

# Razvoj interne poslovne aplikacije u IT poduzeću

---

**Rudić, Martin**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Economics and Business / Sveučilište u Zagrebu, Ekonomski fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:148:652935>

*Rights / Prava:* [Attribution-NonCommercial-ShareAlike 3.0 Unported/Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 3.0](#)

*Download date / Datum preuzimanja:* **2024-08-10**



*Repository / Repozitorij:*

[REPEFZG - Digital Repository - Faculty of Economics & Business Zagreb](#)



Sveučilište u Zagrebu,

Ekonomski fakultet Zagreb

Diplomski sveučilišni studij Poslovne ekonomije – Menadžerska  
informatika

**Martin Rudić**

***Razvoj interne poslovne aplikacije u IT  
poduzeću***

**DIPLOMSKI RAD**

**Mentor: prof. dr. sc. Mario Spremić**

Zagreb, rujan 2020.

Sveučilište u Zagrebu,  
Ekonomski fakultet Zagreb  
Diplomski sveučilišni studij Poslovne ekonomije – Menadžerska  
Informatika

**Martin Rudić**

***Development of an internal business  
application in an IT company***

**MASTER THESIS**

**Mentor: prof. dr. sc. Mario Spremić**

Zagreb, rujan 2020.

Martin Rudić

Ime i prezime studenta

## IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad  
(vrsta rada)

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, što pokazuju korištene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem također da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Student:

U Zagrebu, 17.09.2020

*M. Rudić*

(potpis)

## SADRŽAJ

1.	UVOD.....	1
1.1.	Predmet i cilj rada .....	1
1.2.	Izvori podataka i metoda prikupljanja .....	1
1.3.	Sadržaj i struktura rada .....	1
2.	VAŽNOST POSLOVNE ANALIZE .....	3
2.1.	Objašnjenje pojma poslovna analiza.....	3
2.2.	Nastajanje, uloga i predmet poslovne analize .....	4
2.2.1.	Funkcioniranje i ostvarivanje ciljeva poduzeća.....	4
2.2.2.	Upravljanje poduzećem .....	5
2.2.3.	Mjesto i uloga poslovne analize u procesu upravljanja poduzećem .....	7
2.2.4.	Poslovna analiza i planiranje i kontrola.....	8
2.3.	Vrste poslovne analize .....	9
2.3.1.	Eksterna i interna analiza .....	10
2.3.2.	Ekstenzivna i intenzivna analiza .....	10
2.3.3.	Stalna i povremena analiza .....	11
2.3.4.	Kratkoročna i dugoročna analiza.....	11
2.4.	Poslovna analiza kao faza projekta .....	12
2.4.1.	Bihevioralne vještine i osobine .....	13
2.4.2.	Poslovna znanja.....	14
2.4.3.	Tehnike.....	14
3.	MODELIRANJE BAZE PODATAKA .....	15
3.1.	Objašnjenje pojma modeliranje podataka.....	15
3.1.1.	Modeliranje podataka.....	15
3.2.	Objašnjenje pojma baze podataka.....	17
3.3.	Konceptualno modeliranje podataka.....	18
3.3.1.	Model entiteti-veze.....	18
3.3.2.	Izrada modela entitet-veze .....	20
3.4.	Logičko modeliranje podataka .....	21
3.4.1.	Relacijski model podataka .....	21
3.4.2.	Normalizacija.....	22

