

Programsko inženjerstvo

Ak. god. 2023./2024.

BytePit

Dokumentacija, Rev. 1

Grupa: *Koder kolege*

Voditelj: *Petra Kelković*

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

Nastavnik: *Hrvoje Nuić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
2.1 Primjeri u L ^A T _E Xu	11
3 Specifikacija programske potpore	14
3.1 Funkcionalni zahtjevi	14
3.1.1 Obrasci uporabe	17
3.1.2 Sekvencijski dijagrami	19
3.2 Ostali zahtjevi	20
4 Arhitektura i dizajn sustava	21
4.1 Baza podataka	21
4.1.1 Opis tablica	21
4.1.2 Dijagram baze podataka	22
4.2 Dijagram razreda	23
4.3 Dijagram stanja	24
4.4 Dijagram aktivnosti	25
4.5 Dijagram komponenti	26
5 Implementacija i korisničko sučelje	27
5.1 Korištene tehnologije i alati	27
5.2 Ispitivanje programskog rješenja	28
5.2.1 Ispitivanje komponenti	28
5.2.2 Ispitivanje sustava	28
5.3 Dijagram razmještaja	29
5.4 Upute za puštanje u pogon	30
6 Zaključak i budući rad	31
Popis literature	32

Indeks slika i dijagrama	33
Dodatak: Prikaz aktivnosti grupe	34

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Personalizirana naslovna stranica te header i footer.	Dora Bilić-Pavlinović	27.10.2023.
0.2	Opis projektnog zadatka	Dora Bilić-Pavlinović, Matea Cvetković	24.08.2013.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zah-tjevi i dodatak A	*	25.08.2013.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	*	26.08.2013.
0.8	Povijest rada i trenutni status implementa-cije, Zaključci i plan daljnog rada	*	28.08.2013.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatak	Autori	Datum
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	*	14.09.2013.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predloška dokumentacije	*	28.09.2013.

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 (dodatak, 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

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 slikica 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 zadatka 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 and njim. Tada će iz korisnikove perspective 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.

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 dodajući 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 2.1).

The screenshot shows the BytePit application interface. At the top, there is a navigation bar with icons for 'My previous exams', 'My tickets', 'My stats', 'Playground', 'Code theme', and a dropdown for '2022/2023'. On the right side of the top bar, there is a user profile placeholder, the text 'Ime Prezime Student', and a 'Logout' button. Below the top bar, there is a search bar with the placeholder '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 lists five exams with columns for '#', 'Academic year', 'Exam title, questions no', 'Runs/Max', 'Duration', 'Score ignored', 'Forward only', 'Used in stats', 'Global', 'Public', 'Available for', and 'Start exam'. Each row includes a 'Start' button.

#	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: Međuispit 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 2.2) (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 organizaciju 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 2.3). 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
Incorrect

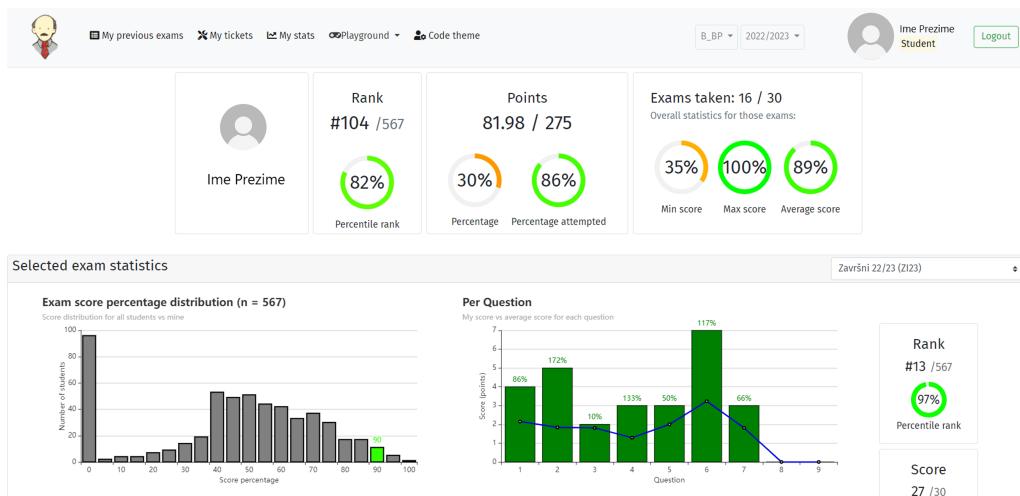
Zadatak 3. (4 bodova)
Administrator baze podataka Streamflix želi stvoriti novu ulogu MOVIE_CLEANER koja će obavljati sljedeće poslove: - pregled svih stupaca iz tablice TRACK - ažuriranje svih neključnih atributa iz tablice TRACK osim tipa sadržaja - brisanje redaka iz tablice TRACK
pri čemu su navedeni 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 uzima 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 kojima su učitali korisnici (slika 2.4)... 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 zadatake za vježbu (slika 2.5) (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 zadataka: u potonjem se nude samo replike stvarnih natjecanja koje se mogu pokrenuti.

