

Programsko inženjerstvo

Ak. god. 2023./2024.

BytePit

Dokumentacija, Rev. 1

Grupa: *Koder kolege*

Voditelj: *Petra Kelković*

Datum predaje: *17. studenoga 2023.*

Nastavnik: *Hrvoje Nuić*

Sadržaj

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Personalizirana naslovna stranica te header i footer.	Dora Bilić-Pavlinović	28.10.2023.
0.2.1	Opis projektnog zadatka - opći opis	Dora Bilić-Pavlinović	28.10.2023.
0.2.2	Opis projektnog zadatka - slične aplikacije i druge primjene	Matea Cvetković	29.10.2023.
0.2.3	Opis projektnog zadatka - final touches	Filip Mohler	31.10.2023.
0.3.1	Funkcijski zahtjevi	Petra Kelković, Mislav Korotaj	30.10.2023.
0.4.1	Obrasci upotrebe (UC dijagrami) - prvi dio	Mislav Korotaj	31.10.2023.
0.4.2	Obrasci upotrebe - drugi dio	Mislav Korotaj, Nives Ostojić, Filip Mohler	31.10.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
0.4.3	Obrasci upotrebe - treći dio	Filip Mohler	3.11.2023.
0.4.4	Obrasci upotrebe - dijagrami	Filip Mohler	6.11.2023.
0.5.1	Sekvencijski dijagrami	Petra Buršić	6.11.2023.
0.5.2	Sekvencijski dijagrami - nastavak	Petra Buršić	7.11.2023.
0.5.3.	Sekvencijski dijagrami - finalno	Petra Buršić	10.11.2023.
0.6	Unos sastanka: podjela uloga Plan daljnog rada	svi	20.10.2023.
0.7	Opis arhitekture sustava i baze podataka	Nives Ostojić	10.11.2023.
0.8	Ispravak sekvencijskih dijagrama Dodani ostali zahtjevi	Filip Mohler	13.11.2023.
0.9	Popravljena dokumentacija na sastanku	svi	14.11.2023.
0.10	Dijagrami razreda + opis	Matea Cvetković	15.11.2023.
0.11	ER dijagram baze i opis	Filip Mohler	16.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	16.11.2023.

2. Opis projektnog zadatka

Tema našeg projektnog rada je izrada web aplikacije "BytePit" koja omogućuje korisnicima sudjelovanje u programerskim natjecanjima i provjeru riješenih zadataka. Ideja je da naša stranica ima sve potrebno za obavljanje natjecanja poput registracije korisnika, uključivanje u natjecanje, pribavljanje zadataka, vrednovanje priloženih rješenja, prikaz dosadašnjih uspjeha natjecatelja i još mnogo toga.

Neregistrirani korisnik može se registrirati definirajući registrira li se kao **voditelj** ili **natjecatelj**. Za registraciju korisnika potrebno je unijeti

- korisničko ime
- fotografiju
- lozinku
- ime
- prezime
- email adresu

Uspješnost registracije potvrđuje se preko email adrese dok voditelja dodatno potvrđuje i administrator.

Neregistrirani korisnik na web stranici može vidjeti kalendar s natjecanjima kojima je moguće pristupiti te pregledati zadatke na stranici. Također, omogućen mu je uvid u profile natjecatelja i voditelja. Svi registrirani korisnici automatski nasljeđuju sve mogućnosti koje neregistrirani korisnici imaju.

Natjecatelj prisustvuje natjecanjima te dobiva uvid u svoje rezultate. U svrhu pripreme za spomenuta natjecanja, na stranici su dostupni zadatci za vježbu te je dostupna opcija virtualnog natjecanja.

Voditelj ima veće ovlasti od natjecatelja. U njegovim rukama leži zadatak učitavanja novih zadataka na web aplikaciju te organizacija natjecanja. Kada voditelj izradi natjecanje, ono postaje vidljivo u kalendaru dostupnom natjecateljima. Dakle, zadatak voditelja je odrediti težinu natjecateljskog ispita, broj zadataka i dostupno vrijeme za rješavanje istih te postavljanje termina ispita. Ukoliko želi, voditelj može učitati sličicu pehara.

Administrator ima, naravno, najviše ovlasti među navedenima. On može vidjeti popis svih registriranih korisnika zajedno s njihovim osobnim podatcima te

im on onda dodjeljuje prava i po potrebi mijenja osobne podatke. Također, može uređivati sve zadatke i natjecanja koja su voditelji postavili na aplikaciju. Administratorova dužnost je ne zloupotrebljavati osobne podatke korisnika, što je i kažnjivo zakonom.

Na profilu natjecatelja prikazana je statistička obrada njegovih dosadašnjih uspjeha. Stoga mu na profilu možemo vidjeti koliko je zadatka uspješno riješio a koliko ih je pokušavao riješiti te koliko je natjecanja osvojio. Za svako osvojeno natjecanje, na profilu će mu biti prikazana po jedna sličica pehara. Profil voditelja prikazuje popis zadataka koje je on učitao te natjecanja koje je on organizirao.

Provedba natjecanja

Kada dođe vrijeme koje je voditelj postavio kao početak natjecanja, zadatci ispita postaju vidljivi aktivnim natjecateljima. Za svaki zadatak natjecatelji prilažu datoteke s programskim kodom. Na ispitu stoje postavljena vremenska ograničenja za trajanje ispita i nakon njegovog isteka objavljuju se rezultati. Rezultati se prikazuju oblikom rang liste svih učesnika poredanih silazno po prikupljenom broju bodova. Pri kalkulaciji broja bodova uzima se u obzir postotak točnosti i isteklo vrijeme. Onima koji su se plasirali na prva tri mesta pridodaje se slika pehara na njihovom profilu. Natjecatelju je po završetku ispita pridodan i uvid u sva priložena rješenja nekog drugog natjecatelja. Također, može vidjeti statistiku svakog pojedinih zadataka uključujući prosječno vrijeme rješavanja, popis natjecatelja koji su ga rješavali i sl.

Virtualno natjecanje

Virtualno natjecanje je koncept osmišljen kako bi natjecatelji mogli provjeriti koliko su se dobro pripremili za nadolazeće natjecanje. Dakle, kada korisnik želi provjeriti svoju spremnost samo ode u kalendar, odabere neko natjecanje koje je provedeno u prošlosti te pokrene virtualno natjecanje nad njim. Tada će iz korisnikove perspektive sve izgledati kao da on sudjeluje na tom natjecanju. Bit će mu pružen isti ispit i po završetku rješavanja biti će rangiran među natjecateljima koji su taj ispit službeno rješavali. Korisnik tako dobiva informaciju kakav bi bio njegov rezultat da je taj dan uistinu sudjelovao na natjecanju.

Slične aplikacije

Već postojeća aplikacija vrlo slična ovoj je Edgar koji se koristi na FER-u za provođenje

ispita i laboratorijskih vježbi. S obzirom na to da je svrha te aplikacije ipak drugačija od naše, postoje neke značajne razlike. Dok se za registrirani pristup našoj aplikaciji korisnik sam prijavljuje i čeka potvrdu administratora, u Edgaru to čini administrator samostalno dodavajući korisnike (kojima se kasnije dodijele njihovi pristupni podaci). Zbog same razlike u namjeni, predana rješenja se drugačije budu (nekim stalnim brojem bodova, bez ovisnosti o vremenu). Također, studentu prijavljenom u sustav nije omogućen pregled tuđih rješenja kao što je to slučaj u našoj aplikaciji, kao ni pristup pojedinačnim zadacima: moguće je samo pokrenuti probni ispit ili vježbu, bez mogućnosti odabira pojedinog zadatka (slika ??).

The screenshot shows the BytePit application's user interface. At the top, there is a navigation bar with links for 'My previous exams', 'My tickets', 'My stats', 'Playground', 'Code theme', and a dropdown for 'B_BP' and '2022/2023'. On the right side, there is a user profile section with a placeholder 'Ime Prezime Student' and a 'Logout' button. Below the navigation bar, there is a search bar with 'Password...' and a 'Start exam' button. The main content area displays two sections: 'You have unsubmitted private exams:' and 'Public exams:'. The 'You have unsubmitted private exams:' section shows one entry: '# 1 Title: 3. laboratorijska vježba Started at: 2023-05-25 19:03:52 Available until: 2023-06-02 11:00:00 Can continue? Can't continue: expired.' The 'Public exams:' section shows five entries:

#	Academic year	Exam title, questions no	Runs/Max	Duration	Score ignored	Forward only	Used in stats	Global	Public	Available for	Start exam
1	2022/2023	Vježba: Završni 22/23 9 question(s)	0/0	3 hours						9 years expires: 2023-09-27 12:00:00	
2	2022/2023	Vježba: Meduispit 22/23 8 question(s)	0/0	2 hours						9 years expires: 2023-09-27 12:00:00	
3	2019/2020	Vježba: Završni ispit 19/20 8 question(s)	0/0	3 hours						9 years expires: 2023-09-27 12:00:00	
4	2020/2021	Vježba: Dekanski rok 20/21 10 question(s)	0/0	2 hours						9 years expires: 2023-09-27 12:00:00	
5	2020/2021	Vježba: Jesenski rok 20/21	0/0	3 hours						9 years	

Slika 2.1: Edgar: početna stranica i popis vježbi za ispite

Ostale funkcionalnosti BytePita vrlo su slične Edgaru: uloge natjecatelja i studenta su slične, oni mogu učitavati i provjeravati svoj kod, pokrenuti probni ispit (slika ??) (u BytePitu virtualno natjecanje) kao i pristupiti ispitu (odnosno natjecanju). Koncept natjecanja i ispita vrlo je sličan - korisnicima su dostupni svi ispitni zadaci istovremeno, a po završetku se ti zadaci objavljuju na stranici za vježbu. Ono što u BytePitu predstavlja uloga voditelja, u Edgaru je asistent/profesor koji ima ovlasti objavljivanja tj. izrade zadataka i organizacije ispita (odabir zadataka, trajanja). U Edgaru čak postoji i stranica sa statistikom koja prikazuje uspješnost u odnosu na druge studente, postotak točno riješenih zadataka i sl.(slika ??). BytePit ima stranicu slične namjene, ali ipak s drugačijim podacima: na njoj natjecatelj može vidjeti tuđa rješenja i njihovu uspješnost, kao i svoj rang.

Vježba Završni 22/23 (293554)

1 (-) 2 (-) 3 (-) 4 (-) 5 (-) 6 (-) 7 (-) 8 (-) 9 (-)

ime Prezime
e-mail
JMBAG

02 : 29 : 48

Submit

3. [4 pts Correct] [0 pts Unanswered] [0 pts Incorrect]

Zadatak 3. (4 bodova)

Administrator baze podataka Streamflix želi stvoriti novu ulogu MOVIE_CLEANER koja će obavljati sljedeće poslove: - pregled svih stupaca iz tablice TRACK - ažuriranje svih neključnih atributa iz tablice TRACK osim tipa sadržaja - brisanje redaka iz tablice TRACK

pri čemu su navedeni poslovi obavljaju isključivo nad redcima iz tablice TRACK vezanim uz filmske sadržaje koje je u zadnje dvije godine gledalo manje od pet različitih korisnika (profila).

