

Programsko inženjerstvo

Ak. god. 2021./2022.

CodeShark

Dokumentacija, Rev. 0.2

Grupa: *DomeFanClub*

Voditelj: *Marko Damjanić*

Datum predaje: 26. 10. 2021.

Nastavnik: *Hrvoje Nuić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	9
3.1.2 Sekvencijski dijagrami	15
3.2 Ostali zahtjevi	16
4 Arhitektura i dizajn sustava	17
4.1 Baza podataka	17
4.1.1 Opis tablica	17
4.1.2 Dijagram baze podataka	18
4.2 Dijagram razreda	19
4.3 Dijagram stanja	20
4.4 Dijagram aktivnosti	21
4.5 Dijagram komponenti	22
5 Implementacija i korisničko sučelje	23
5.1 Korištene tehnologije i alati	23
5.2 Ispitivanje programskog rješenja	24
5.2.1 Ispitivanje komponenti	24
5.2.2 Ispitivanje sustava	24
5.3 Dijagram razmještaja	25
5.4 Upute za puštanje u pogon	26
6 Zaključak i budući rad	27
Popis literature	28
Indeks slika i dijagrama	29

Dodatak: Prikaz aktivnosti grupe

30

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	26.10.2021.
0.2	Napravljen opis projekta	*	7.11.2021.

2. Opis projektnog zadatka

Rješavanjem programskih zadataka i kompetitivnim programiranjem današnji programeri razvijaju svoje logičke vještine, ubrzavaju brzinu razvoja svoga koda, pišu kvalitetniji kod i lakše briljiraju kod firmi koje cijene visoke kompetitivnosti programera.

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "CodeShark" koja će korisniku omogućiti da rješava programske zadatke prik ladne njegovim interesima i razini znanja, te da se natječe na natjecanjima organiziranih od strane voditelja. Zadatak voditeljima je stvarati zadatke korisnicima te organizirati natjecanja koja će najbolje natjecatelje nagraditi trofejima prikazanim na profilu korisnika.

Korisnici na stranici mogu pregledavati i rješavati zadatke, natjecati se u natjecanjima koja su dostupna u kalendaru, te pratiti profile ostalih natjecatelja i voditelja natjecanja. Neregistriranim korisnicima se prilikom pokretanja zadataka ili tijekom pokušaja prijave na natjecanje otvara stranica za log-in pomoću već registriranog profila (potrebno upisati korisničko ime i lozinku) te ako nemaju već registriran profil imaju opciju registracije. Za registraciju novog profila potrebni su sljedeći podaci:

- korisničko ime
- fotografija
- lozinka
- ime
- prezime
- email adresa
- željena uloga na koju se prijavljuje (voditelj natjecanja ili natjecatelj)

Registracija se završava potvrdom preko email adrese. Automatski se korisniku dodjeljuju prava natjecatelja a za ulogu i prava voditelja mora dodatno potvrditi i administrator. Registrirani korisnik može pregledavati, mijenjati osobne podatke i izbrisati svoj korisnički račun.

Natjecatelj može pristupiti stranici s zadacima za vježbanje, na kojoj je prikazan popis svih postojećih zadataka koji su već bili iskorišteni na jednom od natjecanja (javni zadatci) te ostalih zadataka koje su voditelji dodali. Zadaci se mogu pretraživati po korisničkom imenu autora, sortirati po težini, statusu riješenosti te raznim drugim kategorijama podjele. Pritiskom na zadatak otvara se stranica tog zadatka gdje mu je omogućeno učitavanje programskog rješenja u aplikaciju. Učitani program se ovisno o potrebi prevodi u računalni kod, te potom provjerava na temelju zadanih primjera. Izlaz programa mora biti jednak rješenju i izvesti se unutar zadanog vremena. Broj bodova koje je natjecatelj ostvario se dodjeljuje proporcionalno broju točno riješenih primjera.

Natjecatelj također može pristupiti stranici s natjecanjima. Na njoj je prikazan interaktivni kalendar s svim budućim, trenutnim i prošlim natjecanjima. Također sadrži i uređenu listu natjecanja koja se može sortirati po autoru, državi za koju je namijenjeno te klasi natjecanja (za početnike, veterane i otvoren upad). Pritiskom na natjecanje otvara se stranica tog natjecanja s svim bitnim detaljima. Ako se to natjecanje još nije održalo, natjecatelj se može na njega prijaviti a kada bude vrijeme početka natjecanje se može pridružiti.

