Oblikovanje programske potpore

Ak. god. 2015./2016.

Sustav za rezervaciju soba i apartmana u turitičkom naselju

Dokumentacija, Rev. *1*

Grupa: *Mihajlo*

Voditelj: *Damjan Miko*

Datum predaje: *19.11.2015.*

Nastavnik: *Dr.sc Miljenko Krhen*

Sadržaj

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

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

[3. Pojmovnik 8](#_Toc431806047)

[4. Funkcionalni zahtjevi 9](#_Toc431806048)

[5. Ostali zahtjevi 10](#_Toc431806049)

[6. Arhitektura i dizajn sustava 11](#_Toc431806050)

[6.1. Svrha, opći prioriteti i skica sustava 11](#_Toc431806051)

[6.2. Dijagram razreda s opisom 18](#_Toc431806052)

[6.3. Dijagram objekata 19](#_Toc431806053)

[6.4. Ostali UML dijagrami 20](#_Toc431806054)

[7. Implementacija i korisničko sučelje 21](#_Toc431806055)

[7.1. Dijagram razmještaja 21](#_Toc431806056)

[7.2. Korištene tehnologije i alati 22](#_Toc431806057)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 23](#_Toc431806058)

[7.4. Ispitivanje programskog rješenja 24](#_Toc431806059)

[7.5. Upute za instalaciju 25](#_Toc431806060)

[7.6. Korisničke upute 26](#_Toc431806061)

[8. Zaključak i budući rad 27](#_Toc431806062)

[9. Popis literature 28](#_Toc431806063)

[Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 29](#_Toc431806064)

[Dodatak B: Dnevnik sastajanja 30](#_Toc431806065)

[Dodatak C: Prikaz aktivnosti grupe 32](#_Toc431806066)

[Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 35](#_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 | Napravljen predložak | Ćirić | 22.10.2015. |
| 0.11 | Napisan opis zadatka u detalje | Ćirić | 23.10.2015. |
| 0.3 | Napisan opis obrazaca uporabe | Franjić | 28.10.2015. |
| 0.4 | Dodani Use Case dijagrami | Grubišić | 1.11.2015. |
| 0.5 | Dodani nefunkcionalni zahtjevi | Smoljanić | 5.11.2015. |
| 0.6 | Dodan opis MVC modela | Miko | 8.11.2015. |
| 0.7 | Osmišljena baza podatak | Šušnjara | 12.11.2015. |
| 0.8 | Dodan dijagram razreda | Raguž | 17.11.2015. |
| 0.81 | Dodan dijagram objekta | Raguž | 18.11.2015. |
| 0.9 | Dodani sekvencijski dijagrami | Grubišić | 18.11.2015. |
| 0.91 | Napisani zaključak i dodaci | Miko | 19.11.2015 |
| **1.0** | **Verzija samo s bitnim dijelovima za 1. ciklus** | **Ivošević** | **19.11.2015.** |

# Opis projektnog zadatka

Cilj ovog projekta je implementacija informacijskog sustava koji će biti dostupan na web stranici turističkog naselja “Kod nas je najljepše”. Sustav će pružati mogućnost pregleda zauzeća i rezervacije svih smještajnih jedinica. Za svaku jedinicu bit će moguće vidjeti detaljan opis i fotografije.

Implementirani sustav će koristiti prvenstveno turistima koji će na taj način moći unaprijed vidjeti ponudu smještaja, doznati potrebne informacije te rezervirati željenu smještajnu jedinicu. Također će biti od koristi vlasniku turističkog naselja jer će na taj način moći prezentirati ponudu smještaja. Osim toga, pomoći će i u poslovanju jer će sva evidencija vezana za smještajnu jedinicu, registrirane korisnike te izvještaje o iznajmljivanju biti pohranjena na jednom mjestu.

Turističko naselje „Kod nas je najljepše“ ima na raspolaganju 4 objekta s ukupno 16 apartmana kapaciteta 2-4 osobe, 6 apartmana s kapacitetom 6-8 osoba i 8 soba za 2-3 osobe s tušem i WC-om. Pola navedenih smještajnih kapaciteta ima pogled na more, a pola ima pogled na park i borovu šumu. Kako bi turisti bili što zadovoljniji omogućene su im i dodatne usluge kao što su rezervirano parkirno mjesto, bežični pristup internetu te satelitski TV prijamnik. Svi smještajni kapaciteti otvoreni su od 1. svibnja do 30. rujna u godini što je idealno za obiteljski turizam, umirovljenike, te odmor s obzirom da je to kraj ljetne sezone. Nažalost, kućni ljubimci nisu dozvoljeni radi lakšeg održavanja i higijene.

