SVEUČILIŠTE U DUBROVNIKU

ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

Studij Primijenjeno/poslovno računarstvo

**Projekt u 6. semestru**

Akademska godina 20**17**./20**18**.

**Evidencija bike sharing sustava**

Dokumentacija, Rev. 1.00

Student: Veljko Ramljak, 0275047565

Nastavnik: prof. dr. sc. Mario Miličević

Dubrovnik, *15.9.2018*

**Sadržaj**

[1 Opis projektnog zadatka 3](#_Toc503891969)

[2 Rječnik podataka 4](#_Toc503891970)

[3 Projektni zahtjevi 5](#_Toc503891971)

[3.1 Funkcionalni zahtjevi 5](#_Toc503891972)

[3.2 Zahtjevi na sučelja 5](#_Toc503891973)

[3.3 Nefunkcionalni zahtjevi 5](#_Toc503891974)

[4 Arhitektura i dizajn sustava 6](#_Toc503891975)

[4.1 Arhitektura sustava 6](#_Toc503891976)

[4.2 Detaljni opis sustava 6](#_Toc503891977)

[4.3 Model podataka 6](#_Toc503891978)

[5 Implementacija sustava 7](#_Toc503891979)

[5.1 Dijagram razmještaja 7](#_Toc503891980)

[5.2 Korištene tehnologije i alati 7](#_Toc503891981)

[5.3 Izvorni kod programskog rješenja 7](#_Toc503891982)

[5.4 Testiranje programskog rješenja 7](#_Toc503891983)

[5.5 Upute za instalaciju 7](#_Toc503891984)

[5.6 Korisničke upute 7](#_Toc503891985)

[6 Zaključak 8](#_Toc503891986)

[7 Reference 9](#_Toc503891987)

**Zapisi o revizijama**

| **Revizija** | **Datum** | **Opis** |
| --- | --- | --- |
| 1.00 | 15.9.2018 | Prva verzija dokumentacije |
|  |  |  |

# Opis projektnog zadatka

Projektni zadatak je izrada samostojeće (desktop) aplikacije koja omogućava evidenciju i prikaz podataka o iznajmljivanju bicikala kod bike sharing-a. Aplikacija omogućava unos novih bicikala te stanica gdje se mogu preuzeti i ostaviti. Potom prikaz statistike korištenja bicikala.

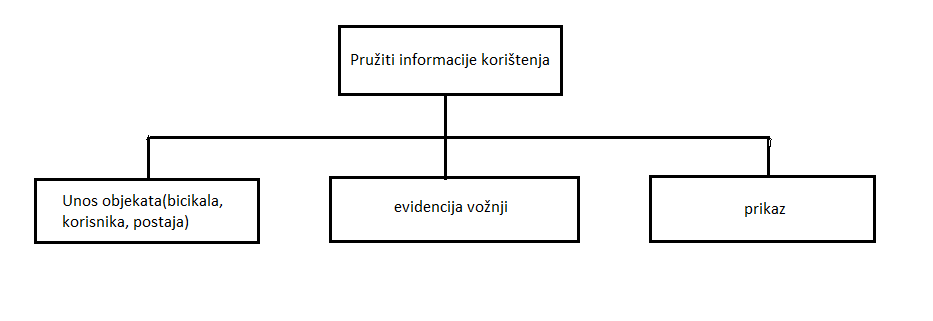
Cilj aplikacije je pokazati koja su bicikla najisplativija te u koje vrijeme je najisplativije iznajmljivati, a kod korisniče strane iskaz popularnosti bicikla te vrijeme vožnji biciklista. Dakle na osnovu evidencije će se pokazati da li je potrebno nabaviti još bicikala popularnog modela, popularne veličine (npr. dječji ili za odrasle). Odnosno prodati bicikl koje nije popularno i koje se ne iznajmljuje.

Namjena je samo za korištenje u bike sharing-u sa malo modifikacije koristila bi svim iznajmljivačima ne samo bicikala nego i ostalih iznajmljivača npr. romobila, auta, motora itd. Te neodlučni korisnici bi znali što je popularno te bi se lakše odabrali.

Bilo bi poželjno da se aplikacija proširi sa mobilnom verzijom za korisnike kako bi znali koliko je udaljena koja stanica te sve imali na dohvat ruke. Gdje u futurističnoj verziji ne bi bilo potrebe za radnicima. A sustav bi bio za iznajmljivanje svega mogućeg te bi se lako mogli registrirati i korisnici koji bi htjeli iznajmljivati.

