Software Design

Ac. yr. 2019./2020.

Festival organisation

Documentation, Rev. *<1 ili 2>*

Group: *Festival organisation*

Coordinator: *Bartol Bilic*

Turn-in Date: *<¡Day>. <Month>. <Year>.*

Supervisor: *Hrvoje Nuic*

Contents

1 Documentation Change Log **3**

2 Project Description **5**
2.1 General Idea 6
2.2 Position in the market - the competition 6
2.3 In-Depth Description 7
2.4 User roles 7
2.4.1 Administrators 7
2.4.2 Creators 7
2.4.3 Organiser 7
2.4.4 Worker 8
2.5 Functions and happenings of the system 8
2.5.1 Festival 8
2.5.2 Auction 9
2.5.3 Job 9
2.5.4 Activity 9
2.5.5 Cards 9
2.5.6 Card Format 10
2.6 System implementation targets 10
2.7 Project Scope and Targeted Users 11
2.8 Changes, upgrades, adaptability of the Application 11
2.9 Primjeri u LaTeXu 11

3 Software Specification **15**
3.1 Funkcionalni zahtjevi 15
3.1.1 Obrasci uporabe 16
3.1.2 Sekvencijski dijagrami 17
3.2 Ostali zahtjevi 18

4 Architecture and System Design **19**
4.1 Baza podataka 19

4.1.1 Opis tablica 19
4.1.2 Dijagram baze podataka 20
4.2 Dijagram razreda 21
4.3 Dijagram stanja 22
4.4 Dijagram aktivnosti 23
4.5 Dijagram komponenti 24

5 Implementation and User Interface **25**
5.1 Koritene tehnologije i alati 25
5.2 Ispitivanje programskog rjeenja 26
5.2.1 Ispitivanje komponenti 26
5.2.2 Ispitivanje sustava 26
5.3 Dijagram razmjetaja 27
5.4 Upute za putanje u pogon 28

6 Conclusion and Outline of Planned Future Work **29**

Popis literature **30**

Image and diagram index **31**

Dodatak: Prikaz aktivnosti grupe **32**

1. Documentation Change Log

Kontinuirano osvjeavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predloak.	Ivoevi	22.08.2013.
0.2	Dopisane upute za povijest dokumentacije. Dodane reference.	Jovi	24.08.2013.
0.5	Dodan *Use Case* dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	Ivoevi	25.08.2013.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	Grudeni	26.08.2013.
0.8	Povijest rada i trenutni status implementacije, Zakljuci i plan daljnjeg rada	Ivoevi	28.08.2013.
0.9	Opisi obrazaca uporabe	Jovi	07.09.2013.
0.10	Preveden uvod	Jovi	08.09.2013.
0.11	Sekvencijski dijagrami	uak	09.09.2013.
0.12.1	Zapoeo dijagrame razreda	Horvat	10.09.2013.
0.12.2	Nastavak dijagrama razreda	Horvat	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Ivoevi	11.09.2013.
1.1	Ureivanje teksta – funkcionalni i nefunkcionalni zahtjevi	Grudeni Jovi	14.09.2013.
1.2	Manje izmjene:Timer - Brojilo vremena	Grudeni	15.09.2013.
1.3	Popravljeni dijagrami obrazaca uporabe	Jovi	15.09.2013.
1.5	Generalna revizija strukture dokumenta	Ivoevi	19.09.2013.
1.5.1	Manja revizija (dijagram razmjetaja)	Jovi	20.09.2013.

Rev.	Opis promjene/dodatka	Autori	Datum
2.0	Konani tekst predloka dokumentacije	Ivoevi	28.09.2013.

Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Izmeu tih revizija mogu postojati manje revizije ve prema tome kako se dokument bude nadopunjavao. Oekuje se da nakon svake znaajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafikih sadraja) dokumenta se to zabiljei kao revizija. Npr., revizije unutar prvog ciklusa e imati oznake 0.1, 0.2, , 0.9, 0.10, 0.11.. sve do konane revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

2. Project Description

dio 1. revizije

Na osnovi projektnog zadatka detaljno opisati korisnike zahtjeve. to jasnije opisati cilj projektnog zadatka, razraditi problematiku zadatka, dodati nove aspekte problema i potencijalnih rjeenja. Oekuje se minimalno 3, a poeljno 4-5 stranica opisa. Teme koje treba dodatno razraditi u ovom poglavlju su:

