

Programsko inženjerstvo

Ak. god. 2020./2021.

Zamjena soba

Dokumentacija, Rev. 2.

Grupa: *PiccoloGrupa*

Voditelj: *Maria Katić*

Datum predaje: *14.1.2021.*

Nastavnik: *Ivana Cepetić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	20
3.2 Ostali zahtjevi	23
4 Arhitektura i dizajn sustava	24
4.1 Baza podataka	25
4.1.1 Opis tablica	26
4.1.2 Dijagram baze podataka	31
4.2 Dijagram razreda	32
4.3 Dijagram stanja	34
4.4 Dijagram aktivnosti	35
4.5 Dijagram komponenti	37
5 Implementacija i korisničko sučelje	38
5.1 Korištene tehnologije i alati	38
5.2 Ispitivanje programskog rješenja	39
5.2.1 Ispitivanje komponenti	39
5.2.2 Ispitivanje sustava	42
5.3 Dijagram razmještaja	46
5.4 Upute za puštanje u pogon	47
6 Zaključak i budući rad	48
Popis literature	49
Dodatak: Prikaz aktivnosti grupe	50

Indeks slika i dijagrama

55

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Katić	31.10.2020.
0.2	Dodan Opis projektnog zadatka	Katić	5.11.2020.
0.3	Arhitektura, dijagram baze podataka, dijagram razreda - treba doraditi.	Ivezić	8.11.2020.
0.4	Specifikacija programske potpore, funkcionalni zahtjevi, ostali zahtjevi	Ružić	10.11.2020.
0.5	Nadopuna - Arhitektura, dijagram baze podataka, dijagram razreda	Ivezić	10.11.2020.
0.6	Nadopuna opisa projektnog zadatka	Katić	11.11.2020.
0.7	Nadopuna specifikacija programske potpore, funkcionalni zahtjevi i ostali zahtjevi	Ružić	11.11.2020.
0.8	Dijagrami obrazaca uporabe, sekvencijski dijagrami	Ružić	12.11.2020.
0.9	Arhitektura, dijagram baze podataka, dijagram razreda - finalna verzija.	Ivezić	12.11.2020.
1.0	Opis projektnog zadatka, specifikacija programske potpore, arhitektura i dizajn sustava (baza podataka, dijagrami razreda)	Ivezić, Katić, Ružić	13.11.2020.
1.1	Izmjena baze podataka	Ivezić	7.1.2021.
1.2	Dodan dijagram komponenti	Katić	9.1.2021.
1.3	Dodani dijagram stanja i dijagram aktivnosti	Ružić	9.1.2021.
1.4	Napisan zaključak i korištene tehnologije i alati	Tolj	9.1.2021.
1.5	Nadopuna Arhitekture	Ivezić	10.1.2021.
1.6	Dodan dijagram razmještaja	Ružić	11.1.2021.
1.7	Izmjena UC dijagrama	Ružić	12.1.2021.
1.8	Napisane upute za puštanje u pogon	Katić	12.1.2021.

Rev.	Opis promjene/dodatka	Autori	Datum
1.9	Dodano ispitivanje komponenti	Bokan	13.1.2021.
1.9.1	Dodano ispitivanje sustava	Katić	14.1.2021.
2.0	Konacna verzija		14.1.2021.

2. Opis projektnog zadatka

Cilj projektnog zadatka je izrada web aplikacije „Zamjena soba“ koja omogućuje studentima koji žive u studentskim domovima da s lakoćom zamijene sobu s nekim drugim zainteresiranim studentom. Trenutno se ovaj problem rješava usmenim dogovorom studenata ili, još češće, preko društvenih mreža. Nakon toga studenti moraju ići u nadležni Studentski centar da im se odobri zamjena, što također komplicira cijeli postupak. Zamjena soba na ovaj način jako je neorganizirana te zbog nemogućnosti pregleda svih soba koje su dostupne za zamjenu ponekad se dogodi da studenti ne uspiju pronaći idealno rješenje.

Aplikacija „Zamjena soba“ predstavlja soluciju za navedene probleme. Prilikom ulaska u aplikaciju otvara se početna stranica na kojoj se nalaze predani oglasi iz različitih gradova i studentskih domova. Neregistrirani korisnik može gledati oglase, ali preduvjet za korištenje bilo koje druge funkcionalnosti je registracija i/ili prijava u sustav. Neregistriranom korisniku omogućeno je prijavljivanje u sustav s postojećim računom ili izradom novog računa (registracija). Ako se korisnik prijavljuje s postojećim računom potrebno je unijeti korisničko ime i lozinku nakon čega sustav provjera oba podatka i prijavljuje korisnika u sustav. Za kreiranje novog računa potrebno je unijeti:

- ime
- email
- JMBAG
- korisničko ime
- lozinku

Korisnici pri registraciji dobivaju ulogu Studenta, a djelatnicima SC-a su korisnički podaci uneseni direktno u bazu podataka te ne trebaju vršiti registraciju u sustav.

Studentu se prilikom prijave u aplikaciju otvara početna stranica na kojoj se nalaze predani oglasi drugih studenata. Student može otići na stranicu „Predaj oglas“ na kojoj navodi značajke svojeg mjesta u domu:

- grad
- dom
- paviljon
- kategorija sobe
- kat
- najbliža menza

te također navodi značajke mjesta u domu koje želi. Prilikom izrade oglasa za vlastito mjesto mora popuniti sve značajke, a za mjesto koje želi može ostaviti jednu ili više značajki prazno (ako mu ta značajka nije bitna). Student sve svoje oglase može vidjeti na stranici „Moji oglasi“. Tamo se nalaze aktivni i neaktivni oglasi, a aktivni oglasi mogu se uređivati (npr. student može promijeniti značajke domskog mjesta koje želi dobiti). U jednom trenutku samo jedan korisnikov oglas može biti aktivan. Nakon što student unese značajke domskog mjesta koje želi, početna stranica mu se mijenja. Prikazuju se oglasi osoba koje žele onu sobu u domu koju je student oglasio te koje nude sobu koja ima značajke koje je student označio. Također na početnoj stranici su dostupne i sobe koje su u tzv. lancima razmjene (sobe za koje se mogu ostvariti zamjene s tri ili više studenata, a da svatko dobije što želi). Kad se pojave novi oglasi koji odgovaraju navedenim kriterijima, osim što će se pojaviti na početnoj stranici, student će također dobiti obavijest mailom: „Imate nove kandidate za zamjenu sobe u domu!“ s linkom za pregled istih. Sve dostupne oglase student može „lajkati“, i to po stupnjevima:

1. sviđa mi se
2. sviđa mi se jako
3. to je to

ili označiti „Ne prikazuj više ovaj oglas“. Kada su obje uključene strane „lajkale“ zamjenu soba, oba studenta bit će obaviještena mailom u kojem će se od njih tražiti da daju konačnu potvrdu zamjene soba. Student će dobiti link u mailu te klikom na taj link otvorit će se stranica za zamjenu na kojoj će student imati mogućnost pregledati zamjenu i prihvatiti željenu sobu. Oba studenta moraju prihvatiti željenu sobu da bi se zamjena dogodila. Nakon što oba studenta prihvate zamjenu, djelatniku SC-a ta zamjena biti će vidljiva i moći će je evidentirati u Studentskom centru.

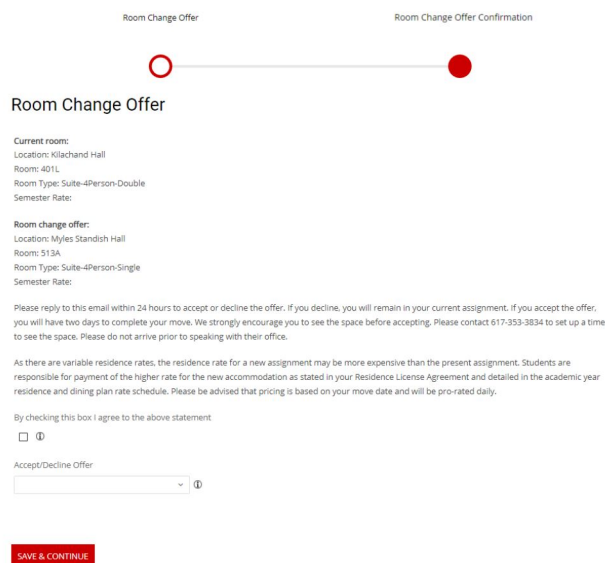
Djelatniku SC-a prilikom prijave u sustav otvaraju se sve zaključane zamjene za studentske domove koji se nalaze u gradu za koji je taj Studentski centar nadležan. Nakon što djelatnik SC-a pogleda neku zamjenu i zabilježi je u sustav SC-a, može je označiti kao obavljenju te se kasnije obavljene zamjene više nigdje ne prikazuju.

Rješenje je trenutno izvedeno samo za državne studentske domove, ali moglo bi se prilagoditi za sve studentske domove (privatne i državne) , privatne sobe i stanove u kojima stanuju studenti.

Projektni zadatak mogao bi se nadograditi na više načina. Jedan od načina je da student prilikom unosa željenih značajki u domu ima mogućnost navođenja 3 ili 5 tipova soba koje ga zanimaju tako da značajke mjesta u domu koje ga najviše zanima stavi na prvo mjesto. Na taj način bi student dobio više izbora prilikom traženja sobe za zamjenu.

Ovaj projekt uvelike bi olakšao i ubrzao zamjenu soba studentima. Neka sveučilišta u svijetu već su razvila aplikacije koje omogućavaju studentima slične mogućnosti. Jedan primjer aplikacije je aplikacija Sveučilišta u Bostonu koja svojim studentima nudi mogućnost oglašavanja vlastite i pronalaska željene sobe. Za razliku od naše aplikacije koja nudi mogućnost korisnicima da sami pregledavaju oglase i označavaju one koje im se sviđaju, aplikacija Sveučilišta u Bostonu nema tu mogućnost. Sustav sam pronalazi moguće zamjene i šalje ih na mail zainteresiranim studentima koji nakon toga trebaju provesti potvrdu.

Slika 2.1 prikazuje stranicu aplikacije za zamjenu soba Sveučilišta u Bostonu na kojoj je prikazana ponuda za zamjenu soba koju je student dužan potvrditi ukoliko želi da se zamjena ostvari. Ova funkcionalnost implementirana je i u našoj aplikaciji.



The form is titled "Room Change Offer Confirmation" and is part of a process shown in a progress bar at the top with two steps: "Room Change Offer" (current) and "Room Change Offer Confirmation".

Room Change Offer

Current room:
Location: Kilachand Hall
Room: 401L
Room Type: Suite-4Person-Double
Semester Rate:

Room change offer:
Location: Myles Standish Hall
Room: 513A
Room Type: Suite-4Person-Single
Semester Rate:

Please reply to this email within 24 hours to accept or decline the offer. If you decline, you will remain in your current assignment. If you accept the offer, you will have two days to complete your move. We strongly encourage you to see the space before accepting. Please contact 617-353-3834 to set up a time to see the space. Please do not arrive prior to speaking with their office.

As there are variable residence rates, the residence rate for a new assignment may be more expensive than the present assignment. Students are responsible for payment of the higher rate for the new accommodation as stated in your Residence License Agreement and detailed in the academic year residence and dining plan rate schedule. Please be advised that pricing is based on your move date and will be pro-rated daily.

