MJERENJE OPSEGA PROGRAMSKE POTPORE POMOĆU FUNKCIJSKI ORIJENTIRANE METRIKE

Maršić, Nikolina

Master's thesis / Specijalistički diplomski stručni

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:506086

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Informacijska tehnologija

NIKOLINA ŽABIĆ

Z A V R Š N I R A D

MJERENJE OPSEGA PROGRAMSKE POTPORE POMOĆU FUNKCIJSKI ORIJENTIRANE METRIKE

Split, listopad 2019.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Informacijske tehnologije

Predmet: Upravljanje kvalitetom i metrika

Z A V R Š N I R A D

Kandidat: Nikolina Žabić

Naslov rada: Mjerenje opsega programske potpore pomoću funkcijski orijentirane metrike

Mentor: mr.sc. Karmen Klarin, viši predavač

Split, listopad 2019.

Sadržaj

SAŽETAK 1

SUMMARY 1

1. UVOD 3

2. MATEMATIČKI MODEL ZA IZRAČUN FUNKCIJSKIH TOČAKA 4

2.1 Funkcijski orijentirana metrika 4

2.2 Funkcijske točke 4

2.3 Logička transakcija 5

2.3.1 Računanje funkcionalnosti logičke transakcije 6

2.3.2 Primjer računanja funkcijskih točaka 7

2.4 Opće karakteristike sustava 10

2.5 Postupak procjene funkcijskih točaka 13

3 RAZVOJ INFORMACIJSKOG SUSTAVA 15

3.1 Faze razvoja informacijskog sustava 15

3.2 Razvoj projekta 16

3.2.1 Razvoj projekta pomoću dijagrama UML jezika 17

4. ANALIZA POSLOVNOG SUSTAVA 18

4.1 Opis poslovnog sustava 18

4.2 Korisnički zahtjevi 18

4.3 Prikaz poslovnog sustava pomoću dijagrama 19

4.3.1 Dijagram obrasca uporabe 19

4.3.2 Scenarij obrasca uporabe 22

5. OBLIKOVANJE INFORMACIJSKOG SUSTAVA 26

5.1 Dijagram aktivnosti 26

5.2 Dijagram stanja 28

6. IZRADA APLIKACIJE 29

6.1 Relacijski model podataka 29
6.2 Priprema razvojnog okruženja 30
6.3 Obrasci aplikacije 30
6.4 Izvješća aplikacije 36
6.2 Prikaz procjene za novi projekt 41
7. ZAKLJUČAK 46
LITERATURA 47
POPIS SLIKA ..... **Pogreška! Knjižna oznaka nije definirana.**
POPIS TABLICA ..... **Pogreška! Knjižna oznaka nije definirana.**

SAŽETAK

Prije početka razvoja programske podrške potrebno je procijeniti i izmjeriti koliko je programsko rješenje složeno i koje će vrijeme biti potrebno za njegovu realizaciju. Prvi dio zadatka mjerenja opsega programske potpore pomoću funkcijski orijentirane metrike bio je izračunati broj funkcijskih točaka za testne projekte koji su gotovi i imaju unesene logičke transakcije. U drugom dijelu zadatka na osnovu procijenjenih vrijednosti napravljeno je predviđanje za novi projekt.

Tijek procjene se nastavlja nakon što su u bazi pohranjeni i izračunati podatci, te je stvorena mogućnost da se za novi uneseni projekt može procijeniti broj funkcijskih točaka. S dobivenom procijenjenom funkcijskih točka voditelj projekta može ići u daljnje pregovaranje o cijeni projekta, potrebnom vremenu, radnom snagom itd.

Nakon što se novi projekt završi i u bazi su sve unesene LT, popravlja se statistika tj. vrijednosti procijenjenih FT postaju sve bliže stvarnim vrijednostima funkcijskim točkama.

Ključne riječi: razvoj programa, metrika, funkcijske točke

SUMMARY

MEASURING SOFTWARE SUPPORT BY FUNCTIONALLY ORIENTED METRIC

Prior to the beginning of software development, it is necessary to evaluate and measure how complex the software solution is and how long it is going to take to implement it. The first part of the task of measuring the scope of software using a function-oriented metrics was to calculate the number of function points for test projects that are completed and have logical transactions entered. In the second part of the task, an estimate for the new project was made based on the estimated values.

The assessment process continues after the data have been stored and calculated in the database, and the possibility was created that the number of function points can be

estimated for the newly entered project. With the obtained function point estimate, the project manager can further negotiate the project cost, time required, manpower, etc.

After the new project is completed and all the LTs have been entered into the database, the statistics are improved, i.e. the values of the estimated FTs become closer to the actual values of the function points.

Keywords: program development, metrics, function points

1. UVOD

Kako bi se dobila potpuna i jasna slika o temi: „Mjerenje opsega programske potpore pomoću funkcijski orijentirane metrike“ koja će biti obrađena u nastavku rada bitno je za početak upoznati se pojmom: „mjerenja i metrike“.

U današnje vrijeme na tržištu se praktički svakodnevno pojavljuju novi proizvođači s novim programskim rješenjima. Kako bi proizvođač zadobio željeni udio na tržištu, on mora ponuditi kvalitetan proizvod po pristupačnoj cijeni. U tom slučaju da bi postigao dobar omjer cijene i kvalitete potrebno je gubitke raznih resursa svesti na minimum. Tako se prije početka razvoja programske podrške mjeri i procjenjuje koliko je programsko rješenje složeno i koje će vrijeme biti potrebno za njegovu realizaciju.

Kada se mjerenje vrši na ovakav način, dakle prije nego je proces realizacije započeo, tada govorimo o procjeni opsega programskog rješenja.

Praktični dio ovog završnog rada podržava funkcionalnost evidentiranja FT nakon što je projekt završen, a uz to podržava i već spomenutu funkcionalnost procjene opsega.

Do danas je razvijen veliki broj metrika, a najpoznatija i najkorištenija je metrika funkcijskih točaka koju je razvio Albrecht 1979. godine. Navedena metrika bit će korištena i u ovom završnom radu.

U poglavlju „Funkcijski orijentirana metrika “ ovoga rada detaljno je objašnjena metrika funkcijskih točaka. Iznesen je kratak povijesni prikaz nastanka i razvoja metrike.

Treće poglavlje odnosi se na razvoj informacijskog sustava. Pomoću faza razvoja informacijskog sustava slijedno je prikazan razvoj zadanog projekta.

U četvrtom poglavlju pod nazivnom „Analiza informacijskog sustava“ opisani su korisnički zahtjevi i tijek razvoja projekta pomoću dijagrama obrasca uporabe i scenarija obrasca uporabe.

Kroz peto poglavlje „Oblikovanje informacijskog sustava“ opisano je upravo oblikovanje sustava pomoću UML dijagrama aktivnosti i dijagrama stanja.

Šesto poglavlje odnosi se na izradu korisničke aplikacije i tehnologiju koja je korištena u samoj realizaciji. Opisan je također i relacijski model podataka. Uz to predstavljena je i funkcionalnost aplikacije.

2. MATEMATIČKI MODEL ZA IZRAČUN FUNKCIJSKIH TOČAKA

Programsko rješenje je pojam za mjeru programskih značajki koje se mogu prebrojavati. Programska mjerenja su važna jer uključuju mjerenje performansi softvera, planiranje radnih sati, mjerenje produktivnosti, provjera konzistentnosti zahtjeva i mnoga druga mjerenja. Programska rješenja povezana su s četiri funkcije upravljanja: planiranjem, organizacijom, kontrolom i napretkom.

Već u uvodu upoznali smo se s tim da se mjerenje opsega programske potpore može podijeliti u dvije kategorije:

- Direktno mjerenje
- Indirektno mjerenje

Direktno mjerenje odnosi se na cijenu proizvoda, uloženi rad, programske retke, brzinu izvođenja i slično.

Indirektno mjerenje odnosi se na funkcionalnost proizvoda, kvalitetu, složenost, učinkovitost, pouzdanost i slično.

2.1 Funkcijski orijentirana metrika

Funkcijski orijentirana softverska metrika je indirektna mjera softvera, a to znači da se funkcionalnost ne može direktno mjeriti. Neovisna je o programskom jeziku koji se koristi za realizaciju softvera. Funkcijski orijentiranu metriku prvi je predložio Albrecht 1979. godine koji je za danu metriku predložio također i mjeru, nazvanu funkcijska točka.

Mjeru funkcijskih točaka Albrecht je izveo pomoću empirijskog odnosa utemeljenog na izravnim mjerama domene informacijskog softvera i procjene njegove složenosti.

2.2 Funkcijske točke

Funkcijska točka (FT, engl. *function point*) je jedinica mjere pomoću koje se može izraziti količina poslovnih funkcionalnosti koje će korisniku pružiti programsko rješenje.

Osnova za primjenu FT su funkcionalni korisnički zahtjevi, gdje se broj FT pridjeljuje ovisno o složenosti zahtjeva.

Do danas je razvijeno nekoliko desetaka metoda za mjerenje funkcijskih točaka, od kojih je, uz IFPUG metriku, tek nekoliko postiglo široku primjenu u praksi, a to se prije svega odnosi na *Mark* II (Mk II) funkcijske točke i COSMIC funkcijske točke. Metodu funkcijskih točaka *Mark* II objavio je Charles Symons 1988. godine.

Nedostatak koji je Symons vidio proučavajući Albrechtov model, bilo je to što taj model ne može mjeriti funkcionalnost svih tipova softvera. Također, Symons je smatrao da bi datoteke koje se koriste u više transakcija trebalo brojati više puta, a ne samo jednom.

Metoda „Točke slučajeva korištenja“ također je jedna od tehnika procjene softvera koja se koristi za mjerenje veličine softvera pomoću slučajeva korištenja tehnike UML (engl. *Unified Modeling Language*). Rezultat procjene metodom TSK vrednuje se preko jedinice vremena potrebne da bi se izradila jedna TSK. Razvijena je od strane Gustava Karnera 1993. godine. Njegov prijedlog mjere bio je faktor od 20 čovjek/sati za svaku točku.

2.3 Logička transakcija

Logička transakcija (LT) može se definirati kao jedinstvena kombinacija slijeda funkcionalnosti ulaz → proces → izlaz, Slika 8. Pokreće ju jedinstveni događaj, a to može biti potreba za informacijom ili bilo koji drugi događaj značajan za korisnika. Za već gotove programe kao što je u ovome radu i slučaj mogu se odrediti osnovni tipovi LT na način da se promatraju osnovni tipovi sučelja programskog rješenja. U nastavku je jedan takav primjer podjele:

- Modul (maska za pregled, unos, ažuriranje i brisanje) – to je skupina logičkih transakcija vezanih uz interakciju korisnika sa bazom podataka (BP). Ova vrsta LT koristi se u svrhu pregleda, unosa, ažuriranja ili brisanja podataka iz baze.
- Izvješće – spada u skupinu LT veznih uz ispis podataka iz baze, s prikazom na ekran, pisač ili datoteku. Postoje slučajevi kada se izvješće može sastojati od više LT, stoga treba i o tome voditi računa.
- Obrada – skupina LT koje se odvijaju bez vanjskog utjecaja. Pokreće ih neki događaj, a on može biti uzrokovan vremenski, akcijom korisnika ili slično.

U ovom slučaju obrade su uglavnom složene, pa ih treba raščlaniti na jednostavnije LT.

- Zajednički proces – ova vrsta LT spada u one koje se pojavljuju više puta u programu.

Na ovaj način su definirani tipovi logičke transakcije. Dakle osim tipova postoje i vrste LT, a to se upravo odnosi na pregled, unos, ažuriranje i brisanje. Modul kao tip LT podržava sve vrste LT, kao i obrada. Kod izvješća je slučaj da ovaj tip LT podržava samo pregled kao vrstu logičke transakcije, jednako kao i zajednički proces.

2.3.1 Računanje funkcionalnosti logičke transakcije

Za prikupljanje podataka važno je izmjeriti svaku logičku transakciju gotovog programa. Proces računanja odvija se prema zadanom matematičkom modelu. Kako bi se uspješno obavilo brojanje funkcionalnosti gotovog programa definirani su koraci prema kojima se obavlja brojanje, a navedeni su u nastavku:

- Prepoznati proces, na osnovu projektne dokumentacije.
- Raščlaniti procese na logičke transakcije i odrediti tip za svaku logičku transakciju. Tipovi LT kako je već spomenuto su: Modul, Izvješće, Obrada i Zajednički proces.
- Zatim za svaku LT odrediti broj entiteta (N_E), broj ulaznih (N_I) i broj izlaznih (N_O) atributa.
- Na kraju za svaku LT potrebno je izračunati broj funkcijskih točaka.

Funkcionalnost jedne logičke transakcije (engl. *Information Processing Size*, IPS) računa se prema formuli:

$$IPS_T = 0{,}58 * N_I + 1{,}66 * N_E + 0{,}26 * N_O \qquad (1)$$

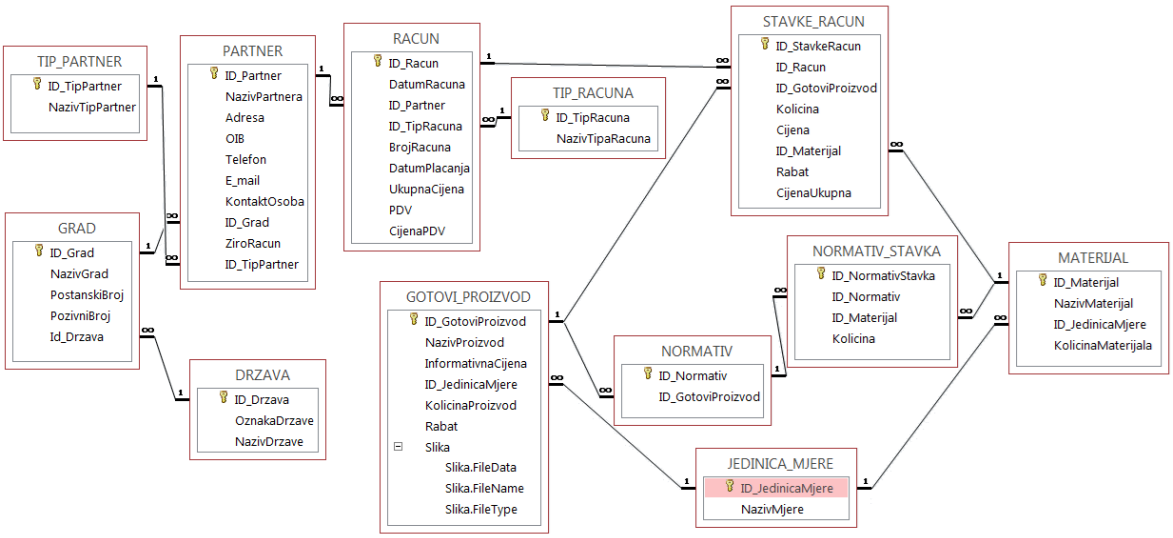
Koeficijenti koji su navedeni u formuli (1) su industrijski standard, a predstavljaju težinske faktore. Ulazni podaci za navedenu formulu su:

- broj tipova ulaznih podataka (N_I)
- broj referenciranih tipova entiteta (N_E)
- broj tipova izlaznih podataka (N_O)

2.3.2 Primjer računanja funkcijskih točaka

U nastavku će konkretnim primjerom biti potkrijepljena priča o opisanom procesu za računanje funkcionalnosti logičke transakcije. Primjer je preuzet iz završnog rada [8], riječ je o Access aplikaciji pod nazivom „Proizvodnja kovanih ograda“.