Informacijski sustav ima četiri vrste korisnika:

-vlasnik sustava (1 vlasnik)

-administrator (maksimalno 3 administratora)

-registrirani korisnik (neogranič broj)

-neregistrirani korisnik.

Vlasnik sustava nakon inicijalnog puštanja sustava u produkciju upisuje podatke o svim smještajnim jedinicama. Svaka smještajna jedinica se nalazi u određenom objektu, ima odgovarajuću oznaku, kapacitet, stranu/pogled i posebni odvojeni sadržaj u kojem se nalazi detaljan opis i fotografije smještajne jedinice. Podatke o vlasniku upisuje informatička kuća koja je radila sustav.

Nakon prvog spajanja na sustav, vlasnik mora upisati svoju adresu elektroničke pošte i broj telefona. Vlasnik sustava definira i administratore sustava s njihovim ovlastima. Podatke o administratorima sustava može upisivati i mijenjati samo vlasnik sustava.

Administrator sustava može po zahtjevu vlasnika, ali i posebnom pisanom zahtjevu korisnika mijenjati datume rezervacije i smještajne jedinice registriranih korisnika, kao i njihove dodatne usluge. Nakon prve rezervacije registriranog korisnika, administrator u roku od 3 dana nakon rezervacije, predmetnu rezervaciju mora potvrditi ili poništiti.

Registrirani korisnik može odabrati:

-željenu smještajnu jedinicu

-dane za koje ju želi rezervirati

-broj odraslih osoba koje će biti u jedinici

-broj djece koja će biti u jedinici

-dob djece (0-1 / 2-7 / 8-14 godina)

-dodatne usluge

Prilikom registracije korisnik mora unijeti:

-ime

-prezime

-adresu(ulicu, kućni broj, grad i zemlja)

-adresu elektroničke pošte

-broj telefona(nije obavezno)

Neregistrirani korisnik može pregledavatisadržaj web stranice te tako dobiti podatke o smještajnim objektima i jedinicama, kao i njihovo zauzeće ili raspoloživost u danima u kojima je turističko naselje otvoreno. Za rezervaciju smještajnih jedinica potrebna je registracija.

Nakon što se korisnik registrira, na njegovu adresu elektroničke pošte se šalje pozdravna poruka i traži se potvrda registracije. Nakon potvrde registracije, korisniku se na adresu elektroničke pošte šalju pristupni podaci s kojima se korisnik prijavljuje u sustav.

Prilikom rezervacije smještajnih jedinica sustav mora paziti na nemogućnost preklapanja termina, kao i raditi optimizaciju rezervacija. To znači da mora prilikom rezervacije paziti da se rezervacije nastavljaju jedna na drugu s najmanjim mogućim brojem dana kada je neka smještajna jedinica prazna. Tako bi se ostvario maksimalni profit turističkog naselja.

Nakon postupka rezervacije smještajne jedinice, registriranom korisniku se šalje poruka elektroničkom poštom o primitku rezervacije i informacijom da će ga u roku od tri dana kontaktirati predstavnik turističkog naselja.

Sustav omogućuje vlasniku pregled zauzeća svih smještajnih jedinica po danima zauzeća, pregled i rangiranje gradova i zemalja iz kojih dolaze gosti i posebne usluge koje su najviše tražene, pregled zauzeća jedinica po kalendarskom razdoblju, kao i ovisnost zahtjeva na posebne usluge o broju gostiju i zemlji iz koje oni dolaze.Na taj način bi se mogla unaprijediti ponuda turističkog naselja, a te podatke mogla bi koristiti i lokalna turistička zajednica radi vođenja statistike o turistima i njihovim navikama, te željama.

Sustav mora omogućiti istovremeni rad vlasnika sustava, svih administratora i neograničenog broja registriranih korisnika. Prilikom rada vlasnik i administratori sustava moraju moći vidjeti broj i imena trenutno aktivnih drugih administratora i broj trenutno aktivnih registriranih korisnika.