Contest status								<input type="checkbox"/> my only	<input type="checkbox"/> friends only
#	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	rajvirsingh192002	1890B - Qingshan Loves Strings	Python 3	Accepted	170 ms	41700 KB		
230279325	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.

2.1 Primjeri u L^AT_EXu

Ovo potpoglavlje izbrisati.

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti L^AT_EXA koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u L^AT_EXu - https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf
- L^AT_EX projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

podcrtani tekst, podebljani tekst, nagnuti tekst

primjer primjer primjer primjer primjer primjer

- primjer
- primjer
- primjer
 - 1. primjer
 - 1.a primjer
 - b primjer
 - 2. primjer

primjer url-a: <https://www.fer.unizg.hr/predmet/proinz/projekt>

posebni znakovi: # \$ % & { } _ | < > ^ ~ \

naslov unutar tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	

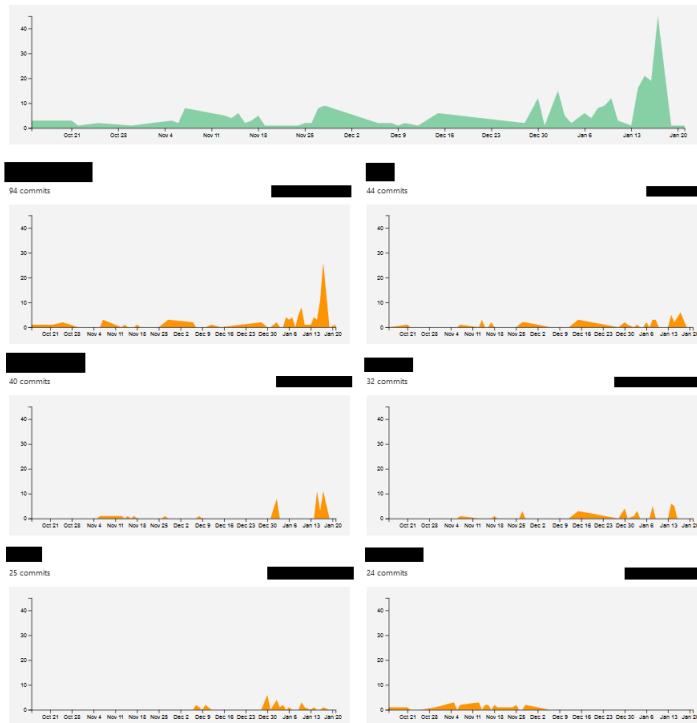
Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

