

Programsko inženjerstvo

Ak. god. 2023./2024.

BytePit

Dokumentacija, Rev. 1

Grupa: *Rade*

Voditelj: *Marko Bolt*

Datum predaje: *<dan>. <mjesec>. <godina>.*

Nastavnik: *Hrvoje Nuić*

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	10
3.1.2 Sekvencijski dijagrami	21
3.2 Ostali zahtjevi	24
4 Arhitektura i dizajn sustava	25
4.1 Baza podataka	27
4.1.1 Opis tablica	27
4.1.2 Dijagram baze podataka	34
4.2 Dijagram razreda	35
4.3 Dijagram stanja	38
4.4 Dijagram aktivnosti	39
4.5 Dijagram komponenti	40
5 Implementacija i korisničko sučelje	41
5.1 Korištene tehnologije i alati	41
5.2 Ispitivanje programskog rješenja	42
5.2.1 Ispitivanje komponenti	42
5.2.2 Ispitivanje sustava	42
5.3 Dijagram razmještaja	43
5.4 Upute za puštanje u pogon	44
6 Zaključak i budući rad	45
Popis literature	46
Indeks slika i dijagrama	47

Dodatak: Prikaz aktivnosti grupe

48

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Marko Bolt	04.11.2023.
0.2	Napravljen opis projekta	Marko Bolt	05.11.2023.
0.3	Napravljeni funkcionalni zahtjevi	Jakov Vinožganić Teo Rado- lović	09.11.2023.
0.4	Napravljeni obrasci uporabe	Filip Bernt	10.11.2023.
0.5	Napravljeni sekvencijski dijagrami	Filip Bernt Fran Sipić	10.11.2023.
0.6	Napravljeni ostali zahtjevi	Jure Franj- ković	12.11.2023.
0.7	Napravljena arhitektura i dizajn sustava	Jure Franj- ković	13.11.2023.
0.8	Napravljen opis baze podataka	Filip Bernt Fran Sipić	14.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.9	Napravljen dijagram razreda	Jure Franj- ković	15.11.2023.
1.0	Napravljena finalna inačica	Filip Bernt	17.11.2023.

Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

2. Opis projektnog zadatka

dio 1. revizije

Aplikacija ByteBit inovativno je rješenje namijenjeno provjeri programerskih vještina, omogućavajući korisnicima sudjelovanje u programskim natjecanjima i vježbanje rješavanja zadataka. Aplikacija se može implementirati kao web ili desktop platforma koristeći objektno-orijentirane programerske jezike, sa podrškom za kompilaciju i evaluaciju rješenja u najmanje jednom programskom jeziku.

Glavne funkcionalnosti aplikacije uključuju:

- **Pregledavanje zadataka i kalendar natjecanja:** Korisnici mogu pregledavati dostupne zadatke i kalendar natjecanja, što omogućava planiranje sudjelovanja i pripremu.
- **Profili korisnika:** Natjecatelji imaju profile s prikazom statistika njihovih rezultata, uključujući broj točno riješenih i isprobanih zadataka, kao i pehare za osvojena natjecanja. Profili voditelja sadrže informacije o učitanim zadacima i organiziranim natjecanjima.
- **Registracija korisnika:** Registracija zahtijeva osnovne podatke kao što su korisničko ime, fotografija, lozinka, ime, prezime i email adresa. Administrator potvrđuje registraciju, a u slučaju voditelja natjecanja potrebna je i dodatna verifikacija.
- **Natjecanja:** Početkom natjecanja, zadaci postaju vidljivi aktivnim natjecateljima. Nakon završetka, zadaci su dostupni svima. Postoji mogućnost slanja datoteka s programskim kodom i objavljivanje rang lista po bodovima, uzimajući u obzir vrijeme i točnost rješenja.
- **Prikaz rješenja:** Natjecatelji mogu vidjeti rješenja drugih natjecatelja nakon natjecanja, dok su tijekom natjecanja vidljiva samo ako su sami točno riješili zadatak.
- **Vježbanje:** Natjecatelji mogu vježbati zadatke i učitavati rješenja, s automatskom provjerom točnosti i vremenskih ograničenja.
- **Virtualna natjecanja:** Natjecatelji mogu stvoriti vlastita virtualna natjecanja

odabirom prošlih natjecanja ili nasumičnim odabirom zadataka, a rezultati se uspoređuju s originalnim.

- **Uloga voditelja:** Voditelji imaju mogućnost učitavanja novih zadataka, organiziranja natjecanja, i odabira zadataka koji će biti aktivni tijekom natjecanja, uz mogućnost uređivanja vlastitih zadataka.
- **Administrativne funkcije:** Administratori imaju ovlasti za upravljanje korisničkim računima i uređivanjem svih zadataka i natjecanja, bez mijenjanja prethodno ostvarenih rezultata.

Trenutno postoje brojne platforme poput Codeforces, LeetCode, HackerRank i drugih koje nude slične funkcionalnosti. Razlike u odnosu na BytePit bi se mogle ogledati u specifičnostima kao što su lokalizacija (jezik i regionalni fokus), specijalizacija za određene programerske jezike ili tehnologije, ili u razini personalizacije korisničkog iskustva. Slična rješenja i razlike:

- **Codeforces:** Fokusira se na konkurentno programiranje s redovitim natjecanjima. BytePit bi mogao uključivati više edukativnih resursa i prilagodljiva virtualna natjecanja.
- **LeetCode:** Specijaliziran za pripremu tehničkih intervjua. BytePit može nuditi širi raspon zadataka, uključujući one van konteksta intervjua.
- **HackerRank:** Osim izazova u programiranju, nudi i podršku za tvrtke u regrutiranju. BytePit bi mogao dodati vrednost kroz specifične profile natjecanja i detaljnu statistiku. Sličan koncept, ali s fokusom na poslovne korisnike. BytePit bi mogao biti više usmjeren na edukaciju i natjecanja.

BytePit bi mogao privući širok spektar korisnika, od početnika u programiranju koji traže mjesto za učenje i prakticiranje, do iskusnih programera koji žele oštriti svoje vještine ili sudjelovati u natjecanjima, te edukativnih institucija koje traže platformu za organiziranje natjecanja ili kao alat u nastavi. Ova raznolikost korisnika stvara dinamično okruženje u kojem se znanje i iskustvo neprestano razmjenjuju, pružajući beskrajne mogućnosti za učenje i napredak. S jedne strane, početnici imaju priliku učiti kroz praksu, rješavajući stvarne probleme i dobivajući povratne informacije od iskusnijih kolega. S druge strane, iskusni programeri mogu se suočavati s izazovnijim problemima i sudjelovati u zahtjevnijim natjecanjima, što ih stavlja u poziciju mentora, potičući ih na daljnji razvoj i dijeljenje znanja. Edukativne institucije mogu koristiti BytePit za stvaranje prilagođenih

tečajeva i kurikuluma, pružajući studentima praktično iskustvo koje je ključno za njihov razvoj. Poslodavci koji traže talentirane programere mogli bi koristiti platformu za identifikaciju potencijalnih kandidata, provodeći natjecanja ili izazove koji služe kao dio procesa zapošljavanja ili kroz analizu rješenja i pristupa problemima koje programeri koriste na platformi.

