



Prolećni semester 2017/18
PROJEKTNII ZADATAK

Rent-a-bike sistem SANA

STUDENTI

Ana Sajić 2773

Sara Davidović 1924

PROFESOR

Prof. dr Ljubomir Lazić

ASISTENT

Boro Mijović

Beograd, 2018.

Sadržaj

Revizija projekta	4
1. Pokretanje, planiranje i realizacija Projekta	5
1.1. Ciljevi i zadaci.....	5
1.2. Sistemski iskazi opsega.....	6
1.2.1. Generalni zahtevi.....	6
1.2.2. Poboljšanja	6
1.3. Kontekst Sistema	6
1.4. Najveća Ograničenja.....	7
2. Procene obima i složenosti Projekta	8
2.1. Istorijski podaci korišćeni za procenu obima i troškova.....	8
2.2. Modeli i Tehnike primenjenih procena, rezultati obima i troškova	10
2.2.1 Procena veličine bazirana na linijama koda	10
2.2.2 Procena veličine bazirana na funkcionalnoj analizi (FP).....	12
2.2.3 Procena veličine bazirana na COCOMO II modelu	14
2.2.4 Procena veličine bazirana na BLIC metodologiji.....	16
2.2.5 Analiza izvršenih procena	17
2.2.6 Procena metodom “rule of thumb”	18
2.3 Procena Resursa Projekta.....	21
2.3.1. Minimalni hardverski zahtevi	21
2.3.2. Minimalni softverski zahtevi.....	22
2.3.3. Budžet.....	23
3. Menadžment rizika	28
3.1. Opseg i namera RMMM aktivnosti.....	28
3.2. Menadžment rizika organizacione uloge.....	28
3.3. Identifikacija i Opis rizika (verovatnoća i efekat)	29
3.4. Upravljanje rizicima	30
3.4.1 Direktor upravljanja promenama	30
3.4.2 Inicijatori promena	30
3.4.4 Savetodavni odbor za promene – CAB	30
3.4.5 Kreatori promena	30
3.4.6 Tester promena	31
3.4.7 Implementator promena.....	31

3.4.8 Recenzent promena	31
3.5 Identifikacija izvora rizika	32
Tabela 3.5.1 – Identifikacija i analiza rizika	32
Tabela 3.5.2 – Akcija smanjenja gubitka usled rizika	33
Tabela 3.5.2 – Akcije oporavka usled rizika	34
Tabela 3.5.3 – Intenzitet rizika	35
Tabela 3.5.4 – Klasifikacija rizika	36
4. Raspored aktivnosti projekta	37
4.1. Isporučenja i tačke donošenja odluka	37
4.1. Vremenski okvir	38
4.2. Strukturalna podela posla (WBS)	39
5. Organizacija projektnog tima	40
5.1. Struktura tima	40
5.2. Dodatne odgovornosti.....	41
6. Mehanizmi praćenja i kontrole	41
6.1. Mehanizmi osiguranja kvaliteta	41
6.2. Kontrola i menadžment promena	43
7. Upravljanje komunikacijom na projektu	43
7.3 Plan izveštavanja	45
7.4 Stejholderi.....	45
7.5 Informacije	46
7.6 Distribucija.....	47
7.7 Plan dokumentacije.....	48
7.8 Plan za rešavanje problema	49
7.8.1 Izveštavanje problema	49
7.8.2 Uloge.....	50
7.8.3 Nedeljni sastanak	51
8. ZAKLJUČAK.....	52
9. LITERATURA	52
10. PRILOZI.....	52
10.1. Reference	53
10.2. Korišćeni softverski alati.....	53

Revizija projekta

Datum	Opis	Autor	Komentari
18.05.2018.	Predaja I faze projekta	Ana Sajić Sara Davidović	Ispravite greške i ubrzajte sa radom. Čim završite određeno poglavlje pošaljite na reviziju.
04.06.2018	Predaja II faze projekta	Ana Sajić Sara Davidović	
13.06.2018	Pregled II faze projekta	Ana Sajić Sara Davidović	

1. Pokretanje, planiranje i realizacija Projekta

Pokretanje, planiranje i realizacija projekta će biti prikazana u sledećim poglavljima u nastavku teksta.

1.1. Ciljevi i zadaci

Koji problem ovaj projekat rešava?

Ova projekat ima za cilj iznajmljivanje bicikala na nivou grada Beograda. Korisnik će biti u mogućnosti da pomoću ovog sistema izabere model, kategoriju i vreme kada želi iznajmiti bicikl (Slika 1.1.1).

Cilj projekta (šta treba biti postignuto u određenom vremenskom periodu) Kreiranje veb sajta za iznajmljivanje bicikala	
Rezultat projekta (šta treba biti isporučeno u planiranom vremenskom periodu) Isporuka projekta podrazumeva online veb sajt na adresi www.rentabikebg.rs kroz period od 4 meseca. Nakon toga bi trebalo testirati sve funkcionalnosti i ukoliko neka funkcionalnost ne radi, to je potrebno ispraviti.	
Uspešnost projekta(šta su očekivanja zainteresovanih strana)	
Zainteresovane strane(Stakeholders)	Kriterijumi uspešnosti
Agencija	Očekivanja su da sajt bude završen na vreme. Da bude jednostavan za upotrebu kako bi korisnici mogli lako da iznajme bicikl. Takođe, nakon toga se očekuje da sajt poboljša njihovo poslovanje.
Programer	Očekuje da se ispoštuje dogovor koji je sklopljen. Da ispalata bude u roku. Da ne bude nekih dodatnih zahteva, nakon isporuke.

Slika 1.1.1 – Ciljevi i zadaci

1.2. Sistemski iskazi opsega

Koji je opseg ovog projekta?

U ovom sistemu imamo dva tipa korisnika, tako da je potrebno je da sistem ispunjava zahteve za svaki tip korisnika. S obzirom na to da je aplikacija zamišljena da bude web aplikacija, odlučujemo se za MVC arhitekturu. MVC deli sve ono što jedna aplikacija radi na tri dela. Ovakvo gledanje na funkcionalnost - takozvano razdvajanje zaduženja, smatra se i inače dobrom praksom.

1.2.1. Generalni zahtevi

U zavisnosti od tipova funkcionalnosti koje treba da se obezbede putem ove aplikacije možemo razlikovati dve grupe korisnika. Prvu grupu bi predstavljali klijenti koji imaju potrebe za uslugama agencije rent a bike, dok administrator ima zadatak stavljanja novih informacija vezanih za iznajmljivanje, i vodi računa o tome da se svaka nova mušterija se upisuje u bazu podataka da bi se evidentirali iznajmljeni bicikli.

1.2.2. Poboljšanja

Poboljšanje koje naš sistem može da doprinese jeste samoj rent-a-bike prodavnici, jer na taj način će olakšati poslovanje. Kada pogledamo, svaki rent-a-bike vodi evidenciju na nekim papirima, ili u nekom zastarelom programu ili možda vodi 2-3 ili više istih rent-a-bike prodavnica, pa mu je neophodno da vodi računa da ukoliko u jednom nema konkretan model bicikala, bude u recimo drugoj prodavnici koja se nalazi svega nekoliko metara dalje. Na ovaj način prodavac ne gubi klijentelu i ima odgovarajuć informacioni sistem.

1.3. Kontekst Sistema

Sajt se sastoji od sledećeg:

- admin panel koji koristi admin(unos bicikala, izmena bicikala, brisanje bicikala, pregled bicikala, pregled korskika)
- korskik panel za korisnike(pretraga bicikala, rentiranje bicikala)

1.4. Najveća Ograničenja

Aplikacija nije od životne važnosti, ali jeste od poslovne, stoga moramo voditi računa da se aplikacija obezbedi kako treba i pravovremenom, kako ne bi došlo do pronevere ili curenja podataka.

Vreme

Za razvoj aplikacije će biti potrebno 14 nedelja. Razvoj aplikacije obuhvata razvoj source code-a, kompletiranje celokupne dokumentacije, izradu svih artefakata vezanih za izradu projekta, kao i potrebno istraživanje tržišta i potrebnih tehnologija.

Korisnički interfejs

Korisnički interfejs bi trebalo da bude jasan i jednostavan. Prilikom pojavljivanja grešaka, sistem treba da prikaže poruke o greškama i na taj način upozori i obavesti korisnika o tipu greške, kao o tome gde je greška nastala.

2. Procene obima i složenosti Projekta

Procenjujemo obim i složenost po sledećim poglavljima:

2.1. Istorijski podaci korišćeni za procenu obima i troškova

From: <https://www.ibm.com/developerworks/library/qsm/>

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70
COBOL *	61	55	23	297
Cognos Impromptu Scripts +	47	42	30	100
Cross System Products (CSP) +	20	18	10	38
Cool:Gen/IEF *	32	24	10	82
Datastage	71	65	31	157
Excel *	209	191	131	315
Focus *	43	45	45	45
FoxPro	36	35	34	38
HTML *	34	40	14	48
J2EE *	46	49	15	67
Java *	53	53	14	134
JavaScript *	47	53	31	63
JCL *	62	48	25	221
LINC II	29	30	22	38
Lotus Notes *	23	21	19	40
Natural *	40	34	34	53
.NET *	57	60	53	60
Oracle *	37	40	17	60
PACBASE *	35	32	22	60
Perl *	24	15	15	60
PL/I *	64	80	16	80
PL/SQL *	37	35	13	60
Powerbuilder *	26	28	7	40
REXX *	77	80	50	80
Sabretalk *	70	66	45	109
SAS *	38	37	22	55
Siebel *	59	60	51	60
SLOGAN *	75	75	74	75
SQL *	21	21	13	37
VB.NET *	52	60	26	60
Visual Basic *	42	44	20	60

Slika 2.1.1 – Tabela QSM gde vidimo funkcionalne poene za HTML i JavaScript

FP relationships

The relationship between lines of code and function points depends upon the programming language that is used to implement the software and the quality of the design [Pre97]. The following table presents rough estimates for some old-fashioned programming languages. The paper describing these results was published in 1983 [AlG83]. I will include some very preliminary estimates for Java and php to the table. These estimates come from Summer 2005 Software Engineering Projects where both FP and lines of code were calculated.

Programming language	LOC/FP (average estimate)
assembly language	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
php	67
Java	31
object-oriented languages	30
fourth generation languages (4GLs)	20
code generators	15
spreadsheets (excel programming)	6
graphical languages (icons) (draw-a-program languages)	4

Slika 2.1.2 – Vidimo funkcionalne poene za PHP

Na Slici 2.1.1 i Slici 2.1.2 vidimo da je za korišćenje JavaScripte funkcionalni poen 47, za HTML je 34, dok je za php 67.

