

Programsko inženjerstvo

Ak. god. 2023./2024.

Digitalni poster

Dokumentacija, Rev. 1

Grupa: *Dipol*

Voditelj: *Vedran Lugarić*

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

Nastavnik: *Miljenko Krhen*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	6
3.1 Funkcionalni zahtjevi	6
3.1.1 Obrasci uporabe	8
3.1.2 Sekvencijski dijagrami	9
3.2 Ostali zahtjevi	10
4 Arhitektura i dizajn sustava	11
4.1 Baza podataka	12
4.1.1 Opis tablica	12
4.1.2 Dijagram baze podataka	13
4.2 Dijagram razreda	14
4.3 Dijagram stanja	15
4.4 Dijagram aktivnosti	16
4.5 Dijagram komponenti	17
5 Implementacija i korisničko sučelje	18
5.1 Korištene tehnologije i alati	18
5.2 Ispitivanje programskog rješenja	19
5.2.1 Ispitivanje komponenti	19
5.2.2 Ispitivanje sustava	19
5.3 Dijagram razmještaja	20
5.4 Upute za puštanje u pogon	21
6 Zaključak i budući rad	22
Popis literature	23
Indeks slika i dijagrama	24

Dodatak: Prikaz aktivnosti grupe

25

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak, opis projektnog zadatka i funkcionalni zahtjevi	Daniel Rodić, Lovro Petrović	1.11.2023.
0.2	Arhitektura i dizajn sustava.	Nina Kolobarić	8.11.2023.

2. Opis projektnog zadatka

dio 1. revizije

Svrha ovog projekta je razvoj programske potpore za web aplikaciju "Digitalni poster" koja će se koristiti u svrhe održavanja stručnih konferencija. Aplikacija će omogućiti pregled radova sudionika i dati im mogućnosti ocjenjivanja pojedinog rada. Svaki stručni rad bit će prikazan u obliku postera, koji se sastoji od stručnog rada i/ili izlaganja autora, te će autori moći dodati odgovarajuću prezentaciju za svoj rad. Svi će događaji biti fotografirani, a odabrane fotografije će tijekom konferencije biti dostupne svim registriranim korisnicima.

Prilikom prvog pokretanja aplikacije prikazuje se...

Za registraciju potrebni su sljedeći podaci:

- *ime i prezime*
- *e-mail adresa*
- *lozinka*
- *broj mobitela*

Prilikom registracije u sustav korisniku se automatski dodjeljuje status posjetitelja te mu se naknadno može dodijeliti status autora, voditelja ili administratora. Svaki registrirani korisnik može pristupiti svojim osobnim podacima koje može mijenjati te može obrisati svoj račun.

Posjetitelj konferencije ima minimalna prava u samoj aplikaciji, to jest posjetitelj može pregledati termine konferencija, mjesto konferencija te vremenske prilike u mjestu konferencije. U terminu konferencije posjetitelj može pristupiti samoj konferenciji prijavom za istu. Nakon prijave u konferenciju posjetitelju je dostupno direktno video praćenje trenutnih događanja u glavnoj konferencijskoj dvorani te su mu dostupni promotivni materijali pokrovitelja konferencije. Također posjetitelju je dodijeljno pravo glasa za točno jedan poster koji predstavlja pojedinog autora. Glasovanje je moguće samo tijekom određenog vremenskog razdoblja, odnosno u terminu održavanja konferencije. Kada završi sam postupak glasovanja posjetitelju kao i svim registriranim korisnicima dostupni su rezultati glasovanja. Posjetitelja će se elektroničkom poštom obavijestiti o

mjestu i vremenu održavanja dodjele nagrada za prva tri mjesta. Sve dostupne fotografije s konferencije posjetitelj će moći spremati lokalno na svoj uređaj.

Autor ima slična prava kao posjetitelj. Razlikuju se po tome što autor ne može glasati kako bi se izbjegli konflikti i subjektivnost. Autor sve potrebne materijale šalje voditelju putem elektroničke pošte. Autori nakon završetka glasanja dobivaju obavijest o svom rangu, a autori prva tri rangirana rada dobivaju pozivnicu za dodjelu nagrada.

Voditelj konferencije vrši prijavu autora, radova i postera za svoju konferenciju te su mu dostupni svi podaci istih. Također voditelj mora organizirati konferenciju na način da rezervira dvorane te od administratora traži unos svih podataka, datum, vrijeme i mjesto, u aplikaciju. Odabrane fotografije dostupne svim korisnicima za spremanje lokalno na uređaj ovjerene su od strane voditelja. Voditelj nakon glasanja sastavlja rang listu koju šalje adminu koji ju čini dostupnom svim korisnicima te obavjesti svakog autora o njegovom rangu. Dodjela nagrada za prva tri mjesta također je organizirana od strane voditelja koji osigurava novčane nagrade za svakog od pobjednika ovisno o samom rangu autora. Također mora rezervirati dvoranu za isti događaj.