4.	PROGRAMSKO INŽENJERSTVO KAO TEMELJ USPJEŠNOG RAZVOJA SOFTWARE-A .....	23
4.1.	Uvod u programsko inženjerstvo .....	23
4.2.	Objašnjene pojma algoritma .....	23
4.3.	Modeliranje poslužitelja (Backend as a Service - BaaS) .....	24
4.3.1.	Backend as a Service (Baas) .....	24
4.4.	Modeliranje sučelja (frontend) .....	25
5.	OSIGURANJE KVALITETE PROGRAMSKE PODRŠKE.....	28
5.1.	Specifičnost programske podrške i kvalitete .....	28
5.1.1.	Što je kvaliteta programske podrške?.....	28
5.2.	Životni ciklus programske podrške .....	29
5.2.1.	Vodopadni model.....	29
5.2.2.	Prototip .....	31
5.2.3.	Spiralni model .....	31
5.3.	Testiranje programske podrške .....	33
5.3.1.	Testiranje „crna kutija“ .....	34
5.3.2.	Testiranje „bijela kutija“ .....	35
5.3.3.	Testiranje „siva kutija“ .....	36
5.3.4.	Agilno testiranje .....	37
5.3.5.	Ad hoc testiranje .....	38
6.	PRIMJER PROJEKTA - STUDIJ SLUČAJA .....	39
6.1.	Uvod .....	39
6.2.	Primjer funkcionalne specifikacije .....	39
6.3.	Primjer baze podataka .....	48
6.4.	Primjer programskog inženjerstva .....	51
6.5.	Primjer testiranja .....	53
6.6.	Prikaz aplikacije .....	55
7.	ZAKLJUČAK .....	59
	LITERATURA .....	60
	POPIS SLIKA I TABLICA.....	62
	Popis slika.....	62

# 1. UVOD

## 1.1. Predmet i cilj rada

Predmet ovog rada je unaprijediti procese unutar poduzeća kako bi korisnici (zaposlenici) bili zadovoljniji i učinkovitiji. Najučinkovitije unapređenje procesa odvija se se izradom poslovnih aplikacija koje pospješuju rad zaposlenih i samog poduzeća. U današnje doba često se spominje pojam „digitalizacija“. Digitalizacija je svuda oko nas, susrećemo se s njome svaki dan i u raznim segmentima života: privatnim, poslovnim i društvenim. Važnost digitalizacije u poslovnom svijetu je ta što poboljšava, unaprjeđuje i ubrzava poslovne procese. Primjenom digitalizacije se štedi mnogo vremena, a uštedom vremena stvara se prilika za druge radnje koje ostvaraju financijsku korist.

Cilj ovog rada je prikazati izradu poslovne aplikacije, od samih početaka gdje se izrađuju skice same aplikacije, pisanja specifikacije, izrade baze podataka, programiranja i testiranje aplikacije do konačnog produkta koji je spreman za korištenje.

## 1.2. Izvori podataka i metoda prikupljanja

Teoretski dio ovog rada zasniva se na sekundarnom istraživanju. U okviru sekundarnoga istraživanja korištene su različite knjige iz područja poslovne analize, baze podataka, programiranja i testiranja, znanstveni članci i web-portali.

Zaključci u ovome radu bit će doneseni na temelju prikaza projekta u kojem je autor sudjelovao. Projekt će biti prikazan slikama zaslona i imat će kratke interpretacije kako bi se čitateljima pobliže objasnio izgled i funkcionalnosti poslovne aplikacije.

## 1.3. Sadržaj i struktura rada

Sadržaj rada sastoji se od dva dijela: teoretski dio i studiju slučaja. Teoretski dio se sastoji od četiri poglavlja. Nakon uvoda, u prvom poglavlju „Važnost poslovne analize“ bit će definiran pojam poslovne analize, koji su predmeti poslovne analize, vrste poslovne analize i poslovna analiza kao faza projekta.

Drugo poglavlje „Modeliranje baze podataka“ sadržavat će objašnjenje pojma informacijski sustavi i pojma baze podataka. Detaljnije će biti objašnjeno konceptualno i logično modeliranje podataka.

Tema trećeg poglavlja je „Programsko inženjerstvo kao temelj uspješnog razvoja software-a“. U ovom poglavlju će se govoriti o programskom inženjerstvu te koje su vrste programskih

jezika. Bit će objašnjen pojam algoritma. U zadnjem dijelu ovog poglavlja će se definirati modeliranje poslužitelja i sučelja te će se napraviti usporedba tih dvaju modeliranja.

U posljednjem teoretskom poglavlju „Osiguranje kvalitete programske podrške“ bit će objašnjeno koje su specifičnosti programske podrške i kvalitete, životni ciklus programske podrške te vrste testiranja.