.

2.2. Modeli i Tehnike primenjenih procena, rezultati obima i troškova

Od velike nam je važnosti da dobro procenimo očekivane rezultate i troškove za dalji rad.

2.2.1 Procena veličine bazirana na linijama koda

Pošto u našem projektu je rađena aplikacija koja se radi u 3 jezika PHP, JavaScript i HTML, morale smo da podelimo 100% broj linija koda na 3 dela odnosno u procentima: php60%, javaScript 15% i HTML 25%.

LOC je broj linija koda, koji se računa prema sledećoj formuli:

$$LOC = \frac{P + (E * 4) + O}{6}$$

P —> Pesimistično predstavljanje

E —> Predpostavljeno

O —> Optimistično

PREDVIĐANJE BROJA LINIJA KODA		
PESIMISTIČNO	PREDPOSTAVLJENO	OPTIMISTIČNO
24.000,00	20.000,00	19.000,00

Tabela 2.2.1.1.-Predviđanje broja linija koda

Na osnovu tabele 2.2.1.1 koristimo formulu da izračuna broj linija koda:

$$\begin{aligned}
 LOC &= \frac{P + (E * 4) + O}{6} \\
 LOC &= \frac{24.000 + (20.000 * 4) + 19.000}{6} \\
 LOC &= \frac{24.000 + 80.000 + 19.000}{6} \\
 LOC &= \frac{123000}{6} \\
 LOC &= 20.500
 \end{aligned}$$

Na osnovu broja linija koda, dalje ćemo računati fp (Function point), za svaki jezik koji smo koristile u toku izrade projekta. Podelu koju smo uradile je :

Php=60%, javaScript=15% i html=25%

$$FP(\text{php})=20.500*0,6/67=183.58$$

$$FP(\text{js}) = 20.500*0.15/47=65.43$$

$$FP(\text{html})=20.500*0.25/34=150.74$$

$$FP=183.58+65.43+150.74$$

$$FP \text{ ukupno}=399.74$$

2.2.2 Procena veličine bazirana na funkcionalnoj analizi (FP)

Rađenje prosečne veličine bazirane na funkcionalnoj analizi radile smo uz pomoć alata koji se nalazi u referenci dokumenta. Kako bi smo to uradile, najpre smo morale da popunimo tabelu koja se nalazi na slici 2.2.2.1, a nakon toga smo zadale ostatak alata i dobile rezultat Slika 2.2.2.2.

Help		Simple	Average	Complex
?	Number of User Inputs	<input type="text" value="10"/>	<input type="text" value="5"/>	<input type="text" value="22"/>
?	Number of User Outputs	<input type="text" value="5"/>	<input type="text" value="9"/>	<input type="text" value="1"/>
?	Number of User Inquiries	<input type="text" value="2"/>	<input type="text" value="5"/>	<input type="text" value="10"/>
?	Number of Files	<input type="text" value="4"/>	<input type="text" value="5"/>	<input type="text" value="1"/>
?	Number of External Interfaces	<input type="text" value="3"/>	<input type="text" value="2"/>	<input type="text" value="1"/>

Slika 2.2.2.1- Rezultat za funkcionalne poene

		0	1	2	3	4	5
1.	Does the system require reliable backup and recovery?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
2.	Are data communications required?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Are there distributed processing functions?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Is performance critical?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Will the system run in an existing, heavily utilized operational environment?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Does the system require on-line data entry?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

		0	1	2	3	4	5
8.	Are the master files updated on-line?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
9.	Are the inputs, outputs, files, or inquiries complex?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Is the internal processing complex?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Is the code designed to be reusable?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	Are conversion and installation included in the design?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13.	Is the system designed for multiple installations in different organizations?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
14.	Is the application designed to facilitate change and ease of use by the user?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The number of function points for your project is

Slika 2.2.2.2-Prikaz dobijanja FP broja.

FP=448.4

$LOC(\text{php}) = 448.4 * 67 * 0.6 = 18,025.68$

$LOC(\text{js}) = 448.4 * 47 * 0.15 = 3,161.22$

$LOC(\text{html}) = 448.4 * 34 * 0.25 = 3,811.40$

LOC (ukupno)=24.998,30

Na osnovu Slike 2.2.2.2, za naš sistem je izračunata vrednost 448.8 funkcionalnih poena. Dalje računamo potencijalni broj linija koda za :

2.2.3 Procena veličine bazirana na COCOMO II modelu

Izračunavanje kalkulacije prema COCOMO II modelu, pronašle smo alat na internetu koji smo i iskoristile (Slika 2.2.3.1). Njegova referenca se može pronaći u dokumentu na kraju.

Software Size Sizing Method **Source Lines of Code** ▼

SLOC % Design Modified % Code Modified % Integration Required Assessment and Assimilation (0% - 8%) Software Understanding (0% - 50%) Unfamiliarity (0-1)

New

Reused

Modified

Software Scale Drivers

Precedentedness **Nominal** ▼ Architecture / Risk Resolution **Nominal** ▼ Process Maturity **Nominal** ▼

Development Flexibility **Very High** ▼ Team Cohesion **Nominal** ▼

Software Cost Drivers

Product

Required Software Reliability **High** ▼

Data Base Size **Very High** ▼

Product Complexity **Nominal** ▼

Developed for Reusability **Very High** ▼

Documentation Match to Lifecycle Needs **Nominal** ▼

Personnel

Analyst Capability **Nominal** ▼

Programmer Capability **Very High** ▼

Personnel Continuity **Nominal** ▼

Application Experience **Nominal** ▼

Platform Experience **High** ▼

Language and Toolset Experience **High** ▼

Platform

Time Constraint **Extra High** ▼

Storage Constraint **Extra High** ▼

Platform Volatility **Very High** ▼

Project

Use of Software Tools **High** ▼

Multisite Development **Very High** ▼

Required Development Schedule **High** ▼

Maintenance **Off** ▼

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

Results

Software Development (Elaboration and Construction)

Effort = 184.3 Person-months
Schedule = 26.7 Months
Cost = \$221127

Total Equivalent Size = 20230 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	11.1	3.3	3.3	\$13268
Elaboration	44.2	10.0	4.4	\$53071
Construction	140.0	16.7	8.4	\$168057
Transition	22.1	3.3	6.6	\$26535

Staffing Profile

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	1.5	5.3	14.0	3.1
Environment/CM	1.1	3.5	7.0	1.1
Requirements	4.2	8.0	11.2	0.9
Design	2.1	15.9	22.4	0.9
Implementation	0.9	5.7	47.6	4.2
Assessment	0.9	4.4	33.6	5.3
Deployment	0.3	1.3	4.2	6.6

Your output file is http://csse.usc.edu/tools/data/COCOMO_June_11_2018_14_13_40_13076.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Slika 2.2.3.1 – Cocomo tabela

Prema Cocomo modelu možemo videti da smo dodali novih 20,000 linija koda, iskoristili smo 200 starih i modifikovali 1000. Takodje smo modifikovale 20% dizajna. Ostalo se može videti na Slika 2.2.3.1.

LOC=20,230

FP php = 20,230*0.6 = 181.16

FP js = 20,230*0.15 = 64.56

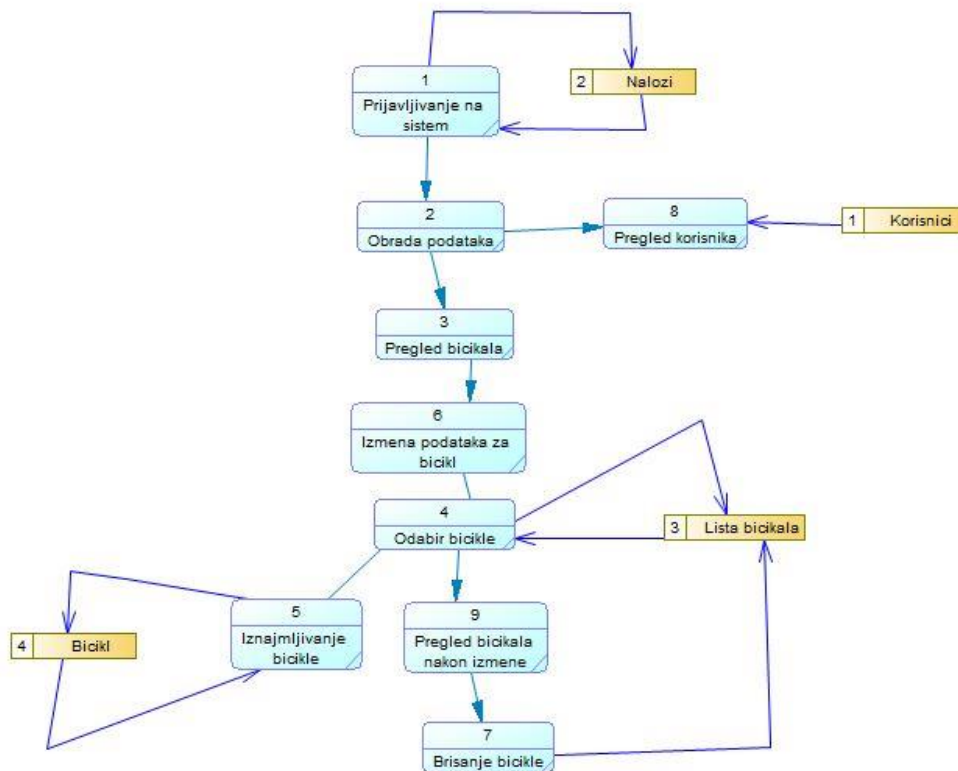
FP html = 20,230*0.25 = 148.75

FP ukupno=394.48

2.2.4 Procena veličine bazirana na BLIC metodologiji

Dijagram predstavlja prikaz projekata i izvora podataka Slika 2.2.4.1.

