

Programsko inženjerstvo

Ak. god. 2023./2024.

KuhajIT

Dokumentacija, Rev. <1 ili 2>

Grupa: *wall-e-zohari*

Voditelj: *Renato Brašnić*

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

Nastavnik: <Ime i prezime nastavnika zaduženog za vašu grupu>

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Cilj i opis projekta na temelju dobivenog zadatka	5
2.2	Potencijalna korist projekta	5
2.3	Slična, već postojeća rješenja	6
2.3.1	kuhar.ba	6
2.4	Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje	7
2.5	Mogućnost prilagodbe rješenja	7
2.6	Opseg projektnog zadatka	8
2.7	Moguće nadogradnje projektnog zadatka	8
2.7.1	Podržavanje hardverskog skenera kodova	8
2.7.2	Opcije za pristupačnost	9
3	Specifikacija programske potpore	10
3.1	Funkcionalni zahtjevi	10
3.1.1	Obrasci uporabe	11
3.1.2	Sekvencijski dijagrami	12
3.2	Ostali zahtjevi	13
4	Arhitektura i dizajn sustava	14
4.1	Baza podataka	14
4.1.1	Opis tablica	14
4.1.2	Dijagram baze podataka	15
4.2	Dijagram razreda	16
4.3	Dijagram stanja	17
4.4	Dijagram aktivnosti	18
4.5	Dijagram komponenti	19
5	Implementacija i korisničko sučelje	20
5.1	Korištene tehnologije i alati	20

5.2	Ispitivanje programskog rješenja	21
5.2.1	Ispitivanje komponenti	21
5.2.2	Ispitivanje sustava	21
5.3	Dijagram razmještaja	22
5.4	Upute za puštanje u pogon	23
6	Zaključak i budući rad	24
	Popis literature	25
	Indeks slika i dijagrama	26
	Dodatak: Prikaz aktivnosti grupe	27

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Predložak postavljen na repozitorij	*	20.10.2023.
0.2	Prva verzija opisa projektnog zadatka	*	26.10.2023.
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.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
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

2.1 Cilj i opis projekta na temelju dobivenog zadatka

Cilj projekta je razviti web aplikaciju koja omogućuje dijeljenje recepata te tako pomaže korisnicima pri dijeljenju kulinarskih ideja, suradnji s drugim kreatorima recepata, organizaciji vlastite dijetete i otkrivanju novih načina za pripremu jela. Tako korist od nje mogu imati profesionalni kuhari, nutricionisti, ali i kuhari amateri te osobe koje radi zdravstvenog stanja moraju prilagođavati dijetu. Novopridošlim korisnicima se prikazuju recepti sortirani od novijih prema starijima, što omogućuje da bez registracije imaju pregled kulinarskih trendova i odluče hoće li je nastaviti koristiti. Moguće se registrirati kao klijent, kuharski entuzijast ili nutricionist. Korisnik može administratorima poslati zahtjev za registraciju koji, jednom kad je odobren, dopušta korisniku da pretražuje profile i recepte na temelju dijetete koju prate. Entuzijasti kreiraju recepte i kuharice, koje su skupovi recepata, dok nutricionisti kreiraju dijetete koje nameću ograničenja na recepte. Korisnici neposredno prije kuhanja mogu unijeti prehrambeni artikl QR kodom ili izborom iz kataloga te tako filtrirati recepte u kojima se on nalazi. Tada se recepti sortiraju prema tome koliko dobro prate dijetu korisnika, odnosno koliko poštuju njena ograničenja. Korisnici također vide statistiku povijesti unosa nutritivnih vrijednosti. Što se implementacije tiče, aplikacija mora biti objektno orijentirana.

2.2 Potencijalna korist projekta

Osim očite koristi da korisnici među sobom dijele i objavljuju recepte i dijetete, korist se može naći i u statistici unosa nutritivnih vrijednosti te vrstama recepata koju prate. Uzmimo u obzir da aplikacija „zaživi“, odnosno dostigne broj korisnika dovoljan da predstavlja državu ili neki njen dio. Tada bi vladajući mogli iskoristiti informaciju o tome što ljudi najviše jedu i subvencionirati domaće proizvođače tih proizvoda jer je njihova prodaja izglednija. S druge strane, mogli bi ciljati na raznolikost prehrane pa subvencionirati one koji proizvode hranu koja je u manjku. Profit tada nije tako izgledan kao u prvom slučaju, ali bi financijska motivacija

navela proizvođače da se preusmjere te tako dođe do veće dostupnosti raznolike prehrane na tržištu. S obzirom da aplikacija vodi računa o preferencijama korisnika i dijetama koje odabiru, novi vlasnici restorana i drugih lokala koji poslužuju hranu bi mogli to iskoristiti da se specijaliziraju, odnosno nude proizvode u skladu s nekom dijetom. Ljudi koji ne smiju konzumirati određene namirnice ili skupove namirnica radi zdravstvenog stanja ili uvjerenja ne mogu jesti u većini lokala. Podatci o dijetama i receptima koje korisnici vole omogućuju vlasnicima novih lokala da ponude takvim osobama obroke u obliku specijalnih menija ili pak specijaliziraju cijeli lokal za tu dijetu.