Studija slučaja započinje uvodom u kojem će se opisati predmet i cilj projekta. Zatim slijedi prikaz dijelova funkcionalne specifikacije i njezinih opisa, prikaz baze podataka i njezin ER model, prikaz programskog inženjerstva i linije koda te prikaz testiranja. Na kraju će biti prikazana cijela aplikacija.

Nakon studije slučaja, na kraju rada bit će donesen zaključak.



## 2. VAŽNOST POSLOVNE ANALIZE

### 2.1. Objašnjenje pojma poslovna analiza

Prema Janku Tintoru, poslovna analiza individualna znanost metodološke naravi. Promatranjem poduzeća i njezinog poslovanja izražena je posebnost. Poduzeće uvijek treba ostvariti određeni stupanj razvitka. Poslovna analiza bavi se pravilima i pretpostavkama načina razmišljanja, najprije poduzeća kao objekta, a potom kao dugotrajnog i aktivnog subjekta.<sup>1</sup>

Prema Općoj enciklopediji, analiza je rastavljanje, odnosno raščlanjivanje cjelovitog, kompleksnog, na njegove sastavne dijelove, te se naznačuje bitna razlika između formalno – logičkog i dijalektičkog shvaćanja analiza. Po ovom posljednjem, “sastavni elementi predmeta do kojih je došlo analizom ostavljaju se u svojoj međusobnoj izoliranosti, danosti, jedan pored drugog. Odnos među njima nije odnos indiferentne razlike nego suprotnosti, on uključuje i moment borbe, protustavljanja, i moment jedinstva, međusobnog prožimanja”, uslijed čega se analiza “ne može ni u mislima izdvojiti od sinteze – Svako raščlanjivanje je u isti mah i razlikovanje i identificiranje različitog.”<sup>2</sup>

Schumpeter govori da je riječ “rastaviti” samo jezični ekvivalent za “analizirati” i, u stvari, pogađa samo poseban tip aktivnosti za koju mislimo da označava termin “analiza”. Međutim, duh Stavka je važan, a ne riječi. Stavak kao cjelina jasno izražava činjenicu da je Aristotel svjesno primjenjivao analitičku metodu.<sup>3</sup>

---

<sup>1</sup> Tintor, J. (2009): Poslovna analiza: operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia: Poslovni dnevnik, Zagreb

<sup>2</sup> Krleža, M. (1996): Opća enciklopedija Leksigrafskog zavoda, I. svezak, Jugoslavenski leksikografski zavod, Zagreb, str. 116.

<sup>3</sup> Schumpeter, J.A., (1975): Povijest ekonomske analize; Prijevod – dr. Zvonimir Baletić i dr. Marijan Hanžeković, Informator, Zagreb

## 2.2. Nastajanje, uloga i predmet poslovne analize

### 2.2.1. Funkcioniranje i ostvarivanje ciljeva poduzeća

Prema Batarelu, poduzeće je kompleksna organizirana tvorevina koja funkcionira pretvarajući inpute (ono što od okoline preuzima) i resurse u outpute (ono što predaje okolini) ili rezultate tog procesa (proizvode usluge...), čija je tržišna vrijednost veća od vrijednosti uloženi ili utrošeni resursa.<sup>4</sup>

Poduzeće se treba racionalno ponašati, tako da uštedi što više mogućih resursa jer ću poduzeću više ostati za svakojake vlastite potrebe. Postavlja se pitanje: koristi li se maksimalno racionalno svoje raspoložive resurse, može li se i kako s njima još efikasnije proizvoditi svoje proizvode i usluge i postoji li neki, još bolji način koji bi dao još povoljniji vrijednosni odnos rezultata i ulaganja, odnosno veću pozitivnu razliku između vrijednosti outputa i inputa.<sup>5</sup>

Poduzeće „preispituje“ odvija li se njegova djelatnost u skladu iliti sukladno ekonomskim principima i „razmatra“ eventualne nove mogućnosti efikasnog stvaranja spomenute razlike. Kako se ni jedno poduzeće, ne osniva da bi se, ubrzo, likvidiralo, propalo već da bi „preživjelo“, opstalo u svojoj okolini, ono se, nema dvojbe, mora kontinuirano racionalno „ponašati“, ekonomizirati svojim resursima ili proizvodnim faktorima, da bi moglo dugoročno postizati što povoljniji odnos vrijednosti outputa i inputa, što je temeljna pretpostavka tog preživljavanja poduzeća.<sup>6</sup>

Budući da nema „preživljavanja“ ili opstanaka bez „rasta“ tj. razvoja poduzeća. Općeniti se cilj poduzeća (preživljavanje ili opstanak u okolini), redefinira u preživljavanje kroz rast, odnosno opstanak kroz razvoj, a operacionalizira kao konkretni zadatak dugoročnog maksimiziranja

---

<sup>4</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb.

