

# Programsko inženjerstvo

Ak. god. 2020./2021.

## Pub kvizovi

Dokumentacija, Rev. 1

Grupa: *RoyalStandard*

Voditelj: *Branimir Stanković*

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

Nastavnik: *Manuela Lukić*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Primjeri u $\text{\LaTeX}$ u . . . . .	9
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>13</b>
3.1	Funkcionalni zahtjevi . . . . .	13
3.1.1	Obrasci uporabe . . . . .	15
3.1.2	Sekvencijski dijagrami . . . . .	19
3.2	Ostali zahtjevi . . . . .	20
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>21</b>
4.1	Baza podataka . . . . .	23
4.1.1	Opis tablica . . . . .	23
4.1.2	Dijagram baze podataka . . . . .	23
4.2	Dijagram razreda . . . . .	24
4.3	Dijagram stanja . . . . .	25
4.4	Dijagram aktivnosti . . . . .	26
4.5	Dijagram komponenti . . . . .	27
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>28</b>
5.1	Korištene tehnologije i alati . . . . .	28
5.2	Ispitivanje programskog rješenja . . . . .	29
5.2.1	Ispitivanje komponenti . . . . .	29
5.2.2	Ispitivanje sustava . . . . .	29
5.3	Dijagram razmještaja . . . . .	30
5.4	Upute za puštanje u pogon . . . . .	31
<b>6</b>	<b>Zaključak i budući rad</b>	<b>32</b>
	<b>Popis literature</b>	<b>33</b>

**Indeks slika i dijagrama** 34

**Dodatak: Prikaz aktivnosti grupe** 35

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Šetka	28.10.2022.
0.2	Dodani opis projektnog zadatka i funkcionalni zahtjevi. Ažuriran dnevnik sastajanja.	Šetka	31.10.2022.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	*	25.08.2013.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	*	26.08.2013.
0.8	Povijest rada i trenutni status implementacije, Zaključci i plan daljnjeg rada	*	28.08.2013.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.2	Manje izmjene:Timer - Brojilo vremena	*	15.09.2013.
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predloška dokumentacije	*	28.09.2013.

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (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

U velikoj konkurenciji različitih pub kvizova kojoj i sami svjedočimo, želja je svakog sastavljača privući što veći broj ekipa i učiniti kviz što je moguće zanimljivijim. Svaki pojedini igrač ili tim želi pronaći pub kviz na kojem mu najbolje odgovara koncept pitanja i gdje je atmosfera jednostavno priča za sebe. Kako bi se povezali sastavljači, kvizaši, ali i oni koji to tek žele postati ili jednostavno probati nešto novo, cilj je razviti aplikaciju koja će znatno olakšati taj proces i ponuditi dodatne funkcionalnosti.

Glavni je cilj ove aplikacije omogućiti sastavljačima kvizova objavu nadolazećih pub kvizova koje organiziraju i osigurati igračima da vide koliko se raznolikih kvizova održava u njihovoj blizini, kao i kada se održavaju oni za koje su najviše zainteresirani. S obzirom na to da često postoje osobe koje žele igrati neki pub kviz, ali nažalost nemaju ekipu, aplikacija će riješiti i taj problem. Igrač koji nema ekipu, a ipak želi sudjelovati na pub kvizu, moći će u aplikaciji pronaći tim kojemu bi najviše odgovarao kao suigrač i tako zaigrati kviz.

Neregistrirani korisnik imat će mogućnost isključivo pregleda pub kvizova, ali s ograničenim uvidom u detalje objavljenih događaja. Moći će vidjeti naziv kviza, ime kafića, vrijeme održavanja te vrstu kviza, ali za sve ostale pojedinosti prvo se mora registrirati u sustav.

