

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	9
3 Specifikacija programske potpore	12
3.1 Funkcionalni zahtjevi	12
3.1.1 Obrasci uporabe	13
3.1.2 Sekvencijski dijagrami	14
3.2 Ostali zahtjevi	15
4 Arhitektura i dizajn sustava	16
4.1 Baza podataka	16
4.1.1 Opis tablica	16
4.1.2 Dijagram baze podataka	17
4.2 Dijagram razreda	18
4.3 Dijagram stanja	19
4.4 Dijagram aktivnosti	20
4.5 Dijagram komponenti	21
5 Implementacija i korisničko sučelje	22
5.1 Korištene tehnologije i alati	22
5.2 Ispitivanje programskog rješenja	23
5.2.1 Ispitivanje komponenti	23
5.2.2 Ispitivanje sustava	23
5.3 Dijagram razmještaja	24
5.4 Upute za puštanje u pogon	25
6 Zaključak i budući rad	26
Popis literature	27

Indeks slika i dijagrama	28
Dodatak: Prikaz aktivnosti grupe	29

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, zadaci 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 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, u Edgaru to čini administrator dodajući korisnike i nakon toga šaljući im podatke. Zbog same razlike u namjeni, predana rješenja se drugačije boduju (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. Ostale funkcionalnosti BytePita vrlo su slične Edgaru: uloge natjecatelja i studenta su slične, oni mogu učitavati i provjeravati svoj kod, pokrenuti Probni ispit (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).

Slične str,korisnost,upotreba str.

The screenshot shows the BytePit platform interface. At the top, there is a navigation bar with links for 'My previous exams', 'My tickets', 'My stats', 'Playground', 'Code theme', and user information ('B_BP', '2022/2023', 'Logout'). Below the navigation bar, there is a search bar labeled 'Password...' and a 'Start exam' button. A message 'You have unsubmitted private exams:' is displayed, followed by a table showing one private exam entry:

#	Title	Started at	Available until	Can continue?
1	3. laboratorijska vježba	2023-05-25 19:03:52	2023-06-02 11:00:00	Can't continue: expired.

Below this, there is a section titled 'Public exams:' showing a list of public 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'. The table contains 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	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 years expires: 2023-09-27 12:00:00	<button>Start</button>
2	2022/2023	Vježba: Meduispit 22/23 8 question(s)	0/0	2 hours	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 years expires: 2023-09-27 12:00:00	<button>Start</button>
3	2019/2020	Vježba: Završni ispit 19/20 8 question(s)	0/0	3 hours	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 years expires: 2023-09-27 12:00:00	<button>Start</button>
4	2020/2021	Vježba: Dekanski rok 20/21 10 question(s)	0/0	2 hours	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 years expires: 2023-09-27 12:00:00	<button>Start</button>
5	2020/2021	Vježba: Jesenski rok 20/21	0/0	3 hours	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 years	<button>Start</button>

Slika 2.1: Početna stranica i popis vježbi za ispite

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 0 pts 0 pts

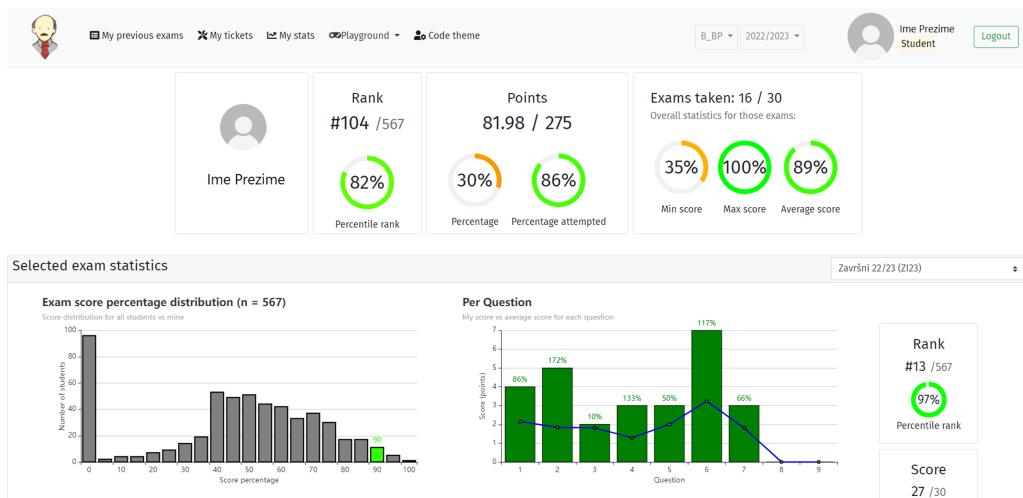
Zadatak 3. (4 boda)
Administrator baze podataka Streamflix želi stvoriti novu ulogu MOVIE_CLEANER koja će obavljati sljedeće poslove: - pregled svih stupaca iz tablice TRACK - ažuriranje svih neključnih atributa iz tablice TRACK osim tipa sadržaja - brisanje redaka iz tablice TRACK
pri čemu se navedeni 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).
Napomrite niz 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 referiraju 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: Probni ispit