BytePit bi trebao biti projektiran s mogućnostima prilagodbe, što bi uključivalo konfigurabilne module za natjecanja, personalizirane profile korisnika, i fleksibilan sustav ocjenjivanja. Prilagodba omogućava korisnicima da kreiraju jedinstveno iskustvo koje odgovara njihovim specifičnim potrebama i stilu učenja ili rada. Personalizirani profili omogućavaju korisnicima da predstave svoje vještine i postignuća, kao i da prate svoj napredak kroz različite metrike i statistike. Fleksibilan sustav ocjenjivanja omogućava različite forme evaluacije, od standardnog bodovanja do složenih algoritama koji mogu vrednovati efikasnost koda ili kreativno rješavanje problema. Ovo omogućava platformi da raste i evoluiraju zajedno sa svojim korisničkim bazama i tehnološkim trendovima, ostajući relevantna i privlačna za široki spektar korisnika, od individualaca i obrazovnih institucija do korporacija i tehničkih zajednica.

Opseg projekta uključuje dizajniranje i implementaciju korisničkih sučelja, koje moraju biti intuitivna i pristupačna, osiguravajući ugodno iskustvo za sve korisnike, neovisno o njihovom tehničkom znanju ili iskustvu u programiranju. Backend servisi za upravljanje bazom podataka moraju biti robustni i optimizirani za brzu obradu upita, što je ključno za održavanje visoke performanse platforme čak i pod velikim opterećenjem. Sustav za evaluaciju koda mora biti precizan i pouzdan, sposoban za brzo i efikasno procjenjivanje raznovrsnih programskih rješenja koja korisnici predaju. Sustav za autentifikaciju i autorizaciju korisnika treba osigurati da su svi podaci sigurni i da pristup platformi

Nadogradnje projekta mogu uključivati:

- Integraciju s vanjskim servisima poput GitHub-a za uvoz postojećeg koda.
- Implementaciju naprednijih algoritama za detekciju plagijata.
- Proširenje podrške za više programskih jezika i tehnoloških okvira.
- Razvoj mobilne aplikacije za pristup platformi.
- Uvođenje umjetne inteligencije za personalizirane preporuke zadataka na te-

melju korisnikovih prethodnih aktivnosti i uspjeha.

- "Gamification" sustav za nagrađivanje korisnika za postignuća i napredak.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

Navesti **dionike** koji imaju **interes u ovom sustavu** ili **su nositelji odgovornosti**. To su prije svega korisnici, ali i administratori sustava, naručitelji, razvojni tim.

Navesti **aktore** koji izravno **koriste** ili **komuniciraju sa sustavom**. Oni mogu imati inicijatorsku ulogu, tj. započinju određene procese u sustavu ili samo sudioničku ulogu, tj. obavljaju određeni posao. Za svakog aktora navesti funkcionalne zahtjeve koji se na njega odnose.

Dionici:

1. Dionik 1
2. Dionik 2
3. ...

Aktori i njihovi funkcionalni zahtjevi:

1. Aktor 1 (inicijator) može:
 - (a) funkcionalnost 1
 - (b) funkcionalnost 2
 - i. podfunkcionalnost 1
 - ii. podfunkcionalnost 2
 - (c) funkcionalnost 3
2. Aktor 2 (sudionik) može:
 - (a) funkcionalnost 1
 - (b) funkcionalnost 2

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregled početne stranice

- **Glavni sudionik:** Natjecatelj, voditelj, neregistrirani korisnik
- **Cilj:** Odlučiti se za opciju registracije ili prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Prikazuje se početna stranica
 2. Odabir prijave ili registracije
 3. Prosljeđivanje na odabranu opciju

UC2 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Odlučiti se za opciju registracije kao voditelj ili natjecatelj
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Prikazuje se stranica za registraciju
 2. Neregistrirani korisnik ispunjava zadane podatke
 3. Odabire opciju registracije za voditelja ili natjecatelja
 4. Potvrđivanje registracije putem email-a
 5. Dodatna potvrda administratora za novog voditelja
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav odbija odobravanje registracije
 2. Korisnik mijenja podatke u ispravan unos ili odustaje od registracije

UC3 - Prijava u sustav

- **Glavni sudionik:** Natjecatelj, voditelj
- **Cilj:** Prijava u sustav i pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Prikazuje se stranica za prijavu
 2. Unos emaila i lozinke
 3. Pristup korisničkoj stranici
- **Opis mogućih odstupanja:**
 - 2.a Neispravan unos emaila ili lozinke
 1. Sustav odbija pristup te nudi ponovni pokušaj za prijavu

UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Natjecatelj, voditelj
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Pritisak korisnika na svoj profil
 2. Prikaz osobnih podataka

UC5 - Pregled zadataka

- **Glavni sudionik:** Natjecatelj, voditelj
- **Cilj:** Pregledati zadatke na stranici
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik pritiskom na dostupne zadatke dobiva pristup
 2. Korisnik može pregledavati već objavljene zadatke

UC6 - Pregled kalendara

- **Glavni sudionik:** Natjecatelj, voditelj
- **Cilj:** Pregledati kalendar
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik na svojoj stranici ima pristup kalendaru
 2. Korisnik unutar kalendara može pregledati datume s dostupnim natjecanjima

UC7 - Pregled profila drugih korisnika

- **Glavni sudionik:** Natjecatelj, voditelj
- **Cilj:** Pregledati profile drugih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik na svojoj stranici odabire profil koji želi vidjeti
 2. Ako je pritisnuti profil natjecatelj prikazuje se statistika o zadacima i osvojeni pehari
 3. Ako je pritisnuti profil voditelj prikazuje se popis učitanih zadataka i kalendar s popisom natjecanja

UC8 - Sudjelovanje u natjecanju

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Prikazati natjecatelju zadatke za odabrano natjecanje
- **Sudionici:** Baza podataka, natjecatelj
- **Preduvjet:** Korisnik je prijavljen u sustav i postoji natjecanje u kojem može sudjelovati
- **Opis osnovnog tijeka:**
 1. Natjecatelj sudjeluje u natjecanju
 2. Završetkom natjecanja zadaci postaju dostupni svim korisnicima za vježbanje
 3. Objava rang liste i dodjeljivanje pehara za prva tri mjesta
- **Opis mogućih odstupanja:**
 - 2.a Pokušaj sudjelovanja u natjecanju korisnika koji nije natjecatelj
 1. Sustav odbija sudjelovanje voditelja u natjecanju

UC9 - Kreiranje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Voditelj stvara natjecanje
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Korisnik je prijavljen u sustav i odobren kao voditelj
- **Opis osnovnog tijeka:**
 1. Voditelj ispunjava formu za izradu natjecanja
 2. Voditelj odabire željene zadatke za natjecanje
 3. Natjecanje se kreira i sprema u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Pokušaj kreiranja natjecanja korisnika koji nije voditelj
 1. Sustav odbija kreiranje natjecanja i šalje poruku da korisnik nema ovlasti za taj postupak

UC10 - Kreiranje zadatka

- **Glavni sudionik:** Voditelj
- **Cilj:** Voditelj stvara zadatak
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Korisnik je prijavljen u sustav i odobren kao voditelj
- **Opis osnovnog tijeka:**
 1. Voditelj ispunjava formu za izradu zadatka
 2. Voditelj unosi naziv, opis zadatka, testne primjere, broj bodova koji nosi zadatak te ih prosljeđuje sustavu
 3. Voditelj opciju privatnog zadatka, on će postati javan tek nakon završetka natjecanja
 4. Sustav pohranjuje zadatak u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Pokušaj kreiranja zadatka korisnika koji nije voditelj
 1. Sustav odbija kreiranje zadatka i šalje poruku da korisnik nema ovlasti za taj postupak

