Oblikovanje programske potpore

Ak. god. 2014./2015.

*Sustav restorana*

Dokumentacija, Rev. *<1 ili 2>*

Grupa: *Avengers*

Voditelj: *Vilim Stubičan*

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

Nastavnik: *<Ime i prezime nastavnika>*

Popis članova grupe i zaduženja

*Vilim Stubičan* – razrada arhitekture sustava, osmišljanje i kreiranje baze podataka, opis razreda i dijagram razreda, dijagram objekata, implementacija programskog rješenja, korištene tehnologije i alati

*Oleg Jakovljev* – funkcionalni zahtjevi, implementacija programskog rješenja, upute za korištenje

*Domagoj Korman –* implementacija programskog rješenja, testiranje programskog rješenja, isječci programskog koda za opise temeljnih funkcionalnosti sustava

*Lana Murar –* dnevnik sastajanja, funkcionalni zahtjevi, testiranje programskog rješenja, upute za korištenje

*Nejra Muslić –* ostali zahtjevi, pojmovnik, testiranje programskog rješenja

*Marko Novina –* arhitektura sustava, pojmovnik

*Filip Popić –* ostali zahtjevi, implementacija programskog rješenja, testiranje programskog rješenja

Sadržaj

[1. DNEVNIK PROMJENA DOKUMENTACIJE](#h.gjdgxs)

[2. OPIS PROJEKTNOG ZADATKA](#h.30j0zll)

[3. RJEČNIK POJMOVA](#h.1fob9te)

[4. FUNKCIONALNI ZAHTJEVI](#h.3znysh7)

[5. OSTALI ZAHTJEVI](#h.2et92p0)

[6. ARHITEKTURA I DIZAJN SUSTAVA](#h.tyjcwt)

[**6.1.** **Svrha, opći prioriteti i skica sustava**](#h.3dy6vkm)

Svrha i opći prioriteti sustava

Grafičko sučelje

Datotečni sustav

Baza podataka

[**6.2.** **Dijagram razreda s opisom**](#h.1t3h5sf)

[**6.3.** **Dijagram objekata**](#h.4d34og8)

[**6.4.** **Ostali UML dijagrami**](#h.2s8eyo1)

Komunikacijski dijagram

Dijagram stanja

Dijagram aktivnosti

Dijagram komponenti

[7. IMPLEMENTACIJA I KORISNIČKO SUČELJE](#h.17dp8vu)

[**7.1.** **Dijagram razmještaja**](#h.3rdcrjn)

[**7.2.** **Korištene tehnologije i alati**](#h.26in1rg)

[**7.3.** **Isječak programskog koda vezan za temeljnu funkcionalnost sustava**](#h.lnxbz9)

[**7.4.** **Ispitivanje programskog rješenja**](#h.35nkun2)

[**7.5.** **Upute za instalaciju**](#h.1ksv4uv)

[**7.6.** **Korisničke upute**](#h.44sinio)

[8. ZAKLJUČAK](#h.2jxsxqh)

[9. REFERENCE](#h.z337ya)

[DODATAK A: DNEVNIK SASTAJANJA](#h.3j2qqm3)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Napravljen predložak. | Stubičan | 29.10.2014. |
| 0.2 | Napravljen pojmovnik | Popić | 30.10.2014. |
| 0.3 | Uređen opis projektnog zadatka | Jakovljev | 30.10.2014. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Opis projektnog zadatka

Cilj projekta koji obavlja ova grupa jest implementacija sustava rezervacija i naplate mjesta u restoranima koji omogućuje uvid u ponude restorana te poslovanje istih na dnevnoj bazi. [NUMBER] je glavnih dionika sustava prema čijim će se potrebama i načinima korištenja oblikovati sustav, a oni su: gosti, ugostitelji, dobavljači, posjetitelji web portala i konobari.

