Oblikovanje programske potpore

Ak. god. 2018./2019.

*Dostavljaona*

Dokumentacija, Rev. *0.1*

Grupa: *BitSoftTechnologies*

Voditelj: *Luka Mrković*

Datum predaje: *<dan>. <mjeseca> <godine>.*

Nastavnik: *Hrvoje Nuić*

Sadržaj

[1. Dnevnik promjena dokumentacije 3](#_Toc431806045)

[2. Opis projektnog zadatka 5](#_Toc431806046)

[3. Pojmovnik 6](#_Toc431806047)

[4. Funkcionalni zahtjevi 7](#_Toc431806048)

[5. Ostali zahtjevi 8](#_Toc431806049)

[6. Arhitektura i dizajn sustava 9](#_Toc431806050)

[6.1. Svrha, opći prioriteti i skica sustava 10](#_Toc431806051)

[6.2. Dijagram razreda s opisom 11](#_Toc431806052)

[6.3. Dijagram objekata 12](#_Toc431806053)

[6.4. Ostali UML dijagrami 13](#_Toc431806054)

[7. Implementacija i korisničko sučelje 14](#_Toc431806055)

[7.1. Dijagram razmještaja 14](#_Toc431806056)

[7.2. Korištene tehnologije i alati 15](#_Toc431806057)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 16](#_Toc431806058)

[7.4. Ispitivanje programskog rješenja 17](#_Toc431806059)

[7.5. Upute za instalaciju 18](#_Toc431806060)

[7.6. Korisničke upute 19](#_Toc431806061)

[8. Zaključak i budući rad 20](#_Toc431806062)

[9. Popis literature 21](#_Toc431806063)

[Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 22](#_Toc431806064)

[Dodatak B: Dnevnik sastajanja 23](#_Toc431806065)

[Dodatak C: Prikaz aktivnosti grupe 24](#_Toc431806066)

[Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 26](#_Toc431806067)

*Sadržaj bi se trebao automatski osvježavati prema tekstu (desni klik, „Update Field“) ako se bude držalo zadanih formata poglavlja.*

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Dokumentacija spremna za popunjavanje | Mrković | 13.10.18. |
|  |  |  |  |
|  |  |  |  |

*Moraju postojati glavne revizije (verzije, inačice) 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, (ukoliko bude više od toga nastavlja se sa 0.10, 0.11…) sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd. Ukoliko se želi naglasiti veličina promjene/dodatka unutar revizije može se napraviti različita „granulacija“ u označavanju revizija. Npr. nakon 0.1 može biti 0.11, 0.12, 0.15 pa tek onda 0.2. Ili se za veće promjene/zahvate može nakon 0.1 odmah skočiti na 0.5. itd. Bitno je da na krajevima ciklusa revizije koje se predaju budu s oznakama* ***1.0*** *i* ***2.0*** *(Preporuka je da te revizije budu podebljane unutar liste).*

# Opis projektnog zadatka

Cilj ovog projekta je razviti aplikaciju za organizaciju dostave hrane *Fina i brza hrana.* Aplikacija će korisniku omogućiti pregled i naručivanje hrane iz više različitih restorana. Restoranima će aplikacija pružiti platformu na kojoj će moći predstaviti i prodavati svoju ponudu.

Unutar aplikacije, ovisno o dodijeljenim dozvolama, razlikujemo sljedeće vrste korisnika:

* Neregistrirani korisnik
* Klijent
* Vlasnik restorana
* Dostavljač
* Dispečer
* Administrator

Neregistrirani korisnici mogu pregledavati ponudu i podatke o svim, u aplikaciji dostupnim, restoranima. Također, svi neregistrirani korisnici imaju opciju stvaranja korisničkog računa.