UC11 - Predaja rješenja zadatka

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecatelj predaje svoje rješenje zadatka
- **Sudionici:** Baza podataka, natjecatelj
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Natjecatelj šalje zahtjev za evaulaciju rješenja
 2. Evaulator prosljeđuje compileru zahtjev
 3. Compiler pokreće testne primjere nad programom
 4. Evaulator računa broj bodova nakon dobivenih rezultata od compilera
 5. Sustav ažurira broj bodova za natjecatelja i pohranjuje ih u bazu
- **Opis mogućih odstupanja:**
 - 2.a Natjecatelj ne predaje rješenje zadatka
 1. Natjecatelj ne predaje rješenje zadatka te se ne pokreće proces ocjenjivanja već dodjeljuje 0

UC12 - Pregled učitanih rješenja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecatelj pregledava rješenja drugih natjecatelja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i natjecanje u kojem je sudjelovao je završilo
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire zadatak unutar natjecanja
 2. Prikazuje mu se popis svih natjecatelja koji su učitali neko rješenje
 3. Rješenja su prikazana sortirano
 4. Ako je natjecatelj potpuno točno riješio zadatak pojavljuje se opcija za preuzimanje rješenja

UC13 - Vježbanje zadataka

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecatelj ima mogućnost vježbanja već objavljenih zadataka
- **Sudionici:** Baza podataka, natjecatelj
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire popis objavljenih zadataka
 2. Natjecatelj pritiskom otvara zadatak koji želi vježbati
 3. Nakon rješavanja zadatka učitava programski kod u aplikaciju
 4. Natjecatelju se dodjeljuju bodovi proporcionalno broju točno riješenih primjera
- **Opis mogućih odstupanja:**
 - 2.a Natjecatelj ne predaje rješenje zadatka
 1. Natjecatelj ne predaje rješenje zadatka te se ne pokreće proces ocjenjivanja već dodjeljuje 0

UC14 - Izrada virtualnog natjecanja od prošlih natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecatelj ima mogućnost vježbanja na već objavljenim natjecanjima
- **Sudionici:** Baza podataka, natjecatelj
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Natjecatelj otvara popis prošlih natjecanja
 2. Natjecatelj odabire neko natjecanje
 3. Natjecanje je aktivno samo za njega
 4. Nakon završetka natjecatelja se rangira po službenim rješenjima

UC15 - Izrada virtualnog natjecanja od nasumičnih zadataka

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecatelj ima mogućnost vježbanja natjecanja uz pomoć objavljenih zadataka simuliranih u natjecanje
- **Sudionici:** Baza podataka, natjecatelj
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire opciju natjecanja napravljenog od nasumičnih objavljenih zadataka
 2. Zadatci se spremaju u natjecanje ravnomjerno po količini bodova
 3. Natjecanje je aktivno samo za njega
 4. Završetkom natjecanja ima pristup rješenjima zadataka

UC16 - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator ima mogućnost pregleda svih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregleda korisnika
 2. Prikazuje se lista ispravno registriranih korisnika

UC17 - Uređivanje vlastitih natjecanja i zadataka

- **Glavni sudionik:** Voditelj
- **Cilj:** Mogućnost izmjene zadataka i natjecanja koje je napravio voditelj
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Korisnik je prijavljen u sustav i odobren je kao voditelj
- **Opis osnovnog tijeka:**
 1. Voditelj odabire natjecanje ili zadatak
 2. Ako je autor pruža mu se opcija za uređivanje
 3. Nakon uređivanja sprema promjene, koje ne utječu na prethodne rezultate

UC18 - Uređivanje svih natjecanja i zadataka

- **Glavni sudionik:** Administrator
- **Cilj:** Mogućnost izmjene svih zadataka i natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire natjecanje ili zadatak
 2. Administrator mijenja zadatke ili natjecanja
- **Opis mogućih odstupanja:**
 - 2.a Administrator nakon izmjene ne pohrani promjene
 1. Sustav obavještava administratora da nije pohranio promjene

UC19 - Promjena prava pristupa

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti razinu pristupa korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator pronalazi željenog korisnika
 2. Administrator mijenja razinu prava korisnika
- **Opis mogućih odstupanja:**
 - 2.a Administrator nakon izmjene ne pohrani promjene
 1. Sustav obavještava administratora da nije pohranio promjene

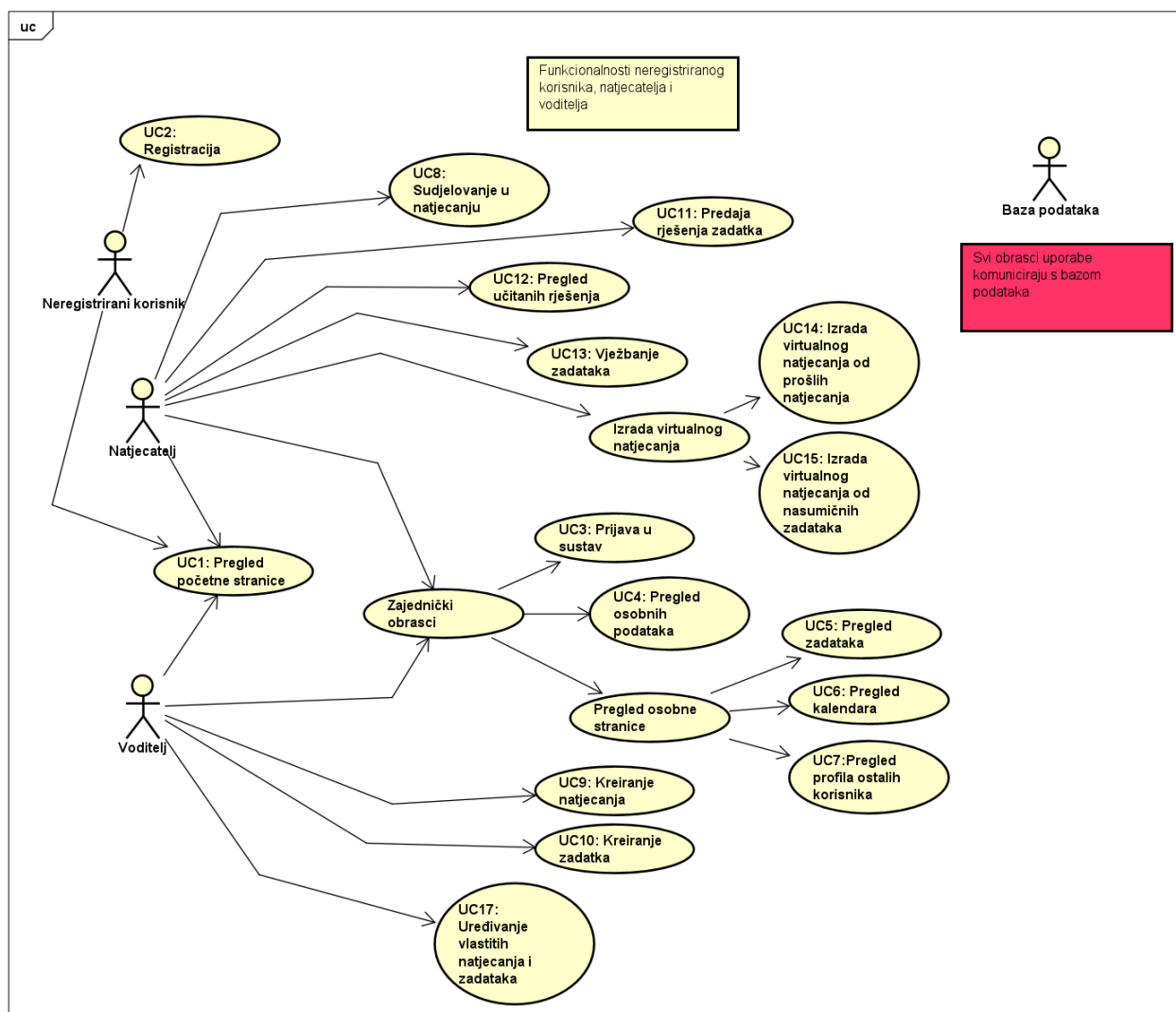
UC20 - Promjena osobnih podataka

- **Glavni sudionik:** Administrator
- **Cilj:** Mogućnost izmjene osobnih podataka korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator pronalazi željenog korisnika
 2. Administrator mijenja osobne podatke korisnika
 3. Administrator sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Administrator nakon izmjene ne pohrani promjene
 1. Sustav obavještava administratora da nije pohranio promjene