<sup>5</sup> Ibid.

<sup>6</sup> Tintor, J. (2009): Poslovna analiza: operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia: Poslovni dnevnik, Zagreb

pozitivne razlike ili viška između vrijednosti rezultata i vrijednosti ulaganja za njegovo postignuće.<sup>7</sup>

### 2.2.2. Upravljanje poduzećem

Upravljanje poduzećem je kreativna aktivnost ili djelatnost dovođenja poduzeća iz stanja u kojem jest u stanje u kojem se želi da bude djelovanjem na ili mijenjanjem njegovih varijabli, zasnovana na procesu obrade podataka relevantnih za stjecanje i prenošenje znanja o poduzeću i okolini poduzeća kao pretpostavki i utvrđivanju ciljeva i izbora između raznih mogućnosti ili putova razvoja, te realizacije, praćenja i nadziranja provedbe najpovoljnije odluke.<sup>8</sup>

Poduzeće je nezavisna cjelina te jedan od tri pravna oblika trgovačkog društva. Odabirom organizacijsko pravnog oblika poduzeća se ujedno bira i način poreza koje će to poduzeće plaćati. Postoje tri vrste oblika poslovanja: trgovačko društvo, obrti i slobodna zanimanja te zadruge. Fokus je na trgovačkim društvima. Nakon jednostavnog društva sa ograničenom odgovornošću (j.d.o.o) i društva s ograničenom odgovornošću (d.o.o.), poduzeće prelazi na dionički oblik poslovanja. Dionički oblik poslovanja je bitan za svako poduzeće jer plasmanom dionica na tržište kapitala, poduzeće prikuplja dodatan kapital za poslovanje.<sup>9</sup>

Tipurić ističe da dioničari riskiraju gubitak samo onih novčanih sredstava kojeg su utrošili na kupnju dionica. Vlasnici mogu utjecati na upravljanje. Oni su u mogućnosti direktno utjecati na upravljanje nego preko upravljačkog odbora poduzeća.<sup>10</sup>

Nadalje, kako Batarelo objašnjava, upravljanje se – uobičajeno je reći – realizira ili ostvaruje kroz odlučivanje. Shvatilo ga se kao izbor između raznih inačica ili kao proces, u svakom mu slučaju, mora, kao odlučujuća faza, prethoditi ili biti njegov sastavni dio, odgovarajuća

---

<sup>7</sup> Batarelo, A. Ž. (2008.): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb

<sup>8</sup> Ibid.

<sup>9</sup> Kim, K. A., Nofsinger, J. R., Mohr, D. J. (2009): Corporate Governance. Prentice Hall, New Jersey

<sup>10</sup> Tipurić, D. (2007): Nadzorni odbor i korporativno upravljanje Sinergija nakladništvo d.o.o., Zagreb

informacijsko-analitička priprema. Gole činjenice, naime, bez čovjekove spoznajne analitičke, povezane s praktičnom međusobnog komuniciranja, nedovoljne su i za pripremu i za izbor i za realizaciju odluka, kao i nadziranje te realizacije.<sup>11</sup>

Na stjecanju i priopćivanju stečenog znanja utemeljena uloga upravljanja poduzećem ili odlučivanja o egzistenciji poduzeća, prema kapitalističkoj podjeli rada (jedni, ukratko, rade i spoznaju probleme, drugi, umjesto njih odlučuju), pripada posebnoj, za „čisto“ odlučivanje kvalificiranoj ili osposobljenoj skupini ljudi – menadžmentu.<sup>12</sup>