# Rječnik podataka

Cilj aplikacije je prvenstveno informirati pružatelja usluga iznajmljivanja. Tok procesa je unos bicikala, terminala te korisnika u sustav te potom evidencija vožnji koje su opisane spomenutim objektima i vremenom početka vožnje.



Slika Domena poslovnog procesa

Riječ **terminal** se koristi za stanicu(postaju) gdje se mogu iznajmiti ili ostaviti bicikli.

# Projektni zahtjevi

## Funkcionalni zahtjevi

Jedini sudionici su radnici koji iznajmljuju bicikala:

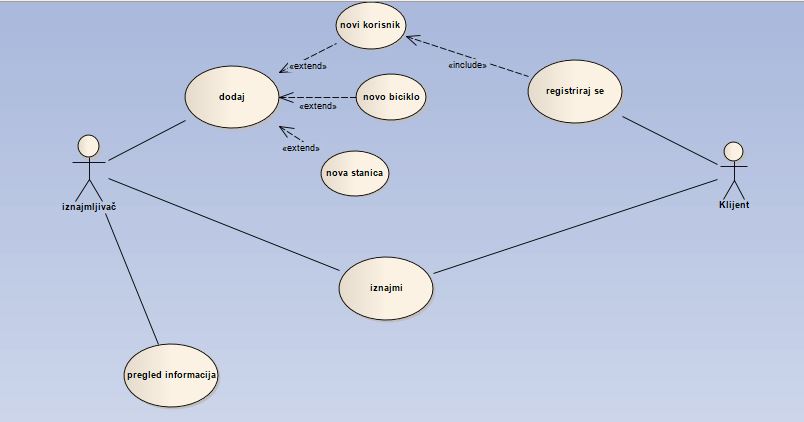
Iznajmljivač trebaju:

* dodati/ukloniti: bicikl, terminal, korisnika
* iznajmiti bicikl
* pregledat izvještaj

Klijent:

* Registrirati se
* Iznajmiti bicikl

|  |  |  |
| --- | --- | --- |
| Rb. | Opis | Prioritet |
| 1. | Unos/brisanje bicikla, terminala, korisnika. | 1 |
| 2 | evidencija vožnje. | 1 |
| 3. | Na aplikaciju se mora moći spojiti s bilo koje lokacije tako da baza podataka mora biti spremljena na internetu. | 1 |
| 4. | Prikaz frekvencije korištenja modela bicikala, veličine pojedinog bicikla, vrijeme početka vožnje. | 2 |
| 5. | Odabir terminala. | 2 |
| 6. | Treba omogućiti pregled trenutnih bicikala u vožnji. | 2 |
|  |  |  |



Slika Dijagram slučajeva korištenja (Use case) [1]

## Zahtjevi na sučelja

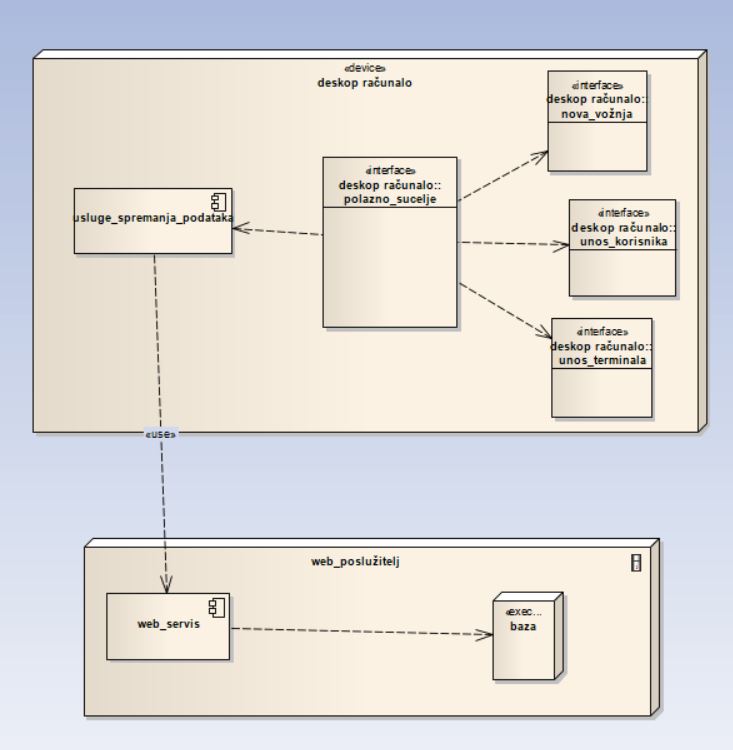
|  |  |
| --- | --- |
| Rb. | Opis |
| 7. | Vezano uz funkcionalni zahtjev 1: Aplikacija treba omogućiti izbor terminala na kojem se radi. |
| 8. | Na početnom zaslonu treba omogućiti prikaz 3. i 4. funkcionalnog zahtjeva. |
|  |  |