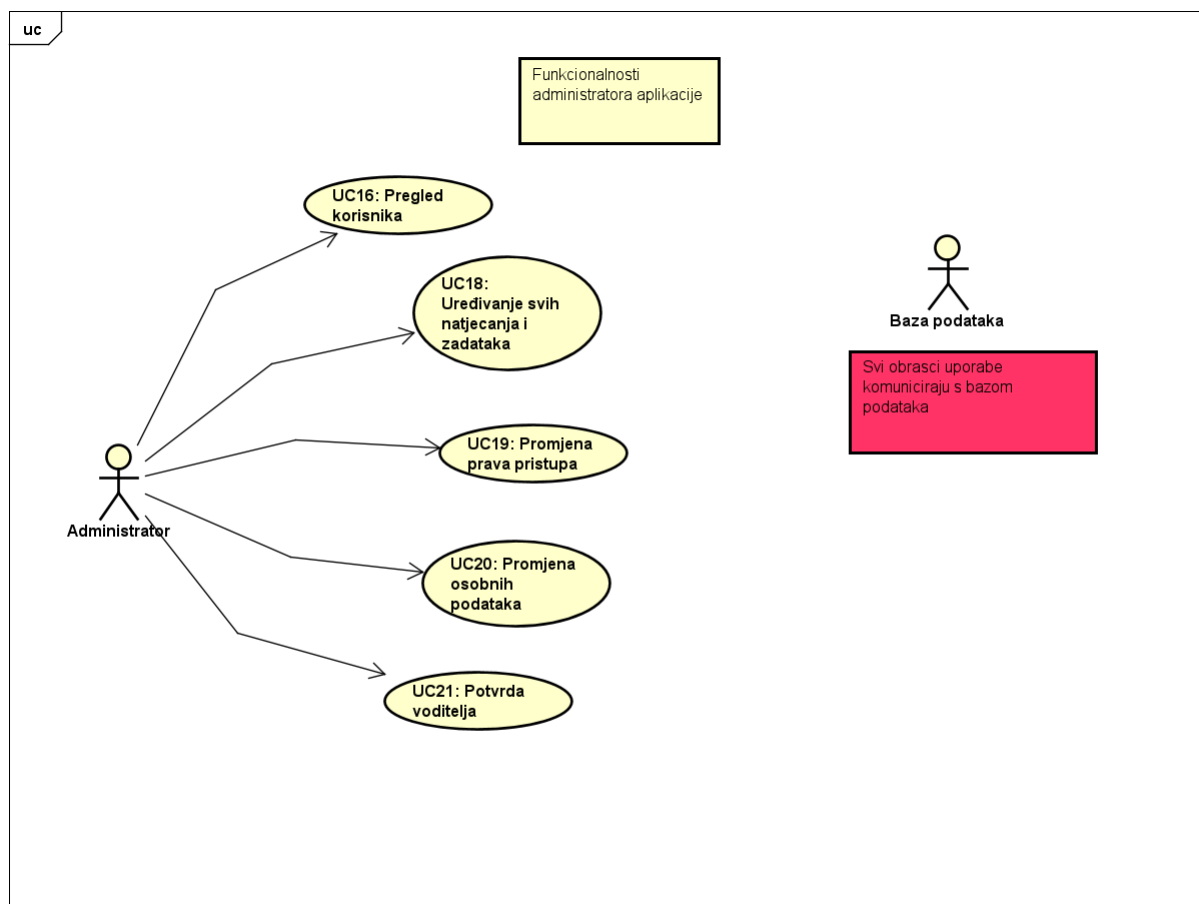
UC21 - Potvrda voditelja

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator potvrđuje voditelja
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Korisnik je prijavljen u sustav i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator pronalazi željenog korisnika
 2. Administrator mu odobrava pravo voditelja
 3. Baza podataka se ažurira

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost natjecatelja, voditelja i neregistriranog korisnika