Registrirani korisnik može biti ili sastavljač ili igrač kviza. Svaka od ovih uloga ima drugačiji skup mogućnosti u aplikaciji. Iznimno, ako je jedan korisnik sastavljač pub kvizova, ali isto tako i sudjeluje na nekim drugima kao igrač, on može imati obje uloge u aplikaciji. Prilikom registracije korisnik najprije odabire ulogu na temelju koje ispunjava prilagođenu formu. Ako se registrira kao igrač ili istovremeno kao igrač i sastavljač onda unosi sljedeće podatke:

- *Ime*
- *Prezime*
- *Nadimak*

- **Email adresa**
- **Lozinka**
- **Područja znanja za kviz**
- **Imaš ekipu?**
  - [DA] **Naziv ekipe**
  - [NE]
- **Slika**
- **Broj telefona**

Pri čemu su podebljani podaci obvezni za unijeti. Ako ipak odabere samo ulogu sastavljača, onda unosi ove podatke:

- **Ime**
- **Prezime**
- **Nadimak**
- **Email adresa**
- **Lozinka**
- **Slika**
- **Broj telefona**

U slučaju da korisnik ima svoju pub kviz ekipu, pri registraciji će odabrati vrijednost DA za polje Imaš ekipu? te će morati obvezno unijeti naziv svoje ekipe. Inače, ako nema ekipu odabrat će opciju NE.

Sastavljač pub kvizova, ujedno je i organizator događaja (pub kviza) te kroz aplikaciju ima mogućnost objaviti nadolazeći događaj tako da prilikom objave unese sve potrebne podatke bitne za kviz:

- **Naziv kviza**
- **Kratki opis**
- **Ime kafića**
- **Vrijeme održavanja**
- **Lokacija**
- **Maksimalan broj ekipa**
- **Iznos kotizacije**
- **Nazivi nagrada**
- **Vrsta kviza**
- **Informacije o sastavljaču**

Pri kreiranju nove objave događaja potrebno je paziti da sastavljač ne smije objaviti više događaja koji su u isto vrijeme (s preklapanjem termina). Ako korisnik sastavljač ima i ulogu igrača, treba dodatno paziti da u isto vrijeme događaja koji je korisnik objavio, isti korisnik nije prijavljen kao igrač na neki od kvizova. Sastavljač može vidjeti sve objave pub kvizova u aplikaciji na pregledu "Svi pub kvizovi", a svoje objave vidi na pregledu "Moji pub kvizovi". Uz to, sastavljač može pregledati i uređivati svoj profil na pregledu "Moj profil" s podacima unesenim pri registraciji.

Igrač pub kvizova ima dvije glavne mogućnosti u aplikaciji, a to su pronalazak ekipe (samo za one igrače koji ju nemaju) i prijava svoje ekipe na neki od objavljenih kvizova. Ako pri registraciji korisnik navede da nema vlastitu ekipu, onda u aplikaciji ima mogućnost pronaći ju tako da odabere opciju „Pronađi tim“ i aplikacija će ga spojiti s ekipom kojoj najviše odgovara (većina područja znanja u kojima je dobar nedostaju ekipi s kojom će ga aplikacija spojiti). Kada je korisniku pronađen tim, dolazi mu obavijest s nazivom ekipe i brojem članova. Također, svakom igraču te ekipe dolazi obavijest da su dobili novog igrača i popis njegovih boljih područja znanja s kojima će doprinijeti timu. Igrači koji imaju svoju ekipu mogu je napustiti odabirom opcije "Napusti tim". Igrač može vidjeti sve objave pub kvizova u aplikaciji na pregledu "Svi pub kvizovi", a događaje na koje je prijavljen može vidjeti na pregledu "Moji pub kvizovi". Također, igrači mogu prijaviti svoju ekipu na kviz, tako da 1 igrač iz ekipe prijavljuje cijelu ekipu na određeni kviz. Pri tome treba paziti da jedna ekipa ne bude prijavljena na više različitih kvizova u isto vrijeme. Kada igrač uspješno prijavi svoju ekipu na kviz, njemu i njegovim suigračima dolazi obavijest da su prijavljeni na novi kviz i na objavi za taj događaj broj slobodnih mjesta se ažurira (jedno slobodno mjesto manje). Uz to, igrač može pregledati i uređivati svoj profil na pregledu "Moj profil" s podacima unesenim pri registraciji.

