Oblikovanje programske potpore

Ak. god. 2018./2019.

*Dostavljaona*

Dokumentacija, Rev. *0.3*

Grupa: *BitSoftTechnologies*

Voditelj: *Luka Mrković*

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

Nastavnik: *Hrvoje Nuić*

Sadržaj

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

[2. Opis projektnog zadatka 4](#_Toc431806046)

[3. Pojmovnik 6](#_Toc431806047)

[4. Funkcionalni zahtjevi 7](#_Toc431806048)

[5. Ostali zahtjevi 19](#_Toc431806049)

[6. Arhitektura i dizajn sustava 20](#_Toc431806050)

[6.1. Svrha, opći prioriteti i skica sustava 21](#_Toc431806051)

[6.2. Dijagram razreda s opisom 22](#_Toc431806052)

[6.3. Dijagram objekata 23](#_Toc431806053)

[6.4. Ostali UML dijagrami 24](#_Toc431806054)

[7. Implementacija i korisničko sučelje 25](#_Toc431806055)

[7.1. Dijagram razmještaja 25](#_Toc431806056)

[7.2. Korištene tehnologije i alati 26](#_Toc431806057)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 27](#_Toc431806058)

[7.4. Ispitivanje programskog rješenja 28](#_Toc431806059)

[7.5. Upute za instalaciju 29](#_Toc431806060)

[7.6. Korisničke upute 30](#_Toc431806061)

[8. Zaključak i budući rad 31](#_Toc431806062)

[9. Popis literature 32](#_Toc431806063)

[Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 33](#_Toc431806064)

[Dodatak B: Dnevnik sastajanja 34](#_Toc431806065)

[Dodatak C: Prikaz aktivnosti grupe 35](#_Toc431806066)

[Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 37](#_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. |
| 0.11 | Dodani dionici i njihovi funkcionalni zahtjevi | Šterbić | 15.10.18. |
| 0.12 | Rad na obrascima uporabe | Ramljak | 19.10.18. |
| 0.13 | Rad na obrascima uporabe | Matešić | 20.10.18 |
| 0.14 | Rad na obrascima uporabe | Barišić | 20.10.18 |
| 0.15 | Dodani sekvencijski dijagrami | Šterbić | 20.10.18 |
| 0.2 | Dodan opis, spojeni obrasci uporabe, formatiranje | Mrković | 21.10.18. |
| 0.3 | Dorađeni obrasci uporabe | Ramljak | 26.10.18 |

*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

**Dionici:**

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

**Aktori i njihovi funkcionalni zahtjevi:**

* Neregistrirani korisnik, inicijator
  + Može pregledati proizvode svih prijavljenih restorana (ime proizvoda, opis, cijena i slika)
  + Može saznati općenite informacije o restoranu (ime restorana, lokacija na karti, kontakt telefon i slika)
  + Može kreirati novi korisnički račun
* Klijent, inicijator
  + Svaki korisnik koji koristi aplikaciju samo radi naručivanja hrane
  + Može u aplikaciji odabrati željene proizvode iz kataloga, spremiti ih u košaricu te završiti kupovinu
  + Može preko aplikacije pratiti trenutnu poziciju narudžbe
  + Može predložiti vlastiti restoran u aplikaciju
* Vlasnik restorana, inicijator
  + Korisnik koji ima ista prava kao i klijent samo mu je dodatno omogućeno upravljanje nad informacijama vlastitog restorana
  + Može unositi, mijenjati i brisati proizvode iz menija vlastitog restorana
  + Može pregledavati popis svih trenutno aktivnih narudžbi i narudžbi koje su se dostavile tijekom prošla 2h
* Dispečer, inicijator
  + Korisnik čiji je zadatak rasporediti zaprimljene narudžbe slobodnim dostavljačima i odrediti redoslijed kojim će obaviti dostavu
  + Ima pristup popisu svih neraspoređenih narudžbi, popisu svih aktivnih dostavljača i njima dodijeljenih zadataka.
  + Može dostavljaču dodijeliti zadatak uzmi narudžbu, ostavi narudžbu i idi na novu lokaciju
* Dostavljač, sudionik
  + Korisnik čija je uloga izvršavati zadatke dodijeljene sa strane dispečera
  + Ima pristup informacijama o pozicijama, rutama i redoslijedu dodijeljenih zadataka
  + Pri izvršenju zadatka mora ga u aplikaciji označiti kao gotov
* Administrator, inicijator
  + Registrirani korisnik s najvišim ovlastima
  + Ima neograničen pristup bazi podataka
  + Korisnicima može mijenjati razinu pristupa aplikaciji
  + Može dodavati, mijenjati i brisati proizvode u katalozima restorana
* Baza podataka, sudionik
  + Pohranjuje podatke o svim registriranim korisnicima i njihovim ovlastima
  + Pohranjuje sve podatke o restoranima, narudžbama i rutama

