

Programsko inženjerstvo

Ak. god. 2023./2024.

BytePit

Dokumentacija, Rev. <1 ili 2>

Grupa: *Looney Codes*

Voditelj: *Vedran Ćutić*

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

Nastavnik: *Hrvoje Nuić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	13
3.2 Ostali zahtjevi	14
4 Arhitektura i dizajn sustava	15
4.1 Baza podataka	15
4.1.1 Opis tablica	15
4.1.2 Dijagram baze podataka	16
4.2 Dijagram razreda	17
4.3 Dijagram stanja	18
4.4 Dijagram aktivnosti	19
4.5 Dijagram komponenti	20
5 Implementacija i korisničko sučelje	21
5.1 Korištene tehnologije i alati	21
5.2 Ispitivanje programskog rješenja	22
5.2.1 Ispitivanje komponenti	22
5.2.2 Ispitivanje sustava	22
5.3 Dijagram razmještaja	23
5.4 Upute za puštanje u pogon	24
6 Zaključak i budući rad	25
Popis literature	26
Indeks slika i dijagrama	27

Dodatak: Prikaz aktivnosti grupe

28

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Preuzet i postavljen predložak.	Nikola Vlahović	20.10.2023.
0.2	Dodan opis projektnog zadatke. Upisani članovi.	Marko Varga	21.10.2023.
0.3	Nadopisan opis projektnog zadatka.	Marina Hrbud	22.10.2023.

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

2. Opis projektnog zadatka

Projektni zadatak jest razvoj programske podrške za ostvarenje mrežne aplikacije BytePit. Aplikacija predstavlja platformu za rješavanje programerskih zadataka i provođenje natjecanja u programiranju. Aplikacija BytePit pruža ključne mogućnosti kao što su:

- *registracija natjecatelja i voditelja natjecanja*
- *dijeljenje osmišljenih zadataka iz natjecateljskog programiranja*
- *rješavanje zadataka i evaluacija rješenja na temelju testnih primjera*
- *kreiranje i provođenje natjecanja.*

Zbog ovih je mogućnosti aplikacija BytePit od posebnog interesa za odgojno-obrazovne ustanove koje alat mogu koristiti kao dodatak u nastavi predmeta povezanih uz programiranje (provođenje ispita ili domaćih zadaća), sudionike natjecanja u programiranju koji na platformi mogu u slobodno vrijeme vježbati i usavršavati svoje vještine te udruge i druga tijela koja se bave organizacijom natjecanja iz programiranja ili pak promocijom računalne znanosti. IT kompanije bi ovu aplikaciju mogle upotrebljavati kako bi ispitale relevantna znanja i programerske sposobnosti potencijalnih novih zaposlenika.

Aplikacija na početnoj stranici mora nuditi mogućnost registracije novim korisnicima. Proces registracije, uz mogućnost odabira uloge (voditelj natjecanja ili natjecatelj), zahtijeva unos korisničkog imena, fotografije, lozinke, imena, prezimena i adrese e-pošte. Za završetak registracije potrebno je otvoriti poveznicu dobivenu putem e-pošte. Voditeljevu registraciju mora dodatno odobriti administrator, koji pored toga ima ovlasti za upravljanje registriranim korisnicima i njihovim podacima, uključujući dodjelu prava i promjene osobnih podataka.

Tri su osnovna tipa korisnika aplikacije:

- **natjecatelji** — Natjecatelji su osnovni korisnici aplikacije i njezina glavna ciljna skupina. Osnovne usluge koje im se pružaju su sudjelovanje u programerskim natjecanjima, rješavanje zadataka i praćenje svog napretka. Na zahtjev mogu pristupiti zadacima (tijekom ili izvan termina natjecanja) . Mogu

slati programska rješenja na evaluaciju, a nakon natjecanja pratiti i rangiranje s obzirom na ostale natjecatelje. Na profilima natjecatelja, vidljive su različite statistike, uključujući broj točno riješenih zadataka, broj pokušaja rješavanja zadataka te prikaz osvojenih pehara za natjecanja. Natjecatelji mogu organizirati i virtualna natjecanja, koja se temelje na prošlim natjecanjima u kalendaru. Ta natjecanja su aktivna samo za njih, a rangiraju se prema službenim rezultatima originalnih natjecanja. Također, mogu kreirati virtualna natjecanja s nasumično odabranim zadacima, ravnomjerno raspoređenim prema težini zadataka.