Implementirani sustav će koristiti svim korisnicima, prvenstveno gostima u smislu da će im omogućiti potrebne informacije za organiziranje ugodne večeri u jednom od restorana. Također, ugostiteljima će se olakšati poslovanje jer će sve podatke o svojim poslovnicama, njihovim zalihama, rezervacijama i prometom koji se odvija unutar restorana događa imati u jednom sustavu, a također će im omogućiti i pregled te izmjenu menija kao i praćenje stanja zaliha i dobave namirnica. Vlasnici restorana će moći i voditi evidenciju o financijskom stanju ugostiteljskog objekta i prihodima. Biti će moguće i pratiti narudžbe pojedinih menija ili obroka i namirnica potrebnih za pripremu istih. Dobavljačima će sustav pomoći u organizaciji poslovanja jer će na jednoj lokaciji moći pratiti sve nadolazeće narudžbe od ugostitelja. Podaci o aktualnim rezervacijama moraju biti dostupne konobarima kako bi mogao klijente rasporediti po slobodnim stolovima. Sustav ocjenjivanja pojedine lokacije omogućuje recenzentima da iste podatke iskoriste u svojim istraživanjima, ali ujedno i gostima da im omogući lakši odabir za svoju ponude. I registrirani i neregistrirani posjetitelji portala će moći ocijeniti pojedine restorane, no registriranim korisnicima se prati statistika i prikazuju se ocjene dodijeljene pojedinim restoranima. Ocjene restorana će se pokazivati na izborniku restorana uz njihove ponude i opcije za rezervacije mjesta ili menija. Trenutno sada postoji više sustava koji imaju samo neke od navedenih funkcionalnosti, ali niti jedan koji obuhvaća sve navedeno te sustav itekako ima svrhu i ima smisla krenuti u njegovu realizaciju. Ovakav će sustav objediniti sve potrebne funkcije za upravljanje restoranom i omogućiti potencijalnim klijentima lako i pristupačno upoznavanje s ponudama restorana i rezervaciju mjesta. Sustav će također omogućiti i jednostavnu izmjenu ponude restorana na dnevnoj bazi te olakšati upravljanje ugostiteljima i koordinaciju dobavljača.

Gosti, ugostitelji i dobavljači moraju biti registrirani korisnici, čime dobivaju prava na korištenje sustava. Prilikom registracije, korisnik popunjava sljedeće podatke kroz web obrazac:

* Tip računa (gost, ugostitelj, dobavljač)
* Korisničko ime
* Lozinka
* Email adresa

Opcionalno:

* Kontakt broj
* Kreditna kartica preko koje će se vršiti naplata

Još jedna od korisnih funkcionalnosti koje će sustav pružati ugostiteljima je grupiranje pojedinih jela ili jelovnika po vrstama, na primjer stvaranje pojedinih grupa za vegeterijanska jela, salate, alkoholna pića, egzotične koktele ili specijalitete. Osim grupiranja po vrsti, biti će moguće i stvoriti posebne kategorije jela poput specijaliteta, jela s popustom ili brzih jela. Uz pojedine stavke biti će priloženi i cjenici kako bi korisnici imali funkcionalan i privlačan pregled ponude. Fizičke osobe ili osobe koje u ime vlastite tvrtke nude navedene sadržaje kroz sustav nazvat ćemo jednim imenom Ugostiteljima. Ugostitelji mogu za svaku svoju poslovnicu unijeti ponudu jela i menija koristeći *Content Management System (dalje, CMS)* . Kada ugostitelj postavi svoju poslovnicu aktivnom, to znači da pruža uslugu rezervacije mjesta u istoj poslovnici. Korisnik koji se odluči za rezervaciju mora ispuniti sljedeće podatke:

* Datum i vrijeme rezervacije,
* Broj sjedećih mjesta,
* Način naručivanja hrane
  + Online : narudžba se kreira na stranici
  + U restoranu : narudžba se obavlja prilikom dolaska u restoran

Opcionalno:

* Napomena vezana uz rezervaciju

Ukoliko ugostitelj želi oglašavati svoju poslovnicu, dužan je putem CMS-a unijeti sljedeće podatke:

* Jedinstveno ime poslovnice,
* Adresa,
* Kontakt,
* Kapacitet
  + Web obrazac za više tipova stolova (2, 4, 6… sjedećih mjesta i broj istih)
* Radno vrijeme