Administrator sustava uz ovlasti svih korisnika ima i neke dodatne mogućnosti u aplikaciji, a to su: blokiranje korisnika koji krše pravila sustava (na primjer imaju nepristojan nadimak ili naziv ekipe), odobravanje ili zabranjivanje objava za pub kvizove koje prethodno kreiraju sastavljači te mogućnost naknadnog brisanja istih događaja. Također, administrator može dodati administratorska prava drugim korisnicima.



Aplikacija za sve prijavljene korisnike treba omogućiti pretraživanje objavljenih događaja te filtriranje prema određenim parametrima. Funkcionalnost pretraživanja ostvarena je okvirom za pretraživanje (eng. search bar), pomoću kojeg prijavljeni korisnik može pronaći određeni događaj unoseći njegov naziv ili ime kafića u kojem se kviz održava. Filtriranje se dodatno ostvaraju na način da korisnik odabire željene vrijednosti ili raspon vrijednosti za svoju udaljenost od kafića gdje se kviz održava, iznos kotizacije, vrstu kviza ili slično.

Također, svi igrači pub kvizova imaju mogućnost uvida u statističke detalje na pregledu "Statistika", u kojem mogu vidjeti i usporediti prikazane podatke u obliku grafa ili tablice, a bilježi se prosječan broj ekipa po kvizu za nekog sastavljača, broj održanih kvizova u posljednjih tjedan dana te prosječna popunjenost kviza za pojedinu vrstu kviza.

Pretpostavka je da je 1 igrač fiksno u 1 ekipi te da ekipa ima maksimalno 5 članova.

Ovaj projekt potencijalno bi mogao koristiti za organizaciju i prijave na sve postojeće pub kvizove u Hrvatskoj, a posebna prednost je što bi sve objave kvizova bile razvrstane, pregledne i skupljene na jednom mjestu kako bi korisnici mogli brzo i jednostavno dobiti bitne informacije o pojedinom događaju. Zainteresirani skup korisnika činili bi svi sastavljači i organizatori kvizova, članovi ekipa te igrači koji žele ići na neki kviz, ali još nemaju suigrače. Opseg projektnog zadatka ponajprije obuhvaća funkcionalnosti objave kvizova, prijave i pronalazak ekipe, a moguće je i dobiti dodatne informacije o pojedinom kvizu ili sastavljaču. Također, projektni zadatak mogao bi se nadograditi tako da sastavljači mogu dodati slike, poredak ekipa i nekoliko pitanja sa svojih bivših događaja te se na taj način predstaviti svim igračima koji pregledavaju njihove profile. Dodatno proširenje bilo bi uvođenje recenzija za kvizove i sastavljače te komentari i osvrta o pojedinim događajima. Iako ne postoje programska rješenja koja nude sve mogućnosti kao ova aplikacija, postoje različite web stranice gdje se može pronaći popis kvizova za određeno mjesto i datum te osnovne informacije o kreiranim događajima. Kao jednu od najpopularnijih navodimo <https://www.pubquizzers.com/index.php>.



Slika 2.1: Primjer sličnog rješenja

## 2.1 Primjeri u $\text{\LaTeX}$ u

*Ovo potpoglavlje izbrisati.*

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti  $\text{\LaTeX}$ a koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u  $\text{\LaTeX}$ u - [https://www.fer.unizg.hr/\\_download/repository/LaTeX-upute.pdf](https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf)
- $\text{\LaTeX}$  projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

