

Programsko inženjerstvo

Ak. god. 2020./2021.

Zamjena sobe

Dokumentacija, Rev. 2

Grupa: *springRolice*

Voditelj: *Mateja Iveta*

Datum predaje: *14. 1. 2021.*

Nastavnik: *Ivana Cepetić*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	3
2.1 Opseg projekta	3
2.2 Postojeća slična rješenja	6
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	17
3.2 Ostali Zahtjevi	19
4 Arhitektura i dizajn sustava	20
4.1 Baza podataka	21
4.1.1 Opis tablica	22
4.1.2 Dijagram baze podataka	30
4.2 Dijagram razreda	31
4.3 Dijagram stanja	34
4.4 Dijagram aktivnosti	36
4.5 Dijagram komponenti	37
5 Implementacija i korisničko sučelje	38
5.1 Korištene tehnologije i alati	38
5.2 Ispitivanje programskog rješenja	40
5.2.1 Ispitivanje komponenti	40
5.2.2 Ispitivanje sustava	40
5.3 Dijagram razmještaja	45
5.4 Upute za puštanje u pogon	46
6 Zaključak i budući rad	50
Indeks slika i dijagrama	52

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napisani obrasci uporabe	Bortas	02.11.2020.
0.2	Napisan Opis projektnog zadatka	Horvat	03.11.2020.
0.5	Dodani Use Case dijagrami i sekvencijski dijagrami	Bortas	04.11.2020.
0.7	Dodan opis baze, tablica i ER dijagram	Horvat	11.11.2020.
0.8	Dodani dijagrami razreda	Bortas	12.11.2020.
1.0	Dodana aktivnost članova i provjera dokumentacije	Horvat	13.11.2020.
1.1	Dodan dijagram stanja, aktivnosti i dijagram razmještaja	Horvat	6.1.2021.
1.2	Dodan opis tehnologija i alata	Horvat	7.1.2021.
1.3	Ažuriran opis baze podataka i dodan ažuriran ER dijagram	Horvat	11.1.2021.
1.4	Dodan dijagram komponenti	Horvat	11.1.2021.
1.5	Dodan zaključak	Horvat	11.1.2021.
1.6	Dodani ažurirani dijagrami razreda	Horvat	12.1.2021.
1.7	Dodani grafovi aktivnosti s GitLaba	Horvat	13.1.2021.
1.8	Dodan opis testiranja	Horvat	13.1.2021.
1.9	Dodane upute za puštanje u pogon	Horvat	14.1.2021.
2.0	Konačna verzija	Horvat	14.1.2021.

2. Opis projektnog zadatka

2.1 Opseg projekta

Glavni je cilj razvoja aplikacije "Zamjena sobe" svim studentima koji su tijekom studiranja smješteni u studentskim domovima ponuditi rješenje za dva najveća problema s kojima se susreću. To su traženje sobe, to jest osobe za zamjenu te sam proces prijave te zamjene u Studentskim centrima.

Nakon provedenog postupka natječaja za smještaj i objave liste ostvarenja prava na studentski smještaj, studenti imaju mogućnost zamjene doma ili sobe unutar dodijeljenog doma. Studenti potencijalne zamjene traže uglavnom nasumično, usmenim putem ili putem raznih foruma i društvenih mreža, na posebnim grupama za pojedine gradove i studentske domove. Ovakvim izuzetno nepraktičnim i nestrukturiranim načinom zamjene vrlo je lako da se dva, moguće idealna oglasa, nikada ne povežu. Također, studentima se ne omogućuje postavljanje i traženje zamjene po više željenih kriterija.

Ova bi aplikacija bila dostupna svima, ne samo studentima na određenom fakultetu ili gradu. Neregistrirani korisnik ima mogućnost pregleda svih oglasa i glavnih informacija o pojedinom oglasu: ime, prezime i profilna slika osobe koja je predala oglas, grad, paviljon i broj ponuđene sobe te željene kriterije nove sobe. Za korištenje svih ostalih funkcionalnosti neregistrirani korisnik može se registrirati, to jest stvoriti novi račun ili se prijaviti u sustav s već stvorenim računom. Kod prijave u sustav potrebno je upisati korisničko ime ili e-mail adresu i lozinku. Za registraciju korisnik mora upisati sljedeće podatke:

- ime
- prezime
- korisničko ime
- e-mail adresa
- lozinka

- jmbag
- grad u kojem studira

Prijavljenom korisniku omogućen je pregled njegova profila na kojemu može vidjeti i uređivati svoje osobne podatke i sliku profila, ili izbrisati svoj račun. Također, na pregledu "Moji oglasi" ima uvid u vlastite predane oglase koji mogu biti aktivni ili neaktivni. Aktivni oglas može dodatno uređivati ili ga učiniti neaktivnim.

Kod predaje oglasa za zamjenu sobe student o sobi koju 'nudi' navodi sljedeće informacije:

- dom
- paviljon
- kat
- broj kreveta
- tip kupaonice
- komentar

Dodatno, student u tekstualno polje ima opciju upisa dodatnih značajki sobe, poput najbliže menze i slično. Osim informacija o svojoj sobi, student navodi i informacije o sobi koju želi, to jest željene kriterije. Jedino se ograničenje odnosi na grad - student ne može tražiti zamjenu u drugom gradu. Ostali kriteriji mogu biti i višestruki, na primjer može označiti da mu odgovaraju sobe s jednim, dva ili tri kreveta.

Nakon što preda oglas, student ima uvid u ostale aktivne oglase koji odgovaraju njegovim željenim kriterijima. Ti oglasi mogu biti prikazani kao pojedinačni ili ulančani. Sve ponuđene oglase student može ocijeniti po stupnjevima podudaranja kriterija:

- 1 - sviđa mi se
- 2 - sviđa mi se jako
- 3 - to je to
- Ne prikazuj više ovaj oglas

Osim pregleda ponuđenih oglasa, student također ima uvid u zaprimljene ponude za zamjenu - koji su studenti zainteresirani za njegov oglas te njihove oglase.

Sustav periodički pretražuje nove dodane oglase i ažurira uparivanja po kriterijima. Kao što je već spomenuto, uz izravne, pojedinačne zamjene sustav nudi i potencijalne lance razmjene, ako takvi postoje.

Nakon što oba studenta (ili svi studenti u slučaju lanca razmjene) ocijene razmjenu, sustav im šalje e-mail poruku kojom se traži da konačno potvrde zamjenu soba. Poveznica na određenu zamjenu sadržana je u e-mail poruci.

Sve su potvrđene zamjene dostupne na uvid ovlaštenim djelatnicima pojedinog Studentskog centra. Nakon prijave u sustav zaključane zamjene izravno evidentiraju u sustavu Studentskog centra, bez potrebe za osobnom intervencijom studenata.

2.2 Postojeća slična rješenja

Mnoga strana sveučilišta imaju slična rješenja za zamjenu soba, ali unutar svojih privatnih sustava koji su dostupni samo studentima toga sveučilišta. Jedan je takav primjer 'Room Swap' sveučilišta *The University of North Carolina at Chapel Hill*. 'Room Swap' je također dostupan samo studentima na sveučilištu 'North Carolina'. Glavna je razlika između ovog sustava i naše aplikacije u tome što putem *Room Swapa* student ne može provesti 'pravu' zamjenu sobe – može izabrati neku slobodnu sobu na kampusu ili sobu koja nije do kraja popunjena, ne može se direktno zamijeniti s drugom osobom, što bi naša aplikacija omogućavala.

Student u sustavu *Room Swap* prvo izabire zgradu te se klikom na zgradu otvara pregled katova i dostupnih slobodnih soba.

Room Selection - Room Swap (Step 9 of 9)

Main | Renew Residence Hall | General Room Selection | **Room Swap**

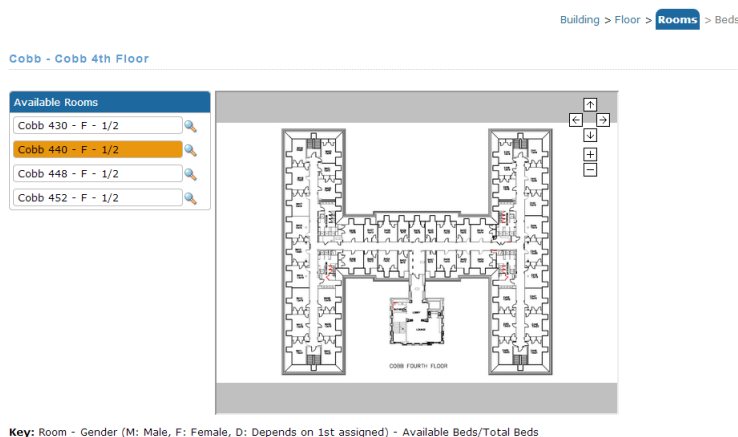
Building > **Floor** > Rooms > Beds

Cobb

Floor/Suite	Single Occupancy Spaces	Double Occupancy Spaces	Triple Occupancy Spaces
Cobb 1st Floor	0	1	1
Cobb 2nd Floor	0	1	1
Cobb 4th Floor	0	4	0

Slika 2.1: Prikaz katova i dostupnih slobodnih soba na pojedinom katu

Još je jedna razlika u odnosu na našu aplikaciju to što nisu sve sobe dostupne svima. Na primjer, postoje 'First Year Experience Buildings' koje su dostupne samo brucosima. Također, sobe s više kreveta podijeljene su po spolu. Nakon izbora kata aplikacija prikazuje tlocrt toga kata sa svim dostupnim sobama.



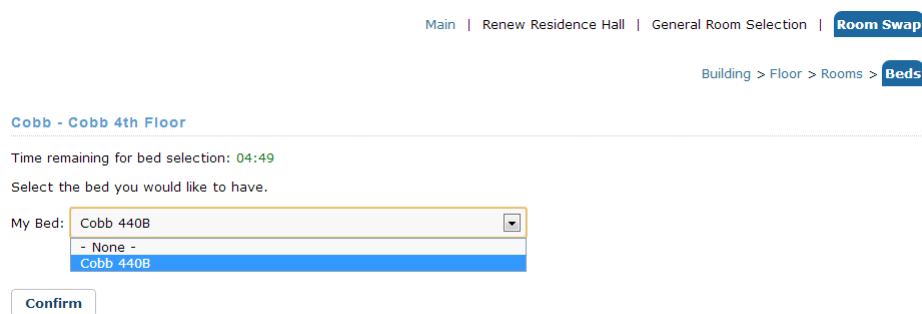
Slika 2.2: Tlocrt odabranog kata

Klikom na sobu moguće je prikaz detalja o sobi ali i studentu kojemu je trenutno dodijeljena soba. Naša bi aplikacija, uz osnovne informacije o sobi poput doma, paviljona, kata i vrste, nudila i mogućnost detaljnijeg opisa - tip kupaonice, blizina menza itd.



Slika 2.3: Detalji o sobi i studentima u sobi

Nakon odabira sobe pokreće se odbrojavanje od 5 minuta unutar kojih student mora izabrati krevet u sobi i konačno potvrditi svoj potpuni odabir. Ako se odabir unutar 5 minuta ne potvrdi zamjena se neće provesti. Naša aplikacija ne bi postavljala vremensko ograničenje na potvrdu zamjene, već bi se čekala potvrda obje strane.



Slika 2.4: Odbrojavanje i konačni odabir

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Studenti
2. Djelatnici Studentskih centara
3. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) vidjeti sve oglase
 - (b) se registrirati, za stvaranje korisničkog računa potrebni su: ime, prezime, korisničko ime, lozinka, e-mail, JMBAG, grad u kojem studira
2. Registrirani korisnik (inicijator) može:
 - (a) se prijaviti u sustav
 - (b) stvoriti oglas
 - (c) vidjeti svoj profil i mijenjati podatke
 - (d) izbrisati svoj profil
 - (e) vidjeti oglase pojedinačne ili ulančane koji odgovaraju njegovim kriterijima
 - (f) pregledati sve svoje oglase (aktivne i neaktivne)
 - (g) uređivati svoje oglase
 - (h) učiniti svoje oglase aktivnim i neaktivnim
 - (i) brisati svoj oglas
 - (j) "lajkati" oglase po stupnjevima:
 - i. 1-sviđa mi se
 - ii. 2-sviđa mi se jako
 - iii. 3-to je to
 - iv. Ne prikazuj više ovaj oglas

3. Djelatnik SC-a (inicijator) može:

- (a) se prijaviti u sustav
- (b) pregledati sve zaključane zamjene

4. Timer (inicijator) može:

- (a) pokrenuti uparivanje studenata s oglasima koji odgovaraju njegovim kriterijima

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o oglasima i sobama

6. Poslužitelj (sudionik):

- (a) obrađuje zahtjeve korisnika

3.1.1 Obrasci uporabe

UC1 -Pregled oglasa

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:**Pregledati oglase soba
- **Sudionici:** baza podataka
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 1. Prilikom otvaranja aplikacije sustav prikazuje sve oglase

UC2 -Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:**Registrirati korisnika i ovlastiti ga
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** dostupnost poslužitelja, korisnik nije registriran
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku "Registracija"
 2. Sustav otvara stranicu za registraciju
 3. Korisnik unosi potrebne podatke
 4. Sustav provjerava ispravnost podataka
 5. Sustav registrira korisnika
 6. Korisnik prima podatke o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Korisnik unosi neispravne podatke ili već zauzeto korisničko ime ili e-mail
 1. Sustav obavještava korisnika o pogrešci
 2. Sustav vraća korisnika na stranicu za registraciju sa crveno označenom greškom
 3. Korisnik mijenja neispravne podatke te završava unos ili odustaje od registracije

UC3 -Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik, Djelatnik SC-a
- **Cilj:**Prijaviti se u sustav
- **Sudionici:** baza podataka
- **Preduvjet:** Korisnik se je registrirao

- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Sustav provjerava ispravnost podataka
 3. Sustav korisniku otvara početnu stranicu i korisnik ima pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Unos neispravnih podataka
 1. Sustav obavještava o neispravnim podacima i vraća na stranicu za prijavu s crveno označenom pogreškom
 2. Korisnik ispravlja podatke

UC4 -Objavljivanje oglasa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Objaviti oglas
- **Sudionici:** baza podataka,poslužitelj
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Objavi novi oglas"
 2. Sustav otvara stranicu za objavljivanje oglasa
 3. Korisnik upisuje potrebne podatke za sobu koju nudi i kriterije za sobu koju traži. Sve podatke odabire iz padajućih izbornika osim polja za proizvoljni komentar
 4. Korisnik pritišće tipku "objavi"
 5. Sustav pohranjuje oglas u bazu

UC5 -Pregled mogućih zamjena

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Pregledati ponuđene moguće zamjene koje odgovaraju kriterijima korisnika
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen i ima aktivan oglas
- **Opis osnovnog tijeka:**
 1. Korisnik otvara početnu stranicu
 2. Sustav prikazuje sve oglase koji odgovaraju njegovim kriterijima

UC6 - Uparivanje oglasa

- **Glavni sudionik:** Timer
- **Cilj:** Upariti studenta s oglasima koji odgovaraju njegovim kriterijima
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 1. Timer pokreće uparivanje svakih 5 sati
 2. Sustav za svakog studenta stvara listu svih oglasa koji direktno ili lančano odgovaraju njegovim kriterijima

UC7 -Lajkanje oglasa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** "Lajkati" oglase soba koje korisnika zanimaju
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen, ima aktivan oglas i barem jedan oglas odgovara njegovim kriterijima
- **Opis osnovnog tijeka:**
 1. Korisnik oglase koji ga zanimaju označava ("lajka") po stupnjevima od 1 do 3
 2. Sustav sprema njegov odabir

UC8 -Mjenjanje Lajka

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Promijeniti razinu "lajka" ili maknuti lajk
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen, ima aktivan oglas, barem jedan oglas odgovara njegovim kriterijima
- **Opis osnovnog tijeka:**
 1. Korisnik odabire neki drugi stupanj "lajka" ili odustaje od "lajka" micanjem oznake
 2. Njegov odabir se sprema u bazu podataka

UC9 -Pregled mojih oglasa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Vidjeti sve svoje oglase
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen i ima oglas

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Moji oglasi"
2. Sustav otvara stranicu gdje se prikazuju svi korisnikovi oglasi

UC10 -Uređivanje oglasa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Urediti već postojeći oglas
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen i ima aktivan oglas
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi pod "Moji oglasi"
 2. Sustav prikazuje stranicu sa listom njegovih oglasa
 3. Korisnik odabire opciju uredi
 4. Korisnik mijenja podatke ili atribut je li oglas aktivan ili neaktivan
 5. Korisnik potvrđuje promjene odabirom opcije spremi
 6. Promjene se spremaju u bazu podataka

UC11 -Brisanje oglasa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Izbrisati oglas
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen i ima oglas
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi pod "Moji oglasi"
 2. Sustav prikazuje stranicu sa listom njegovih oglasa
 3. Korisnik odabire opciju izbriši
 4. Sustav briše oglas iz baze podataka

UC12 -Pregled profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Vidjeti svoje osobne podatke
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi pod "Moj profil"
 2. Sustav otvara stranicu korisnikova profila

UC13 -Promjena profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Vidjeti svoje osobne podatke
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi pod "Moj profil"
 2. Sustav otvara stranicu korisnikova profila
 3. Korisnik odabire opciju "uredi"
 4. Korisnik mijenja osobne podatke
 5. Korisnik pritišće tiku "spremi promjene"
 6. Sustav sprema promjene

UC14 -Brisanje profila

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Vidjeti svoje osobne podatke
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi pod "Moj profil"
 2. Sustav otvara stranicu korisnikova profila
 3. Korisnik odabire opciju "izbriši profil"
 4. Sustav briše korisnikov profil i otvara početnu stranicu

UC15 -Potvrđivanje zamjene

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Potvrditi zamjenu sobe
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Korisnik je prijavljen, sve uključene strane su "lajkale" oglas
- **Opis osnovnog tijeka:**
 1. Korisnik dobiva obavijest da su sve strane "lajkale" zamjenu
 2. Korisnik pomoću linka iz obavijesti dolazi do oglasa
 3. Korisnik potvrđuje zamjenu
 4. Potvrda se sprema u bazu podataka

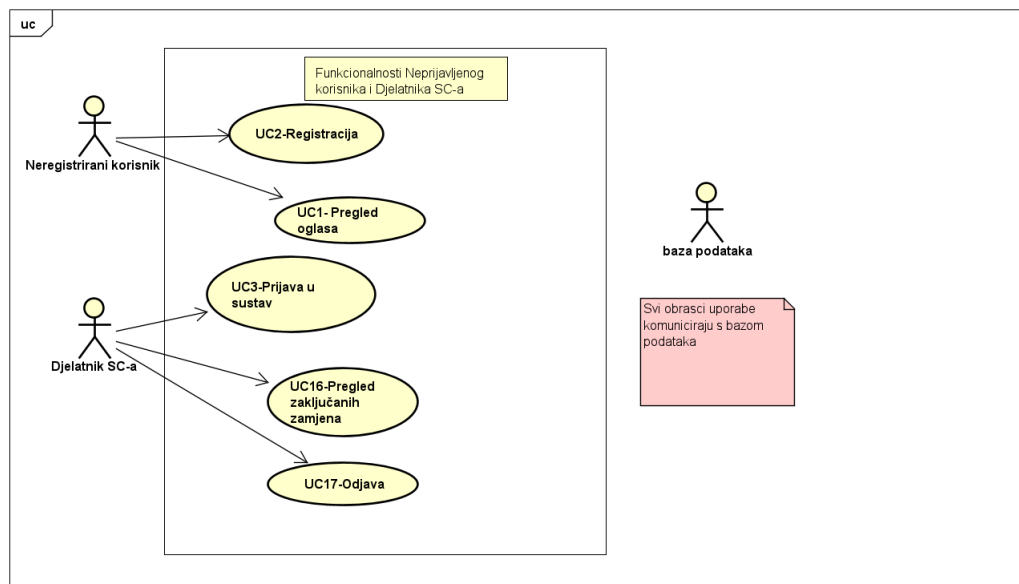
UC16 -Pregled zaključenih zamjena

- **Glavni sudionik:** Djelatnik SC-a
- **Cilj:** Vidjeti sve zaključane zamjene kako bi se mogle provesti u sustavu SC-a
- **Sudionici:** baza podataka, poslužitelj
- **Preduvjet:** Djelatnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Sustav na početnoj stranici prikazuje listu svih zaključenih zamjena
 2. Korisnik označuje sve zamjene koje je proveo
 3. Sustav miče označene zamjene sa stranice

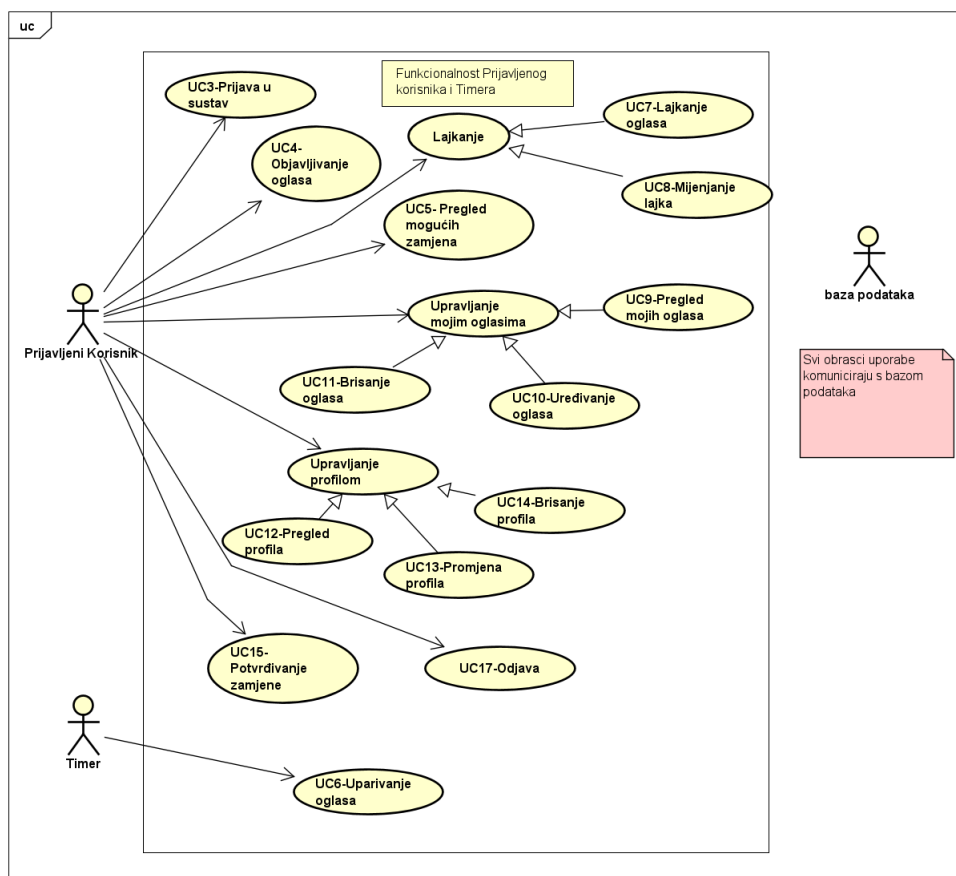
UC17 -Odjava

- **Glavni sudionik:** Djelatnik SC-a, Prijavljeni korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** poslužitelj
- **Preduvjet:** Djelatnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku "odjavi se"
 2. Sustav odjavljuje korisnika te otvara početnu stranicu

Dijagrami obrazaca uporabe



Slika 3.1: Dijagrami obrazaca uporabe, funkcionalnost neregistriranog korisnika i djelatnika SC-a

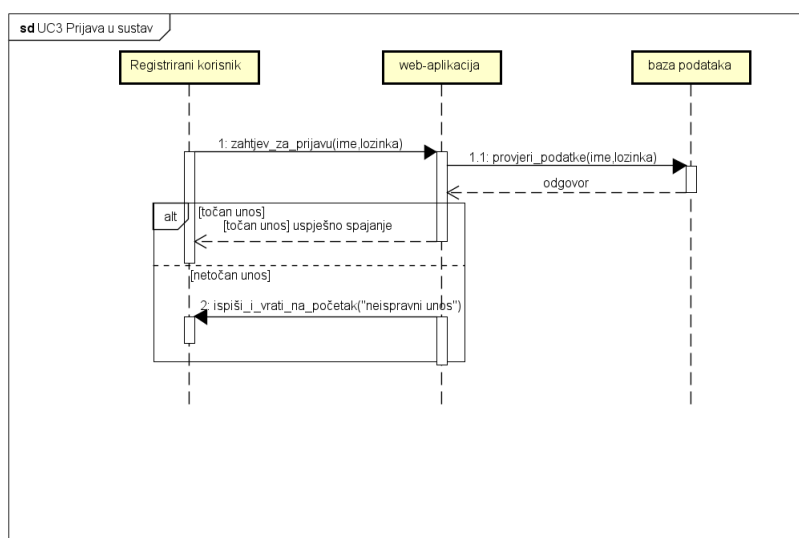


Slika 3.2: Dijagrami obrazaca uporabe, funkcionalnost prijavljenog korisnika i timera

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC3(Prijava u sustav)

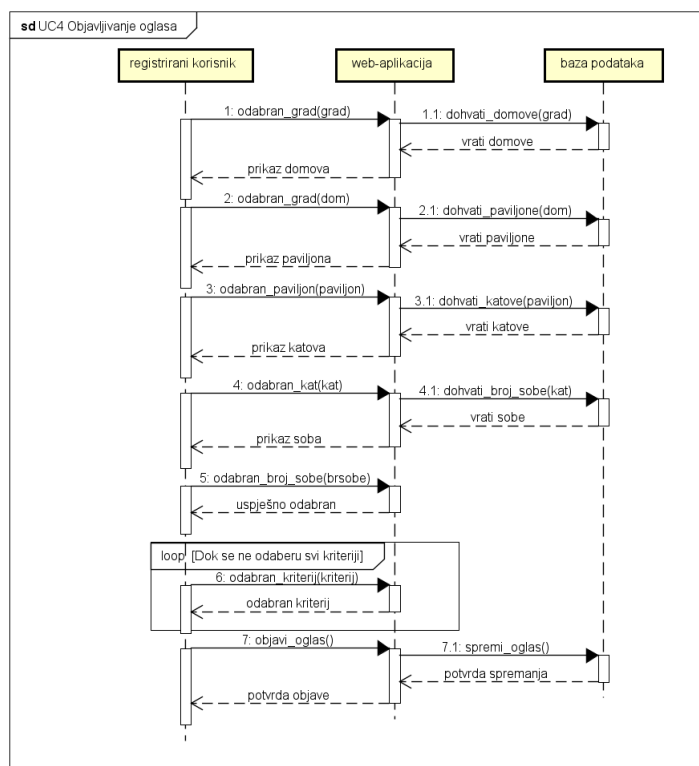
Neprijavljeni korisnik šalje zahtjev za prijavu s korisničkim imenom i lozinkom. Provjerava se ako je korisničko ime u bazi podataka. Ako ime ne postoji dojavljuje se greška u suprotnom provjerava se ako mu odgovara unesena lozinka. Ako lozinka odgovara tada se korisniku dodjeljuju ovlasti u suprotnom se dojavljuje greška.