Procena na osnovu DFD dijagrama



Slika 2.2.4.1 -DFD dijagram

Na našem dijagramu imamo 9 projekata i 4 izvora podataka. Procena broja linija koda koju ćemo ovde izračunati prema datom dijagramu sa slike 2.2.4.1, računamo prema sledećoj formuli:

- Broj po dijagramu (9 projekata i 4 izvora podataka) = 13
- Prosečnim brojem programa po procesu (subjektivno) = 3
- Brojem linija koda po program(subjektivno)= 550

$$LOC = 13 * 3 * 550$$

$$LOC = 21.450$$

Da bi smo izračunali funkcionalne poene, računamo tako što borjem koeficijenta programskog jezika JavaScript (47), PHP(67) i HTML (34) delimo broj koji smo dobili u predhodnom računanju:

$$FP(\text{php})=21.450*0.6/67=192.09$$

$$FP(\text{js})=21.450*0.15/47=64.56$$

$$FP(\text{html})=21.450*0.25/34=148.75$$

$$FP(\text{ukupno})=192.09+64.56+148.75$$

$$FP(\text{ukupno})=405.40$$

2.2.5 Analiza izvršenih procena

Na osnovu predhodne 4 procene veličine softvera, različitom metodologijom, prilikom izračunavanja došli smo do sledećih rezultata:

ANALIZA JavaScript					
	Po broju linija koda	Po funkcionalnim poenima	COCOMO II	BLIC metodologija	Capers Jones
LOC	20,500.00	24,998.30	20,230.00	21,450.00	
FP	399.74	448.40	394.48	405.40	

Slika 2.2.5.1 -DFD Analiza izvršenih procena za JavaScript

Prema Slici 2.2.5.1 vidimo da su nam odsutapanja u LOC-u od 20.000-25.000=, a FP je od 390-449.

$$LOC=(20.500+24.998,3+20.230+21.450)/4 = 21,795$$

$$FP=(399,74+448,40+394,48+405,40)/4 = 412.01$$

2.2.6 Procena metodom “rule of thumb”

Ovu tehniku uveo je Capers Jones. Utvrdio je dvanaest pravila za procenu softvera. Kako bismo mogli da koristimo ovu tehniku moramo prvo izračunati broj FP ili broj linija koda. U našem slučaju, odlučili smo se za korišćenje funkcionalnih tačaka jer je procenjivanje tačnije.

12 Pravila Capers Jones-a:

Estimacija veličine izvornog koda

Programski jezik	Procenat udela u softveru	Koeficijent programskog jezika	Broj linija koda
PHP	60%	67	19.323,2
JavaScript	15%	47	2.904,6
HTML	25%	34	3.502
Ukupan broj linija koda (LOC)			25.729,2

Tabela 2.2.6.1-Pravila capers jones

Pravilo 1 - Estimacija veličine izvornog koda.

Na osnovu analize izvršenih procena iz poglavlja 2.2.5 gde smo dobili da je broj FP = **412.01** možemo da izračunamo LOC:

$$LOC(PHP) = 412.01 * 67 * 0,7 = 19.323,2$$

$$LOC(JavaScript) = 412.01 * 47 * 0,15 = 2.904,6$$

$$LOC(HTML) = 412.01 * 34 * 0,25 = 3.502$$

Pravilo 2 – Estimacija dokumentacije. Pomoću drugog pravila vrši se procena količine dokumentacije vezane za softverski paket. Izražava se u broju strana a izračunava se na sledeći način: Dokumentacija = $FP^{1.15} = 412.01^{1.15} = 1.016,59$

Pravilo 3- Estimacija odudaranja korisničkih zahteva. Odudaranje korisničkih zahteva u proseku iznosi oko 3% mesečno. Kako bi se izbegla neslaganja potrebno je definisati u ugovoru i promeniti troškove ovog problema na vreme.

Pravilo 4- Estimacija broja slučajeva testiranja Po četvrtom pravilu izračunavanje estimiranih test slučajeva glasi: Broj test slučajeva= $FP^{1.2} = 412.01^{1.2} = 1373,69$ test slučajeva

Pravilo 5- Estimacija mogućnosti dolaska do greške. Način na koji se vrši estimacija mogućnosti defekta u našem softveru je sledeći: Mogućnost greške= $FP^{1.25} = 412.01^{1.25} = 1856,24$ mogućnosti

Pravilo 6 – Estimacija efikasnosti otklanjanja greške. Po navedenom pravilu svaki vid testiranja pronaći i otkloniti 35 % u tom trenutku postojećih grešaka.

Pravilo 7- Estimacija efikasnosti organizovanog otklanjanja grešaka. Po datom pravilu svaka inspekcija dizajna pronaći i otkloniti 70% u tom trenutku postojećih grešaka, što je mnogo efikasniji ali i skuplji i zahtevniji način otklanjanja grešaka od prethodnog. Druga stavka ovog pravila glasi: Svaka inspekcija koda će pronaći i otkloniti 65% u tom trenutku postojećih grešaka.

Pravilo 8- Estimacija efikasnosti otklanjanja greške nakon. Tester programeri zaduženi za održavanje softvera mogu ispraviti 9 grešaka za jedan čovek mesec.

Pravilo 9- Estimacija trajanja realizacije projekta. Deveto pravilo glasi: Funkcionalne tačke stepenovane na 0,4 predviđaju približno vreme razvoja u kalendarskim mesecima, dakle:
Ukupno vreme = $FP^{0.4} = 412.01^{0.4} = 11,11$ kalendarskih meseci

Pravilo 10- Estimacija potrebnih ljudi za realizaciju projekta. Funkcionalne tačke podeljene sa 150 predviđaju približan broj ljudi potrebnih za izradu projekta. Za našu komponentu, taj broj izračunava se ovako: Broj ljudi = $FP/150 = 412.01/150 = 2.75$ - tj. 3 čoveka

Pravilo 11- Estimacija ljudi potrebnih za održavanje softvera. Funkcionalne tačke podeljene sa 750 predviđaju približan broj ljudi potrebnih za održavanje projekta. Za našu komponentu, taj broj izračunava se ovako: Broj ljudi za održavanje = $FP/750 = 412.01 / 750 = 1$ čovek

Pravilo 12- Estimacija napora. Ovo pravilo predstavlja kombinaciju pravila 9 i 10. Primenjuje se na sledeći način: Napor = vreme potrebno za izradu projekta * br.ljudi = $11.11 * 3 = 33,3$ čovek-meseci

2.3 Procena Resursa Projekta

U daljem tekstu vidćemo kako se procenjuju resursi projekta.

2.3.1. Minimalni hardverski zahtevi

Ovakav sistem u početku ne zahteva bolje performanse. Od uspeha aplikacije zavise dalja proširanjje hardverskog dela sistema.

Zahtev	Minimalna vrednost
Memorija	4.0 GB
Tip Procesora	32 bit Or 64 bit
Brzina Procesora	1.6 GHz*1
Swap Mesto	1.0 GB
Mesto na Hard Disku	500 MB
USB input	3.0

Tabela 2.3.1.1-Tabela zahteva hardvera

2.3.2. Minimalni softverski zahtevi

Windows XP je minimalni Windows operativni sistem kako bi aplikacija bila instalirana na desktop računarima. Minimalni Android operativni sistem potreban za pokretanje aplikacije biće 4.0.3.

Zahtev	Minimalna vrednost
Operativni sistem	Nije bitan
Mreža	Potrebno je angažovati profesionalca koji će odraditi posao najbolje moguće.
Web browser	Podrška postoji za Google Chrome, Safari, Operu i Mozilu.,
Konfiguracija okruženja	Nije potrebno
Podešavanje bezbednosnog zida	Sistem ne zavisi od konfiguracije bezbednosnog zida
Mobilna verzija	Koristi se kao web aplikacija, tako da jedino što nam treba je da mobilni koristi web browser i da može da pristupi internetu.

Tabela 2.3.1.2-Tabela zahteva softvera

2.3.3. Budžet

Kada je reč o definisanju budžeta, njega određuju menadžeri i direktor projekta. U budžet su uključena sva sredstva koja su neophodna za realizaciju projekta u novčanom smislu, a određuju se uz pomoć raznih tehnika za predviđanje potencijalnog budžeta i troškova projekta. Pre svega, neophodno je izvršiti procenu novčanih sredstava koji se odnose na plate zaposlenih po radnom satu.

Prilikom definisanja budžeta, uzimamo u obzir da postoje tri tipa resursa, a to su: posao, materijal i trošak. Na slici broj 2.3.3.1. vidimo sledeće: Menadžer projekta pripada tipu work, a plata mu je 10 dolara po satu, dok je za prekovremeni rad 13 dolara. Dalje vidimo da su zaposlena i dva programera, koji imaju platu 8 dolara po satu, a za prekovremeni rad dobijaju 9 dolara.

Analitičar ima 9 dolara po satu, dok za prekovremeni rad dobija 10 dolara. Grafički dizajner ima 6 dolara po satu, a za prekovremeni rad 8 dolara. Dalje vidimo da su zaposlena i dva testera koji imaju 6 dolara po satu, a za prekovremeni rad dobijaju 9 dolara.

Sledeće što vidimo jesu resursi tipa materijal. Cena servera iznosi 1500 dolara, cena hardvera iznosi 3000 dolara, Intelij IDEA iznosi 50 dolara, Microsoft Project iznosi 70 dolara, Power Designer iznosi 40 dolara i Photoshop iznosi 100 dolara.

Poslednja stavka u tabeli jeste obuka i ona pripada tipu trošak.

Menadžer projekta	Work	M	100%	\$10,00/hr	\$13,00/hr	\$0,00	Prorated	Standard
Developer 1	Work	D1	100%	\$8,00/hr	\$9,00/hr	\$0,00	Prorated	Standard
Developer 2	Work	D2	100%	\$8,00/hr	\$9,00/hr	\$0,00	Prorated	Standard
Analitičar	Work	A	100%	\$9,00/hr	\$10,00/hr	\$0,00	Prorated	Standard
Grafički dizajner 1	Work	G1	100%	\$6,00/hr	\$8,00/hr	\$0,00	Prorated	Standard
Tester 1	Work	T1	100%	\$6,00/hr	\$9,00/hr	\$0,00	Prorated	Standard
Tester 2	Work	T2	100%	\$6,00/hr	\$9,00/hr	\$0,00	Prorated	Standard
Server	Material	S		\$1.500,00		\$0,00	Prorated	
Hardver	Material	H		\$3.000,00		\$0,00	Prorated	
Intelij IDEA	Material	I		\$50,00		\$0,00	Prorated	
Microsoft Project	Material	M		\$70,00		\$0,00	Prorated	
Power Designer	Material	Pd		\$40,00		\$0,00	Prorated	
Photoshop	Material	P		\$100,00		\$0,00	Prorated	
Obuka	Cost	O					Prorated	

Slika 2.3.3.1: Tabela raspodele resursa

Na slikama 2.3.3.2, 2.3.3.3, 2.3.3.4. vidimo sledeće:

iskorišćenost resursa i koliko su oni vremenski angažovani na određenim aktivnostima u projektu i cenu njihovog rada.