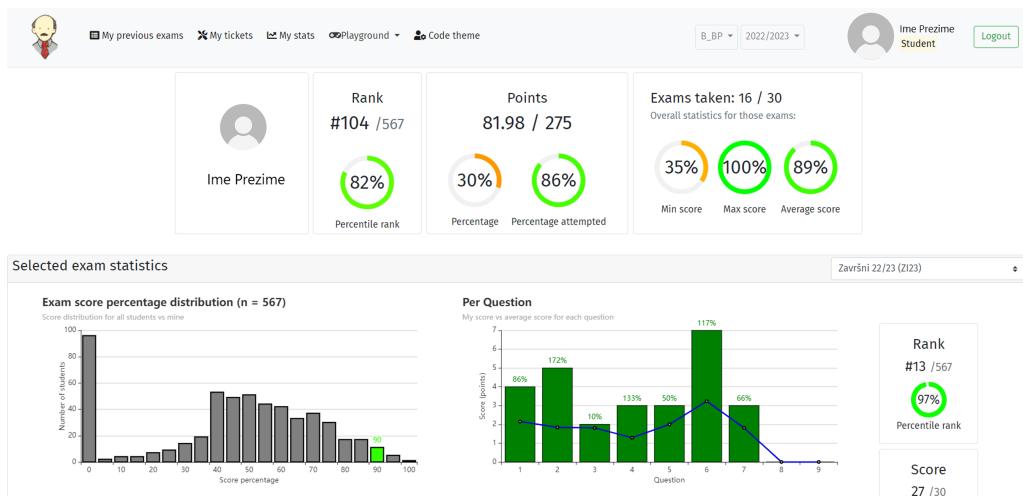
Napomita se da naredbi kojima će administrator ovo omogućiti te potom ulogu dodati postojecem korisniku naziva PERO. Pretpostavite slijedeće - tablica TRACK nalazi se u PUBLIC shemi baze podataka STREAMFLIX - vremenski interval provjere broja gledanja kao referentne trenutke uzmite trenutak izvodjenja upita i trenutak početka gledanja - uloga MOVIE_CLEANER u svojim upitima će za navedene poslove koristiti virtualnu tablicu naziva TRACK_CLEANER - brisanjem sadržaja redaka tablice TRACK provesti će se kaskadno brisanje redaka koji referenciraju navedene retke - korisnik PERO postoji u sustavu, ali nema mogućnost spajanja na navedenu bazu niti pristup navedenoj shemi

NAPOMENA: U zadatku se ne smiju koristiti procedure i okidači.

Save

Edgar: On-Line Exam Web Application © Developed with ❤️ @FER

Slika 2.2: Edgar: probni ispit



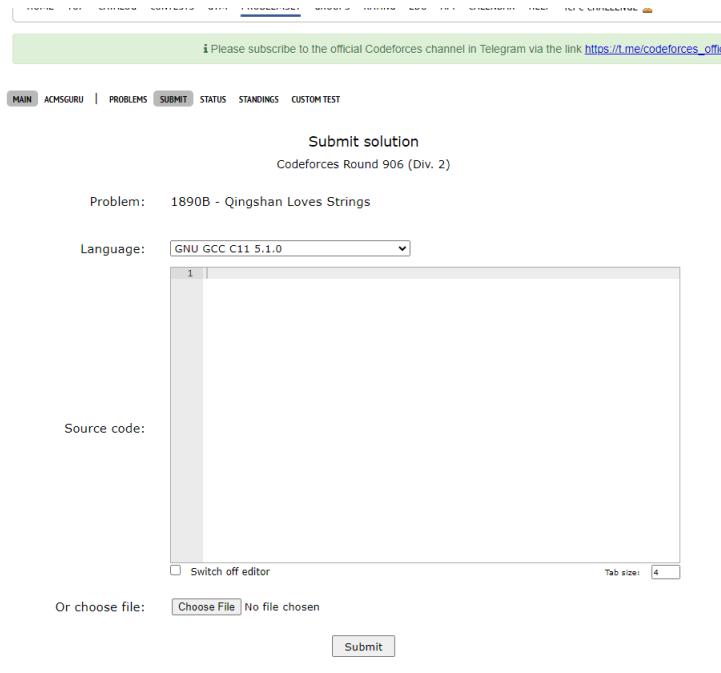
Slika 2.3: Edgar: stranica sa statistikom

Osim Edgara, postoji još niz aplikacija sličnih BytePitu, a jedna od njih je Codeforces, web aplikacija koja omogućuje sudjelovanje u online natjecanjima. Gotovo i da nema razlike među ovim aplikacijama: na profilima korisnika vidljiva je njihova statistika, omogućen je pristup virtualnim natjecanjima koja simuliraju prava, vidljiva je lista problema kao i njihovih rješenja koja su učitali korisnici (slika ??)... Ta rješenja nisu uvijek vidljiva, vidljivost ovisi o postavkama natjecanja tako je da ovisno o sudjelovanju nekim korisnicima onemogućen pregled predanih rješenja. Nasuprot tomu, u BytePitu rješenja može dohvatiti samo natjecatelj koji je i sam točno riješio zadatak. Bitna razlika u ovom je slučaju također i to što Codeforces omogućava svim korisnicima da učitaju zadatke, koji potom prolaze dodatne pro-

vjere da bi se utvrdila njihova ispravnost, dok je u BytePitu ta mogućnost otvorena samo voditeljima, i to bez dodatnih provjera nakon objave zadatka. Na profilima korisnika koji su zadatke učitali ti zadaci nisu vidljivi (u BytePitu se oni nalaze na profilima voditelja). Slično kao u našoj aplikaciji, nakon natjecanja moguće je na profilima korisnika vidjeti njihova rješenja i rezultate testova ali čak i bez registracije: svaki korisnik vidi rješenja svakog korisnika te nije potrebna registracija. Ono što registracija omogućuje je, dakako, sudjelovanje u natjecanjima i virtualnim natjecanjima te izvršavanje i predaja koda za riješene zadatke za vježbu (slika ??) (neregistrirani korisnik može vidjeti tekst zadatka, ali ne može izvršiti kod i time provjeriti točnost rješenja). Ono što BytePit omogućuje, a Codeforces ne je mogućnost virtualnog natjecanja koje se sastoji od nasumičnih zadatka: u potonjem se nude samo replike stvarnih natjecanja koje se mogu pokrenuti.

Contest status							
#	When	Who	Problem	Lang	Verdict	Time	Memory
230466190	Oct/30/2023 09:20 UTC+1	eugalt	1890B - Qingshan Loves Strings	Python 3	Accepted	187 ms	41700 KB
230465212	Oct/30/2023 09:10 UTC+1	eugalt	1890B - Qingshan Loves Strings	Python 3	Accepted	62 ms	0 KB
230205402	Oct/29/2023 04:29 UTC+1	eugalt	1890B - Qingshan Loves Strings	Python 3	Accepted	62 ms	100 KB
230492736	Oct/30/2023 13:15 UTC+1	rajvirslinigh192002	1890B - Qingshan Loves Strings	Python 3	Accepted	170 ms	41700 KB
230379325	Oct/29/2023 15:18 UTC+1	Sparkle_Twilight	1890B - Qingshan Loves Strings	Python 3	Accepted	46 ms	100 KB
230279886	Oct/28/2023 20:14 UTC+1	eugalt	1890B - Qingshan Loves Strings	Python 3	Accepted	46 ms	0 KB
230215478	Oct/28/2023 16:18 UTC+1	eugalt	1890B - Qingshan Loves Strings	Python 3	Accepted	62 ms	0 KB
230383099	Oct/29/2023 15:48 UTC+1	Leo25Darklight	1890B - Qingshan Loves Strings	PyPy 3-64	Accepted	202 ms	7400 KB
230222712	Oct/28/2023 16:31 UTC+1	_M_H_M_	1890B - Qingshan Loves Strings	Python 3	Accepted	77 ms	0 KB
230192760	Oct/28/2023 15:48 UTC+1	emrakul	1890B - Qingshan Loves Strings	PyPy 3	Accepted	249 ms	5600 KB
230274516	Oct/28/2023 19:29 UTC+1	MdNazmulHossain	1890B - Qingshan Loves Strings	Python 3	Accepted	61 ms	0 KB
230211866	Oct/28/2023 16:12 UTC+1	just_sai	1890B - Qingshan Loves Strings	PyPy 3-64	Accepted	670 ms	10900 KB
230377483	Oct/29/2023 15:03 UTC+1	Aldibek	1890B - Qingshan Loves Strings	PyPy 3-64	Accepted	670 ms	10900 KB

Slika 2.4: Codeforces: rješenja različitih korisnika



Slika 2.5: Codeforces: mogućnost izvršavanja koda

BytePit je aplikacija koja svakako može imati širu primjenu od ovdje opisane: dok je trenutna verzija aplikacije pogodna uglavnom za programerska natjecanja, s manjim preinakama ona bi se mogla koristiti u razne svrhe. Svakako bi bila dobra ideja koristiti aplikaciju kao svojevrsni test pri zapošljavanju programera odnosno za selekciju najboljih kandidata: kandidati bi dobili podatke za pristup i od njih bi se tražilo da riješe određen broj zadataka (naravno, drugačije vrste od natjecateljskih). Tako bi se lakše probralo bolje kandidate koji ulaze u uži izbor za određenu poziciju. Prilagodbom težine zadataka, BytePit bi mogao postati i platforma za vježbu i učenje programiranja. Naravno, u tom bi slučaju bilo potrebno osmisiliti i kratke tečajeve programiranja, kao što to postoji na npr. Codecademy-u. Aplikacija bi se mogla proširiti i na način da bude slična gore predstavljenoj aplikaciji Edgar: mogla bi biti platforma za ispite iz programiranja, kako na fakultetu, tako i u osnovnim i srednjim školama.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Naručitelj
2. Voditelj natjecanja
3. Natjecatelj
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) pregledavati programske zadatke objavljene na stranici
 - (b) vidjeti kalendar s dostupnim natjecanjima
 - (c) pregledavati profile drugih korisnika (natjecatelja i voditelja natjecanja)
 - (d) sortirati učitane zadatke na profilu voditelja natjecanja
 - (e) poslati zahtjev za registracijom za koji mora priložiti sljedeće informacije: uloga za koju se prijavljuje (voditelj natjecanja ili natjecatelj), korisničko ime, fotografija, lozinka, ime, prezime i email adresa
2. Natjecatelj (inicijator) može:
 - (a) za vrijeme trajanja natjecanja:
 - i. vidjeti aktualne zadatke
 - ii. poslati datoteku s programskim kodom za svaki zadatak
 - (b) nakon natjecanja:
 - i. vidjeti popis učitanih rješenja drugih natjecatelja
 - ii. za svaki pojedini zadatak vidjeti popis svih natjecatelja koji su učitali rješenje za taj zadatak, broj točnih primjera po najboljem učitavanju od natjecatelja i prosječno vrijeme izvršavanja po primjeru

- iii. dohvatiti učitano rješenje za pojedini zadatak ukoliko je on u potpunosti točno riješen
- (c) vježbati prethodno objavljene zadatke
- i. učitati rješenje zadatka u aplikaciju
- (d) pokrenuti virtualno natjecanje:
- i. odabirom prošlog natjecanja iz kalendara
 - ii. odabirom opcije rješavanja nasumičnih zadataka iz prethodnih natjecanja
3. Voditelj natjecanja (inicijator) može:
- (a) učitati nove zadatke u aplikaciju
 - (b) organizirati natjecanje:
 - i. odabire vrijeme početka i završetka
 - ii. određuje broj zadataka
 - iii. odlučuje koji su zadaci aktivni
 - iv. po želji učitava sličicu pehara
 - (c) izraditi zadatak
 - (d) uređivati vlastito objavljene zadatke i natjecanja (to ne mijenja prijašnje rezultate)
4. Administrator (inicijator) može:
- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
 - (b) mijenjati dodijeljena prava registriranim korisnicima
 - (c) mijenjati osobne podatke registriranih korisnika
 - (d) potvrditi/odbiti registracijski zahtjev za ulogu voditelja
 - (e) uređivati sve zadatke i natjecanja
5. Baza podataka (sudionik):
- (a) pohranjuje sve podatke o registriranim korisnicima
 - (b) briše podatke o korisnicima koji nisu potvredili registracijski mail unutar 24 sata
 - (c) briše podatke o voditeljima koje nije potvratio administrator unutar 7 dana
 - (d) pohranjuje opise svih zadataka i njihova rješenja
 - (e) pohranjuje sve podatke o već završenim natjecanjima i njihove rang liste