Jednom kada korisnik stvori korisnički račun, automatski postaje klijent. Kao klijent, može iz ponuda restorana odabrati željene proizvode koje tada može dodati u košaricu. U košarici, klijent ima mogućnost uklanjanja proizvoda od kojih je odustao, te promjene željene količine svakog proizvoda. Kada je zadovoljan odabirom, klijent finalizira narudžbu unosom lokacije dostave na karti. Također, klijent na karti može pratiti trenutnu lokaciju narudžbe (dostavljača). Klijent može predložiti svoj restoran. Nakon unosa svih potrebnih podataka o restoranu, i nakon odobrenja od strane administratora, klijent postaje Vlasnik restorana.

Svaki vlasnik restorana može unositi stavke (proizvode) u meni svog restorana. Također, može uređivati (mijenjati detalje i brisati) već postojeći meni restorana. Aplikacija vlasniku šalje e-mail sa detaljnim opisom narudžbe svaki put kada klijent finalizira narudžbu u vlasnikovom restoranu. Kao vlasnik restorana, također ima dostupnu listu svih trenutno aktivnih narudžbi (u izradi i dostavi), kao i narudžbi dostavljenih u zadnja dva sata.

Dostavljač prilikom prijave postaje aktivan. U aplikaciji vidi popis zadataka koje mu je dispečer dodijelio. Ti zadaci ne moraju nužno biti iz istih restorana. Također, na karti vidi iscrtanu rutu, pozicije i redoslijed zadataka. Dostavljač mora, nakon završetka zadatka isti označiti gotovim.

Dispečer dodjeljuje zadatke trenutno aktivnim dostavljačima. U aplikaciji vidi listu svih trenutno neraspoređenih narudžbi. Na karti vidi početne i odredišne lokacije neraspoređenih narudžbi, trenutne pozicije svih aktivnih dostavljača i rute kojima oni već moraju proći. Svaka početna lokacija neraspoređene narudžbe pokraj sebe ima prikazano vrijeme preuzimanja (vrijeme kada će biti spremna).

Administrator ima pristup popisu svih aktivnih korisnika, te njihovim osobnim podacima. Mora provjeriti, i ukoliko je restoran valjan, potvrditi prijedloge restorana od strane korisnika. Može svakom korisniku promijeniti razinu pristupa aplikaciji (dozvole). Također, može mijenjati, unositi i brisati proizvode u ponudama svih restorana.

U ovom projektu, fokus nije razvoj API-ja za komuniciranje sa servisima koji će biti potrebni za ostvarenje zadatka, već implementacija objektne strukture koja će omogućiti jednostavno i sigurno funkcioniranje aplikacije. Sve dorade već postojećih open-source API-ja bit će izvršene samo u slučaju da trenutna rješenja ne nude funkcionalnosti potrebne za izradu aplikacije. Uz objektnu strukturu u pozadini, značajan napor bit će usmjeren u izradu i dizajn korisničkog sučelja, te prikaz interaktivnih mapa unutar aplikacije.

Po završetku projekta, aplikacija bi trebala predstavljati primamljivo rješenje svim malim restoranima koji žele povećati vidljivost i promet.

*Na osnovi projektnog zadatka opisati korisničke zahtjeve. Što jasnije opisati cilj projektnog zadatka, i opširnije opisati problematiku zadatka, razraditi točke postavljenog problema, dodati nove aspekte problema i potencijalnih rješenja (već prema želji).*

*Za pomoć pogledati reference navedene u poglavlju „Reference“, ili sve što se nađe na Webu da nudi dobre smjernice u tom pogledu.*

*Npr., neke od mogućih točaka bile bi (ne nužno tim redoslijedom ni tekstom):*

*Cilj i potencijalna korist ovog projekta je …*

*Ono što već postoji iz domene primjene …*

*Problem koji treba riješiti …*

*Potrebno je ostvariti ….*

*Mogućnosti za implementaciju rješenja su …*

*Predviđa se izrada…*

*Ključne točke kod implementacije sustava su …*

*Rješenja koji projekt nudi odnosi se na domenu …*

*Skup korisnika koji bi mogli biti zainteresirani za razvijeni sustav su…*

*Ono što predviđeno rješenje ne obuhvaća je …*

*Nakon završetka projekta dobro bi bilo razmotriti mogućnosti za nadogradnje u pogledu …*