Slika 1. prikazuje pripadajući model baze podataka nad kojim će biti izvedeni koraci prema kojima se obavlja brojanje.

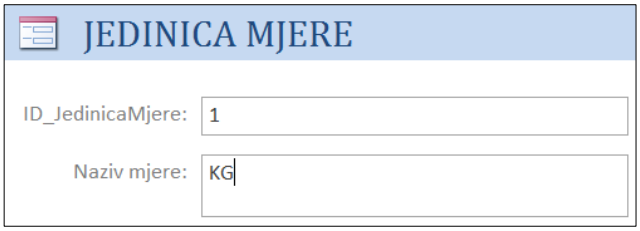


Slika 1: Model baze podataka za projekt "Proizvodnja kovanih ograda"

Na osnovu projektne dokumentacije prepoznati su procesi i za primjer su izabrane tri unosne maske i jedno izvješće.

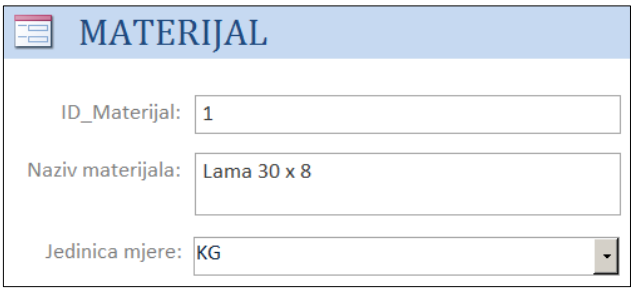
Slika 2. prikazuje modul jednostavni kao tip i podtip programskog rješenja tj. tip logičke transakcije, a koji služi za prikaz jedinica mjere. ID_JedinicaMjere je broj koji se automatski generira. Naziv mjere je vrijednost koja se unosi.

Modul kao tip logičke transakcije podržava sve vrste LT. Na Slika 6. Su prikazane prebrojane vrijednosti *input-a*, entiteta i *output-a*, te su prema formuli (1) su izračunate vrijednosti pojedine vrste logičke transakcije.



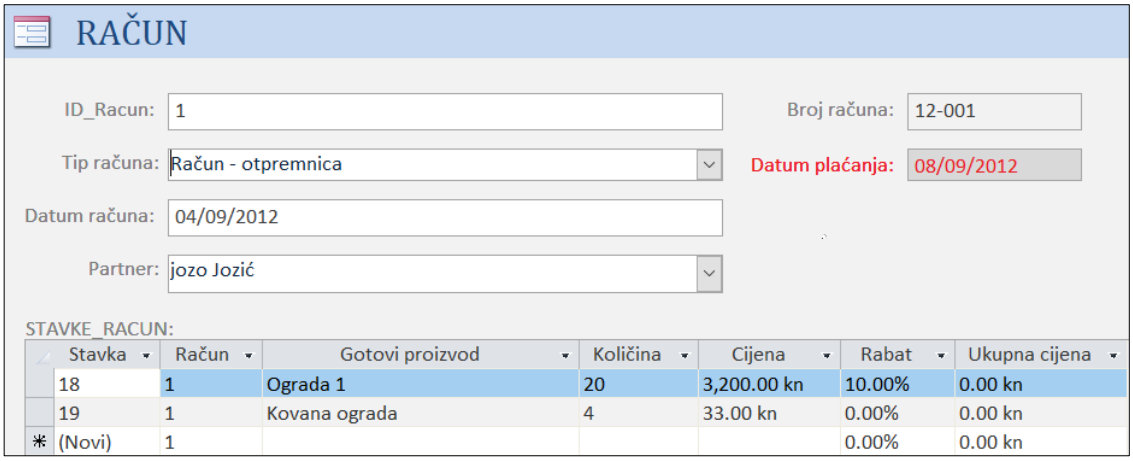
Slika 2: Unosna maska „JEDINICA MJERE“

Na Slika 3. prikazana je također forma unosne maske naziva „MATERIJAL“. Kao tip i podtip sučelja u Tablica 1.**Pogreška! Izvor reference nije pronađen.** programskog rješenja je modul jednostavni. Slika 6. prikazuje broj *input-a*, entiteta i *output-a* koji su pridruženi pojedinoj vrsti logičke transakcije.



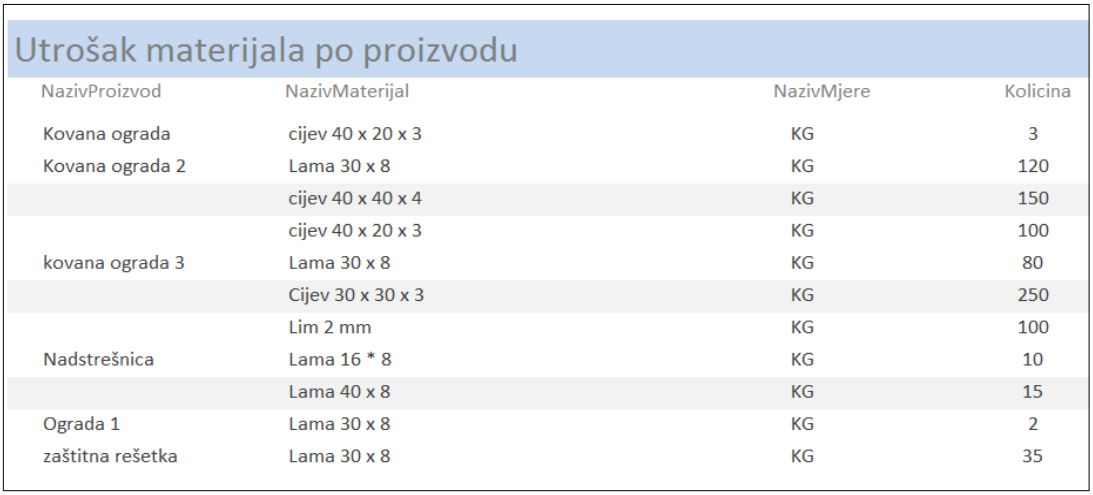
Slika 3: Unosna maska "MATERIJAL"

Forma „RAČUN“ prikazana na Slika 4. prema tipu i podtipu logičke transakcije je modul složeni što bi značilo da ima složenu obradu podataka (generiranje stavki računa). Izračunate vrijednosti svake pojedine vrste logičke transakcije kao i pridruženi brojevi *input-a*, entiteta i *output-a* su prikazani na Slika 6.



Slika 4: Unosna maska "RAČUN"

Izvješće o utrošku materijala po proizvodu prikazano na Slika 5. Prema tipu logičke transakcije spada u „Izvješća“ u ovome slučaju izvješće prosječno. Poznato je da se odnosi na podtip logičke transakcije „prosječno“ jer se podaci odbijaju iz više entiteta, ali nema obrade podataka. Slika 6. nam prikazuje broj *input-a*, entiteta i *output-a*. Vrijednosti su unesene samo za vrstu LT „Pregled“ obzirom da je u radu već ranije spomenuto kako tip LT „Izvješće“ kao vrstu LT podržava samo pregled.



Slika 5: Izvješće "Utrošak materijala po proizvodu"

Ukupan zbroj svih logičkih transakcija predstavlja ukupnu funkcionalnost sustava (IPS). Formulom prikazano na sljedeći način:

$$IPS = \Sigma\, IPS_T \qquad (2)$$

U ovom primjeru ukupna funkcionalnost sustava tj. ukupna funkcionalnost izabranih unosnih maski i izvješća prikazana je na Slika 6. pod nazivom „Suma“.

		N input	N entitet	N output	FT
Jedinica mjere	R - pregled	2	1	2	3.34
	C - unos	2	1	2	3.34
	U - ažuriranje	1	1	2	2.76
	D - brisanje	1	1	1	2.5
Materijal	R - pregled	3	2	3	5.84
	C - unos	3	3	3	7.5
	U - ažuriranje	2	3	3	6.92
	D - brisanje	2	2	2	5
Račun	R - pregled	1	2	13	7.28
	C - unos	6	2	13	10.18
	U - ažuriranje	4	2	13	9.02
	D - brisanje	1	1	1	2.5
Utrošak materijala po proizvodu	R - pregled	1	5	4	9.92
	C - unos	0	0	0	0
	U - ažuriranje	0	0	0	0
	D - brisanje	0	0	0	0
				Suma:	**76.1**

Slika 6: Način mjerenja FT za primjer "Proizvodnja kovanih ograda"

2.4 Opće karakteristike sustava

Izračunavanje funkcijskih točaka radi se u dvije faze:

- Računanje funkcionalnosti LT
- Računanje općih karakteristika sustava

Za izračunavanje funkcijskih točaka osim mjerenja ukupne funkcionalnosti sustava (IPS) pokazano na prethodnom primjeru Slika 6., na ukupan broj FT imaju utjecaja i opće karakteristike sustava, Slika 7. Svaku pojedinu karakteristiku potrebno je ocijeniti vrijednošću od 0-5. Vrijednosti pokazuju utjecaj na sustav:

- 0 – nema utjecaja,
- 1 – neznatan utjecaj,
- 2 – umjereni utjecaj,

- 3 – prosječan utjecaj,
- 4 – značajan utjecaj,
- 5 – potpun utjecaj.

Zbrajanjem svih ocjena za svaku pojedinu karakteristiku dobije se broj koji predstavlja ukupni faktor utjecaja TDI (engl. *Total Degree of Influence*) za cijeli sustav i prikazan je na Slika 7 označen kao „UKP“. Na osnovu izračunatog faktora TDI može se izračunati faktor prilagodbe tehničke složenosti TCA (engl. *Technical Complexity Adjustment*) prema formuli (3).

$$TCA = 0{,}65 + (0{,}01 * TDI) \qquad (3)$$

U jednadžbi se koriste koeficijenti koji su industrijski standard.

Nakon izračunate ukupne funkcionalnosti sustava (IPS) prema formuli (2) i faktora prilagodbe tehničke složenosti (TCA) prema formuli (3) može se izračunati ukupan broj funkcijskih točaka (FT) za cijelo programsko rješenje prema formuli (4), a na Slika 7. izračunati broj prikazan je pod nazivom „UnFT“.

$$FT = IPS * TCA \qquad (4)$$

Za cijelo programsko rješenje „Proizvodnje kovanih ograda“ ukupan broj funkcijskih točaka prema formuli (4) prikazano je na Slika 7.

LOGICKA_TRANSAKCIJA

FUNKCIJSKE TOČKE PROJEKTA

14 October 2019
15:48:57

PROJEKT **RADIONICA KOVANIH OGRADA**

KORISNIČKI ZAHTJEVI	STVARNE FT	PROCJENJENE FT
Unos matrijala	43.9	42
Ispis otpremnice	11.96	10
Stanje na skladištu	19.16	10
Unos novog partnera	56.02	42
Unos materijala	31.32	24
Narudžba materijala	48.9	42
Unos normativa	15.46	42
Pregled stanja materijala	10.02	10
Pregled proizvoda	6.04	10
Izrada narudžbenice za materijal	55.2	42
Izrada računa	57.4	42
Izrada izvješća	14.5	14
Izrada ponude	32.84	32
Unos gotovog proizvoda	35.9	42
Popis artikala	6.94	10
UnFT (Ukupno neprilagođenih FT)	**445.58**	**414.00**
FT = UnFT * (0.65 + (0.01* UKP))	**521.32**	

OPĆE KARAKTERISTIKE PROJEKTA

UKUPNA OCJENA OPĆIH KARAKTERISTIKA PROJEKTA

Prijenos/Komunikcija podataka (Dana communication)	5
Raspodijeljena obrada (Distributed dana processing)	4
Konfiguriranje (Heavily used configuration)	3
Učestalost transakcija (Transaction rate)	5
Interaktivni unos podataka (On-line dana entry)	5
Dizajn obzirom na učinkovitost korisnika (End-user efficiency)	5
Dinamičko ažuriranje (On-line update)	4
Složena obrada (Compplex processing)	3
Korisnost u drugim aplikacijama (Reusability)	5
Lakoća uvođenja (Installation ease)	4
Lakoća promjene (Operational ease)	5
Raširenost (Multiple sites)	4
UKP (Ukupna ocjena općih karakteristika projakta)	**52**

Slika 7: Ukupan broj FT za projekt „Proizvodnja kovanih ograda“

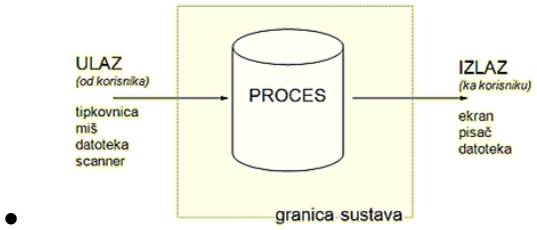
2.5 Postupak procjene funkcijskih točaka

U prethodnom dijelu poglavlja objašnjeno je mjerenje opsega brojanjem funkcijskih točaka nakon što je programsko rješenje napravljeno i jasno je koje LT postoje i od kojih se elemenata sastoje. Nastavak priče slijedi i bit će opisana procjena FT za novi projekt.

Uobičajeno je da se procjena odvija na početku razvoja IS-a. Početak može biti definiram stvarnim početkom, odnosno samim ugovaranjem projekta, dok u nekim slučajevima početak bude pomaknut u fazu analize ili oblikovanja.

Kada se procjena radi prilikom ugovaranja projekta potrebno je puno više znanja i vještine da se obavi procjena. Stoga nije slučajno da se s procjenom počinje u nekoj od idućih faza razvoja kada je znatno lakše raditi procjenu obzirom da je dokumentirano puno više detalja koji mogu pomoći u brojanju elemenata LT. Procjena FT je subjektivna. Tako se aktivnost prikupljanja podataka kako bi se izvela potrebna metrika može promotriti u dvije razine:

- Odrediti osnovne tipove funkcionalnosti (ulaz, izlaz i proces) prikazano na Slika 8. Ukoliko se mjerenje FT radi u nekoj od kasnijih faza recimo u fazi analize razvoja programske potpore u tom slučaju su ulazi, izlazi i procesi, prikazano na Slika 8. određeni modelima podataka i procesa korištenih u metodi razvoja (primjerice, EV model).
- Često se procjena zahtjeva i ranije, no tada su modeli podataka i procesa općeniti i bez dovoljno osnovnih parametara na kojima bi se mogla primijeniti metoda.