Slika 2.3: Stranica sa statistikom

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	

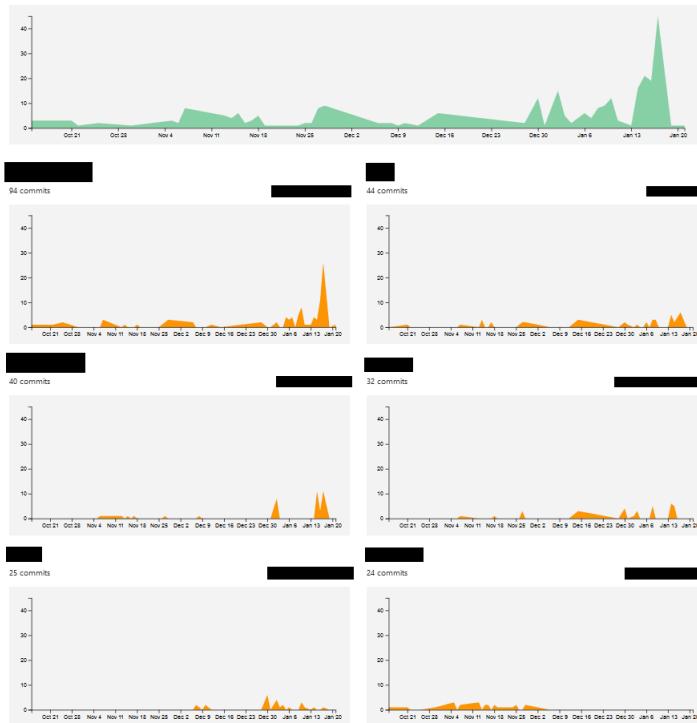
Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

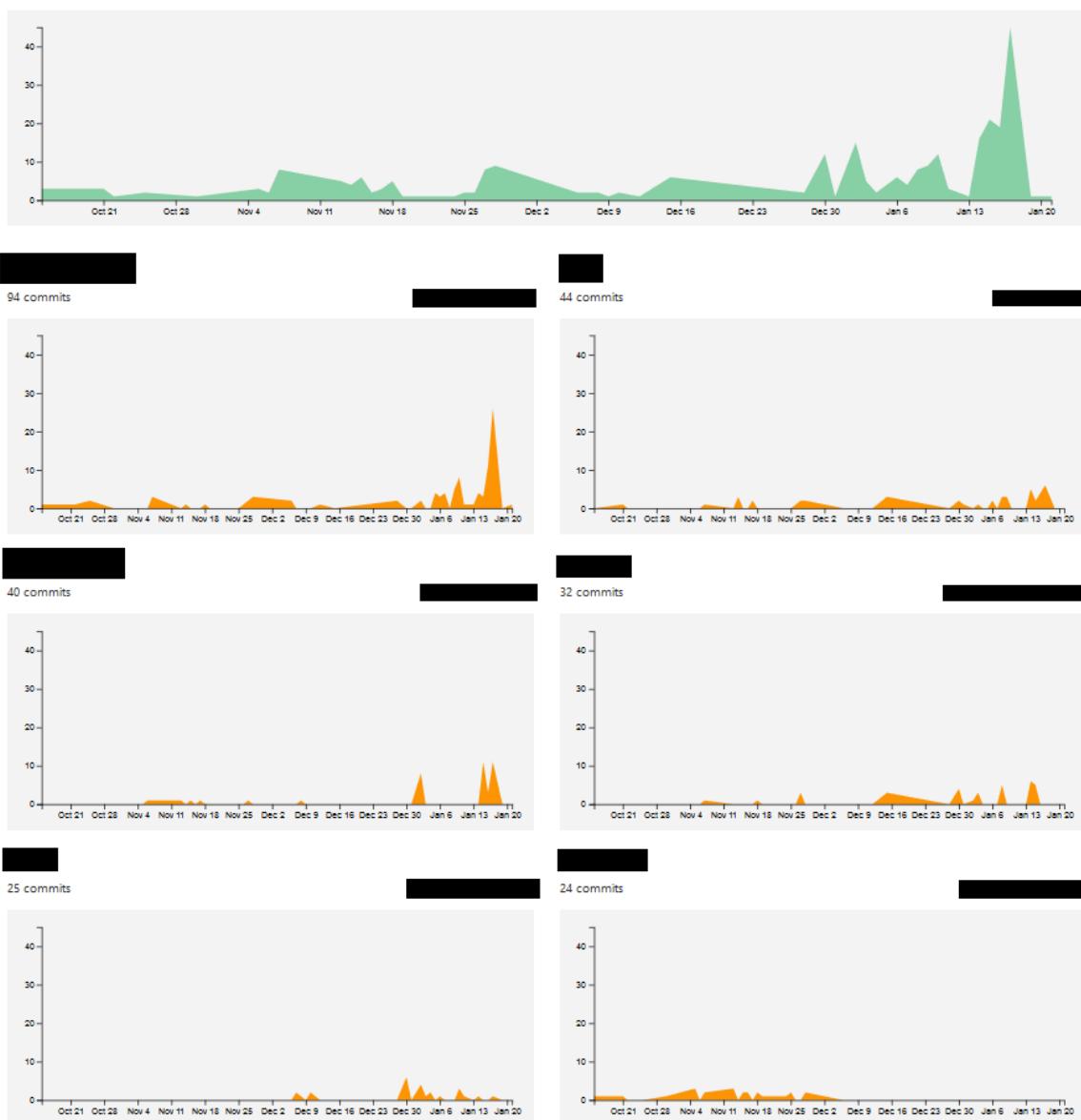
naslov unutar tablice		
ime	VARCHAR	
primjer	VARCHAR	

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 2.4: Primjer slike s potpisom



Slika 2.5: Primjer slike s potpisom 2

Referenciranje slike 2.5 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. Dionik 1
2. Dionik 2
3. ...

Aktori i njihovi funkcionalni zahtjevi:

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

3.1.1 Obrasci uporabe

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>

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	Početna stranica i popis vježbi za ispite	7
2.2	Probni ispit	8
2.3	Stranica sa statistikom	8
2.4	Primjer slike s potpisom	10
2.5	Primjer slike s potpisom 2	11

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: 29. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 29. 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.