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registracija novog korisnika
- **Sudionici:** Baza podataka, Administrator
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Otvara se forma u koji upisuje podatke:
 - (a) korisničko ime
 - (b) fotografija
 - (c) lozinka
 - (d) ime i prezime
 - (e) email adresa
 - (f) odabire: voditelj natjecanja / natjecatelj
 3. Upisuje i odabire potrebne podatke
 4. Korisnik dobiva poruku da potvrdi registraciju putem email adrese
 5. Korisniku se šalje mail za potvrdu registracije
 6. Korisnik potvrđuje registraciju putem linka te se otvara stranica s porukom dobrodošlice
- **Opis mogućih odstupanja:**
 - 2.a Email/korisničko ime su već zauzeti
 1. Korisnik dobiva poruku da je mail/korisničko ime u upotrebi
 2. Traži se ponovni upis podataka
 - 2.f Korisnik se registrira kao "voditelj natjecanja"
 1. Administrator dobiva poruku u kojoj (ne)potvrđuje registraciju
 - 5.a Korisnik ne potvrđuje email unutar 24 sata
 1. Korisnikov profil se briše iz baze podataka

UC2 - Prijava u sustav

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Dobiti pristup korisničkom sustavu
- **Sudionici:** Baza podataka

- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju prijava
 2. Unos korisničkog imena i lozinke
 3. Potvrda od sustava o ispravnim podatcima
 4. Pristup korisničkim opcijama
- **Opis mogućih odstupanja:**
 - 3.a Neispravno korisničko ime ili lozinka
 1. Korisnik dobiva poruku o neispravnom korisničkom imenu ili lozincki

UC3 - Pregled zadataka za vježbu

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled svih zadataka u aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju Zadatci za vježbu
 2. Otvara se stranica s popisom svih zadataka
 3. Korisnik odabire zadatak
 4. Otvara se stranica sa odabranim zadatkom te se prikazuje tekst zadatka
- **Opis mogućih odstupanja:**
 - 4.a Neregistrirani korisnik želi predati rješenje zadatka
 1. Korisnik dobiva poruku da je za nastavak potrebna registracija

UC4 - Pregled kalendara

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled kalendara sa svim natjecanjima
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju kalendar
 2. Otvara se stranica sa prikazom mjesečnog kalendara
- **Opis mogućih odstupanja:**
 - 2.a Neregistrirani korisnički želi pristupiti natjecanju
 1. Korisnik dobiva poruku da je za nastavak potrebna registracija

UC5 - Pregled svih korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled pojedinog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju korisnici
 2. Korisniku se nude dvije opcije:
 - (a) Natjecatelji
 - (b) Voditelji

UC6 - Pregled svih natjecatelja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled podataka pojedinog natjecatelja
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. UC5 - Pregled svih korisnika
 2. Korisnik odabire opciju natjecatelji
 3. Korisnik dobije listu svih natjecatelja

UC7 - Podatci o natjecatelju

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled podataka pojedinog natjecatelja
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. UC6 - Pregled svih natjecatelja
 2. Korisnik odabire profil natjecatelja
 3. Prikazuju se podatci o natjecatelju:
 - (a) Statistika o broju točno riješenih zadataka
 - (b) Statistika o broju isprobanih zadataka
 - (c) Pehari za osvojena natjecanja

UC8 - Pregled svih voditelja

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregled podataka pojedinog voditelja
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. UC5 - Pregled svih korisnika
 2. Korisnik odabire opciju voditelji
 3. Korisnik dobije listu svih voditelja

UC9 - Podatci o voditelju

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled podataka pojedinog voditelja
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. UC8 - Pregled svih voditelja
 2. Korisnik odabire profil voditelja
 3. Prikazuju se podatci o voditelju:
 - (a) Svi objavljeni zadatci voditelja
 - (b) Kalendar sa svim natjecanjima voditelja

UC10 - Sudjelovanje u natjecanju

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljeni korisnik: natjecatelj
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire aktivno natjecanje
 2. Otvara se stranica s natjecanjem
 3. U trenutku početka natjecanja, svi zadaci postaju vidljivi
 4. Natjecatelj rješava zadatke unutar vremenskog ograničenja
 5. Natjecatelj predaje rješenja zadatka

UC11 - Pregled rješenosti drugih natjecatelja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pregled rješenja i statistika drugih natjecatelja
- **Sudionici:** Baza podataka

- **Preduvjet:** Natjecatelj pristupio natjecanju
- **Opis osnovnog tijeka:**
 1. Natjecatelj otvara pojedini zadatak
 2. Prikazuju se sljedeći podatci:
 - (a) Svi natjecatelji koji su učitali neko rješenje
 - (b) Broj točnih primjera natjecatelja
 - (c) Prosječno vrijeme izvršavanja po primjeru
 - (d) Gumb za dohvrat učitanog rješenja
- **Opis mogućih odstupanja:**
 - 2.d Natjecatelj nije u potpunosti točno riješio zadatak
 1. Sustav onemogućuje gumb

UC12 - Virtualno natjecanje

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Simulacija natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljeni korisnik: natjecatelj
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire opciju virtualno natjecanje
 2. Natjecatelju se otvaraju dvije opcije:
 - (a) Prethodno natjecanje
 - (b) Nasumično odabrani zadaci

UC13 - Prethodno natjecanje

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Simulacija prethodnog natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljeni korisnik: natjecatelj
- **Opis osnovnog tijeka:**
 1. UC12 - Virtualno natjecanje
 2. Natjecatelj odabire opciju Prethodno natjecanje
 3. Preusmjeravanje na kalendar
 4. Natjecatelj bira natjecanje
 5. Natjecatelja se po završetku natjecanja rangira u odnosu na službene rezultate originalnog natjecanja

UC14 - Nasumično odabrani zadaci

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Simulacija prethodnog natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljeni korisnik: natjecatelj
- **Opis osnovnog tijeka:**
 1. UC12 - Virtualno natjecanje
 2. Natjecatelj odabire opciju Nasumično odabrani zadaci
 3. Aplikacija ravnomjerno odabire zadatke po težini

UC15 - Rješavanje zadataka za vježbu

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Vježbanje zadataka za natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire opciju zadatci za vježbu
 2. Otvara se stranica s popisom svih zadataka
 3. Korisnik odabire zadatak
 4. Otvara se stranica sa zadatkom
 5. Natjecatelj može pogledati rješenje zadatka
 6. Nakon rješavanja, natjecatelj odabire opciju za evaluaciju rješenja
 7. Natjecatelj dobiva informaciju o točnosti svog rješenja

UC16 - Izrada natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Napraviti natjecanje za korisnike
- **Sudionici:** Natjecatelji, baza podataka
- **Preduvjet:** Prijavljeni korisnik: voditelj
- **Opis osnovnog tijeka:**
 1. Voditelj odabire opciju izrade natjecanja
 2. Odabire vrijeme početka i završetka natjecanja
 3. Bira broj zadataka
 4. Unosi tekst zadatka
 5. Bira broj bodova za zadatak
 6. Odabire po želji sličicu pehara

UC17 - Uređivanje svojih natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Ispraviti pogreške ili dodati zadatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljeni korisnik: voditelj
- **Opis osnovnog tijeka:**
 1. Voditelj odabire opciju uredi natjecanje
 2. Otvara se lista natjecanja koje je voditelj izradio
 3. Voditelj odabire željeno natjecanje
 4. Voditelj radi promjene
 5. Sprema promjene

UC18 - Pregled i uređivanje korisničkih podataka

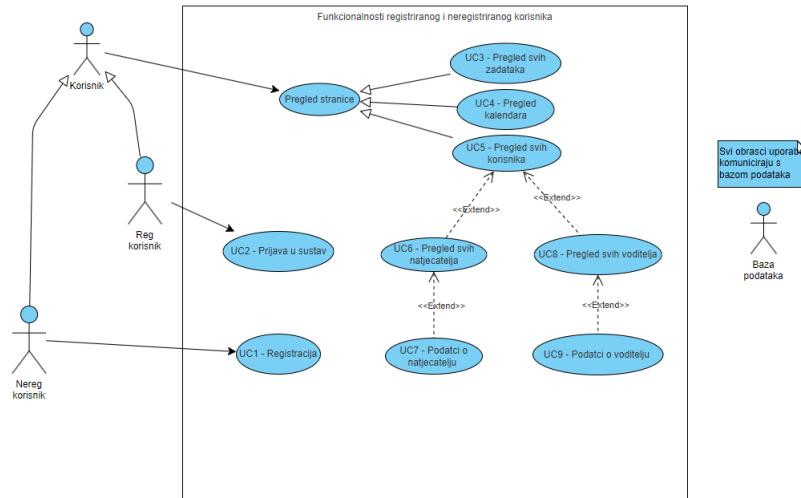
- **Glavni sudionik:** Administrator
- **Cilj:** Učinkovita administracija i održavanje sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik: administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled svih registriranih korisnika
 2. Otvara se stranica s popisom svih registriranih korisnika i njihovih podataka
 3. Administrator mijenja podatke ili prava korisnika
 4. Sprema promjene
 5. Korisniku se šalje mail o promijeni njegovih podataka.
- **Opis mogućih odstupanja:**
 - 3.a Administrator narušava integritet jedinstvenosti (npr. mijenja korisničko ime u već postojeće)
 1. Administrator dobiva poruku da je korisničko ime već u uporabi
 2. Traži se ponovni upis podataka

UC19 - Uređivanje svih natjecanja/zadataka

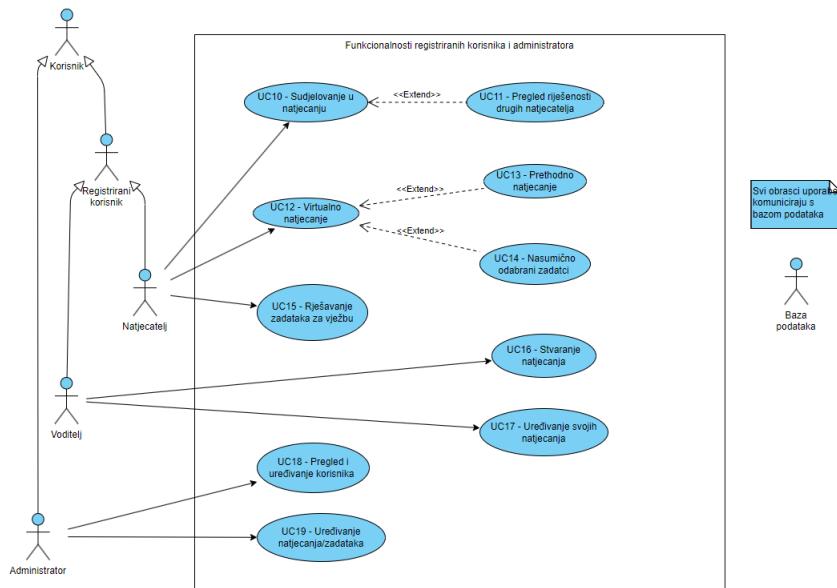
- **Glavni sudionik:** Administrator
- **Cilj:** Ispravljanje grešaka ili nesporazuma u natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik: administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled svih natjecanja

2. Otvara se stranica s popisom svih natjecanja
3. Administrator odabire natjecanje
4. Otvaraju se podaci o natjecanju i popis zadataka tog natjecanja
5. Administrator uređuje podatke o natjecanju ili zadatke tog natjecanja
6. Administrator spremi promjene

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost reg. i nereg. korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost voditelja, natjecatelja i administratora

