

# Programsko inženjerstvo

Ak. god. 2023./2024.

## Digitalni poster

Dokumentacija, Rev. <1 ili 2>

Grupa: *Posterheimer*

Voditelj: *Dominik Papeš*

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

Nastavnik: *Miljenko Krhen*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>4</b>
<b>3 Specifikacija programske potpore</b>	<b>7</b>
3.1 Funkcionalni zahtjevi . . . . .	7
3.1.1 Obrasci uporabe . . . . .	8
3.1.2 Sekvencijski dijagrami . . . . .	9
3.2 Ostali zahtjevi . . . . .	10
<b>4 Arhitektura i dizajn sustava</b>	<b>11</b>
4.1 Baza podataka . . . . .	11
4.1.1 Opis tablica . . . . .	11
4.1.2 Dijagram baze podataka . . . . .	12
4.2 Dijagram razreda . . . . .	13
4.3 Dijagram stanja . . . . .	14
4.4 Dijagram aktivnosti . . . . .	15
4.5 Dijagram komponenti . . . . .	16
<b>5 Implementacija i korisničko sučelje</b>	<b>17</b>
5.1 Korištene tehnologije i alati . . . . .	17
5.2 Ispitivanje programskog rješenja . . . . .	18
5.2.1 Ispitivanje komponenti . . . . .	18
5.2.2 Ispitivanje sustava . . . . .	18
5.3 Dijagram razmještaja . . . . .	19
5.4 Upute za puštanje u pogon . . . . .	20
<b>6 Zaključak i budući rad</b>	<b>21</b>
<b>Popis literature</b>	<b>22</b>
<b>Indeks slika i dijagrama</b>	<b>23</b>

**Dodatak: Prikaz aktivnosti grupe**

24

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	V. Javornik	22.10.2023.
0.2	Dodan osnovni opis projektnog zadatka	V. Javornik	23.10.2023.

*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

### *dio 1. revizije*

*Na osnovi projektnog zadatka detaljno opisati korisničke zahtjeve. Što jasnije opisati cilj projektnog zadatka, razraditi problematiku zadatka, dodati nove aspekte problema i potencijalnih rješenja. Očekuje se minimalno 3, a poželjno 4-5 stranica opisa. Teme koje treba dodatno razraditi u ovom poglavlju su:*

- *potencijalna korist ovog projekta*
- *postojeća slična rješenja (istražiti i ukratko opisati razlike u odnosu na zadani zadatak). Dodajte slike koja predočavaju slična rješenja.*
- *skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje.*
- *moгуćnost prilagodbe rješenja*
- *opseg projektnog zadatka*
- *moгуće nadogradnje projektnog zadatka*

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije „Digitalni poster“ koja će olakšati pregled radova sudionicima stručne konferencije prikazom svakog rada i/ili izlaganja odgovarajućim posterom.

Prilikom pokretanja sustava prikazuje se [izgled početne stranice: prikaz konferencija koje se nalaze u blizini]

Neregistriranom korisniku odabirom konferencije otvaraju se opće informacije o konferenciji: [organizator, pokrovitelji, digitalni poster]. Uz to, navedena su sva izlaganja [mogu biti razvrstana po kategorijama ili vremenu predavanja]: naziv rada, autor, opis. Neregistriranom korisniku je omogućeno prijavljivanje u sustav s postojećim računom (potrebno je upisati korisničko ime i lozinku) ili kreiranjem novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- korisničko ime
- lozinka
- ime
- prezime
- email adresa

- *broj mobitela, mjesto i sl. ako će biti potrebno za funkcionalnost*

Registracijom u sustav korisniku se dodjeljuju prava posjetitelja konferencije, a naknadno mu se mogu dodijeliti prava autora ili administratora [*biti će potrebno nadodati ostale dogovorene vrste korisnika*]. Registrirani korisnik može pregledati, mijenjati osobne podatke i izbrisati svoj korisnički račun.

Posjetitelj prijavom i dolaskom na stručnu konferenciju dobiva jedinstvenu lozinku za pristup sustavu. Unošenjem te lozinke u aplikaciju otvara se [*početna stranica konferencije na kojoj se trenutno nalazi, može biti pregled postera, video sadržaj ili izabrane fotografije*]. Nakon prijave dostupni su im promotivni materijali pokrovitelja konferencije. Posjetitelji mogu pratiti trenutna događanja u glavnoj konferencijskoj dvorani pomoću direktnog videa. Imaju mogućnost glasovanja za svaki pojedini poster koji predstavlja svakog pojedinog predavača na konferenciji. Svaki posjetitelj može glasovati samo jednom tijekom određene konferencije. Glasovanje je moguće samo tijekom određenog razdoblja koje je određeno danima i vremenom održavanja konferencije. Nakon završenog postupka glasovanja, obavljaju se rezultati koji su dostupni svim registriranim korisnicima.

Izabrane fotografije dostupne su registriranim korisnicima tijekom konferencija. Fotografije korisnici mogu spremati lokalno na svoj uređaj. Dostupan im je dio s informacijama o mjestu održavanja konferencije koji između ostalog sadrži podatke o trenutnim vremenskim uvjetima i vremenskoj prognozi za navedenu lokaciju.

Uz posjetitelja postoje još  $x$  vrste korisnika, a to su

- autor rada
- administrator
- *moгуći korisnici*

organizator konferencije (ima širi spektar mogućnosti za rad u aplikaciji, omogućeno mu je ažuriranje podataka o izlaganjima, dodaje ili briše sudionike)

pokrovitelj (dostupna im je analitika vezana uz njihov sadržaj)

hotelijer (spomenut tijekom prve laboratorijske vježbe)

Autor koji sudjeluje na stručnom skupu elektroničkom poštom dostavljaju sve potrebne materijale sistemskom administratoru. Autor prima obavijest o rangi svojeg rada prema glasovima posjetitelja. Porukom elektroničke pošte ih se poziva na dodjelu nagrade za prva tri nagrađena rada. Sve sudionike se elektroničkom poštom obavještava o mjestu i vremenu dodjele nagrade.

Administrator sustava ima najveće ovlasti. On ima pristup bazi s popisom registriranih korisnika i njihovim podacima te ih može brisati. Obavlja prijavu autora, radova i postera. Ima mogućnost ažuriranja tih podataka. Dostupni su mu svi podaci te može definirati sve potrebne uvjete za ispravan rad sustava.

Sustav treba podržavati rad više korisnika u stvarnom vremenu.

## 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 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 Selenium<sup>1</sup>. 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.*

---

<sup>1</sup><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: 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.*

	Dominik Papeš	Fran Androić	Ante Batić	Valeria Javornik	Mario Perhat	Dario Tomšić
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						
Upute za puštanje u pogon						
Dnevnik sastajanja						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	<b>Dominik Papeš</b>	<b>Fran Androić</b>	<b>Ante Batić</b>	<b>Valeria Javornik</b>	<b>Mario Perhat</b>	<b>Dario Tomšić</b>
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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*