By checking this box I agree to the above statement
☐ ⓘ

Accept/Decline Offer
 ⓘ

SAVE & CONTINUE

Slika 2.1: Aplikacija za zamjenu soba Sveučilišta u Bostonu

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Studenti
2. Djelatnici studentskog centra
3. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani klijent (inicijator) može:
 - (a) vidjeti predane oglase
 - (b) registrirati se u sustav, stvoriti novi korisnički račun i kasnije se prijaviti u sustav
2. Registrirani korisnik (student) (inicijator) može:
 - (a) prijaviti se u sustav i odjaviti se iz sustava
 - (b) predati oglas za zamjenu sobe sa značajkama njegove sobe i željama sobe za zamjenu
 - (c) pregledati i uređivati vlastiti profil
 - (d) vidjeti svoje aktivne i neaktivne oglase
 - (e) uređivati svoje aktivne oglase (promijeniti detalje oglasa ili ga učiniti ne-aktivnim)
 - (f) vidjeti oglase drugih korisnika te ih „lajkati“ ili označiti da se više ne prikazuje
 - (g) pregledati oglase koji odgovaraju njegovim kriterijima
 - (h) pregledati zaprimljene ponude za zamjenu
 - (i) potvrditi zamjenu soba
3. Djelatnik studentskog centra (sudionik) može:
 - (a) vidjeti potvrđene (zaključane) zamjene između studenata

- (b) označiti potvrđene zamjene koje je unio u sustav studentskog centra kao obavljene

4. Baza podataka (sudionik) može:

- (a) pohranjuje sve podatke o korisnicima
- (b) pohranjuje sve podatke o predanim oglasima
- (c) pohranjuje sve podatke o zamjenama

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregledaj oglase

- **Glavni sudionik:** neregistrirani klijent, registrirani korisnik
- **Cilj:** pregled ponude predanih oglasa
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. neregistrirani klijent/registrirani korisnik otvara aplikaciju
 2. sustav prikazuje predane oglase
 3. neregistrirani klijent/registrirani korisnik listanjem aplikacije („scrollanjem“) prema dolje može vidjeti još oglasa

UC2 - Registriraj korisnika

- **Glavni sudionik:** neregistrirani klijent
- **Cilj:** stvaranje korisničkog računa koji omogućuje prijavu u sustav i predaju oglasa
- **Sudionici:** baza podataka
- **Preduvjet:** ispravna e-mail adresa i važeći JMBAG
- **Opis osnovnog tijeka:**
 1. neregistrirani klijent u izborniku koji se nalazi u gornjem desnom kutu aplikacije odabire opciju za registraciju („Registriraj se“)
 2. sustav neregistriranog klijenta vodi na stranicu za registraciju na kojoj traži njegove korisničke podatke
 3. klijent unosi korisničko ime, e-mail adresu, JMBAG, ime i prezime te lozinku
 4. klijent odabire opciju za registraciju („Registriraj se“) koja se nalazi ispod unešenih podataka
 5. sustav prikazuje prikladnu poruku o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 3.a odabrano korisničko ime je zauzeto ili je već postoji račun sa istom e-mail adresom
 1. sustav prikazuje prikladnu poruku „Korisničko ime je zauzeto“ ili „Već postoji račun sa ovom e-mail adresom“

3.b korisnik nije unio sve potrebne podatke za registraciju (korisničko ime, e-mail adresu, JMBAG, ime i prezime ili lozinku)

1. sustav prikazuje prikladnu poruku o potrebi za ispunjavanjem svih polja o korisničkim podacima

UC3 - Prijavi korisnika

- **Glavni sudionik:** registrirani korisnik (student ili djelatnik studentskog centra)
- **Cilj:** pristupanje korisničkom sučelju, predaja oglasa
- **Sudionici:** baza podataka
- **Preduvjet:** registracija
- **Opis osnovnog tijeka:**
 1. registrirani korisnik u izborniku u gornjem desnom kutu aplikacije odabire opciju za prijavu („Prijavi se“)
 2. sustav korisnika vodi na stranicu za prijavu na kojoj traži korisničko ime i lozinku te mu ispod podataka koje traži nudi gumb za prijavu („Prijavi se“)
 3. korisnik upisuje tražene podatke te odabire opciju za prijavu („Prijavi se“)
 4. sustav obavještava korisnika o uspješnoj prijavi te mu dozvoljava pristup korisničkom sučelju
- **Opis mogućih odstupanja:**
 - 4.a neispravan unos korisničkog imena ili lozinke
 1. sustav obavještava korisnika o neuspjeloj prijavi te mu nudi ponovni pokušaj

UC4 - Odjavi korisnika

- **Glavni sudionik:** registrirani korisnik (student ili djelatnik studentskog centra)
- **Cilj:** odjava iz korisničkog sučelja
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava
- **Opis osnovnog tijeka:**
 1. registrirani korisnik u izborniku u gornjem desnom kutu aplikacije odabire opciju za odjavljivanje („Odjava“)

2. sustav odjavljuje korisnika te ga vodi na početnu stranicu gdje može vidjeti oglase svih korisnika, a u gornjem desnom kutu aplikacije, u izborniku ima opcije „Početna“, „Prijava se“ i „Registriraj se“

UC5 - Predaj oglas

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** predaja vlastitog oglasa u sustav
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava
- **Opis osnovnog tijeka:**
 1. registrirani korisnik odabire opciju za pregled vlastitih oglasa („Moji oglasi“) koja se nalazi u izborniku u gornjem desnom kutu aplikacije
 2. sustav vodi korisnika na stranicu na kojoj se nalaze svi neaktivni oglasi korisnika te mu na gornjem dijelu ekrana nudi opciju za stvaranje novog oglasa
 3. korisnik odabire opciju za stvaranje novog oglasa
 4. sustav vodi korisnika na stranicu za predaju novog oglasa, na kojoj od njega traži značajke (grad, dom, paviljon, kategorija sobe, kat, najbliža menza) njegove sobe u domu (ispunjavanje svih podataka o sobi koju korisnik nudi je obavezno)
 5. korisnik ispunjuje tražene podatke te odabire opciju „Predaj oglas“ koja se nalazi ispod kućica za upis traženih podataka
 6. sustav korisniku ispisuje poruku o uspješnoj predaji oglasa te ga vraća na stranicu „Moji oglasi“
 7. ukoliko je korisnik imao aktivan oglas prilikom predaje novog oglasa, on predajom novog oglasa automatski postaje neaktivan jer jedan korisnik može imati najviše jedan aktivan oglas istovremeno
 8. korisnik zatim vidi svoj novi aktivan oglas i sve neaktivne oglase
- **Opis mogućih odstupanja:**
 - 6.a korisnik nije unio sve obavezne podatke (grad, dom, paviljon, kategorija sobe, kat, najbliža menza) o sobi koju nudi
 1. korisnik ne može predati oglas dok ne ispuni sve obavezne podatke
 2. sustav ispisuje odgovarajuću obavijest

UC6 - Pregledaj vlastite oglase

- **Glavni sudionik:** registrirani korisnik

- **Cilj:** pregled vlastitih predanih oglasa
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava
- **Opis osnovnog tijeka:**
 1. registrirani korisnik odabire opciju za pregled vlastitih oglasa koja se nalazi u izborniku u gornjem desnom kutu aplikacije („Moji oglasi“)
 2. sustav vodi korisnika na stranicu na kojoj se nalaze svi neaktivni oglasi korisnika, aktivan oglas (ako takav postoji) te opcija za stvaranje novog oglasa
 3. ukoliko korisnik ima više oglasa može ih pogledati listanjem aplikacije („scrollanjem“) prema dolje
 4. korisnik se može vratiti na početnu stranicu klikom na „Početna“ u izborniku u gornjem desnom kutu aplikacije
- **Opis mogućih odstupanja:**
 - 2.a nepostojanje oglasa
 1. sustav ispisuje prikladnu poruku o nepostojanju oglasa
 2. sustav nudi korisniku da stvori oglas

UC7 - Uredi oglas

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** promjena detalja oglasa
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava, postojanje oglasa
- **Opis osnovnog tijeka:**
 1. registrirani korisnik odabire opciju za pregled vlastitih oglasa koja se nalazi u izborniku u gornjem desnom kutu aplikacije („Moji oglasi“)
 2. sustav vodi korisnika na stranicu na kojoj se nalazi aktivan oglas ukoliko postoji i svi neaktivni oglasi korisnika
 3. korisnik odabire opciju za uređivanje oglasa u donjem desnom kutu oglasa kojeg želi urediti („uredi“)
 4. sustav korisnika vodi na stranicu „Uredi oglas“ na kojoj korisnik može promijeniti bilo koji detalj vezan za oglas
 5. korisnik po želji mijenja podatke o oglasu
 6. sustav na dnu stranice za uređivanje oglasa nudi korisniku opcije „Spremi promjene“
 7. korisnik odabirom opcije „Spremi promjene“ mijenja detalje oglasa

8. nakon odabira opcije „Spremi promjene“ ili odabira izlaska iz stranice za uređivanje oglasa (X u gornjem desnom kutu), sustav vraća korisnika na stranicu „Moji oglasi“ gdje se sada nalazi izmjenjen oglas(ukoliko je korisnik odabrao opciju „Spremi promjene“), ili oglas koji je isti kao i prije (ukoliko je korisnik odabrao izaći bez promjena)
9. korisnik se može vratiti na početnu stranicu klikom na „Početna“ u izborniku u gornjem desnom kutu aplikacije

UC8 - Uredi profil

- **Glavni sudionik:** registrirani korisnik (student ili djelatnik studentskog centra)
- **Cilj:** promjena detalja profila
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava
- **Opis osnovnog tijeka:**
 1. korisnik odabire opciju „Uredi profil“ u izborniku u gornjem desnom kutu aplikacije
 2. sustav prikazuje korisniku njegove korisničke podatke (korisničko ime, ime i prezime, JMBAG, e-mail adresu)
 3. korisnik odabire opciju „uredi“ koja se nalazi ispod svih podataka
 4. sustav omogućuje korisniku promjenu bilo kojeg od podataka ili promjenu lozinke
 5. korisnik mijenja željene podatke
 6. sustav korisniku nudi opcije „Ažuriraj podatke“ i „Ažuriraj lozinku“
 7. korisnik odabire jednu od ponuđenih opcija
 8. sustav ponovno prikazuje korisniku njegove korisničke podatke koji su izmjenjeni ili isti(ukoliko je mijenjao lozinku), ovisno o korisnikovom ranijem odabiru
 9. korisnik se može vratiti na početnu stranicu klikom na „Početna“ u izborniku u gornjem desnom kutu aplikacije

UC9 - Reagiraj na oglas

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** izraziti želju za zamjenom sobe s korisnikom koji je dao oglas ili spriječiti ponovno pojavljivanje određenog oglasa
- **Sudionici:** baza podataka