3.1.2 Sekvencijski dijagrami

1) Registracija korisnika

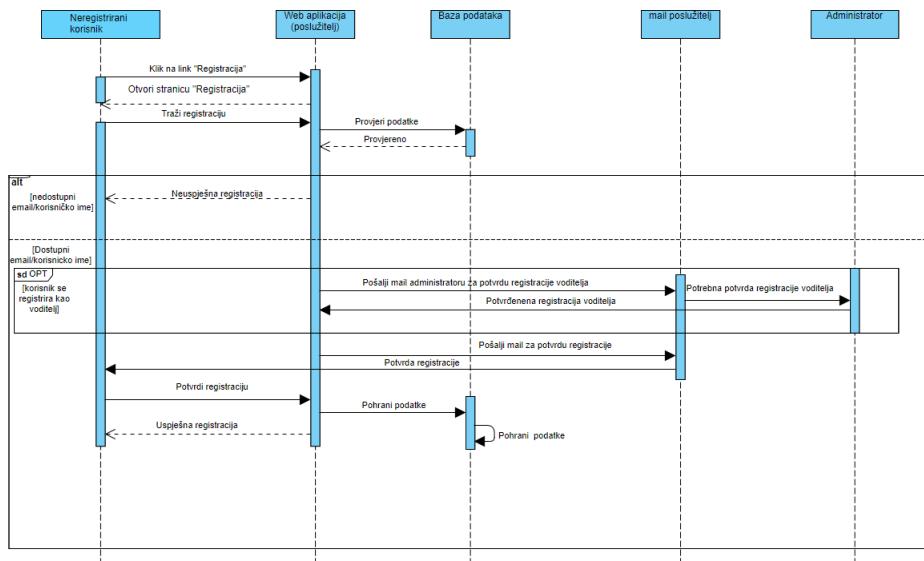
Opis dijagrama: Korisnik započinje registraciju odabirom opcije "Registracija" na korisničkom sučelju. Nakon što korisnik odabere tu opciju, poslužitelj aplikacije šalje zahtjev korisniku da unese sljedeće podatke:

- (a) korisničko ime
- (b) fotografiju
- (c) lozinku
- (d) ime i prezime
- (e) email adresu
- (f) odabir: voditelj natjecanja / natjecatelj

Nakon što korisnik unese navedene podatke, šalje zahtjev za registraciju putem korisničkog sučelja. Nakon toga, poslužitelj provjerava unesene podatke. Provjerava se dostupnost e-mail adrese i korisničkog imena u bazi podataka. U slučaju da su e-mail adresa ili korisničko ime već zauzeti, poslužitelj obavještava korisnika da registracija nije uspjela. U slučaju da su e-mail ili korisničko ime adresa dostupni, korisniku se šalje zahtjev za potvrdom registracije putem e-maila. Korisnik potvrđuje registraciju - podaci se trajno pohranjuju u bazu podataka, a poslužitelj obavještava korisnika da je registracija uspješna.

U slučaju da se korisnik registrira kao "voditelj natjecanja," tada se e-mail zahtjev za potvrdom registracije šalje i administratoru.

Slika dijagrama:



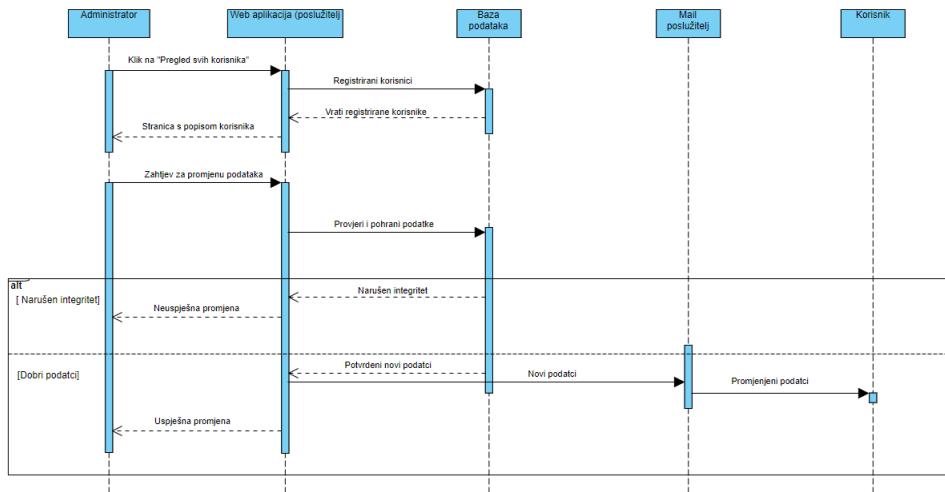
Slika 3.3: Sekvencijski dijagram registracije novog korisnika

2) Pregled i uređivanje korisnika

Opis dijagrama: Administrator šalje zahtjev za pregled korisnika i njihovih podataka tako da klikne na link "Pregled svih korisnika". Poslužitelj šalje upit bazi podataka za podatke registriranih korisnika, a baza podataka vraća podatke o registriranim korisnicima. Otvara se stranica s popisom registriranih korisnika i njihovim podacima koje je moguće urediti ili izbrisati. Kada administrator pošalje zahtjev za izmjenom podataka nekog korisnika, baza podataka provjerava je li narušen integritet jedinstvenosti, odnosno je li administrator promijenio korisničko ime u drugo korisničko ime koje već postoji u bazi podataka. Iz ovog scenarija proizlaze dva slučaja:

- Promjenom nije narušen integritet - novi podaci se pohranjuju u bazu podataka. Administratoru se javlja da je promjena uspješna, te se istovremeno šalje e-mail putem mail-poslužitelja korisniku s njegovim novim podacima.
- Promjenom je narušen integritet - administratoru se javlja da promjena nije uspjela (korisničko ime već postoji u bazi podataka).

Slika dijagrama:



Slika 3.4: Sekvencijski dijagram pregleda i uređivanja korisnika

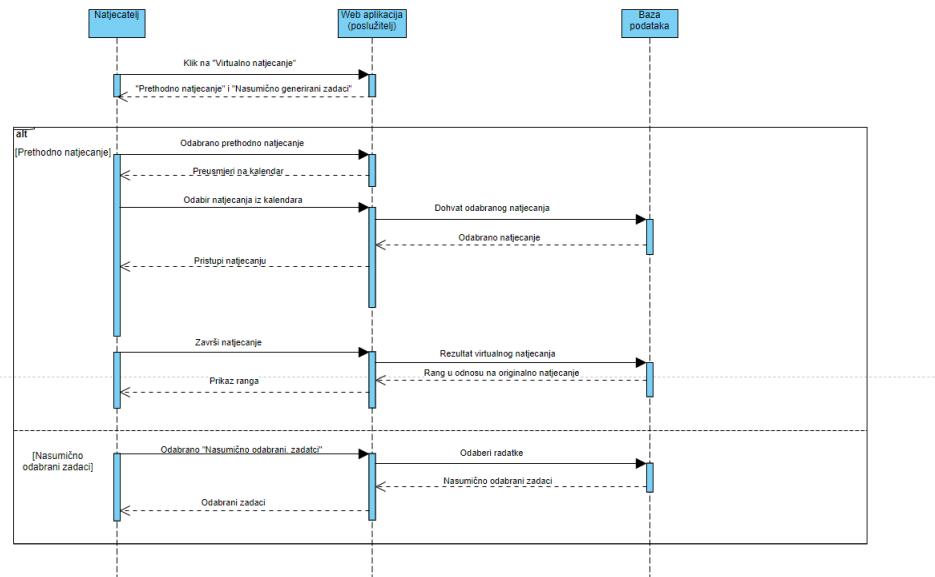
3) Virtualno natjecanje

Opis dijagrama: Kada natjecatelj klikne na link/gumb "Virtualno natjecanje", poslužitelj mu nudi mogućnost odabira između "Prethodno natjecanje" i "Nasumično odabrani zadaci".

Ako natjecatelj odabere "Prethodno natjecanje", poslužitelj ga preusmjerava na kalendar natjecanja, gdje natjecatelj može birati prethodna natjecanja.

Ako natjecatelj odabere opciju "Nasumično odabrani zadaci", poslužitelj iz baze podataka dohvaća zadatke i odabire ih ravnomjerno prema težini. Nakon odabiranja zadataka, zadaci se prikazuju natjecatelju.

Slika dijagrama:



Slika 3.5: Sekvencijski dijagram virtualnog natjecanja

3.2 Ostali zahtjevi

- Sustav treba biti funkcionalan na bilo kojem web pregledniku
- Sustav mora biti brz i responzivan
- Dohvat zadataka ili korisnika iz baze podataka mora se obaviti u konačnom vremenu (manji od 5 sekunde)
- Sustav mora podržavati hrvatsku abecedu pri unosu i prikazu tekstualnog sadržaja
- Sustav mora podržavati rad jednog ili više korisnika istovremeno
- Pristup sustavu mora biti omogućen preko javne mreže pomoću HTTPS
- Nadogradnja sustava ne smije narušiti postojeće funkcionalnosti sustava
- Sustav podržava format slike jpeg (maksimalna veličina 1048576 bajtova)

4. Arhitektura i dizajn sustava

Arhitektura ima 3 podsustava:

- *Web preglednik*
- *Web aplikacija na web poslužitelju*
- *Baza podataka - PostgreSQL*

Web preglednik je ujedino i prevoditelj koda koji korisniku omogućuje pregleđavanje sadržaja web aplikacije i interakciju s istim.

Web poslužitelj prima HTTP (engl. Hyper Text Transfer Protocol) zahtjeve od klijenta (preglednika) koji sadrže informacije o tome što klijent traži, kao što su primjerice URL, GET, POST... i vraća dohvaćeni resurs ako ga ima te vraća statusni kod koji daje informacije o uspješnosti zahtjeva.

Tehnologije korištene u našoj aplikaciji jesu Spring Boot i React. Aplikacija se sastoji od serverske komponente napisane u Javi (Spring Boot) i klijentske komponente napisane u JavaScriptu (React). Za razvojno okruženje koristimo IntelliJ, a baza koju koristimo za spremanje podataka o registriranim korisnicima i sve informacije o natjecanjima je PostgreSQL.

Web aplikacija temelji se na arhitekturi Model-View-Controller (MVC). Ova arhitektura omogućuje organizaciju aplikacije u tri ključne komponente:

- **Model:** Ova komponenta predstavlja poslovnu logiku i podatke aplikacije. Model je implementiran u Java programskom jeziku (Spring Boot) i odgovoran je za upravljanje podacima, komunikaciju s bazom podataka te izračune i obrade podataka.
- **View:** Komponenta za prikaz podataka korisnicima. Prikazi se generiraju u Reactu, a omogućuju korisnicima interakciju s aplikacijom putem web preglednika. Prikazi se oblikuju pomoću HTML-a, CSS-a i JavaScripta.
- **Controller:** Kontroler je posrednik između Modela i Viewa. Ova komponenta upravlja korisničkim zahtjevima, prima ulazne podatke od korisnika te izvršava odgovarajuće akcije u Modelu. Kontroler također određuje koji

prikaz treba biti poslan korisnicima.

4.1 Baza podataka

Koristimo relacijsku bazu podataka čije su gradivne jedinke tablice definirane imenom i skupom atributa za jednostavno upravljanje podacima. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- *Korisnik*
- *Natjecanje*
- *Zadatak*
- *Zadaci na natjecanju*
- *Primjeri za evaluaciju*
- *Slike*

4.1.1 Opis tablica

Korisnik - entitet sadržava sve važne informacije o korisniku: Korisničko ime, lozinku, ime, prezime, identifikator slike, email, tip korisnika (natjecatelj/voditelj), individualni sažetak, potvrda registracije, potvrda administratora. U vezi je One-To-One sa entitetom slika

Korisnik		
id	INT	jedinstveni identifikator korisnika
username	VARCHAR	izabrano korisničko ime
email	VARCHAR	e-mail adresa korisnika
image-id	INT	identifikator slike korisnika
confirmation-hash	VARCHAR	individualni kod za registraciju
name	VARCHAR	ime korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Korisnik		
lastname	VARCHAR	prezime korisnika
user-type	VARCHAR	natjecatelj ili voditelj natjecanja
password	VARCHAR	šifra korisnika
confirmed	BOOLEAN	potvrđena registracija preko korisničkog emaila
confirmed-by-admin	BOOLEAN	potvrđena registracija od strane administratora

Natjecanje - entitet sadržava sve važne informacije o natjecanju: Id natjecanja, vrijeme početka, vrijeme završetka, broj zadataka, id voditelja natjecanja, identifikator pehara, ime natjecanja i je li virtualno. U odnosu je Many-To-One s entitetom korisnik preko atributa voditelja natjecanja, u vezi je One-To-One sa entitetom slika.