Resource Name ▼	Work ▼	Cost ▼	Add New Column ▼
▲ Menadžer projekta	120 hrs	\$1.200,00	
Izrada biznis plana	20 hrs	\$200,00	
Identifikacija stejkholdera	8 hrs	\$80,00	
Postavljanje ciljeva	16 hrs	\$160,00	
Odabir tehnologije	4 hrs	\$40,00	
Planiranje vremena	8 hrs	\$80,00	
Planiranje troškova	4 hrs	\$40,00	
Sastanak projektnog tima	4 hrs	\$40,00	
Kontrola izmena u planu razvoja	16 hrs	\$160,00	
Razgovor sa korisnicima	32 hrs	\$320,00	
Objavljivanje app i doc	8 hrs	\$80,00	
▲ Developer 1	42 hrs	\$336,00	
Specificiranje zahteva	20 hrs	\$160,00	
Razvoj logike i funkcija	20 hrs	\$160,00	
Sastanak projektnog tima	2 hrs	\$16,00	
▲ Developer 2	46 hrs	\$368,00	
Istrazivanje tehnologija	40 hrs	\$320,00	
Odabir tehnologije	4 hrs	\$32,00	
Sastanak projektnog tima	2 hrs	\$16,00	
▲ Analitičar	122 hrs	\$1.098,00	
Izrada biznis plana	20 hrs	\$180,00	
Specificiranje zahteva	20 hrs	\$180,00	
Identifikacija stejkholdera	8 hrs	\$72,00	

Slika 2.3.3.2 – Iskorišćenost resursa

Resource Name ▼	Work ▼	Cost ▼	Add New Column ▼
<i>Analiza ocekivanja</i>	16 hrs	\$144,00	
<i>Planiranje ljudskih resursa</i>	4 hrs	\$36,00	
<i>Planiranje troskova</i>	4 hrs	\$36,00	
<i>Planiranje rizika</i>	2 hrs	\$18,00	
<i>Validacija zahteva</i>	32 hrs	\$288,00	
<i>Provera troskova</i>	8 hrs	\$72,00	
<i>Podnosenje izvestaja o</i>	8 hrs	\$72,00	
▲ Grafički dizajner 1	12 hrs	\$72,00	
<i>Dizajniranje izgleda aplikacije</i>	12 hrs	\$72,00	
▲ Tester 1	28 hrs	\$168,00	
<i>Planiranje kvaliteta</i>	8 hrs	\$48,00	
<i>Testiranje</i>	12 hrs	\$72,00	
<i>Provera kvaliteta</i>	8 hrs	\$48,00	
▲ Tester 2	12 hrs	\$72,00	
<i>Testiranje</i>	12 hrs	\$72,00	
▲ Server	3	\$4.500,00	
<i>Razvoj</i>	1	\$1.500,00	
<i>Pracenje I kontrola</i>	1	\$1.500,00	
<i>Zatvaranje</i>	1	\$1.500,00	
▲ Hardver	1,25	\$3.750,00	
<i>Inicijacija</i>	0,2	\$600,00	
<i>Planiranje I dizajniranje</i>	0,3	\$900,00	
<i>Razvoj</i>	0,6	\$1.800,00	

Slika 2.3.3.3 – Iskorišćenost resursa II deo

<i>Zatvaranje</i>	0,15	\$450,00	
▲ Intelij IDEA	2	\$100,00	
<i>Razvoj</i>	2	\$100,00	
▲ Microsoft Project	5	\$350,00	
<i>Inicijacija</i>	1	\$70,00	
<i>Razvoj</i>	2	\$140,00	
<i>Pracenje I kontrola</i>	2	\$140,00	
▲ Power Designer	5	\$200,00	
<i>Razvoj</i>	4	\$160,00	
<i>Pracenje I kontrola</i>	1	\$40,00	
▲ Photoshop	3	\$300,00	
<i>Razvoj</i>	3	\$300,00	
▲ Obuka		\$550,00	
<i>Obuka korisnika</i>		\$550,00	

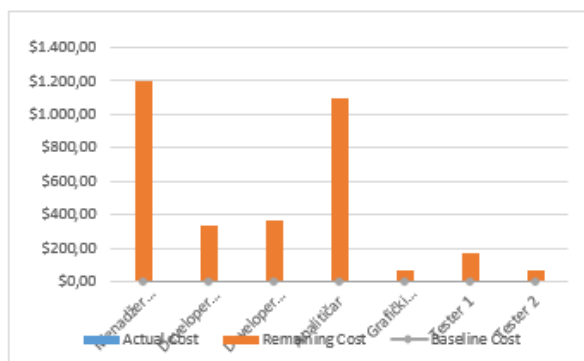
Slika 2.3.3.4 – Iskorišćenost resursa III deo

Dalje, na slikama 2.3.3.5, 2.3.3.6, 2.3.3.7 možemo videti i grafički prikaz troškova.

RESOURCE COST OVERVIEW

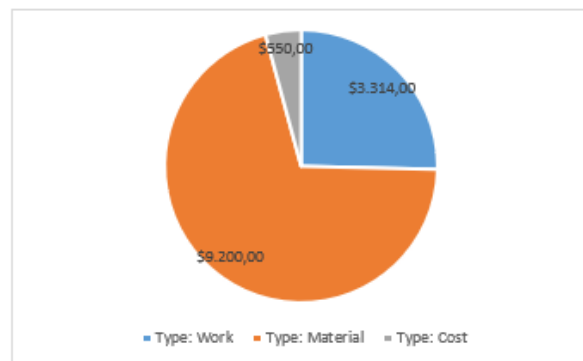
COST STATUS

Cost status for work resources.



COST DISTRIBUTION

How costs are spread out amongst different resource types.



COST DETAILS

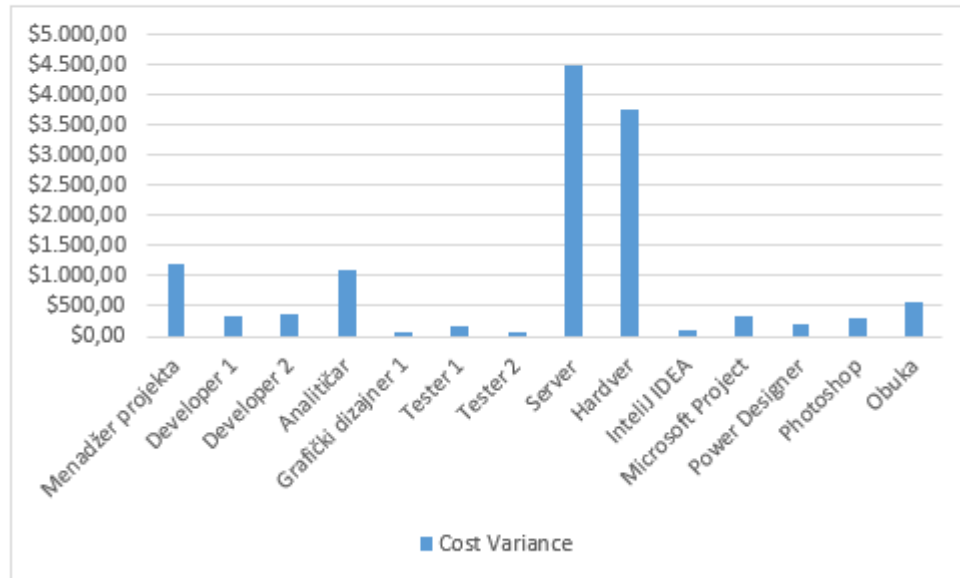
Cost details for all work resources.

Name	Actual Work	Actual Cost	Standard Rate
Menadžer projekta	0 hrs	\$0,00	\$10,00/hr
Developer 1	0 hrs	\$0,00	\$8,00/hr
Developer 2	0 hrs	\$0,00	\$8,00/hr
Analitičar	0 hrs	\$0,00	\$9,00/hr
Grafički dizajner 1	0 hrs	\$0,00	\$6,00/hr
Tester 1	0 hrs	\$0,00	\$6,00/hr
Tester 2	0 hrs	\$0,00	\$6,00/hr

Slika 2.3.3.5 –Grafički prikaz troškova

RESOURCE COST VARIANCE

Cost variance for all the work resources.

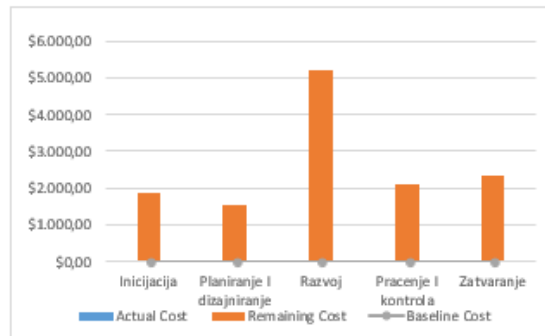


Slika 2.3.3.6 –Tabela variranja troškova

TASK COST OVERVIEW

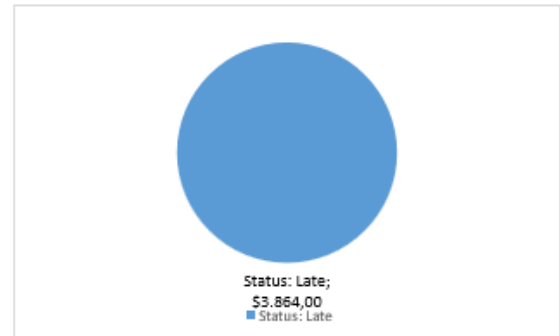
COST STATUS

Cost status for top-level tasks.



COST DISTRIBUTION

How costs are spread out amongst tasks based on their status.



COST DETAILS

Cost details for all top-level tasks.

Name	Fixed Cost	Actual Cost	Remaining Cost	Cost	Baseline Cost	Cost Variance
Inicijacija	\$0,00	\$0,00	\$1.846,00	\$1.846,00	\$0,00	\$1.846,00
Planiranje i dizajniranje	\$0,00	\$0,00	\$1.550,00	\$1.550,00	\$0,00	\$1.550,00
Razvoj	\$0,00	\$0,00	\$5.214,00	\$5.214,00	\$0,00	\$5.214,00
Pracenje i kontrola	\$0,00	\$0,00	\$2.104,00	\$2.104,00	\$0,00	\$2.104,00
Zatvaranje	\$0,00	\$0,00	\$2.350,00	\$2.350,00	\$0,00	\$2.350,00

Slika 2.3.3.7 –Prikaz troškova po fazama

3. Menadžment rizika

Menadžment rizika je opisan u daljem tekstu.