- **Preduvjet:** registracija, prijava, postojanje aktivnog oglasa
- **Opis osnovnog tijeka:**
 1. registrirani korisnik želi reagirati na određeni oglas
 2. korisnik odabire jednu od opcija za reagiranje na oglas koje su prikazane simbolom 1, 2 ili 3 srca, tj. kantom za smeće
 3. opcije koje su prikazane simbolima srca označuju stupanj svidanja (1-malo, 3-jako), dok opcija koja je prikazana simbolom kante za smeće označuje da korisnik ne želi da mu se taj oglas više prikazuje
 4. ako je oglas „lajkan“ sustav šalje e-mail korisniku čiji je oglas „lajkan“ koji sadrži poruku „Imate nove kandidate za zamjenu sobe u domu!“ te link na stranicu „Dobiveni lajkovi“ koja se nalazi u izborniku u gornjem desnom kutu aplikacije gdje korisnik može vidjeti aktivne oglase svih korisnika koji su „lajkali“ njegov aktivan oglas

UC10 - Pregledaj oglase koji odgovaraju kriterijima

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** pronalazak sobe za zamjenu
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava, postojanje aktivnog oglasa
- **Opis osnovnog tijeka:**
 1. korisnik na početnoj stranici odabire opciju „Filtracija oglasa“
 2. korisnik unosi podatke o željenoj sobi te odabire opciju „Pretraži“
 3. korisnik također može odabrati i opcije „Spremi filter“ koja mu omogućuje spremanje unešenog filtera i „Prikaži spremljene filtere“ koja mu omogućuje prikaz do sada spremljenih oglasa
 4. korisnik može reagirati na bilo koji od oglasa koji su dobiveni filtriranjem

UC11 - Pregledaj zaprimljene ponude za zamjenu

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** pregled oglasa onih korisnika koji žele zamjeniti sobu sa korisnikom
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava, postojanje aktivnog oglasa
- **Opis osnovnog tijeka:**
 1. korisnik odabire opciju „Dobiveni lajkovi“ koja se nalazi u izborniku u gornjem desnom kutu aplikacije

2. sustav vodi korisnika na stranicu na kojoj mu prikazuje oglase studenata koji su „lajkali“ njegov oglas
 3. korisnik pregledava zaprimljene ponude za zamjenu te po želji reagira na njih
- **Opis mogućih odstupanja:**
 - 2.a nema novih kandidata za zamjenu
 1. sustav ispisuje prikladnu poruku

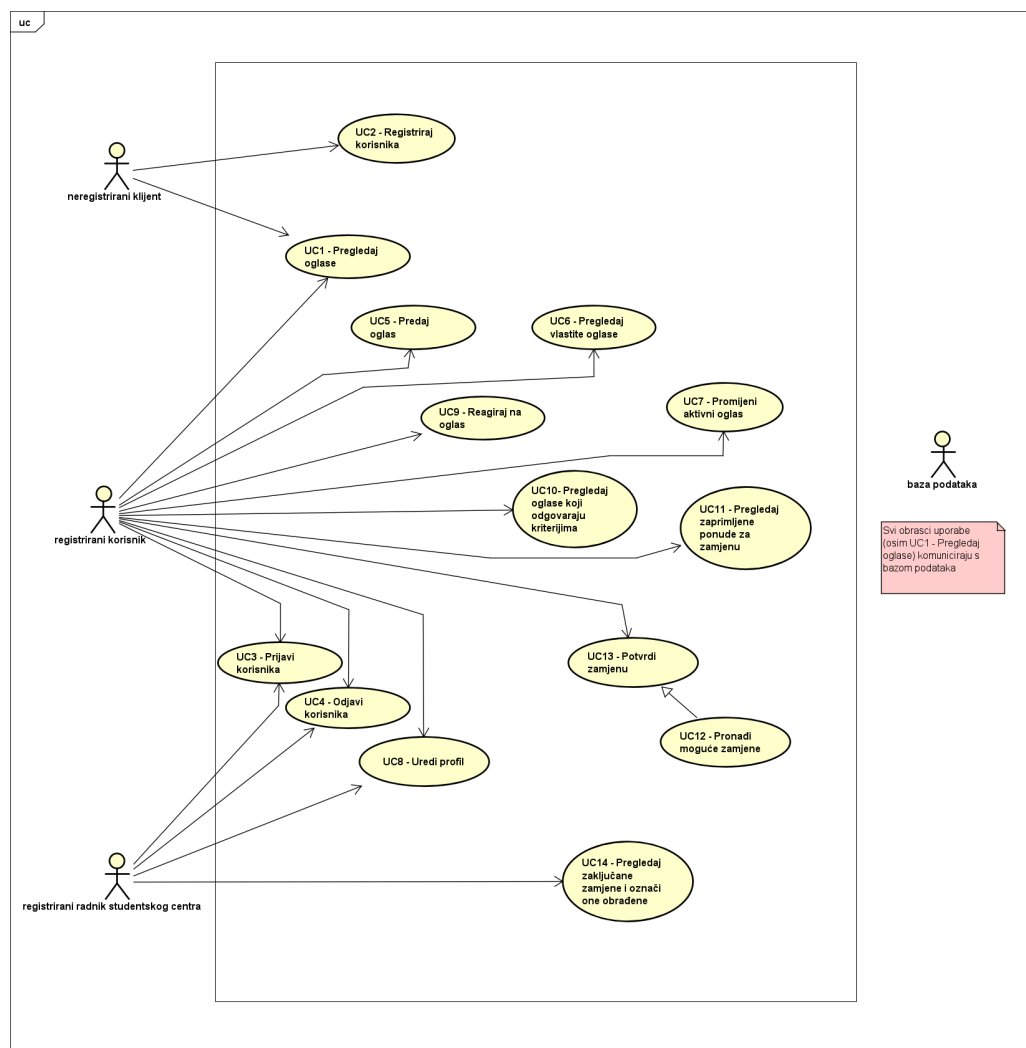
UC12 - Potvrdi zamjenu

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** konačni odabir zamjene soba
- **Sudionici:** baza podataka
- **Preduvjet:** registracija, prijava, postojanje aktivnog oglasa, međusobno „lajkanje“ oglasa dvoje korisnika
- **Opis osnovnog tijeka:**
 1. korisnik odlazi na stranicu „Moguće zamjene“ koja se nalazi u izborniku u gornjem desnom kutu aplikacije
 2. korisnik dolazi na stranicu na kojoj se prikazuje oglas ponuđene sobe za potvrdu zamjene (oglas može ponovno pogledati) te se ispod njega nalaze opcije „Potvrdi zamjenu“ i „Odbij zamjenu“
 3. korisnik potvrđuje zamjenu odabirom opcije „Potvrdi zamjenu“ ili ju odbija odabirom opcije „Odbij zamjenu“
 4. ukoliko svi korisnici uključeni u zamjenu potvrde zamjenu, djelatnik studentskog centra dobiva podatke o zamjeni, a ukoliko barem jedan od korisnika uključenih u zamjenu odbije zamjenu, svi uključeni u zamjenu dobivaju e-mail o neuspjelom ostvarenju zamjene
- **Opis mogućih odstupanja:**
 - 3.a svi korisnici uključeni u zamjenu nisu potvrdili niti odbili zamjenu
 1. sustav nakon 7 dana briše ponudu za potvrdu ili odbijanje zamjene te se ona automatski smatra neostvarenom
 2. sustav šalje svim korisnicima uključenim u zamjenu e-mail o neuspjelom ostvarenju zamjene
 - 3.b jedan ili više korisnika uključenih u zamjenu je učinio svoj oglas neaktivnim, a drugi korisnik (korisnici) su potvrdili zamjenu
 1. sustav šalje svim korisnicima uključenim u zamjenu e-mail o neuspjelom ostvarenju zamjene

UC13 - Pregledaj zaključane zamjene i označi one obrađene

- **Glavni sudionik:** radnik studentskog centra
- **Cilj:** potvrda zamjene soba u sustavu studentskog centra i označavanje obrađenih zamjena
- **Sudionici:** baza podataka
- **Preduvjet:** prijava, postojanje zaključenih zamjena
- **Opis osnovnog tijeka:**
 1. radnik studentskog centra odabire opciju „Prikaži zaključane zamjene“ koja se nalazi u izborniku u gornjem desnom kutu aplikacije
 2. sustav radniku studentskog centra prikazuje jednu po jednu zaključanu zamjenu te ispod svake zamjene nudi opciju „Zamjena obrađena“
 3. radnik studentskog centra unosi zaključanu zamjenu u sustav studentskog centra te odabire opciju „Zamjena obrađena“ kako bi označio obrađene zamjene
 4. sustav unosi podatke o obrađenoj zamjeni u bazu podataka, korisnicima kojima su sobe uspješno zamjenjene šalje prikladan e-mail te obrađenu zamjenu više ne prikazuje radniku studentskog centra
 5. sustav radniku studentskog centra prikazuje iduću zaključanu zamjenu