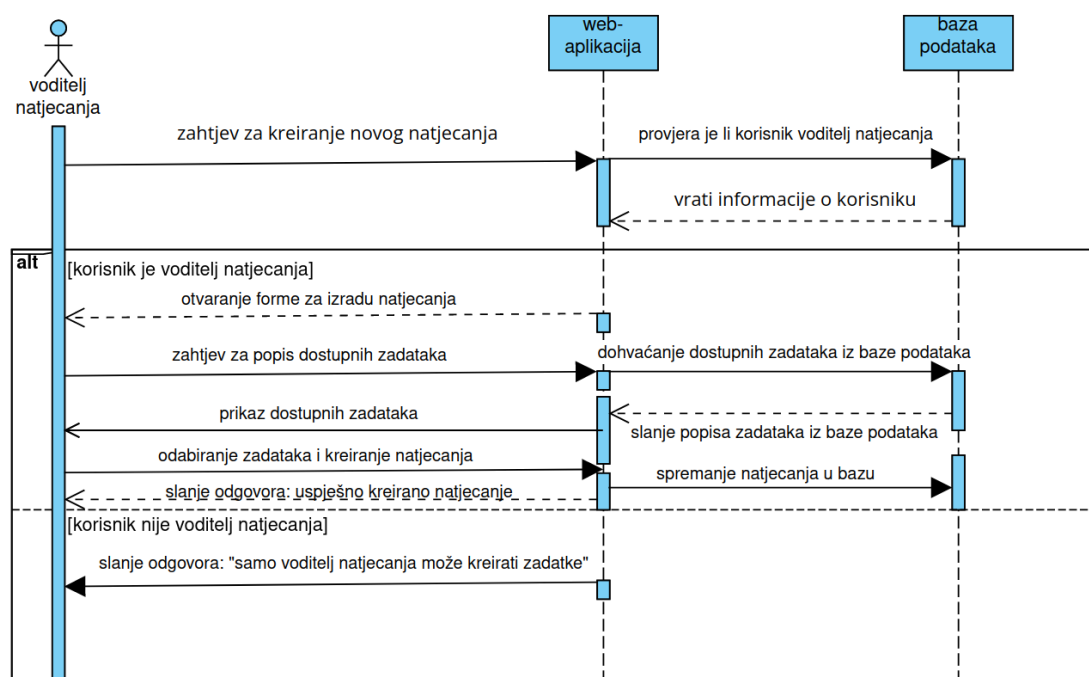


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

UC9 Kreiranje natjecanja

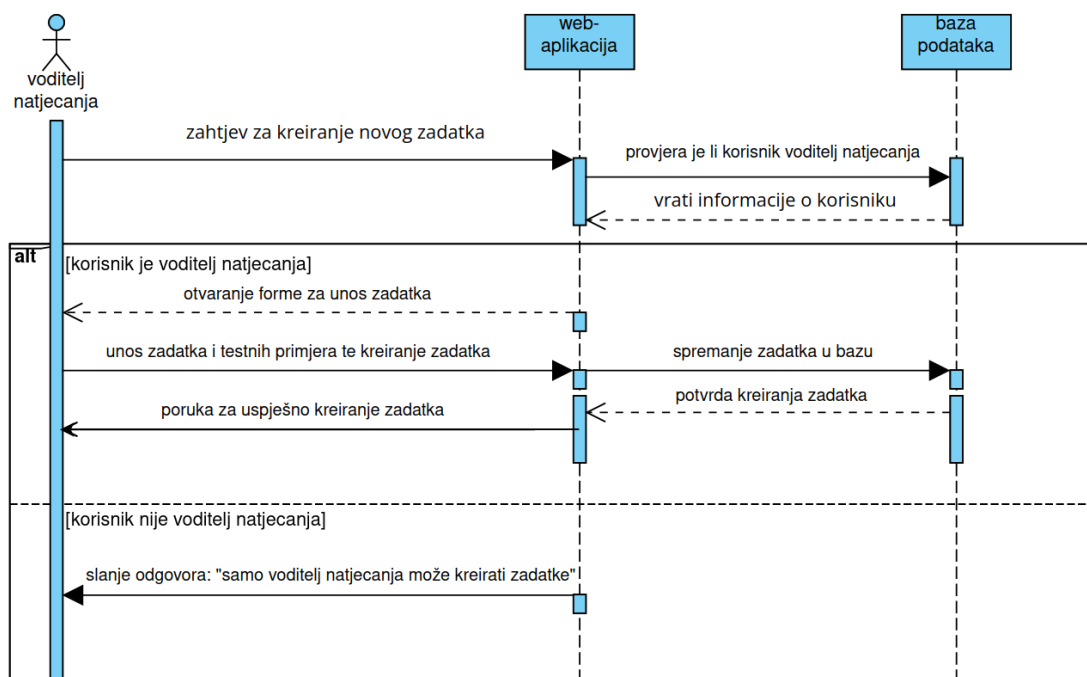
Voditelj natjecanja započinje proces kreiranja natjecanja slanjem zahtjeva poslužitelju. Poslužitelj provjerava informacije o korisniku u bazi podataka kako bi utvrdila je li korisnik ovlašten za kreiranje natjecanja. Ako je korisnik voditelj natjecanja na web-aplikaciji otvara se forma za izradu natjecanja, voditelj šalje zahtjev za popis dostupnih zadataka, poslužitelj ih dohvaća iz baze podataka i prikazuje ih voditelju. Zatim voditelj odabire željene zadatke i kreira natjecanje. Poslužitelj sprema informacije o natjecanju u bazu podataka i šalje povratnu informaciju voditelju o uspješnom kreiranju natjecanja. Ako korisnik nije voditelj natjecanja poslužitelj šalje odgovor voditelju da samo voditelji natjecanja mogu kreirati zadatke.



Slika 3.3: Sekvencijski dijagram kreiranja novog natjecanja

UC10 Kreiranje zadatka

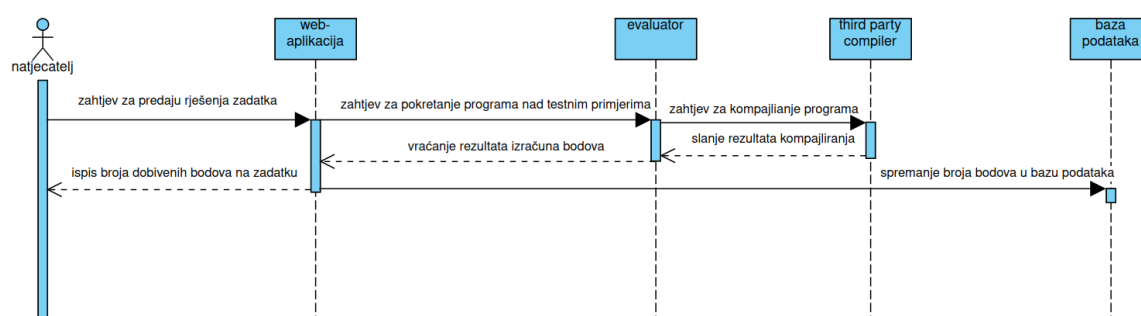
Voditelj natjecanja počinje proces kreiranja novog zadatka slanjem zahtjeva poslužitelju koji provjerava je li korisnik voditelj natjecanja. Ako je korisnik voditelj natjecanja poslužitelj omogućuje pristup formi za unos zadatka. Voditelj unosi detalje zadatka i testne primjere te šalje podatke. Poslužitelj prosljeđuje spremanje zadatka u bazu podataka te šalje potvrdu voditelju o uspješnom kreiranju zadatka. Ako korisnik nije voditelj natjecanja poslužitelj šalje poruku korisniku da samo voditelj natjecanja može kreirati zadatke.



Slika 3.4: Sekvencijski dijagram kreiranja novog zadatka

UC11 Predaja rješenja

Natjecatelj započinje proces predaje rješenja zadatka slanjem zahtjeva poslužitelju koji proslijedi zahtjev evaluatoru za pokretanje programa nad testnim primjerima. Evaluator proslijedi zahtjev third party compileru. Compiler pokreće program nad testnim primjerima i šalje rezultate evaluatoru. Evaluator izračunava broj dobivenih bodova i šalje ih Poslužitelju koji ažurira natjecatelja s brojem dobivenih bodova te šalje rezultate i broj bodova bazi podataka za spremanje.



Slika 3.5: Sekvencijski dijagram predaje rješenja zadatka

3.2 Ostali zahtjevi

- Cijeli sustav namijenjen je upotrebi na engleskom jeziku
- Sustav mora biti u mogućnosti podržati veći broj korisnika zbog svoje namjene
- Upotreba gotove aplikacije zamišljena je preko web preglednika
- Aplikacija mora implementirati autentikaciju i autorizaciju svakog korisnika
- Osjetljivi podaci od korisnika se moraju pretvarati u hash prije spremanja u bazu
- Greška u jednom djelu sustava ne smije onemogućiti korištenje drugih usluga
- Odgovori sustava prema korisniku moraju biti brzi i ne dulji od nekoliko sekundi

4. Arhitektura i dizajn sustava

Arhitektura cijelog sustava može se podijeliti na četiri ključna dijela:

- Web preglednik
- Web aplikacija
- Baza podataka
- Code runner

Web preglednik omogućuje interakciju između korisnika i aplikacije. Svaki se korisnikov zahtjev događa na web pregledniku i prosljeđuje aplikaciji na obradu. Osim slanja zahtjeva, korisniku se omogućuje bolji pregled aplikacije pomoću apstraktnog koda kojeg web preglednik pretvara u lako shvatljive strukture.

Aplikacija se sastoji od dva glavna dijela, frontenda i backenda. Tehnologije koje se koriste na backendu uključuju programski jezik **Python** i **fastApi** web framework koji omogućuje stvaranje aplikacije temeljene na **REST** arhitekturi i rutera na kojem su definirane rute za slanje zahtjeva i komunikaciju korisnika i aplikacije. Frontend aplikacije izrađen je korištenjem JavaScript librarya **React**. On omogućuje korisniku jednostavan pregled aplikacije i jednostavno korištenje aplikacije, odnosno slanje zahtjeva.