Slika 3.3: Sekvencijski dijagram za UC3

Obrazac uporabe UC4(Objavljivanje oglasa)

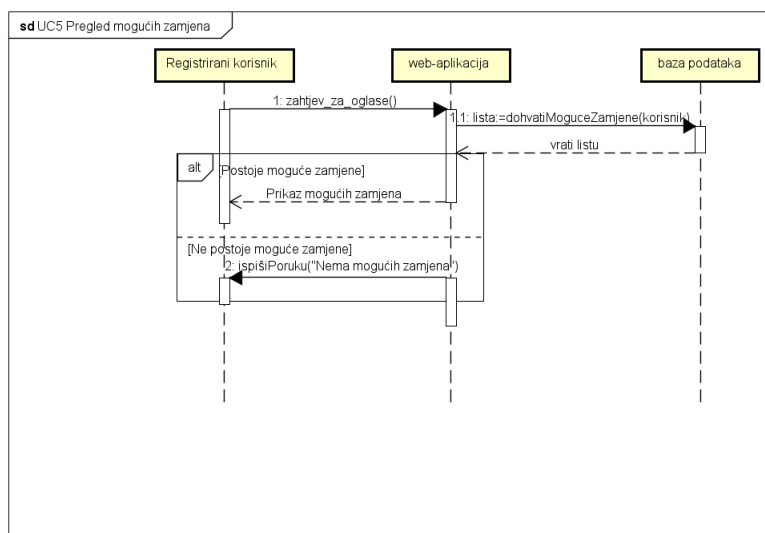
Korisnik odabire opciju "Objavi novi oglas". Zatim u obrascu iz padajućih izbornika bira grad zatim mu se nude svi domovi. Nakon odabira doma prikazuju mu se paviljoni i tako dalje za kat i broj sobe. Zatim korisnik označava iste kriterije za sobu koju traži. Pritiskom na oznaku spremi sprema se oglas u bazu podataka.



Slika 3.4: Sekvencijski dijagram za UC4

Obrazac uporabe UC5(Pregled mogućih zamjena)

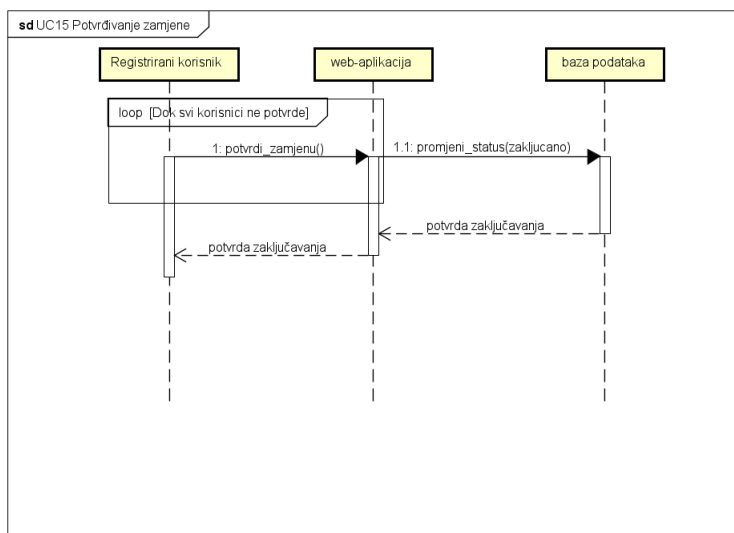
Korisnik nakon što napravi oglas odlazi na početnu stranicu gdje dobiva listu svih oglasa koji odgovaraju njegovim kriterijima.



Slika 3.5: Sekvencijski dijagram za UC5

Obrazac uporabe UC15(Potvrđivanje zamjene)

Korisnik potvrđuje zamjenu odabirom opcije "zaključaj zamjenu". Potvrda se sprema u bazu podataka promjenom statusa oglasa tek kad sve strane potvrde zamjenu.



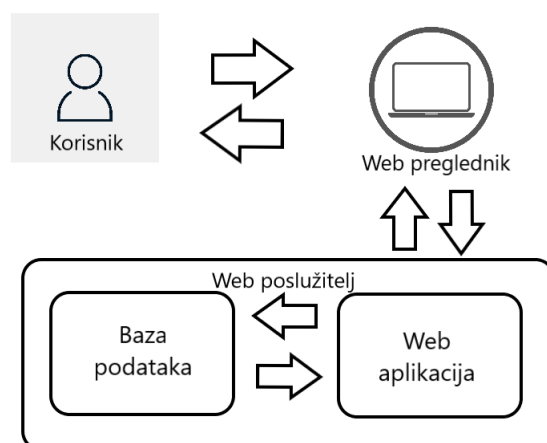
Slika 3.6: Sekvencijski dijagram za UC15

3.2 Ostali Zahtjevi

1. Sustav treba omogućiti rad više korisnika u stvarnom vremenu
2. Sustav bi se trebao moći koristiti bez dodatnih uputa
3. Sustav i korisničko sučelje trebaju podržavati hrvatsku abecedu
4. Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
5. Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
6. Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
7. Aplikacija treba biti izvedena kao web aplikacija prilagođena (engl. responsive) mobilnom uređaju

4. Arhitektura i dizajn sustava

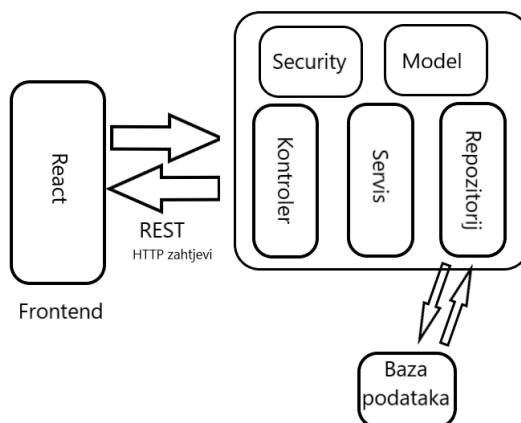
Arhitektura sustava je web aplikacija kojoj će korisnici pristupiti pomoću web preglednika. Odлучili smo se na takvu arhitekturu jer je cilj sustava da bude što jednostavniji za korištenje i da mu se može pristupiti sa svih mjesta.



Slika 4.1: Skica sustava

Programski jezik koji smo odabrali za izradu aplikacije je Java s razvojnim okvirom Spring Boot te JavaScript. Odabrano razvojno okruženje je IntelliJ IDEA. Aplikacija je organizirana u dva sloja: frontend i backend. Za izradu frontenda koristi se React - JavaScript biblioteka koja služi za izradu jednostraničnih aplikacija. Frontend i backend međusobno komuniciraju pomoću RESTa. REST se bazira na HTTP protokolu. Backend se sastoji od pet komponenti:

- Kontroler - služi za komunikaciju s frontendom. Zaprima HTTP zahtjev te određuje koja će se funkcionalnost izvršavati
- Servis - u njima se odvijaju poslovne logike i sve funkcionalnosti aplikacije
- Repozitorij - dohvaća i sprema podatke u bazu podataka
- Model - opisuju entitete iz baze
- Security - omogućava autentikaciju i autorizaciju



Slika 4.2: Skica aplikacije

4.1 Baza podataka

Za potrebe sustava za zamjenu soba koristit ćemo relacijsku bazu podataka koja nam omogućuje oblikovanje objekata iz stvarnog svijeta pomoću povezanih tablica - relacija. Svaka je tablica definirana vlastitim nazivom i skupom različitih atributa koji je opisuju. Glavna je zadaća baze podataka pohrana, brzo pronalaženje i dohvaćanje te dodavanje i brisanje podataka. Baza podataka ovog sustava sastoji se od entiteta:

- Student
- Oglas
- Soba
- Grad
- Dom
- Paviljon
- StudentskiCentar
- Obavijest
- ZaposlenikSc
- TrazeniUvjeti

- Lajk
- StudentObavijesti
- Par
- Kandidat
- DomoviUvjeti
- PaviljoniUvjeti
- TrazeniUvjetiBrojKreveta
- TrazeniUvjetiKategorija
- TrazeniUvjetiKatovi
- TrazeniUvjetiTipKupaonice

4.1.1 Opis tablica

Primarni ključevi entiteta u tablicama su označeni podebljanim fontom dok su strani ključevi označeni u nakošenom obliku.

Student Entitet sadrži informacije o korisniku aplikacije - studentu. Sadrži sljedeće atribute: identifikator studenta, korisničko ime, JMBAG, ime, prezime, e-mail adresu, lozinku, oznaku za primanje mailova, hash kod lozinke, identifikator oglasa, identifikator grada i identifikator sobe. Entitet je u vezi *Many-to-Many* s entitetom Obavijest preko identifikatora obavijesti, u vezi *One-to-One* s entitetom Oglas preko atributa identifikator oglasa, u vezi *Many-to-One* s entitetom Grad preko atributa identifikator grada te u vezi *One-to-One* s entitetom Soba preko identifikatora sobe.

Student		
idKorisnik	UUID	jedinstveni identifikator studenta (korisnika)
korisnickoIme	VARCHAR	jedinstveno korisničko ime
jmbag	VARCHAR	jedinstveni JMBAG studenta
ime	VARCHAR	ime studenta

Student		
prezime	VARCHAR	prezime studenta
email	VARCHAR	e-mail adresa studenta
hashLozinke	VARCHAR	hash lozinka
obavijestiNaMail	BOOLEAN	oznaka želi li student primiti obavijesti na mail
oglas	UUID	identifikator studentovog oglasa
idGrad	UUID	identifikator grada u kojem student stanuje
idSoba	UUID	identifikator studentove sobe

Oglas Entitet sadrži informacije koje su vezane uz oglas koji student predaje. Sadrži attribute: ID oglasa, naslov oglasa, datum objave oglasa, status oglasa te identifikator sobe, identifikator studenta i identifikator traženih uvjeta. Entitet je u vezi *One-to-One* s entitetom Student preko atributa identifiaktora studenta, u vezi *One-to-Many* s entitetom Obavijest preko atributa identifikatora oglasa, u vezi *One-to-Many* s entitetom Kandidat preko identifikatora kandidata oglasa, u vezi *One-to-One* s entitetom Soba preko identifikatora sobe te u vezi *One-to-One* s entitetom TrazeniUvjeti preko identifikatora traženih uvjeta.

Oglas		
idOglas	UUID	jedinstveni identifikator oglasa
naslov	VARCHAR	naslov oglasa
objavljen	DATE	datum objavljivanja oglasa
statusOglasa	INTEGER	status oglasa
idSoba	UUID	identifikator sobe
idStudent	UUID	identifikator studenta
idTrazeniUvjeti	UUID	identifikator traženih uvjeta

Soba Entitet sadrži informacije o sobi u studentskom domu. Sadrži attribute: identifikator sobe, kat na kojemu se soba nalazi, broj kreveta i vrsta kupaonice koja pripada sobi, dodatni komentar te identifikator paviljona i oglasa. Entitet je u vezi *Many-to-One* s entitetom Paviljon preko atributa identifikator paviljona, u vezi *One-to-One* s entitetom Oglas preko atributa identifikator oglasa te u vezi *One-to-One* s entitetom Student preko identifikatora studenta.

Soba		
idSoba	INTEGER	identifikator sobe
kat	INTEGER	kat na kojemu se soba nalazi
brojKreveta	VARCHAR	broj kreveta u sobi
tipKupaonice	VARCHAR	vrsta dostupne kupaonice
komentar	TEXT	dodatni komentar o sobi
idPaviljon	UUID	identifikator paviljona kojemu soba pripada
<i>idOglas</i>	UUID	identifikator oglasa

Grad Entitet sadrži informacije o pojedinom gradu. Sadrži attribute: identifikator grada i naziv te identifikator studentskog centra tog grada. Entitet je u vezi *One-To-One* s entitetom Studentski Centar preko atributa identifikator studentskog centra, u vezi *One-to-Many* s entitetom Dom preko identifikatora grada, u vezi *One-to-Many* s entitetom TrazeniUvjeti preko identifikatora grada te u vezi *One-to-Many* s entitetom Student preko identifikatora grada.

Grad		
idGrad	UUID	jedinstveni identifikator grada
naziv	VARCHAR	ime grada
<i>idSc</i>	UUID	identifikator gradskog studentskog centra

Dom Entitet sadrži sve važne informacije o pojedinom studentskom domu. Sadrži attribute: identifikator doma, naziv doma, identifikator grada u kojemu se dom nalazi te oznaka ima li dom vlastitu menzu. Entitet je u vezi *Many-to-One* s entitetom Grad preko atributa identifikatora grada, u vezi *One-To-Many* s entitetom Paviljon preko identifikatora doma te u vezi *Many-to-Many* s entitetom TrazeniUvjeti.

Dom		
idDom	UUID	jedinstveni identifikator studentskog doma
naziv	VARCHAR	naziv studentskog doma
imaMenzu	BOOLEAN	oznaka ima li dom vlastitu menzu

Dom		
<i>idGrad</i>	UUID	identifikator grada u kojemu se dom nalazi

Paviljon Entitet sadrži sve informacije o pojedinom paviljonu studentskog doma. Sadrži attribute: identifikator paviljona, naziv, broj katova, kategorija u koju paviljon spada te identifikator doma. Entitet je u vezi *Many-to-One* s entitetom Dom preko atributa identifikatoa doma, u vezi *One-to-Many* s entitetom Soba preko indetifikatora paviljona te u vezi *Many-to-Many* s entitetom TrazeniUvjeti.

Paviljon		
idPaviljon	UUID	jedinstveni identifikator paviljona
naziv	VARCHAR	naziv paviljona
brojKatova	INTEGER	broj katova u paviljonu
kategorija	VARCHAR	kategorija kojoj paviljon pripada
<i>idDom</i>	UUID	identifikator doma u kojemu se paviljon nalazi

StudentskiCentar Entitet sadrži informacije o studentskom centru. Sadrži attribute: identifikator studentskog centra, naziv te identifikator grada u kojemu se studentski centar nalazi. Entitet je u vezi *One-to-One* s entitetom Grad preko atributa identifikatora grada te u vezi *One-to-Many* s entitetom Zaposlenik SC preko identifikatora studentskog centra.