Administrator sustava ima najveće ovlasti te definira sve uvjete za ispravan rad sustava. Dostupni su mu svi podaci svakog registriranog korisnika kojima može promijeniti ovlasti ili im obrisati račun. Admin odobrava i vrši zahtjeve koje dobije od voditelja. Zadužen je za konstatno ažuriranje kalendara događaja te mora paziti da ne dođe do konflikata, odnosno da se dvije konferencije ne održavaju u isto vrijeme na istom mjestu. Uz to admin može obrisati svaki prijavljeni poster koji nije primjeren iz bilo kojeg razloga.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

Dionici:

1. Administrator
2. Voditelj
3. Autor
4. Posjetitelj
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) stvoriti novi račun registracijom za koju su mu potrebni ime i prezime, e-mail, korisničko ime, lozinka i broj mobitela
 - (b) pregledati kalendar događaja, odnosno može vidjeti:
 - i. vrijeme konferencije
 - ii. mjesto konferencije
2. Posjetitelj (inicijator) može:
 - (a) upravljati svojim osobnim podacima
 - (b) pokrenuti direktno video praćenje događaja u glavnoj dvorani
 - (c) glasati za točno jedan poster
 - (d) spremati fotografije lokalno na svoj uređaj
 - (e)
3. Autor (inicijator) može:
 - (a) ???
4. Voditelj (inicijator) može:

- (a) prijaviti autora
- (b) prijaviti stručni rad
- (c) prijaviti poster
- (d) postavljati fotografije tijekom konferencije

5. Administrator (inicijator) može:

- (a) vidjeti sve podatke svakog registriranog korisnika
- (b) obrisati korisnički račun ili mu promijeniti status
- (c) dodati i/ili izbrisati konferenciju s kalendara
- (d) dodati i/ili izbrisati poster

6. Baza podataka (sudionik):

- (a) sadrži podatke o svim korisnicima i njihovom statusu
- (b) sadrži podatke o svim održanim ili budućim konferencijama
- (c) sadrži podatke o svakom posteru, autoru, broju glasova i krajnjem rang

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

Arhitektura sustava ima hijerarhijsku strukturu u kojoj svaki sloj komunicira isključivo s neposredno susjednim slojevima. Naš sustav sastoji se od pet glavnih slojeva: Korisničko sučelje, Kontroler, Servis, Repozitorij i Baza podataka. Korisničko sučelje, ili User Interface (UI) omogućava interakciju između korisnika i računala. Korisničko sučelje našeg sustava razvijeno je uz pomoć Reacta, JavaScript biblioteke koja olakšava stvaranje korisničkih sučelja. Korisničko sučelje šalje zahtjeve kontroleru temeljem korisničkih akcija i koristi JSON (JavaScript Object Notation) datoteke za prijenos podataka. Kontroler, koristeći REST API (Representational State Transfer), upravlja zahtjevima vanjskih korisnika i odgovara na njih. U većini slučajeva, kontroler radi s podacima u JSON formatu. Servis je odgovoran za upravljanje i obradu podataka dobivenih od korisničkog sučelja putem kontrolera i baze podataka putem repozitorija. Repozitorij se koristi za komunikaciju s bazom podataka i sadrži funkcionalnosti za pronalaženje određenih objekata iz baze.

Arhitektura se može podijeliti na tri osnovna podsustava: Web poslužitelj, Web aplikacija i Baza podataka. Web preglednik omogućuje korisnicima pregledavanje web-stranica i pristup multimedijalnom sadržaju na internetu. Svaki web preglednik djeluje kao prevoditelj, interpretirajući web-stranice napisane u kodu i prikazujući ih korisnicima na razumljiv način. Korisnici šalju zahtjeve web poslužitelju putem web preglednika, a web poslužitelj igra ključnu ulogu u radu web aplikacije. Njegova primarna zadaća je omogućiti komunikaciju između korisnika i aplikacije putem HTTP (Hypertext Transfer Protocol) protokola, standardnog načina prijenosa informacija na webu. Web poslužitelj pokreće web aplikaciju i proslijeđuje joj korisničke zahtjeve.

Korisnici koriste web aplikaciju za obradu svojih zahtjeva. Web aplikacija obrađuje te zahtjeve i, ovisno o njihovoj prirodi, pristupa bazi podataka putem repozitorija. Nakon obrade zahtjeva, web aplikacija preko web poslužitelja vraća odgovor u obliku HTML dokumenta koji korisnici vide u svom web pregledniku. Za razvoj web aplikacije koristi se programski jezik Python zajedno s .NET radnim okvirom i JavaScriptom, a razvojno okruženje je Microsoft Visual Studio. Arhitektura sustava temelji se na konceptu Model-

View-Controller (MVC), arhitekturnog obrasca koji se često koristi u razvoju softverskih aplikacija kako bi se postigla jasna organizacija i odvojenost različitih dijelova aplikacije. Sastoji se od tri osnovne komponente:

- *Model: predstavlja središnju komponentu sustava. On je odgovoran za upravljanje podacima logikom i pravilima aplikacije. Neovisan je o korisničkom sučelju i često sadrži dinamičke podatkovne strukture koje predstavljaju stanje aplikacije. Kada se dogodi promjena u podacima ili stanju aplikacije, Model obavještava ostale komponente sustava o tim promjenama.*
- *View: komponenta odgovorna za prikaz podataka korisnicima. To uključuje sve vizualne elemente sučelja, kao što su grafovi, tablice, forme i slično. View omogućava korisnicima da vide i koriste podatke iz Modela na način koji im je razumljiv.*
- *Controller: komponenta koja prima ulazne podatke od korisnika ili drugih izvora i upravlja njima. Kontrolira korisničke zahtjeve i daljnju interakciju s Modelom i View-om. Kada korisnik izvrši neku radnju, Controller reagira na tu akciju i donosi odluke o tome kako će se to odraziti na Model i kako će se ažurirati View.*

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

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: 8. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 8. studenoga 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.

	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							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
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.