Natjecanje		
id	INT	jedinstveni identifikator natjecanja
image-id	INT	identifikator slike pehara
date-time-of-beginning	TIMESTAMP	vrijeme početka natjecanja
date-time-of-ending	TIMESTAMP	vrijeme završetka natjecanja
competition-maker-id	INT	id voditelja natjecanja
number-of-problems	INT	broj zadataka u natjecanju
name	VARCHAR	ime natjecanja
isVirtual	BOOLEAN	virtualno natjecanja

Zadatak - entitet sadržava sve važne informacije o zadatku. Sadrži atribute id, trajanje zadatka, booleanski atribut is-private, broj bodova koje je moguće ostvariti, tip problema (težinu zadatka), tekst zadatka, naslov i id korisnika koji je napravio

zadatak. S entitetom Korisnik je u odnosu Many-To-One preko atributa problem-maker-id.

Zadatak		
id	INT	jedinstveni identifikator zadatka
problem-maker-id	INT	identifikator vlasnika zadatka
duration	NUMERIC	trajanje zadatka
is-private	BOOLEAN	provjerava je li zadatak objavljen
problem-type	INT	težina zadatka
text	VARCHAR	tekst zadatka
title	NUMERIC	naslov zadatka
points	INT	broj bodova zadatka

Zadaci na natjecanju - entitet sadržava informacije o tome koje natjecanje ima koje zadatke. Atributi: id natjecanja i id zadatka. On predstavlja vezu Many-To-Many natjecanja i zadatka

Zadaci na natjecanju		
competition-id	INT	identifikator natjecanja (natjecanje.id)
problem-id	INT	identifikator zadatka (problem.id)

Primjeri za evaluaciju - entitet sadržava id zadatka te vrijednost i ključ, odnosno na ovaj način se mapiraju rješenja zadataka i služi za evaluaciju rješenja korisnika.

Primjeri za evaluaciju		
problem-id	INT	jedinstveni identifikator zadatka
vrijednost	VARCHAR	očekivani izlaz programa
ključ	VARCHAR	ulaz programa

Slike - entitet sadržava id slike te bajt zapis slike. Na ovaj način spremamo

slike korisnika i pehara.

Slike		
id	INT	jedinstveni identifikator slike
data	BYTEA	binarni kod slike

Tablica sudionika natjecanja - entitet sadržava id korisnika i id natjecatelja te informaciju do kada korisnik može rješavati natjecanje. Ova tablica predstavlja Many-To-Many vezu natjecanja i korisnika.

Tablica sudionika natjecanja		
competition_id	INT	identifikator natjecanja (natjecanje.id)
user_id	INT	identifikator korisnika
access_time	TIMESTAMP	rok završetka rješavanja natjecanja za korisnika

Rang lista - entitet sadržava id korisnika i id natjecatelja te korisnikov rang. Ova tablica predstavlja Many-To-Many vezu natjecanja i korisnika.

Rang lista		
user_id	INT	identifikator korisnika
placement	INT	rang korisnika
competition_id	INT	identifikator natjecanja (natjecanje.id)

Rezultati natjecanja - entitet sadržava jedinstven identifikator, id natjecatelja, korisničko ime natjecatelja, i osvojene bodove te korisnikov rang. Ova tablica predstavlja Many-To-Many vezu natjecanja i korisnika.

Rezultati natjecanja		
id	INT	jedinstven identifikator
points	NUMERIC	osvojeni bodovi
competition_id	INT	identifikator natjecanja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

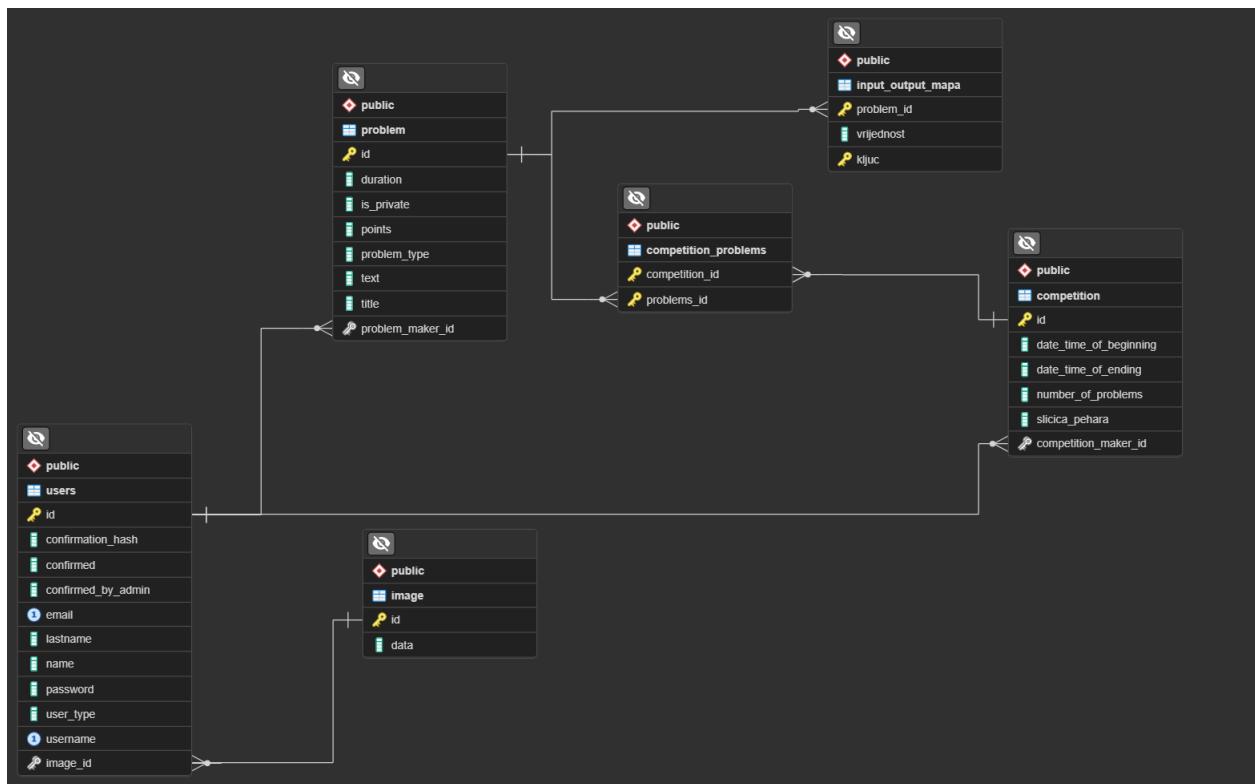
Rezultati natjecanja		
username	VARCHAR	korisničko ime

Učitana programska rješenja - entitet sadržava jedinstven identifikator, bitove pohranjenog rješenja, vrijeme od početka rješavanja do predaje, korisničko ime, identifikator zadatka, ostvarene bodove, identifikator natjecanja, postotak točnosti i radi li se o virtualnom učitanju. Ova tablica predstavlja Many-To-Many-Many vezu natjecanja, korisnika i zadatka.

Učitana programska rješenja		
id	INT	jedinstven identifikator
file_data	BYTEA	binarni kod rješenja
time_taken	INT	vrijeme od početka rješavanja do predaje
problem_id	INT	identifikator zadatka
username	VARCHAR	izabrano korisničko ime
points	NUMERIC	ostvareni bodovi
competition_id	INT	identifikator natjecanja (natjecanje.id)
percentage_of_total	DOUBLE	postotak točnosti
is_virtual	BOOLEAN	virtualno učitanje

4.1.2 Dijagram baze podataka

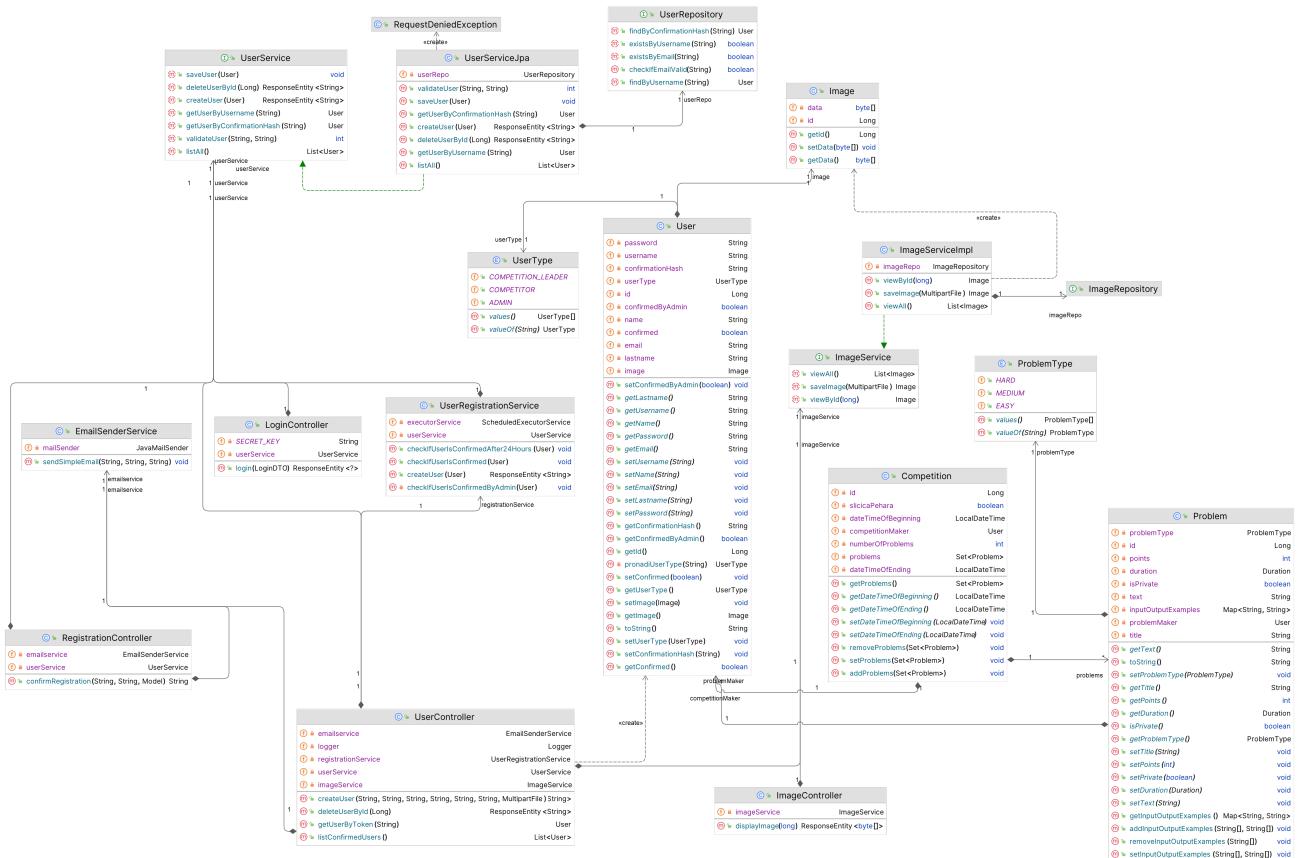
ER dijagram baze podataka prikazan na slici ?? sadrži vizualne slike ključeva: primarni ključ prikazan je ikonom ključa žute boje, strani ključ sivom bojom, a atributi označeni brojem 1 su alternativni ključevi. Prvi redak svake tablice entiteta predstavlja vidljivost tablice u bazi podataka, u drugom se retku nalazi ime tablice, a u preostalim retcima su ispisani atributi.



Slika 4.1: Er dijagram baze podataka

4.2 Dijagram razreda

U nastavku su prikazani razredi na kojima je zasnovan backend aplikacije. Lako je uočljivo da većina njih u nazivu sadrži naziv entiteta iz baze, te riječi Service i Controller.