U trenutku početka natjecanja svi zadatci postaju vidljivi aktivnim natjecateljima. Za svaki zadatak, natjecatelj može poslati datoteku s programskim kodom. Završetkom natjecanja objavljuje se rang lista svih natjecatelja po ostvarenom broju bodova. Natjecateljima se na temelju postignuća za prva tri mjesta dodjeljuje pehar. Za izračun osvojenih bodova na natjecanju potrebno je uzeti u obzir provedeno vrijeme za rješavanje zadatka i postotak točnih primjera.

Natjecatelj nakon natjecanja može vidjeti i popis svih učitanih rješenja od nekog drugog natjecatelja. Slično, svaki zadatak ima popis svih natjecatelja koji su učitali neko rješenje za taj zadatak, broj točnih primjera po najboljem učitavanju od natjecatelja, prosječnom vremenu izvršavanja po primjeru i gumb za dohvat učitano rješenja koji se aktivira samo onim natjecateljima koji su već potpuno točno riješili zadatak.

Ako se natjecanje već održalo natjecatelj može sebi pokrenuti virtualno natjecanje. Virtualno natjecanje je aktivno samo za natjecatelja koji ga je pokrenuo. Pri isteku ograničenog vremena ili po želji natjecatelja virtualno natjecanje završava. Pri završetku se natjecatelja rangira u usporedbi s službenim rezultatima korisnika koji su prisustvovali tom natjecanju.

Natjecatelj isto tako može napraviti virtualno natjecanje gdje aplikacija nasumično odabere zadatke, ali tako da ravnomjerno rasporedi težine zadataka prema količini bodova pojedinog zadatka.

Postoje još 2 vrste korisnika, oboje s većim pravima od natjecatelja, a to su:

- voditelj
- administrator

Voditelj ima mogućnosti učitati nove zadatke u aplikaciju i organizirati natjecanja. Za postaviti novi zadatak potrebno je:

- naziv zadatka
- broj bodova (težina 1-5)
- vremensko ograničenje izvršavanja programa
- tekst zadatka
- primjeri za evaluaciju (ulaz i izlaz programa)
- privatnost zadatka

Privatni zadatak automatski postaje javan završetkom natjecanja.

Voditelj može izraditi natjecanje i ono svima postaje vidljivo u kalendaru natjecanja. Za organizaciju natjecanja su mu potrebni:

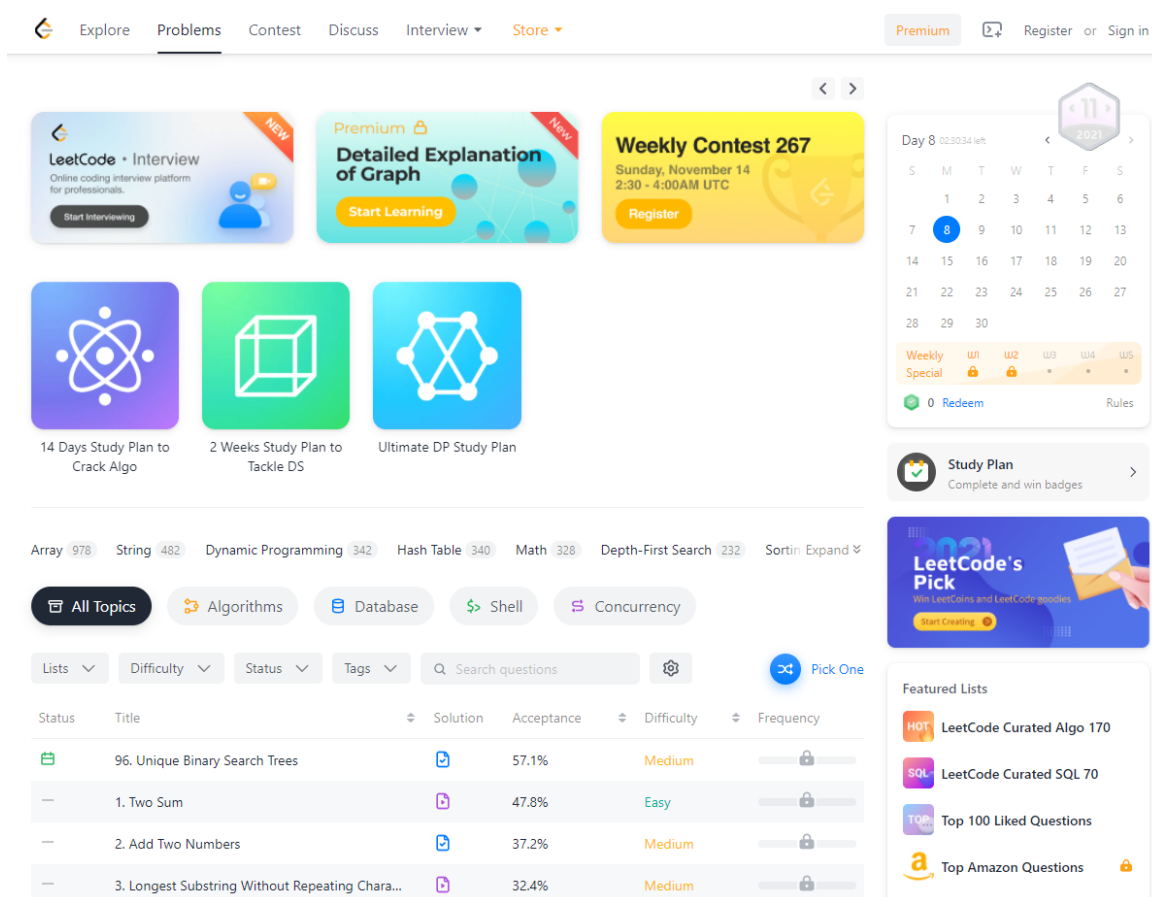
- naziv natjecanja
- tekst natjecanja
- vrijeme početka i završetka natjecanja (max 48 sati)
- broj zadataka
- odabrani privatni zadaci koje je voditelj učitao
- sličica pehara
- država natjecanja

Voditelj može uređivati vlastito objavljene zadatke i natjecanje. Uređivanjem zadatka se ne mijenjaju prethodno ostvareni rezultati na tom zadatku.

Administrator sustava ima najveće ovlasti. On može vidjeti popis svih registriranih korisnika i njihovih osobnih podataka te im mijenjati dodijeljena prava i osobne podatke. Ima potpun pristup bazi zadataka i natjecanja te ih ima pravo obrisati.

Uz vlastiti profil, u web aplikaciji postoji stranica s listom profila preko koje se mogu pogledati statistike i trofeji drugih korisnika. Na profilu natjecatelja su ispisane statistike o broju točno riješenih i broju isprobanih zadataka. Na profilima je također kalendar natjecanje na kojem je korisnik prisustvovao, a za voditelje i ona natjecanja koja su organizirali. Profili voditelja također sadrže popis njihovih učitanih zadataka s mogućnošću sortiranja.

Od sličnih rješenja istaknut je <https://leetcode.com/>. Za razliku od leetcode-a CodeShark ima mogućnosti virtualnih natjecanja koja pridonose boljoj pripremi korisnika. CodeShark se također više fokusira na svojom jednostavnosti i interaktivnosti.



Slika 2.1: <https://leetcode.com/problemset/all/>

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

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

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

Dionici:

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

Aktori i njihovi funkcionalni zahtjevi:

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

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC15 - Dodavanje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Dodati natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire karticu "Natjecanja"
 2. Voditelj dodaje novo natjecanje
 3. Voditelj unosi potrebne podatke o natjecanju
 4. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Voditelj unosi podatke u nedozvoljenom formatu
 1. Sustav obavještava voditelja o neuspjelom upisu podataka
 2. Voditelj popravljiva neispravne unose i stvara natjecanje ili odustaje od stvaranja natjecanja

UC16 - Pokretanje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Pokrenuti natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire karticu "Natjecanja"
 2. Voditelj odabire jedno od svojih natjecanja
 3. Voditelj pokreće natjecanje

UC17 - Prijava na natjecanje

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Prijaviti se na nadolazeće natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen
- **Opis osnovnog tijeka:**

1. Natjecatelj odabire karticu "Natjecanja"
2. Natjecatelj odabire jedno od nadolazećih natjecanja
3. Natjecatelj se prijavljuje na odabrano natjecanje