**Opis obrazaca uporabe:**

* **UC1:** *Registracija*
  + **Glavni sudionik:** Anonimni korisnik
  + **Cilj:** Izrada korisničkog računa
  + **Sudionici:** Baza podataka, poslužitelj
  + **Preduvjet:** Valjani e-mail
  + **Rezultat:** Izrađen korisnički račun
  + **Željeni scenarij:**
    1. Korisnik odabire poveznicu za registraciju
    2. Korisnik unosi sve potrebne podatke i potvrđuje svoj unos
    3. Poslužitelj provjerava ispravnost unesenih podataka i postojanje u bazi podataka
    4. Poslužitelj prihvaća podatke i upisuje u bazu podataka podatke o novom korisniku
  + **Mogući drugi scenarij:**
    1. Odabrano već zauzeto korisničko ime
    2. Unesen neispravan e-mail
* **UC2:** *PrijaviSe*
  + **Glavni sudionik:** Anonimni korisnik
  + **Cilj:** Prijava u aplikaciju
  + **Sudionici:** Baza podataka, poslužitelj
  + **Preduvjet:** Postoji korisnički račun
  + **Rezultat:** Korisnik je prijavljen
  + **Željeni scenarij:**

1. Korisnik odabire opciju za prijavu u aplikaciju
2. Korisnik upisuje podatke i lozinku
3. Poslužitelj provjerava ispravnost unesenih podataka.
4. Korisnik se prijavljuje u aplikaciju
   * **Mogući drugi scenarij:**
     1. Korisnik je upisao krive podatke i lozinku

* **UC3:** *DodajProizvodUKošaricu*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Dodavanje proizvoda iz kataloga u košaricu
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Korisnik je prijavljen u aplikaciju
  + **Rezultat:** Proizvod je dodan u košaricu
  + **Željeni scenarij:** 
    1. Prijavljen klijent odabire opciju dodaj proizvod
    2. Unosi se promjena u bazi podataka
    3. U košarici se pojavljuje odabrani proizvod
* **UC4:** *UređivanjeKošarice*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Uređivanje košarice
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Klijent je prijavljen u aplikaciju te ima proizvod u košarici
  + **Rezultat:**  Stanje u košarici je promijenjeno
  + **Željeni scenarij:** 
    1. Klijent odabire opciju uređivanja košarice
    2. Klijent mijenja količinu proizvoda i/ili briše proizvod iz košarice
    3. Promjene se spremaju u bazu podataka
* **UC5:** *PotvrdiNarudžbu*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Potvrditi narudžbu iz košarice
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Klijent je prijavljen u aplikaciju, ima proizvode u košarici te je označio lokaciju dostave
  + **Rezultat:** Obavljena narudžba
  + **Željeni scenarij:** 
    1. Klijent odabire opciju potvrdi narudžbu
    2. Klijent upisuje lokaciju na koju želi da se narudžba dostavi
    3. Narudžba se unosi u bazu podataka
  + **Mogući drugi scenarij:**