Skup korisnika koji bi mogli biti zainteresirani osim turističkog naselja „Kod nas je najljepše“ su razna turistička mjesta s turističkim zajednicama te policijska uprava. Na taj način bi vrlo jednostavno i učinkovito mogli prikupljati statistiku o turistima i voditi evidenciju, te rukovati podacima.

Implementirani informacijski sustav bi trebao imati i mogućnost. Uz male izmjene u bazi podataka, te implementacije novih zahtjeva od korisnika (ukoliko oni postoje) sustav bi se mogao primijeniti i kod ostalih turističkih naselja koja traže takvu uslugu. Prilikom implementacije bi se trebalo paziti da kod bude što apstraktniji, te bi se na taj način omogućile jednostavne promjene i lakše održavanje.

Za što kvalitetniji razvoj i oblikovanje sustav odabrali smo objektno usmjerenu arhitekturu čije su odlike jednostavnije održavanje sustava, stabilnost i pouzdanost. S obzirom na to, za implementaciju je korišten MVC (Model-View-Controller) oblikovni obrazac koji razdvaja slojeve aplikacije na podatkovni, logički te prezentacijski dio.

# Pojmovnik

**MVC (Model-View-Controller):** vrsta je arhitekture namijenjena izoliranju programske logike (engl. business logic) od ulaza (engl. input) i prezentacijskog djela, omogućava nezavisan razvoj, ispitivanje i održavanje odreñene aplikacije

**SQL:** Strukturirani upitni jezik za upravljanje bazom podataka. Omogućava modificiranje podataka, aritmetičke operacije s podacima i postavljanje različitih upita. SQL je standardiziran preko standarda ANSI i ISO.

**Java:** Objektno orijentirani programski jezik, koji se može iskoristiti za izradu web aplikacija.

**UML (Unified/Universal Modeling Language):** standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

**Eclipse:** je programska razvojna okolina (IDE) pisana u Javi, a može se koristiti za razvoj aplikacija u raznim programskim jezicima kao što su Java, Ada, C, C++, COBOL, Perl, PHP, Python, [R](https://hr.wikipedia.org/w/index.php?title=R_(programski_jezik)&action=edit&redlink=1), Ruby (uključujući [Ruby on Rails](https://hr.wikipedia.org/w/index.php?title=Ruby_on_Rails&action=edit&redlink=1) okolinu), [Scala](https://hr.wikipedia.org/w/index.php?title=Scala_(programski_jezik)&action=edit&redlink=1), [Clojure](https://hr.wikipedia.org/w/index.php?title=Clojure&action=edit&redlink=1) i Scheme. Isto tako, može se koristiti za razvoj dijelova aplikacije [Mathematica](https://hr.wikipedia.org/w/index.php?title=Mathematica&action=edit&redlink=1). Razvojna okolina (IDE) često se naziva Eclipse ADT (Ada Development Toolkit) za Adu, Eclipse CDT za [C/C++](https://hr.wikipedia.org/w/index.php?title=C/C%2B%2B_(programski_jezik)&action=edit&redlink=1), Eclipse JDT za Javu i Eclipse PDT za PHP.

**MySQL:** besplatan, open source sustav za upravljanje bazom podataka. Uz PostgreSQL MySQL je čest izbor baze za projekte otvorenog koda, te se distribuira kao sastavni dio serverskih Linux distribucija, no također postoje inačice i za ostale operacijske sustave poput Mac OS-a, Windowse itd.

**Hibernate:** objektno/relacijska tehnologija koja povezuje objekte iz Jave s relacijskim tablicama baze podataka.

# Funkcionalni zahtjevi

# Ostali zahtjevi

* Sustav mora omogućiti istovremeni rad vlasnika sustava, svih administratora i neograničenog broja registriranih korisnika.
* Prilikom rada vlasnik i administratori sustava moraju moći vidjeti broj i imena trenutno aktivnih drugih administratora i broj trenutno aktivnih registriranih korisnika.
* Sustav ima jednog vlasnika i do najviše 3 administratora.
* Broj registriranih korisnika je neograničen.
* Sustav treba biti ostvaren tako da omogućuje jednostavno korištenje bez potrebe za dodatnim detaljnim uputama.
* Nepravilno i nepredviđeno ponašanje korisnika unutar korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava.
* Stabilnost rada sustava ne bi trebala biti narušena nekritičnim iznimkama i pogreškama.
* Nadogradnje sustava ne smiju narušiti postojeću funkcionalnost sustava.
* Pretraživanje baze podataka ne bi trebalo trajati više od 5 sekundi.