Radi lakše organizacije menija, Ugostitelj ima mogućnost kreiranja kategorija jela za svaku poslovnicu pojedinačno. Sve kategorije zahtijevaju sljedeće podatke:

* Ime kategorije
* Stanje (aktivna ili neaktivna)

Za svaku poslovnicu Ugostitelj može unositi neograničen broj jela u ponudi. Svako dodavanje jela zahtjeva:

* Ime jela
* Cijena
* Skup namirnica
  + Namirnica
  + Normativ odabrane namirnice
* Kategorija
* Dostupno

Ugostitelj ima ugrađenu podršku za obradu rezervacija. Kada korisnik završi rezervaciju, dobije unikatni kod koji je identifikator njegove rezervacije. S tim kodom u restoranu potvrđuje rezervaciju. Ugostitelj ima mogućnost obrade rezervacije. Prilikom toga, mijenja sljedeće stavke rezervacije:

* Stanje rezervacije (rezervirano, plaćeno, otkazano)
* Način plaćanja

Za narudžbe koje se obavljaju u restoranu:

* Meni

Svaki ugostitelj ima uvid u trenutno stanje zaliha u svakoj poslovnici. U skladu s time, može kreirati narudžbu namirnica od dobavljača. Prilikom postavljanja narudžbe, Ugostitelj mora unijeti sljedeće podatke:

* Za svaku namirnicu
  + Količina
  + Dobavljač

Dobavljači imaju uvid u narudžbe koje su zaprimili od Ugostitelja. Za mogućnost primitka narudžbe, svaki dobavljač mora odabrati proizvođača određene namirnice.

Kako bi Ugostitelji znali s kime započinju poslovni odnos, imaju uvid u listu svih Dobavljača koja sadrži osnovne podatke o istima.

Svi korisnici imaju mogućnost postavljanja upita pojedinim Ugostiteljima preko poslovne jedinice koju Ugostitelj oglašava. Tim putem korisnici mogu saznati dodatne informacije o poslovnici. Pregled svih restorana će biti pretraživ po ključnim vrijednostima.

Sustav podržava i ocjenjivanje pojedinih poslovnica. Ocijeniti restoran može bilo koji registrirani korisnik te ima mogućnost ostavljanja pismenog komentara. Sustav ocjenjivanja redovito računa ocjenu restorana te prikazuje istu na pregledu poslovnice.

Kako bi sustav oblikovali i razvili što kvalitetnije, izabrana je objektno orijentirana arhitektura zbog odlika stabilnosti, modularnosti i pouzdanosti koje se vrlo dobro uklapaju u sve zahtjeve sustava. Sustav ima dvije osnovne zadaće: klijent postavlja upit, a poslužitelj omogućava izvršavanje upita. [MODEL = KLIJENT-POSLUŽITELJ ?]

Centralni dio sustava je svakako baza podataka koja će ujediniti funkcionalnosti raznih dionika i povezati ih u dinamičku mrežu u stvarnom vremenu. Velike količine podataka o normativima jela, rezervacijama, narudžbama, ponudama restorana, itd., potrebno je spremiti negdje. U tu svrhu, sustav koristi relacijsku bazu podataka čija je struktura organizirana na način da se poklapa s predodžbom akcija koje se događaju unutar poslovanja restorana. Tip baze podataka koju smo odabrali je MySql.