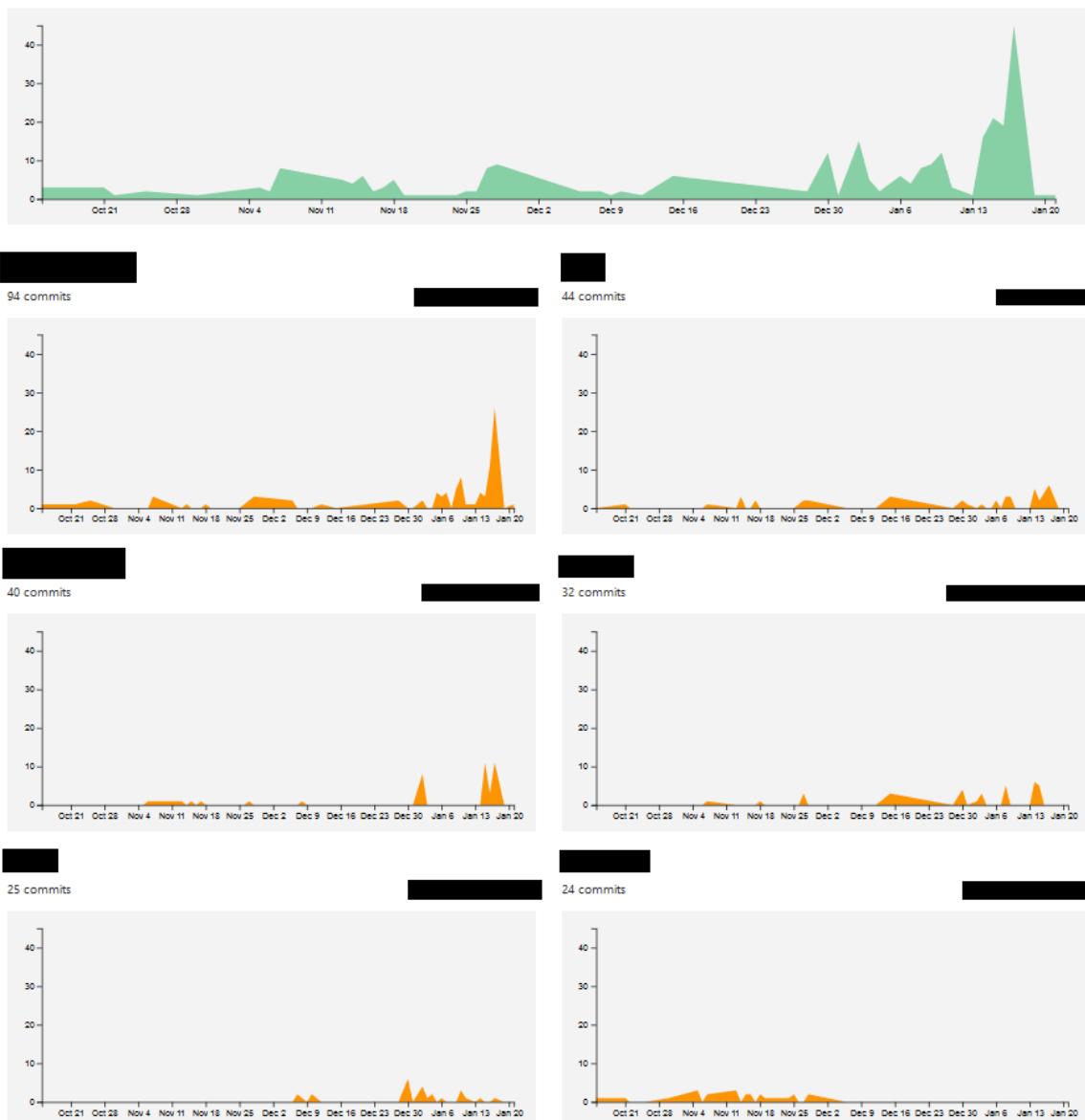
naslov unutar tablice		
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	



Slika 2.6: Primjer slike s potpisom



Slika 2.7: Primjer slike s potpisom 2

Referenciranje slike 2.7 u tekstu.

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. 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)
 - i. na profilu natjecatelja može vidjeti statistike o broju točno riješenih zadataka, broju isprobanih zadataka i pehare za osvojena natjecanja
 - ii. na profilu voditelja može vidjeti popis učitanih zadataka i kalendar s popisom objavljenih 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 (???sta ovo znaci u tekstu???)

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

3. Voditelj natjecanja (inicijator) može:

(a) učitati nove zadatke u aplikaciju

(b) organizirati natjecanje:

i. odabire vrijeme početka i završetka

ii. odlučuje broj zadataka

iii. odlučuje koji su zadaci aktivni

iv. po želji učitava sličicu pehara (??)