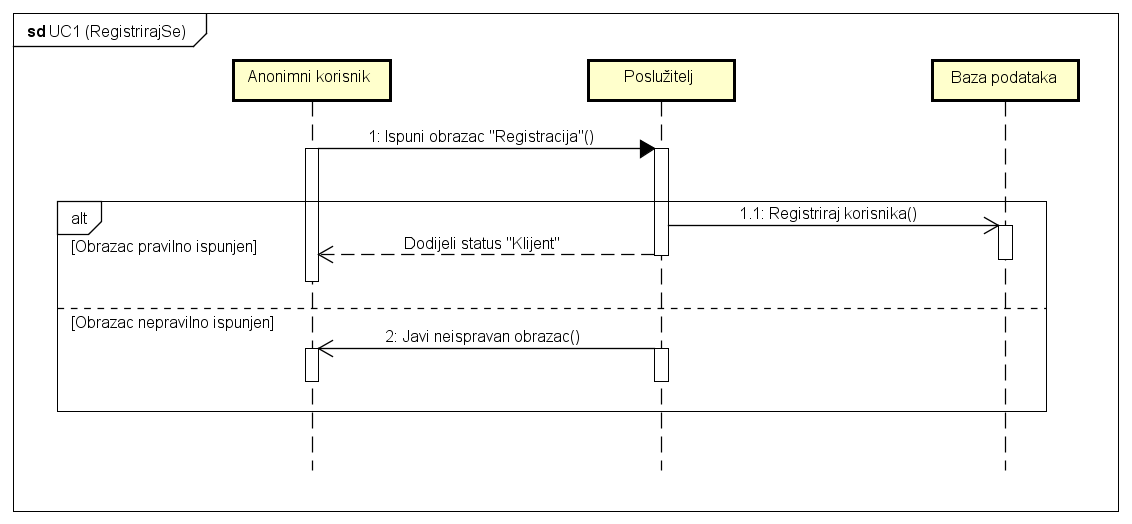
1. Korisnik nema proizvoda u košarici
2. Korisnik nije upisao lokaciju