Web stranica će biti implementirana pomoću aktualnih web tehnologija – HTML, CSS, JavaScript, jQuery, PHP, MySQL. Ključ uspješne izrade funkcionalnog proizvoda je jednostavno i pristupačno sučelje koje će korisnicima omogućiti lako ostvarivanje željenih akcija i restoranskom osoblju jednostavan pregled potrebnih funkcija. PHP je jedan od jednostavnijih tehnologija kreirana upravo za razvoj web aplikacije, dok je AngularJS vrlo moćan alat za laganu implementaciju objektno orijentiranog pristupa u web aplikaciju. Bitno svojstvo kod kojeg će u implementaciji trebati posvetiti pažnju je sigurnost podataka i netransparentnost zaštićenih korisničkih pojedinosti. Ovako ostvaren sustav će uvelike pojednostaviti vođenje restorana i objediniti mnoge zadaće koje bi inače morao obavljati skup aplikacija. Time će se korisnicima sustava efektivno smanjiti troškovi poslovanja i kompleksnost korištenja programske potpore u poslovanju ugostiteljskih objekata a klijentima omogućiti jednostavan i bezbolan način pregleda ponuda, rezervacije mjesta kao i uvid u ocjene pojedinih restorana i trenutne ponude ili popuste.

Arhitektura sustava će se bazirati na već spomenutom objektno orijentiranom pristupu, koristeći *MVC (Model – View – Controller)* oblikovni obrazac kojim se odvaja prezentacijski, logički i podatkovni dio aplikacije.

AngularJS MVC i baza podataka će biti detaljnije opisani u nastavku.

Jedna od prvih ključnih točaka u implementaciji sustava su ostvarivanje osnovnih funkcija koje sustav pruža gostima restorana i dizajn korisničkog sučelja. Funkcionalnost baze podataka koja leži u centru sustava je još jedna ključna točka projekta pošto se na bazu nadovezuju sve funkcije koje dionici mogu obavljati. Ključne točke projekta će se također odnositi na dodavanje mogućnosti ugostitelja i sučelja za konobare.

# Pojmovnik

**AngularJS**: Strukturni radni okvir (eng. framework) za dinamičke web aplikacije. Izvrsno organizira JavaScript kod te omogućuje izradu brzih web stranica koji se u paru sa jQuery-ijem izvrsno nadopunjavaju. Jednostavan je za učenje te omogućava sintaksna proširenja HTML-a doprinoseći lakšoj čitljivosti koda i povećavajući samu učinkovitost rješenja.

**CSS**: (eng. CSS - Cascading Style Sheets) Stilski jezik koji se rabi za opis reprezentacije dokumenta napisanog pomoću HTML jezika. Oblikuje izgled i raspored HTML-a bez potrebe za prikazivanjem njegovog sadržaja.

**HTML**: (eng. HTML - HyperText Markup Language) Prezentacijski jezik za izradu web stranica. Svaki hipertekst dokument sadržava sadržaj te hiperveze na ostale sadržaje koje web preglednik jednostavno dobavlja. Osnovna građa svake HTML web stranice su znakovi (eng. tags) koji opisuju kako i gdje će se nešto prikazati u web pregledniku poštujući uređenu hijerarhijsku strukturu.

**JavaScript**: Najrašireniji skriptni programski jezik kojeg podržavaju svi poznati preglednici. Služi za razvoj klijent-server Internet aplikacija. JavaScript je jezik interpreter, što znači da se ne prevodi (eng. compiling), već se odmah slijedno izvršava. Omogućava izravno kodiranje unutar HTML stranica. Osim brzine i preglednijeg web sučelja, osvježavanje podataka bez potrebe za osvježavanjem cijele stranice (Ajax) te mogućnost trenutnog pretraživanja podataka su neke od značajnih prednosti JavaScript-a.

**jQuery**: Brza, jednostavna ali sadržajno bogata JavaScript biblioteka. Omogućava jednostavno korištenje i manipulaciju nad HTML dokumentima te pojednostavljuje mnoge usluge poput upravljanja događajima (eng. event handling), upravljanje animacijama te korištenje Ajax-a. Unaprijed pripremljenom programskom potporom značajno se olakšava implementacija različitih servisa koji rade neovisno o web pregledniku.

**MySQL**: Besplatan sustav za upravljanje bazom podataka. MySQL baze podataka su relacijske baze podataka, iznimno brze i pogodne za skladištenje velikih količina podataka, zbog čega se često u praksi i koriste. MySQL ima podršku za rad sa brojnim programskim jezicima poput PHP-a koji je isto javno dostupan, što ih čini široko dostupnima.