Slika 8: Tipovi funkcionalnosti za metodu analize FT

- Odrediti tipove izvedenih grupa funkcionalnosti na osnovu povratnih informacija tj. mjerenja nakon izrađenog programa. U ovom slučaju se izbjegava mjerenje funkcionalnosti svake pojedine logičke transakcije u trenutku procjene za novi

program. Naglasak je u ovom slučaju na tome da se prije procjene napravi što više mjerenja osnovnih funkcionalnosti za gotove proizvode kako bi izvedene grupe funkcionalnosti bile što vjerodostojnije. Jedan od načina primjene opisanog postupka je taj da se u procjeni koriste iskustva prethodnih brojanja FT. To bi značilo da veće iskustvo i veći broj procijenjenih projekata daje bolju statistiku za nove procjene.

Rezultati mjerenja gotovih programskih rješenja grupirani su u nekoliko grupa i kombinacije tih grupa, Tablica 1. daju dovoljno raznolike tipove funkcionalnosti. Svakom tipu funkcionalnosti pridružen je prosječan broj FT izračunat na osnovu odgovarajućih LT gotovih projekata Slika 28. Ove prosječne vrijednosti su različite za svakog korisnika aplikacije.

Tablica 1: Tablica tipova i složenosti logičke transakcije

Tip i složenost logičke transakcije	***Opis***
Modul jednostavni	Novi modul koji uključuje više entiteta i na sebi NEMA nikakvu obradu podataka.
Modul prosječni	Modul prosječni uključuje više entiteta i na sebi ima obradu podataka.
Modul složeni	Novi složeni modul se sastoji od više logičkih cjelina i na sebi ima složenu obradu podataka.
Obrada jednostavni	Jednostavna obrada podataka.
Obrada prosječni	Modul koji uključuje obradu podataka, a odnosi se na generiranje vrijednosti .
Obrada složeni	Modul koji uključuje složenu obradu podataka.
Izvješće jednostavni	Izvještaj koji koristi podatke iz jednostavne forme. U većini slučajeva odnosi se na ispis podataka iz baze.
Izvješće prosječni	Izvještaj koji koristi podatke iz jednostavne forme i NEMA nikakve obrade podataka.
Izvješće složeni	Izvještaj koji koristi podatke iz jednostavne forme i IMA obradu podataka u prikazu. (ukupna cijena, brojFT …)
Zajednički proces jednostavni	Novi modul koji na sebi sadrži *combobox*, ili neku drugu od jednostavnih kontrola (*textbox…*)
Zajednički proces složeni	Novi složeni modul uključuje jedan entitet. U ovu kategoriju ulaze i složene kontrole koje se pojavljuju na nekoliko mjesta.

3 RAZVOJ INFORMACIJSKOG SUSTAVA

Svaki sustav koji se izrađuje nalazi se u nekom širem sustavu čiji je dio i s kojim je u relaciji. Jedan dio sustava koji pak nije obuhvaćen većim sustavom naziva se okolinom sustava. Veza između sustava i okoline predstavlja se ulazima u slučaju kada okolina predaje sustavu informacije, odnosno izlazima kada ih sustav predaje okolini.

Općenito za informacijski sustav (IS) može se reći da je jedan od podsustava organizacijskog sustava, a svrha mu je: prikupljanje, obrada, pohranjivanje i distribucija informacija, koje su potrebne za praćenje rada i upravljanje sustavom. Jasno se može primjetiti da se informacijski sustav sastoji od programa koji prihvaćaju informacije o stanju organizacijskog sustava i upisuju ih u bazu podataka, te da se sastoji i od programa koji čitaju podatke iz BP i kreiraju izvješća.

3.1 Faze razvoja informacijskog sustava

Praćenje razvoja informacijskog sustava organizirano je kroz slijed zadataka i aktivnosti. Prema tome izgradnju informacijskog sustava može se podijeliti u nekoliko faza:

- Planiranje i definiranje zahtjeva za projektiranje i izgradnju IS-a
- Analiza poslovnog sustava
- Projektiranje novog sustava (oblikovanje i dizajn)
- Izrada IS-a
- Testiranje i uvođenje sustava u primjenu
- Održavanje i prilagođavanje informacijskog sustava

Planiranje i definiranje zahtjeva važno je za uspostavu ciljeva i vizije informacijskog sustava.

Faza analize je bitna za otkrivanje zahtjeva korisnika prema informacijskom sustavu i odnosi se na planiranje kako te zahtjeve ostvariti.

Projektiranje novog sustava odnosi se pak na oblikovanje, tj. uspostavlja dizajn i definira oblik budućeg sustava.

Izrada je faza u kojoj se informacijski sustav izrađuje na način da bi se osiguralo ponašanje sustava u skladu sa zahtjevima.

Korak testiranja i uvođenja sustava u primjenu testira sustav za buduće korištenje.

Održavanje je prilagođavanje novog informacijskog sustava. Uvode se novi poslovni sustavi ili mijenjaju postojeći. Bez pravilnog održavanja novi informacijski sustav ne može biti koristan za poslovnu organizaciju.

3.2 Razvoj projekta

Razvoj projekta započinje planiranjem, izborom programskih alata, baze podataka, a u obzir se mogu uzeti i postojeći programski proizvodi. Nakon završenog procesa prikupljanja zahtjeva od korisnika velikim dijelom nam je poznato što korisnik očekuje od sustava, pa utoliko se može započeti sa razvojem.

Projekt koji će biti opisan u nastavku rada kao program za izradu koristi Access. Pomoću Accessa omogućena je izrada baze podataka. Dobro dizajnirani predlošci uvelike olakšavaju stvaranje vlastite BP. Upiti omogućuju jednostavno pronalaženje podataka koji su nam u datom trenutku potrebni. Također Access nudi mogućnost stvaranja obrazaca za jednostavan unos podataka, a zatim se ti podatci mogu grupirati i prikazati u izvješćima. Nedostatak je prilična ograničenost prilikom dizajniranja sučelja ili nekih većih funkcionalnosti, no za jednostavnije projekte i aplikacije svakako je dobar izbor.

Uz forme kako je već spomenuto postoji mogućnost stvaranja upita (engl. *Query*) koji prikazuje podatke iz tablica na definirani način u *Access* programu. Sve polazi od popunjavanja tablica s atributima te prikazivanjem veza između istih.

Aplikacija se sastoji od manje baze podataka. U bazu podataka su pohranjeni podaci o projektima, korisničkim zahtjevima, općim karakteristikama, logičkim transakcijama i broju funkcijskih točaka.

Visual Basic kao programski jezik korišten je prilikom povezivanja formi i definiranja pojedinih funkcionalnosti aplikacije. Detaljnije o tome, kao i grafički primjeri, opisani su u poglavlju „Izrada aplikacije“.

3.2.1 Razvoj projekta pomoću dijagrama UML jezika

Ujedinjeni jezik za modeliranje (engl. *Unified Modeling Language*, kraće: UML), je skup grafičkih prikaza, jezik opće namjene koji se koristi za modeliranje računalnih sustava. Računalni sustav modelira se raznim dijagramima, od kojih se svaki koristi za prikaz sustava iz drukčije perspektive.

Dijagrami unutar UML-a mogu se podijeliti na statičke i dinamičke. Statički dijagrami ne uzimaju u obzir vremensku komponentu sustava, nego daju sliku cijelog sustava ili pojedinih dijelova u nekom trenutku. Kod dinamičkih dijagrama težnja je uključiti međudjelovanje sudionika i vremensku komponentu u opis sustava kako bi se modelirali sljedovi događaja unutar sustava.

Pomoću slijedećih UML tehnika napravljeno je modeliranje sustava, a to su: dijagram obrazaca uporabe (engl. *Use – case diagrams*), sekvencijski dijagram (engl. *Sequence diagram*) i dijagram aktivnosti (engl. *Activity diagram*).

4. ANALIZA POSLOVNOG SUSTAVA

Analiza poslovnog sustava je proces ustanovljavanja poslovnih zahtjeva i resursa informacijskog sustava potrebnih za odvijanje poslovnih procesa. Cilj analize je definirati detaljni opis zahtjeva prema informacijskom sustavu (zahtjeve korisnika).

4.1 Opis poslovnog sustava

Mjerenje opsega poslovnog sustava bazirano je na funkcijski orijentiranoj metrici i ima za cilj prikazati funkcionalnost proizvoda, kvalitetu i složenost proizvoda uz pomoć funkcionalnih mjerenja tj. metode mjerenja nazvane funkcijske točke.

Funkcijski orijentirana metrika temelji se na mjerenju funkcionalnosti isporučenog programskog proizvoda. Planira se dobiti uvid u opseg gotovog proizvoda. Može se koristiti i kao pomoć u upravljanju i kontroli razvoja programske potpore, ali isto tako i za prilagođavanje i/ili poboljšavanje od faze planiranja pa sve do isporuke projekta.

Izradom informacijskog sustava ove vrste može se doprinijeti aktivnostima i odlukama u upravljanu projektom.

Brojanje funkcijskih točaka nakon što je programsko rješenje gotovo je objektivno, budući nam je jasno koje LT postoje i od kojih se elemenata sastoje.

Poslovni sustav mjerenja opsega programske potpore podržan je aplikacijom. Aplikacija će korisnicima dati uvid u rezultate mjerenja gotovih programskih rješenja koja su grupirana u nekoliko skupina, a potom kroz razne kombinacije tih skupina ujedno i uvid u razne tipove funkcionalnosti. Rezultat prikaza je iskustvo prethodnih brojanja FT, što ujedno znači da bogatije iskustvo i veći broj proizvoda će dati bolju statistiku za nove procjene.

Najvažniji dio ovog poslovnog sustava je upravo statistika. Gdje će se nakon svakog brojanja FT moći poboljšati statistika i regrupirati tipove funkcionalnosti.

4.2 Korisnički zahtjevi

Aplikacija treba korisniku omogućiti slijedeće informacije:

- Unos podataka o projektu, evidenciju projekta.
- Unos korisničkih zahtjeva za projekt.
- Unos pripadajućih logičkih transakcija za pojedini korisnički zahtjev.
- Unos, ažuriranje, brisanje općih karakteristika po pojedinom projektu.
- Dodjeljivanje odgovarajućih općih karakteristika pojedinom projektu.
- Unos ocjene za svaku odabranu opću karakteristiku.
- Definiranje tipa i podtipa za svaku logičku transakciju.
- Izračun prosječne vrijednosti za svaku pojedinu logičku transakciju.
- Izračun neprilagođenih funkcijskih točaka, kao i procjenu funkcijskih točaka za novi projekt i pripadajuće korisničke zahtjeve.
- Omogućiti pregled svih projekata sa izračunatim vrijednostima procijenjenih i stvarnih funkcijskih točaka.
- Popisa općih karakteristika i korisničkih zahtjeva za određeni projekt.
- Prikaz izračunatih funkcijskih točaka uz popis korisničkih zahtjeva i općih karakteristika uz vrijednosti procijenjenih i stvarnih funkcijskih točaka.
- Popis tipova i podtipova LT i pripadajućih prosječnih FT.

4.3 Prikaz poslovnog sustava pomoću dijagrama

Da bi poslovni procesi bili dobro definirani i modelirani potrebno je odabrati odgovarajući alat za modeliranje. U nastavku će kroz razne dijagrame biti grafički predočene aktivnosti poslovnog sustava.

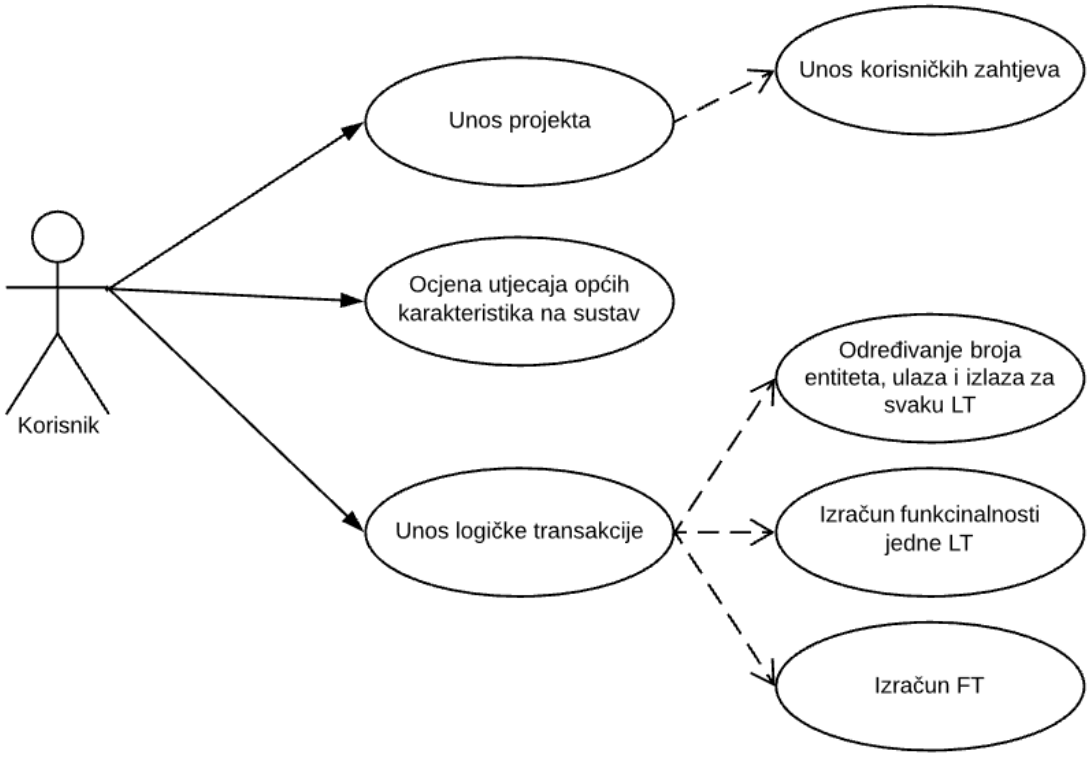
4.3.1 Dijagram obrasca uporabe

Dijagram obrasca uporabe koristi se kako bi se prikazalo ponašanje sustava. Obrasci uporabe također mogu poslužiti definiciji ponašanja korisnika pri komunikaciji sa sustavom.

Ponašanje sustava kroz obrasce uporabe opisano je pomoću obrasca uporabe (engl. *use case*), sudionika koji predstavljaju apstrakciju korisnika sustava, ili neki drugi vanjski sustav, vezama među sudionicima, te pomoću scenarija koji je tekstualni opis obrasca uporabe.