- *potencijalna korist ovog projekta*
- *postojea slina rjeenja (istraiti i ukratko opisati razlike u odnosu na zadani zadatak). Dodajte slike koja predoavaju slina rjeenja.*
- *skup korisnika koji bi mogao biti zainteresiran za ostvareno rjeenje.*
- *mogunost prilagodbe rjeenja*
- *opseg projektnog zadatka*
- *mogue nadogradnje projektnog zadatka*

Za pomo pogledati reference navedene u poglavlju Popis literature, a po potrebi konzultirati sadraj na internetu koji nudi dobre smjernice u tom pogledu.

2.1 General Idea

The idea of this app is to enable a low to mid size festival organisation in a relatively simple and straight-forward manner that would be easily accessible and understandable even to non-technically educated Users.

The application would be open-source, and would run on the Android OS - a native mobile app.

2.2 Position in the market - the competition

Mainly, the competition consists of either high-profile professional apps, or non-native(non-mobile) apps. Thus, the point of this app is to fill that gap - it's supposed to be a native app, that's relatively simple to use, and very portable and easily deployable.

This would also imply, and goes hand in hand, with the fact that the application would be easy to use and easily accessible to a wide range of people - from highly-trained professional IT Users all the way to non-IT savvy amateur/inexperienced Users.

Because of the low difficulty, and relatively ad rem employability, this app would be more suitable to the lower skill level(entry to mid-level) Users, as it is likely that Pro Users would require a larger scale App for, probably, larger caliber Festivals that they deal with. With that in mind however, this app can be used as a mini, mobile device reminder version of whatever Pro tool is used for Festival organisation.

In the market there is presence of both event organisation, and user-event interface apps. This app is focused on organisation, and not festival-goers. Therefore, from a marketplace standpoint, there will be no impact on the demand of the application due to the selected specialisation.

2.3 In-Depth Description

Application would be used for Festival organisation - music festivals, film festivals, library events, food and alcohol festivals, parties, birthdays, and other kinds of meet-ups. By scope and complexity it would be used for smaller and medium scale Events/Festivals.

The system would be run and moderated by Administrators. The system/platform would allow concurrent usage by multiple Users. These Users are: Administrators, Creators, Organisers, Workers

2.4 User roles

2.4.1 Administrators

Maintain and organize the platform. They curate the Creators. They have complete transparency and access to all data. They can veto and issue bans on: Festivals, Jobs, User registrations, comments, etc... Basically they have complete control over the platform.

2.4.2 Creators

Need to be verified by one of the Administrators. Have the ability to create and edit Festivals, as well as have complete transparency of the Festivals they have created. Can undertake certain actions regarding the details of their Festivals(and Jobs).

They appoint and verify Organisers. To these Organisers they assign Festivals that need to be organised - delegation of Festival planning and execution. Organisers cannot organise concurrent Festivals.

2.4.3 Organiser

Organisers are appointed by the Creators. They manage, plan and execute Festivals on both a macro and micro/detail scale. They are appointed to Festivals by Creators.

They can organise multiple Festivals, and be appointed to those Festivals by multiple Creators - it is only important that none of those Festivals are concurrent.

They organise and manage Jobs and Activities that need to be done for the Festival. Jobs are performed by Workers. Organisers also have the ability to write and read comments on the Worker's walls as to better organise these Jobs.

Job and Activity management is done via Auctions. First Workers apply to these Auctions, and their entries need to be verified by the Organisers. And then the Verified Entries can compete to receive the Job Offer. Should they deem it necessary, Organisers can extend Auctions by one day.

2.4.4 Worker

They perform Jobs and Activities. For these Jobs they first have to apply to the selection process(Auctions) during which their entries are curated by the Organisers(or eventually Administrators or Creators). Upon verification, their entries compete in order for the Worker to receive the Job Offer.

Some Jobs are necessary to be done by multiple Workers. Workers can specify in their Auction Entry how many more Workers would the Job require. Jobs can be done concurrently(parallelised), and Workers can perform Jobs with maximum freedom, so long as these Jobs aren't concurrent - in case they are, the system will immediately veto such entries.