**PHP**: PHP je jedan od najpopularnijih i najmoćnijih skriptnih jezika javne upotrebe za dinamičko generiranje HTML koda, što znači da se pomoću PHP-a može kreirati HTML stranica na serveru prije nego što je ona poslana samom klijentu sa svim pripadajućim sadržajem. PHP je orijentiran na poslužitelja (eng. server) te kada poslužitelj primi zahtjev, pomoću PHP dokumenta generira HTML kod koji šalje korisniku. Svojom sintaksom sličan je mnogim drugim jezicima (na primjer C-u).

# Funkcionalni zahtjevi

**Dionici ovog sustava su :**

* Ugostitelj
* Dobavljač
* Gost
* Posjetitelj web portala
* Konobar

**Aktori i njihovi funkcionalni zahtjevi:**

* Ugostitelj, inicijator
* Oglašava svoju ponudu na web koja sadrži naziv restorana, kratki opis, jelovnik te kapacitet restorana.
* U svakom trenutku ima mogućnost uvida u trenutno stanje zaliha namirnica.
* Kod narudžbe i pripreme hrane motri stanje zaliha, te po potrebi kontaktira dobavljača.
* U ponudi su opisani jelo i piće grupirani po vrstama, te kompletan cjenik.
* Može definirati specifične kategorije jela (“gableci” po danima, posebne ponude, specijaliteti kuće i sl.)
* Dobavljač, inicijator
* Vodi evidenciju restorana i ugostiteljskih objekata te proizvođača hrane tako što ima uvid u stanje zaliha po dobavi i isporuci hrane.
* Gost, inicijator
* Rezervira mjesto, jelovnik i stol.
* Konobar, sudionik
* Pri dolasku gostiju na “pametnom uređaju” provjerava stanje rezervacija i narudžbi te ih smješta za slobodne stolove.
* Ako gost naručuje u restoranu, konobar to evidentira čitanjem bar koda.
* Posjetitelj web portala, inicijator
* Brojčano ocjenjuju i komentiraju restorane ( i registrirani i neregistrirani posjetitelji )
* Registrirani:
* Može samostalno napraviti rezervaciju mjesta.
* Mora dati svoje osobne podatke.
* Opcionalno može ostaviti trajno zabilježene i ostale kontakt podatke kao što je broj telefona i mobitela te podatke o svojim kreditnim karticama kojima se vrši naplata.
* Nezavisna “gastro-inspekcija”, inicijator
* Na svome web portalu recenzira restoran i njegovu ponudu, klasificira ga te posjetiteljima otvara prostor gdje oni mogu ocijeniti i komentirati ponudu restorana.
  1. ***Opis obrazaca uporabe ???***
* UC1 – RezervacijaINaplata?
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC2 – PraćenjeStanjaZaliha
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC3 – PostavljanjeNarudžbi
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC4 – ProcesiranjeNarudžbe
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC5 – UrediProfil
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC6 – RegistracijaILogin?
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC7 – PrikazRestorana
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**
* UC8 – OcjenjivanjeRestorana
* **Glavni sudionik:**
* **Cilj:**
* **Sudionici:**
* **Preduvjeti:**
* **Željeni scenarij:**
* **Mogući drugi scenariji:**

# 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.*

*Crtanje pojedinih dijagrama (dijagram komponenti, dijagram objekata) nije moguće kroz ArgoUML. Studenti se za crtanje dijagrama mogu koristiti bilo kojim dostupnim alatom.*

*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 4) 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 5 A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika..*

# Zaključak

*U ovom poglavlju potrebno je navesti koji je status implementacije u odnosu na postavljene ciljeve, 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 smjernice za daljnji rad kad bi se nastavilo s projektom.*

# Reference

*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. Tigris.org Open Source Software Engineering Tools, <http://readyset.tigris.org/>
6. Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
7. I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
8. Concepts: Requirements, <http://www.upedu.org/upedu/process/gcncpt/co_req.htm>
9. UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
10. Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
11. ArgoUML, <http://argouml.tigris.org/>

# Dodatak A: Dnevnik sastajanja

*U ovom dijelu 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.*