Studentski Centar		
idSc	UUID	jedinstveni identifikator studentskog centra
naziv	VARCHAR	ime studentskog centra
<i>idGrad</i>	UUID	identifikator grada u kojemu se nalazi studentski centar

Obavijest Entitet sadrži informacije o obavijestima koje aplikacija šalje studentima. Sadrži entitete: identifikator obavijesti, tekst, oznaka je li obavijest pročitana, vrijeme slanja obavijesti te lista studenata kojima se obavijest šalje i identifikator oglasa za koji se obavijest šalje. Entitet je u vezi *Many-to-Many* s entitetom Student

te u vezi *Many-to-One* s entitetom Oglas preko identifikatora oglasa.

Obavijest		
idObavijest	UUID	jedinstveni identifikator obavijesti
tekst	VARCHAR	tekst obavijesti
procitana	BOOLEAN	oznaka je li poruka pročitana
vrijeme	DATE	vrijeme slanja obavijesti
idOglas	UUID	identifikator oglasa za koji se obavijest generira

ZaposlenikSC Entitet sadrži informacije o zaposleniku u studentskom centru. Sadrži attribute: identifikator zaposlenika, korisničko ime i lozinka za prijavu u sustav, ime i prezime zaposlenika, e-mail adresa, hash kod lozinke, oznaka o primanju obavijesti na e-mail te identifikator studentskog centra u kojem je zaposlen. Entitet je u vezi *Many-to-One* s entitetom StudentskiCentar preko atributa identifikator studentskog centra.

ZaposlenikSc		
idZaposlenik	UUID	jedinstveni identifikator zaposlenika studentskog centra
korisnickoIme	VARCHAR	jedinstveno korisničko ime zaposlenika studentskog centra
ime	VARCHAR	ime zaposlenika studentskog centra
prezime	VARCHAR	prezime zaposlenika studentskog centra
email	VARCHAR	e-mail adresa zaposlenika studentskog centra
hashLozinke	VARCHAR	hash lozinke
obavijestiNaMail	BOOLEAN	oznaka želi li zaposlenik primiti obavijesti na mail
idSc	UUID	identifikator studentskog centra u kojemu je zaposlen

TrazeniUvjeti Entitet sadrži informacije o sobi koju student traži. Sadrži attribute: identifikator traženih uvjeta, identifikator grada i identifikator oglasa. Entitet je u vezi *Many-to-One* s entitetom Grad preko identifikatora grada, u vezi *Many-to-Many* s entitetom DomoviUvjeti te u vezi *Many-to-Many* s entitetom Pavi-

ljoniUvjeti.

TrazeniUvjeti		
idTrazeniUvjeti	UUID	jedinstveni identifikator skupa traženih uvjeta
<i>idGrad</i>	UUID	identifikator grada
<i>idOglas</i>	UUID	identifikator grada

Lajk Entitet sadrži informacije vezane uz 'lajkove' oglasa. Sadrži attribute: identifikator oglasa i identifikator studenta te ocjenu. Entitet je u vezi *Many-to-One* s entitetom Student preko identifikatora studenta i u vezi *Many-to-One* s entitetom Oglas preko identifikatora oglasa.

Lajk		
idOglas	UUID	jedinstveni identifikator oglasa koji se ocjenjuje
idStudent	UUID	jedinstveni identifikator studenta koji je 'dao lajk'
ocjena	INTEGER	iznos ocjene

StudentObavijesti Entitet sadrži informacije vezane uz obavijesti i studente koji ih primaju. Sadrži attribute: identifikator studenta i identifikator obavijesti.

StudentObavijesti		
<i>studentiIdKorisnik</i>	UUID	identifikator studenta
<i>obavijestiIdObavijest</i>	UUID	identifikator oglasa

Par Entitet sadrži informacije vezane uz par za zamjenu sobe. Sadrži attribute: identifikator para, identifikator prvog oglasa u paru, identifikator drugog oglasa u paru, identifikator zaposlenika studentskog centra koji je potvrdio zamjenu te oznake statusa zamjene - done, ignore, lanac, odobren, prihvatio prvi, prihvatio drugi. Entitet je u vezi *Many-to-One* s entitetom Oglas preko identifikatora oglasa za oba oglasa u paru te u vezi *Many-to-One* s entitetom ZaposlenikSC.

Par		
idPar	INTEGER	identifikator para
done	BOOLEAN	oznaka jesu li oba studenta prihvatila zamjenu
ignore	BOOLEAN	oznaka je li barem jedan student odbio zamjenu
lanac	BOOLEAN	oznaka je li zamjena nije obostrana nego ulančana
odobren	BOOLEAN	oznaka je li djelatnik studentskog centra potvrdio zamjenu
prihvatioDrugi	BOOLEAN	oznaka da je prvi student prihvatio zamjenu
prihvatioPrvi	BOLLEAN	oznaka da je drugi student prihvatio zamjenu
<i>idOglas1</i>	UUID	identifikator jednog oglasa u paru
<i>idOglas2</i>	UUID	identifikator drugog oglasa u paru
<i>zaposlenikKorisnickoIme</i>	UUID	identifikator zaposlenika studentskog centra koji je potvrdio zamjenu

Kandidat Entitet sadrži informacije vezane uz mogućeg 'kandidata', to jest sobe za zamjenu. Sadrži attribute: identifikator kandidata, razinu bliskosti, oznake ignore i stvoren te identifikatore oglasa kandidata i oglasa za kojeg je pronađen kandidat. Entitet je u vezama *Many-to-One* s entitetom Oglas preko identifikatora oglasa i preko identifikatora kandidata oglasa.

Kandidat		
idKandidat	UUID	identifikator kandidata
bliskost	INTEGER	broj zahtjeva koje oglas zadovoljava
ignore	BOOLEAN	oznaka odbijanja kandidata
stvoren	DATE	oznaka prihvata kandidata
<i>idKandOglasa</i>	UUID	identifikator oglasa kandidata
<i>idOglas</i>	UUID	identifikator oglasa za kojeg je pronađen kandidat

DomoviUvjeti Entitet sadrži informaciju o domu za tražene uvjete. Sadrži attribute: identifikator doma i identifikator traženih uvjeta.

DomoviUvjeti		
<i>idDom</i>	UUID	identifikator doma
<i>idTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

PaviljoniUvjeti Entitet sadrži informaciju o paviljonu za tražene uvjete. Sadrži attribute: identifikator paviljona i identifikator traženih uvjeta.

PaviljoniUvjeti		
<i>idPaviljon</i>	UUID	identifikator paviljona
<i>idTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

TrazeniUvjetiBrojKreveta Entitet sadrži informaciju o broju kreveta za tražene uvjete. Sadrži attribute: broj kreveta i identifikator traženih uvjeta.

TrazeniUvjetiBrojKreveta		
brojKreveta	INTEGER	traženi broj kreveta
<i>trazeniUvjetiIdTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

TrazeniUvjetiKategorija Entitet sadrži informacije o kategoriji sobe za tražene uvjete. Sadrži attribute: kategorija i identifikator traženih uvjeta.

TrazeniUvjetiKategorija		
kategorija	INTEGER	tražena kategorija sobe
<i>trazeniUvjetiIdTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

TrazeniUvjetiKatovi Entitet sadrži informaciju o katu na kojemu se nalazi soba za tražene uvjete. Sadrži attribute: katovi i identifikator traženih uvjeta.

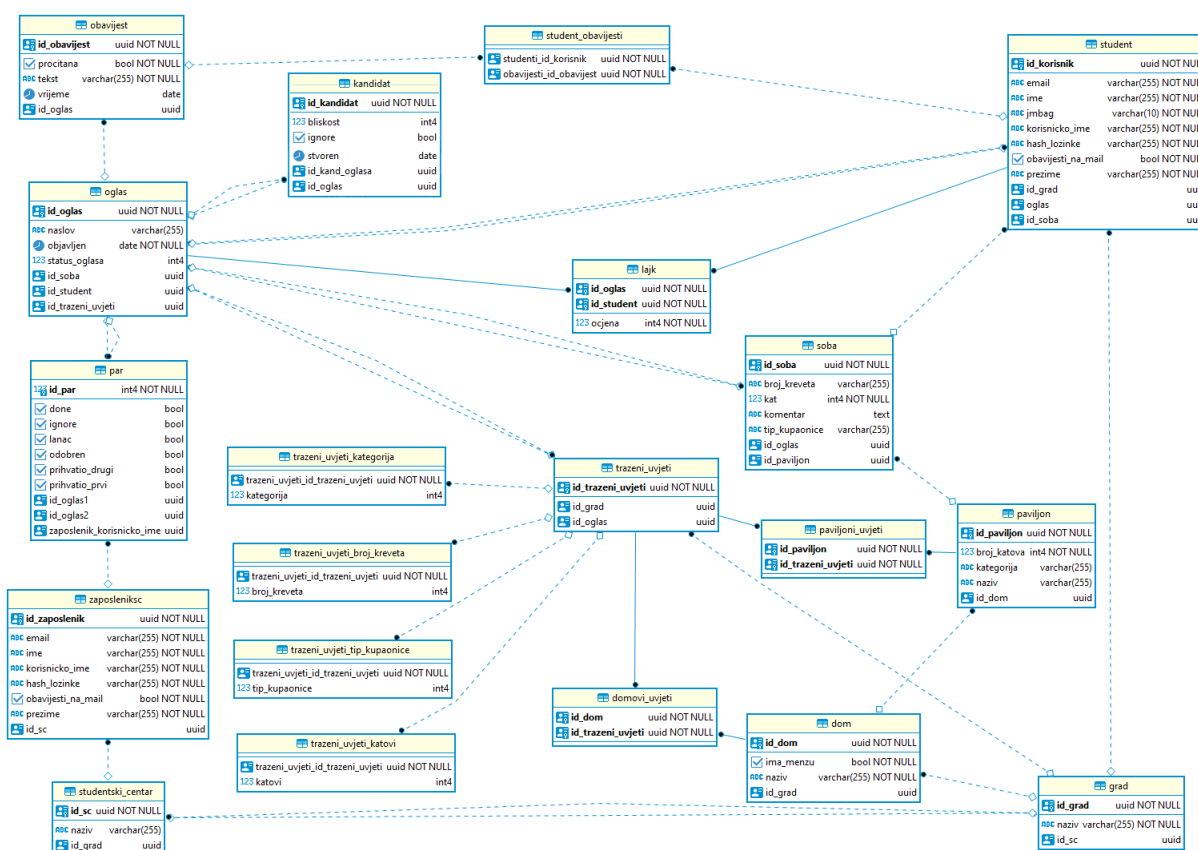
TrazeniUvjetiKatovi		
katovi	INTEGER	traženi kat na kojemu se nalazi soba

TrazeniUvjetiKatovi		
<i>trazeniUvjetiIdTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

TrazeniUvjetiTipKupaonice Entitet sadrži informaciju o tipu kupaonice za tražene uvjete. Sadrži attribute: tip kupaonice i identifikator traženih uvjeta.

TrazeniUvjetiTipKupaonice		
tipKupaonice	INTEGER	traženi tip kupaonice
<i>trazeniUvjetiIdTrazeniUvjeti</i>	UUID	identifikator traženih uvjeta