# Pojmovnik

* **OPEN-SOURCE SOFTWARE**
  + Vrsta programske podrške čiji je izvorni kod izdan pod licencom koja korisnicima dopušta proučavanje, mijenjanje i distribuiranje bilo kome i ubilo koju svrhu
* **API (application programming interface)**
  + Skup određenih pravila i specifikacija koje programeri slijede kako bi se mogli služiti uslugama ili resursima operacijskog sustava ili nekog drugog složenog programa kao standardne biblioteke rutina

*Lista pojmova (napisanih abecednim redom) potrebnih za razumijevanje teksta dokumentacije. Svaki pojam treba biti dosljedno opisan u jednoj ili dvije rečenice da se izbjegnu nejasnoće u specifikaciji sustava i opisu implementacije i rezultata.*

# Funkcionalni zahtjevi

*Funkcionalni zahtjevi su određeni samom specifikacijom sustava.*

*Nekoliko točaka je potrebno popuniti u ovom dijelu dokumentacije:*

* *Navesti koje su dionici koji imaju interes u ovom sustavu ili su nositelji odgovornosti. To su prije svega korisnici, ali i administratori sustava, sponzori, manageri.*
* *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 svakoga od njih treba navesti ulogu i svrhu njihovog djelovanja u sustavu.*
* *Napraviti analizu obrazaca uporabe (engl.* Use Cases*):*
  + *Za sve obrasce koji su u planu za implementaciju napraviti detaljne opise (prema donjem predlošku).*
  + *Nacrtati dijagrame obrazaca uporabe (engl.* Use Case Diagrams*).*
* *Nacrtati sekvencijske dijagrame za sve obrasce uporabe sustava. Uz svaki dijagram dodati odgovarajući opis.*

# Ostali zahtjevi

*Ostali ili nefunkcionalni zahtjevi nisu izravno vezani za primarnu funkciju sustava. Govore o određenim mjerama kvalitete i svojstvima i ograničenjima platforme i procesa ostvarenja sustava.*

*Pod mjerama i zahtjevima na kvalitetu sustava misli se na nekoliko aspekata: neosjetljivost na pogreške, (is)koristivost, pouzdanost, performanse, (p)održivost.*

*Pod ograničenjima se uglavnom misli na svojstva tehnologije i metodologiju implementacije: ograničenja tehnologije, vrijeme izrade, (ne)usvojeni standardi, informacije o procesu razvoja i oblikovanja.*

*Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

# Arhitektura i dizajn sustava

*Ovo je je jedno od bitnijih poglavlja jer se opisuju temelji implementacije sustava. Moguće je razlikovati više aspekata arhitekture i oblikovanja sustava od kojih je većinu potrebno dokumentirati već prema zahtjevima projektnog zadatka. Neki od tih su: stil arhitekture i identifikacija podsustava, preslikavanje na radnu platformu, spremišta podataka, mrežni protokoli, globalni upravljački tok, sklopovsko-programski zahtjevi.*

*U svakoj stavci navedenoj u nastavku potrebno je dati tekstovni opis. te ilustrirati odgovarajućim UML dijagramima. Projektna grupa odlučuje gdje i koje sve dijagrame treba prikazati u pojedinim stavkama, ali cijela točka 6. Arhitektura i dizajn sustava, mora sadržavati barem po jedan:*

*Dijagram razreda*

*Dijagram objekata*

*Dijagram stanja*

*Dijagram aktivnosti*

*Komunikacijski dijagram*

*Dijagram komponenti*

*Ostali UML dijagrami (npr. Sekvencijski za objekte) mogu se koristiti po potrebi.*