* **UC6:** *PratiPoziciju*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Pratiti poziciju dodijeljenog dostavljača
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Klijent je prijavljen u aplikaciju i ima potvrđenu narudžbu
  + **Rezultat:** Klijent može saznati trenutnu poziciju narudžbe
  + **Željeni scenarij:** 
    1. Klijent odabire opciju prati poziciju
    2. Promjena se unosi u bazu podataka
    3. Klijent uspješno prati poziciju narudžbe
* **UC7:** *PredložiRestoran*
  + **Glavni sudionik:** Klijent
  + **Cilj:** Predložiti restoran
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Klijent je prijavljen u aplikaciju
  + **Rezultat:** Prijedlog restorana proslijeđen administratoru
  + **Željeni scenarij:**
    1. Korisnik odabire opciju predlaganje restorana
    2. Korisnik unosi podatka o restoranu
    3. Prijedlog se unosi u bazu podataka
    4. Prijedlog čeka na odobrenje administratora
  + **Mogući drugi scenarij:**
    1. Korisnik nije unio podatke
    2. Obavijesti korisnika da nije unio potrebne podatke
* **UC8:** *UređivanjeMenija*
  + **Glavni sudionik:** Vlasnik restorana, Administrator
  + **Cilj:** Unijeti promjene u meni
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjet:** Vlasnik restorana prijavljen u aplikaciji
  + **Rezultat:** Meni je promijenjen
  + **Željeni scenarij:** 
    1. Vlasnik odabire opciju uređivanja menija
    2. Vlasnik unosi promjene u meni
    3. Promijenjeni meni se sprema u bazu podataka
* **UC9:** *PregledPopisaNarudžbi*
  + **Glavni sudionik:** Vlasnik restorana
  + **Cilj:** U aplikaciji može pregledati popis svih trenutno aktivnih narudžbi(u postupku izrade i dostave) i narudžbi koje su se dostavile tijekom prošla 2h
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjet:** Vlasnik restorana prijavljen u aplikaciji i postoje potvrđene narudžbe
  + **Rezultat:** Omogućen pregled popisa svih trenutno aktivnih narudžbi i narudžbi koje su se dostavile tijekom prošla 2h
  + **Željeni scenarij:** 
    1. Vlasnik odabire opciju pregleda popisa s narudžbama i dostavljačima
    2. Narudžbe se dohvaćaju iz baze podataka
    3. Vlasnik prati obavljanje narudžbi
  + **Mogući drugi scenarij:** 
    1. Nema narudžbi
    2. Dojavi porukom „Nema aktivnih narudžbi niti narudžbi tijekom prošla 2h“
* **UC10:** *PostaviOznakuGotov*
  + **Glavni sudionik:** Dostavljač
  + **Cilj:** Dostavljač nakon završetka nekog zadatka označava taj zadatak kao gotov
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti:** Dostavljač je prijavljen u aplikaciju i prihvatio je izvršenje određene narudžbe
  + **Rezultat:** Dostavljač označio izvršenje zadataka te prelazi na iduću narudžbu
  + **Željeni scenarij:**
    1. Dostavljač odabire opciju postavi oznaku gotovo
    2. Promjena se upisuje u bazu podataka
* **UC11:** *PristupiKarti*
  + **Glavni sudionik:** Dostavljač
  + **Cilj:** Pristupiti karti i saznati pozicije zadanih zadataka
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjeti**:Dostavljač je prijavljen u aplikaciju i ima dodijeljene narudžbe
  + **Rezultat:** Informacije o pozicijama, rutama i redoslijedu dodijeljenih zadataka se iscrtavaju na karti
  + **Željeni scenarij:**
    1. Dostavljač odabire opciju pristupi karti
    2. Dostavljaču se prikazuje karta i informacije o narudžbama
* **UC12:** *IdiNaNovuLokaciju*
  + **Glavni sudionik:** Dispečer
  + **Cilj:** Dostavljač dolazi na novu lokaciju
  + **Sudionici:** Dostavljač, baza podataka, poslužitelj
  + **Preduvjet:** Lokacija postoji
  + **Rezultat:** Dolazak dostavljača na određenu lokaciju
  + **Željeni scenarij:**
    1. Dispečer odabire opciju slanja dostavljača
    2. Dispečer odabire slobodnog dostavljača
    3. Dispečer upisuje lokaciju gdje šalje dostavljača
    4. Dispečer potvrđuje upisano
    5. Dostavljač dolazi na određenu lokaciju
* **UC13:** *DodjelaZadatka*
  + **Glavni sudionik:** Dispečer
  + **Cilj:** Dodjeljivanje zadatka dostavljaču
  + **Sudionici:** Baza podataka
  + **Preduvjet:** nema
  + **Rezultat:** Dostavljač dobio zadatak
  + **Željeni scenarij:**
    1. Dispečer odabire opciju dodijele zadatka dostavljačima
    2. Dispečer odabire dostavljača i zadatak za njega
    3. Dispečer potvrđuje svoj unos
    4. U bazu podataka se sprema zadani zadatak
    5. Dostavljač dobiva svoj zadatak
* **UC14:** *NeraspoređeneNarudžbe*
  + **Glavni sudionik:** Dispečer
  + **Cilj:** Prikazivanje neraspoređenih narudžbi, trenutne pozicije dostavljača i njihove buduće rute
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Nema
  + **Rezultat:** Dispečeru prikazane neraspoređene narudžbe, pozicije dostavljača i njihove buduće rute
  + **Željeni scenarij:**
    1. Dispečer odabire opciju prikaza neraspoređenih narudžbi
    2. Iz baze podataka, dispečeru se prikazuje karta sa neraspoređenim narudžbama, trenutnim pozicijama dostavljača i njihovim budućim rutama
* **UC15:** *OdobriRestoran*
  + **Glavni sudionik:** Administrator
  + **Cilj:** Odobriti restoran kojeg je predložio klijent
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjet:** Klijent predložio vlastiti restoran
  + **Rezultat:** Restoran odobren od administratora
  + **Željeni scenarij:** 
    1. Administrator odabire opciju prihvaćanja prijedloga
    2. Odluka administratora sprema se u bazu podatak
* **UC16:** *PregledPopisaRegistriranihKorisnika* 
  + **Glavni sudionik:** Administrator
  + **Cilj:** Pregled popisa svih registriranih korisnika i njihovih osobnih podataka
  + **Sudionici:** Poslužitelj, baza podataka
  + **Rezultat:** Omogućen pregled popisa svih korisnika i njihovih osobnih podataka u svakom trenutku
  + **Željeni scenarij:** 
    1. Administrator odabire opciju pregleda popisa
    2. Administratoru se prikazuje popis registriranih korisnika
* **UC17:** *PromjenaRazinePristupa*
  + **Glavni sudionik:** Administrator
  + **Cilj:** Promijeniti razinu pristupa korisnika aplikaciji
  + **Sudionici:** Poslužitelj, baza podataka
  + **Preduvjet:** Korisnik je prijavljen i ima administratorske ovlasti
  + **Rezultat:** Razina pristupa korisnika promijenjena
  + **Željeni scenarij:** 
    1. Administrator odabire opciju promjene razine pristupa korisnika
    2. Administrator unosi promjenu
    3. Promjena se sprema u bazu podataka