podcrtani tekst, **podebljani tekst**, *nagnuti tekst*

primjer primjer primjer primjer primjer primjer

- primjer
- primjer

- primjer
  1. primjer
    - 1.a primjer
    - b primjer
  2. primjer

primjer url-a: <https://www.fer.unizg.hr/predmet/proinz/projekt>

posebni znakovi: # \$ % & { } \_ | < > ^ ~ \

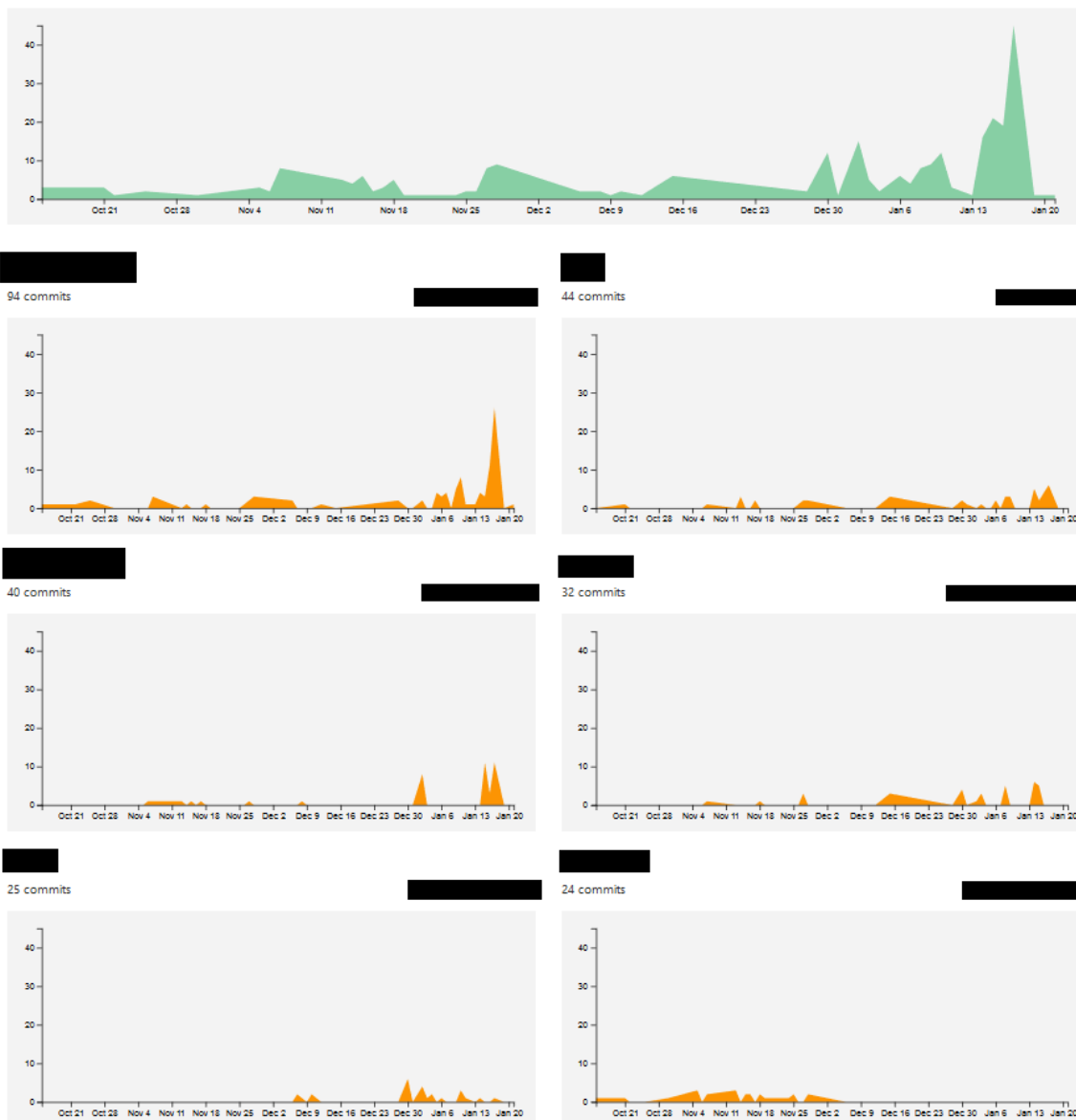
naslov unutar tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	