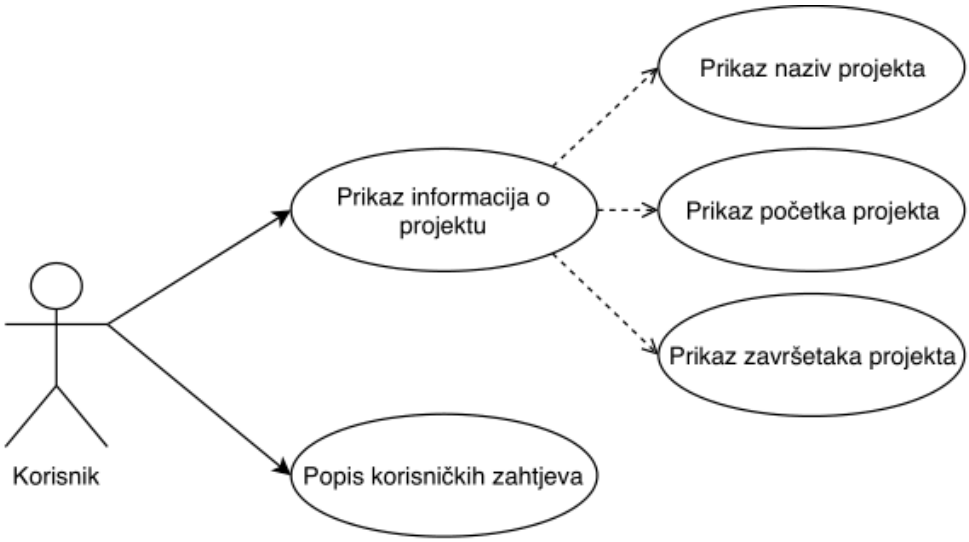
Sudionik u dijagramu obrasca uporabe je vanjski entitet koji je direktno povezan sa sustavom. Jedan sudionik može sudjelovati u više obrazaca uporabe. Sve akcije u sustavu započinje sudionik, on direktno pokreće neke od aktivnosti sustava, ili se pojedina aktivnost odražava direktno na sudionika.

Na Slika 9prikazan je osnovni dijagram obrasca uporabe za sustav Procjene opsega programske potpore. Dijagramom su definirani osnovni obrasci uporabe i sudionik sustava.



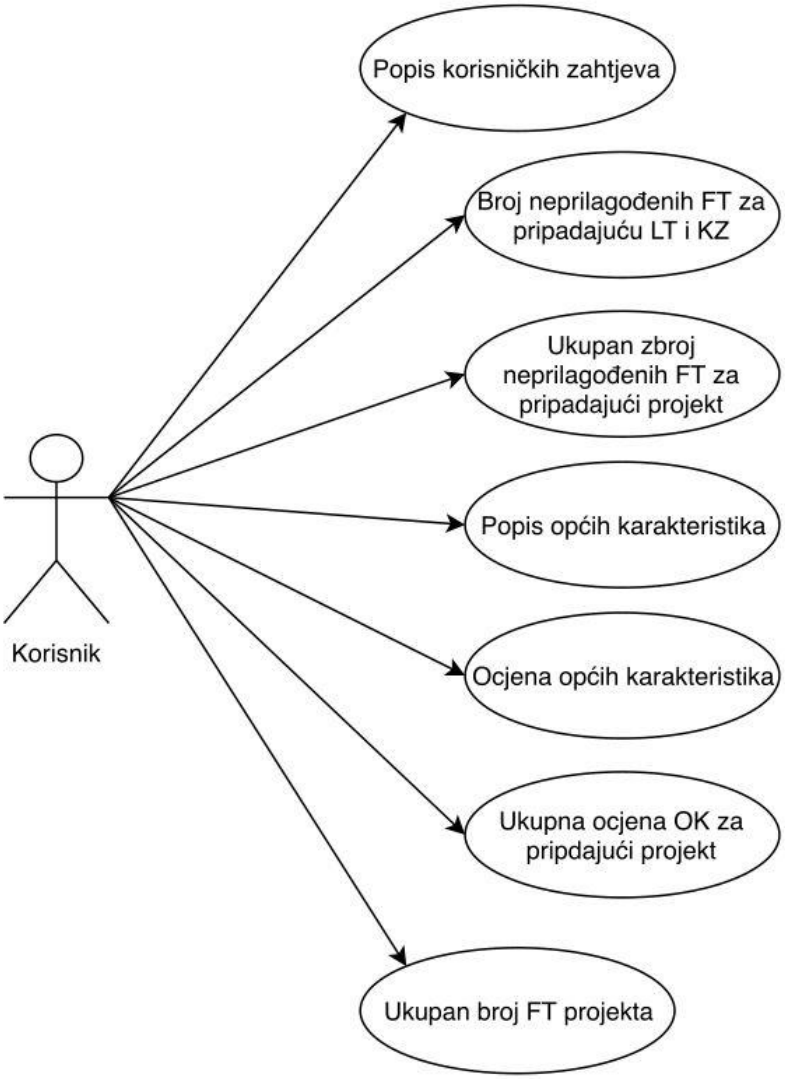
Slika 9: Dijagram obrasca uporabe za mjerenje opsega programske potpore

Slika 10. prikazuje dijagram obrasca uporabe za izvješće „Korisnički zahtjevi za projekt“. Otvaranjem izvješća korisnik ima uvid u informacije o nazivu te početku i završetku projekta kao i njegovim pripadajućim korisničkim zahtjevima.



Slika 10: Dijagram obrasca uporabe za izvješće „Funkcijske točke projekta“

U nastavku na Slika 11. korisniku je omogućeno pomoću izvješća „Funkcijske točke projekta“ grupirati informacije o korisničkim zahtjevima s funkcijskim točkama i općim karakteristikama sa ocjenama, za pripadajući projekt. Prema formuli za izračunavanje ukupnog broja funkcijskih točaka koja je navedena dobiven je broj stvarnih FT projekta.



Slika 11: Dijagram obrasca uporabe za izvješće „Funkcijske točke projekta“

4.3.2 Scenarij obrasca uporabe

Pomoću obrasca scenarija u nastavku teksta opisano je nekoliko karakterističnih slučaja uporabe: unos projekta, ocjena utjecaja općih karakteristika na sustav, unos logičke transakcije, određivanje broja entiteta, ulaza i izlaza za svaku LT, izračun funkcijskih točaka (FT).

U Tablica 2 opisan je scenarij obrasca uporabe za unos projekta. Opisani su koraci u kojima korisnik prikuplja korisničke zahtjeve, definira naziv projekta, te početak i završetak projekta.

Tablica 2: Scenarij 0brasca uporabe za unos projekta

	Obrazac uporabe – Unos projekta
Autor	Nikolina Žabić
Datum	22. ožujka 2019.
Naziv obrasca uporabe	U sustavu se vodi evidencija o projektima i njihovoj realizaciji. Potrebni za stvaranje statistike
Opis	Korisnik unosi novi projekt
Sudionici	Korisnik
Okidači	Procjena opsega projekta
Preduvjeti	Isporučeni projekt
Osnovni scenarij	Provjera postoji li projekt već u sustavu Korisnik dodaje novi projekt u sustav Korisnik određuje datum početka i završetka projekta Korisnik upisuje KZ vezane uz projekt
Izuzeci	A) Projekt postoji u sustavu, ne unosi se ponovno

U nastavku su također opisani scenariji slučajeva uporabe za unos logičke transakcije u Tablica 3 i izvješća funkcijskih točaka projekta u tablici 3.

Tablica 3: Scenarij 0brasca uporabe za unos logičke transakcije

	Obrazac uporabe – Unos logičke transakcije
Autor	Nikolina Žabić
Datum	22. ožujka 2019.
Naziv obrasca uporabe	Unos logičke transakcije
Opis	Korisnik izabire jedan od osnovnih tipova logičke transakcije (modul, obrada, izvješće ili zajednički proces). Zatim se traži od korisnika da upiše ukratko na što se odnosi izabrana logička transakcija.
Sudionici	Korisnik
Okidači	Nema
Preduvjeti	Izabran željeni projekt Izabrana željena opća karakteristika vezana uz projekt
Osnovni scenarij	Korisnik izabire jedan od tipova LT Korisnik opisuje logičku transakciju Korisnik pridjeljuje vrijednosti svakom tipu LT u odnosu na input, entitet i output Sustav automatski računa pojedinačne rezultate Sustav nakon izračunatih rezultata računa sumu
Izuzeci	Nema

Tablica 4: Obrazac uporabe za izvješće funkcijske točke projekta

	Obrazac uporabe – Izvješće funkcijske točke projekta
Autor	Nikolina Žabić
Datum	22. ožujka 2019.
Naziv obrasca uporabe	Funkcijske točke projekta
Opis	Grupiranje podataka o projektu i pripadajućim korisničkim zahtjevima i općim karakteristikama uz sumu funkcijskih točaka
Sudionici	Korisnik
Okidači	Nema
Preduvjeti	Projekt u sustavu
Osnovni scenarij	Korisnik otvara formu „ Logička transakcija“ Korisnik izabire željeni projekt Za zadani projekt sustav pronalazi korisničke zahtjeve i pripadajuće FT Sustav prikazuje formu sa popisom korisničkih zahtjeva i općih karakteristika s pripadajućim ocjenama vezanih uz projekt Računa se i prikazuje ukupan broj neprilagođenih funkcijskih točaka Računa se i prikazuje ukupna ocjena općih karakteristika projekta Računa se i prikazuje ukupan broj FT za projekt (tzv. Procijenjene funkcijske točke)
Izuzeci	Nema

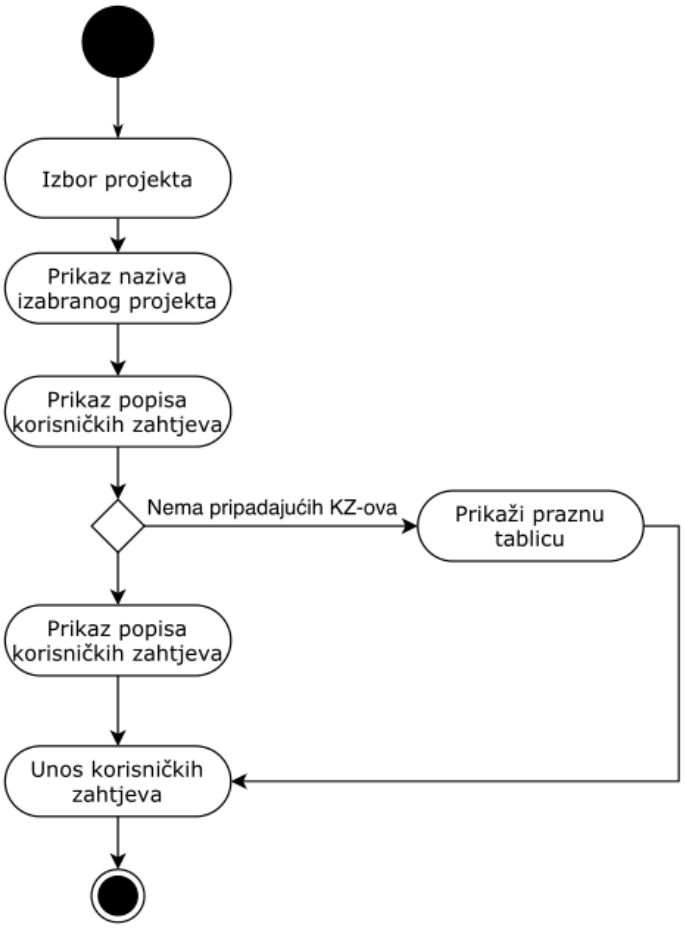
5. OBLIKOVANJE INFORMACIJSKOG SUSTAVA

5.1 Dijagram aktivnosti

To je posebni slučaj dijagrama stanja u kojem su sva stanja akcije ili podakcije. Prikazuje funkcionalnost softverskog sustava iz perspektive njegove unutrašnjosti. Dijagramom aktivnosti prikazuje se radni tok aktivnosti koje se obavljaju u sustavu korak po korak. Aktivnosti se odvijaju slijedno jedna za drugom.

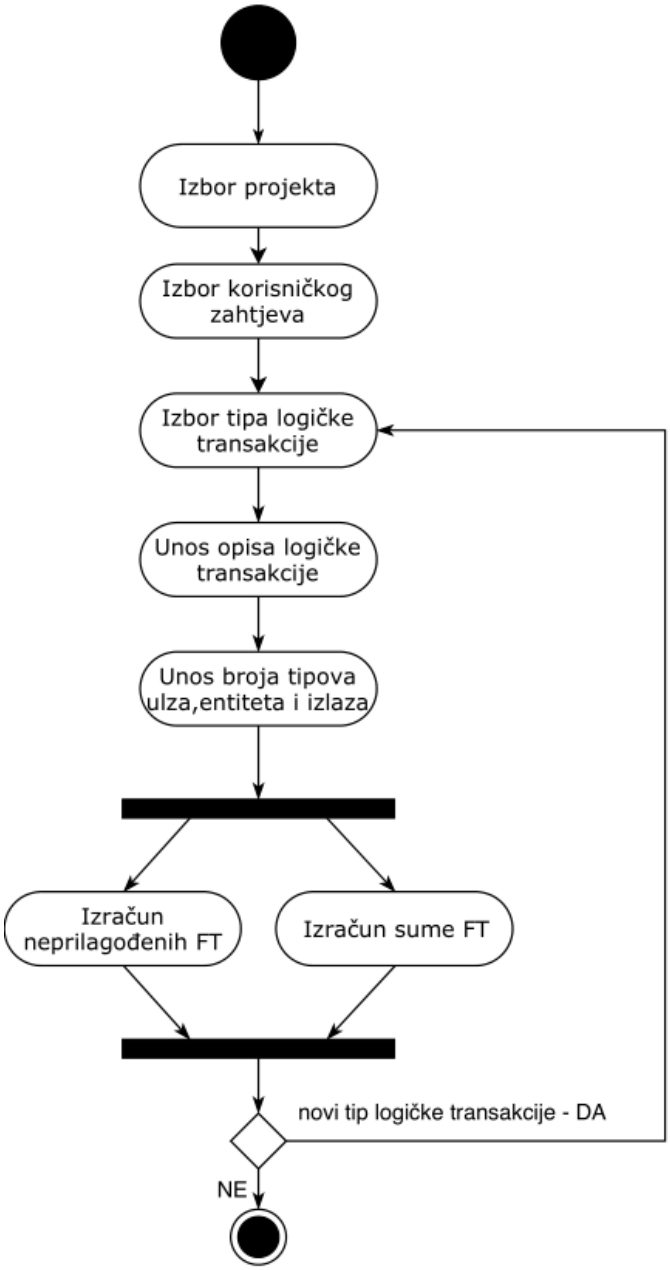
Temeljni elementi dijagrama aktivnosti su: početno i konačno stanje, aktivnosti, prijelaz između aktivnosti, odluka ili uvjetno grananje, račvanje ili skupljanje.