Dijagram obrazaca uporabe

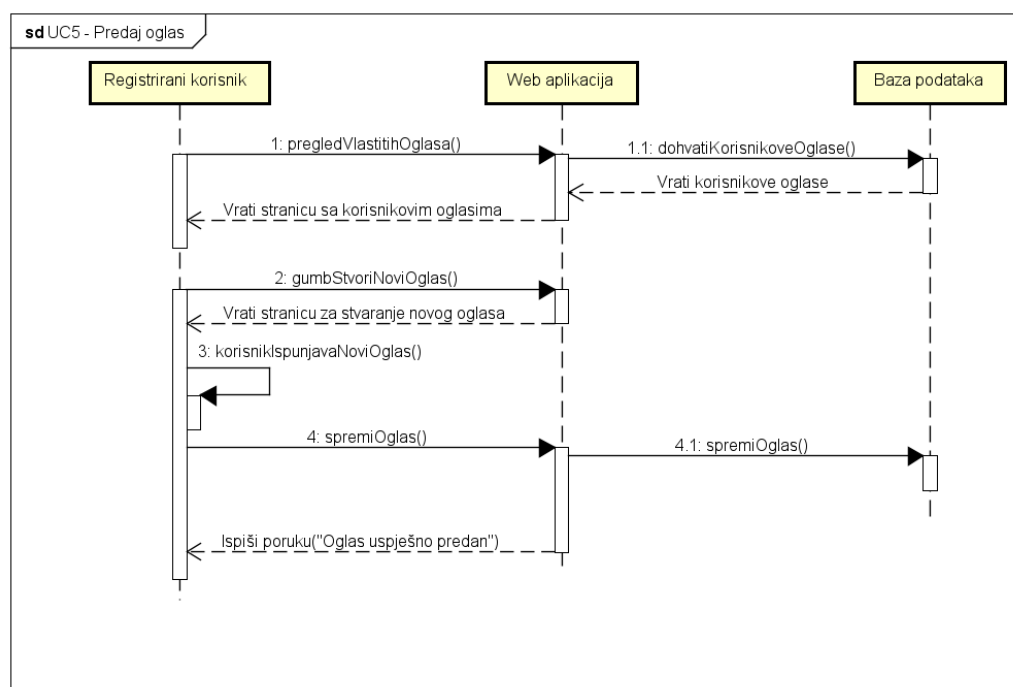


Slika 3.1: Dijagram obrazaca uporabe

3.1.2 Sekvencijski dijagrami

UC5 - Predaj oglas

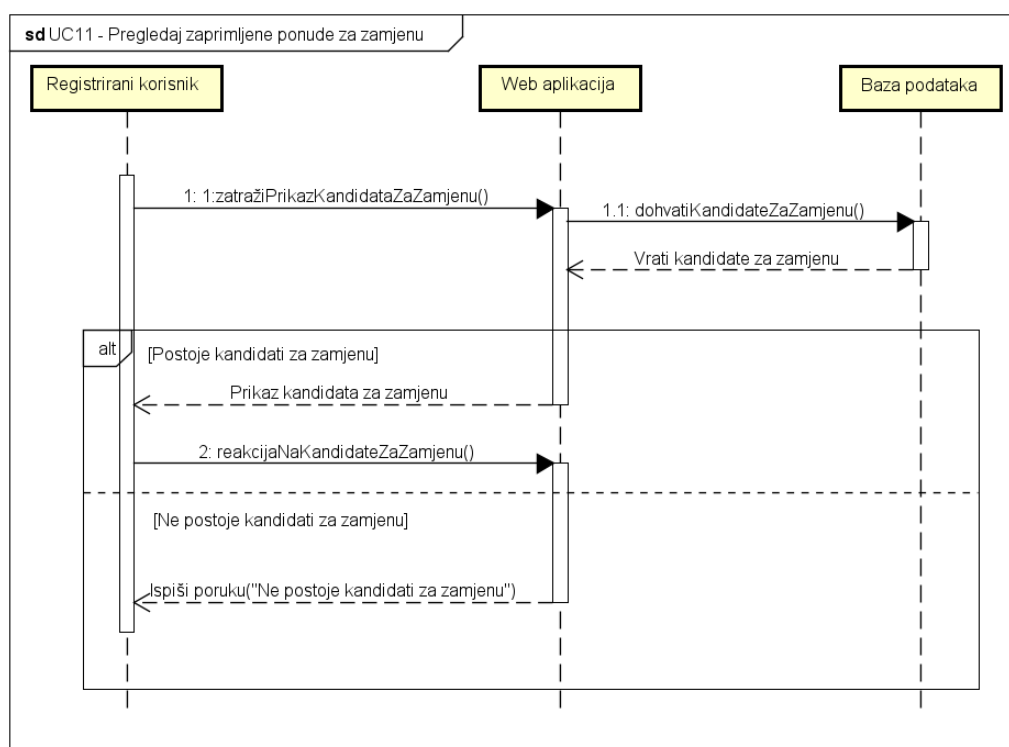
Registrirani korisnik šalje zahtjev za pregled vlastitih oglasa. Poslužitelj dohvaća korisnikove oglase iz baze podataka te ih prikazuje korisniku. Korisnik šalje zahtjev za stvaranje novog oglasa a poslužitelj mu vraća stranicu za stvaranje novog oglasa. Korisnik ispunjava novi oglas te šalje zahtjev za spremanje oglasa. Poslužitelj sprema oglas u bazu podataka i korisniku ispisuje poruku „Oglas uspješno predan“.



Slika 3.2: Sekvencijski dijagram za UC5

UC11 - Pregledaj zaprimljene ponude za zamjenu

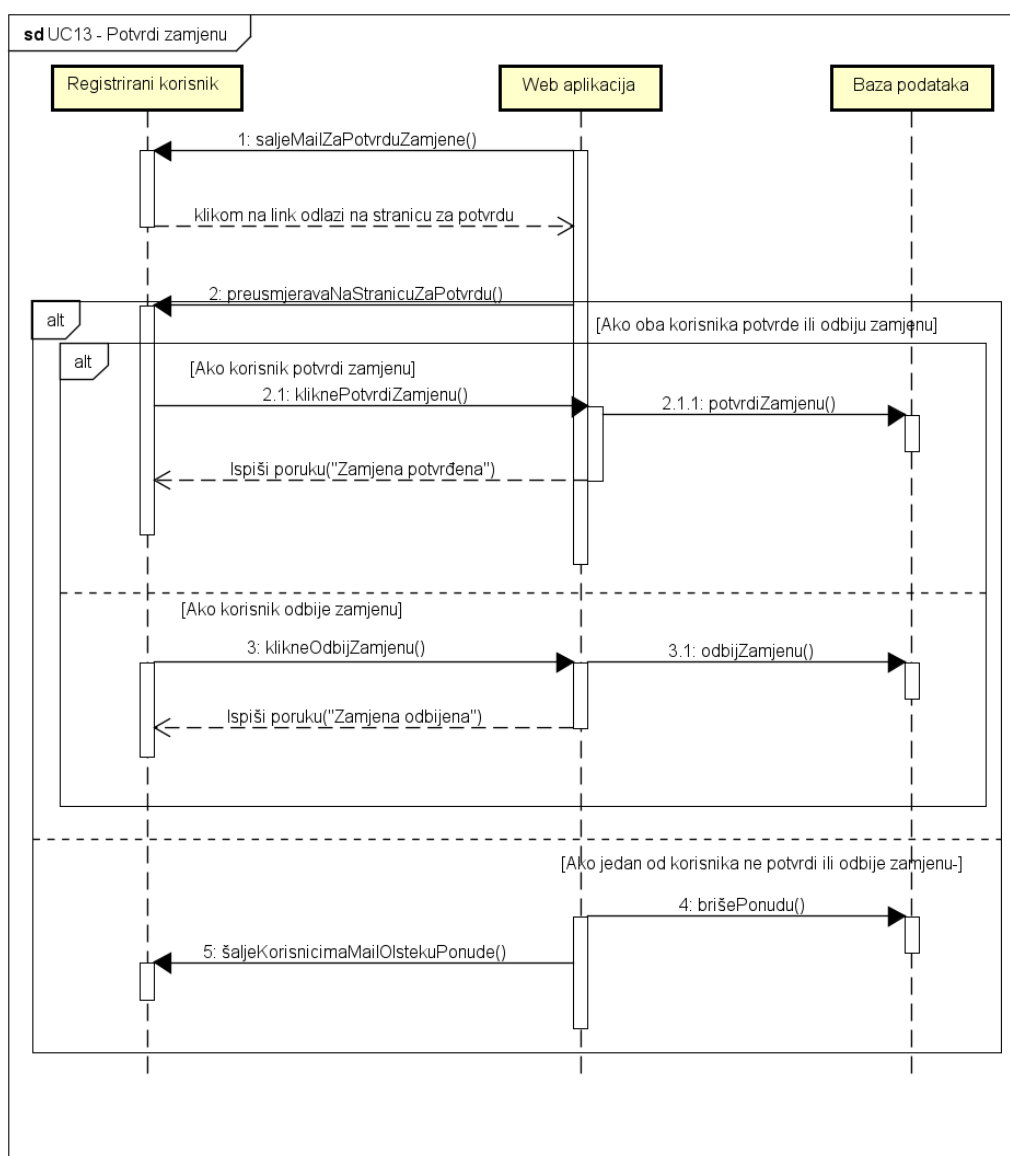
Registrirani korisnik šalje zahtjev za prikaz kandidata za zamjenu kako bi vidio oglase korisnika koji su „lajkali“ njegov oglas i koji bi se htjeli zamjeniti za sobe sa njime. Poslužitelj dohvaća kandidate za zamjenu iz baze podataka i prikazuje ih korisniku. Ukoliko ne postoje kandidati za zamjenu poslužitelj prikazuje poruku „Ne postoje kandidati za zamjenu“.



Slika 3.3: Sekvencijski dijagram za UC11

UC13 - Potvrdi zamjenu

Poslužitelj šalje korisniku e-mail za potvrdu zamjene. Registrirani korisnik klikne na link (odabere ga) te dolazi na web aplikaciju koja ga preusmjerava na stranicu za potvrdu. Korisnik potvrđuje ili odbija zamjenu, poslužitelj sprema promjenu u bazu podataka, te ako svi korisnici uključeni u zamjenu potvrde zamjenu poslužitelj ispisuje poruku „Zamjena potvrđena“, a ako jedan ili više korisnika uključenih u zamjenu odbiju zamjenu poslužitelj svim korisnicima uključenim u zamjenu ispisuje poruku „Zamjena odbijena“. Ukoliko prođe više od 7 dana te netko od korisnika ne potvrdi ili odbije zamjenu, poslužitelj briše podatke o zamjeni iz baze podataka te korisnicima šalje e-mail o isteku ponude za zamjenu.



Slika 3.4: Sekvencijski dijagram za UC13

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Aplikacija treba biti prilagođena (engl. responsive) mobilnom uređaju
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) prilikom unosa i prikaza tekstualnog sadržaja
- Izvršavanje dijela programa u kojemu se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže

4. Arhitektura i dizajn sustava

Pri razvoju web aplikacije Zamjena soba koristi se stil višeslojne arhitekture (engl. multi-layer architecture). To znači da uključuje više logičkih slojeva implementacije programske potpore. Općenito, osnova svih višeslojnih stilova je troslojni stil arhitekture koji sadrži:

- korisnički ili prezentacijski sloj (engl. presentation layer) – sloj s korisničkim sučeljem, odnosno onaj koji korisnik vidi i koji mu služi za unos podataka; uneseni podaci se mogu pohraniti u bazu ili u datotečni sustav
- sloj poslovne logike (engl. logic layer) – sloj s implementacijom poslovnih procesa i izračuna; pomoću njega se odvija sva komunikacija s bazom; on obrađuje i generira dinamički sadržaj
- podatkovni sloj (engl. data layer) – sloj za pohranu podataka u bazu ili u datotečni sustav.

No, klijentsku i poslužiteljsku stranu moguće je organizirati u više slojeva, pri čemu svaki sloj pruža uslugu nekom drugom. U slučaju naše aplikacije, u kojoj koristimo radni okvir Spring Boot, bilo je potrebno dodati više slojeva, te su slojevi arhitekture naše aplikacije:

- sloj korisničke strane – implementiran u JavaScriptu, koristi knjižnicu React koja omogućuje prikaz korisničkog sučelja. Pisan je u radnoj okolini Microsoft Visual Studio Code.
- sloj nadglednika (engl. controller) – povezuje korisničku stranu s poslužiteljskom stranom.
- sloj usluge (engl. service) – obavlja svu poslovnu logiku i potrebne izračune.
- sloj domene (engl. domain) – ima razrađeni model podataka domene
- sloj za pristup podacima (engl. data access object, DAO) – omogućuje spremanje i dohvat podataka iz baze podataka te razmjenu tih podataka sa slojem domene

- sloj baze podataka – omogućuje pohranu podataka u relacijsku bazu PostgreSQL

Svi slojevi osim prvog i zadnjeg, pisani su na radnoj platformi Eclipse u programskom jeziku Java.

Primijetimo da slojevi naglednika, usluge i domene su analogni MVC arhitekturi. Odnosno sloj domene (domain) predstavlja model, sloj usluge (service) predstavlja view, te sloj naglednika (controller) predstavlja controller.

Izabrali smo baš tu arhitekturu kako bismo mogli bolje podijeliti rad među članovima tima (princip podijeli pa vladaj).

Nakon stavljanja aplikacije na server pomoću platforme Heroku, web poslužitelj je taj koji isporučuje sadržaj web aplikacije. To izvršava preko HTTP protokola na portu 8080.

Komunikacija među slojevima izvršava se u pozadini putem određenih portova (vrata). Sloj korisničke strane povezan je sa slojem nadglednika preko aplikacijskog programskog sučelja (API) na portu 3000. Sloj nadglednika, usluge i domene te sloj za pristup podacima su zajedno integrirani, odnosno povezuje ih radni okvir Spring Boot. Naposljetku, sloj za pristup podacima je povezan sa slojem baze podataka preko porta 5432.