2.5 Functions and happenings of the system

Here some inner elements of the system will be defined, along with their attributes and characteristics.

2.5.1 Festival

Festivals are events that are being organised. Their attributes are:

1. Name
2. Description
3. Location
4. Planned start-end time

2.5.2 Auction

Auctions are part of the process where Workers apply to Jobs. First Workers need to turn in their entries, which are then verified by the Organisers/Creators, and moderated by Administrators. Upon successful verification/moderation, Workers' entries finally enter the Auction competition where the lowest bid ends upon the expiration of the Auction period. If need be, Organisers and Creators can extend the Auction period by one day.

Auctions have the following attributes:

1. Price
2. Comment
3. Number of Workers needed(Workforce Quantity)
4. Estimated time to Completion(ETC)

2.5.3 Job

Jobs are performed by Workers. They are constituted of Activities that need to be done in order for the Job to be completed. Jobs can require multiple Workers. The lowest bid Worker is assigned the Job. Licitations usually last 1 day, but if necessary Organisers and/or Creators can extend them by an additional day.

Jobs are to be given a sequential order of execution. They can be parallelised as well. Multiple Workers cannot work on concurrent Jobs.

2.5.4 Activity

Jobs consist of multiple Activities that need to be done in order for the Job to be completed - as already defined above. Activities help break down Jobs into manageable, and easily organised chunks. Since Jobs can be done by multiple Workers, as well as be parallelised, Activities can help with that task as they would be the elementary particle, the smallest unit of work that needs to be done and distributed throughout the network of Workers that would be performing the given Job.

2.5.5 Cards

*Participants in the organisation of the Festival would each receive a **SINGLE** card - with an exception of multiple Workers working on the same Job - that will be explained*

further below. Workers also have another specificity - their Cards feature some additional information.

A single Card can be printed for a single User and a single Festival. Therefore, the same User is able to download and print multiple Cards for multiple Festivals.

2.5.6 Card Format

The cards can be downloaded in a .pdf format - with the dimensions 10x7cm

Everyone's Cards feature the following information:

1. User picture
2. Name and Surname
3. Name and Logo of the Festival
4. QR code(MD5 Hash):
   (a) User's Name
   (b) Name of the Festival

*Workers' Cards contain **ADDITIONAL** information:*

1. Time and Location of the Job
2. QR code(MD5 Hash):
   (a) ID of the Job
   (b) ID of the User

In case a Job needs multiple Workers - multiple Cards will be printed for all the Workers.

2.6 System implementation targets

*The Platform is meant to enable **MULTIPLE** Users **CONCURRENT** access to, and usage of the Platform. This would make festival organising a dynamic and fast environment. As such, a modern object-oriented programming language will be used for implementation.*

Since the application will be developed for Android OS, we have chosen Java as the programming language, and SQLite has been chosen for the database software.

2.7 Project Scope and Targeted Users

The software would be targeted towards people organising a festival and/or participating in its execution - administrators, technicians, investors, musicians, other kinds of artists, influencers, ...

The software, as said before, is aimed at primarily festivals of smaller sizes, but could be employed for festivals up to certain 'medium' size. Especially with improvements, it could be a viable free, open-source and easily protable alternative for both medium-sized festivals and small-sized festivals.

2.8 Changes, upgrades, adaptability of the Application

The application as it stands currently has some unfortunate restrictions on the freedom that Administrators, Creators and Organisers. It would be possible to modify these restrictions as to allow a greater freedom of Festival organisation, but still keeping some in place as to prevent abuse and misuse. Certain functionalities could also be added in order to make the app more useful to mid-sized Festivals.

A few changes that would probably help tremendously is to implement an intra-platform messaging service, feedback system, and support/ticked service. This would allow the communication within the App between Users. It would also provide a way for the Developers to easily diagnose, track and reproduce bugs, as well as see what changes, ideas and updates Users would like to see in the App. Finally, a support/ticket system would help alleviate any frustration that Users could suffer due to bugs and/or failures of the platform.

2.9 Primjeri u LaTeXu

Ovo potpoglavlje izbrisati.