Dijagramom aktivnosti za unos korisničkog zahtjeva za pripadajući projekt na Slika 13 prikazani su koraci prema kojima korisnik unosi korisničke zahtjeve vezane uz projekt.



Slika 12: Dijagram aktivnosti za unos korisničkih zahtjeva vezanih uz projekt

Na Slika 13. prikazan je dijagram aktivnosti za unos pripadajuće logičke transakcije i izračun funkcijskih točaka za svaku logičku transakciju. Opisani su koraci u kojima korisnik izabire projekt i pripadajući korisnički zahtjev, zatim odabire vrtu logičke transakcije i na kraju dobije sumu funkcijskih točaka za izabranu logičku transakciju

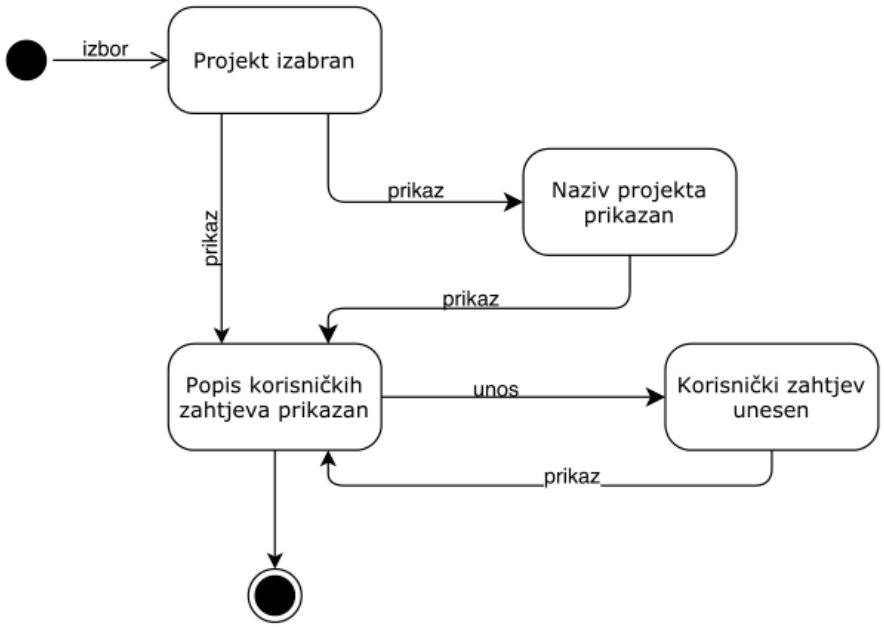


Slika 13: Dijagram aktivnosti za unos pripadajuće logičke transakcije i izračun funkcijskih točaka za svaku LT

5.2 Dijagram stanja

Služi za modeliranje osnovnih stanja unutar sustava. Prikazuje prijelaze između stanja u sustavu koji su prouzrokovani događajima. U jednom stanju entitet zadovoljava neki uvjet, obavlja neku aktivnost ili čeka na događaj.

Slika nam prikazuje dijagram stanja za unos korisničkih zahtjeva vezanih uz projekt.



Slika 14: Dijagram stanja za unos korisničkih zahtjeva uz projekt

6. IZRADA APLIKACIJE

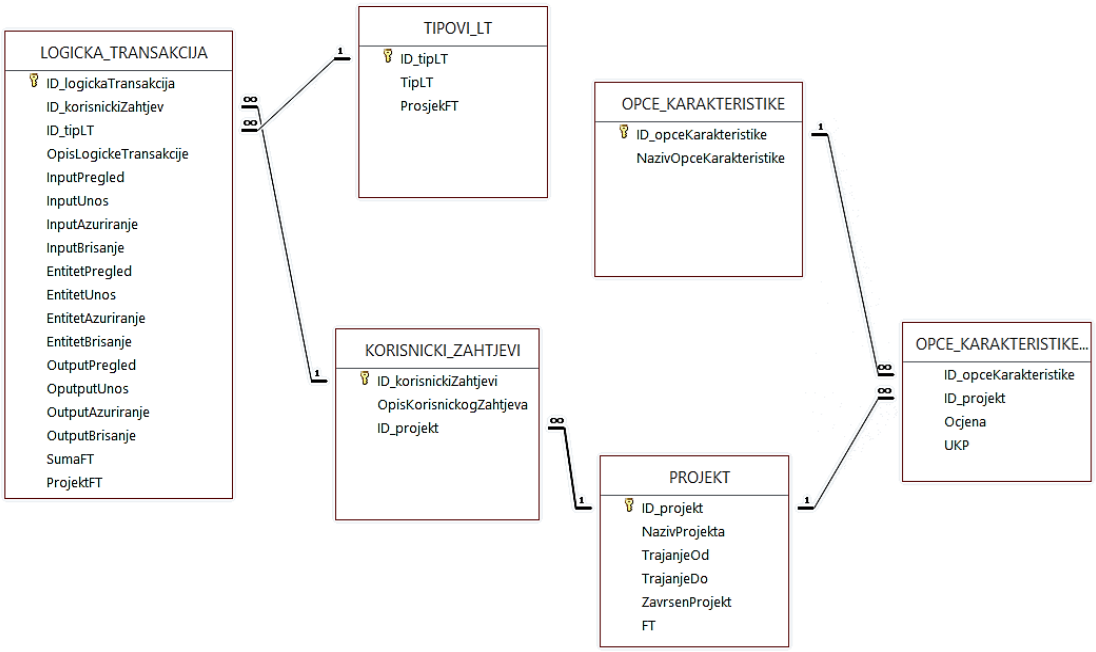
Aplikacija je zamišljena kao pomoć u procjeni opsega programske potpore koristeći metriku funkcijskih točaka. Pokretanjem aplikacije korisniku se otvara početni prozor i pruža nekoliko mogućnosti za izbor koje su ugrađene u samu aplikaciju. Prilikom odabira neke od ponuđenih mogućnosti na ekranu se pojavljuje prozor sa implementiranim funkcionalnostima.

U nastavku će izrađena aplikacija biti detaljnije opisana i priča kroz teoriju potkrijepljena je odgovarajućim slikama sučelja aplikacije.

6.1 Relacijski model podataka

Osnovna zamisao relacijskog modela baze podataka jest skup podataka rastaviti na manje logički povezane cjeline i podatke svake cjeline spremiti u zasebnu tablicu. Već od prije su nam poznati osnovni elementi baze podataka kao i veze među entitetima stoga se u ovom poglavlju neće detaljno opisivati.

Na Slika 15 je prikazana struktura baze podataka za procjenu opsega programske podrške pomoću funkcijski orijentirane metrike.



Slika 15: Relacijski model baze podataka

6.2 Priprema razvojnog okruženja

Za razvoj ove aplikacije bilo je potrebno imati pristup Microsoft Access programu za upravljanje relacijskom bazom podataka. Korištena verzija u ovom slučaju je Microsoft Access 2016.

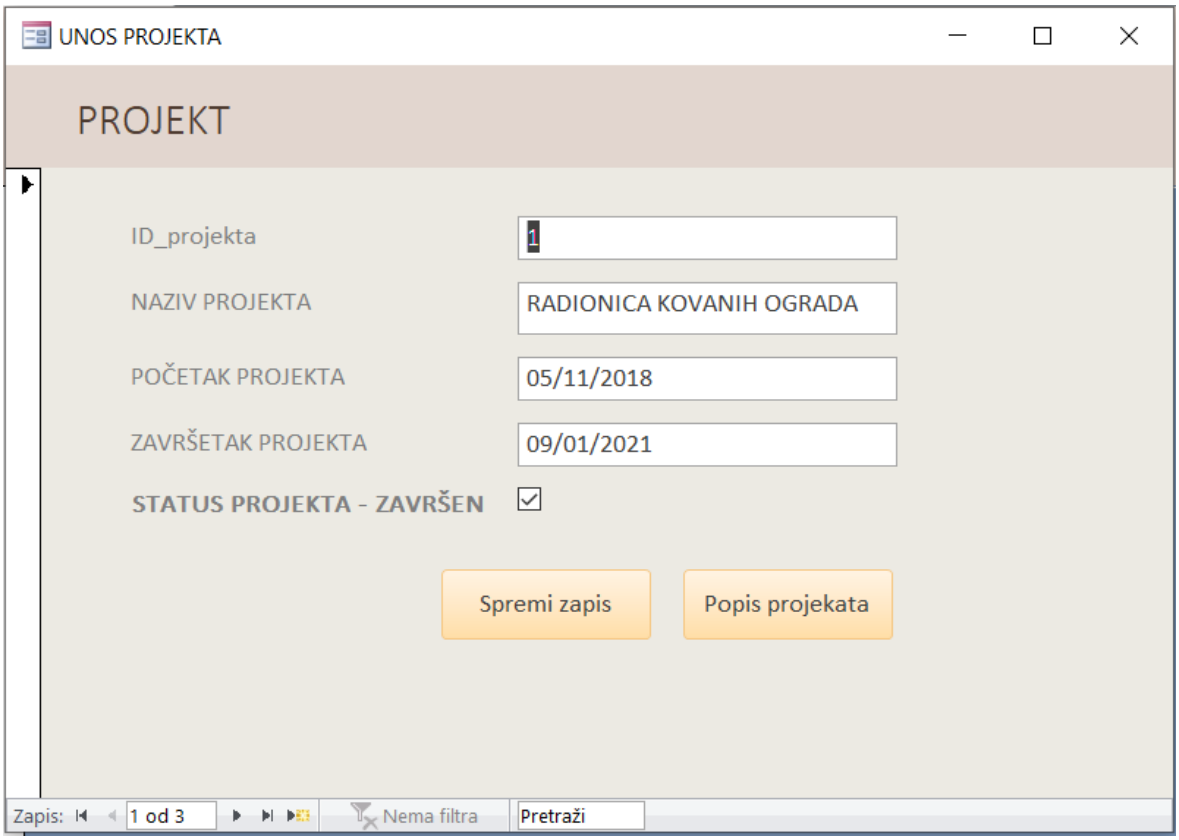
U razvoju navedene aplikacije korišten je kao alat za upravljanje informacijama jer može sadržavati i analizirati ogromnu količinu informacija koje su u međusobnom odnosu. Relativno je intuitivan program za oblikovanje cjelovitog programskog rješenja. Upravljanje podacima omogućeno je kroz obrasce, dok izvješća predstavljaju opći sažetak željenih informacija, a kreiraju se automatski iz podataka pohranjenih u bazi.

6.3 Obrasci aplikacije

Ulaskom u aplikaciju pokreće se početna forma koja se sastoji od glavnog izbornika i pripadajućih komponenti što je vidljivo na Slika 16**.**

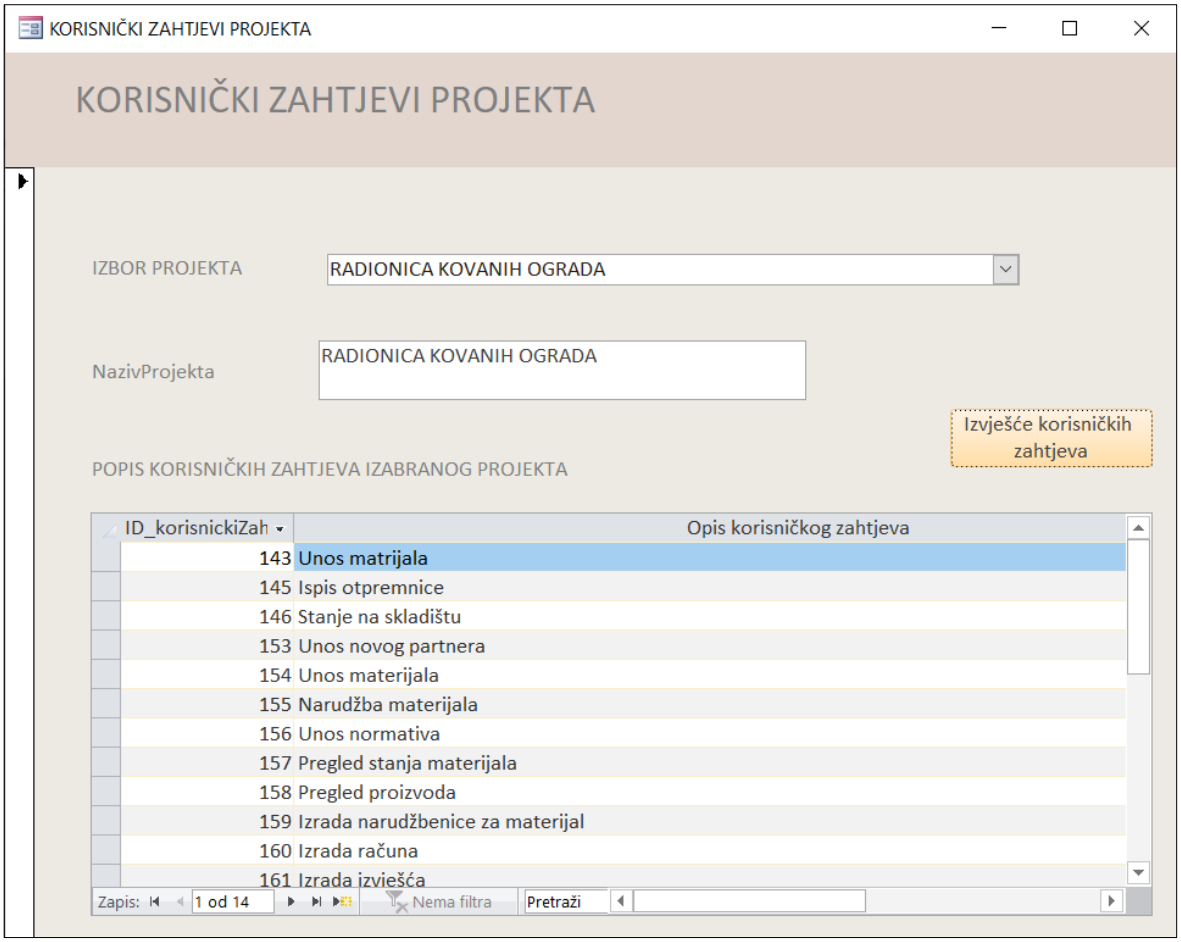
Slika 16: Izgled početne forme aplikacije

Izborom „Unos projekta“ otvara se forma koja omogućuje pregled projekata koji se nalaze u bazi kao i mogućnost unosa novog projekta kako je rečeno u samom nazivu izbornika. Također se na formi nalazi kontrola za definiranje statusa projekta. Osim navedenih elemenata forme nalazi se i tipka za spremiti zapise. U nastavku na Slika 17. prikazan je izgled forme „Unos projekta“.