- **voditelji** — Voditelji natjecanja organiziraju i upravljaju programerskim natjecanjima na platformi. To uključuje postavljanje natjecanja, dodavanje zadataka i praćenje rezultata sudionika. Prilikom kreiranja novih natjecanja moraju postaviti parametre, kao što su datum objave i trajanje natjecanja, dodati zadatke i po želji slike pehara. Dodavanje programerskih zadataka u natjecanja uključuje i postavljanje njihovih karakteristika – bodova i vremenskog ograničenja, ali i testnih primjera. Profili voditelja sadrže popis zadataka koje su učitali, s mogućnošću sortiranja.
- **administratori** — Administratori imaju ovlasti za upravljanje cijelom platformom, uključujući i korisničke račune. Upravljanje registriranim korisnicima i njihovim podacima uključuje dodjelu prava (primjerice potvrdu registracije voditelja) i promjene osobnih podataka korisnika. Administrator ima ovlasti za uređivanje svih zadataka i natjecanja pri čemu se se ne mijenjaju prethodno ostvareni rezultati.

Središnji dijelovi aplikacije su *zadaci* i *natjecanja*. Programerski zadaci su ključna komponenta aplikacije i pružaju izazove korisnicima koji žele razvijati svoje programerske sposobnosti. Svaki zadatak dolazi s nazivom, jedinstvenim opisom, brojem bodova koji se mogu osvojiti i vremenskim ograničenjem za rješavanje. Zadatak može biti postavljen kao privatan ili javan, a svaki privatni zadatak dodan u natjecanje, po završetku istog, postaje javan. Natjecatelji biraju zadatke koji ih zanimaju (ili koji su dio natjecanja) i šalju svoje rješenje u obliku izvornog programskog koda. Evaluacija rješenja (omogućena samo za jedan programski jezik) temelji se na usporedbi generiranih izlaza programa s očekivanim izlazima iz testnih primjera. Dodijeljeni broj bodova proporcionalan je postotku točnih primjera. Za svaki zadatak dostupan je popis svih natjecatelja koji su poslali rješenje za taj zadatak.

tak. Ovaj popis sadrži informacije o broju točno riješenih primjera, prosječnom vremenu izvršavanja po primjeru te omogućava pristup učitanim rješenjima. Važno je napomenuti da opcija za pristup učitanim rješenjima aktivira samo za one natjecatelje koji su uspješno i potpuno riješili zadatak.

Natjecanja predstavljaju srce platforme, potičući korisnike da pokažu svoje programerske vještine i natječu se s drugima. Voditelji natjecanja imaju ključnu ulogu u organizaciji ovih događaja. Oni kreiraju i definiraju ključne elemente natjecanja. Natjecatelji se prijavljuju za sudjelovanje u natjecanjima, a tijekom trajanja rješavaju zadatke i šalju svoja rješenja. Početkom natjecanja prijavljeni natjecatelji mogu pristupiti svim zadacima koji se nalaze u njemu. Nakon završetka natjecanja, rezultati se prikazuju na profilima natjecatelja, a osvajači pehara nagrađuju se za svoje uspjehe. Također, natjecatelji imaju mogućnost pregledavanja svih rješenja koja su drugi natjecatelji poslali za isti zadatak.

Rješenje za ovu aplikaciju biti će ostvareno pomoću razvojnog okvira Springboota i JavaScript biblioteka React kojom će biti modelirano korisničko sučelje.

Moguća poboljšanja za aplikaciju BytePit uključuju unaprjeđenje korisničkog sučelja radi olakšane navigacije, uvođenje ocjena i komentara za zadatke i natjecanja, implementaciju sustava za korisničku komunikaciju putem chatova ili foruma, bolje statističke podatke i analitiku na korisničkim profilima, integraciju s razvojnim alatima, raznolikost vrsta zadataka, napredniju evaluaciju rješenja s podrškom za različite programske jezike, te dodatak edukativnih materijala kako bi se potaknulo učenje i razvoj programerskih vještina među korisnicima.

Primjeri sličnih rješenja kroz koje ćemo proći su:

- *Edgar*
- *Sphere online judge*
- *Codeforces*

Kao i u sustavima *Edgar*, *SPOJ* i *Codeforces*, korisnici će moći pregledavati zadatke na stranici, te će moći na svom profilu vidjeti statistiku u odnosu na preostale natjecatelje, te svoj rank. Također mogu na kalendaru vidjeti vrijeme odvijanja budućih natjecanja.

list of classical problems

ID	NAME	QUALITY	USERS	ACC %	DIFFICULTY C I
1	Life, the Universe, and Everything	👍 294	207076	32.89	{ }
2	Prime Generator	👍 616	75846	15.94	15 13
3	Substring Check (Bug Funny)	👍 9	1057	7.99	41 13
4	Transform the Expression	👍 257	41908	49.17	1 1
5	The Next Palindrome	👍 450	19715	10.82	25 18
6	Simple Arithmetics	👍 10	2978	12.49	30 9
7	The Bulk!	👍 6	476	16.32	30 24
8	Complete the Sequence!	👍 30	4249	38.15	16 19
9	Direct Visibility	👍 4	247	12.48	30 21
10	Complicated Expressions	👍 11	1515	32.43	16 17
11	Factorial	👍 171	56448	44.98	1 1
12	The Game of Master-Mind	👍 1	717	49.89	17 17