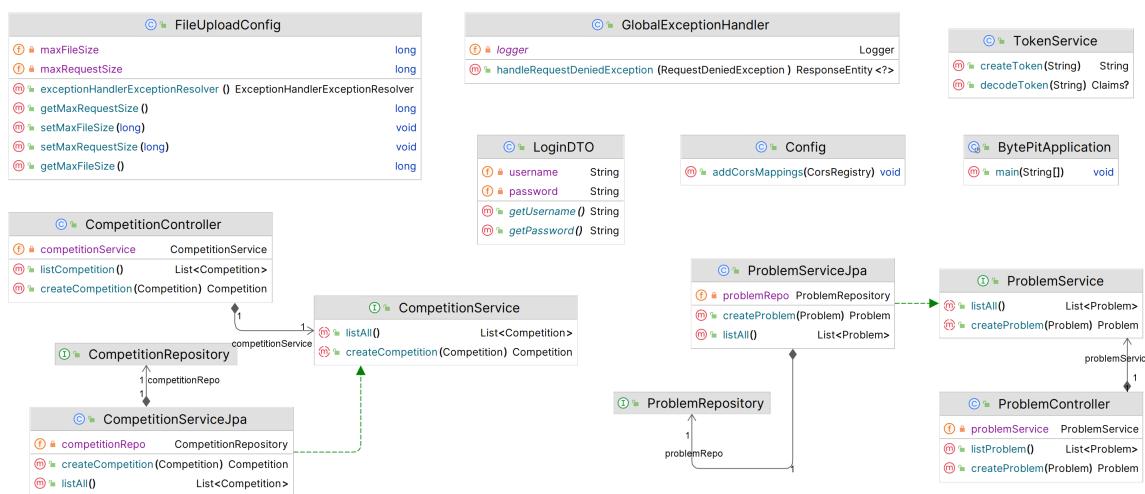
Slika 4.2: Dijagram razreda: razredi koji predstavljaju strukturu baze

Razredi User, Competition, Image i Problem sa slike ?? preslikavaju atribute svakog od entiteta iz baze. Razred User predstavlja registriranog korisnika, čiji UserType može biti COMPETITOR, COMPETITION LEADER i ADMIN (samo jedan, glavni korisnik), a koji pri registraciji unosi podatke prikazane na dijagramu (koji su vlastiti atributi razreda User). Razred Problem predstavlja zadatak koji je korisnik s tipom voditelj unio u sustav: na dijagramu se vidi ta poveznica. Jedna ili više instanci razreda Problem je dio Natjecanja tj. razreda Competition. Jedan korisnik može sudjelovati na više natjecanja, kao što i na jednom natjecanju sudjeluje jedan ili više korisnika. Tu je i razred Image koji služi za pohranu slike u bazu podataka.

Razredi imena Service služe za baratanje objekatima razreda User, Competition itd. Razredi imena Controller obrađuju HTTP zahtjeve poslane na server i vraćaju zatražene podatke u .json obliku.

Strelice sa slike na jednostavan i intuitivan način prikazuju poveznice između razreda User, UserController i UserService. Metode koje su u razredima implementirane služe za baratanje entitetima u bazi: primarno su to razredi koji u svome nazivu sadrže 'Service' kao što je prije rečeno. Razred UserController oslanja se na ta dva prethodno navedena razreda da bi mogao ispuniti zaprimljene HTTP zahtjeve, bilo da je to samo informacija o korisnicima ili izmjena podataka.

Slika ?? prikazuje još neke Controller i Service razrede koji povezuju bazu i server. Prikazan je TokenService razred koji omogućava autentifikaciju korisnika, a time sadrži i informacije o njegovim ovlastima u aplikaciji.



Slika 4.3: Dijagram razreda: ostali razredi Controller i Service te konfiguracija

dio 2. revizije

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

4.3 Dijagram stanja

UML-dijagram stanja je ponašajni UML-dijagram kojim se prikazuje diskretno ponašanje objekta ili sustava putem prelazaka između konačnog broja stanja, a često se za cijelokupni model stanja i prijelaza između stanja koristi izraz stroj stanja. Ova vrsta dijagrama se koristi za modeliranje ponašanja entiteta tijekom vremena, naglašavajući odgovor na dogadaje i okidače. Na dijagramu stanja, slika ?? prikazane su sve funkcionalnosti koje natjecatelj može.

Proces kreće prijavom korisnika kao natjecatelj. Na početnoj stranici natjecatelj može pristupiti aktivnom natjecanju, otici na kalendar natjecanja, korisnike, rezultate ili na vježbe. U bilo kojem slučaju korisnik može doći u stanje "Početna stranica", "Vježba", "Korisnici", "Kalendar natjecanja", "Rezultati" također iz svih stanja se može odjaviti.

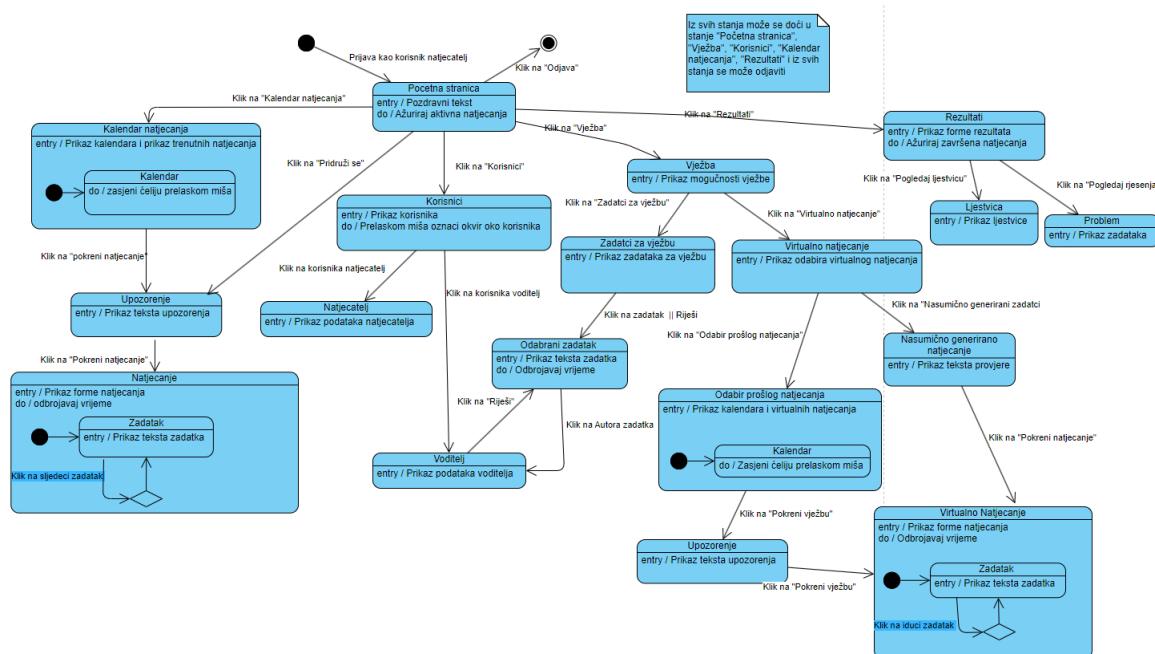
Odabirom kalendara stranica preusmjerava na kalendar s prikazanim natjecanjima, u koliko su aktivni. U slučaju aktivnog natjecanja, natjecatelj može pristupiti natjecanju odabirom na "Pokreni natjecanje". Tim odabirom javlja se poruka upozorenja koja naglašava kako se korisnik nebi smije koristiti nedopuštenim sredstvima. Odabirom "Pokreni natjecanje", natjecanje se aktivira te počinje brojač vremena. Korisnik može vidjeti tekst svakog zadatka, preći na sljedeći zadatak, testirati rješenje, učitati svoj kod te završiti natjecanje. Završetkom natjecanja natjecatelj vidi rang listu svih natjecatelja koji su pristupili tom natjecanju na stranici rezultati.

Odabirom korisnici, stranica prikazuje sve registrirane korisnike. Klikom na bilo kojeg korisnika prikazuju se podaci o korisnicima. Za natjecatelja prikazuju se podaci o broju točno riješenih zadataka, broju isprobanih zadataka i iscrtani su pehari za osvojena natjecanja. S druge strane profili voditelja sadrže popis učitanih zadataka s mogućnošću sortiranja i kalendar s popisom objavljenih natjecanja. Odabirom na zadatak od pojedinog voditelja, moguće je vidjeti i riješiti taj zadatak.

Odabirom vježba stranica nudi dva načina vježbe a to su virtualno natjecanje i zadatci za vježbu. Klikom na zadatci za vježbu prikazuju se svi zadatci objav-

ljeni od različitih voditelja. Klikom na zadatak ili na riješi započinje odbrojavanje vremena te natjecatelj može riješiti pojedini zadatak i testirati točnost svog koda. Odabirom virtualnog natjecanja stranica nudi: odabir pošlog natjecanja ili nasumično generirano natjecanje. U slučaju nasumično generiranog natjecanja stranica poručuje kako je vrijeme ograničeno te klikom na pokreni natjecanje, natjecatelj započinje natjecanje. Odabirom prošlog natjecanja, prikazuje se kalendar u kojem korisnik može odabrati prijašnje natjecanje. Postupak provedbe natjecanja za oba slučaja virtualnog natjecanja je isti kao i za normalno natjecanje.

Proces završava odjavom natjecatelja.



Slika 4.4: Dijagram stanja

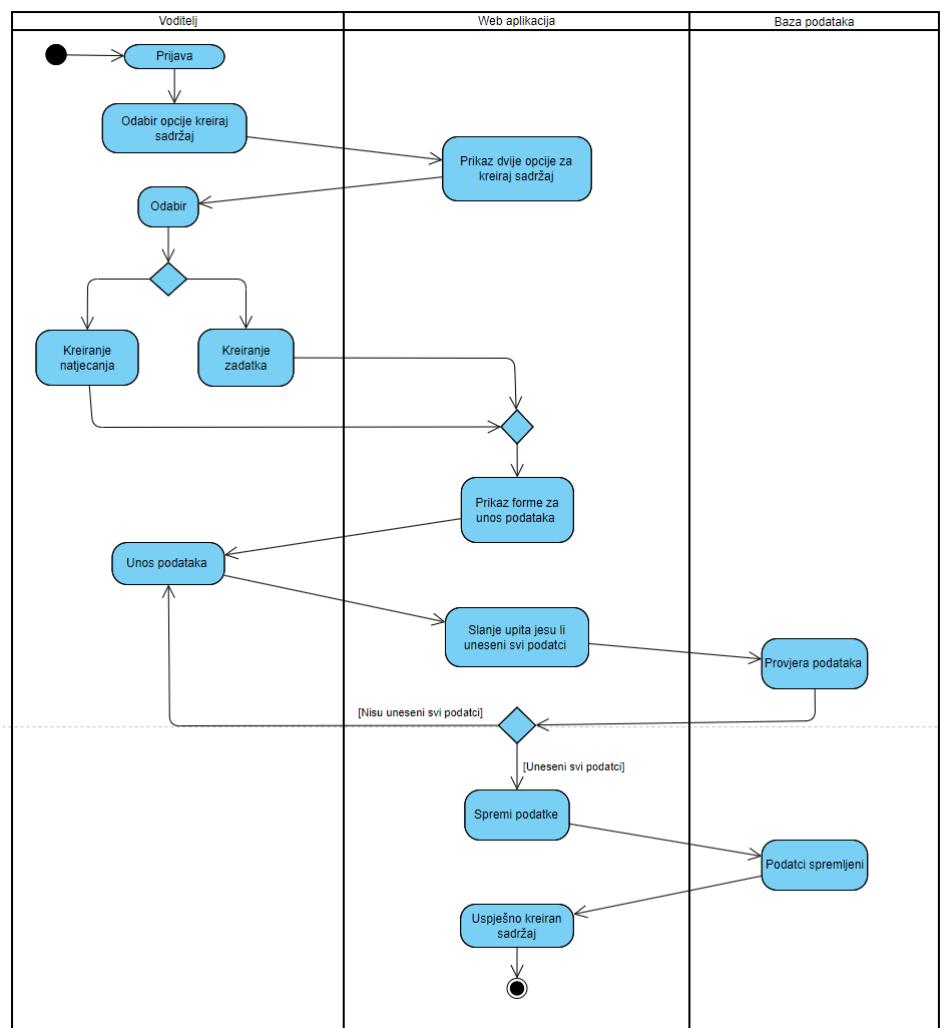
4.4 Dijagram aktivnosti

Dijagram aktivnosti je vrsta UML-dijagrama koja se u programskom inženjerstvu koristi za modeliranje i grafički prikaz dinamičkog ponašanja sustava. Na njemu se prikazuje izvođenje aktivnosti kroz niz akcija koje čine upravljačke tokove i tokove objekata, s naglaskom na slijed i uvjete toka. Na dijagrame aktivnosti, slika ?? prikazan je proces kako voditelj stvara zadatak za vježbu ili natjecanje.