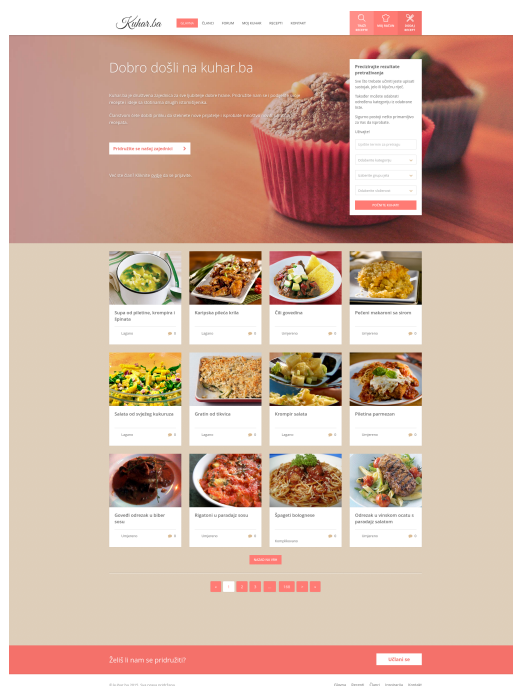
2.3 Slična, već postojeća rješenja

Kriteriji da rješenje smatramo sličnim:

- Aplikacijaweb mjesto mora biti forum, to jest mora imati mogućnost registriranja korisnika koji onda mogu raspravljati.
- Tema diskusije mora biti hrana.
- Mora postojati mogućnost filtriranja recepata prema više od jednog kriterija, ne nužno istovremeno.

2.3.1 kuhar.ba

<https://kuhar.ba> – datum pristupa: 26.10.2023. Web mjesto na naslovnoj stranici ima mogućnost filtriranja recepata prema složenosti te kategorijama i grupama jela. Moguće se registrirati te potom dodavati recepte, komentirati ih te raspravljati na forumu. Recepti imaju fotografiju, popis sastojaka, detaljni opis koraka i oznake.



Slika 2.1: prednja strana web mjesta kuhar.ba

2.4 Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje

- kulinarski entuzijasti
- nutricionisti
- klijenti
 - profesionalni kuhari
 - kuhari amateri
 - osobe kojima je potrebna specijalna dijeta radi zdravstvenih razloga ili uvjerenja

2.5 Mogućnost prilagodbe rješenja

Mana trenutnog rješenja je da ograničeno na objektno orijentirano pisanje jer to otežava skaliranje u budućnosti. Radi se o web (ili mobilnoj) aplikaciji koja treba imati sučelje koje je lako za razumjeti i koristi se u kuhinji, gdje, uz uobičajene laptopa, mobitele i tablete, korisnici mogu imati i ostale pametne uređaje poput frižidera ili pećnica. Svi ti uređaji imaju web preglednik (ili operacijski sustav

sličan Androidu ili nekoj konfiguraciji Linux-a) te su potencijalni klijenti. Što je više klijenata, baza podataka u pozadini je veća, a time i broj objekata, koji potiču problem upravljanja memorijom. Preporučeni programi, Java i Python te Javascript, koji izvršavaju preglednici, svi redom koriste garbage collector. Činjenica je da taj proces šteti brzini rada aplikacije jer algoritam za „čišćenje“ mora proći cijelu memoriju. Što više objekata, to više posla za taj algoritam pa stoga i veći period vremena koji sustav ili dijelovi sustava moraju provesti čekajući. Još k tomu niti dodavanje radne memorije ne pomaže jer to također algoritmu zadaje dodatan posao. Razumljivo je da se u vidu projekta na fakultetu objektno-orijentirana paradigma koristi u svrhu izrade dijagrama razreda (i drugih dijagrama, ali poglavito njega). Međutim, činjenica je da će takvo rješenje u upotrebi naići na probleme radi svega navedenog.

2.6 Opseg projektnog zadatka

Projekt obuhvaća:

- izradu web sučelja
- bazu podataka i komunikaciju s njom
- skeniranje QR kodova te njihovo generiranje
- izradu statističkog izvještaja na temelju podataka

2.7 Moguće nadogradnje projektnog zadatka

2.7.1 Podržavanje hardverskog skenera kodova

S obzirom da aplikacija koristi kodove kao identifikator proizvoda, implementacija skeniranja hardverskim skenerom je moguća nadogradnja projektnog zadatka. Takvo rješenje bi zahtijevalo financijski trošak, ali bi poboljšalo rad aplikacije i olakšalo implementaciju. Umjesto „čupanja“ koda iz datoteke sa slikom, uz gotov skener bi bilo dovoljno pročitati kod kao niz znakova na portu na korisničkom uređaju. To bi uzrokovalo znatno manju potrošnju mrežnog prometa i smanjilo vrijeme postavljanja koda na poslužitelj u usporedbi sa slikom. Također bi se rasteretio poslužitelj jer u tom slučaju nema potrebe za izvršavanjem algoritma za raspoznavanje koda iz slike.

2.7.2 Opcije za pristupačnost

Korištenje aplikacije svodi se uglavnom na čitanje recepata. Stoga, bilo bi opravdano implementirati funkcionalnost korisničkog sučelja koja olakšava korištenje korisnicima koji imaju poteškoće s raspoznavanjem boja i slova. Dakle, riječ je o osobama s velikom dioptrijom, vrstama daltonizma i disleksijom. Kao funkcionalnost bi trebalo uvesti način rada u visokom kontrastu te mogućnost odabira fonta koji olakšava čitanje. Također bi bilo dobro implementirati pomični kursor kako bi se korisnik lakše orijentirao 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. 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 Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

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

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

Kontinuirano osvježavanje

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

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	prednja strana web mjesta kuhar.ba	7
-----	--	---

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: u ovom formatu: 26. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 26. 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.

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.