Slika 2.2: Primjer slike s potpisom



Slika 2.3: Primjer slike s potpisom 2

Referenciranje slike 2.3 u tekstu.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### *dio 1. revizije*

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

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

#### **Dionici:**

1. Naručitelj
2. Korisnik aplikacije
  - (a) Igrač kviza
  - (b) Sastavljač kviza
3. Administrator
4. Razvojni tim

#### **Aktori i njihovi funkcionalni zahtjevi:**

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
  - (a) registrirati se u sustav kao igrač, sastavljač ili oboje istovremeno
  - (b) pregledati osnovne informacije o događaju (pub kvizu)
2. Igrač kviza (inicijator) može:
  - (a) prijaviti se u sustav
  - (b) vidjeti sve objavljene događaje na pregledu "Svi pub kvizovi"
  - (c) vidjeti događaje na koje je prijavljen na pregledu "Moji pub kvizovi"

- (d) pronaći ekipu za kviz
- (e) napustiti svoju ekipu
- (f) prijaviti svoju ekipu na kviz
- (g) pregledati svoj profil
- (h) uređivati svoj profil
- (i) pregledati profil sastavljača
- (j) pretraživati i filtrirati događaje
- (k) vidjeti statističke podatke

3. Sastavljač kviza (inicijator) može:

- (a) prijaviti se u sustav
- (b) objaviti novi događaj (pub kviz)
- (c) vidjeti sve objavljene događaje na pregledu "Svi pub kvizovi"
- (d) vidjeti svoje objavljene događaje na pregledu "Moji pub kvizovi"
- (e) pregledati svoj profil
- (f) uređivati svoj profil
- (g) pretraživati i filtrirati događaje
- (h) vidjeti statističke podatke

4. Administrator (inicijator) može:

- (a) sve isto što mogu i ostali korisnici sustava
- (b) blokirati korisnike koji krše pravila sustava
- (c) odobriti ili zabraniti objavu koju je kreirao sastavljač
- (d) brisati objavljene događaje
- (e) davati administratorska prava drugim korisnicima

5. Baza podataka (sudionik) može:

- (a) pohranjivati sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjivati sve podatke o događajima (kvizovima)

### 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.*

#### UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi tražene podatke
  3. Podaci se spremaju u bazu podataka
  4. Korisnika se preusmjerava na početnu stranicu
- **Opis mogućih odstupanja:**
  - 2.a Korisnik unosi neispravne podatke (zauzeti ili neispravni e-mail ili korisničko ime, unos podataka u nedopuštenom formatu)
    1. Sustav upozorava korisnika na neispravnost unesenih podataka i vraća ga na stranicu za registraciju.
    2. Korisnik mijenja podatke ili odustaje od registracije

#### UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Pristupiti korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent posjeduje korisnički račun
- **Opis osnovnog tijeka:**
  1. Korisnik unosi korisničko ime i lozinku
  2. Podaci prolaze kroz provjeru



3. Korisnik dobija pristup korisničkom sučelju
- **Opis mogućih odstupanja:**
  - 2.a Uneseni su neispravni korisničko ime ili lozinka
    1. Sustav obavještava korisnika da su uneseni pogrešni podaci i vraća ga na stranicu za prijavu

#### UC3 - Pregled podataka korisničkog profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati podatke korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Moj profil“.
  2. Korisnik pregledava podatke profila.