**Sekvencijski dijagrami:**

**Obrazac uporabe UC1 (*RegistrirajSe*):**

Anonimni korisnik ispunjuje obrazac za registraciju. Ako je obrazac ispravno ispunjen onda anonimni korisnik postaje klijent. Ako je obrazac nepravilno ispunjen poslužitelj ispisuje prikladnu poruku anonimnom korisniku.



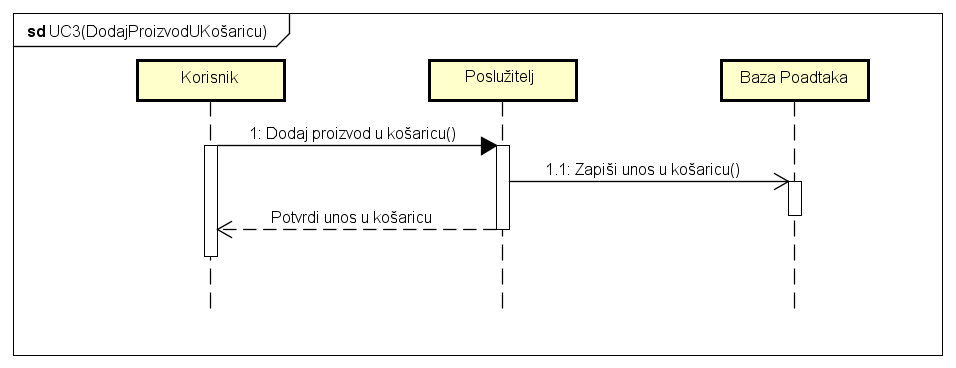
**Obrazac uporabe UC2 (*PrijaviSe*):**

Korisnik ispunjuje obrazac za prijavu. Ako su podaci dani u obrascu ispravni korisniku je dan pristup aplikaciji. Ako podaci nisu ispravni poslužitelj ispisuje prikladnu poruku.



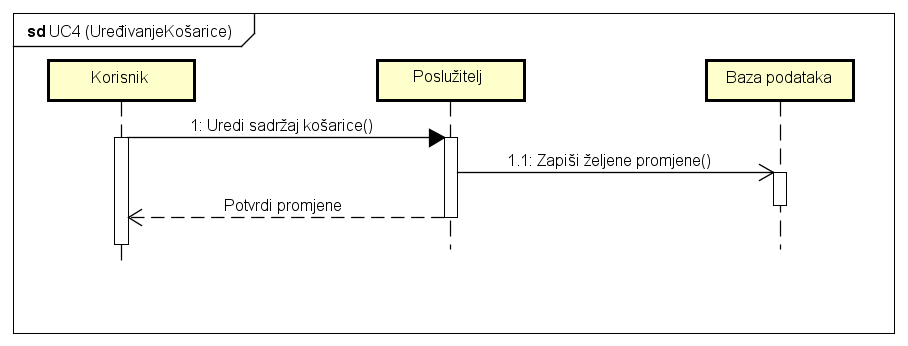
**Obrazac uporabe UC3 (*DodajProizvodUKošaricu*):**

Korisnik šalje sučelju zahtjev za unos proizvoda u košaricu. Poslužitelj javi bazi podataka da zapiše unos u košaricu te pošalje korisniku potvrdu o uspješnom unosu u košaricu.



**Obrazac uporabe UC4 (*UređivanjeKošarice*):**

Korisnik uređuje košaricu (mijenja količinu proizvoda ili uklanja proizvod) preko poslužitelja. Tada poslužitelj zahtjeva od baze podataka da zapiše promjene te pošalje korisniku potvrdu o uspješnoj provedbi promjena.



*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.*