Proces započinje prijavom korisnika u sustav kao voditelj. Na početnoj stranici,

klikom na "Kreiraj sadržaj", web aplikacija prikazuje 2 moguća odabira: kreiranje zadatka ili kreiranje natjecanja. Voditelj odabire što želi kreirati, te nakon odabira web aplikacija prikazuje formu za unos potrebnih podataka. Forma za stvaranje zadataka razlikuje se od forme za kreiranje natjecanja, ali bez obzira na odabranu, web aplikacija izvršava isti postupak.

Nakon što voditelj unese potrebne podatke, web aplikacija provjerava s bazom podataka jesu li uneseni svi potrebni podaci. U protivnom, web aplikacija vraća voditelja na ponovni unos podataka, gdje mora dopuniti izostavljene podatke. Kada su svi podaci uneseni, web aplikacija sprema kreirani sadržaj u bazu podataka te označava uspješno kreiran sadržaj.

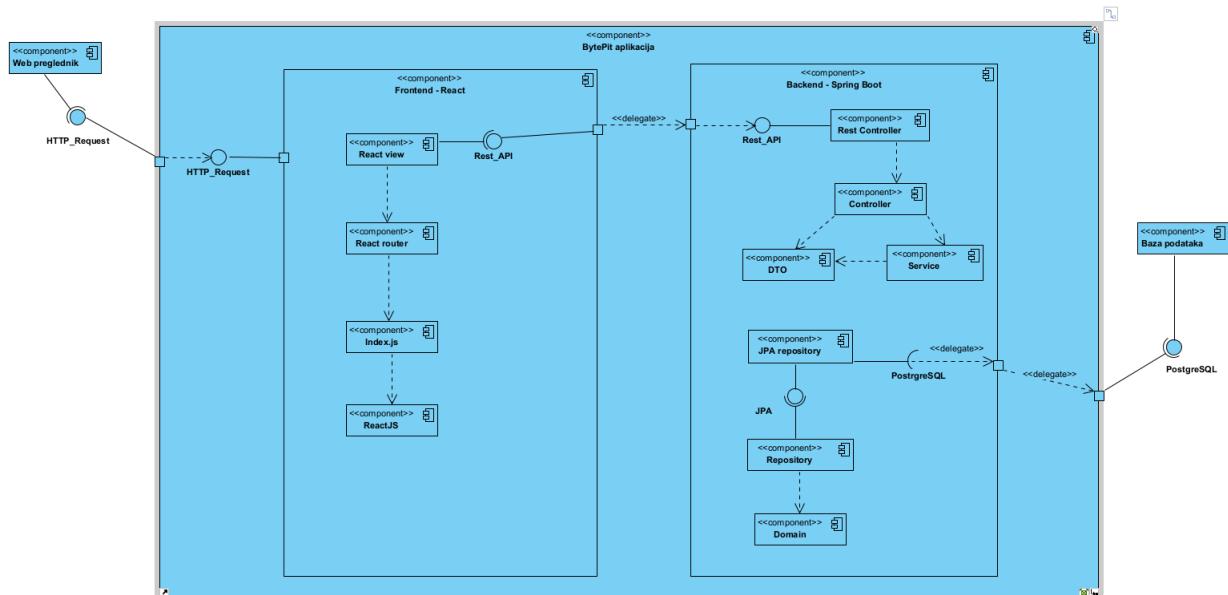


Slika 4.5: Dijagram aktivnosti

4.5 Dijagram komponenti

UML-dijagrami komponenti su vrsta strukturnih UML-dijagrama koji prikazuju organizaciju i odnose komponenti koje čine programsku potporu. Pružaju vizualni prikaz arhitekture sustava, naglašavajuci modularnu strukturu i interakcije između komponenti. Dijagram komponenti na slici ?? opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini.

U web aplikaciji BytePit postoje dvije velike komponente: Frontend - React i Backend - Spring Boot. Te dvije komponente kumuniciraju preko sučelja REST API koje je zaduženo za komunikaciju frontend dijela aplikacije s backend dijelom aplikacije. U React komponenti sučelje REST API spaja se na komponentu React view koja komunicira s ostalim Frontend komponentama. Sve frontend komponente ovise o komponenti ReactJS koja je zadužena za dohvatanje React biblioteka. S druge strane, u komponenti Spring Boot REST API je spojeno na komponentu Rest Controller koji je glavna komponenta za backend dio aplikacije. Sučelje JPA s vezom "ball and soccet" povezuje komponente Repository i JPA repository, JPA repository omogućuje komunikaciju s bazom podataka. Web aplikacija se spaja na vanjeske komponente: web preglednik preko sučelja HTTP Request koji podržava HTTP protokol i sve njegove operacije (Get, Post, Put, Delete), također na bazu podataka preko sučelja postgreSQL.



Slika 4.6: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

U izradi dokumentacije i aplikacije koristili smo sljedeće tehnologije i alate:

- **pgAdmin** je open-source alat za upravljanje bazama podataka PostgreSQL. Koristi se za sve osnovne operacije na bazi podataka, kao što su kreiranje, ažuriranje i brisanje tablica, upita i drugih objekata. Također omogućuje napredne operacije, kao što su izvoz i uvoz podataka, upravljanje sigurnosnim ovlastima i optimizacija performansi. Više o pgAdminu možete saznati na <https://www.pgadmin.org/>.
- **React** je JavaScript biblioteka za izradu web sučelja (frontend razvoj). Koristi se za izgradnju dinamičnih i interaktivnih web stranica i aplikacija koje reagiraju na korisničke interakcije. React je jednostavan za učenje i korištenje, a također je vrlo fleksibilan.. Više o Reactu možete saznati na <https://reactjs.org/>.
- **Spring Boot** je open-source framework za Java backend razvoj. Koristi se za izgradnju RESTful web servisa i mikroservisa. Također se koristi za automatizaciju mnogih zadataka koji su inače potrebni za izgradnju web servisa, kao što su konfiguracija, upravljanje ovisnostima i sigurnost. Spring Boot je lagan i brz, a također je vrlo fleksibilan i prilagodljiv. Više o Spring Bootu možete saznati na <https://spring.io/projects/spring-boot>.
- **GitHub** je web platforma za upravljanje izvornim kodom. Koristi se za spremanje, dijeljenje i suradnju na izvornom kodu. Siguran je i pouzdan, a također vrlo fleksibilan. Više o GitHubu možete saznati na <https://github.com/>.
- **WhatsApp** je mobilna aplikacija za razmjenu trenutnih poruka. Koristi se za komunikaciju unutar tima pomoći tekstualnih, glasovnih ili video poruka. WhatsApp je jednostavan za korištenje i pouzdan, a također je vrlo popularan. Više o WhatsAppu možete saznati na <https://www.whatsapp.com/>.
- **Visual Paradigm Online** je online alat za izradu UML dijagrama. Koristi se za modeliranje softverskih sustava i drugih sustava. Visual Paradigm Online

je jednostavan za korištenje i ima široku paletu funkcionalnosti. Više o Visual Paradigm Onlineu možete saznati na <https://online.visual-paradigm.com/>.

- **LateX** je programski jezik za pisanje strukturiranih tekstova i njihov automatski slog i prijelom u dokumente profesionalne kvalitete spremne za tisk. Omogućuje precizno kontroliranje izgleda dokumenta, uključujući veličinu i oblik slova, razmake, margine i druge elemente. LaTeX je popularan među znanstvenicima i inženjerima za pisanje tehničkih dokumenata. Više o LaTeXu možete saznati na <https://www.latex-project.org/>.

Zaključak

U izradi dokumentacije i aplikacije korištene su suvremene tehnologije i alati koji su omogućili izgradnju kvalitetnog i funkcionalnog proizvoda.

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 odabralih 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 kod svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnem 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

¹ <https://www.seleniumhq.org/>

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:

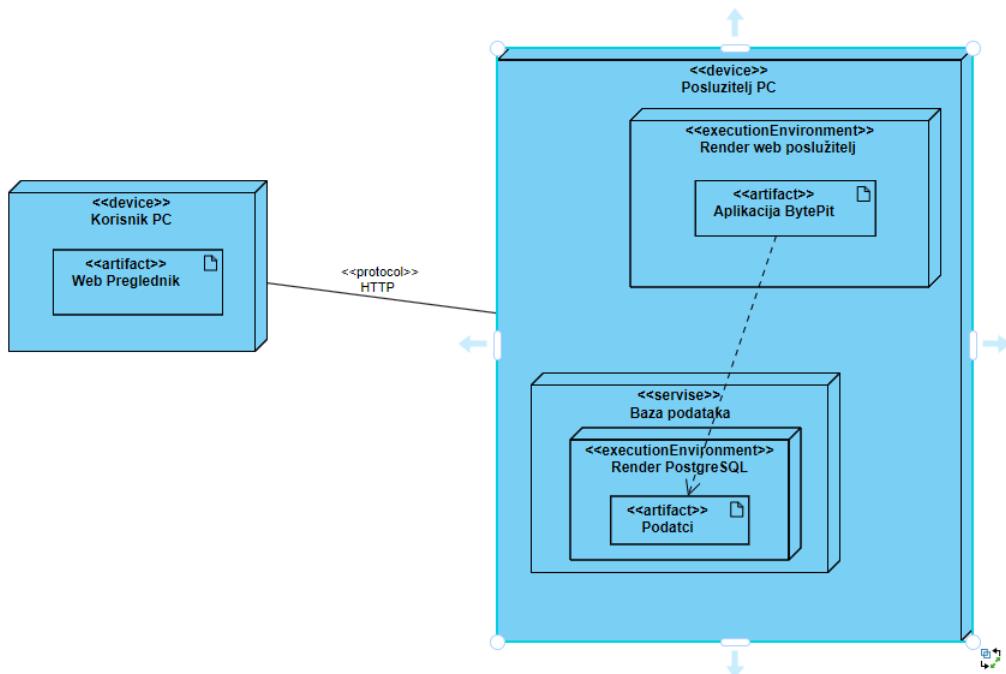
- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsко sučelje.

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

5.3 Dijagram razmještaja

UML-dijagrami razmještaja su vrsta strukturnih UML- dijagrama koji prikazuju fizicku arhitekturu i konfiguraciju razmještaja programskog sustava.

U ovoj aplikaciji korisnikovo računalo preko web preglednika pristupa poslužiteljovom računalu pomoću HTTP protokola. Na poslužiteljskom računalu se nalazi naša aplikacija BytePit i baza podataka s kojom aplikacija komunicira. Sustav je baziiran na arhitekturi "klijent – posluzitelj"



Slika 5.1: Dijagram razmjestaja

5.4 Upute za puštanje u pogon

Postavljanje aplikacije u pogon putem usluge Render zahtijeva pažljivo provedene korake uključujući kreiranje baze podataka, kreiranje backend-a te kreiranje frontenda, čime se osigurava siguran i učinkovit deploy.