UC18 - Pristupanje natjecanju

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pristupiti natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav i na natjecanje koje je trenutno aktivno
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od trenutno aktivnih natjecanja na koje je prijavljen
 3. Natjecatelj pristupa natjecanju

UC19 - Odjava s natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Odjaviti se s natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav i na natjecanje
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od nadolazećih natjecanja na koje je prijavljen
 3. Natjecatelj se odjavljuje s odabranog natjecanja

UC20 - Uređivanje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Urediti natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire karticu "Natjecanja"
 2. Voditelj odabire jedno od svojih natjecanja
 3. Voditelj uređuje pojedinosti svog natjecanja
 4. Promjene se upisuju u bazu podataka

- **Opis mogućih odstupanja:**

3.a Voditelj unosi podatke u nedozvoljenom formatu

1. Sustav obavještava voditelja o neuspjelom upisu podataka
2. Voditelj popravljajući neispravne unose i uspješno uređuje natjecanje ili odustaje od uređivanja natjecanja

UC21 - Pregled sudionika na natjecanju

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati natjecatelje koji su sudjelovali na natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu "Natjecanja"
 2. Korisnik odabire jedno od prošlih natjecanja
 3. Prikazuje se lista natjecatelja koji su sudjelovali na odabranom natjecanju

UC22 - Brisanje natjecanja

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati postojeće natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Natjecanja"
 2. Administrator odabire jedno od natjecanja
 3. Administrator briše odabrano natjecanje
 4. Natjecanje se briše iz baze podataka

UC23 - Dohvat učitanoj rješenja zadatka s natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Dohvatiti učitano rješenje zadatka nekog od natjecatelja koji je sudjelovao na natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati potpuno točno riješen konkretan zadatak
- **Opis osnovnog tijeka:**

1. Natjecatelj odabire karticu "Natjecanja"
2. Natjecatelj odabire jedno od prošlih natjecanja
3. Natjecatelj odabire jedan od zadataka s odabranog natjecanja
4. Natjecatelj dohvaća rješenje zadatka

UC24 - Završetak natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Završiti s natjecanjem
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i morao je pristupiti natjecanju
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od trenutno aktivnih natjecanja na koje je prijavljen
 3. Natjecatelj pristupa natjecanju
 4. Nakon rješavanja zadataka, a prije isteka vremena, natjecatelj završava natjecanje

UC25 - Pregled ostalih korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled profila ostalih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu "Korisnici"
 2. Prikazuje se lista svih registriranih natjecatelja i voditelja

UC26 - Promjena ovlasti korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Promjeniti ovlasti korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire korisnika
 3. Administrator mijenja odabranom korisniku ovlasti

4. Promjene se upisuju u bazu podataka

UC27 - Pregled statistike pojedinog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati statistiku pojedinog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu "Korisnici"
 2. Korisnik odabire određenog korisnika s liste
 3. Prikazuje se statistika odabranog korisnika

UC28 - Pregled svih učitanih rješenja nekog natjecatelja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pregledati sva učitana rješenja natjecatelja koji je sudjelovao na natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od prošlih natjecanja
 3. Natjecatelj odabire jednog od natjecatelja koji je sudjelovao na natjecanju
 4. Prikazuju se sva učitana rješenja odabranog natjecatelja

UC29 - Promjena osobnih podataka korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti osobne podatke korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire određenog korisnika
 3. Administrator mijenja željene osobne podatke korisniku
 4. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**

3.a Voditelj unosi podatke u nedozvoljenom formatu

1. Sustav obavještava administratora o neuspjelom upisu podataka
2. Administrator popravljiva neispravne unose i uspješno mijenja osobne podatke odabranog korisnika ili odustaje od promjena

UC30 - Potvrda promjene uloge voditelja

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi promjenu uloge voditelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire određenog korisnika
 3. Administrator potvrđuje korisniku promjenu uloge u voditelja natjecanja
 4. Promjene se upisuju u bazu 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-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	https://leetcode.com/problemset/all/	7
-----	---	---

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: 16. studenoga 2021.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 16. studenoga 2021.
- 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.

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
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							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Dijagram razmještaja							
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, pritiskom na stavku Contributors.