Kontejneri se koriste unutar aplikacije kako bi poboljšali korisničko iskustvo, optimizirali performanse sustava te osigurali skalabilnost za povećani broj korisnika. Glavni se dio aplikacije s ruterima i logikom za komunikaciju s bazom nalazi u vlastitom kontejneru kao i glavna baza, evaluator i blob storage.

Za bazu podataka se koristi **SQL** specifično **PostgreSQL**. Glavna baza se nalazi u vlastitom kontejneru kojemu backend aplikacije pristupa kako bi spremio informacije u nju. Na backendu se koristi **psycopg** PostgreSQL adapter koji omogućuje komunikaciju između baze i backenda.

Evaluator komunicira s vanjskim *Code runnerom* za kompajliranje kodova koje korisnik šalje na backend. Ovaj ključni dio sustava odgovoran je za izvršavanje kôda unutar sigurnog okruženja, čime se osigurava izolacija korisničkih skripti od ostatka aplikacije. Kroz ovu komunikaciju, *Evaluator* prima kôd od korisnika, proslijeđuje ga *Code runneru* na izvršavanje te zatim interpretira rezultate kako bi ih pravilno integrirao natrag u aplikaciju.

Dodatno, *evaluator* koristi **blob storage** pomoću kojeg dohvaća podatke koji su mu potrebni za evaluaciju određenog zadatka. *Evaluator* ima pristup informacijama poput ulaznih podataka za testiranje, očekivanih rezultata te drugih specifičnih parametara vezanih uz pojedini zadatak.

4.1 Baza podataka

Za implementaciju našeg sustava koristit ćemo PostgreSQL, relacijsku bazu podataka. Ovaj tip baze podataka posebno je pogodan za modeliranje kompleksnih struktura koje odražavaju realni svijet, zahvaljujući svojoj sposobnosti da efikasno upravlja relacijama, odnosno tablicama. Svaka tablica u bazi unikatno je definirana svojim nazivom i skupom atributa, čime se omogućava precizno strukturiranje podataka. Jedna od ključnih prednosti PostgreSQL baze je njena brzina i efikasnost u obradi podataka, uključujući pohranu, ažuriranje i dohvaćanje informacija. Naša baza podataka obuhvatit će niz entiteta koji su specifično prilagođeni zahtjevima naše aplikacije.

Popis entiteta: users, competitions, problems, competition_participations, problems_on_competitions, problem_results, trophies i verification_tokens.

4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ

Users

Entitet **users** sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: *id*, *role*, *username*, *image_url*, *password_hash*, *email*, *name*, *surname*, *is_verified*, *created_on*. Ovaj entitet je u *many-to-many* vezi s entitetom *competitions* preko tablice *competition_participations*. Ovaj entitet je u *one-to-many* vezi s entitetom *verification_tokens* preko atributa *email*.

users		
id	UUID	jedinstveni brojčani identifikator
role	ENUM	uloga korisnika (admin, voditelj natjecanja ili natjecatelj)
username	VARCHAR(32)	korisničko ime
image	BYTEA	slika
password_hash	VARCHAR(255)	hash šifre korisnika
email	VARCHAR(128)	adresa e-pošte korisnika
name	VARCHAR(64)	ime korisnika
surname	VARCHAR(64)	prezime korisnika
is_verified	BOOLEAN	bool vrijednost koja govori je li korisnik verificirao adresu e-pošte
created_on	TIMESTAMP	timestamp trenutka kreiranja korisnika

Slika 4.1: Entitet **users**

Problems

Entitet **problems** sadržava sve važne informacije o zadacima. Sadržava attribute: *id*, *name*, *description*, *example_input*, *example_output*, *is_hidden*, *num_of_points*, *runtime_limit*, *tests_dir*, *is_private*, *created_on*. Ovaj entitet je u *many-to-many* vezi s entitetom *competitions* preko tablice *problems_on_competitions*. Ovaj entitet je u *one-to-many* vezi s entitetom *problem_results* preko atributa *id*.

problems		
id	UUID	jedinstveni brožčani identifikator
name	TEXT	naziv zadatka
example_input	TEXT	primjer unosa
example_output	TEXT	primjer izlaza
is_hidden	BOOLEAN	bool vrijednost koja govori je li skriven zadatak ili ne
num_of_points	REAL	broj bodova na zadatku
runtime_limit	TIME	ograničenje vremena izvođenja rješenja zadatka
description	TEXT	opis zadatka
tests_dir	TEXT	direktorij s testovima za ispravljanje rješenja
is_private	BOOLEAN	bool vrijednos koja govori je li privatan zadatak
created_on	TIMESTAMP	timestamp trenutka kreiranja zadatka

Slika 4.2: Entitet **problems**

Competitions

Entitet **competitions** sadržava sve važne informacije o natjecanjima. Sadržava attribute: *id*, *name*, *description*, *start_time*, *end_time*, *parent_id*, *problems*. Natjecanja mogu imati svoja nadređena natjecanja. Ovaj entitet je u *many-to-many* vezi s entitetom *users* preko *competition_participations* tablice. Ovaj entitet je u *many-to-many* vezi s entitetom *problems* preko *problems_on_competitions* tablice.

competitions		
id	UUID	jedinstveni brojčani identifikator
name	TEXT	naziv natjecanja
description	TEXT	opis natjecanja
start_time	TIMESTAMP	vrijeme početka natjecanja
end_time	TIMESTAMP	vrijeme završetka natjecanja
parent_id	UUID	identifikator natjecanja-roditelja
problems	UUID[]	svi zadaci u natjecanju

Slika 4.3: Entitet **competitions**

Competition_participations

Entitet **competition_participations** sadržava sve važne informacije o sudjelovanju korisnika na natjecanjima. Sadržava attribute: *id*, *user_id*, *competition_id*, *num_of_points*. Ovaj entitet je u *many-to-many* vezi s entitetom *users*. Ovaj entitet je u *many-to-many* vezi s entitetom *competitions* preko atributa *competition_id*.

competition_participations		
id	UUID	jedinstveni brojčani identifikator
user_id	UUID	jedinstveni brojčani identifikator korisnika, users.id
competition_id	UUID	jedinstveni brojčani identifikator natjecanja, competitions.id
num_of_points	REAL	broj bodova korisnika na natjecanju

Slika 4.4: Entitet **competition_participations**

Problems_on_competitions

Entitet **problems_on_competitions** sadržava informacije o zadacima koji pripadaju pojedinim natjecanjima. Sadržava attribute: *id*, *problem_id*, *competition_id*, *num_of_points*.

problems_on_competitions		
id	UUID	jedinstveni brojčani identifikator
problem_id	UUID	jedinstveni brojčani identifikator zadatka, problems.id
competition_id	UUID	jedinstveni brojčani identifikator natjecanja, competitions.id
num_of_points	REAL	broj bodova na zadatku u natjecanju

Slika 4.5: Entitet **problems_on_competitions**

Problem_results

Entitet **problem_results** sadrži rezultate pojedinih korisnika na pojedinim zadacima. Sadržava attribute: *id*, *user_id*, *problem_id*, *competition_id*, *num_of_points*, *average_runtime*, *is_correct*, *source_code*. Ovaj entitet je u *one-to-many* vezi s entitetom *users* preko *user_id* te *one-to-many* vezi s entitetom *problems* preko atributa *problem_id*. Ovaj entitet je u *one-to-many* vezi s entitetom *competitions* preko atributa *competition_id*.