## Nefunkcionalni zahtjevi

|  |  |
| --- | --- |
| Rb. | Opis |
| 9. | Aplikacija se treba moći instalirati samostalno na Windows OS. |
| 10. | Za spremanje podataka koristi se MySQL baza podataka. |
|  |  |

# Arhitektura i dizajn sustava

## Arhitektura sustava

**

Slika 3 Arhitektura sustava [1]

Slika 3 prikazuje arhitekturu sustava. Aplikacija koristi dvoslojnu arhitekturu. Većina funkcionalnosti smještena je na samostojećem klijentu, dok su dodatne funkcionalnosti smještene na web poslužiteljima odnosno dostupne su pozivom određenog web servisa.

Po pokretanju programa pokreće se glavna forma na kojoj se prikazuje trenutni terminal, bicikla u vožnji, te graf frekvencije. Iz glavne forme u pri vrhu imamo menu bar s koje se mogu odabrati ostale funkcionalnosti izdavanja vožnje te dodavanje i brisanje bicikala, korisnik i terminala.

Pod sustav baze podataka pomoću web servisa omogućava pristupanje MySQL bazi podataka na serveru. Tako da je dostupna svim korisnicima.

## Detaljni opis sustava

*<Ovo potpoglavlje treba sadržavati detaljniji opis sustava. Dijagrami moraju biti u skladu sa skicom arhitekture iz prethodnog potpoglavlja.*

### Dijagram klasa

*Ako se koristi objektno-orijentirani razvoj, treba izraditi dijagram klasa (engl. class diagram) ili više njih – ovisno o složenosti sustava. U dijagramu obavezno navesti javne atribute klase te vrstu i kardinalnost veza između klasa (i sučelja). Naznačiti nazive, tipove podataka i razine pristupa elemenata. Za svaku klasu kratko navesti i opisati osnovne odgovornosti te klase.*

*Ako se ne koristi objektno-orijentirani razvoj, u ovom potpoglavlju umjesto dijagrama klasa treba izraditi odgovarajući dijagram koji pokazuje detaljnije elemente sustava.>*

### Ostali dijagrami

*<U ovom potpoglavlju po potrebi i u dogovoru s mentorom se uključuju dodatni dijagrami koji prikazuju detaljniju arhitekturu sustava. Primjerice dijagram slijeda (engl. sequence diagram), dijagram aktivnosti (engl. aActivity diagram), dijagram stanja (engl. state machine diagram), dijagram komponenti (engl. component diagram),… >*

## Model podataka

*<Ovo potpoglavlje treba sadržavati model podataka u grafičkom obliku. Ako se koristi relacijska baza podataka skicirati tablice i veze među njima u obliku relacijskog modela. Ako se ne koristi relacijska baza podataka, onda treba koristiti odgovarajući model koji pokazuje podatke i vezu među njima. Pored dijagrama, tekstom treba opisati podatke (primjerice opisati svaku tablicu i njezine atribute – za što je namijenjena tablica, koja je uloga svakog stupca tablice, koji su ključevi, i sl.).>*

# Implementacija sustava

*<Ovo poglavlja opisuje detalje implementacije sustava.>*

## Dijagram razmještaja

*<U ovom potpoglavlju treba prikazati cjelokupan fizički izgled sustava i mjesta na koja će se smjestiti njegove komponente, odnosno na koji su način dijelovi sustava povezani u cjelinu. Za prikaz koristiti dijagram razmještaja (engl. deployment diagram), a po potrebi dodatno tekstom pojasniti detalje razmještaja.>*

## Korištene tehnologije i alati

*<U ovom potpoglavlju treba pobrojiti sve tehnologije, alate, radne okvire, vanjske biblioteke i sl. koji su korišteni pri izradi sustava te ukratko opisati njihovu važnost, mjesto i način primjene u projektu.>*

## Izvorni kod programskog rješenja

*<Izvorni kod realiziranog rješenja zajedno sa svim testnim i konfiguracijskim datotekama treba biti dostupan mentoru i drugim zainteresiranim nastavnicima tijekom čitavog rada na projektu. Kod treba verzionirati sustava pomoću Git-a [1] korištenje Bitbucket [2] online sustava za dijeljenje programskog koda koji omogućuje besplatno korištenje malim timovima do 5 korisnika, s neograničenim brojem privatnih repozitorija.*