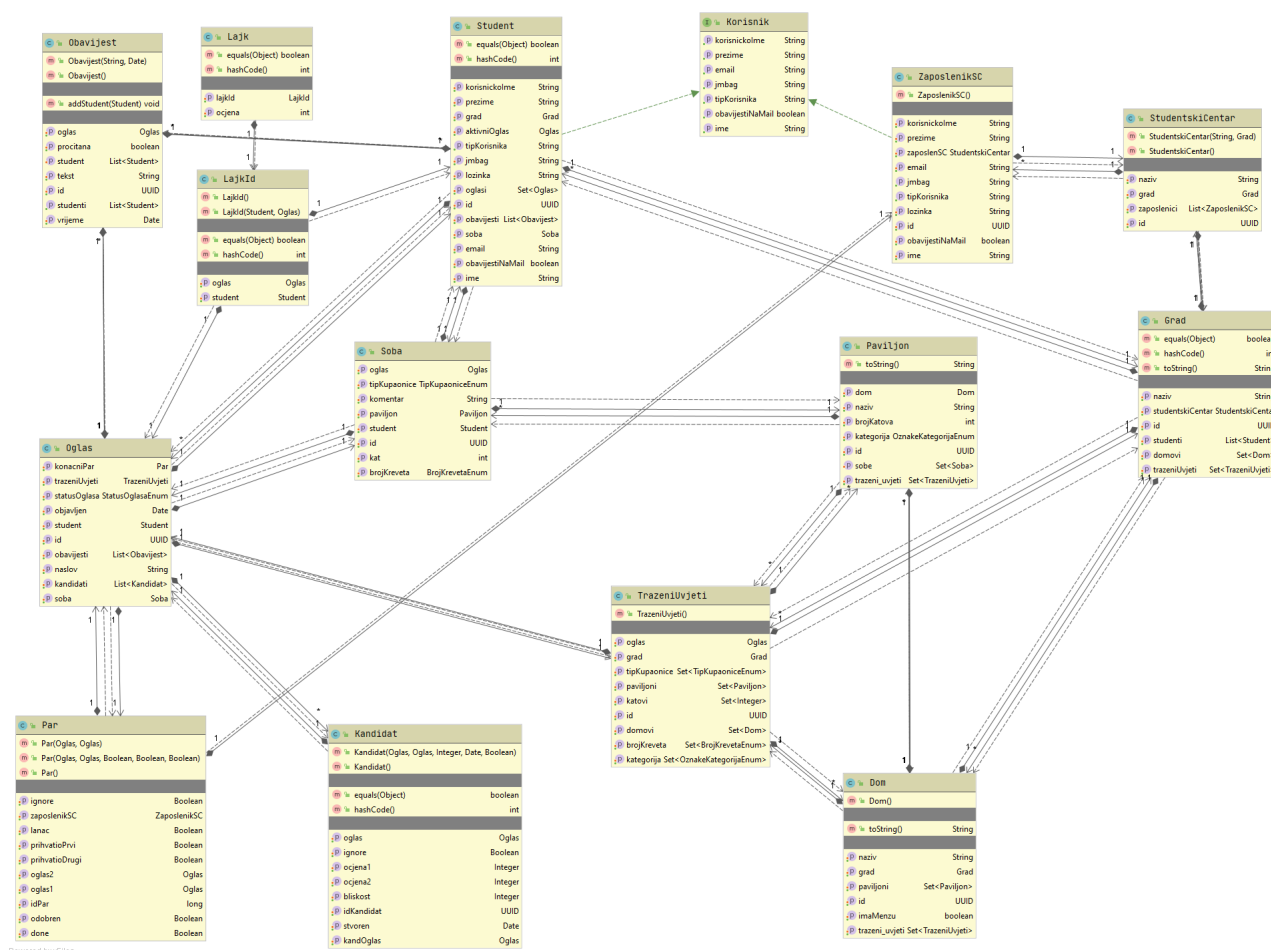
4.1.2 Dijagram baze podataka



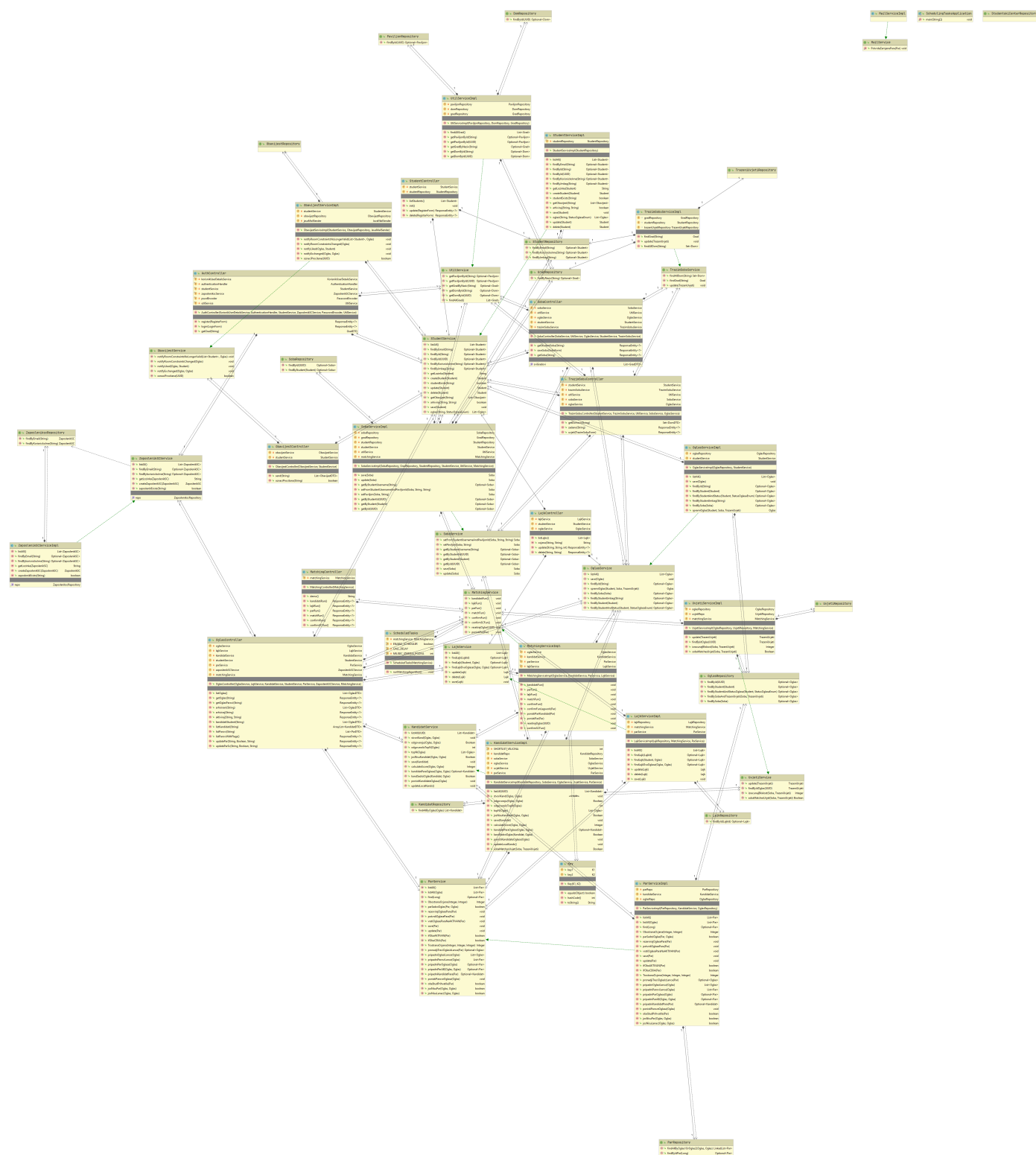
Slika 4.3: ER dijagram baze podataka

4.2 Dijagram razreda

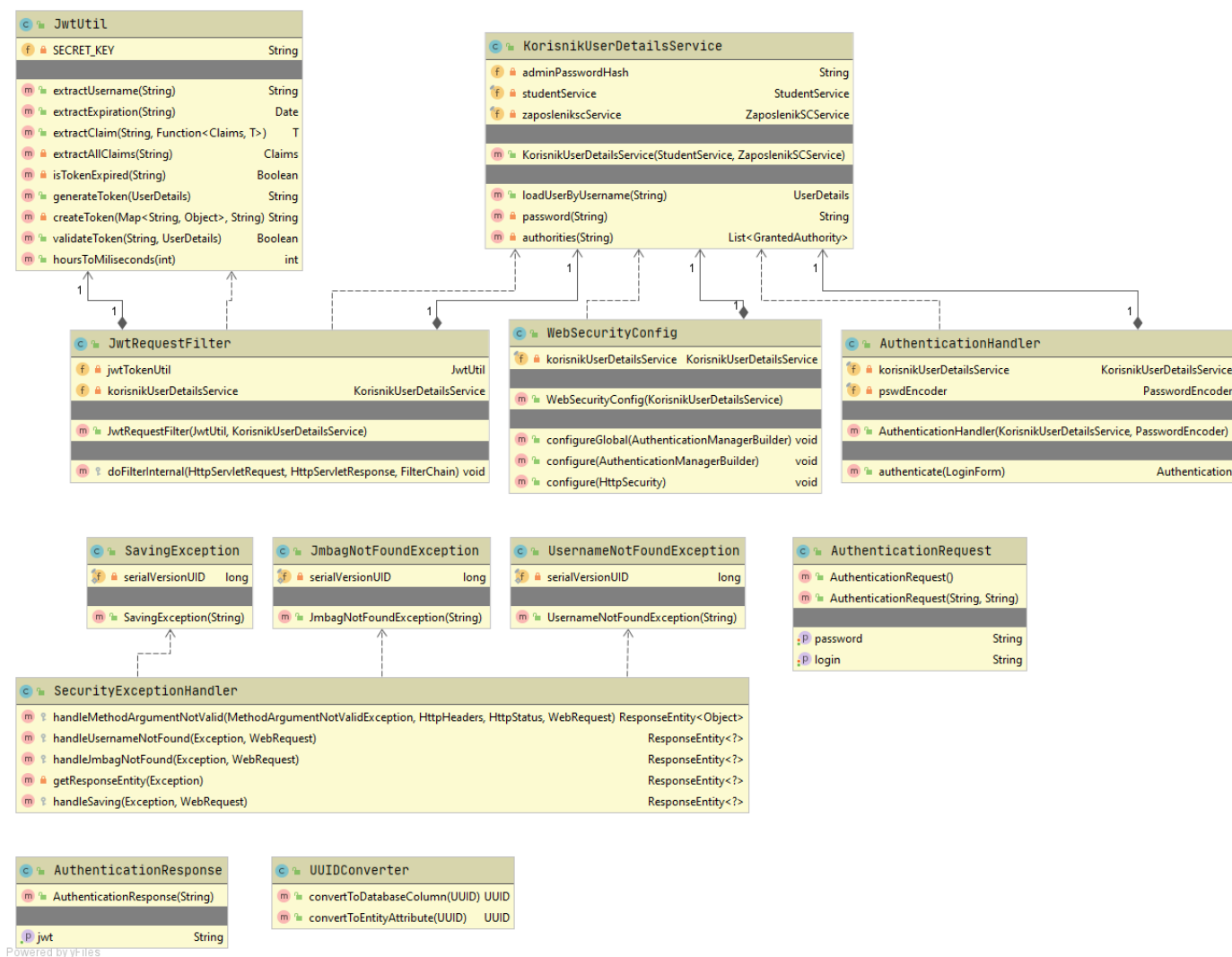
Na slikama su prikazani razredi backenda. Na slici 4.4. prikazani su razredi Modela. Modeli opisuju entitete u bazi podataka. Na slici 4.5. prikazani su dijelovi Kontroler, Servis i Repozitorij. Kontroler `AuthController` manipulira s `KorisnikDTO` što je *Data transfer object* (DTO) kojeg šaljemo na frontend kako bi mogli pamtit i trenutnog korisnika. Varijable `KorisnikDTOa` su dohvaćene metodama iz razreda `Model`. Kontroleri također pozivaju servise. Servisi obavljaju logiku aplikacije te za to koriste podatke iz razreda `Model`. Kontroleri i servisi pozivaju Repozitorij. To su sučelja koja nasljeđuju sučelje `JpaRepository` koje ima ugrađene metode za dohvat i spremanje podataka u bazu. Na slici 4.6. prikazan je dio `Security`. Razrede iz dijela `Security` pozivaju kontroleri za autentikaciju podataka.



Slika 4.4: Dijagrami razreda - dio Model



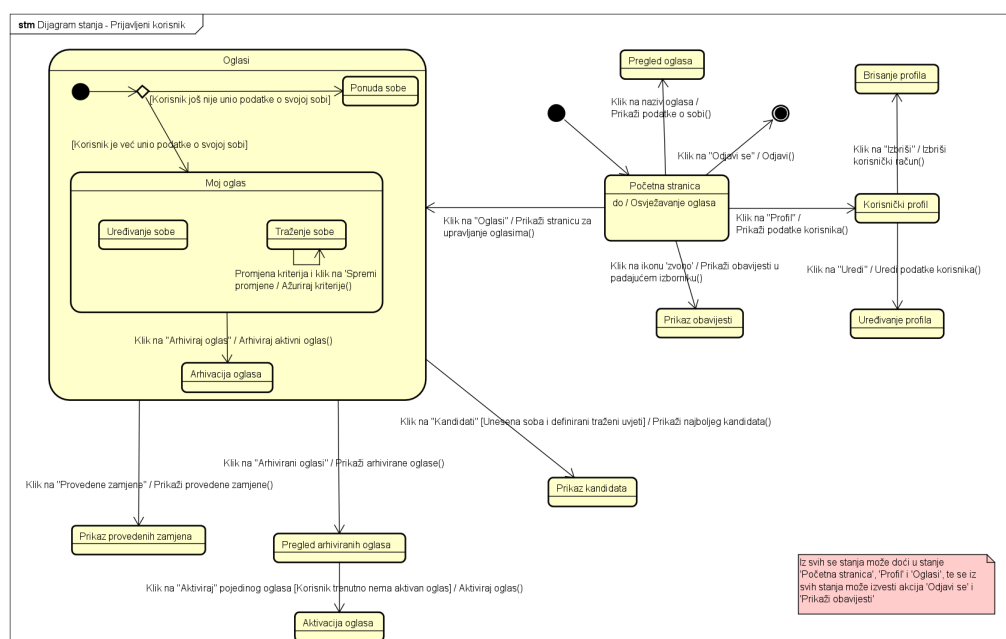
Slika 4.5: Dijagrami razreda - dio Kontroler, Servis i Repozitorij