problem_results		
id	UUID	jedinstveni brožani identifikator
problem_id	UUID	jedinstveni brožani identifikator zadatka, problems.id
competition_id	UUID	jedinstveni brožani identifikator natjecanja, competitions.id
user_id	UUID	jedinstveni brožani identifikator korisnika, users.id
average_runtime	TIME	prosječno vrijeme izvođenja programa
is_correct	BOOLEAN	bool vrijednost točnosti zadatka
num_of_points	REAL	korisnikov broj bodova na zadatku
source_code	TEXT	izvorni kod korisnika koji je predan kao rješenje

Slika 4.6: Entitet **problem_results**

Trophies

Entitet **trophies** sadržava informacije o trofejima koje korisnici mogu osvojiti. Sadržava attribute: *id*, *competition_id*, *user_id*, *position*, *icon*. Ovaj entitet je u *one-to-many* vezi s entitetom *users* preko atributa *user_id* te *one-to-many* vezi s entitetom *competitions* preko atributa *competition_id*.

trophies		
id	UUID	jedinstveni brojčani identifikator
competition_id	UUID	jedinstveni brojčani identifikator natjecanja, competitions.id
user_id	UUID	jedinstveni brojčani identifikator korisnika, users.id
position	SMALLINT	pozicija korisnika na ljestvici
icon	TEXT	ikona pehara

Slika 4.7: Entitet **trophies**

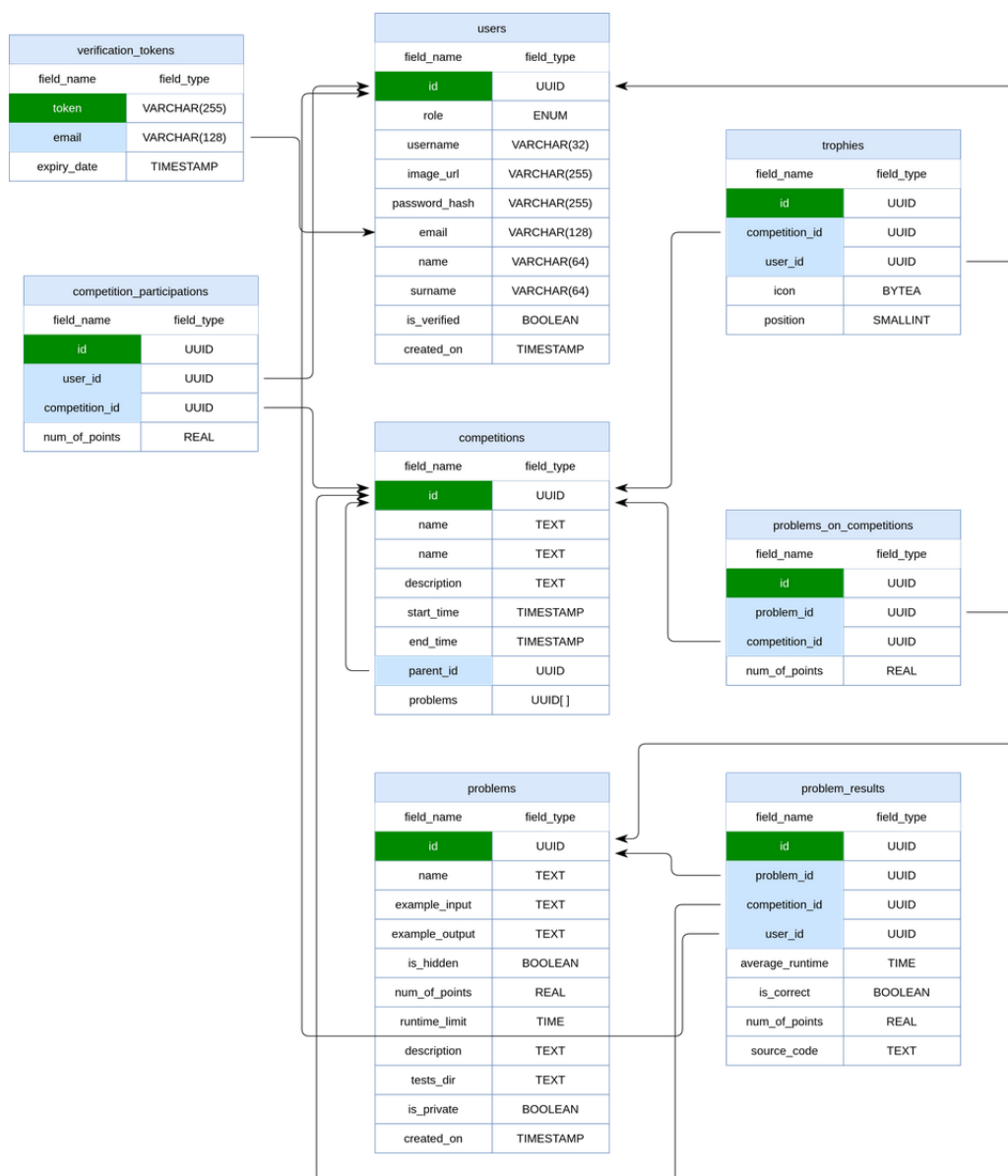
Verification tokens

Entitet **verification_tokens** sadržava informacije o tokenima za verifikaciju korisnika. Sadržava attribute: *token*, *email*, *expiry_date*. Ovaj entitet je u *one-to-many* vezi s entitetom *users* preko atributa *email*.

verification_tokens		
token	VARCHAR(255)	verifikacijski token
email	VARCHAR(128)	adresa e-pošte korisnika
expiry_date	TIMESTAMP	datum isteka verifikacijskog tokena