Prije samog puštanja aplikacije u pogon potrebno je provesti tri ključna koraka:

- Konfiguracija baze podataka**

- Ovaj proces uključuje definiranje environmental varijabli unutar konfiguracije razvojnog okruženja (IDE). U našem slučaju, konfiguracija se nalazi u datoteci src/main/resources/application.properties. U ovoj datoteci specificiramo parametre kao što su korisničko ime, lozinka, URL baze podataka te određujemo port poslužitelja. Primjer konfiguracije:

```
# Port na kojem će se vrtiti api
# Obavezno izloziti, ovu varijablu koristi Render
server.port=${PORT:8080}

# Korijenska putanja ("prefiks") za sve zahtjeve na backend - preporuča se postaviti ovo zbog proxy konfiguracije
# Ako je npr. u controlleru navedena putanja /test, moci će joj se pristupiti pomocu putanje /api/test
server.servlet.context-path=/api

# Koristi se samo kao primjer koristenja environment varijable unutar TestController klase
# SERVER_MESSAGE je sada environment varijabla koja će se mapirati na property "message", ako nije postavljena uzima se default vrijednost "Hello from backend! "
message=${SERVER_MESSAGE:Hello from backend! }

# Lokacija Liquibase master chagelog-a
spring.liquibase.change-log=classpath:/db/changelog/changelog-master.xml

# Konfiguracija baze podataka
# Izlaganje environment varijabli je nuzno da bismo mogli postaviti adresu, korisnicko ime i lozinku baze podataka na produkciji
# Stavljanje credentialsa proizvodnje baze podataka direktno u kod je jako losa praksa!
spring.datasource.password=${DB_PASS:password}
spring.datasource.username=${DB_USERNAME:username}
spring.datasource.url=${DB_URL:jdbc:postgresql://localhost:5432/db}
spring.datasource.driverClassName=${DB_DRIVER:org.postgresql.Driver}
```

Slika 5.2: Environment Varijable u IDE-u

2. Priprema backend-a za deploy

- Nakon postavljanja baze podataka, slijedi priprema backend-a za deploy. Ovaj korak uključuje dodavanje Dockerfile skripte za izgradnju i pokretanje aplikacije. U našem slučaju, koristimo Maven, pa je Dockerfile konfiguriran na sljedeći način:

```
# Container za izgradnju (build) aplikacije
FROM openjdk:17-alpine AS builder

# Kopiranje izvornog koda u container
COPY ../../.mvn .mvn
COPY ../../mvnw .
COPY ../../pom.xml .
COPY ../../src src
RUN chmod +x mvnw

# Pokretanje builda
```

```
RUN ./mvnw clean package

FROM openjdk:17-alpine

COPY --from=builder target/*.jar /app.jar

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "/app.jar"]
```

- Ukoliko se mijenja lokacija Dockerfilea paziti na putanje unutar COPY naredbi u Dockerfile skripti. U nasem slucaju Docker file se nalazi u root direktoriju Docker/Maven/Dockerfile te su po toj putanji pisane naredbe COPY.

3. Priprema frontenda za deploy

- Potrebno je dodati potrebne dependencije u package.json datoteku. U terminalu je potrebno izvršiti sljedeće naredbe:

```
npm install http-proxy-middleware
npm install dotenv
npm install express
```

- Potrebno je kreirati Proxy.js, u src direktorij, koji služi kao proxy server za lokalni development (preusmjeruje api pozive na localhost:8080). Ovo je primjer koda:

```
const { createProxyMiddleware } =
  require("http-proxy-middleware");

module.exports = function (app) {
  app.use(
    "/api",
    createProxyMiddleware({
      target: "http://localhost:8080/",
      changeOrigin: true,
    })
);
```

```
};

---


```

- Stvoriti app.js, u root direktorij, koja sadrži Express server za produkcijski proxy i posluživanje frontenda. Ovo je primjer koda:

```
const express = require("express");
const { createProxyMiddleware } =
    require("http-proxy-middleware");
require("dotenv").config();
const path = require("path");

const app = express();

// Configuration
const { PORT, HOST, API_BASE_URL } = process.env;

// Proxy
app.use(
    "/api",
    createProxyMiddleware({
        target: API_BASE_URL,
        changeOrigin: true,
    })
);

app.use(express.static(path.join(__dirname,
    'build')));

app.listen(PORT, HOST, () => {
    console.log(`Starting Proxy at
${HOST}:${PORT}`);
});

app.get("*", async (req, res) => {
    res.sendFile(path.join(__dirname, 'build',
        'index.html'))
});

---


```

- Izmjeniti package.json skripte i dodati specifične konfiguracije:

```
"build": "yarn install && react-scripts build",
"start-prod": "node app.js",
"engines": {
  "node": ">=18.18.0 <19.0.0"
}
```

Sada, nakon završene pripreme, možemo započeti deploy.

1. Kreiranje baze podataka U rander dashboardu:

- New →PostgreSQL
- Postaviti ime baze i opcionalno username za korisnika baze (password je automatski generiran)
- Region Frankfurt
- Create Database

2. Kreiranje backend-a U Render dashboardu:

- New →Web Service
- Povezati GitHub račun, nakon čega su za odabir dostupni svi projekti na koje imate prava pristupa
- Stisnuti connect pored odgovarajućeg projekta
- Postaviti ime za servis (postat će dio web adrese)
- Postaviti root directory (u nasem slučaju: dev)
- Environment Docker
- Region Frankfurt
- Na dnu proširiti *advanced*
- Dodati potrebne environment varijable (vidi sliku ??), kopirati vrijednosti iz postavki baze na renderu (vidi sliku ??). Također dodati kopirane environment varijable u IDE okruzenje (vidi sliku ??)

Environment Variables Optional
Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more](#).

DB_PASS	value	Generate	Delete
DB_USERNAME	value	Generate	Delete
DB_URL	value	Generate	Delete
+ Add Environment Variable			

Slika 5.3: Environment Varijable za backend

Connections

Hostname	dpg-clav11mnt67s73f76bag-a
Port	5432
Database	koderkolege
Username	koderkolege_user
Password
Internal Database URL
External Database URL
PSQL Command

Slika 5.4: Postavke baze podataka na renderu

- Postaviti putanju za Dockerfile ovisno koji se package manager koristi (u ovom slučaju ./docker/maven/Dockerfile, vidi sliku ??)

Dockerfile Path
Path to your Dockerfile relative to the repository root. This is *not* relative to your Docker build context. For example, [./subdir/Dockerfile](#).

BytePit/	./docker/maven/Dockerfile
Edit	

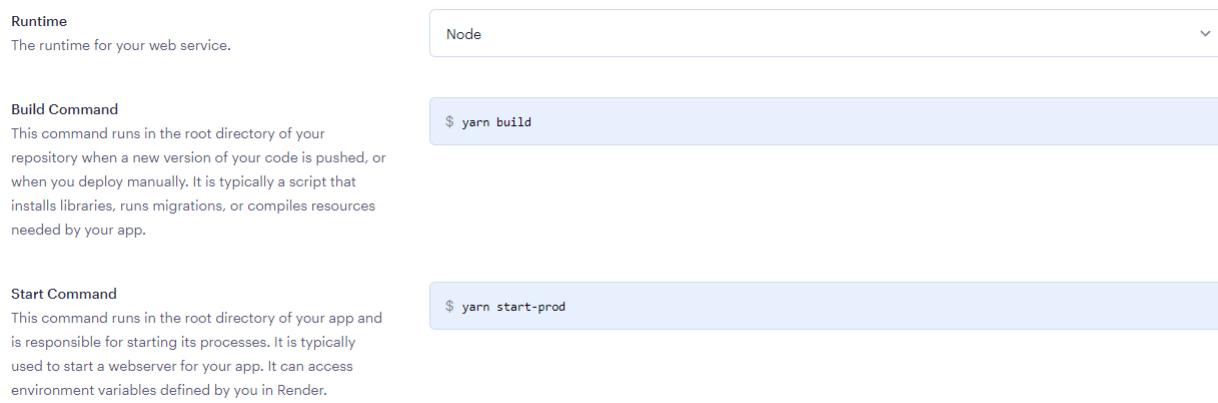
Slika 5.5: Putanja za Dockerfile

- Stisnuti Create Web Service

3. Kreiranje frontenda

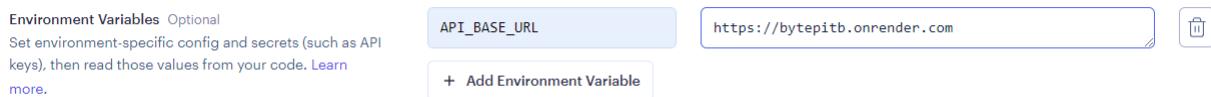
- New → Web Service

- Povezati GitLab racun, nakon cega su za odabir dostupni svi projekti na koje imate prava pristupa
- Stisnuti connect pored odgovarajućeg projekta
- Postaviti ime za servis (postat će dio web adrese)
- Postaviti root directory (u nasem slučaju: dev)
- Environment Node
- Region Frankfurt
- Build Command postaviti na yarn build, a Start Command yarn start-prod (vidi sliku ??)



Slika 5.6: Build Comand za Node enviroment

- Na dnu proširiti *advanced*
- Dodati potrebne environment varijable - API BASE URL postaviti na adresu deployanog backenda aplikacije dostupnu na Render dashboardu (vidi sliku ??)



Slika 5.7: Environment Varijable za frontend

- Stisnuti Create Web Service

Nakon uspješnog deploja, naša aplikacija sada bezbrižno operira na Render poslužitelju.

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

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20. listopada 2023.
- Prisustvovali: Petra Buršić, Filip Mohler, Matea Cvetković, Dora Bilić-Pavlinović, Petra Kelković, Mislav Korotaj, Nives Ostojić
- Teme sastanka:
 - Prvotno okupljanje tima
 - Familiarizacija s temom projekta

2. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: Petra Buršić, Filip Mohler, Matea Cvetković, Dora Bilić-Pavlinović, Petra Kelković, Mislav Korotaj, Nives Ostojić
- Teme sastanka:
 - Podjela uloga među članovima tima

3. sastanak

- Datum: 14. studenoga 2023.
- Prisustvovali: Petra Buršić, Filip Mohler, Matea Cvetković, Dora Bilić-Pavlinović, Petra Kelković, Mislav Korotaj, Nives Ostojić
- Teme sastanka:
 - Pregled dokumentacije, podjela ostatka posla

4. sastanak

- Datum: 16. studenoga 2023.
- Prisustvovali: Petra Buršić, Filip Mohler, Matea Cvetković, Dora Bilić-Pavlinović, Petra Kelković, Mislav Korotaj, Nives Ostojić
- Teme sastanka:
 - Deploy aplikacije

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Petra Kelković	Petra Buršić	Nives Ostojić	Matea Cvetković	Dora Bilić-Pavlinović	Mislav Korotaj	Filip Mohler
Upravljanje projektom	3	3	3	5	6	3	3
Opis projektnog zadatka				5	6		
Funkcionalni zahtjevi	3					3	1
Opis pojedinih obrazaca						3	6
Dijagram obrazaca							2
Sekvencijski dijagrami		6					2
Opis ostalih zahtjeva							2
Arhitektura i dizajn sustava			5		2	2	1
Baza podataka			7			3	1
Dijagram razreda				2			3
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati						6	
Ispitivanje programskog rješenja	2		4		5	5	
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Petra Kelković	Petra Buršić	Nives Ostojić	Matea Cvetković	Dora Bilić-Pavlincović	Mislav Korotaj	Filip Mohler
Upute za puštanje u pogon	10	10	10	10	10	10	10
Dnevnik sastajanja					1		
Zaključak i budući rad							
Popis literature							
<i>upoznavanje s korištenim tehnologijama</i>	7	6	6	7	6	6	6
<i>izrada osnovnih prikaza</i>	25	6		3			
<i>izrada baze podataka</i>					3		
<i>spajanje s bazom podataka</i>			5		4		
<i>spajanje fronta i backenda</i>		14	10	14		3	
<i>back end</i>			25		30	20	

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, pritisom na stavku Contributors.