3.1. Opseg i namera RMMM aktivnosti

U cilju smanjenja mogućnosti za pojavu defekata, tim projektnih menadžera ima dužnost da usko sarađuje međusobno i svaki problem nastoji da reši dogovorom. Tim projekt menadžera je u obavezi da po izbijanju bilo kakvog problema koji može ugroziti projekat odmah sazove sastanak i pronađe najbolje rešenje za otkalnjavanje takvog problema. Sistem će se razvijati prema utvrđenim zahtevima, a o bilo kakvoj potrebi za izmenom zahteva ili definisanjem novih zahteva razvojni tim će odlučivati na sastancima koji će se specijalno organizovati za tu priliku. Podneti zahtevi će se razmotiri, i o prihvatanju ili odbacivanju istih tim će biti pisano obavešten.

3.2. Menadžment rizika organizacione uloge

Svaki član projektnog tima ima zadatak da teži ka tome da se projekat završi u predviđenom roku, na jasno definisan način i sa raspoloživim sredstvima.

Odgovornosti projektnog tima su:

- ❖ Tim projekt menadžera ima dužnost da pre početka razvoja kreira odgovarajući i izvodljiv raspored aktivnosti i zadataka za razvojni tim.
- ❖ Svaki član tima ima zadatak da svakog dana proverava raspored svojih aktivnosti, radi na njihovom izvršenju i beleži napredak svog rada i o tome obavesti tim na petnaestominutnim sastancima koji se održavaju svakog jutra na početku radnog vremena.
- ❖ Svaki član tima je dužan da o kašnjenju i problemima u radu obavesti tim na petnaestominutnim sastancima koji se održavaju svakog jutra na početku radnog vremena.
- ❖ Projekt menadžer ima obavezu da po pojavi problema reaguje i upozna ostatak tima sa datim problemom.

3.3. Identifikacija i Opis rizika (verovatnoća i efekat)

Rizik korisnika aplikacije:

Ukoliko sami korisnici aplikacije nisu zadovoljni istom, ili aplikacija ne naiđe na odobravanje ciljne grupe korisnika, postoji rizik da projekat neće biti realizovan prema očekivanjima.

Rizik konkurencije:

Ukoliko se za vreme razvoja aplikacije na tržištu pojavi slična ili čak i bolja aplikacija namenjena istoj ciljnoj grupi, to može imati katastrofalne posledice po uspeh aplikacije samim tim i uspeha projekta u celini.

Rizik razvoja:

Ukoliko se kasni sa realizacijom zadataka pri razvoju projekta, projekt menadžer je zadužen da problem reši ili preraspodeli zadatke. Pošto je reč o generičkom projektu koji se razvija iz sopstvenih sredstava, probijanje roka utiče na uspeh celog projekta.

Rizik zaposlenih:

Ukoliko neko od programera u razvojnom timu napusti tim u toku trajanja projekta, to može u velikoj meri ugroziti sam uspeh projekta, a probijanje roka takođe utiče na budžet projekta. Ako dizajner odluči da napusti tim, projekt menadžer mora istog trenutka angažovati novog dizajnera kako projekat ne bi bio ugrožen.

Rizik povećanja veličine projekta:

Svaki dodatni zahtev ili zahtev za izmenom će produžiti vreme izrade sistema i povećati njegov obim. Ovo utiče i na povećanje budžeta. Samim tim, dodavanje novih funkcionalnosti će biti predmet razmatranja nakon puštanja u rad aplikacije.

3.4. Upravljanje rizicima

3.4.1 Direktor upravljanja promenama

Ono što je glavna uloga direktora jeste odobrenje bilo kakve promene. Direktor takođe treba da komunicira sa svim članovima tima i zajedno sa njima učestvuje u eventualnim promenama. Samim tim što je zadužen za odobrenje promene, može se reći da je njegova uloga menadžerska jer on upravlja celokupnim projektom.

3.4.2 Inicijatori promena

Inicijator promena može biti bilo koja osoba u timu koja izloži ono što se treba promeniti. Ipak, svaka moguća i potencijalna promena se mora obrazložiti svim članovima kako bi bili upućeni šta eventualna promena donosi. Takođe, poželjno je da inicijatori promena predlože promenu u okviru svoje oblasti za koju su zaduženi kako ne bi došlo do nesuglasica.

3.4.4 Savetodavni odbor za promene – CAB

Savetodavni odbor za promene predstavlja grupu koja zajedno sa menadžerom promena deluje u svojstvu savetnika. Sastavljeni su od pojedinaca koji se smatraju relevantnim za donošenje odluka.

3.4.5 Kreatori promena

Kreatori promena su zaduženi za sprovođenje promena korišćenjem određenih alata i tehnologija koje su im na raspolaganju. Neophodno je osigurati sav potreban hardver, softver, licence koje su na raspolaganju kako bi što pre kreirali promenu pre slanja na testiranje.

3.4.6 Tester promena

Što se uloge testera tiče, za to je zadužen test inženjer. Svaka promena mora proći kroz testiranje, a potrebno je da tester promena ne bude creator promena iz razloga što je na taj način bolje i tačnije testiranje.

3.4.7 Implementator promena

Implementatori promena su oni koji sprovode promenu tj vrše implementaciju. U zavisnosti od vrste promene, zavisi i implementator. Ukoliko je promena u bazi podataka, onda je to osoba zadužena za upravljanje bazom podataka. Ukoliko postoji promena u sistemu, onda su to programeri, a ukoliko je reč o dokumentaciji onda je to menadžer.

3.4.8 Recenzent promena

To su grupa ljudi koje zajedno sa menadžerom promena vrši recenziju realizovanih promena i zatvara zahtev za promenu.

3.5 Identifikacija izvora rizika

Rizik pod rednim brojem 2 iz tabele 3.5.1 ima najveću mogućnost da nastane rizik, jer radnici nisu predvidljivi i mogu da promene mišljenje, raspoloženje kad god im je volja. Stoga radnici i njihova nemarnost su najčešći rizik svakog poslovanja, jer su nepredvidljivi. Svaki radnik može da nas košta i do 2000e po radniku. On iznosi 900 jedinica.

Redni broj	Identifikacija izvora rizika	Verovatnoća rizika (P)	Uticaj gubitaka usled rizika (C)	Izloženost rizika (RE)
1.	Opseg projekta – loše definisan projekta, cilj, plan... Ne shvatanje šta je klijent želeo i šta programer može da omogućiti.	0,26	1000e	260
2.	Nemarnost radnika -ukoliko u trenutku rada odпустimo radnika ili radnik da otkaz ili ukoliko radnici svakodnevno kasne ili prave prevelike pauze u svom radu.	0,45	2000e	900
3.	Česte promene zahteva – klijent želi da menja zahteve.	0,28	1100e	308
4.	Otkazivanje baze podataka -ukoliko baza podataka otkáže može biti kobno koliko za zaposlene toliko i za korisnike.	0,05	500e	250
5.	Testiranje softvera -loši testovi, kasno započeto testiranje, nefunkcionalni ili jednostavno loša procena programera da test nije neophodan.	0,20	800e	160

Tabela 3.5.1 – Identifikacija i analiza rizika

Drugi je česte promene zahteva. Ukoliko klijent stalno želi da menja svoje zahteve, ne samo da odugovlači naše vreme, već zaustavlja i mnoge projekte u našoj firmi što šteti ugled naše firme, kašnjenje na drugim projektima. Dakle, dodavanjem ne planiranih zahteva može nas koštati vremena mnogim projektima, zbog čega procenjujem da će ovaj rizik biti na drugom mestu i svaki zahtev može nas koštati i do 1100e po projektu što se nama ne isplati. Dakle, ovde će izloženost rizika imati 308 jedinica.

Kod tabele 3.5.1 mozemo videti akciju smanjenja gubitka usled rizika kako ne bi došlo do 100% štete, uvek moramo da se obezbedimo za ono čuveno "Ne daj Bože". Dakle, možemo videti šta možemo da preduzmemo da se ne bi dogodilo.

<i>Redni broj</i>	<i>Rizik</i>	<i>Akcija smanjenja gubitaka usled rizika</i>
1.	Opseg projekta – loše definisan projekta, cilj, plan... Ne shvatanje šta je klijent želeo i šta programer može da omogućiti.	Popraviti specifikacije i sprovesti ih u akciju.
2.	Nemarnost radnika -ukoliko u trenutku rada otpustimo radnika ili radnik da otkaz ili ukoliko radnici svakodnevno kasne ili prave prevelike pauze u svom radu.	Na vreme otkazivati radnike koji nisu sposobni za rad u smislu nemarni su i kasne (ukoliko nisu eksperti u svom poslu) u suprotnom uvesti radne subote i/ili skinuti dnevnicu sa plata.
3.	Česte promene zahteva – klijent želi da menja zahteve.	Ne izvršavanje dodatih zahteva klijenata.
4.	Otkazivanje baze podataka -ukoliko baza podataka otkáže može biti kobno koliko za zaposlene toliko i za korisnike.	Neka vrsta online besplatne baze podataka.
5.	Testiranje softvera -loši testovi, kasno započeto testiranje, nefunkcionalni ili jednostavno loša procena programera da test nije neophodan.	Dovođenje profesionalne radne snage na neodređeno vreme.

Tabela 3.5.2 – Akcija smanjenja gubitka usled rizika

Akcije oporavka usled rizika (Tabela 3.5.2) su naša pomoć u koliko dođe zaista do rizika. Takođe, može da pomogne i u budućnosti da se izbegnu moguće štete.

<i>Redni broj</i>	<i>Rizik</i>	<i>Akcija oporavka usled rizika</i>
1.	Specifikacija projekta – loše definisan projekta, cilj, plan... Ne shvatanje šta je klijent želeo i šta programer može da omogućiti.	Uvođenje stručnog kadra koji će napraviti specifikacije projekta i koja će navesti tačno šta kako ko gde i kad treba da uradi.
2.	Nemarnost radnika -ukoliko u trenutku rada odпустimo radnika ili radnik da otkaz ili ukoliko radnici svakodnevno kasne ili prave prevelike pauze u svom radu.	Otkaz nemarnim radnicima (zabušantima) i/ili uvođenje radne subote i/ili skidanje dnevnica sa plate.
3.	Česte promene zahteva – klijent želi da menja zahteve.	Ugovorom se obezbediti da ne dođe do čestih promena.
4.	Otkazivanje baze podataka -ukoliko baza podataka otkáže može biti kobno koliko za zaposlene toliko i za korisnike.	Prelazak na neko sigurno rešenje kao što je cloud.
5.	Testiranje softvera -loši testovi, kasno započeto testiranje, nefunkcionalni ili jednostavno loša procena programera da test nije neophodan.	Obezbeđivanje dodatne edukacije za osoblje ovog kadra.