U nastavku se nalaze razliiti primjeri kako koristiti osnovne funkcionalnosti LaTeXa koje su potrebne za izradu dokumentacije. Za dodatnu pomo obratiti se asistentu na projektu ili potraiti upute na sljedeim web sjeditima:

- Upute za izradu diplomskog rada u LaTeXu - `https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf`

- 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
    2. primjer

primjer url-a: `https://www.fer.unizg.hr/predmet/opp/projekt`

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

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

Table 2.3: Naslov ispod tablice.

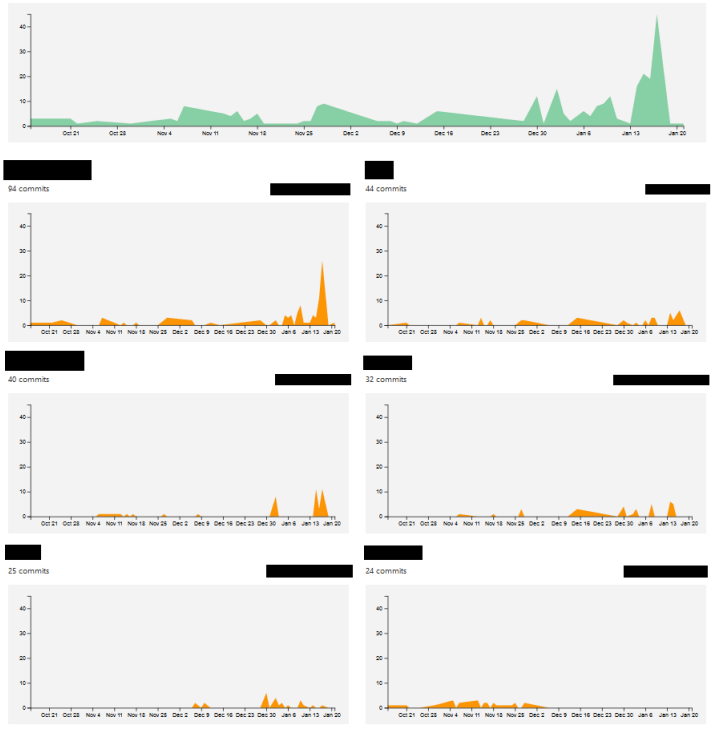

Figure 2.1: Primjer slike s potpisom

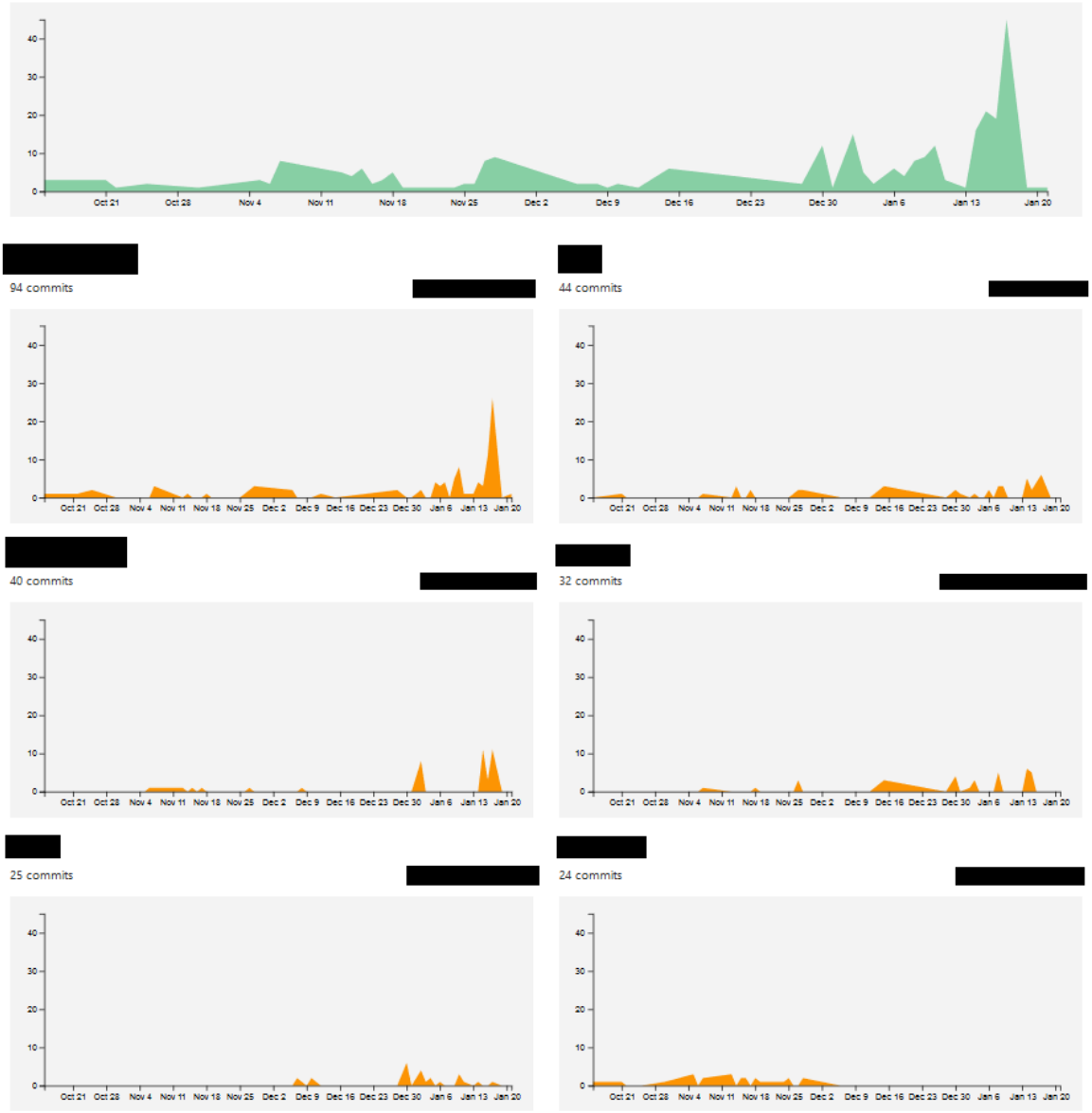

Figure 2.2: Primjer slike s potpisom 2

3. Software Specification

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, naruitelji, razvojni tim.*

*Navesti **aktore** koji izravno **koriste** ili **komuniciraju sa sustavom**. Oni mogu imati inicijatorsku ulogu, tj. zapoinju odreene procese u sustavu ili samo sudioniku ulogu, tj. obavljaju odreeni 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) moe:
   (a) funkcionalnost 1
   (b) funkcionalnost 2
      i. podfunkcionalnost 1
      ii. podfunkcionalnost 2
   (c) funkcionalnost 3
2. Aktor 2 (sudionik) moe:
   (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 predloku. Ukoliko u nekom koraku moe doi do odstupanja, potrebno je to odstupanje opisati i po mogunosti ponuditi rjeenje kojim bi se tijek 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 etiri>
  5. <opis korak pet>
- **Opis moguih odstupanja:**

  2.a <opis mogueg scenarija odstupanja u koraku 2>
    1. <opis rjeenja mogueg scenarija korak 1>