4.1 Baza podataka

Koristi se relacijska baza podataka, ona se sastoji od skupa povezanih tablica odnosno relacija. Izabrali smo DBMS (database management system) PostgreSQL. Entiteti baze su:

- *student*
- *sc*
- *dom*
- *menza*
- *soba*
- *filter*
- *grad*

- *oglas*
- *lajkani_oglas*
- *moguca_zamjena*
- *zakljucane_zamjene*

4.1.1 Opis tablica

Student

- u ovoj tablici se nalazi popis svih studenata i djelatnika studentskih centara.

Primarni ključ: id

student		
id	BIGINT	svakom korisniku se automatski generira id (identifikacijski broj)
username	VARCHAR	korisnik ga sam odabire, jedinstven je
email	VARCHAR	email korisnika, jedinstven je
JMBAG	VARCHAR	JMBAG korisnika
name	VARCHAR	ime i prezime korisnika
user_role	INT	sustav dodjeljuje ovlasti kako bi razlikovao studente od SC djelatnika
password	VARCHAR	hash lozinke
locked	BOOLEAN	osposobljenost ili neosposobljenost korisnikovog računa
enabled	BOOLEAN	aktiviranost ili deaktiviranost korisnikovog računa
potvrđeni_lanac	BIGINT	id lanca u kojem je potvrđen, dog ga nema vrijednost mu je null

Studentski centar

Primarni ključ: idSC

Strani ključ: grad_id_grada

SC		
idSC	BGIINT	svakom studentskom centru se automatski dodjeljuje id
ime	VARCHAR	naziv studentskog centra
grad_id_grada	INT	grad u kojem se nalazi studentski centar

Dom

Primarni ključ: *id_dom*

Strani ključ: *menza_naziv_menze, sc_idsc, grad_id_grada*

dom		
id_dom	BIGINT	svakom studentskom domu se automatski dodjeljuje id
ime;c doma	VARCHAR	naziv doma
grad_id_grada	INT	grad u kojem se dom nalazi
najbliza_menza_naziv_menze	VARCHAR	naziv najblize menze
sc_idSC	BIGINT	id studentskog centra u kojem se nalazi

Menza

- najbliža menza studentskom domu

Primarni ključ: *naziv_menze*

Strani ključ: *grad_id_grada*

menza		
naziv_menze	VARCHAR	naziv menze
grad_id_grada	INT	id grada u kojem se nalazi

Soba

- studentova sobe

Primarni ključ: *id_soba*

Strani ključ: *id_doma_id_dom, id_studenta_id*

soba		
id_soba	BIGINT	svakoj sobi se automatski dodjeljuje jedinstveni id
kat	SMALLINT	kat na kojem se nalazi
kategorija_sobe	SMALLINT	kategorija sobe studentskog doma (od 1 do 7)
paviljon	SMALLINT	paviljon u kojem se nalazi
id_doma_id_dom	BIGINT	id doma u kojem se nalazi
id_studenta_id	BIGINT	id studenta koji raspolaže tom sobom

Filter

- služi za filtriranje oglasa na početnoj stranici

Primarni ključ: id_filter

Strani ključ: id_doma_id_dom, id_studenta_id

filter		
id_filter	BIGINT	svakome filteru se automatski dodjeljuje jedinstveni id
kat	SMALLINT	kat koji se pretražuje
kategorija_sobe	SMALLINT	kategorija sobe koja se pretražuje
paviljon	SMALLINT	paviljon koji se pretražuje
id_doma_id_dom	BIGINT	id doma koji se pretražuje
student_id	BIGINT	id studenta koji koji je napravio filter

Grad

Primarni ključ: id_grada

grad		
id_grada	INT	svakome gradu se automatski dodjeljuje jedinstveni id
ime_grada	VARCHAR	naziv grada

Oglas

- oglas kojime student nudi svoju sobu

Primarni ključ: id_oglas

Strani ključ: dom_id_dom, student_id

oglas		
id_oglas	BIGINT	svakome oglasu se automatski dodjeljuje jedinstveni id
aktivan	BOOLEAN	student može deaktivirati aktivan oglas
kat	SMALLINT	kat na kojem se nalazi studentova soba
kategorija_sobe	SMALLINT	kategorija studentove sobe
paviljon	SMALLINT	paviljon u kojem se nalazi studentova soba
dom_id_dom	BIGINT	id doma u kojem se nalazi studentova soba
student_id	BIGINT	id studenta koji je sastavio oglas

Lajkani oglas

- lajkovi koji povezuju studenta i oglas koji je on lajkao

Primarni ključ: *oglas_id, student_id*

lajkani_oglas		
oglas_id	BIGINT	id lajkanog oglasa
student_id	BIGINT	id studenta koji je lajkao oglas
stupanj_lajkanja	INT	Može biti: 1, 2 ili 3 ovisno o pritisutom broju srca. Također može biti 0 za uklanjanje lajka te 4 za opciju "ne prikazuj više"
procitano	BOOLEAN	Služi za obavješćavanje autora oglasa o novim dobivenim lajkovima

Moguća zamjena

- služi za prikaz ostvarivih zamjena korisniku

Primarni ključ: *id_lanca, redni_broj*

Strani ključ: *oglas_id_oglas*

moguca_zamjena		
id_lanca	BIGINT	svakome lancu zamjena se automatski dodjeljuje jedinstveni id
redni_broj	SMALLINT	Redni broj oglasa u lancu zamjena
oglas_id_oglas	BIGINT	id oglasa studentovog oglasa u lancu zamjena
procitano	BOOLEAN	Služi za obavješćavanje autora oglasa o novim mogućim zamjenama

Zaključane zamjene

- potvrđene zamjene soba od strane studenata, koje se prikazuju djelatniku studentskog centra

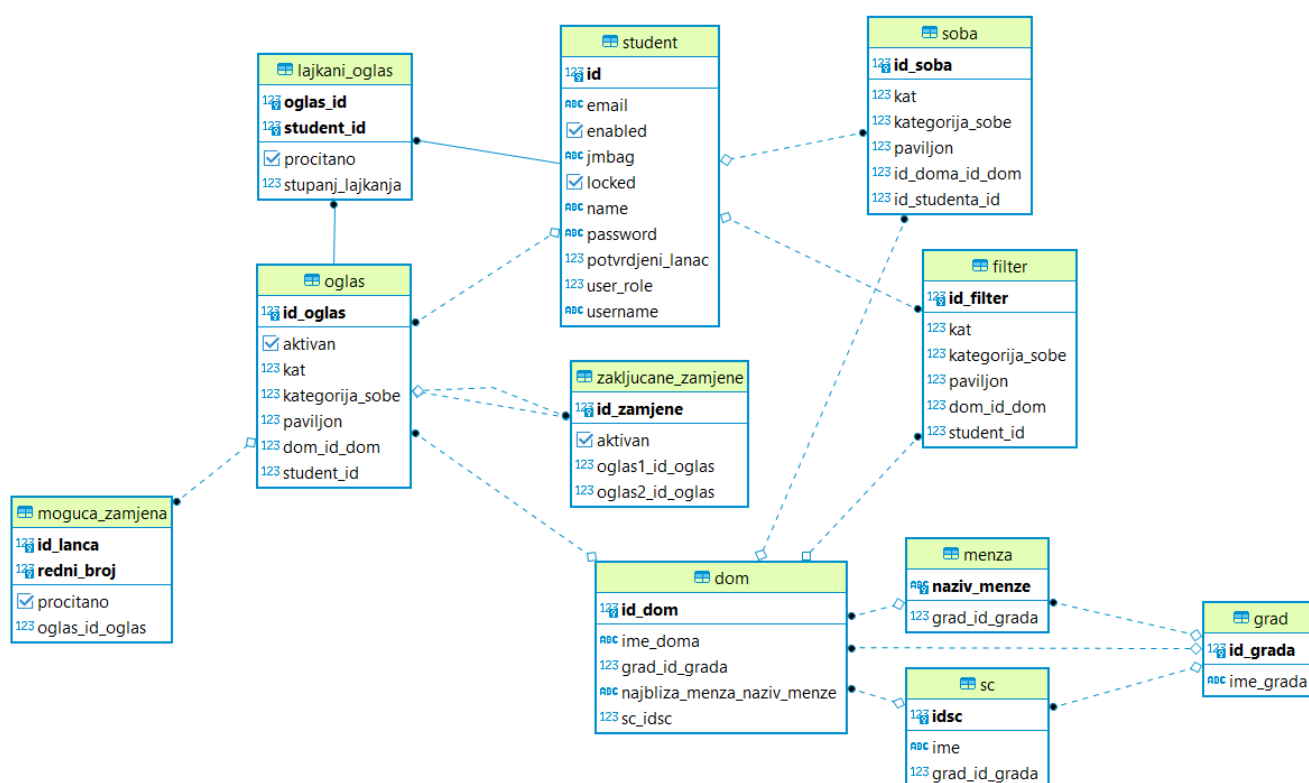
Primarni ključ: id_zamjene

Strani ključ: oglas1_id_oglas, oglas2_id_oglas

zakljucane_zamjene		
id_zamjene	BIGINT	svakoj zaključanoj zamjeni se automatski dodjeljuje jedinstveni id
aktivan	BOOLEAN	izvršenost zamjene
oglas1_id_oglas	BIGINT	id oglasa 1 u zamjeni
oglas2_id_oglas	BIGINT	id oglasa 2 u zamjeni

4.1.2 Dijagram baze podataka

Priložen je ER-model (entity relationship model) baze podataka. Strani ključevi su prikazani crtom među tablicama. Plava točka označava da u toj tablici se nalazi strani ključ koji je u povezanoj tablici primarni ključ.

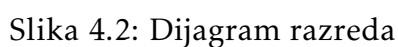


Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

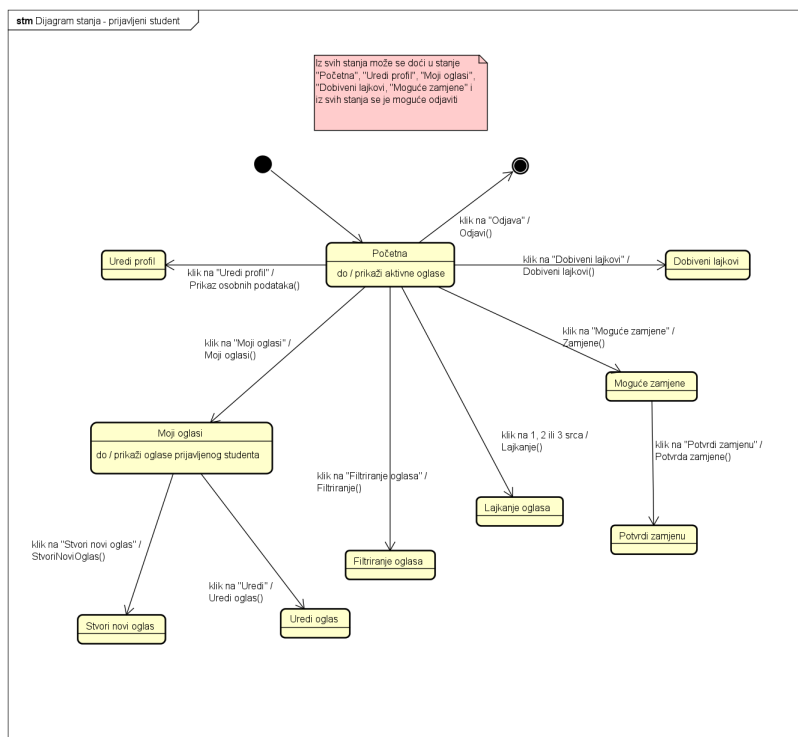
Na slici 4.1 je prikazan potupuni dijagram razreda. Stavke su grupirane prema odgovarajućem paketu unutar kojeg se nalaze. Zbog lakšeg prikaza, kružićima su označena sučelja.