Tabela 3.5.2 – Akcije oporavka usled rizika

Na osnovu tabele broj 3.5.3 možemo da raspoznamo rizik za naš sistem. Dakle, postoje rizici visoki i niski rizici.

<i>Verovatnoća</i>	<i>Posledice</i>				
	Beznačajno (1000 <)	Malo (1000-4000)	Umereno (4000-7000)	Veliko (7000-11000)	Katastrofalno (> 11000)
Skoro izvesno >90%	900	3850	6750	11550	90000
Verovatno (60% - 90%]	750	3000	6250	10500	75555
Umereno (25% - 60%]	500	2000	5500	9000	50000
Neizvesno (10% - 25%]	200	1500	5020	8520	22222
Retko 10% <	100	1250	4990	7845	12000

Tabela 3.5.3 – Intenzitet rizika

Prihvaljivo	
Umereno	
Zabrinjavajuće	

Legenda

Prema klasifikaciji rizika (Tabela 3.5.4) za nas softver ustanovile sam da su rizici spali u kategoriju gde je uticaj prihvatljiv i može se manipulirati kako bi se dobili željeni rezultati.

Redni broj	Identifikacija izvora rizika	Klasa rizika
1.	Opseg projekta – loše definisan projekta, cilj, plan... Ne shvatanje šta je klijent želeo i šta programer može da omogućiti.	Prihvaljivo
2.	Nemarnost radnika -ukoliko u trenutku rada odpuštimo radnika ili radnik da otkaz ili ukoliko radnici svakodnevno kasne ili prave prevelike pauze u svom radu.	Umereno
3.	Česte promene zahteva – klijent želi da menja zahteve.	Prihvaljivo
4.	Otkazivanje baze podataka -ukoliko baza podataka otkaze može biti kobno koliko za zaposlene toliko i za korisnike.	Prihvaljivo
5.	Testiranje softvera -loši testovi, kasno započeto testiranje, nefunkcionalni ili jednostavno loša procena programera da test nije neophodan.	Prihvaljivo























Tabela 3.5.4 – Klasifikacija rizika











4. Raspored aktivnosti projekta

Vremenski raspored aktivnosti prikazuje aktivnost kad koja počinje i kada se koja završava.

Takođe je moguće videti koliko dana koja traje Slika 4.1.1.

4.1. Isporučenja i tačke donošenja odluka

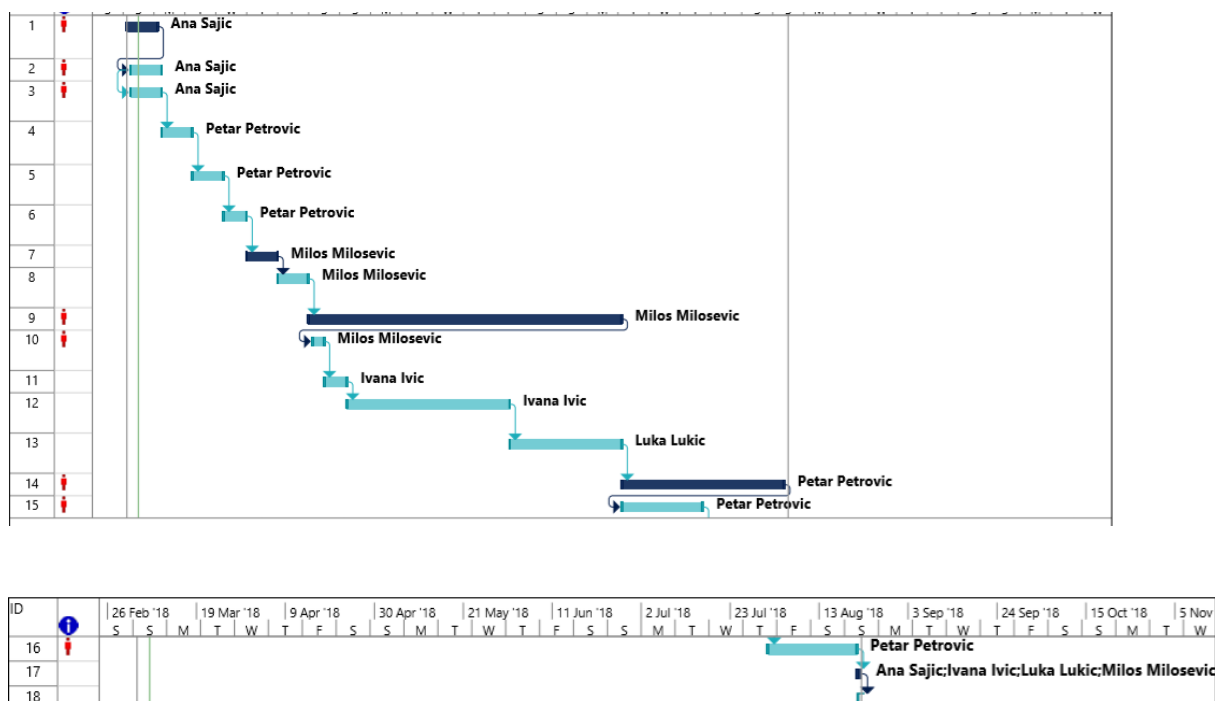
		Menadžement projekta	6 days	Mon 5.3.18	Sun 11.3.18		Ana Sajić
		Priprema biznis plana	6 days	Mon 5.3.18	Sun 11.3.18		Ana Sajić
		Priprema specifikacije zahteva	6 days	Mon 5.3.18	Sun 11.3.18		Ana Sajić
		Planiranje i dizajniranje	6 days	Mon 12.3.18	Sun 18.3.18		Petar Petrovic
		Istraživanje i čitanje dokumentacije	6 days	Mon 12.3.18	Sun 18.3.18		Petar Petrovic
		Dizajn grafičkog korisničkog interfejsa	4 days	Thu 15.3.18	Tue 20.3.18		Petar Petrovic
		Dizajn baze podataka	6 days	Tue 20.3.18	Tue 27.3.18		Milos Milosevic
		Kreiranje konceptualnog modela	6 days	Tue 20.3.18	Tue 27.3.18		Milos Milosevic
		Razvoj	58 days	Tue 27.3.18	Thu 14.6.18		Milos Milosevic
		Implementacija baze podataka	3 days	Tue 27.3.18	Thu 29.3.18		Milos Milosevic
		Razvoj prototipa	4 days	Fri 30.3.18	Wed 4.4.18		Ivana Ivic
		Back-end programiranje	30 days	Thu 5.4.18	Wed 16.5.18		Ivana Ivic

		Izrada klijentske aplikacije	21 days	Wed 16.5.18	Wed 13.6.18		Luka Lukic
		Testiranje	30 days	Wed 13.6.18	Tue 24.7.18		Petar Petrovic
		Alfa testiranje	15 days	Wed 13.6.18	Tue 3.7.18		Petar Petrovic
		Beta testiranje	15 days	Tue 3.7.18	Mon 23.7.18		Petar Petrovic
		Puštanje u rad	1 day	Mon 23.7.18	Mon 23.7.18		Ana Sajić; Ivana Ivic
		Održavanje					

Slika 4.1 - Aktivnosti projekta

4.1. Vremenski okvir

Dijagram sa slike 4.1.2 prikazuje nam aktivnosti koje treba izvršiti kako bi smo ostvarili cilj odnosno napravili softver koji zadovoljava interes klijenta. Vidimo i članove tima koji će izvršiti određene aktivnosti.



Slika 4.1.2 - Vremenski okvir

4.2. Strukturalna podela posla (WBS)

Work break down structure predstavlja hijerarhijsko razlaganje ukupnog obima posla koji treba da obavi projektni tim, kako bi ostvario projektne ciljeve i postigao adekvatne rezultate.

U našem slučaju sistem se deli na sledeće zadatke:

Specifikacija zahteva:

- Priprema biznis plana
- Priprema specifikacije zahteva

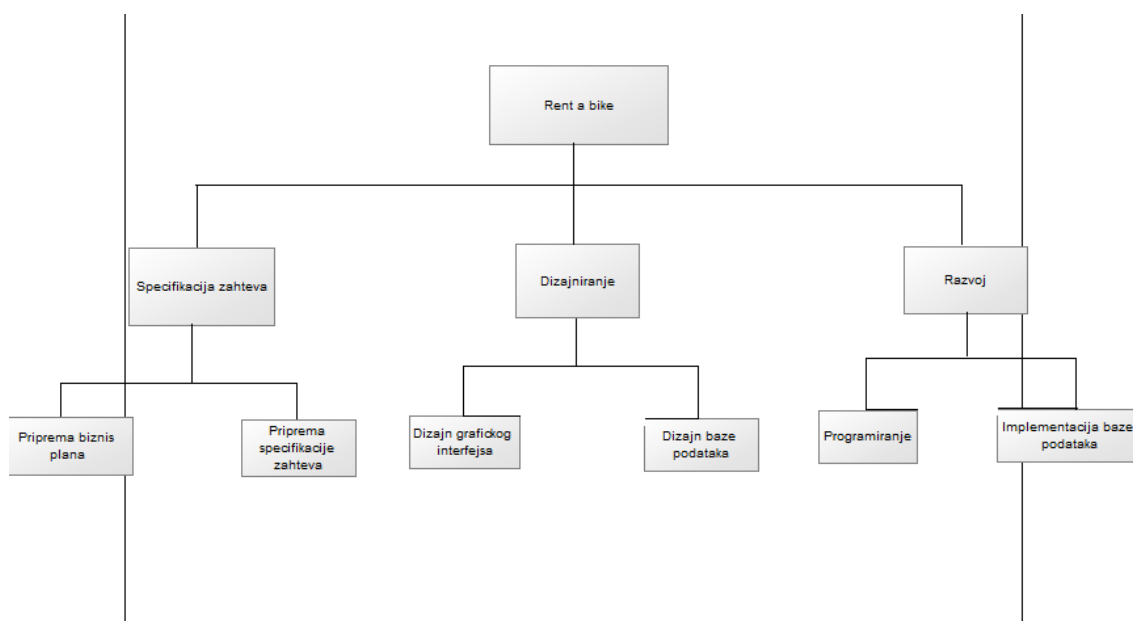
Dizajniranje:

- Dizajn grafičkog interfejsa
- Dizajn baze podataka

Razvoj:

- Programiranje
- Implementacija baze podataka

Na Slici 4.2.3 mozemo videti podelu posla prikazanog u WBS dijagramu.