#### UC4 - Uređivanje podataka korisničkog profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Urediti podatke korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Moj profil“
  2. Korisnik odabire opciju za promjenu podataka
  3. Korisnik mijenja odabrane podatke
  4. Korisnik sprema promjene
  5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 4.a Korisnik ne spremi promjene
    1. Promjene se odbacuju

#### UC5 - Kreiranje nadolazećih događaja

- **Glavni sudionik:** Sastavljač kviza
- **Cilj:** Kreiranje novih kvizova
- **Sudionici:** -
- **Preduvjet:** Sastavljač kviza je prijavljen u sustav
- **Opis osnovnog tijeka:**

1. Sastavljač odabire opciju „Kreiraj novi kviz“
2. Sastavljač popunjava potrebne podatke
3. Sastavljač sprema promjene
4. Novi kviz čeka odobrenje administratora

#### **UC6 - Odobravanje kreiranih događaja**

- **Glavni sudionik:** Administrator
- **Cilj:** Odobriti nove kvizove
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administratoru je dostupan popis novih kvizova koji čekaju odobrenje
  2. Administrator odabire one kvizove koje želi odobriti
  3. Baza podataka se osvježava
  4. Odobreni kvizovi se objavljuju

#### **UC7 - Zabrana kreniranih događaja**

- **Glavni sudionik:** Administrator
- **Cilj:** Zabraniti neke kvizove
- **Sudionici:** -
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administratoru je dostupan popis novih kvizova koji čekaju odobrenje
  2. Administrator odabire one kvizove koje želi zabraniti
  3. Odabrani kvizovi se odbacuju

#### **UC8 - Objava nadolazećih događaja**

- **Glavni sudionik:** Sastavljač kviza
- **Cilj:** Objaviti nove kvizove
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Sastavljač kviza je prijavljen u sustav, administrator je odobrio događaj (kviz) koji je sastavljač kreirao
- **Opis osnovnog tijeka:**
  1. Sastavljač je dobio odobrenje za kreirani kviz
  2. Sastavljač objavljuje kviz koji je vidljiv ostalim korisnicima sustavima

#### **UC9 - Pregled svih objavljenih događaja**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati sve događaje koji su objavljeni
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Svi pub kvizovi“
  2. Svi odobreni kvizovi korisniku se prikazuju na zaslonu

#### UC10 - Pregled svojih objavljenih događaja

- **Glavni sudionik:** Sastavljač kviza
- **Cilj:** Pregledati svoje kreirane kvizove
- **Sudionici:** Baza podataka
- **Preduvjet:** Sastavljač kviza je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Sastavljač kviza bira opciju „Moji pub kvizovi“
  2. Svi sastavljačevi kreirani kvizovi prikazuju mu se na zaslonu

#### **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*

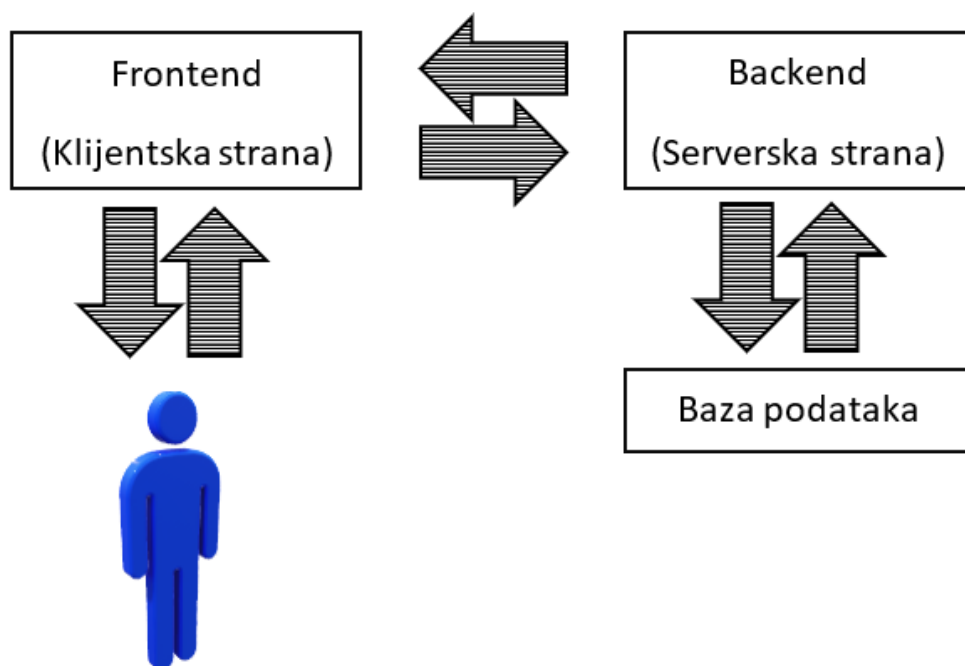
*Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.*