Slika 4.8: Entitet **verification_tokens**

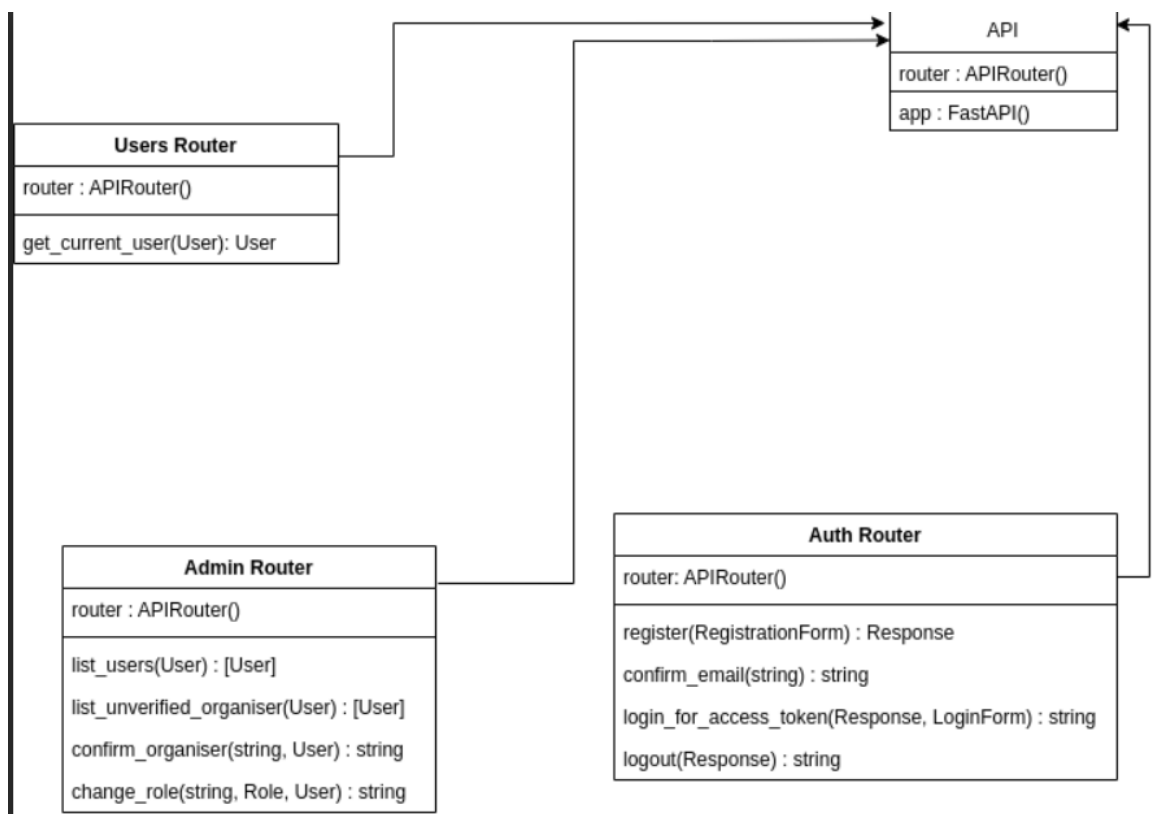
4.1.2 Dijagram baze podataka

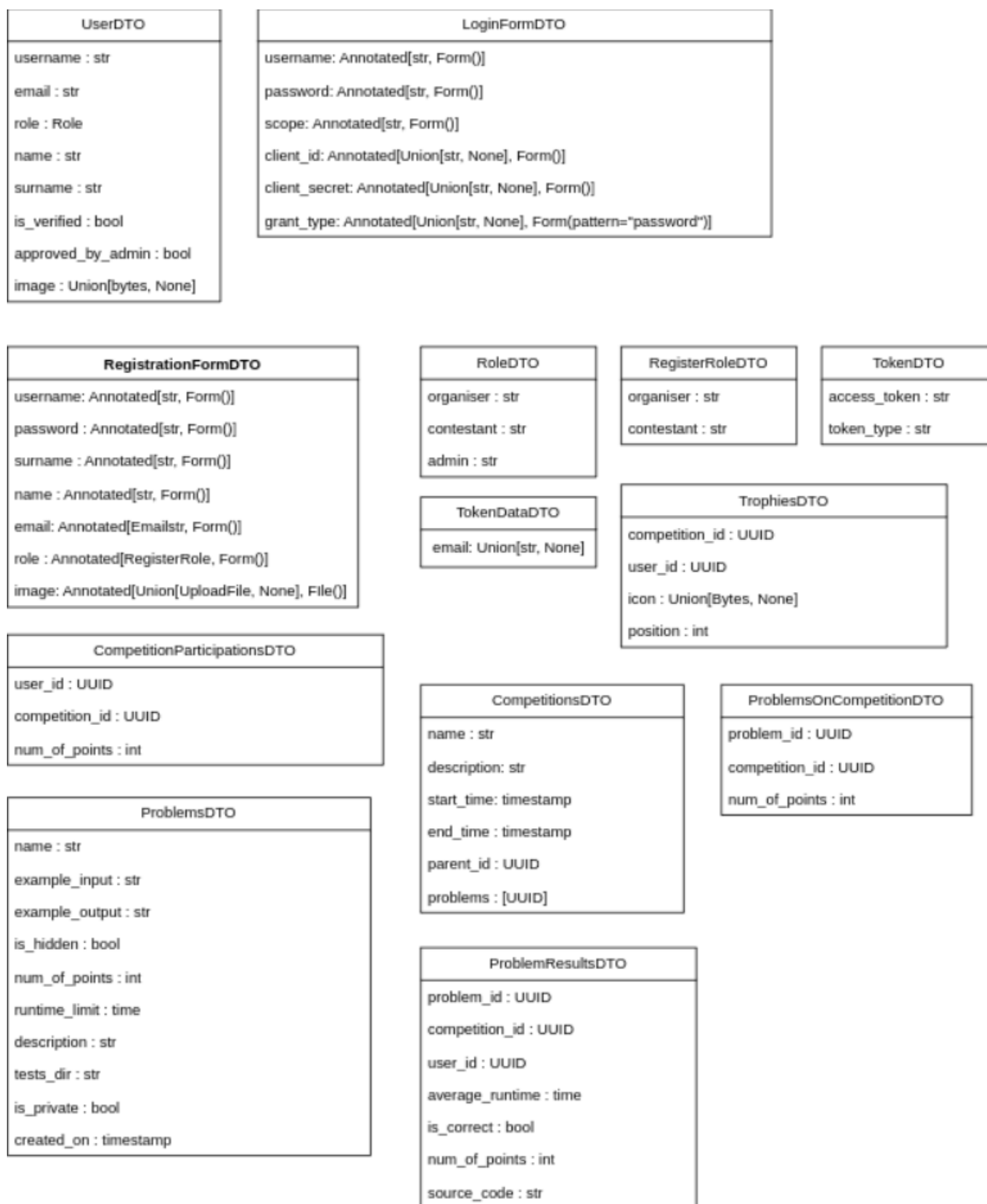


Slika 4.9: Dijagram baze podataka

4.2 Dijagram razreda

Na sljedećim slikama prikazani su dijagrami razreda koji pripadaju backend dijelu aplikacije.





dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

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

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.

5.4 Upute za puštanje u pogon

dio 2. revizije

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Zaključak i budući rad

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

Popis literature

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

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

Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, funkcionalnost natjecatelja, voditelja i ne-registriranog korisnika	19
3.2	Dijagram obrasca uporabe, funkcionalnost administratora	20
3.3	Sekvencijski dijagram kreiranja novog natjecanja	21
3.4	Sekvencijski dijagram kreiranja novog zadatka	22
3.5	Sekvencijski dijagram predaje rješenja zadatka	23
4.1	Entitet users	28
4.2	Entitet problems	29
4.3	Entitet competitions	30
4.4	Entitet competition_participations	30
4.5	Entitet problems_on_competitions	31
4.6	Entitet problem_results	32
4.7	Entitet trophies	33
4.8	Entitet verification_tokens	33
4.9	Dijagram baze podataka	34

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. sastanak

- Datum: u ovom formatu: 18. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 18. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Marko Bolt	Filip Bernt	Teo Matošević	Teo Radolović	Jure Franjković	Fran Sipić	Jakov Vinožganić
Upravljanje projektom	10		4				3
Opis projektnog zadatka	3						
Funkcionalni zahtjevi				3			4
Opis pojedinih obrazaca		2					
Dijagram obrazaca		5					
Sekvencijski dijagrami			1			4	
Opis ostalih zahtjeva					3		
Arhitektura i dizajn sustava					3		
Baza podataka		1				4	
Dijagram razreda					3		
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Marko Bolt	Filip Bernt	Teo Matošević	Teo Radolović	Jure Franjković	Fran Sipić	Jakov VINOŽGANIĆ
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>front end</i>		4	7	7			5
<i>izrada baze podataka</i>			3				4
<i>spajanje s bazom podataka</i>							
<i>back end</i>					6	7	4
<i>deploy</i>	5	1	2				

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.