Slika 4.2.3- WBS

5. Organizacija projektnog tima

5.1. Struktura tima

Razvojni tim će većinu odluka donositi dogovorom. U slučaju sukoba unutar tima, odluku će donesti većinski, shodno vlasničkom udelu u firmi. U nastavku teksta su data imena članova projektnog tima i njihove odgovornosti.

Projekt menadžeri: Ana Sajić i Sara Davidović

- ❖ Kontrola rada i kvaliteta
- ❖ Sastavljanje projektnog plana i verifikovanje druge neophodne dokumentacija

Softver arhitekta: Ivana Ivić

- ❖ Definisanje zahteva za softver
- ❖ Odabir tehnologija

Softver arhitekta: Miloš Milošević

- ❖ Dizajn arhitekture softvera
- ❖ Sastavljanje softver dokumentacije

Programeri: Miloš Milošević, Ivana Ivić

- ❖ Razvoj sistema prema urađenom dizajnu
- ❖ Razvojno testiranje

Testeri i održavanje: Petar Petrović i Milos Milosevic

- ❖ Testiranje aplikacije
- ❖ Integraciono testiranje
- ❖ Održavanje sistema
- ❖ Sastavljanje korisničkog uputstva

Dizajner: Ivana Ivić

- ❖ Izrada grafičko korisničkog interfejsa
- ❖ Pomoć pri sastavljanju korisničkog uputstva

5.2. Dodatne odgovornosti

Članovi projektovog tima će često biti angažovani na zajedničkim aktivnostima ili na aktivnostima koje se preklapaju. Potrebno je izraditi i marketinšku strategiju u čemu će biti uključeni svi članovi tima.

6. Mehanizmi praćenja i kontrole

6.1. Mehanizmi osiguranja kvaliteta

Kako bi se obezbedio kvalitet sistema, proćiće se kroz nekoliko koncepata:

Kvalitet

Odnosi se na merljive karakteristike softvera. Ove karakteristike se upoređuju uz pomoć definisanog standarda.

Kontrola Kvaliteta

Predstavlja seriju inspekcija, pregleda i testiranja korišćenih tokom razvojnog ciklusa softverskog proizvoda.

Ključni koncept je upoređivanje proizvoda sa definisanim i merljivim standardima.

Za cilj ima da obezbedi QA menadžera sa neophodnim podacima o kvalitetu softvera.

Sastoji se iz 2 tipa:

- Kvaliteta dizajna
Karakteristike koje dizajneri određuju za stavku.
Obuhvata: zahteve, specifikacije I dizajn sistema.
- Kvaliteta usklađenosti

Stepen do kog se poštuje specifikacija dizajna. Fokusira se na realizaciji zasnovanoj na dizajnu.

Garancija Kvaliteta

SQA / Software Quality Assurance ili Grancija kvaliteta softvera je aktivnost koja se primenjuje tokom procesa softvera.

Ona obuhvata:

- Pristup upravljanja kvalitetom
- Efektivnu softversko inženjersku tehnologiju
- Formalne tehničke preglede
- Višeslojne strategije testiranja
- Promenu kontrole dokumenta / Document Change Control
- Razvoj standarda softvera i njegovu proceduru kontrole
- Mehanizam merenja i izveštavanja

Troškovi kvaliteta

Uključuje sve troškove koji su nastali u potrazi za kvalitetom ili radom obavljenim za utvrđivanje kvaliteta.

Obuhvata:

- Troškove prevencije
 - *Planiranje kvaliteta*
 - *Formalni tehnički pregledi*
 - *Ispitivanje opreme*
 - *Obuka*
- Procenjene troškove
 - *Inspekcija unutar i izvan procesa*
 - *Kalibracija opreme I održavanje*
 - *Testiranje*
- Troškove neuspeha
 - *Dorade, popravke i analiza neuspeha*
 - *Promene u rezoluciji*
 - *Povratak proizvoda i zamena*
 - *Obezbeđena pomoć i podrška u vidu kontakta (Help line support)*
 - *Radna garancija*

6.2. Kontrola i menadžment promena

Razumevanje i kontrolisanje promena prilikom razvoja softvera je jedan od najvećih izazova razvojnog tima.

Do promena može doći iz više razloga, neki od njih mogu biti planirani a neki i ne.

Neki od razloga dolaska promena:

- Planiranje softverskih zahteva
- Ispravljanje grešaka prilikom testiranja
- Ispravljanje nepredvidivih posledica
- Razrešavanje dobijenih žalbi od strane klijenata

Uz pomoć menadžmenta kontrole, promene ne ometaju već poboljšavaju razvitak softvera. Ovo direktno utiče na kontrolisanje razvoja softvera i obezbeđuje poboljšanje metrika i procesa, povećava zadovoljstvo kupca i smanjuje vreme „provedeno“ na tržištu.

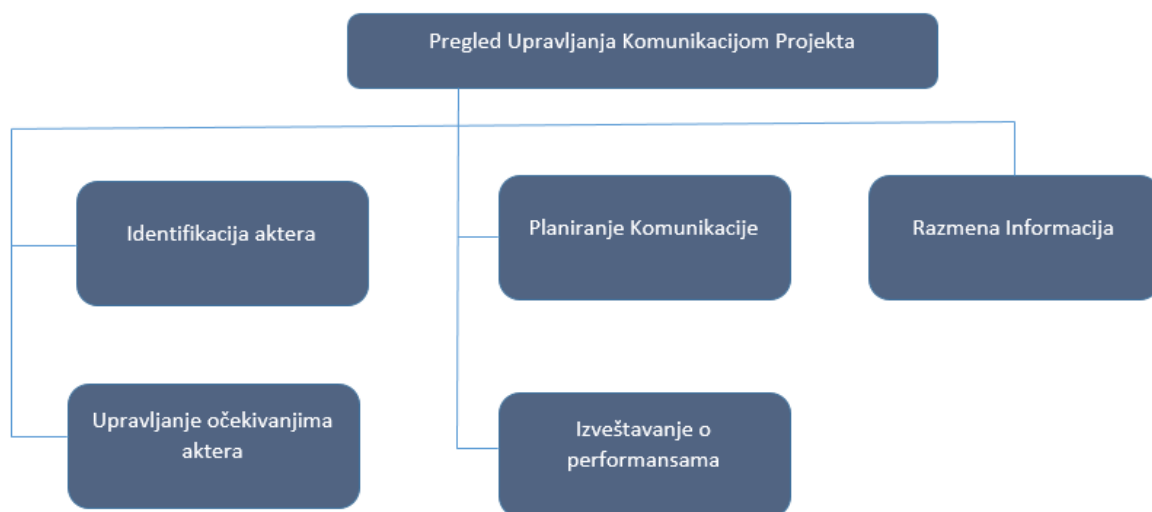
7. Upravljanje komunikacijom na projektu

Upravljanje komunikacijama u projektima predstavlja jednu od funkcionalnih oblasti upravljanja projektima.

Cilj upravljanja komunikacijom projekta jeste da obezbedi prikupljanje, skladištenje, raspoređivanje i raspolaganje informacijama unutar projekta.

Postoji pet glavnih procesa upravljanja komunikacijom:

- ❖ Identifikacija aktera
- ❖ Planiranje komunikacije
- ❖ Razmena informacija
- ❖ Upravljanje očekivanjima aktera
- ❖ Izveštavanje o performansama



Dijagram 7.1. - Upravljanje komunikacijom projekta

Komunikacija je definisana kao socijalna interakcija koja se odvija porukama.

Identifikovana je kao jedan od najvećih razloga za uspeh, tj. neuspeh projekta. Iz ovog razloga menadžer projekta provodi oko 90% svog vremena u komunikaciji sa članovima tima kao i sa svim eksternim stakeholderima.

Kako bi komunikacija bila efektivna, primljena poruka mora biti shvaćena u značenju koje je toj poruci dao pošiljalac.

Druga bitna stavka je otvorenost u komunikaciji, jer poboljšava odnos između članova projektnog tima i utiče na stvaranje međusobnog poverenja članova.

Zbog potrebe komunikacije i putem interneta i dokumentacije informacija koje bi u suprotnom bile izgubljene, rukovodilac projekta povezan je sa svim učesnicima u projektu putem slack i trello alata.

Upravljanje komunikacijom projekta biće detaljno dato kao prilog drugoj fazi projektnog zadatka.

Sadržaje sledeće:

7.1 Organizacija projekta

Vizuelno će prikazati internu i eksternu strukturu ljudi koji imaju interes u realizaciji projekta. U eksternoj strukturi menadžer projekta vrši komunikaciju i prosleđuje informacije između svog tima i kompanije Rent a bike SANA.

U internoj strukturi, resursi komuniciraju međusobno i direktno podnose izveštaje rukovodiocu.

7.3 Plan izveštavanja

U ovom delu se identifikuju stakeholderi u projektu, njihovi informacioni zahtevi, elementi koji se distribuiraju u okviru procesa komunikacije i performanse podataka koje se prosleđuju tokom trajanja projekta.

7.4 Stakeholderi

Zainteresovane strane - Stakeholders	Kriterijum uspešnosti
Projekt menadžeri	Žele da se projekat uspešno zatvori. Da tim usvoji nove veštine, i da zadovoljni klijenti iskažu zainteresovanost za dalju saradnju.
Projektni tim	Očekivanja su da se poštuju ranije dogovoreni uslovi, poštovanje svih postignutih dogovora kao i isplatu u dogovorenom roku. Ako se uslovi promene, da se shodno tome promeni i ugovor, koji će voditi povećanju vremena potrebnom za izradu aplikacije, kao i povećanju isplate.
Korisnici aplikacije	Korisnici očekuju da imaju mogućnost da pregledaju bicikle, i rentiraju onaj koji žele. Želi da mu ceo proces bude intuitivan, lak za korišćenje i da mu upotreba aplikacije ne oduzima više vremena nego što je to ranije bio slučaj.