*Za crtanje UML dijagrama preporuka je koristiti alat Astah Community.*

*Preporuča se da ovo poglavlje podijelite po sljedećim potpoglavljima:*

*6.1. Svrha, opći prioriteti i skica sustava*

*6.2. Dijagram razreda s opisom*

*6.3. Objektni dijagram.*

*6.4. Ostali UML dijagrami*

## Svrha, opći prioriteti i skica sustava

*Potpoglavlje treba sadržavati sljedeće:*

*Koji sustav ili dio sustava ovaj dio dokumentacije opisuje, izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu).*

*Organizacija sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje (ako nije do kraja poznato, dopuniti u 2. ciklusu).*

## Dijagram razreda s opisom

*Potpoglavlje treba sadržavati dijagram razreda, jedan ili više njih, s opisima šta koji razred znači. Ono što je potrebno imati u dokumentaciji prilikom prve predaje je: svi razredi specifični za vaš projekt, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazive atributa razreda, veze i odnosi između razreda, okviri (forme) grafičkog sučelja specifične za pojedini programski jezik, ako su vam poznate.*

*Primjer onoga što nije potrebno imati u dokumentaciji prilikom druge predaje su: tipovi parametara metoda razreda (npr. boolean, int), događaji (npr. klikovi miša).*

*Ako za neke stvari niste sigurni kako će biti implementirane, tada je bolje taj dio izostaviti prilikom prve predaje pa dopuniti na drugoj predaji. Prilikom druge predaje navest ćete sve one razrede sa svim detaljima koje direktno koristite u implementaciji sustava.*

## Dijagram objekata

*Prikazati stanje sustava u određenom trenutku. Dio je Revizije 1.*

## Ostali UML dijagrami

*Ovdje počinju sadržaji Revizije 2.*

*U ovom potpoglavlju potrebno je uvrstiti dodatna četiri dijagrama koji prikazuju arhitekturu sustava i to: komunikacijski dijagram, dijagram stanja, dijagam aktivnosti i dijagram komponenti. Potrebno je dati barem po jedan primjerak svakog od dijagrama. Potrebno je da svaki od dijagrama prikazuje (jedan) bitan dio funkcionalnosti sustava. Dijagram komponenti treba prikazivati sve komponente sustava. Prema potrebi, ovdje se mogu dodati i neki dodatni sekvencijski dijagrami za objekte.*

# Implementacija i korisničko sučelje

*U ovom poglavlju potrebno je:*

*dati dijagram razmještaja (deployment dijagram)*

*navesti koje su tehnologije i alati korišteni u razvoju sustava*

*dati isječak programskog koda koji implementira neku od temeljnih funkcionalnosti u sustavu*

*objasniti kako je ispitano implementirano rješenje i pokazati bar 4 ispitna scenarija*

*dati upute za instalaciju*

*dati upute za korištenje*

## Dijagram razmještaja

*Potrebno je umetnuti dijagram razmještaja i po potrebi ga opisati.*

## Korištene tehnologije i alati

*Navesti sve tehnologije i alate koji su primijenjeni pri izradi projekta te ih ukratko opisati; njihovo značenje i mjesto i način primjene.*

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

*U ovom poglavlju potrebno je prikazati isječak programa koji prema mišljenju studenta ostvaruje temeljnu funkcionalnost u sustavu (ili nekom modulu).*

## Ispitivanje programskog rješenja

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva.*

*Prilikom prezentacije svojih Ispitnih scenarija (minimalno četiri) studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete. Poželjno je da se napravi i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane te da se vidi na koji način sustav reagira kada nešto nije u potpunosti ostvareno.*

*Različiti ulazi za ispitne scenarije trebaju pokrivati temeljnu funkcionalnost nekog modula i nekoliko rubnih uvjeta.*

## Upute za instalaciju

*U ovom poglavlju potrebno je dati upute za instalaciju ostvarenog prototipa.*

## Korisničke upute

*Korisničke upute ovisit će o količini implementirane funkcionalnosti. Očekuje se da upute budu na oko pet A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika.*