## 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



Slika 4.1: Arhitektura sustava

Korisnik aplikaciji pristupa putem web preglednika. Interakciju s aplikacijom ostvaruje preko korisničkog sučelja pomoću kojeg šalje zahtjeve web poslužitelju i prima odgovore.

Programski jezik pomoću kojeg je ostvaren backend web aplikacije je Java, a korišteni radni okvir je Spring Boot. Frontend aplikacije ostvaren je programskim jezikom JavaScript i bibliotekom React.js. Za razvojno okruženje odabran je IntelliJ IDEA. Spring Boot je radni okvir namijenjen stvaranju mikroservisa. Mikro servis je arhitektura koja omogućuje neovisan razvoj više različitih servisa od kojih svaki ima svoj proces.

Web aplikaciju čine tri osnovna dijela:

- **frontend**
- **backend**
- **baza podataka**

Frontend se sastoji od komponenata i logike. Istu komponentu je moguće koristiti za različite namjene (engl. reusability). React.js koristi virtualni DOM (engl. Document Object Model) čiji se sadržaj uspoređuje sa stvarnim DOM-om i na osnovu toga se provode promjene što za posljedicu ima poboljšanje performansi. Struktura ostvarena međusobnim povezivanjem različitih komponenti je stablo.

Backend se sastoji od:

- programskog sučelja za reprezentacijski prijenos stanja (REST API), odnosno Controller-a
- sloja poslovne logike (Service)
- sloja za pristup bazi podataka (Repository)

**Controller** izlaže funkcionalnost web aplikacije kao RESTful web usluge, tj. prima zahtjeve čiji su glavni dijelovi URI, metoda i HTTP zaglavlje, a korisniku šalje odgovor koji se sastoji od statusnog koda, tijela poruke i zaglavlja. U tijelu poruke se nalazi sadržaj kojeg korisnik konzumira nakon što je prikazan u web pregledniku. Komunikaciju sa slojem poslovne logike Controller ostvaruje pomoću umetanja ovisnosti (engl. dependency injection). Dependency injection je obrazac prema kojemu se u određeni objekt/funkciju umeće neki drugi objekt/funkcija na koji se prvobitno spomenuti objekt/funkcija oslanja.

**Service** omogućuje komunikaciju između slojeva Controller i Repository, zadužen je za provjeru ispravnosti podataka. Osim na sloju Service, provjera ispravnosti se obavlja na frontend-u i u bazi podataka. Komunikaciju sa slojem za pristup bazi podataka ostvaruje umetanjem ovisnosti.

**Repository** omogućuje komunikaciju s bazom podataka pomoću SQL-a. Objekti iz relacijske baze podataka pretvaraju se objekte programskog jezika Java korištenjem tehnike ORM.

### ***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
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

2.1	Primjer sličnog rješenja . . . . .	9
2.2	Primjer slike s potpisom . . . . .	11
2.3	Primjer slike s potpisom 2 . . . . .	12
4.1	Arhitektura sustava . . . . .	21

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

- Datum: 12. listopada 2022.
- Prisustvovali: svi
- Teme sastanka:
  - upoznavanje
  - dogovor oko prijedloga nove teme

#### 2. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: svi
- Teme sastanka:
  - sastanak s asistenticom i demonstratorom
  - konačan odabir alata i tehnologija
  - upute o radu

#### 3. sastanak

- Datum: 31. listopada 2022.
- Prisustvovali: svi
- Teme sastanka:
  - opis teme
  - definiranje funkcionalnih zahtjeva
  - podjela zadataka (obraci uporabe, opis arhitekture, model baze podataka)

## 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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*