    2. <opis rjeenja mogueg scenarija korak 2>

  2.b <opis mogueg scenarija odstupanja u koraku 2>

  3.a <opis mogueg scenarija odstupanja u koraku 3>

Dijagrami obrazaca uporabe

Prikazati odnos aktora i obrazaca uporabe odgovarajuim UML dijagramom. Nije nuno 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 najvanije 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 ponaati** i koja **ogranienja** treba potivati (performanse, korisniko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vaem projektu mogu biti: podrani jezici korisnikog suelja, vrijeme odziva, najvei mogui podrani broj korisnika, podrane web/mobilne platforme, razina zatite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije reenice.*

4. Architecture and System Design

dio 1. revizije

Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremita podataka, mrene protokole, globalni upravljaki tok i sklopovsko-programske zahtjeve. Po tokama razraditi i popratiti odgovarajuim skicama:

- *izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zato ste ba odabrali takvu arhitekturu)*
- *organizaciju sustava s najvie razine apstrakcije (npr. klijent-posluitelj, baza podataka, datoteni sustav, grafiko suelje)*
- *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 slino.

4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predloku. Lijevo se nalazi tono ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom oznaite primarni klju. Svjetlo plavom oznaite strani klju

korisnik - ime tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

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 kljuevi moraju biti oznaeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".