Kako menadžment, da bi realizirao svoju upravljačku ulogu, što se, u konačnici, svodi na poduzimanje svega potrebnog za dugoročno postizanje što veće pozitivne razlike između vrijednosti rezultata i vrijednosti ulaganja za njegovo postignuće, ne raspolaže za to dovoljnim specifičnim znanjem tj. angažira stručnjake koji svaki na svom području „vladaju“ teorijskim i praktičnim znanjem i metodama potrebnim da do njega dođu, što ga takvim informacijama opskrbljuju.<sup>13</sup>

Locirani u razne, ovisno o konkretnim uvjetima i potrebama svakog poduzeća, više ili manje integrirane organizacijske jedinice, ti stručnjaci, zapravo, svojim znanjem pružaju potporu iliti pomažu menadžmentu u obavljanju njegovih, tzv. Menadžmentskih ili upravljačkih funkcija, koje mora obaviti da bi poduzeće „preveo“ iz aktualnog u ciljno stanje. To su funkcije planiranja, kontroliranja, analiziranja, organiziranja i usmjeravanja.<sup>14</sup>

---

<sup>11</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb

<sup>12</sup> Ibid.

<sup>13</sup> Ibid.

<sup>14</sup> Ibid.



Slika 1. Shema povratne veze planiranja, kontrole i analize, Batarelo str. 51

### 2.2.3. Mjesto i uloga poslovne analize u procesu upravljanja poduzećem

Batarelo napominje kako poduzeće nije neprofitna ili neka slična, već s općenitim ciljem „preživljavanja“ i „rasta“, odnosno opstanka kroz razvoj, konkretiziranog u zadatku dugoročnog maksimiranja pozitivne vrijednosne razlike outputa i inputa ili što povoljnijeg odnosa vrijednosti rezultata i vrijednosti ulaganja za njegovo postignuće, osnovana organizacija, ono mora kontinuirano ekonomizirati s njima ili „štedjeti“ za to mu potrebne svoje resurse, s jednim „okom“ budno motreći i pitajući se stalno koristi li ih maksimalno ekonomski racionalno, može li i kako s njima još efikasnije i efektivnije proizvoditi i stvarati onu razliku ili „preispitivati“ odvija li se njegova djelatnost sukladno ekonomskim principima tog stvaranja.<sup>15</sup>

Tintor govori kako je ključno stvoriti sistem po kojem bi poduzeće funkcioniralo. Sistem nastaje umrežavanjem ili povezivanjem elemenata pojedinačnih sposobnosti u jedinstvenu strukturu sistema. Kvalitativno gledano, struktura pretvara skup u funkcionalno organiziranu cjelovitost i pretvara odvojene tj. ne povezane elemente u dijelove funkcionalne cjeline, u kompozitne sistema.<sup>16</sup>

<sup>15</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb

<sup>16</sup> Tintor, J. (2009): Poslovna analiza: operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia: Poslovni dnevnik, Zagreb

Također, Družinin i Kontorov napominju važnost potpunosti sistema. Potpunost sistema temelji se na zakonitosti nužne različitosti sposobnosti za transformaciju pojedinačnih sastavnica. Načelno, sistem, pa i poduzeće kao sistem, oblikuje se od različitih elemenata ili komponenti među kojima se razlikuju efektivne, receptivne i refleksivne.<sup>17</sup>

Međutim, raspoloživost komponenti koje omogućuju stvaranje potpunog sistema nije dostatna za nastanak cjeline kao djelatno sposobnog i dugotrajno uspješnog poduzeća. Poduzeće se ne uspostavlja po nekakvom genetskom kodu već smišljenom aktivnosti ljudi koji svrsishodno povezuju, primjerice, strojeve, sirovine, kapital, ljude i njihova znanja i sposobnosti.<sup>18</sup>

Batarelo zaključuje da bez analize nema spoznaje, nema znanja, lime ni njegova priopćavanja tj. „informacija“ o prošlosti, sadašnjosti i budućnosti poduzeća, prijeko potrebnih menadžmentu za postavljanje racionalnih ciljeva i iznalaženje optimalnog načina njihove realizacije; bez nje nema ni strateškog ni operativnog upravljanja poduzećem; nema ni dugoročnog maksimiranja pozitivne razlike između vrijednosti rezultata i vrijednosti ulaganja u njegovo postignuće; nema osmišljenog razvoj, čime ni ostvarenja temeljnog cilja – opstanka kroz razvoj poduzeća.<sup>19</sup>