# Arhitektura i dizajn sustava

## Svrha, opći prioriteti i skica sustava

Izbor odgovarajuće arhitekture programske potpore bitan je korak u samom oblikovanju sustava jer ona predstavlja most između zahtjeva koji su postavljeni na sustav i implementacije sustava. O izboru arhitekture programske potpore ovisi cijela programska potpora, jer dobra arhitektura znači fleksibilnost sustava, dobru prenosivost, laganu mogućnost nadogradnje i jeftino održavanje.

Budući da je zadatak ovog programskog sustava napraviti aplikaciju za turističku zajednicu, koju će koristiti korisnici radi lakšeg uvida u rezervaciju soba i apartmana, web aplikacija se čini kao optimalan izbor. Za razliku od desktop aplikacije, web aplikacija je puno lakša za korištenje, održavanje i nadogradnju (nije potreban sustav nadogradnje aplikacije). Isto tako, radit će na svakoj platformi i sustavu koji ima standardni web preglednik, čime smanjujemo troškove jer nije potreban razvoj za više platformi. Web aplikacija se brine za komunikaciju izmeñu zahtjeva korisnika i baze podataka koja se nalazi na poslužitelju.

Arhitektura našeg sustava sastoji se od sljedećih podsustava:

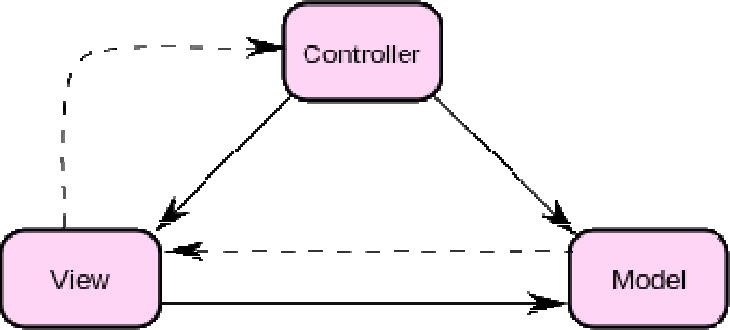
1. Web aplikacija

2. Baza podataka

***Web aplikacija***

Web aplikacija je prozor u naš programski sustav. Ona omogućava prikaz svih podataka iz baze na lijep, strukturiran i siguran način. Pomoću nje se vrši gotovo sva administracija sustava i razmjena podataka.

Za izradu aplikacije smo odabrali MVC (engl *Model – View - Controller*) arhitekturu zato jer omogućava nezavisan razvoj, ispitivanje i održavanje određene aplikacije. Kao jezik u kojem ćemo kodirat izabrali smo javu.



Slika MVC koncept – povezanost dijelova

MVC se sastoji od:

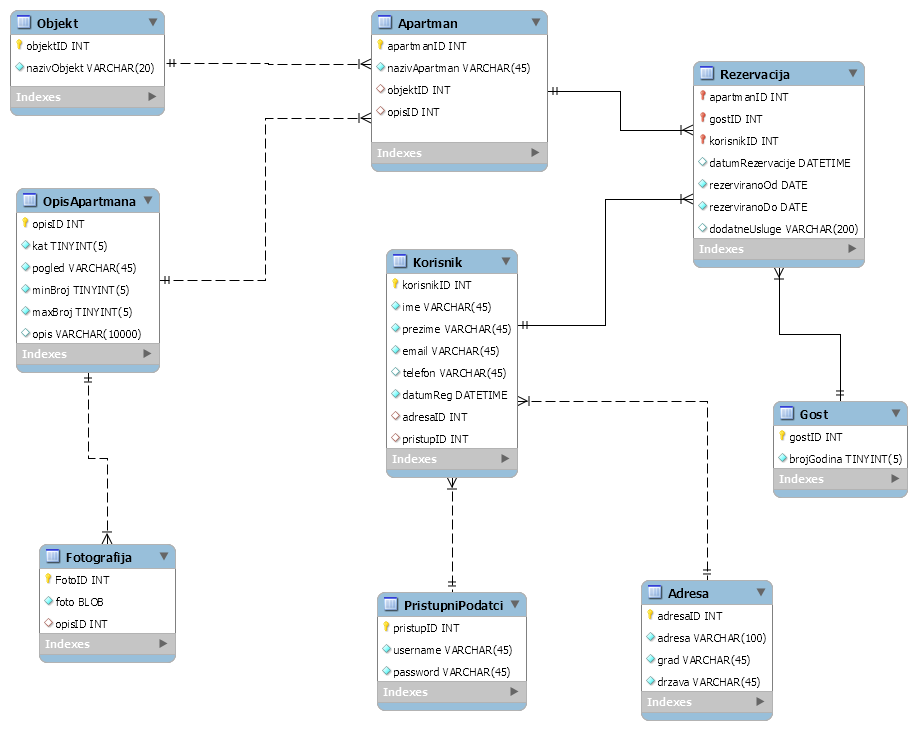
**Model** – predstavlja podatke neke aplikacije. U konkretnoj implementaciji modela(ASP.Net MVC) postoje dvije vrste modela. Jedan prema bazi – kojim se komunicira s bazom, a drugi prema pogledu (View) - koji prenosi podatke do html predloška na kojem se prikazuju

**View** - prikazuje prezentaciju prethodno modeliranih podataka. Konkretno htmlpredložak koji se puni pomoću Razor stroja (engl. *engine*)

**Controller** – služi za upravljanje korisničkim zahtjevima

***Baza podataka***

Bazu podataka koristimo zbog pohranjivanja i međusobnog povezivanja podataka uz isključivanje bespotrebnog stvaranja zalihosti (redundancije). Podaci u bazi podataka su pohranjeni na način neovisan o programima koji ih koriste. Prilikom dodavanja novih podataka, mijenjanja i pretraživanja postojećih podataka primjenjuje se zajednički i kontroliran pristup. Tablice u bazi podataka su strukturirane tako da je omogućena jednostavna nadopuna sa potrebnim informacijama. Također sve tablice svedene su na 3. normalnu formu te je time osigurano da neće biti redundancije podataka. Za izradu baze podataka koristimo MySQL zbog toga što je jednostavan za upravljanje, dovoljno brz, a ujedno i besplatan. Prikaz baze podataka dan je sljedećim ER modelom.



Slika 6.1.1 ER model baze podataka

**OBJEKT**

objektID INT

nazivObjekt VARCHAR(20)

-šifra smještajnog objekta

-naziv smještajnog objekta

Smještajni objekt koji sadrži jednu ili više smještajnih jedinica.

**APARTMAN**

apartmanID INT

nazivApartman VARCHAR(45)

objektID INT

opisID INT

-šifra apartmana -šifra apartmana

-naziv apartmana

-šifra objekta kojem apartman pripada

-šifra opisa apartmana

Apartman koji se može rezervirati, sadrži strani ključ na objekt kojem pripada i na opis u kojem se nalaze detaljne informacije o apartmanu.

**OPISAPARTMANA**

opisID INT

kat TINYINT(5)

pogled VARCHAR(45)

minBroj TINYINT(5)

maxBroj TINYINT(5)

opis VARCHAR(10000)

-šifra opisa apartmana

-kat na kojem se nalazi apartman

-pogled iz apartmana

-minimalan broj ljudi dozvoljen u apartmanu

-maksimalan broj ljudi dozvoljen u apartmanu

-detaljan tekstualni opis apartmana

Pogled iz apartmana može biti na park i borovu šumu ili na more.