Slika 4.6: Dijagrami razreda - dio Security

4.3 Dijagram stanja

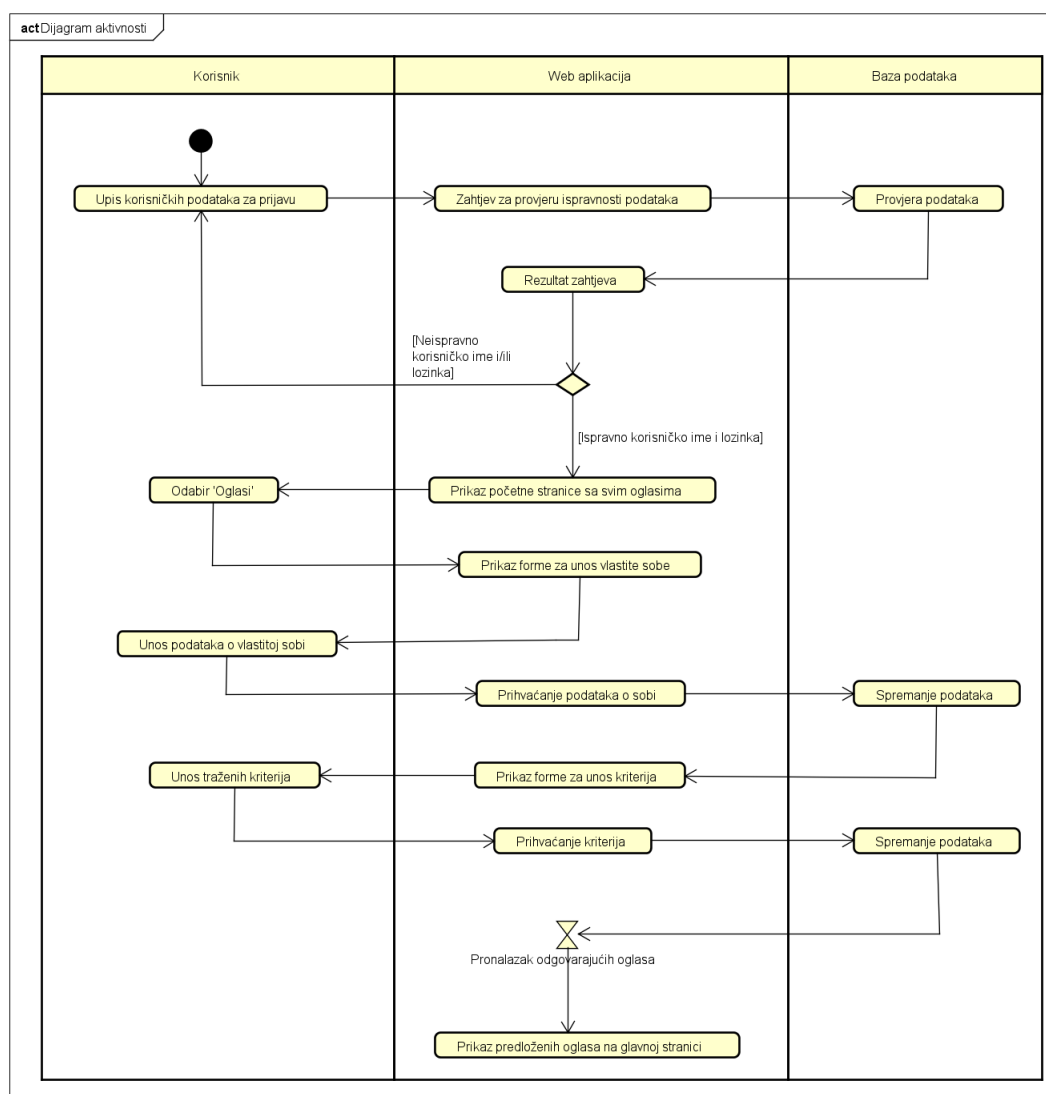
Dijagram stanja ponašajni je UML dijagram koji opisuje dinamično ponašanje nekog dijela sustava prikazom konačnog broja stanja objekata te prijelaze između stanja koji su potaknuti događajima. Na dijagramu na slici 4.7 prikazan je dijagram stanja za prijavljenog korisnika. Kada se korisnik prijavi (ili registrira) prikazuje mu se početna stranica na kojoj su mu vidljivi svi oglasi i oglasi koji odgovaraju njegovim kriterijima, ako ih je postavio. Za svaki oglas korisnik može vidjeti dodatne informacije o sobi koja se nudi. Klikom na "Profil" prikazuju se svi korisnički podaci. Korisnik te podatke može ažurirati (osim podatka JMBAG te korisničkog imena). Također može izbrisati svoj profil. Klikom na ikonu zvona korisnik u padajućem izborniku može vidjeti nepročitane i pročitane obavijesti (ili poruku da nema obavijesti). Ako korisnik prethodno nije unio podatke o svojoj sobi, klikom na "Oglasi" prikazuje mu se forma za ponudu sobe u kojoj upisuje podatke o svojoj sobi. Nakon potvrde unosa prikazuju mu se dvije forme, 'Nudim sobu' i 'Tražim sobu'. U formi za ponudu sobe automatski su uneseni podaci o sobi korisnika, a u formi za traženje sobe korisnik može postaviti željene kriterije. Podaci i o sobi i o kriterijima mogu se promijeniti klikom na "Spremi promjene". Sa stranice Oglasi korisnik klikom na "Kandidati" može vidjeti najbolje kandidate za zamjenu, klikom na "Provedene zamjene" može vidjeti sve prošle zamjene dok mu se klikom na "Arhivirani oglasi" prikazuju prethodno arhivirani oglasi. Ako trenutno ne postoji aktivan oglas, korisnik može aktivirati jedan od arhiviranih oglasa klikom na "Aktiviraj".



Slika 4.7: Dijagram stanja

4.4 Dijagram aktivnosti

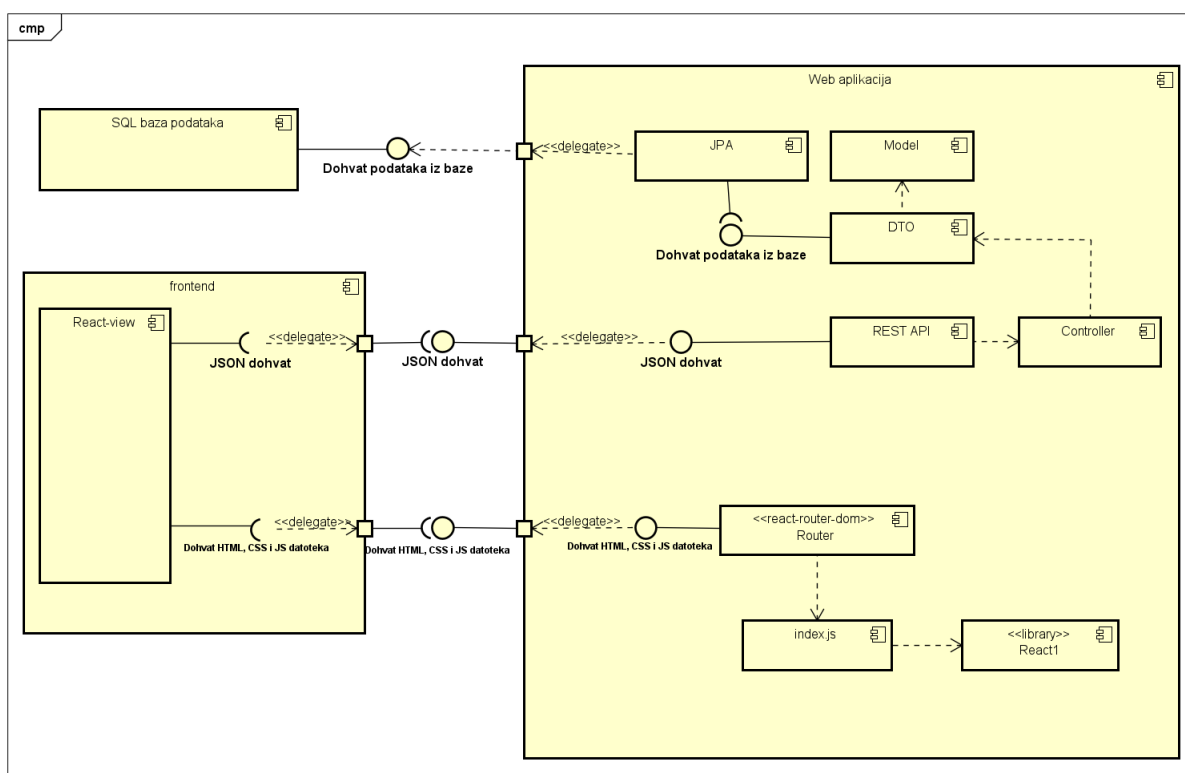
Dijagram aktivnosti ponašajni je UML dijagram kojim se modelira upravljački i podatkovni tok. Kod modeliranja toka upravljanja koristi se *pull* način djelovanja - novi korak slijedni nakon završenog prethodnog. Na dijagramu na slici 4.8 prikazana je aktivnost predaje oglasa koji predaje korisnik koji prethodno nije unio podatke o svojoj sobi. Korisnik se prijavljuje u sustav, odabire stranicu 'Oglasi' te u formu upisuje podatke o svojoj sobi. Nakon toga prikazuje mu se forma za unos željenih kriterija. Korisnikov oglas se pohranjuje te sustav pronalazi odgovarajuće oglase i prikazuje ih korisniku na početnoj stranici.



Slika 4.8: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti strukturni je UML dijagram koji prikazuje organizaciju i međuovisnosti između implementacijskih komponenata i odnos programske potpore prema okolini. Na dijagramu na slici 4.9 prikazana su sučelja i komponente koje čine arhitekturu web aplikacije i poveznice među njima. Arhitektura je organizirana kroz dva sučelja, za *frontend* i *backend* dio aplikacije. Datoteke za *frontend* dio poslužuju se preko sučelja za dohvat HTML, CSS i JS datoteka, a koja će se datoteka poslužiti odlučuje Router na temelju URL-a upita. Komponenta REST API definirana je sučeljem za dohvat JSON podataka, a služi za povezivanje *frontend* i *backend* dijela. JPA komponenta služi za transformaciju podataka između relacijskih baza i objekata u Javi. Podaci iz baze dalje se proslijeđuju u DTO obliku. Na temelju akcija korisnika komponenta React-view komunicira s web aplikacijom i ažurira prikaz.



Slika 4.9: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija među članovima projektnog tima ostvarena je pomoću platforme Microsoft Teams¹ preko koje su održavani video sastanci te pomoću mobilne aplikacije za razmjenu poruka WhatsApp². Za pisanje dokumentacije korišten je programski jezik za dokumente LaTeX³ i LaTeX uređivač Texstudio⁴, za izradu različitih UML dijagrama korišten je alat Astah UML⁵, dok je za izradu ER dijagrama baze podataka korišten alat za baze podataka DBeaver⁶. Za upravljanje inačicama datoteka projekta korišten je distribuirani sustav Git⁷, a repozitorij projektne grupe uspostavljen je na GitLabu⁸. GitLab je web hosting usluga koja podržava rad sustava Git.

Za razvojno okruženje korišten je IntelliJ IDEA⁹, integrirano razvojno okruženje (IDE) koje je razvila tvrtka JetBrains. Ponajviše je namijenjeno za Javu, no podržava i mnogo drugih programskih jezika kao što su Kotlin, Scala, Python i drugi. Koristi se za razvoj, modeliranje i *deployment* web i mobilnih aplikacija.

Za izradu web aplikacije korišten je radni okvir Spring Boot¹⁰ uz arhitekturni stil Rest API¹¹ koji se bazira na HTTP protokolu. Spring Boot je specijalizacija radnog okvira Spring a omogućuje jednostavnije i brže oblikovanje web aplikacija tako što u svojoj automatskoj konfiguraciji neke uobičajene dijelove web aplikacije već ima podešeno, kao na primjer pokretanje Apache Tomcat poslužitelja. Od programskih jezika koriste se objektno orijentirani jezik Java¹² za izradu *backend* sloja

¹<https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-software>

²<https://www.whatsapp.com>

³<https://www.latex-project.org/>

⁴<https://www.texstudio.org/>

⁵<https://astah.net/products/astah-community/>

⁶<https://dbeaver.io/>

⁷<https://git-scm.com/>

⁸<https://about.gitlab.com/>

⁹<https://www.jetbrains.com/idea/>

¹⁰<https://spring.io/projects/spring-boot>

¹¹<https://restfulapi.net/>

¹²<https://www.java.com/en/>

aplikacije te skriptni programski jezik JavaScript¹³ te knjižnica React¹⁴ za izradu *frontend* sloja. Uz Javu smo koristili Apache Maven¹⁵, *open source* alat za tehničko upravljanje Java projektima. Knjižnicu React razvija tvrtka Facebook a pisana je u jeziku JavaScript i sadrži mnogo paketa te je tako fleksibilna za različite aplikacije.

Za rad s bazom podataka korišten je sustav za upravljanje bazom podataka PostgreSQL¹⁶ te alat za mapiranje podataka između relacijske baze i objekata u Javi Hibernate ORM¹⁷. Hibernate implementira standard Java Persistence API te nudi anotacije koje služe za lakši rad s entitetima.

Baza podataka nalazi se na poslužitelju Heroku¹⁸

¹³<https://www.javascript.com/>

¹⁴<https://reactjs.org/>

¹⁵<https://maven.apache.org/>

¹⁶<https://www.postgresql.org/>

¹⁷<https://hibernate.org/orm/>

¹⁸<https://www.heroku.com/>

5.2 Ispitivanje programskog rješenja

dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

Ispitni slučaj: UC2 - Registracija, osnovni tijek

- **Ulaz:**
 1. Korisnik otvara stranicu za registraciju
 2. Korisnik unosi svoje podatke
 3. Korisnik pritišće tipku "Registriraj se"
- **Očekivani rezultat:**
 1. Korisnik se uspješno registrirao
 2. Korisnik je preusmjeren na početnu stranicu
- **Rezultat:**
 1. Korisnik je registriran i preusmjeren na početnu stranicu

```
1. open on / OK
2. setWindowSize on 1288x1040 OK
3. click on linkText=Registracija OK
4. click on name=ime OK
5. type on name=ime with value Ivan OK
6. click on name=prezime OK
7. type on name=prezime with value Kovačić OK
8. click on name=jmbag OK
9. type on name=jmbag with value 0036517345 OK
10. click on name=username OK
11. type on name=username with value ivank OK
12. click on name=email OK
13. type on name=email with value ivan.kovacic@gmail.com OK
14. click on name=lozinka OK
15. type on name=lozinka with value lozinka OK
16. click on name=grad OK
17. select on name=grad with value label=Osijek OK
18. click on name=grad OK
19. click on css=.btn-dark OK
20. runScript on window.scrollTo(0,96) OK
'Test1' completed successfully
```

(a) Prvi ispitni slučaj: provedeni test

Ispitni slučaj: UC2 - Registracija, odstupanje

- **Ulaz:**

1. Korisnik otvara stranicu za registraciju
2. Korisnik unosi podatke i izabire korisničko ime koje već postoji
3. Korisnik pritišće tipku "Registriraj se"

- **Očekivani rezultat:**

1. Sustav vraća korisnika na stranicu za registraciju s upisanim podacima i porukom o pogrešci

- **Rezultat:**

1. Registracija nije uspješna i korisnik je vraćen na stranicu za registraciju s upisanim podacima i porukom o pogrešci

Registracija

Ime

Prezime

JMBAG

JMBAG se sastoji od 10 znamenki

Korisničko ime

Email

Lozinka

Lozinka se treba sastojati od najmanje 5 znakova!

U kojem gradu studiraš?

Već postoji korisnik s tim imenom!