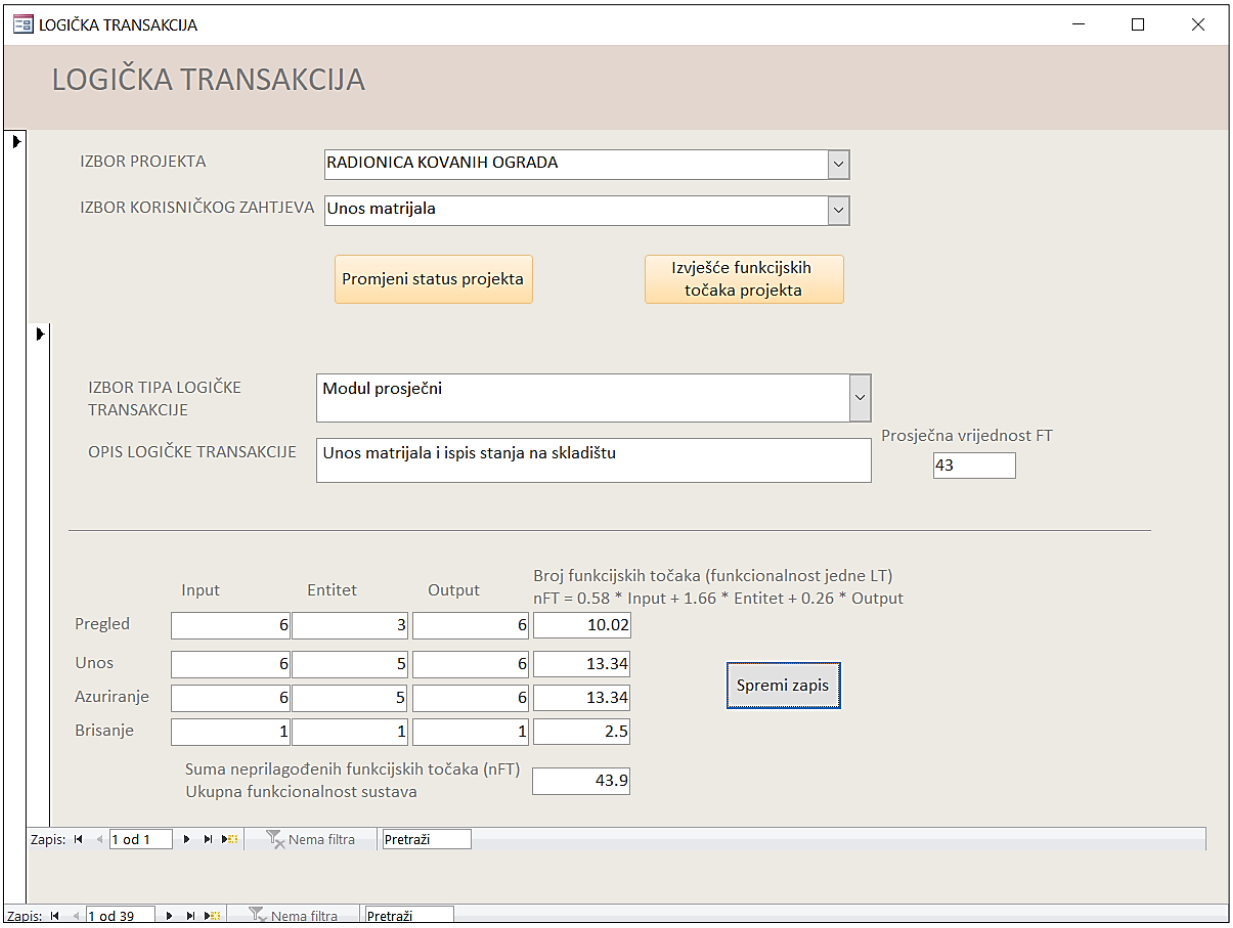
Slika 17: Forma za unos projekta

Kroz formu za unos korisničkih zahtjeva koja je grafički prikazana na Slika 18. omogućen je unos i prikaz korisničkih zahtjev za pripadajući projekt. Tipka „Izvješće korisničkih zahtjeva“ prikazuje popis svih zahtjeva za pripadajući projekt.



Slika 18: Forma za unos korisničkih zahtjeva

Forma na kojoj je za pojedini korisnički zahtjev omogućen unos pripadajuće logičke transakcije prikazana je na Slika 19. Složena forma logičke transakcije sastoji se od jedne glavne forme i jedne podforme. Na samom početku potrebno je izabrati željeni projekt izabrati korisnički zahtjev koji pripada izabranom projektu te za njega unijeti logičku transakciju. Logička transakcija definirana je sa 11 podtipova, Tablica 1. Na formi je prikazano i polje koje prikazuje prosječnu vrijednost LT. Dvije tipke koje se nalaze na formi služe za rad sa podatcima za pripadajuću formu. Detaljno objašnjenje je u nastavku kada bude govora o izvješćima aplikacije.



Slika 19: Forma za unos logičke transakcije za pripadajući korisnički zahtjev

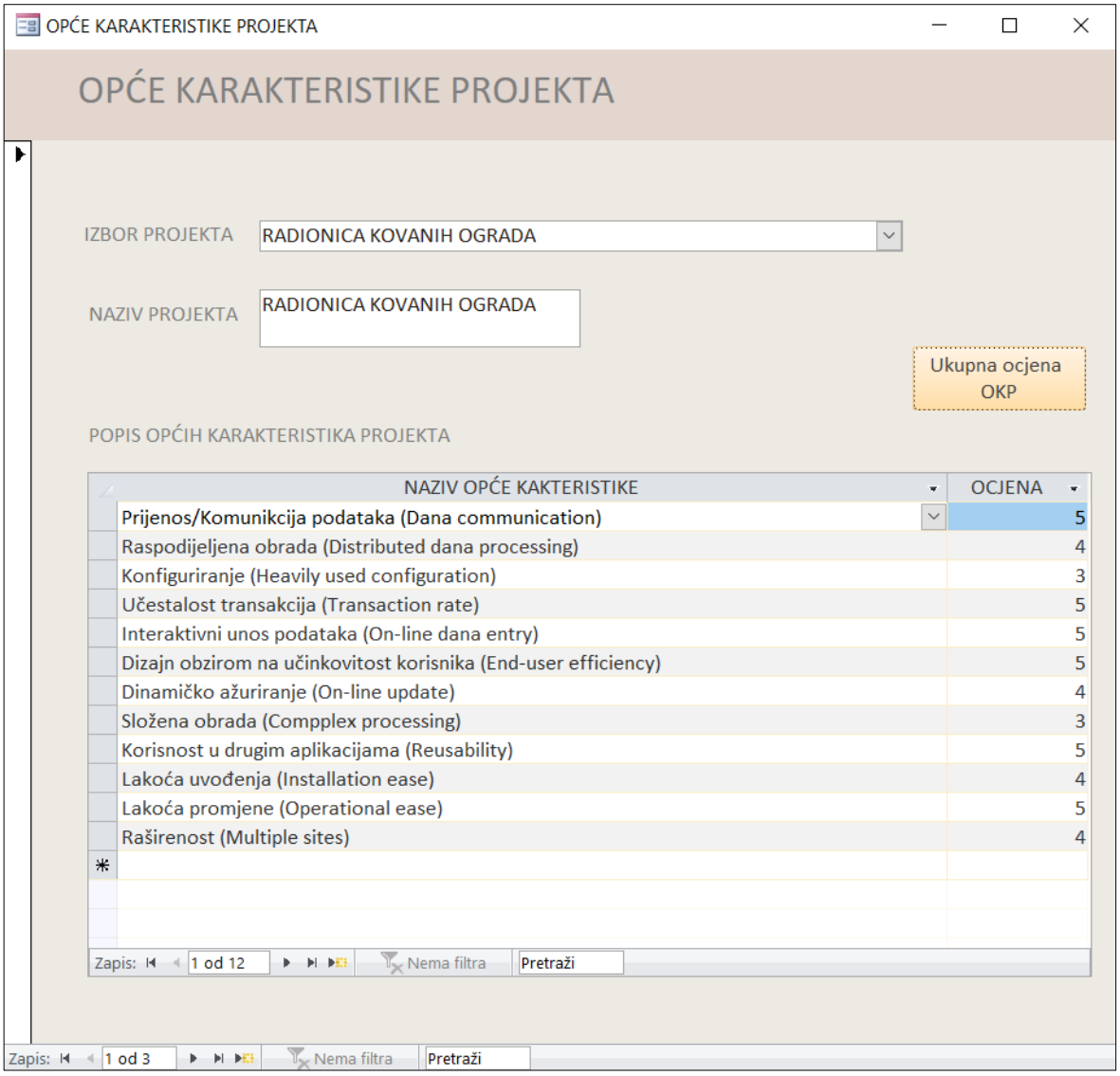
Jedna od preostalih mogućnosti za izbor na početnoj formi je izbornik za opće karakteristike projekta. Forma je osmišljena na način da je omogućeno dodjeljivanje odgovarajućih općih karakteristika pojedinom projektu. Na formi se prikazuju podaci za pojedini projekt i nije ih moguće ažurirati, te se u podformi biraju opće karakteristike iz popisa općih karakteristika Slika 20. i unosi se ocjena za svaku odabranu opću karakteristiku.

Opće karakteristike koje utječu na procjenu FT osim standardom definiranih mogu se unositi prema potrebama korisnika. Opisana forma grafički je prikazana na Slika 21.

POPIS OPĆIH KARAKTERISTIKA

ID_OK	NAZIV OPĆE KARAKTERISTIKE
1	Prijenos/Komunikcija podataka (Dana communication)
2	Raspodijeljena obrada (Distributed dana processing)
3	Konfiguriranje (Heavily used configuration)
4	Učestalost transakcija (Transaction rate)
5	Interaktivni unos podataka (On-line dana entry)
6	Dizajn obzirom na učinkovitost korisnika (End-user efficiency)
7	Dinamičko ažuriranje (On-line update)
8	Složena obrada (Compplex processing)
9	Korisnost u drugim aplikacijama (Reusability)
10	Lakoća uvođenja (Installation ease)
11	Lakoća promjene (Operational ease)
12	Raširenost (Multiple sites)
13	Mogućnost promjena (Facilitate change)
51	Raspodijeljena obrada (Distributed dana processing)

Slika 20: Popis općih karakteristika



Slika 21: Forma za dodjeljivanje odgovarajućih općih karakteristika pojedinom projektu

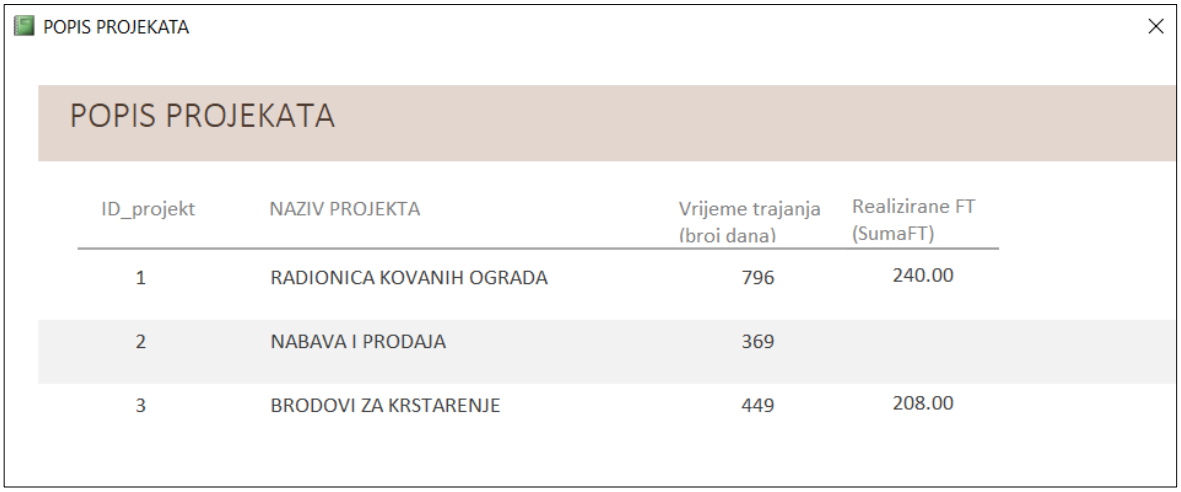
6.4 Izvješća aplikacije

Razvojno okruženje MS Access ima mogućnost stvaranja izvješća. Izvješće (engl. *report*) je dio baze podataka koji omogućava prilagođeni prikaz i način ispisa podataka iz baze podataka.

Sastavni dio izrađene aplikacije također su i izvješća. Pomoću nekoliko izrađenih izvješća grupirane su i prikazane željene informacije koje će detaljnije biti opisane u nastavku.