Tabela 7.3.1 : Prikaz stakeholdera

7.5 Informacije

Stejkholder	Opis	Format	Učestalost
Korisnici aplikacije	korisničko uputstvo	elektronski	na završetku
Projekt menadžer	SRS dokument	papir	po kreiranju
	plan testiranja	elektronski	po kreiranju
	izveštaj o testiranju	elektronski	nakon obavljenog testiranja
	softver dokumentacija	elektronski	po kreiranju
	plan upravljanja rizicima	elektronski	po kreiranju
	plan upravljanja promenama	elektronski	po kreiranju
	plan komunikacije	elektronski	po kreiranju
Projektni tim	projekt menadž. plan	elektronski	po kreiranju
	SRS dokument	elektronski	po kreiranju
	plan testiranja	elektronski	po kreiranju
	softver dokumentacija	elektronski	po kreiranju
	plan komunikacije	elektronski	po kreiranju

Tabela 7.4.1 : Informacioni zahtevi stejkholdera

7.6 Distribucija

Slededa matrica distribucije ilustruje koji učesnici projekta će primati određene elemente komunikacije:

Document/Item	Dist. Method	Projekt Menadžeri	Projektni tim
Plan nedeljnog sastanka	email		x
Izveštaj sa nedeljnog sastanka	email	x	x
SRS dokument	email	x	x
SPMP	email	x	x
STP	email	x	x
SQAP	email	x	
SCMP	email	x	x
Izveštaji o učinku	email	x	
Projekt status	email	x	

Tabela 3. : Distribucija elemenata komunikacije

7.7 Plan dokumentacije

Dat je opis plana dokumentacije za dokumente koji će se isporučivati kao rezultati projekta. Svi dokumenti koji se prosleđuju se nalaze u *Tabela 7.7.1*. Na početku je data definicija kolona koje se nalaze u zaglavlju tabele:

- Dokument -- naziv dokumenta
- Verzija – važeća/trenutna verzija dokumenta
- Pripremio – osoba koja je pripremila dokument
- Potvrdio – osoba koja je odobrila dokument
- Datum pregleda – datum do kada dokument treba da bude spreman za odobravanje
- Lista za distribuciju – ko prima odobrenu verziju dokumenta

Dokument	Verzija	Pripremio	Potvrdio	Datum pregleda	Lista za distribuciju
Ugovor	1.1	Projekt menadžer	Projekt menadžer	15.05.2018.	Projekt menadžeri, CEO, repozitorijum
Projekt plan	1.7	Projekt menadžer	Projekt menadžer	18.05.2018.	Projekt menadžeri, CEO, repozitorijum
SRS dok.	1.2	Projekt menadžer	Projekt menadžer	19.05.2018.	Projekt menadžeri, CEO, projektni tim, repozitorijum
Test slučajevi	1.2	Testeri	Projekt menadžeri	20.05.2018.	Projekt menadžer, projektni tim, repozitorijum
Plan upravljanja rizicima	1.1	Projekt menadžer	Projekt menadžer	20.05.2018.	Projekt menadžer, CEO, repozitorijum
Plan upravljanja konfiguracijom	1.0	Projekt menadžer, CTO	Projekt menadžer	20.05.2018.	Projekt menadžer,

					CEO, repozitorijum
Plan upravljanja komunikacijom	1.2	Projekt menadžer	Projekt menadžer	21.05.2018.	Projekt menadžer, CEO, projektini tim, repozitorijum
Softver dokumentacija	1.2	Projektini tim	Projekt menadžer	22.05.2018.	Projekt menadžer, repozitorijum
Korisničko uputstvo	1.0	Projektini tim	Projekt menadžer	23.05.2018.	Projekt menadžer, korisnici, repozitorijum
Izveštaj o prihvatanju	1.1	Projekt menadžer	Vlasnik softvera, CEO	29.05.2018.	Projekt menadžer, CEO, repozitorijum

Tabela 7.7.1 : Plan dokumentacije

7.8 Plan za rešavanje problema

7.8.1 Izveštavanje problema

Svi problemi moraju biti prosleđeni projekt menadžeru korišćenjem forme koja je dogovorena u projektu. Kada je kompletirana, forma se prosleđuje elektronski putem preko email-a.

7.8.2 Uloge

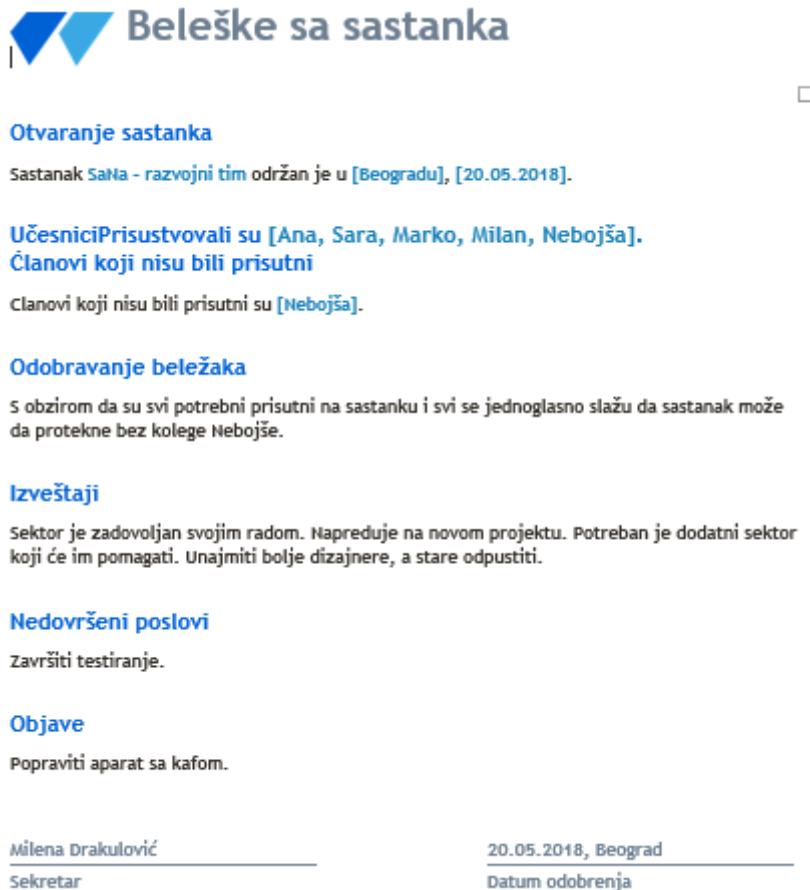
Data je *Tabela 7.8.2.1* koja prikazuje uloge članova tima koji učestvuju u rešavanju problema

Član tima	Uloga
Projekt menadžer	Prihvata izveštaje o problemima Organizuje sastanke Nadgleda proces rešavanja problema Dokumentuje i zatvara problem
SA	Predlaže rešenja problema Vodi proces rešavanja problema
Projektni tim	Predlažu rešenja problema Učestvuju na sastancima Aktivno učestvuju u procesu rešavanja problema

Tabela 7.8.2.1 : Članovi tima za rešavanje problema

7.8.3 Nedeljni sastanak

Jedan od načina vođenja računa o diskusijama vezane za naš sistem je pripremom izveštaja sa nedelnog sastanka. Vođenjem evidencije o sastancima možemo imati uvidaj u naš napredak Slika 7.8.3.1.



Slika 7.8.3.1-primer izveštaja sa nedelnog sastanka

Pisana forma ovakvog dokumenta, može se pronaći u prilogu

[SE325-SaraDavidovic1924-AnaSajic2773-IzvestajSaNedelnogSastanka.docx](#).

8. ZAKLJUČAK

Pirlikom sastavljanja dokumentacije, došle smo do zaključka da je prikupljanje dokumentacije prilično težak posao, jer projektant treba da zna šta traži, gde traži i da zna da li su podaci verodostojni.

Naša dokumentacija za projekat Sana je ispio uspešan, jer smo uspele da skupimo sve što nam treba. Mogle smo da upotrebimo stečeno znanje sa dva predmeta SE201-Uvod u softverko inženjerstvo I IT255-Veb sistemi 1.

9. LITERATURA

- [1] Cole A.: Runaway Project - Causes and effects, Software World, UK, Vol 26. No. 3, 1995.
- [2] SAFU, Rječnik: Ganttov grafikon ili gantogram
- [3] Dohn Kissinger, A Step-by-Step Approach for Planning Your Small Project, ssmall_projects_tep_by_step.pdf
- [4] Luckey T., Phillips J., Software Project Management For Dummies, Wiley Publishing, Inc, 2006
- [5] A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Project Management Institute, Newtown Square, PA. 1996,
- [6] Mavenlink blog, <http://blog.mavenlink.com/new-gantt-baselines-help-evaluate-the-success-of-your-projects-plan>

10. PRILOZI

- **Prilog 1:** Konceptualni model(IT255-Ana-Sajic-2773-SaraDavidovic1924-KonceptualniModel-.pdf)
- **Prilog 2:** Specifikacija zahteva softvera (SE201-PZ-Ana-Sajic-2773-SaraDavidovic1924.docx)
- **Prilog 3:** Izveštaj sa nedeljnog sastanka(SE325-SaraDavidovic1924-AnaSajic2773-IzvestajSaNedeljnogSastanka.docx)
- **Prilog 4:** Master Test SE325-MasterTestPlan-SaraDavidovićAnaSajić2773.docx

10.1. Reference

[1] "Function Point Languages Table ". [Online]. Dostupno:: <http://www.qsm.com/resources/function-point-languages-table> Posećeno: Maj. 10, 2018.

[2] "Function Point Calculator ". [Online]. Dostupno:
<http://groups.umd.umich.edu/cis/course.des/cis375/projects/fp99/main.html> Posećeno: Maj. 10, 2018.

[3] Prof. Dr. Dragan Domazet. Predavanja i Vežbe iz SE201-Uvod u softversko inženjerstvo. Izdanje za nastavnu godinu 2017/2018. Beograd: Univerzitet Metropolitani, 2017. Web. LAMS Univerziteta Metropolitani. Dostupno na: <http://lms.metropolitan.ac.rs>. Posećeno: 10 Maj, 2018.

[4] Prof. dr Ljubomir Lazić. Predavanja i Vežbe iz SE325 UPRAVLJANJE PROJEKTIMA RAZVOJA SOFTVERA. Izdanje za nastavnu godinu 2017/2018. Beograd: Univerzitet Metropolitani, 2017. Web. LAMS Univerziteta Metropolitani. Dostupno na: <http://lms.metropolitan.ac.rs>. Posećeno: 10 Maj, 2018.

[5] Prof. dr Ljubomir Lazić. Predavanja i Vežbe iz IT255 Veb sistemi 1. Izdanje za nastavnu godinu 2017/2018. Beograd: Univerzitet Metropolitani, 2017. Web. LAMS Univerziteta Metropolitani. Dostupno na: <http://lms.metropolitan.ac.rs>. Posećeno: 10 Maj, 2018.

10.2. Korišćeni softverski alati

- Power Designer
- MS project
- Function Point Calculator
- Function Point Calculator
- MS office