Registriraj se
Već si registriran?

```

1. open on / OK
2. setWindowSize on 1288x1040 OK
3. click on linkText=Registracija OK
4. click on name=ime OK
5. type on name=ime with value Ivan OK
6. click on name=prezime OK
7. type on name=prezime with value Kraljević OK
8. click on name=jmbag OK
9. type on name=jmbag with value 0036517658 OK
10. click on name=username OK
11. type on name=username with value ivank OK
12. click on name=email OK
13. type on name=email with value ivan.kraljevic@gmail.com OK
14. click on name=lozinka OK
15. type on name=lozinka with value lozinka OK
16. click on name=grad OK
17. select on name=grad with value label=Zagreb OK
18. click on name=grad OK
19. click on css=.btn-dark OK
20. runScript on window.scrollTo(0,96) OK
'Test1 (1)' completed successfully

```

(a) Drugi ispitni slučaj: rezultat

(b) Drugi ispitni slučaj: provedeni test

Ispitni slučaj: UC3 - Prijava u sustav, osnovni tijek

- **Ulaz:**

1. Korisnik otvara stranicu za prijavu
2. Korisnik unosi svoje podatke
3. Korisnik pritišće tipku "Prijavi se"

- **Očekivani rezultat:**

1. Korisnik se uspješno prijavio
2. Korisnik je preusmjeren na početnu stranicu

- **Rezultat:**

1. Korisnik je prijavljen i preusmjeren na početnu stranicu

```

1. open on / OK
2. setWindowSize on 1353x1040 OK
3. click on linkText=Prijava OK
4. click on name=username OK
5. type on name=username with value ivank OK
6. click on name=password OK
7. type on name=password with value lozinka OK
8. click on css=.btn-dark OK
'UC3PrijavaOsnovno' completed successfully

```

(a) Treći ispitni slučaj: provedeni test

Ispitni slučaj: UC3 - Prijava u sustav, odstupanje

- **Ulaz:**

1. Korisnik otvara stranicu za prijavu

2. Korisnik unosi korisničko ime koje ne postoji

3. Korisnik pritišće tipku "Prijavi se"

- **Očekivani rezultat:**

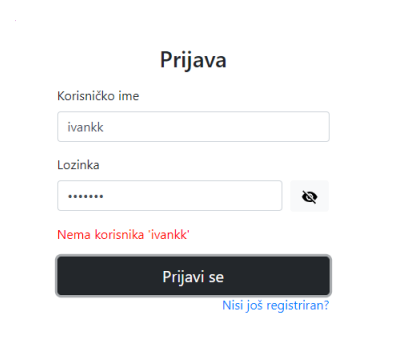
1. Sustav javlja poruku o pogrešci

2. Prijava nije uspješna

- **Rezultat:**

1. Sustav javlja poruku o nepostojanju korisničkog imena

2. Prijava nije uspješna



1. open on / OK
2. setWindowSize on 1288x1040 OK
3. click on linkText=Prijava OK
4. click on name=username OK
5. type on name=username with value ivankk OK
6. click on css=form OK
7. click on name=password OK
8. type on name=password with value lozinka OK
9. click on css=.btn-dark OK
'PrijavaNepostojeciRacun' completed successfully

(a) Četvrti ispitni slučaj: rezultat

(b) Četvrti ispitni slučaj: provedeni test

Ispitni slučaj: UC13 - Promjena profila, osnovni tijek

- **Ulaz:**

1. Korisnik otvara stranicu profila

2. Korisnik pritišće tipku 'Uredi'

3. Korisnik mijenja svoje podatke

4. Korisnik pritišće tipku "Ažuriraj podatke"

- **Očekivani rezultat:**

1. Ažurirani podaci vidljivi su na stranici profila

- **Rezultat:**

1. Ažurirani podaci vidljivi su na stranici profila

Moj profil

Ime

Ivan

Prezime

Kovačić

JMBAG

0036517345

Korisničko ime

ivank

Email

ivan.kovac1@gmail.com

Obavijesti na mail

Ne

Uredi

Izbriši

Moj profil

Ime

Ivan

Prezime

Kovačić

JMBAG

0036517345

Korisničko ime

ivank

Email

ivan.kovacic@gmail.com

Obavijesti na mail

Da

Uredi

Izbriši

1. open on / OK
2. setWindowSize on 1353x1040 OK
3. click on linkText=Profil OK
4. click on css=.btn-dark:nth-child(1) OK
5. click on name=email OK
6. type on name=email with value ivan.kovacic@gmail.com OK
7. click on name=obavijestiNaMail OK
8. click on css=.btn-dark OK

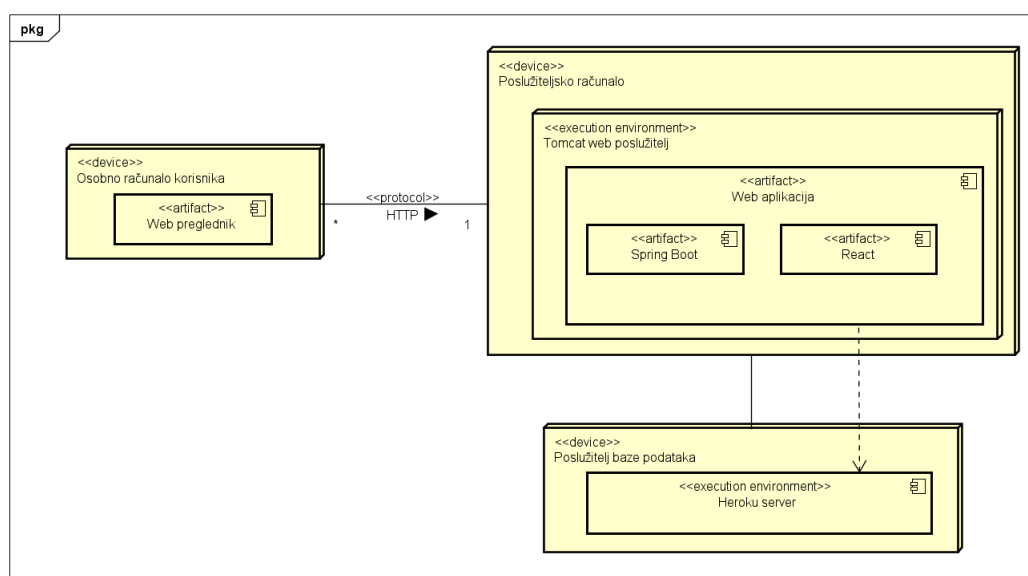
'UC13PromjenaProfila' completed successfully

(b) Peti ispitni slučaj: provedeni test

(a) Peti ispitni slučaj: rezultat

5.3 Dijagram razmještaja

Dijagram razmještaja statički je UML dijagram koji prikazuje i opisuje topologiju sustava te programsku potporu za implementaciju sustava. Korisnik, to jest klijent na svom osobnom računalu koristi web preglednik za pristup web aplikaciji. Na poslužiteljskom računalu nalazi se Tomcat web poslužitelj te PostgreSQL poslužitelj baze podataka. Komunikacija između klijenta i poslužitelja odvija se preko protokola aplikacijskog sloja HTTP.



Slika 5.6: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Git

Ako sustav Git nije instaliran, potrebno je preuzeti instalacijski paket s poveznice <https://git-scm.com/downloads>.

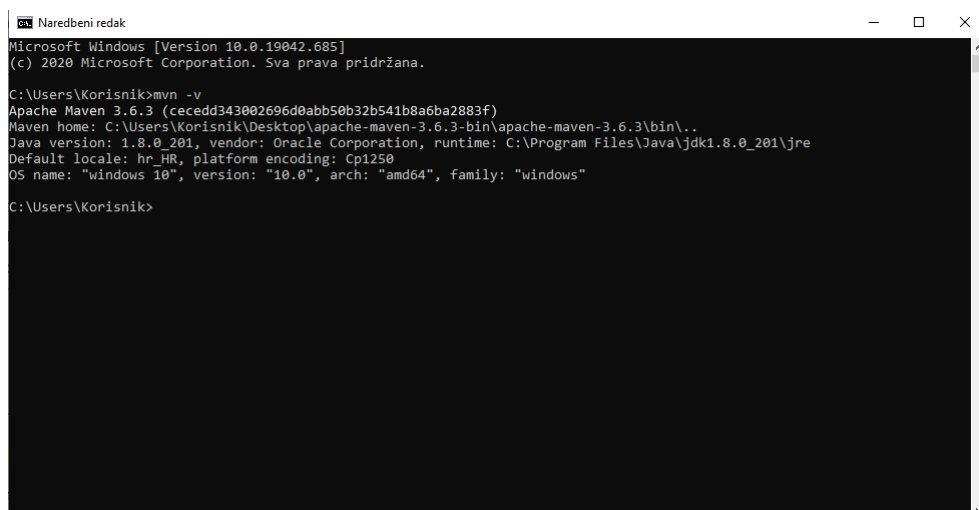
- Instalacija
 1. Pokrenuti preuzeti instalacijski program čime se otvara čarobnjak za konfiguraciju
 2. Odabrati direktorij za instalaciju
 3. Ostaviti zadane izabrane komponente
 4. Za path varijablu odabrati *Git from the command line and also from 3-rd party software*
 5. Odabrati *Use the OpenSSL library*
 6. Odabrati *Checkout windows-style, commit Unix-style line endings* (za operacijski sustav Windows)
 7. Odabrati *Use Windows' default console window*
 8. Odabrati *Default (fast-forward or merge)*
 9. Ostaviti zadane dodatne postavke
 10. Pritisnuti *Install*
- Konfiguracija

Detalji konfiguracije mogu se proučiti na stranici [First-Time Git Setup](#)

Apache Maven

Ako alat Apache Maven nije instaliran, potrebno je preuzeti instalacijski paket s poveznice <https://maven.apache.org/download.cgi>

1. Preuzeti *Binary zip archive*
2. Raspakirati distribuciju arhive u željeni direktorij
3. U Windows Start upisati i odabrati *Uređivanje varijabli okruženja sustava (Edit the system environment variables)*
4. Odabrati *Varijable okruženja (Environment variables)*
5. U PATH sistemsku varijablu dodati bin direktorij stvorenog apache-maven direktorija
6. U naredbenom retku naredbom `mvn -v` provjeriti uspješnost instalacije



```
Naredbeni redak
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. Sva prava pridržana.

C:\Users\Korisnik>mvn -v
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\Users\Korisnik\Desktop\apache-maven-3.6.3-bin\apache-maven-3.6.3\bin\..
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_201\jre
Default locale: hr HR, platform encoding: Cp1250
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Korisnik>
```

Slika 5.7: Ispravna instalacija

Heroku klijent

Na stranici <https://www.heroku.com/> potrebno je napraviti korisnički račun. Nakon toga treba preuzeti instalacijski paket s poveznice <https://devcenter.heroku.com/articles/heroku-cli>.

1. Pokrenuti preuzeti instalacijski program čime se otvara čarobnjak za konfiguraciju
2. Ostaviti zadane komponente
3. Odabrati direktorij za instalaciju
4. U naredbenom retku naredbom `heroku version` provjeriti uspješnost instalacije

Postupak puštanja u pogon

- Stvaranje Heroku servera za backend dio
 1. U naredbenom retku upisati `heroku login` i ulogirati se
 2. Upisati `heroku create <NAME-OF-APP>`
- Stvaranje baze
 1. Otvoriti stranicu <https://www.heroku.com/>
 2. Kliknuti na `NAME-OF-APP` server
 3. Kliknuti na *Resources*
 4. Odabrati *Find more add-ons*
 5. Pronaći i odabrati *Heroku POSTGRES*, odabrati plan *Hobby Dev - Free*

- Konfiguracija backenda
 1. Kliknuti na NAME-OF-APP server
 2. Kliknuti na *Resources*
 3. Kliknuti na *Heroku POSTGRES*
 4. Kliknuti na *Settings* i *View Credentials*
 5. U projektnom direktoriju otvoriti `src/main/resources/application.properties`
 6. U datoteci podesiti podatke s *View Credentials*:
 - `spring.datasource.url=jdbc:postgresql://<Host>`
 - `spring.datasource.username=<User>`
 - `spring.datasource.password=<Password>`
- Puštanje backenda u pogon
 1. U projektnom direktoriju otvoriti datoteku `pom.xml`
 2. U datoteku dodati *Maven Plugin Configuration*

```
<plugin>
<groupId>com.heroku.sdk</groupId>
<artifactId>heroku-maven-plugin</artifactId>
<version>3.0.3</version>
<configuration>
  <appName><NAME-OF-APP>-backend</appName>
  <includeTarget>false</includeTarget>
  <includes>
    <include>target/${project.build.finalName}.jar</include>
  </includes>
  <jdkVersion>${java.version}</jdkVersion>
  <processTypes>
    <web>java $JAVA_OPTS -jar target/${project.build.finalName}.jar</web>
  </processTypes>
</configuration>
</plugin>
```
 3. U naredbenom retku pozicionirati se u *root* backenda projekta
 4. U naredbeni redak upisati `mvn clean heroku:deploy`
- Stvaranje Heroku servera za frontend dio
 1. U naredbenom retku upisati `heroku login` i ulogirati se

2. Upisati heroku `create <NAME-OF-FRONTEND-APP> --buildpack https://github.com/mars/create-react-app-buildpack.git`
 3. Stvoriti prazan direktorij
 4. U naredbenom retku pozicionirati se u stvoreni direktorij i ipisati `git init`
 5. U naredbenom retku upisati heroku `git:remote -a <NAME-OF-FRONTEND-APP>`
 6. U direktorij kopirati *root* frontenda projekta
 7. U naredbenom retku upisati `git add`.
 8. U naredbenom retku upisati `git commit -m "Initital commit"`
 9. U naredbenom retku upisati `git push heroku master`
- Napomena: Za puštanje u pogon frontend dijela u *rootu* front direktorija projekta treba se nalaziti datotetka *static.json* u kojoj treba biti:

```
{
  "root": "build/",
  "clean_urls": false,
  "routes": {
    "/*": "index.html"
  }
}
```

6. Zaključak i budući rad

Naš projektni zadatak bio je razviti web aplikaciju pod nazivom "Zamjena sobe" koja bi studentima omogućila organiziranu razmjenu soba u studentskim domovima, a djelatnicima u studentskim centrima omogućila izravan pristup zamjenama koje mogu provesti. Tijekom dva ciklusa naš je tim uložio mnogo vremena i truda u ostvarenje projektnog cilja. Rad na projektu bio je ostvaren kroz dvije glavne faze.