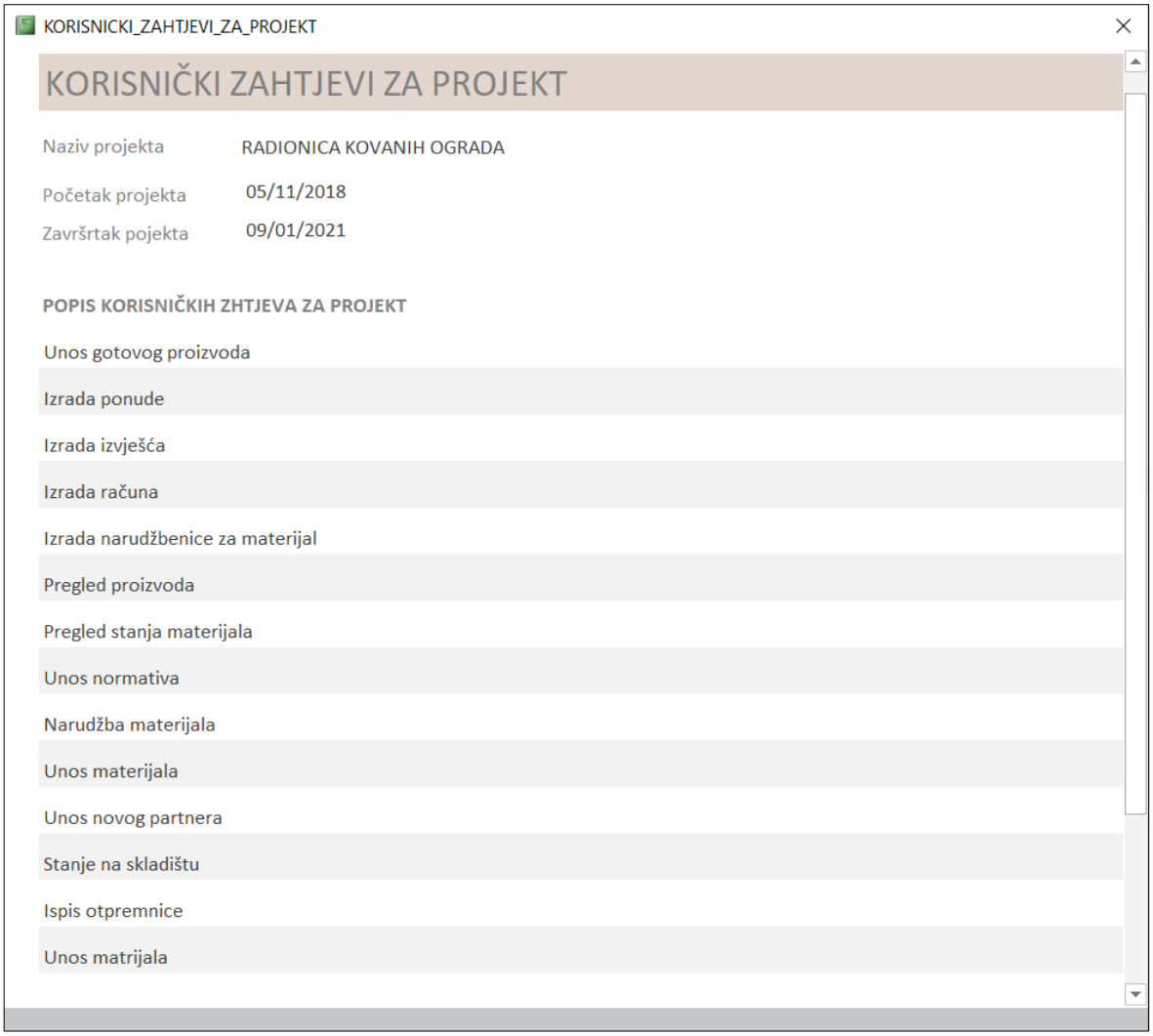
Slika 22. prikazuje izvješće pod nazivom „Popis projekata“. Kako sam naziv kaže radi se o popisu projekata koji se nalaze u bazi. Iz izvješća se može iščitati naziv projekta, vrijeme trajanja pojedinog projekta izraženo u danima i broj realiziranih FT.

Projekt pod rednim brojem dva nema vrijednosti realiziranih FT budući da projekt nije označen kao završen.



Slika 22: Izvješće za popis projekata

Izvješće „Korisnički zahtjevi za projekt“ prikazuje podatke o projektu (naziv, vrijeme početka i završetka projekta), te popis pripadajućih korisničkih zahtjeva. Slika 23. grafički predočuje izgled izvješća.



Slika 23: Izvješće za popis korisničkih zahtjeva za projekt

Izvješće pod nazivom „Ukupna ocjena općih karakteristika zasebno prikazano na Slika 24. dio je glavnog izvješća „Funkcijske točke projekta“.

Na formi koja je prikazana na Slika 21. pritiskom na tipku „Ukupna ocjena OK“ na ekranu nam se prikazuje izvješće sa Slika 24. koje za pripadajući projekt ispisuje popis pripadajućih općih karakteristika s ocjenama i njihovu sumu kao ukupnu ocjenu OK projekta.

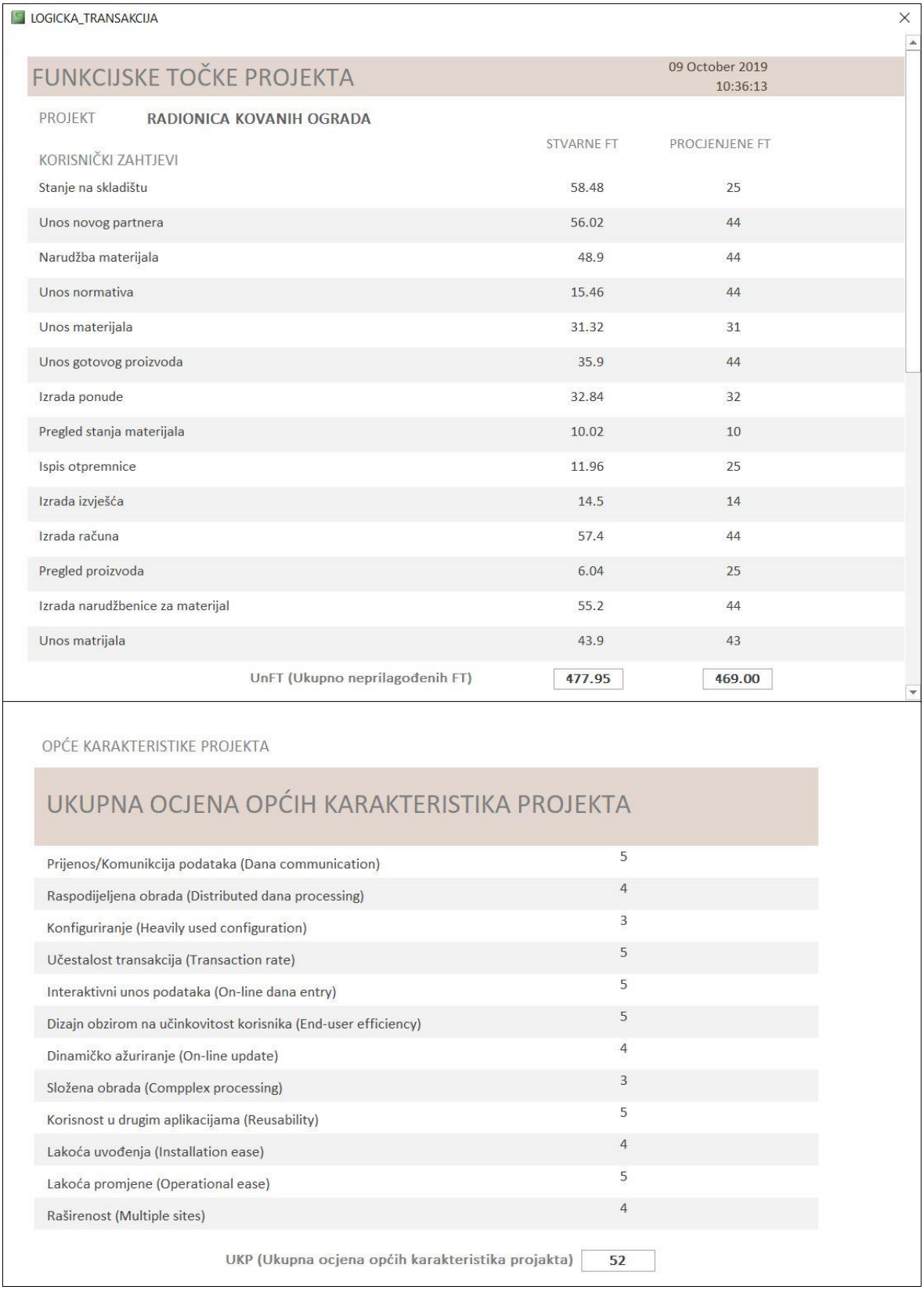
OPCE_KARAKTERISTIKE_PROJEKTA

UKUPNA OCJENA OPĆIH KARAKTERISTIKA PROJEKTA

POPIS OPĆIH KARAKTERISTIKA ZA PROJEKT	OCJENA
Prijenos/Komunikcija podataka (Dana communication)	5
Raspodijeljena obrada (Distributed dana processing)	4
Konfiguriranje (Heavily used configuration)	3
Učestalost transakcija (Transaction rate)	5
Interaktivni unos podataka (On-line dana entry)	5
Dizajn obzirom na učinkovitost korisnika (End-user efficiency)	5
Dinamičko ažuriranje (On-line update)	4
Složena obrada (Compplex processing)	3
Korisnost u drugim aplikacijama (Reusability)	5
Lakoća uvođenja (Installation ease)	4
Lakoća promjene (Operational ease)	5
Raširenost (Multiple sites)	4
UKP (Ukupna ocjena općih karakteristika projakta)	52

Slika 24: Izvješće za opće karakteristike projekta i ukupna ocjena OK

Na Slika 25. prikazano je izvješće „Funkcijske točke projekta“. Za odabrani projekt prikazuje se popis korisničkih zahtjeva sa njihovom sumom i općih karakteristika projekta također s sumom ocjena OK.



LOGICKA_TRANSAKCIJA

FUNKCIJSKE TOČKE PROJEKTA

09 October 2019
10:36:13

PROJEKT **RADIONICA KOVANIH OGRADA**

KORISNIČKI ZAHTJEVI	STVARNE FT	PROCJENJENE FT
Stanje na skladištu	58.48	25
Unos novog partnera	56.02	44
Narudžba materijala	48.9	44
Unos normativa	15.46	44
Unos materijala	31.32	31
Unos gotovog proizvoda	35.9	44
Izrada ponude	32.84	32
Pregled stanja materijala	10.02	10
Ispis otpremnice	11.96	25
Izrada izvješća	14.5	14
Izrada računa	57.4	44
Pregled proizvoda	6.04	25
Izrada narudžbenice za materijal	55.2	44
Unos matrijala	43.9	43
UnFT (Ukupno neprilagođenih FT)	477.95	469.00

OPĆE KARAKTERISTIKE PROJEKTA

UKUPNA OCJENA OPĆIH KARAKTERISTIKA PROJEKTA

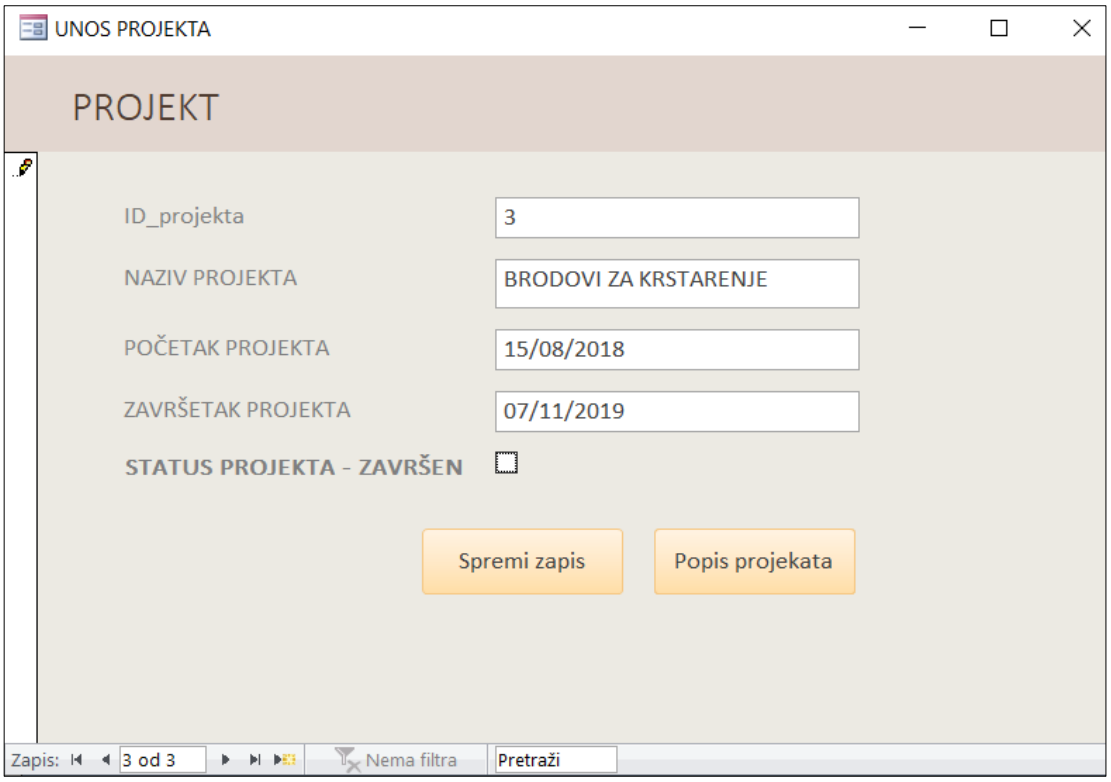
Prijenos/Komunikcija podataka (Dana communication)	5
Raspodijeljena obrada (Distributed dana processing)	4
Konfiguriranje (Heavily used configuration)	3
Učestalost transakcija (Transaction rate)	5
Interaktivni unos podataka (On-line dana entry)	5
Dizajn obzirom na učinkovitost korisnika (End-user efficiency)	5
Dinamičko ažuriranje (On-line update)	4
Složena obrada (Compplex processing)	3
Korisnost u drugim aplikacijama (Reusability)	5
Lakoća uvođenja (Installation ease)	4
Lakoća promjene (Operational ease)	5
Raširenost (Multiple sites)	4
UKP (Ukupna ocjena općih karakteristika projakta)	52

Slika 25: Izvješće za funkcijske točke projekta

6.5 Prikaz procjene za novi projekt

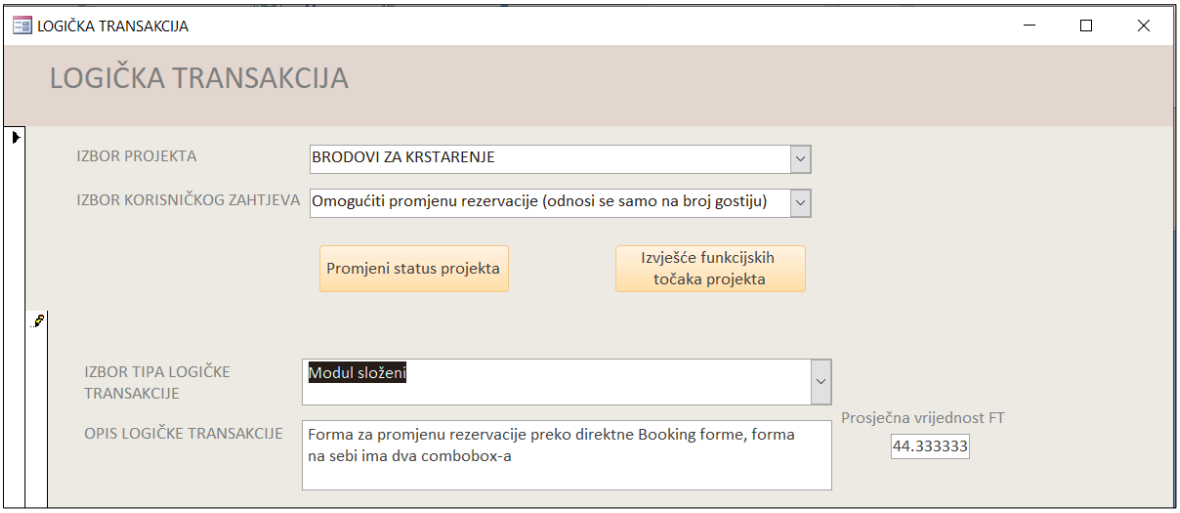
U nastavku rada prikazan je slijed praćenja projekta od početka kada se projekt unese u sustav do trenutka kada se za uneseni projekt pomoću izračuna dobije vrijednost težine projekta izražena u FT. Na samom početku definiraju se korisnički zahtjevi, procjenjuju pripadajuće LT na osnovu prosječnih vrijednosti tipova LT, Slika 28, unosi stvarna težina pojedine LT nakon što je implementirana u programsko rješenje, te se po završetku projekta generiraju nove prosječne vrijednosti tipova LT. Tako korigirane prosječne vrijednosti su polazne točke za procjenu novih projekata.