Slika 2.1: Sustav SPOJ - pregled zadatka na stranici

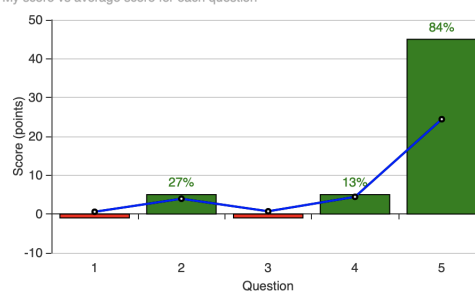
Exam score percentage distribution (n = 31)

Score distribution for all students vs mine



Per Question

My score vs average score for each question



Rank

#6 /31

83%

Percentile rank

Score

53 /70

76%

Percentage

Slika 2.2: Sustav Edgar - pregled statistike ispita

MAIN	ACMSGURU	PROBLEMS	SUBMIT	STATUS	STANDINGS	CUSTOM TEST
------	----------	----------	--------	--------	-----------	-------------

#	When	Who	Problem	Lang	Verdict	Time	Memory
229301890	Oct/22/2023 16:35 ^{UTC+2}	~JAAT~	1883A - Morning	GNU C++17	Accepted	31 ms	0 KB
229301889	Oct/22/2023 16:35 ^{UTC+2}	Part_time_Ray	1850C - Word on the Paper	GNU C++17 (64)	Accepted	15 ms	0 KB
229301887	Oct/22/2023 16:35 ^{UTC+2}	bkifhr9	1873B - Good Kid	GNU C++17	Time limit exceeded on test 1	1000 ms	0 KB
229301886	Oct/22/2023 16:35 ^{UTC+2}	manish36	486A - Calculating Function	GNU C++17	Accepted	15 ms	0 KB
229301885	Oct/22/2023 16:35 ^{UTC+2}	regain0002	1888D2 - Dances (Hard Version)	GNU C++17	Wrong answer on test 2	46 ms	37600 KB
229301883	Oct/22/2023 16:35 ^{UTC+2}	P.N.V.Sumanasree	50A - Domino piling	GNU C++20 (64)	Accepted	30 ms	0 KB
229301882	Oct/22/2023 16:35 ^{UTC+2}	_SAK_	1883E - Look Back	GNU C++17	Accepted	93 ms	1200 KB
229301881	Oct/22/2023 16:35 ^{UTC+2}	ValeriaDubinets	1808A - Lucky Numbers	GNU C++20 (64)	Time limit exceeded on test 2	1000 ms	0 KB
229301880	Oct/22/2023 16:35 ^{UTC+2}	Huihuilucky	1791C - Prepend and Append	GNU C++17	Accepted	31 ms	0 KB
229301879	Oct/22/2023 16:35 ^{UTC+2}	xgyxgy	803B - Distances to Zero	GNU C++20 (64)	Wrong answer on test 13	15 ms	800 KB
229301878	Oct/22/2023 16:35 ^{UTC+2}	vjudge3	1538C - Number of Pairs	Clang++17 Diagnostics	Wrong answer on test 2	514 ms	2100 KB
229301876	Oct/22/2023 16:35 ^{UTC+2}	imaginesabab	520A - Pangram	GNU C++17	Accepted	15 ms	0 KB
229301875	Oct/22/2023 16:35 ^{UTC+2}	gdaaay	1883C - Raspberries	GNU C++20 (64)	Accepted	15 ms	0 KB
229301874	Oct/22/2023 16:35 ^{UTC+2}	rohitrojo	1863A - Channel	GNU C++20 (64)	Wrong answer on test 1	15 ms	0 KB
229301873	Oct/22/2023 16:35 ^{UTC+2}	3inf	1884C - Medium Design	GNU C++17	Accepted	499 ms	7300 KB

Slika 2.3: Sustav Codeforces - pregled statusa na natjecanju

Calendar

Programming Contests Calendar						
Today	◀	▶	October 2023	▼	Print	Week Month Agenda ▾
Sun	Mon	Tue	Wed	Thu	Fri	Sat
Oct 1	2	3	4	5	6	7
8	9	10	11	12	13	14
11:05am CF R (Div. 1, based on 11:05am CF R 902)	4:35pm Educational CF R 156	8:15pm Stream: Teaching Ch...	4:35pm CF R 903 (Div. 3)			
15	16	17	18	19	20	21
9:45pm Stream: No code CF						
22	23	24	25	26	27	28
9:05am CF R 904 (Div. 2) 1:05pm CF R 905						4:35pm CF R
29	30	31	Nov 1	2	3	4
	3:35pm CF R (Div. 2)					