# Zaključak i budući rad

*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.*

# Popis literature

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

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
3. I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
4. T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
5. Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
6. I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
7. Concepts: Requirements, <http://www.upedu.org/upedu/process/gcncpt/co_req.htm>
8. UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
9. Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
10. Astah Community, <http://astah.net/editions/community/>

# Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

*U ovom dijelu dodatka potrebno je ispisati (numerirane) liste slika, dijagrama, tablica, ispisa kôda i ostalih pomoćnih sadržaja iz teksta dokumentacije.*

# Dodatak B: Dnevnik sastajanja

*U ovom dijelu dodatka potrebno je prenijeti sadržaj dnevnika sastajanja, tj. datoteke 'dnevnik.txt' iz repozitorija grupe. Datoteku 'dnevnik.txt' potrebno je voditi kroz repozitorij, a ovdje će to biti preneseno kao dio konačne dokumentacije.*

# Dodatak C: Prikaz aktivnosti grupe

*U ovom dijelu dodatka potrebno je navesti*

* *popis članova grupe i njihovih zaduženja, tj. odrađenih poslova (u postocima ukupno odrađenog posla)*
* *prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je u* BitBucket *sučelju instalirati dodatak* AwesomeGraph *koji daje prikaz grafički prikaz aktivnosti grupe, tj. '*commit' *akcija tijekom trajanja projekta, te na kraju projekta generirane grafove prenijeti u ovo poglavlje dokumentacije.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Popis aktivnosti** | **Članovi grupe** (abecednim redom) | | | | | | |
| Ime Prezime voditelja | Ime Prezime 2 | Ime Prezime 3 | Ime Prezime 4 | Ime Prezime 5 | Ime Prezime 6 | Ime Prezime 7 |
| **Upravljanje projektom** |  |  |  |  |  |  |  |
| **Opis projektnog zadatka** |  |  |  |  |  |  |  |
| **Rječnik pojmova** |  |  |  |  |  |  |  |
| **Opis funkcionalnih zahtjeva** |  |  |  |  |  |  |  |
| **Opis ostalih zahtjeva** |  |  |  |  |  |  |  |
| **Arhitektura i dizajn sustava** |  | | | | | | |
| Svrha, opći prioriteti i skica sustava |  |  |  |  |  |  |  |
| Dijagram razreda s opisom |  | | | | | | |
| Dijagram objekata |  |  |  |  |  |  |  |
| Ostali UML dijagrami |  |  |  |  |  |  |  |
| **Implementacija i korisničko sučelje** |  | | | | | | |
| Dijagram razmještaja |  |  |  |  |  |  |  |
| Korištene tehnologije i alati |  |  |  |  |  |  |  |
| Isječak programskog kôda |  |  |  |  |  |  |  |
| Ispitivanje programskog rješenja |  |  |  |  |  |  |  |
| Upute za instalaciju |  |  |  |  |  |  |  |
| Korisničke upute |  |  |  |  |  |  |  |
| **Plan rada** |  |  |  |  |  |  |  |
| **Pregled rada i stanje ostvarenja** |  |  |  |  |  |  |  |
| **Zaključak i budući rad** |  |  |  |  |  |  |  |
| **Popis literature** |  |  |  |  |  |  |  |
| **Dodaci** |  | | | | | | |
| Indeks |  |  |  |  |  |  |  |
| Dnevnik sastajanja |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

*Napomena: Doprinose u aktivnostima treba navesti u postocima po članovima grupe. Zbroj postotaka u svakom retku treba biti 100%.*

Pregled pohrana kroz vrijeme trajanja projekta (primjer):



# Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

*U ovom poglavlju potrebno je navesti:*

* *(u rev. 1) koji je plan rada za rev. 2,*
* *(u rev. 2) koji je status implementacije u odnosu na postavljene ciljeve, procjenu vremena dovršetka projekta (ako zadatak nije u potpunosti ispunjen), koje bi bile smjernice za daljnji rad kad bi se nastavilo s projektom te u čemu bi se sastojale buduće nadogradnje.*