Unos novog projekta započinje Slika 26.



Slika 26: Forma za unos novog projekta

U fazama izrade projekta dok još uvijek nisu poznati detalji logičke transakcije. Izborom tipa LT i složenosti prikaže se prosječna vrijednost za izabrani tip LT, Slika 27. Prosječna vrijednost LT nalazi se u tablici TIPOVI_LT, slika Slika **28**.



Slika 27: *Unos LT i prikaz pro procijenjene vrijednosti iz tablice TIPOVI_LT*

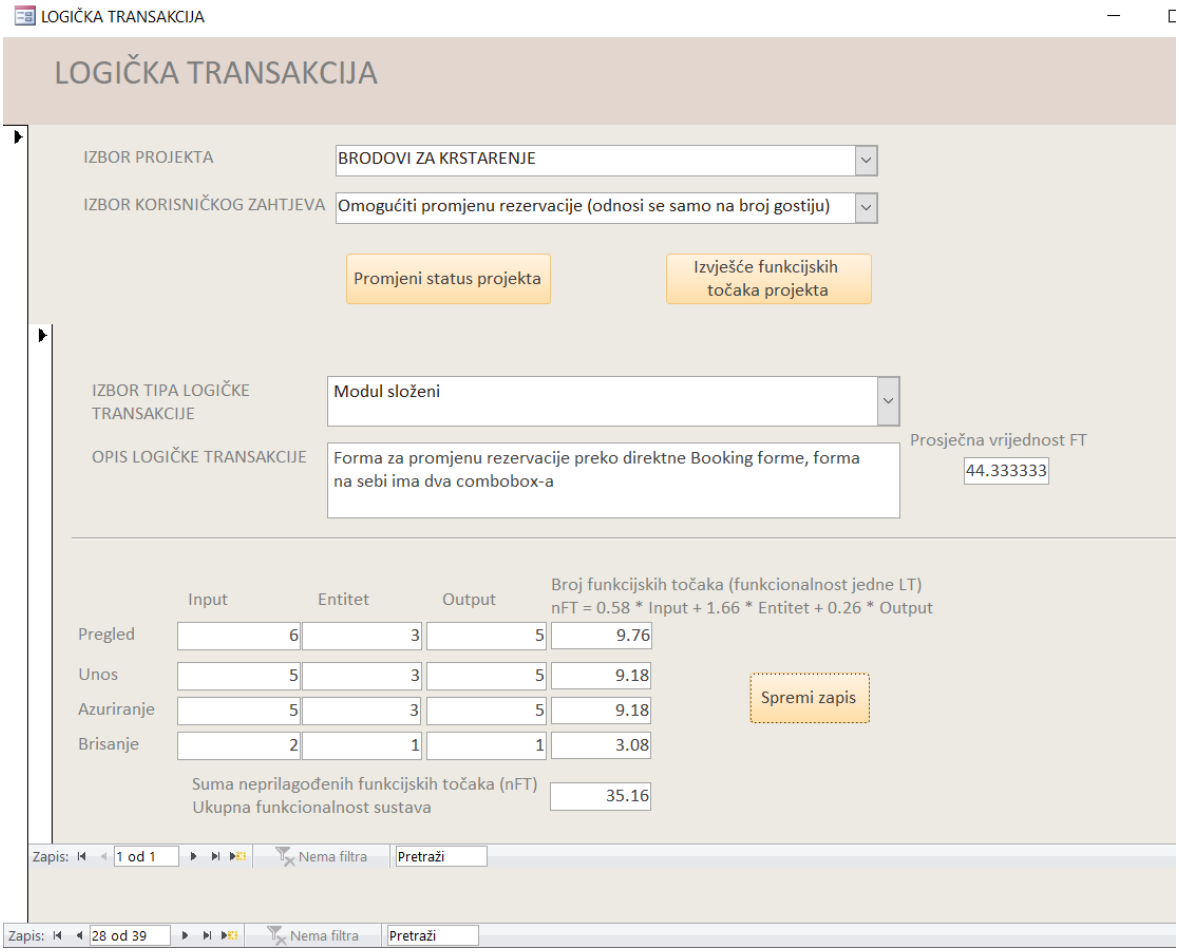
APLIKACIJA | TIPOVI_LT

ID_tipLT	TipLT	ProsjekFT	Kliknite da biste dodali
1	Modul jednostavni	31.00	
2	Modul prosječni	43.00	
3	Modul složeni	44.00	
4	Obrada jednostavni	0.00	
5	Obrada prosječni	32.00	
6	Obrada složeni	94.00	
7	Izvješće jednostavni	25.00	
8	Izvješće prosječni	10.00	
9	Izvješće složeni	14.00	
10	Zajednički proces jednostavni	0.00	
11	Zajednički proces složeni	0.00	
(Novi)		0.00	

Slika 28*: Tablica TIPOVI_LT*

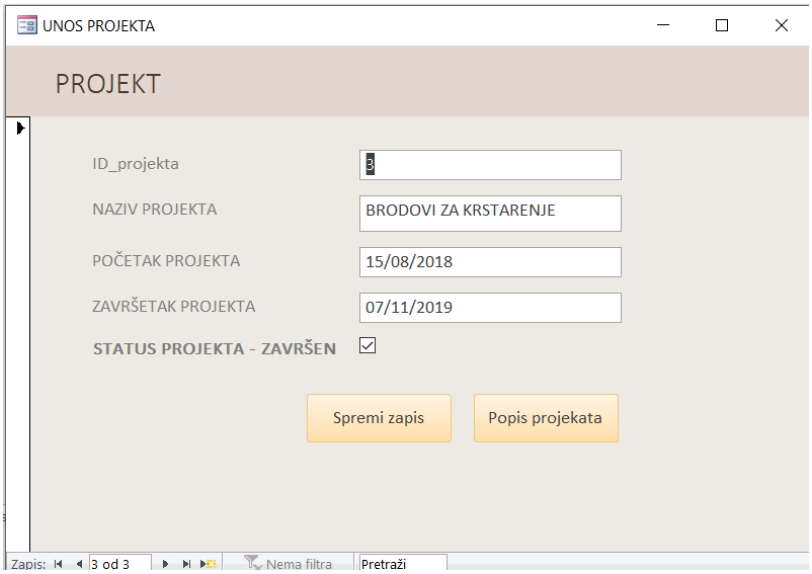
Kako proces izrade programa traje tako postaju poznate i LT te se njihove vrijednosti unose za svaki korisnički zahtjev. Jedan KZ može imati jednu ili više LT.

Popunjavanjem vrijednosti LT prosječna vrijednost im se ne mijenja sve dok se projekt ne označi kao gotov, Slika 29



Slika 29: *Prikaz unosa vrijednosti LT*

Nakon što je projekt označen kao gotov, a to je vidljivo na Slika 30, pritiskom na spremi zapis ažuriraju se vrijednosti u tablici tipova logičkih transakcija.



Slika 30: *Novi dodani projekt označen kao završen.*

Na Slika 31 prikazane su ažurirane prosječne vrijednosti tipova LT.

APLIKACIJA | TIPOVI_LT

ID_tipLT	TipLT	ProsjekFT	Kliknite da biste dodali
1	Modul jednostavni	31.00	
2	Modul prosječni	43.00	
3	Modul složeni	42.00	
4	Obrada jednostavni	0.00	
5	Obrada prosječni	32.00	
6	Obrada složeni	94.00	
7	Izvješće jednostavni	25.00	
8	Izvješće prosječni	10.00	
9	Izvješće složeni	14.00	
10	Zajednički proces jednostavni	0.00	
11	Zajednički proces složeni	0.00	
(Novi)		0.00	

Slika 31: *Ažurirane prosječne vrijednosti tipova LT*

Cjelokupan prikaz FT za projekt vidljiv na posljednjoj slici **Slika 32**. Za svaki korisnički zahtjev prikazan je stvarni broj FT izračunat nakon implementacije (donji dio Slika 29) i usporedno procijenjene FT na osnovu baze znanja iz tablice tipova logičkih transakcija (Slika 28 i gornji dio Slika 29).

U ovom primjeru je razlika procijenjenih i realiziranih (stvarnih) FT oko 20% što je prihvatljivo za projekte razvoja programske potpore. Primjer sa Slika 25. ima još manju razliku procijenjenih i realiziranih FT. Može se zaključiti da firma koja se bavi razvojem programskih rješenja (i koristi ovu aplikaciju) ima dovoljno iskustva i znanja zabilježenog u tablici na slici F koje može kvalitetno koristiti u procjeni novih poslova. Također, to bi značilo i da firma ima projektante i programere s dovoljno iskustva i znanja tako da mogu dobro procijeniti nove korisničke zahtjeve za novi projekt.

Slika 32: Izvješće za funkcijske točke projekta „Brodovi za krstarenje“

7. ZAKLJUČAK

Ovim je radom dan sažeti pregled teme mjerenja opsega programske potpore pomoću funkcijski orijentirane metrike. Kroz rad je teoretski opisana metrika funkcijskih točaka kao metoda za određivanje složenosti gotovog programskog proizvoda. Osim određivanja funkcionalne veličine gotovog programskog proizvoda, metrika funkcijskih točaka primijenjena je i u dijelu koji omogućava procjenu funkcijskih točaka i za buduće projekate.

Aplikacija poput ove koja je izrađena u ovom završnom radu osmišljena je kako bi se automatizirao postupak brojanja funkcijskih točaka. Prema zadanim korisničkim zahtjevima u aplikaciji je omogućeno brojanje FT za gotov proizvod i procjena FT za novi projekt. Dosta prostora je ostalo da bi aplikacija prerasla u informacijski sustav, ali svakako ovako automatizirani način brojanja može voditelju projekta uštedjeti dosta vremena,

Zaključno rečeno, za daljnji razvoj ove aplikacije može se osigurati mjerenje pomoću nekih drugih metoda. Jedna od spomenutih metoda je „Točke slučajeva korištenja“, a sve u svrhu bolje procjene opsega projekta.

LITERATURA

[1] Drummond, I. (1993). *Estimating with Mk II Function Point Analysis.* London: CCTA - Government Centre for Information Systems. Dohvaćeno iz CCTA - Government Centre for Information Systems.

[2] *Functional point.* (2011). Dohvaćeno iz Software_quality_management: https://www.tutorialspoint.com/software_quality_management/software_quality_management_albrechts_function_point_method.htm

[3] Galin, D. (2004). *Software Quality Assurance.* Dohvaćeno iz Software quality assurance: http://desy.lecturer.pens.ac.id/Manajemen%20Kualitas%20Perangkat%20Lunak/ebook/Software%20Quality%20Assurance%20From%20Theory%20to%20Implementation.pdf

[4] Karmen Klarin, K. K. (2002). Informacijski sustavi / bilješke s predavanja. U *Informacijski sustavi.* Split: Veleučilište u Splitu.

[5] Karmen Klarin, K. K. (2002). Projektiranje informacijskih sustava / bilješke s predavanja. U *Projektiranje informacijskih sustava* . Split: Veleučilište u Splitu.

[6] Klarin, K. (2002). Određivanje opsega programske potpore. Split: Veleučilište u Splitu.

[7] Klarin, K. (2018/2019). Bilješke sa predavnja. *Mjerenje neprilagođenih funkcijkih točaka*. Split: Veleučilište u Splitu.

[8] Goran Znidaršić. (2012.). Završni rad *Proizvodnja kovanih ograda,* Odjel za stručne studije, Split

POPIS SLIKA

***Slika 1**: Model baze podataka za projekt "Proizvodnja kovanih ograda"* 7
***Slika 2:** Unosna maska „JEDINICA MJERE“* 8
***Slika 3**: Unosna maska "MATERIJAL"* 8
***Slika 4:** Unosna maska "RAČUN"* 9
***Slika 5:** Izvješće "Utrošak materijala po proizvodu"* 9
***Slika 6:** Način mjerenja FT za primjer "Proizvodnja kovanih ograda"* 10
***Slika 7:** Ukupan broj FT za projekt Proizvodnja kovanih ograda* 12
***Slika 8:**Tipovi funkcionalnosti za metodu analize FT* 13
***Slika 9:** Dijagram obrasca uporabe za mjerenje opsega programske potpore* 20
***Slika 10:** Dijagram obrasca uporabe za izvješće „Funkcijske točke projekta“* 21
***Slika 11:** Dijagram obrasca uporabe za izvješće „Funkcijske točke projekta“* 22
***Slika 12:**Dijagram aktivnosti za unos korisničkih zahtjeva vezanih uz projekt* 26
***Slika 13:** Dijagram aktivnosti za unos pripadajuće logičke transakcije i izračun funkcijskih točaka za svaku LT* 27
***Slika 14:** Dijagram stanja za unos korisničkih zahtjeva uz projekt* 28
***Slika 15:** Relacijski model baze podataka* 30
***Slika 16:** Izgled početne forme aplikacije* 31
***Slika 17:** Forma za unos projekta* 32
***Slika 18:** Forma za unos korisničkih zahtjeva* 33
***Slika 19:** Forma za unos logičke transakcije za pripadajući korisnički zahtjev* 34
***Slika 20:** Popis općih karakteristika* 35
***Slika 21:** Forma za dodjeljivanje odgovarajućih općih karakteristika pojedinom projektu* 36
***Slika 22:** Izvješće za popis projekata* 37
***Slika 23:** Izvješće za popis korisničkih zahtjeva za projekt* 38
***Slika 24:** Izvješće za opće karakteristike projekta i ukupna ocjena OK* 39
***Slika 25:** Izvješće za funkcijske točke projekta* 40
***Slika 26:** Forma za unos novog projekta* 41
***Slika 27:** Unos LT i prikaz pro procijenjene vrijednosti iz tablice TIPOVI_LT* 42
***Slika 28**: Tablica TIPOVI_LT* 42
***Slika 29:** Prikaz unosa vrijednosti LT* 43
***Slika 30:** Novi dodani projekt označen kao završen.* 44
***Slika 31:** Ažurirane prosječne vrijednosti tipova LT* 44

POPIS TABLICA

Tablica 1: *Tablica tipova i složenosti logičke transakcije 14*

Tablica 2: *Scenarij 0brasca uporabe za unos projekta 23*

Tablica 3: *Scenarij 0brasca uporabe za unos logičke transakcije 24*

Tablica 4: *Obrazac uporabe za izvješće funkcijske točke projekta 25*