*Izvorni kod mora biti dobro dokumentiran i osim u opravdanim slučajevima licenciran nekom od licenci otvorenog koda (primjerice GPL, LGPL, BSD,…). Svaka datoteka, klasa te javna varijabla, metoda ili funkcija trebaju biti dokumentirani tako da se iz dokumentacije može prepoznati način njihovog korištenja u drugom dijelu programa. Za primjere komentiranja objektno-orijentiranog koda pogledajte primjerice način na koji dokumentiran Java source kod [3]. Pored toga svi složeniji algoritmi čija namjena ili način realizacije nije intuitivna treba komentirati.*

*U ovom potpoglavlju opisati način pristupa izvornom kodu i strukturu koda (primjerice organizaciju mapa). U ovaj dokument ne treba lijepiti izvorni kod.>*

## Testiranje programskog rješenja

*<U ovom poglavlju treba opisati provedbu testiranja implementiranih funkcionalnosti s prikazom izbora ispitnih slučajeva. Ispitni scenariji trebaju minimalno ispitati temeljnu funkcionalnost sustava navedenu u funkcionalnim zahtjevima.>*

## Upute za instalaciju

*<U ovom poglavlju treba opisati upute za instalaciju realiziranog programskog rješenja.>*

## Korisničke upute

*<U ovom poglavlju treba opisati korištenje svih funkcionalnosti sustava sa stajališta krajnjeg korisnika.>*

# Zaključak

Projektni zadatak bio je izrada samostojeće aplikacije koja omogućuje uvid u podatke iznajmljivanja bicikla. Aplikacija kao polazno stanje uzima postojeće izvještaje vožnji trenutno stanje bicikala i termina ali omogućuje i naknadnu izmjenu podataka (unos novog bicikla, korisnika i terminala te brisanje istih). Aplikacija je trebala biti prenosiva i jednostavna za instalaciju.

Zadatak je ostvaren te je u aplikaciji implementirana tražena osnovna funkcionalnost. Razvijena aplikacija je funkcionalna i stabilna te zadovoljava sve zadane funkcionalne i nefunkcionalne zahtjeve, mada neke ne obavlja jako efikasno što bi bio obavezan zahtjev kod aplikacije u stvarnom poslovnom modelu. Najveći problem se pojavio u dijelu testiranja funkcionalnosti povezivanja grafičkog sučelja s bazom podataka.

Kao i kod svakog projekta ima prostora za napredak te bi se u budućnosti aplikacija mogla unaprijediti. Primjeri mogućih unaprjeđenja su:

* Proširiti aplikaciju tako da se u njoj mogu evidentirati detaljniji podaci svih događaja i objekata.­
* Proširiti platformu sa mobilnom verzijom ili web aplikacijom da korisnici imaju u uvid gdje su stanice, koliko im je preostalo vožnje, prikaz vlastite statistike.
* ­ Omogućiti iznajmljivanje ostalih prijevoznih sredstava, i automatizirati postaju
* ­ Jediničnim testovima pokriti većinu programskog koda

# Reference

*<U ovom poglavlju popišite sve reference i literaturu koju ste koristili pri radu na projektu. Navedite sve dokumente i web adrese koje ovaj dokument referencira (dokumente, standarde, članke, web stranice, …) i poredajte ih prema mjestu navođenja u tekstu. U tekstu se treba pozvati na ovdje popisane reference navođenjem broja reference u uglatim zagradama. Kod navođenja referenci koristite* [*IEEE stil*](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf) *[4]. >*

1. *S. Chacon and B. Straub, Pro Git, 2nd ed. Apress, 2014. [E-book] Dostupno na:*[*https://git-scm.com/book/en/v2*](https://git-scm.com/book/en/v2) *[Pristupljeno: 12.01.2018.].*
2. *Atlassian Bitbucket, Distributed version control system, [Online]. Dostupno na:* [*https://bitbucket.org/*](https://bitbucket.org/) *[Pristupljeno: 12.01.2018.].*
3. *Java Community, Open-source implementation of the Java SE 8 Platform, [Online]. Dostupno na:* [*http://grepcode.com/snapshot/repository.grepcode.com/java/root/jdk/openjdk/8-b132/*](http://grepcode.com/snapshot/repository.grepcode.com/java/root/jdk/openjdk/8-b132/) *[Pristupljeno: 16.01.2018.].*
4. *IEEE, How to Cite References: IEEE Documentation Style, [Online]. Dostupno na:*[*https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf*](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf) *[Pristupljeno: 16.01.2018.].*