Oblikovanje programske potpore

Ak. god. 2018./2019.

*Auto servis*

Dokumentacija, Rev. *1*

Grupa: *LittleSkillzException*

Voditelj: *Ante Žužul*

Datum predaje: *16. studenoga 2018.*

Nastavnik: *Vlado Sruk*

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 | Napravljene početne izmjene predloška i dodan opis projektnog zadatka. | Žužul | 19.10.2018. |
| 0.2 | Dopisane upute za povijest dokumentacije.  Dodane reference. | Jović | 24.08.2013. |
| 0.6 | Dodan *Use Case* dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A | Ivošević | 25.08.2013. |
| 0.7 | Arhitektura i dizajn sustava, algoritmi i strukture podataka | Grudenić | 26.08.2013. |
| 0.8 | Povijest rada i trenutni status implementacije,  Zaključci i plan daljnjeg rada | Ivošević | 28.08.2013. |
| 0.82 | Opisi obrazaca uporabe | Jović | 07.09.2013. |
| 0.88 | Preveden uvod | Jović | 08.09.2013. |
| 0.9 | Sekvencijski dijagrami | Žužak | 09.09.2013. |
| 0.91 | Započeo dijagrame razreda | Horvat | 10.09.2013. |
| 0.92 | Nastavak dijagrama razreda | Horvat | 11.09.2013. |
| 1.0 | Verzija samo s bitnim dijelovima za 1. ciklus | Ivošević | 11.09.2013. |
| 1.1 | Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi | Grudenić | 14.09.2013. |
| 1.11 | Manje izmjene: Timer > Brojilo vremena | Grudenić | 15.09.2013. |
| 1.12 | Popravljeni dijagrami obrazaca uporabe | Jović | 15.09.2013. |
| 1.5 | Generalna revizija strukture dokumenta | Ivošević | 19.09.2013. |
| 1.55 | Manja revizija (dijagram razmještaja) | Jović | 20.09.2013. |
| 2.0 | Konačni tekst predloška dokumentacije | Ivošević | 28.09.2013. |
| 3.0 | Nadopuna poglavlja o arhitekturi | Jović | 25.11.2013. |
| 4.0 | Nadopune po pitanju praćenja aktivnosti u grupi | Ivošević | 01.10.2015. |
| 4.1 | Definirani Dodaci C i D | Ivošević | 05.10.2015. |

*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 učinkoviti informacijski sustav za prijavu korisnika na auto servis. Također, sustav će omogućiti praćenje statusa popravka vozila koji su na servisu duže od jednog dana.

Potrebno je ostvariti primanje na servis vozila samo jednog proizvođača, sve njihove modele. Uz to, vrlo bitno je osigurati određena vremenska ograničenja:

* Radno vrijeme servisa radnim je danom od 7:00 do 18:00 sati.
* Prijava vozila na popravak aktivna je 24 sata dnevno, 7 dana u tjednu.
* Vozila se zaprimaju svakog radnog dana do 10 sati ujutro.
* Preuzimanje vozila moguće je svakog radnog dana poslije 14 sati.

Skup korisnika informacijskog sustava auto servisa sačinjavaju:

* administratori
* ovlašteni serviseri
* registrirani korisnici (članovi)
* neregistrirani (anonimni) korisnici

Administrator sustava može dodavati, brisati i mijenjati informacije o sebi i svim registriranim korisnicima. Administrator upisuje podatke o ovlaštenim serviserima, za svakoga korisničko ime i lozinku.

Ovlašteni serviser, nakon spajanja na sustav, ima pregled svih korisnika prijavljenih za taj radni dan. Dodatno, ima uvid u sve podatke koje je registrirani korisnik upisao. Podatke može mijenjati, ali i dodavati nove (npr. posebne napomene). Nakon završetka pregleda, ovlašteni serviser ispisuje obrazac i daje ga korisniku na potpis. Obrazac se smatra potvrdom predaje vozila ovlaštenom serviseru.

Osobe koje žele koristiti usluge servisa po prvi put, moraju se registrirati tako što upisuju određene podatke o sebi (obavezne i neobavezne). Obavezni podaci su ime, prezime, broj telefona i tip vozila korisnika te godina proizvodnje i registarska oznaka vozila. Prije svake prijave vozila na popravak, registrirani korisnik mora se prijaviti na sustav korisničkim imenom (adresa elektroničke pošte) i lozinkom. Korisniku je u svakom trenutku omogućena izmjena vlastitog korisničkog profila, kao i brisanje istog.

Neregistrirani korisnik može vidjeti osnovne informacije o uslugama servisa, ali ne može prijavljivati vozilo na servis.

Nakon što se registrirani korisnik spoji na sustav, ponuđen mu je izbor ovlaštenog servisera kod kojega se želi prijaviti. Postoje dvije mogućnosti:

* Korisnik JE odabrao željenog servisera – prikazuju mu se slobodni termini odabranog servisera u sljedećih 10 dana.
* Korisnik NIJE odabrao željenog servisera – prikazuju mu se slobodni termini svih servisera u sljedećih 10 dana. Ovisno o odabranom terminu, sustav korisniku sam dodjeljuje servisera.

Nakon odabira termina i dodjele servisera, korisniku se nude standardne usluge na odabir. Međutim, korisnik može i sam dodati neku novu uslugu ili opis problema koji ima. Na kraju prijave korisnik se može izjasniti želi li zamjensko vozilo dok je njegovo na popravku.

Predviđa se trajanje vremenskih intervala za prijavu na servis u koracima od 20 minuta.

Korisniku se nakon uspješne prijave, na njegovu adresu elektroničke pošte, šalje detaljna obavijest o prijavi. Ukoliko korisnik prijavljuje vozilo za popravak u periodu duže od 3 dana, sustav ga automatski podsjeća na prijavu (putem elektroničke pošte) zadnji dan prije odabranog termina za dolazak na servis.

Auto servis ima ukupno 10 zamjenskih vozila na raspolaganju za njihove korisnike. Sustav mora kontrolirati zauzeće vozila i na zahtjev ih rezervirati za određenog korisnika. U slučaju da nema slobodnih vozila, korisnik mora biti obaviješten kako servis nije u mogućnosti osigurati mu zamjensko vozilo.

Jedna od ključnih točaka implementacije sustava je omogućavanje istovremenog rada administratora, ovlaštenih servisera i neograničenog broja registriranih i neregistriranih korisnika.

# Pojmovnik

*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)).*

PROMJENA

## 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.*