Events shown in time zone: Central European Time - Belgrade



Slika 2.4: Sustav Codeforces - kalendar nadolazećih natjecanja

Ukoliko natjecatelj zanima profil nekog drugog natjecatelja, klikom na ime profila će se otvoriti stranica na kojoj je moguće vidjeti statistiku natjecatelja: broj točno riješenih zadataka, pehari, broj svih pokrenutih zadataka. *Codeforces* dodatno daje prikaz aktivnosti natjecatelja u vremenskom periodu od jedne godine.



Slika 2.5: Sustav SPOJ - profil natjecatelja

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Voditelji
2. Natjecatelji
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) vidjeti kalendar s budućim natjecanjima
 - (b) pregledati dostupne zadatke na stranici
 - (c) pregledati profile natjecatelja i voditelja
 - (d) registrirati se u sustav stvaranjem novog korisničkog računa pri čemu odabire jednu od uloga (natjecatelj ili voditelj), a potrebni su mu: korisničko ime, fotografija, lozinka, ime, prezime te email adresa
2. Natjecatelj (inicijator) može:
 - (a) sudjelovati na natjecanju
 - (b) vidjeti rang listu natjecatelja na natjecanju kojeg je i sam bio sudionik
 - (c) vidjeti popis svih učitanih rješenja od ostalih sudionika za prethodno završena natjecanja
 - (d) pristupiti rješavanju već objavljenih zadataka
 - (e) izraditi virtualno natjecanje odabirom nekog prošlog natjecanja ili nasumičnim generiranjem zadataka
 - (f) vidjeti vlastiti profil s osobnim podacima, statistikom o broju točno riješenih zadataka, broju isprobanih zadataka te prikaz pehara za osvojena natjecanja
3. Aktivni natjecatelj (inicijator) može:

- (a) rješavati zadatke i slati datoteke s programskim kodom tijekom natjecanja na kojem se natječe
- (b) osvojiti bodove na natjecanju s obzirom na potrošeno vrijeme za rješavanje zadatka i postotak točnih primjera
- (c) na temelju postignuća za prva tri mjesta osvojiti pehar koji je vidljiv na vlastitom profilu

4. Voditelj (inicijator) može:

- (a) pregledati dostupne zadatke na stranici
- (b) izraditi novi zadatak pri čemu treba definirati: naziv zadatka, broj bodova, vremensko ograničenje, tekst zadatka i primjere za evaluaciju
- (c) organizirati novo natjecanje pri čemu treba definirati: vrijeme početka i završetka, broj zadataka, koji zadaci će biti aktivni te po želji sličicu pehara
- (d) uređivati vlastite prethodno objavljene zadatke te natjecanja
- (e) vidjeti vlastiti profil s osobnim podacima, popisom učitanih zadataka s mogućnošću sortiranja te kalendar s popisom objavljenih natjecanja

5. Administrator (inicijator) može:

- (a) uređivati sve zadatke i natjecanja
- (b) potvrditi voditelja prilikom registracije
- (c) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (d) mijenjati dodijeljena prava i osobne podatke

6. Baza podataka (sudionik) može:

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima te zadacima i natjecanjima
- (b) pohranjuje rezultate natjecanja, rješenja zadataka i statistiku natjecatelja

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 tijekom 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-programске 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 varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ

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 tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

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

4.5 Dijagram komponenti

dio 2. revizije

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

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

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

5.2 Ispitivanje programskog rješenja

dio 2. revizije

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

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

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

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

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

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

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

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

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

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

Kontinuirano osvježavanje

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

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

Indeks slika i dijagrama

2.1	Sustav SPOJ - pregled zadataka na stranici	7
2.2	Sustav Edgar - pregled statistike ispita	7
2.3	Sustav Codeforces - pregled statusa na natjecanju	8
2.4	Sustav Codeforces - kalendar nadolazećih natjecanja	8
2.5	Sustav SPOJ - profil natjecatelja	9

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: u ovom formatu: 23. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 23. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Vedran Ćutić	Antonio Glavaš	Marina Hrbud	Lara Marčec	Jakov Novak	Marko Varga	Nikola Vlahović
Upravljanje projektom							
Opis projektnog zadatka			2			2	
Funkcionalni zahtjevi				1			
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							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Vedran Čutić	Antonio Glavaš	Marina Hrbud	Lara Marčec	Jakov Novak	Marko Varga	Nikola Vlahović
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, pritiskom na stavku Contributors.