(c) izraditi zadatak:

i. unosi naziv zadatka

ii. unosi broj bodova (ovisan o težini zadatka)

iii. određuje vremensko ograničenje izvršavanja zadatka

iv. unosi tekst zadatka

v. unosi testne primjere za evaluaciju (provjeravaju ulaz i izlaz programa)

vi. može zadatak postaviti kao privatni te on nakon završetka natjecanja automatski postaje javan

- (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) može:

- (a) funkcionalnost 1
- (b) funkcionalnost 2

3.1.1 Obrasci uporabe

dio 1. revizije

Opis obrazaca uporabe

Funkcionalne zahtjeve razraditi u obliku obrazaca uporabe. Svaki obrazac je potrebno razraditi prema donjem predlošku. Ukoliko u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijek obrasca vratio na osnovni tijek.

UC<broj obrasca> -<ime obrasca>

- **Glavni sudionik:** <sudionik>
- **Cilj:** <cilj>
- **Sudionici:** <sudionici>
- **Preduvjet:** <preduvjet>
- **Opis osnovnog tijeka:**
 1. <opis korak jedan>
 2. <opis korak dva>
 3. <opis korak tri>
 4. <opis korak četiri>
 5. <opis korak pet>
- **Opis mogućih odstupanja:**
 - 2.a <opis mogućeg scenarija odstupanja u koraku 2>
 1. <opis rješenja mogućeg scenarija korak 1>
 2. <opis rješenja mogućeg scenarija korak 2>
 - 2.b <opis mogućeg scenarija odstupanja u koraku 2>
 - 3.a <opis mogućeg scenarija odstupanja u koraku 3>

1. -Registriraj novog korisnika

- **Glavni sudionik:** neregistrirani korisnik
- **Cilj:** Registracija novog korisnika
- **Sudionici:** back-end
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju

2. Otvara se form u koji upisuje podatke:
 - (a) korisničko ime
 - (b) fotografija
 - (c) lozinka
 - (d) ime i prezime
 - (e) email adresa
 3. Back end provjerava jesu li korisničko ime i email već zauzeti
 4. Ako nisu zauzeti, korisnik se upisuje u bazu sa "zaključanim" profilom
 5. Učitava se stranica na kojoj piše da korisnik treba potvrditi mail
 6. Back end šalje mail potvrde na mail korisnika
 7. Korisnik potvrđuje registraciju
 8. Back end "otključava" profil korisnika
 9. Korisniku se šalje mail o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 3.a Email/korisničko ime su već zauzeti
 1. Korisnik dobiva poruku da je mail/korisničko ime već korišten
 2. Traži se ponovni upis podataka
 - 7.a Korisnik ne potvrđuje email u nekom zadаном roku
 1. Nakon isteka roka, korisnik se briše iz baze podataka

Dijagrami obrazaca uporabe

Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.

3.1.2 Sekvencijski dijagrami

dio 1. revizije

Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.

3.2 Ostali zahtjevi

dio 1. revizije

*Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju **kako se sustav treba ponašati** i koja **ograničenja** treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

4. Arhitektura i dizajn sustava

dio 1. revizije

Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programske zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:

- izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu)
- organizaciju sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje)
- organizaciju aplikacije (npr. slojevi frontend i backend, MVC arhitektura)

4.1 Baza podataka

dio 1. revizije

Potrebno je opisati koju vrstu i implementaciju baze podataka ste odabrali, glavne komponente od kojih se sastoji i slično.

4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime variable 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

korisnik - ime tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

korisnik - ime tablice		
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

4.1.2 Dijagram baze podataka

U ovom potpoglavlju potrebno je umetnuti dijagram baze podataka. Primarni i strani ključevi moraju biti označeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".

4.2 Dijagram razreda

Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.

dio 1. revizije

Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz **generičku funkcionalnost** sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.

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 tijek 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 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 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.

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

2.1 Edgar: početna stranica i popis vježbi za ispite	7
2.2 Edgar: probni ispit	8
2.3 Edgar: stranica sa statistikom	8
2.4 Codeforces: rješenja različitih korisnika	9
2.5 Codeforces: mogućnost izvršavanja koda	10
2.6 Primjer slike s potpisom	12
2.7 Primjer slike s potpisom 2	13

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: u ovom formatu: 31. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 31. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

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							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
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							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

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.