**FOTOGRAFIJA**

fotoID INT

foto BLOB

opisID INT

-šifra fotografije

-fotografija

-šifra opisa kojem fotografija pripada

**KORISNIK**

korisnikID INT

ime VARCHAR(45)

prezime VARCHAR(45)

email VARCHAR(45)

telefon VARCHAR(45)

datumReg DATETIME

adresaID INT

pristupID INT

-šifra korisnika

-ime korisnika

-prezime korisnika

-email adresa korisnika

-telefonski broj korisnika

-datum registracije korisnika u sustav

-šifra adrese korisnika

-šifra pristupnih podataka korisnika

Nakon prijave u sustav preko korisničkog imena i lozinke koji se nalaze u tablici „PristupniPodatci“, sustav ovisno o ulozi, dodjeljuje korisniku određene ovlasti. Registrirani korisnik ima najmanje ovlasti, iznad njega je administrator, a najveće ovlasti ima vlasnik sustava.

Telefonski broj korisnik može ali i ne mora unijeti, ostali podatci su obavezni. AdresaID predstavlja strani ključ na relaciju u kojoj su sadržane detaljne informacije o mjestu prebivališta korisnika. PristupID predstavlja strani ključ na relaciju u kojoj se nalazi podatci za pristup sustavu.

**PRISTUPNIPODATCI**

pristupID INT

username VARCHAR(45)

password VARCHAR(45)

-šifra pristupnih podataka

-korisničko ime za pristup sustavu

-lozinka za pristup sustavu

**ADRESA**

adresaID INT

adresa VARCHAR(100)

grad VARCHAR(45)

drzava VARCHAR(45)

-šifra adrese korisnika

-ulica i kućni broj gdje korisnik živi

-grad u kojem korisnik živi

-država u kojoj korisnik živi

Relacija koja sadrži detaljne podatke o mjestu prebivališta korisnika.

**GOST**

gostID INT

brojGodina TINYINT(5)

-šifra gosta u određenom apartmanu

-broj godina gosta

Relacija sadrži samo informaciju o broju godina gosta. Ta informacija je potrebna zbog određivanja cijene smještaja. Djeca se kategoriziraju u skupine (0-1 / 2-7 / 8-14) godina te se njima naplaćuje manja cijena smještaja od osoba iznad 14 godina.

**REZERVACIJA**

apartmanID INT

gostID INT

korisnikID INT

datumRezervacije DATETIME

rezerviranoOd DATE

rezerviranoDo DATE

dodatneUsluge VARCHAR(200)

-šifra apartmana

-šifra gosta

-šifra korisnika

-datum rezervacije

-datum kada se gosti smještaju u apartman

-datum kada gosti odlaze iz apartmana

-dodatne usluge

Ova relacija sadrži sve potrebne podatke vezane uz rezervaciju smještaja. Oni se upisuju kada korisnik putem WEB aplikacije rezervira određenu smještajnu jedinicu (apartman). ApartmanID je strani ključ na apartman koji se rezervira, gostID je strani ključ na gosta koji će boraviti u apartmanu, korisnikID je strani ključ na korisnika sustava koji je obavio rezervaciju. Dodatne usluge mogu biti i izostavljene ako ih korisnik nije zatražio. Dodatnim uslugama pripadaju: rezervirano parkirno mjesto, bežični pristup internetu i satelitski TV prijamnik.

## 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

Zadatak projekta bio je razviti web aplikaciju koja će pomoći vlasnicima objekata da predstave svoje jedinice, a mi im omogućimo lakše evidentiranje rezervacija i da naši korisnici mogu pregledati i rezervirati apartman ili sobu u objektu koji im se najviše sviđa.

U prvoj fazi obavljene su sve formalnosti poput formiranja i upoznavanja tima, iznošenja ideja i vizija te naposljetku, podjele uloga i zadataka. Općenito, prva faza je služila kao priprema za konačno ostvarenje računalnog sustava. Mogli bismo reći da je u njoj napravljen kostur cijelog sustava. Izneseni su glavni funkcijonalni i nefunkcionalni zahtjevi. Nakon toga smo napravili arhitekturu web stranice, tako da smo koristili model MVC. Prvo smo osmislili kako bi trebala izgledati baza podataka, nakon toga smo izgradili kontrloere, čija je uloga da koristeći bazu podataka radi funckije koje su nam potrebne za rad web stranice. Prilikom izrade grafičkog prikaza kontrolera puno su nam pomogli funckionalni zahtjevi koje smo već prije naveli.