Prva je faza obuhvatila upoznavanje i formiranje razvojnog tima, upoznavanje s dodijeljenim zadatkom te planiranje rada i podjela uloga. U toj su fazi razrađeni funkcionalni i nefunkcionalni zahtjevi koji su, uz tekstualne opise, bili popraćeni i UML dijagramima obrazaca uporabe, sekvencijskim dijagramima, dijagramima razreda te ER dijagramom baze podataka. Uz razradu zahtjeva, definirala se i arhitektura sustava kao i tehnologije koje će se koristiti za samu implementaciju. Ova je faza postavila temelje razvoja i olakšala daljnji rad na našoj aplikaciji.

Druga je faza obuhvatila implementaciju rješenja, testiranje rješenja i izradu konačne dokumentacije. Dokumentirani su UML dijagrami stanja, aktivnosti, komponenti i razmještaja koji prikazuju ključne dijelove programskog rješenja. Bez obzira na dobro definiranu strukturu projekta u prvoj fazi, implementacija je zahtijevala mnogo truda i konstantne komunikacije između članova tima.

Komunikacija se odvijala putem *WhatsApp* za brže konzultacije i dogovaranje online sastanaka koji su se održavali putem *Teamsa*. U timu je vladala pozitivna atmosfera što je olakšavalo rad.

Sudjelovanje i rad na projektu pokazalo se kao vrlo poučno i korisno iskustvo. Članovi tima upoznali su se s mnogim novim alatima i okruženjima, ali i s radom u tim s kojim se dosada u ovoj mjeri nisu susreli. Članovi tima zadovoljni su što su uspjeli ostvariti svoj cilj te što su usvojili mnoge vještine koje će im koristiti u daljnjem obrazovanju i budućem radu.

Popis literature

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

Indeks slika i dijagrama

2.1	Prikaz katova i dostupnih slobodnih soba na pojedinom katu	6
2.2	Tlocrt odabranog kata	6
2.3	Detalji o sobi i studentima u sobi	7
2.4	Odbrojanje i konačni odabir	7
3.1	Dijagrami obrazaca uporabe, funkcionalnost neregistriranog korisnika i djelatnika SC-a	15
3.2	Dijagrami obrazaca uporabe, funkcionalnost prijavljenog korisnika i timera	16
3.3	Sekvencijski dijagram za UC3	17
3.4	Sekvencijski dijagram za UC4	18
3.5	Sekvencijski dijagram za UC5	18
3.6	Sekvencijski dijagram za UC15	19
4.1	Skica sustava	20
4.2	Skica aplikacije	21
4.3	ER dijagram baze podataka	30
4.4	Dijagrami razreda - dio Model	31
4.5	Dijagrami razreda - dio Kontroler, Servis i Repozitorij	32
4.6	Dijagrami razreda - dio Security	33
4.7	Dijagram stanja	35
4.8	Dijagram aktivnosti	36
4.9	Dijagram komponenti	37
5.6	Dijagram razmještaja	45
5.7	Ispravna instalacija	47
6.1	Dijagrami aktivnosti na master grani	59
6.2	Dijagrami aktivnosti na master grani	60

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 7. listopada 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Upoznavanje članova tima
 - Dogovor o tehnologijama. Dogovoreno: Postgres - Spring - React
 - Dogovor o prijedlogu drugog projektnog zadatka

2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Razgovor o funkcionalnim zahtjevima sustava i arhitekturi

3. sastanak

- Datum: 19. listopada 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Definiranje organizacije baze podataka
 - Dogovor o raspodjeli poslova

4. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Ivana Cepetić, Luka Martić
- Teme sastanka:
 - Razgovor o optimalnoj organizaciji baze podataka

5. sastanak

- Datum: 29. listopada 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Podjela uloga za kreiranje baze, izradu kontrolera, servisa i repozitorija, dokumentacije i logina i registracije

6. sastanak

- Datum: 4. studenoga 2020.
- Prisustvovali: Dora Horvat, Dora Bortas, Ivana Cepetić, Luka Martić
- Teme sastanka:
 - Razgovor o trenutnom napretku dokumentacije

7. sastanak

- Datum: 6. studenoga 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Diskusija o trenutnom napretku projekta
 - Podjela poslova koji se moraju obaviti do prve predaje projekta

8. sastanak

- Datum: 10. studenoga 2020.
- : Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Dogovor o načinu kriptiranja i hashiranju lozinki

9. sastanak

- Datum: 11. studenoga 2020.
- : Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Pregled trenutnog stanja projekta
 - Priprema za prvo kolokviranje

10. sastanak

- Datum: 12. studenoga 2020.

- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas, Ivana Cepetić, Luka Martić, Hrvoje Šimić
- Teme sastanka:
 - Pregled generičkih funkcionalnosti aplikacije prije prve predaje

11. sastanak

- Datum: 3. prosinca 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas, Ivana Cepetić, Hrvoje Šimić
- Teme sastanka:
 - Kolokviranje prvog ciklusa projekta

12. sastanak

- Datum: 10. prosinca 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Podjela poslova nakon kolokviranja

13. sastanak

- Datum: 29. prosinca 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Provjera napretka projekta
 - Ponovna podjela zadataka

14. sastanak

- Datum: 31. prosinca 2020.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Provjera napretka projekta

15. sastanak

- Datum: 2. siječnja 2021.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas

- Teme sastanka:
 - Dogovor o implementaciji lajkova

16. sastanak

- Datum: 3. siječnja 2021.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Dogovor o algoritmu uparivanja oglasa

17. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas
- Teme sastanka:
 - Provjera napretka
 - Dogovor o nastavku rada na aplikaciji

18. sastanak

- Datum: 7. siječnja 2021.
- Prisustvovali: Mateja Iveta, Dora Horvat, Vedran Hernaus, Lea Brzica, Matija Holik, Denis Đurašinović , Dora Bortas, Ivana Cepetić
- Teme sastanka:
 - Prezentacija alfa verzije projekta

Tablica aktivnosti

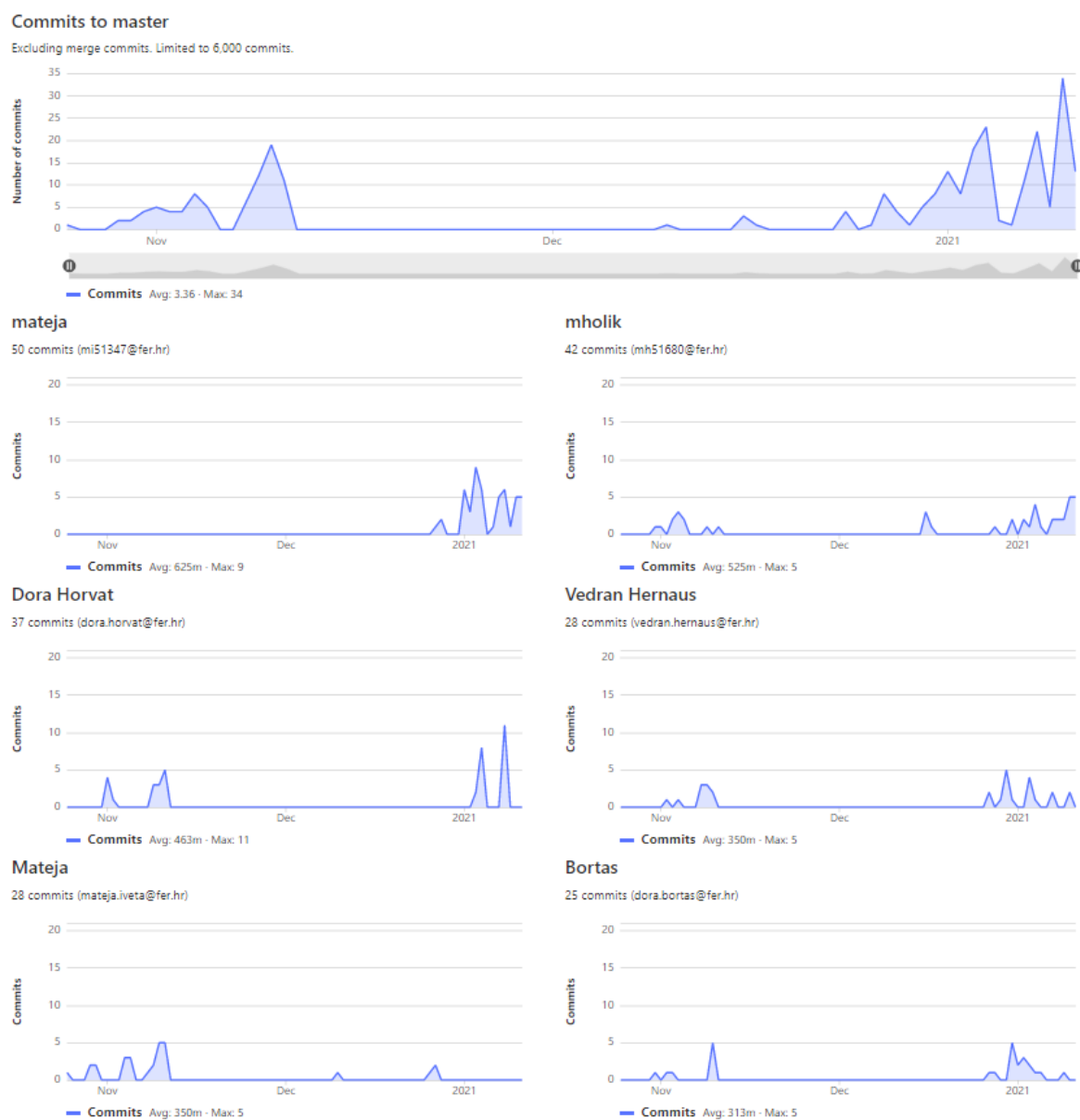
Kontinuirano osvježavanje

Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

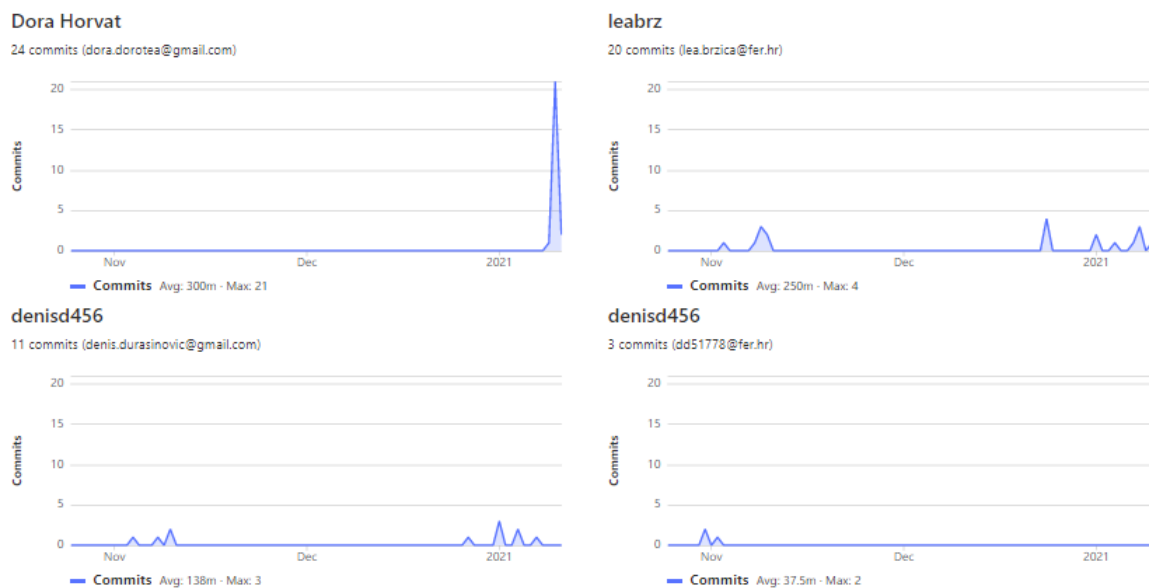
	Mateja Iveta	Dora Bortas	Dora Horvat	Lea Brzica	Vedran Hernaus	Matija Holik	Denis Đurašinović
Upravljanje projektom	5						
Opis projektnog zadatka			6				
Funkcionalni zahtjevi		3					
Opis pojedinih obrazaca		5					
Dijagram obrazaca		5					
Sekvencijski dijagrami		3					
Opis ostalih zahtjeva		1					
Arhitektura i dizajn sustava		1					
Baza podataka			15				
Dijagram razreda		5	5				
Dijagram stanja			8				
Dijagram aktivnosti			8				
Dijagram komponenti			8				
Korištene tehnologije i alati	5						
Ispitivanje programskog rješenja		10					
Dijagram razmještaja			5				
Upute za puštanje u pogon			6				
Dnevnik sastajanja			1				
Zaključak i budući rad			3				
Popis literature							
Izrada frontenda	50	20		50	50	2	

	Mateja Iveta	Dora Bortas	Dora Horvat	Lea Brzica	Vedran Hernaus	Matija Holik	Denis Đurašinović
Izrada backenda	50	10		10	20	50	30
Sigurnost	5					30	
Spajanje na bazu	5					5	5
Modeliranje baze	5	2				10	30
Puštanje u pogon	15						

Dijagrami pregleda promjena



Slika 6.1: Dijagrami aktivnosti na master grani



Slika 6.2: Dijagrami aktivnosti na master grani