JWT (JSON Web Token) služi za identifikaciju korisnika. On dodjeljuje token svakom korisniku. Preko toga se svi korisnikovi zahtjevi mogu ispuniti u web aplikaciji.



4.3 Dijagram stanja

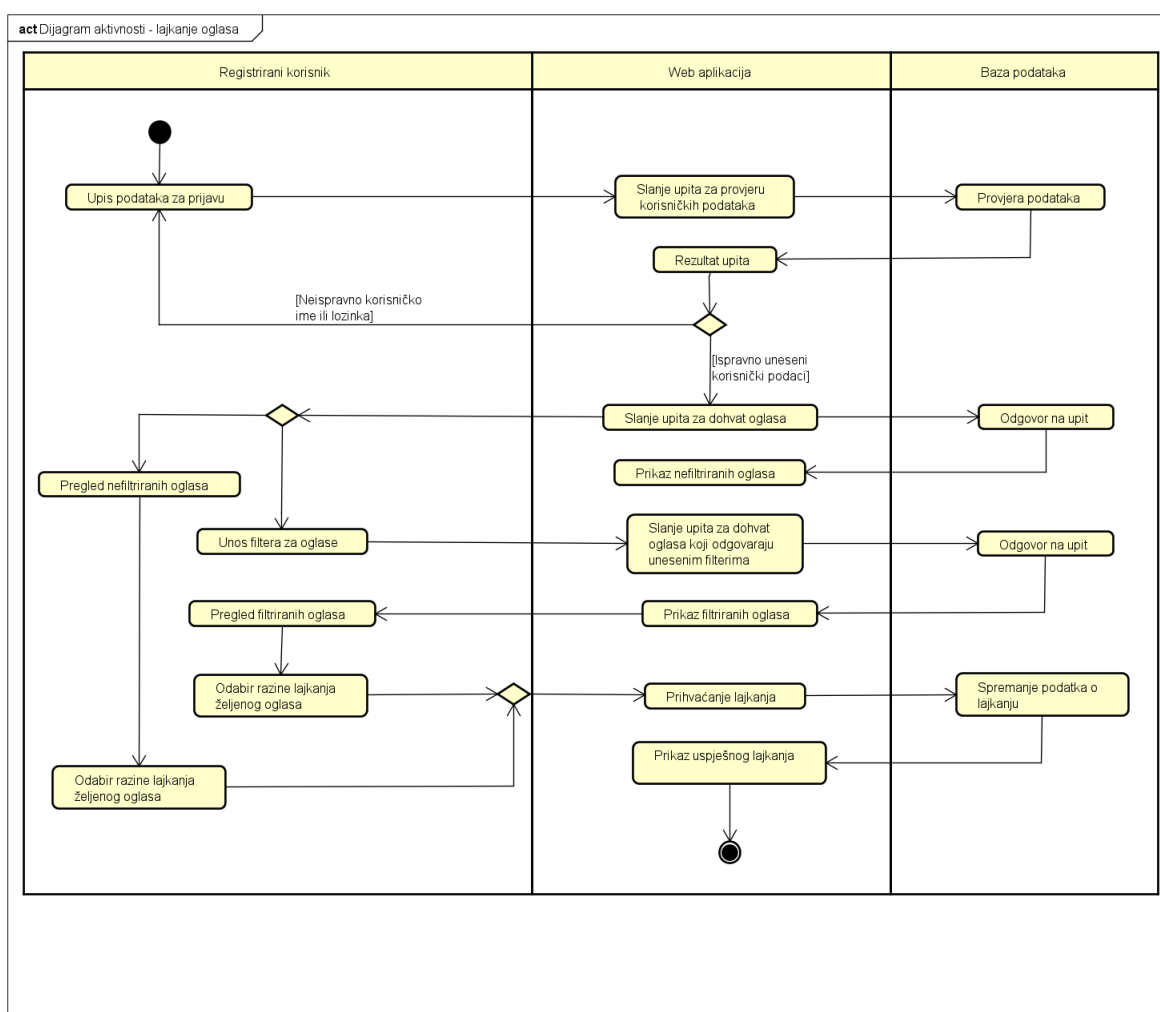
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.6 prikazan je dijagram stanja za registriranog korisnika. Nakon prijave, klijentu se prikazuje početna stranica na kojoj može pregledati oglase te poželji filtrirati oglase. Odabrani oglas korisnik može i lajkati. Korisnik može odabirom različitih opcija i urediti vlastiti profil, pregledati predane oglase, stvoriti novi oglas i urediti predani oglas, pregledati oglase osoba koji su lajkali korisnikov oglas (odabirom opcije "Dobiveni lajkovi"). Ukoliko korisnik odabere opciju "Moguće zamjene", može vidjeti sve oglase koji mu se nude za zaključavanje zamjene te zaključati zamjenu.



Slika 4.3: Dijagram stanja

4.4 Dijagram aktivnosti

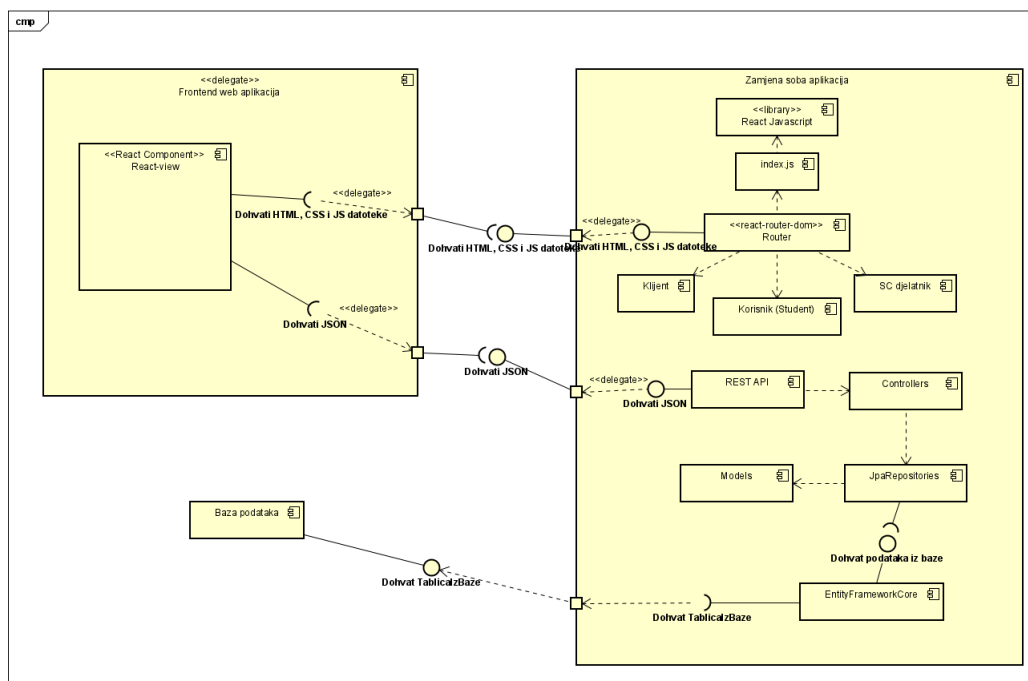
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti 4.7 prikazan je proces lajkanja oglasa. Korisnik nakon prijave u sustav ima uvid u predane oglase. On zatim može odabrati i lajkati neki od prikazanih oglasa ili unijeti filtere za pretraživanje oglasa. Ukoliko korisnik odabere opciju za filtriranje oglasa, sustav će mu prikazati samo one oglase koji odgovaraju njegovim kriterijima te korisnik onda po želji lajka. Nakon što korisnik lajka oglas, promjene se unose u sustav i korisniku se prokazuje uspješno lajkanje u obliku obojanih srca koji su ranije bili bijeli.



Slika 4.4: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici 4.8 opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju frontend dijelu aplikacije. Router je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. Frontend dio se sastoji od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po tipovima aktera koji im pristupaju. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvacaju gotove komponente kao što su gumbi, forme i slično. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju backend dijelu aplikacije. EntityFrameworkCore je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Reactview komponenta preko dostupnih sučelja komunicira s aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.



Slika 4.5: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu ostvarena je pomoću aplikacija WhatsApp i Discord. Za upravljanje i verzioniranje izvornog koda korišten je Git te udaljeni repozitorij na web platformi GitLab. Za razvoj backenda korištena su razvojna okruženja (IDE)

Eclipse i IntelliJ Idea, a za razvoj frontenda Visual Studio Code. Rad sa bazom podataka ostvaren je pomoću sustava PostgreSQL i pgAdmin. Aplikacija je napisana u programskim jezicima Java i JavaScript. Backend aplikacije izrađen je u radnom okviru Spring Boot, koji sadrži brojne biblioteke za rad sa bazom podataka, izradu web aplikacije i sigurnost pri slanju HTTP zahtjeva. Za frontend je korištena biblioteka React koja se koristi za izgradnju korisničkih sučelja. Aplikacija je hostana na cloud platformi Heroku.

<https://www.whatsapp.com>
<https://discord.com>
<https://git-scm.com>
<https://about.gitlab.com>
<https://www.eclipse.org/ide>
<https://www.jetbrains.com/idea>
<https://code.visualstudio.com>
<https://www.postgresql.org>
<https://www.java.com>
<https://www.javascript.com>
<https://spring.io>
<https://reactjs.org>
<https://www.heroku.com>

5.2 Ispitivanje programskog rješenja

Provedeno je automatizirano ispitivanje programskog rješenja. Testiranje komponenti sustava izvedeno je pomoću JUnit testova, a ispitivanje ponašanja sustava pomoću alata Selenium IDE.

5.2.1 Ispitivanje komponenti

Proveli smo 7 JUnit testova koji testiraju razrede i metode u aplikaciji.