Komunikacija među članovima tima, uključujući i voditelja, je bila na visokoj razini. Svi su o svemu bili obaviješteni pravovremeno i prilično jasno. Tim se sastajao skoro svaki tjedan i pričali smo o planu i osmišljali kako bi najbolje bilo osmisliti dokumentaciju i sve važne stvari za ovaj prvi dio. Dobro smo se podijelili u grupe tako da je svaki član imao svoj dio koji je trebao napraviti. Opterečenje po članu je bilo skoro jednako, iako je to teško odrediti. Pošto smo svi združili i ozbiljno shvatili projekt, mislim da nebi trebalo biti problema u daljnoj izradi.

Za izradit nam je još preostala druga vaza, koja je više programskog tipa za razliku od ove prve. Nakon što smo napravili dobru dokumentaciju nebi trebao biti problem u realizaciji web stranice. Među nama su dosta dobri programeri i kad udružimo snage mislim da će to ispast jedna odlična web stranica.

# Popis literature

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

**Ponedjeljak, 19.10.2015., 18:00-20:00**

Prisutni: Juraj Šušnjara, Ivan Ćirić, Ivan Grubišič, Marin Smoljanić, Marko Franjić, Marko Raguž, Damjan Miko

Sadržaj sastanka:

Našli smo se da popričamo o zadatku, raspravljali kako smo ga shvatili, dogovorili se oko programskog jezika u kojem budemo kasnije implementirali zadatak, odabrali smo javu. Također smo se dogovorili kome najviše odgovara koji dio dokumentacije za pisanje.

**Srijeda, 28.10.2015., 16:00-17:00**

Prisutni: Juraj Šušnjara, Ivan Ćirić, Ivan Grubišič, Marin Smoljanić, Marko Franjić, Marko Raguž, Damjan Miko

Sadržaj sastanka:

Dogovorili smo se oko detalje tko će realizirati koji dio zadatka. Podijeljeni poslovi na članoce tima su:

Juraj Šušnjara - osmislit će bazu za projektni zadatak

Ivan Ćirić - opisat će projektni zadatak

Ivan Grubišić - nacrtat će sve potrebne djagrame uporabe i sekvencijske dijagrame

Marin Smoljanić - opisat će sve nefunkcionalne zahtjeve

Marko Franjić - opisat će sve obrasce uporabe

Marko Raguž - izradit će dijagram razreda s opisom i dijagram objekta

Damjan Miko - opisat če MVC arhitekturu i napisat riječnik pojmova

**Utorak, 10.10.2015., 19:00-20:00**

Prisutni: Juraj Šušnjara, Ivan Ćirić, Ivan Grubišič, Marin Smoljanić, Marko Franjić, Marko Raguž, Damjan Miko

Sadržaj sastanka:

Svatko je iznio dio koji je napravio, raspravljali smo o tome, i razjasnili sva pitanja koja smo imali, te se dogovorili oko nekih detalja. Na temelju napravljenog posla dogovorili smo što treba dalje napraviti i kojim redom.

Juraj Šušnjara - promijanit će nekve stvari u bazi podataka

Ivan Ćirić - pomoći će ostalima ako di zapnu

Ivan Grubišić - dovršiti sekvencijske dijagrame

Marin Smoljanić - pomoći će ostalima oko izrade dijagrama

Marko Franjić - ispravit će greške u obrascima upotrebe

Marko Raguž - doradit će započene dijagrame

Damjan Miko - napisat će zaključak

# 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) | | | | | | |
| Damjan Miko | Ivan Ćirić | Ivan Grubišić | Juraj Šušnjara | Marin Smoljanić | Marko Franjić | Marko Raguž |
| **Upravljanje projektom** |  |  |  |  |  |  |  |
| **Opis projektnog zadatka** |  | 100% |  |  |  |  |  |
| **Rječnik pojmova** | 100% |  |  |  |  |  |  |
| **Opis funkcionalnih zahtjeva** |  |  | 50% |  |  | 50% |  |
| **Opis ostalih zahtjeva** |  |  | 50% |  | 50% |  |  |
| **Arhitektura i dizajn sustava** |  | | | | | | |
| Svrha, opći prioriteti i skica sustava | 20% |  |  | 80% |  |  |  |
| Dijagram razreda s opisom |  | | | | | | |
| Dijagram objekata |  |  |  |  |  |  | 100% |
| 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** | 100% |  |  |  |  |  |  |
| **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.*