4.2 Dijagram razreda

Potrebno je priloiti dijagram razreda s pripadajuim opisom. Zbog preglednosti je mogue dijagram razlomiti na vie njih, ali moraju biti grupirani prema slinim razinama apstrakcije i srodnim funkcionalnostima.

dio 1. revizije

Prilikom prve predaje projekta, potrebno je priloiti potpuno razraen dijagram razreda vezan uz ***generiku funkcionalnost*** *sustava. Ostale funkcionalnosti trebaju biti idejno razraene u dijagramu sa sljedeim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zatieni), nazivi atributa razreda, veze i odnosi izmeu 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 priloiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje ***znaajan dio funkcionalnosti*** *sustava. Na primjer, stanja korisnikog suelja i tijek koritenja neke kljune funkcionalnosti jesu znaajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priloiti dijagram aktivnosti s pripadajuim opisom. Dijagram aktivnosti treba prikazivati znaajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priloiti dijagram komponenti s pripadajuim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementation and User Interface

5.1 Koritene 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 znaenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno ***navesti internet poveznicu*** *gdje se mogu preuzeti ili vie saznati o njima.*

5.2 Ispitivanje programskog rjeenja

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 sluajeva. 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 sluajeva*** *u kojima e se ispitati redovni sluajevi, rubni uvjeti te izazivanje pogreke (engl. exception throwing). Poeljno je stvoriti i ispitni sluaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priloiti izvorni kd svih ispitnih sluajeva te prikaz rezultata izvoenja ispita u razvojnom okruenju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristei radni okvir Selenium[1]*. Razraditi* ***minimalno 4 ispitna sluaja*** *u kojima e se ispitati redovni sluajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogreku kako bi se vidjelo na koji nain sustav reagira kada neto nije u potpunosti ostvareno. Ispitni sluaj se treba sastojati od ulaza (npr. korisniko ime i lozinka), oekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih sluajeva pomou radnog okvira Selenium mogue je provesti pomou jednog od sljedea dva alata:

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

Detalji o koritenju alata Selenium bit e prikazani na posebnom predavanju tijekom semestra.

[1] `https://www.seleniumhq.org/`

5.3 Dijagram razmjetaja

dio 2. revizije

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

5.4 Upute za putanje u pogon

dio 2. revizije

U ovom poglavlju potrebno je dati upute za putanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kda dolazi do potpuno postavljene baze podataka i posluitelja 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 raunalo. Ukoliko mobilne i stolne aplikacije komuniciraju s posluiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporuuje se ***naglasiti korake instalacije uporabom natuknica*** *te koristiti to je vie mogue* ***slike ekrana*** *(engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovrenu aplikaciju potrebno je pokrenuti na javno dostupnom posluitelju. Studentima se preporua koritenje neke od sljedeih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Conclusion and Outline of Planned Future Work

dio 2. revizije

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

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

Literature

Kontinuirano osvjeavanje

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

1. Oblikovanje programske potpore, FER ZEMRIS, http://www.fer.hr/predmet/opp
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

Image and diagram index

2.1 Primjer slike s potpisom 13
2.2 Primjer slike s potpisom 2 14

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvjeavanje

U ovom dijelu potrebno je redovito osvjeavati dnevnik sastajanja prema predloku.

1. sastanak
    - Datum: u ovom formatu: November 6, 2019
    - Prisustvovali: I.Prezime, I.Prezime
    - Teme sastanka:
        - opis prve teme
        - opis druge teme
2. sastanak
    - Datum: u ovom formatu: November 6, 2019
    - Prisustvovali: I.Prezime, I.Prezime
    - Teme sastanka:
        - opis prve teme
        - opis druge teme

Tablica aktivnosti

Kontinuirano osvjeavanje

Napomena: Doprinose u aktivnostima treba navesti u satima po lanovima 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							
Koritene tehnologije i alati							
Ispitivanje programskog rjeenja							
Dijagram razmjetaja							
Upute za putanje u pogon							
Dnevnik sastajanja							
Zakljuak i budui rad							
Popis literature							

	Ime Prezime voditelja	**Ime Prezime**	**Ime Prezime**	**Ime Prezime**	**Ime Prezime**	**Ime Prezime**	**Ime Prezime**
Dodatne stavke kako ste podijelili izradu aplikacije							
npr. izrada poetne stranice							
izrada baze podataka							
spajanje s bazom podataka							
back end							

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.