Ispitni slučaj 1: Prvi test testira registraciju novog usera u aplikaciji.

```
@Test
public void testSuccessfulRegistration() throws Exception{

    SignupRequest signupRequest = new SignupRequest();
    signupRequest.setEmail("test@email.com");
    signupRequest.setUsername("testUser");
    signupRequest.setPassword("testPassword");
    signupRequest.setJmbag("0123456789");
    signupRequest.setName("TestniUser");
    signupRequest.setRole(UserRole.STUDENT);

    String json = new ObjectMapper().writeValueAsString(signupRequest);

    this.mockMvc.perform(post("/api/v1/signup").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isOk())
        .andExpect(jsonPath("message").value("User registered successfully!"));
}
```

Ispitni slučaj 2 i 3: Sljedeća dva testa testiraju hoće li se dogoditi iznimka u slučaju ako pokušamo registrirati novog korisnika u sustav ukoliko već postoji korisnik s istim usernameom ili se email već koristi.


```
@Test
public void testUnsuccessfulRegistrationUsernameTaken() throws Exception{

    SignupRequest signupRequest = new SignupRequest();
    signupRequest.setEmail("test2@email.com");
    signupRequest.setUsername("testUser");
    signupRequest.setPassword("testPassword");
    signupRequest.setJmbag("0123456789");
    signupRequest.setName("TestniUser");
    signupRequest.setRole(UserRole.STUDENT);

    String json = new ObjectMapper().writeValueAsString(signupRequest);

    this.mockMvc.perform(post("/api/v1/signup").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isBadRequest())
        .andExpect(jsonPath("message").value("Error: Username is already taken!"));

}
```

```
@Test
public void testUnsuccessfulRegistrationEmailTaken() throws Exception{

    SignupRequest signupRequest = new SignupRequest();
    signupRequest.setEmail("test@email.com");
    signupRequest.setUsername("testUser2");
    signupRequest.setPassword("testPassword");
    signupRequest.setJmbag("0123456789");
    signupRequest.setName("TestniUser");
    signupRequest.setRole(UserRole.STUDENT);

    String json = new ObjectMapper().writeValueAsString(signupRequest);

    this.mockMvc.perform(post("/api/v1/signup").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isBadRequest())
        .andExpect(jsonPath("message").value("Error: Email is already in use!"));

}
```

Ispitni slučaj 4: Test 4 provjerava api koji vraća podatke o trenutno logiranom korisniku.

```
@Test
public void testGetCurrentUserFromAuthToken() throws Exception{
    this.mockMvc.perform(get("/api/v1/currentuser").header("Authorization",
        "Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiI0ZXN0VXNlciIsIm1hdCI6MTYxMDQ2ODk5OStiZW50IjoxNjEwNTU1Mzk5
        + "_RBHaPzI1h6C67daGaBkE6auADDZvIlQRCvOMI-hnLJdsNzVVQpAO65qLk9d_NmocMzj05G89Q"))
        .andDo(print())
        .andExpect(status().isOk())
        .andExpect(jsonPath("username").value("testUser"));
}
```

Ispitni slučaj 5: Test 5 provjerava baca li se odgovarajuća iznimka kada se

pokuša koristiti api za provjeru podataka o trenutno logiranom korisniku, a odgovarajući JWT token nije bio poslan.

```
@Test
public void testAuthTokenNotSent() throws Exception{
    this.mockMvc.perform(get("/api/v1/currentuser"))
        .andDo(print())
        .andExpect(status().is4xxClientError())
        .andExpect(content().string(containsString("Nije poslan token za autorizaciju")));
}
```

Ispitni slučaj 6: Test 6 provodi provjeru logiranja u sustav. Ukoliko se korisnik uspješno ulogira, test prolazi.

```
@Test
public void testSuccessfulLogin() throws Exception{

    LoginRequest login = new LoginRequest();
    login.setUsername("testUser");
    login.setPassword("testPassword");

    String json = new ObjectMapper().writeValueAsString(login);

    this.mockMvc.perform(post("/api/v1/signin").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isOk())
        .andExpect(jsonPath("username").value("testUser"));
}
```

Ispitni slučaj 7: Test 7 provjera što se će se dogoditi u slučaju neispravnih korisničkih podataka za login. Očekivano ponašanje je error kod 400 uz poruku "Bad credentials".

```
@Test
public void testUnsuccessfulLogin() throws Exception{

    LoginRequest login = new LoginRequest();
    login.setUsername("testUser");
    login.setPassword("wrongPassword");

    String json = new ObjectMapper().writeValueAsString(login);

    this.mockMvc.perform(post("/api/v1/signin").contentType(MediaType.APPLICATION_JSON).content(json))
        .andDo(print())
        .andExpect(status().isBadRequest())
        .andExpect(jsonPath("message").value("Bad credentials"));
}
```

5.2.2 Ispitivanje sustava

Ispitivanje sustava provedeno je dodatkom za preglednik Selenium IDE. Testirani su slučajevi registracije, prijave korisnika, objave oglasa i oznake oglasa da se više ne prikazuje.

Ispitni slučaj 1: Registracija korisnika Testirana je registracija korisnika s jedinstvenim podacima te je očekivani rezultat uspješna registracija.

Registration Successful*			
	Command	Target	Value
6	type	id=normal_login_username	Ivan
7	click	id=normal_login_email	
8	type	id=normal_login_email	ivan.ivanic@gmail.com

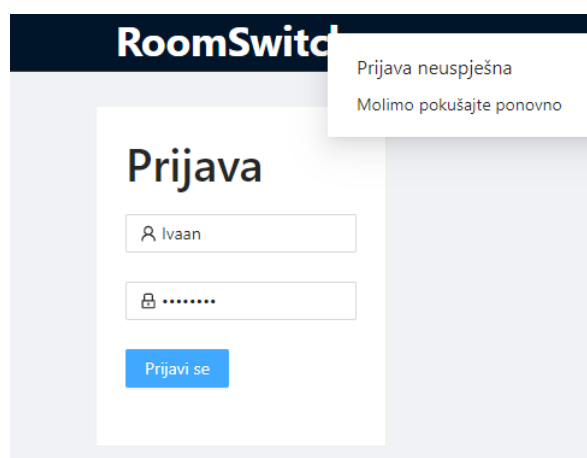
Command	type	//	
Target	id=normal_login_email		
Value	ivan.ivanic@gmail.com		
Description			

RoomSwitchr

Prijava

Ispitni slučaj 2: Prijava nepostojećeg korisnika Testirana je prijava korisnika s nepostojećim korisničkim imenom te je očekivani rezultat odbijanje prijave i ispis poruke u grešci.

	Command	Target	Value
Login Unsuccessful*			
Registration Successful*	6	type	id=normal_login_username
	7	click	id=normal_login_password
	8	type	id=normal_login_password
	Command	type	
	Target	id=normal_login_username	
	Value	Ivaan	
	Description		



Ispitni slučaj 3: Predaja oglasa Testirana je predaja oglasa korisnika koji već ima aktivni oglas. Očekivani rezultat je stvaranje novog aktivnog oglasa, a prethodni aktivni oglas postaje neaktivan.

	Command	Target	Value
Login Unsuccessful*			
Registration Successful*	6	click	css=.ant-btn-primary > span:nth-child...
Submit Ad*	7	click	id=newPost_domID
	8	click	css=.ant-select-item-option-active > ...
	Command	click	
	Target	id=newPost_domID	

RoomSwitch

Uspješna predaja
Novi oglas je uspješno predan!

Novi oglas

Opis sobe

Stjepan Radić

1

1

1

Predaj oglas

Moji oglasi

Stvori novi oglas

Moja Objava

Neaktivan oglas

Dom: Stjepan Radić

Paviljon: 1

Kat: 1

Kategorija sobe: 1

uredi

Moja Objava

Aktivan oglas

Dom: Cvjetno Naselje

Paviljon: 1

Kat: 1

Kategorija sobe: 1

uredi





Ispitni slučaj 4: Ne prikazuj više ovaj oglas Testirano je označavanje oglasa da se više ne prikazuje te je očekivani rezultat početna stranica na kojoj nema

označenog oglasa.





RoomSwitchr

> Filtriranje oglasa

Dom: Dr. Ante Starčević
Paviljon:5
Kat:1
Kategorija sobe:1





Dom: Laščina
Paviljon:1
Kat:3
Kategorija sobe:2

RoomSwitchr

> Filtriranje oglasa


Dom: Laščina
Paviljon:1
Kat:3
Kategorija sobe:2

Moja Objava

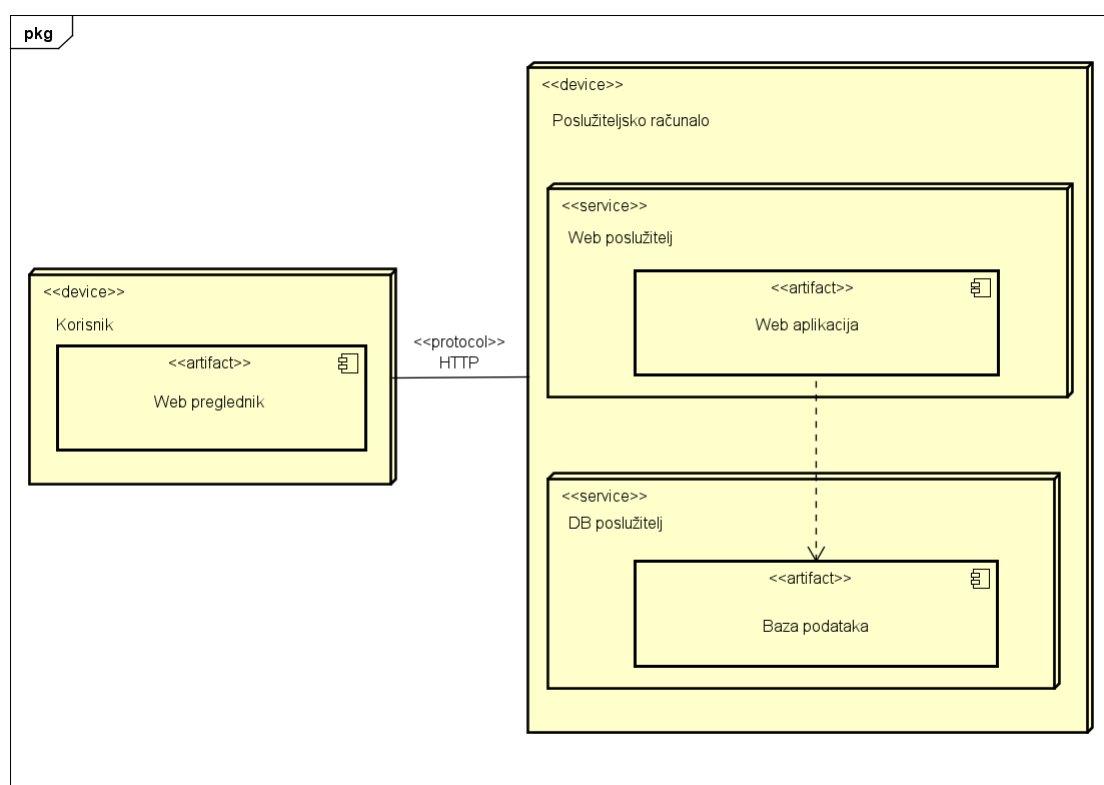
Aktivan oglas

Dom: Cvjetno Naselje
Paviljon: 1
Kat: 3
Kategorija sobe: 2

 uredi

5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. Na poslu žiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Klijenti koriste web preglednik kako bi pristupili web aplikaciji. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala korisnika (student ili djelatnik studentskog centra) i poslužitelja odvija se preko HTTP veze.



Slika 5.1: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Potrebno je preuzeti PostgreSQL bazu podataka. Nakon toga potrebno je provesti standardnu instalaciju te se prijaviti s korisničkim imenom „postgres“ i lozinkom „bazepodataka“. Nakon instalacije, potrebno je pokrenuti pgAdmin i konfigurirati prvu konekciju na bazu. Potrebno je prijaviti se s istim korisničkim imenom i lozinkom kao i prilikom instalacije. U bazi podataka bi već trebala postojati tablica „postgres“. Server baze mora biti pokrenut. U daljnjim uputama opisan je način podizanja baze.

Kako bi se aplikacija pokrenula potrebno je instalirati podršku za bazu podataka, podršku za prevođenje i pokretnaje aplikacija napisanih u programskom jeziku Java te podršku za prevođenje i pokretnaje aplikacija napisanih u programskom jeziku JavaScript.

Prilikom instalacije podrške za bazu podataka potrebno je preuzeti aplikaciju pgAdmin s poveznice <https://www.pgadmin.org/download/>. Nakon preuzimanja, potrebno je provesti standardnu instalaciju te se prijaviti s korisničkim imenom „postgres“ i lozinkom „bazepodataka“. Nakon instalacije, potrebno je pokrenuti pgAdmin i konfigurirati prvu konekciju na bazu. Potrebno je prijaviti se s istim korisničkim imenom i lozinkom kao i prilikom instalacije. U bazi podataka bi već trebala postojati tablica „postgres“. Server baze tijekom korištenja aplikacije mora biti pokrenut. Na računalu je također potrebno instalirati podršku za programski jezik Java (<http://jdk.java.net/15>). Također kako bi se mogli pokretati programi napisani u JavaScriptu potrebno je instalirati Node.js (<https://nodejs.org/en/download>). Osim podrški za pokretanje programa, potrebno je instalirati i razvojnu okolinu. Najjednostavnije je instalirati IDE Intelij jer se u njemu mogu pokretati programi napisani i u jeziku Java i u jeziku JavaScript. Projekt se može preuzeti s poveznice <https://gitlab.com/mariakatic/piccologrupo>. Nakon uspješne instalacije svega od navedenog, konačno se može pokrenuti aplikacija. Potrebno je instalirati npm pakete pokretanjem naredbe `npm install`. Frontend se pokreće naredbom `npm start`, a backend pritiskom tipke Run u razvojnom okruženju. Nakon pokretanja programa, aplikacija se može pregledati u pregledniku na poveznici <http://localhost:3000>. Na gore opisan način pokreće se aplikacija neovisno o tome je li deployana. Ova aplikacija hostana je na cloud platformi Heroku te se također može pronaći na linku <https://switchr.herokuapp.com>.

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije za olakšanu zamjenu soba u studentskim domovima. Aplikacija automatizira pronalazak mogućih kandidata za zamjenu te informiranje studentskog centra. Izrada projekta izvršena je kroz dva koraka. Prvi korak uključivao je okupljanje tima, analiziranje zahtjeva zadatka, dogovor o raspodjeli posla i pisanje dokumentacije. Izrada obrazaca i dijagrama te detaljan opis rada aplikacije pomogli su pri ubrzanju razvoja u drugom koraku. Također je omogućeno razdvajanje posla na backend i frontend jer je njihova interakcija unaprijed određena. Što se tiče same aplikacije, izrađen je kostur na koji su se mogle brže i učinkovitije dodavati tražene funkcionalnosti. Drugi korak bio je sama izrada aplikacije u obliku pisanja izvornog koda i dizajniranja korisničkog sučelja. U ovom koraku rađeno je i temeljito testiranje funkcionalnosti što je dovelo do brojnih inačica rješenja tokom ispravljanja grešaka. Dokumentacija je u drugoj fazi nadopunjena i proširena po potrebi zbog novih ideja i unaprijed nepredviđenih slučajeva korištenja. Moguće proširenje mogućnosti aplikacije je prijava zamjena dogovorenih izvan aplikacije radi lakše interakcije sa studentskim centrom, naprednije upravljanje oglasima i filtriranjem te poboljšanje sigurnosti podataka. Također je moguće izraditi i mobilnu aplikaciju kojom bi se povećala pristupačnost usluge koju projekt pruža. Sudjelovanje na ovom projektu svim je članovima tima pružilo iskustvo u timskom radu na većem projektu. Shvatili smo potrebu za dobrom dokumentacijom i koordinacijom između članova pri svakom koraku razvoja. Iako završen projekt ima još mnogo prostora za poboljšanje zbog neiskustva sa ovakvom vrstom zadatka, zadovoljni smo onime što smo postigli.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. ERDPlus, <https://erdplus.com/>
4. Astah Community, <http://astah.net/editions/uml-new>
5. <https://www.bu.edu/housing/assignments/changes/requests/room-change-requests-walkthrough/>

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 29. rujna 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - odabran voditelj tima i ime tima

2. sastanak

- Datum: 8. listopada 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom i demosom
 - analiza zadatka

3. sastanak

- Datum: 12. listopada 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - dogovor oko alata i tehnologije
 - podjela rada

4. sastanak

- Datum: 21. listopada 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom i demosom
 - analiza zadatka
 - nejasnoće u implementaciji generičkih funkcionalnosti

5. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić

- Teme sastanka:
 - ispravljanje pogrešaka u implementaciji

6. sastanak

- Datum: 5. studenog 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom i demosom
 - razrješavanje nejasnoća oko implementacije

7. sastanak

- Datum: 10. studenog 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom
 - pregled dokumentacije

8. sastanak

- Datum: 12. studenog 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom i demosom
 - demonstracija generičkih funkcionalnosti aplikacije

9. sastanak

- Datum: 14. prosinca 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - podjela zadataka između članova

10. sastanak

- Datum: 22. prosinca 2020.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - pregled trenutno napravljenog
 - razjašnjenje nejasnoća

11. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić

- Teme sastanka:
 - razjašnjenje nejasnoća
 - analiza traženih zadataka
 - podjela ostatka posla i dokumentacije

12. sastanak

- Datum: 7. siječnja 2021.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - sastanak s asistenticom i demosom
 - demonstracija alfa verzije

13. sastanak

- Datum: 8. siječnja 2021.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - pregled trenutno napravljenog
 - popravljanje grešaka u kodu

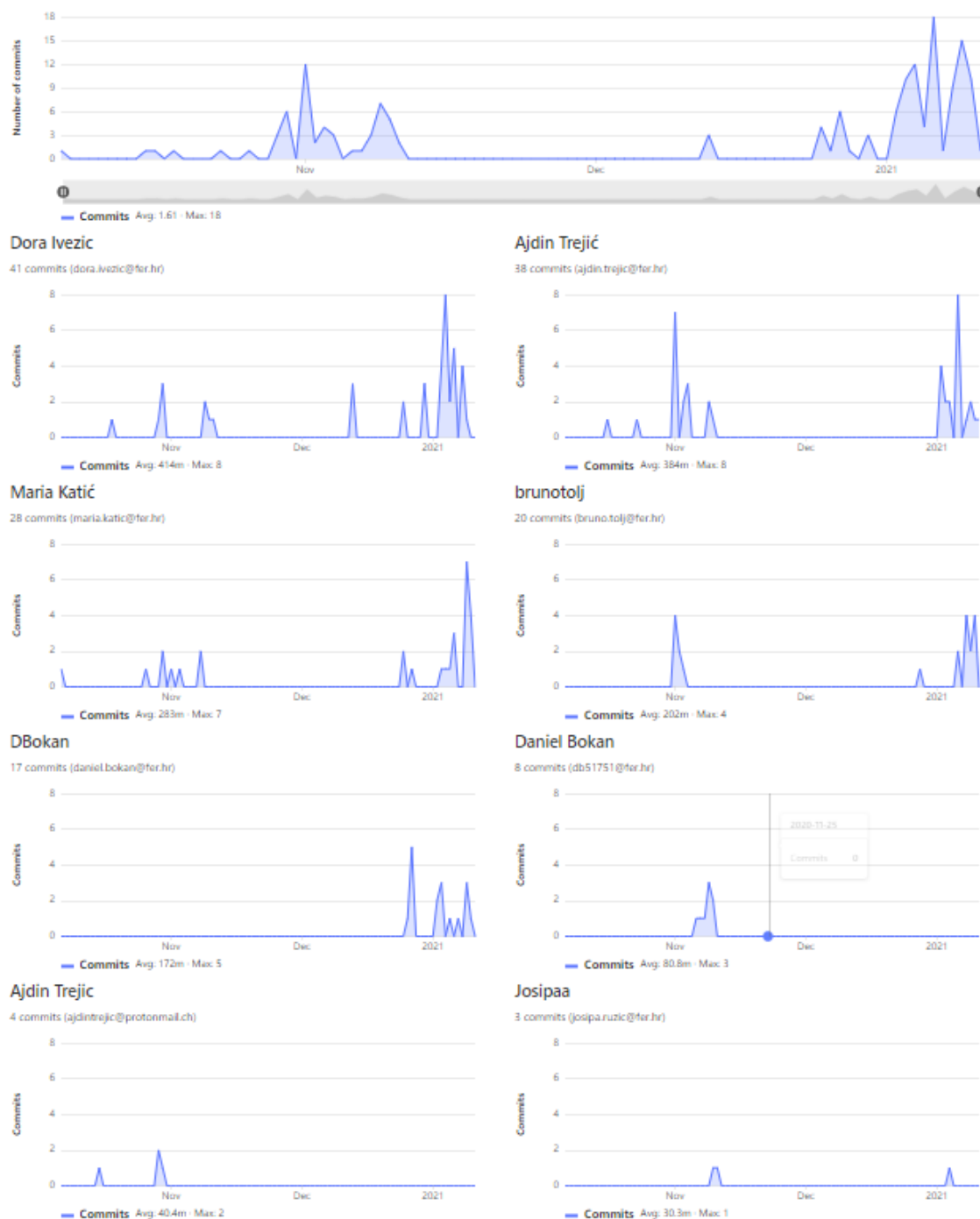
14. sastanak

- Datum: 13. siječnja 2021.
- Prisustvovali: D.Ivezić, A.Trejić, B.Tolj, J.Ružić, D.Bokan, M.Katić
- Teme sastanka:
 - spajanje backenda i frontenda

Tablica aktivnosti

	Maria Katić	Ajdin Trejić	Dora Ivezić	Daniel Bokan	Josipa Ružić	Bruno Tolj	x
Upravljanje projektom	2						
Opis projektnog zadatka	2						
Funkcionalni zahtjevi					1		
Opis pojedinih obrazaca					8		
Dijagram obrazaca					1		
Sekvencijski dijagrami					2		
Opis ostalih zahtjeva					1		
Arhitektura i dizajn sustava			3				
Baza podataka			4				
Dijagram razreda			4				
Dijagram stanja					2		
Dijagram aktivnosti					2		
Dijagram komponenti	1						
Korištene tehnologije i alati						1	
Ispitivanje programskog rješenja	1			2			
Dijagram razmještaja					1		
Upute za puštanje u pogon		1					
Dnevnik sastajanja	1						
Zaključak i budući rad						1	
Popis literature	1						
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>		40	40				
<i>izrada baze podataka</i>	1	6					
<i>spajanje s bazom podataka</i>				4			
<i>back end</i>	40	3		40	10	40	

Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti na repozitoriju

Indeks slika i dijagrama

2.1	Aplikacija za zamjenu soba Sveučilišta u Bostonu	8
3.1	Dijagram obrazaca uporabe	19
3.2	Sekvencijski dijagram za UC5	20
3.3	Sekvencijski dijagram za UC11	21
3.4	Sekvencijski dijagram za UC13	22
4.1	Dijagram baze podataka	31
4.2	Dijagram razreda	33
4.3	Dijagram stanja	34
4.4	Dijagram aktivnosti	36
4.5	Dijagram komponenti	37
5.1	Dijagram razmještaja	46
6.1	Prikaz aktivnosti na repozitoriju	54