#### 2.2.4. Poslovna analiza i planiranje i kontrola

Važnost ove cjeline najbolje opisuje ulogu poslovnog analitičara u IT poduzeću prilikom izrade poslovne aplikacije. Batarelo spominje kako osim u navedenim tzv. Vertikalnim, poslovna je analiza u horizontalnim vezama i odnosima s nizom drugih znanstvenih disciplina, a osobito je važna međuovisnost s poslovnim planiranjem i kontrolom reprodukcijskog procesa.

---

<sup>17</sup> Družinin, V. V., Kontorov, D.S. (1976): Problemi sistemologii, Svojetskoje radio, Moskva, str.48.

<sup>18</sup> Tintor, J. (2009): Poslovna analiza: operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia: Poslovni dnevnik, Zagreb

<sup>19</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb

Planiranje je, ukratko, oblikovanje ekonomske budućnosti poduzeća, u kojoj postavljene ciljeve valja, tek ostvariti. Analiza, drugačije rečeno, prethodi planu, a potom prati njegovu realizaciju zbog eventualnih korekcija. Veza i međuovisnost analize i planiranja više je nego očigledna.<sup>20</sup>

### 2.3. Vrste poslovne analize

Prema Brautu razlikuju se sljedeće vrste ekonomske analize:<sup>21</sup>

- Eksterne ekonomske analize – interne ekonomske analize
- Kompleksne ekonomske analize – posebne ekonomske analize
- Kvalitativne ekonomske analize – kvantitativne ekonomske analize
- Statičke ekonomske analize – dinamičke ekonomske analize

Prema Krajčeviću razlikuju se sljedeće vrste ekonomske analize:<sup>22</sup>

- Eksternu i internu
- Statičku i dinamičku
- Kvalitativnu i kvantitativnu
- Analizu prema raznim ciljevima u koje ubraja: analizu strukture, analizu razvoja, analizu odstupanja, analizu zavisnosti, analizu rezultata

Od ostalih poznatijih analitičara, Popović, vrste analize navodi „polazeći uglavnom od ciljeva analize i mogućnosti koje analitičaru stoje na raspolaganju. Prema Popoviću sljedeće vrste ekonomske analize:<sup>23</sup>

- Eksterna i interna
- Ekstenzivna i intenzivna
- Kompleksna i parcijalna
- Stalna i povremena

---

<sup>20</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb

<sup>21</sup> Braut, R. (1966): Ekonomska analiza poslovanja poduzeća, I.dio Visoka privredna škola u Zagrebu, Zagreb

<sup>22</sup> Krajčević, F. (1975): Analiza poslovanja organizacije udruženog rada Zagreb

<sup>23</sup> Popović, Ž. (1979): Ekonomska analiza poslovanja, Zagreb

- Analiza za kraće i duže razdoblje
- Analiza statike i dinamike
- Analiza kvalitete i kvalitete

### 2.3.1. Eksterna i interna analiza

Bit ove analize je dokumentacija koja je dostupna analitičaru (pojedincu ili stručnom timu koji obavlja analizu). Ako se za analizu koristi samo dokumentacija namijenjena javnosti (račun dobiti i gubitka, bilanca i slično), radit će se o eksternoj analizi. Ako analitičar neograničeno raspolaže potrebnom dokumentacijom, radit će se o internoj analizi.<sup>24</sup>

### 2.3.2. Ekstenzivna i intenzivna analiza

O podjeli analize prema kriteriju intenziteta (jačine, snage), na ekstenzivnu i intenzivnu, može se i vrijedi odmah reći, nema ništa ni u doajena hrvatske poslovne analize (Braut i Krajčević), a ni njihova suvremenika Vasiljevića. Ona bi, dakle, spadala u novije klasifikacije vrsta analize.<sup>25</sup>

Premda će, još 1972. godine u svojoj, kod Krajčevića prihvaćenoj i obranjenoj doktorskoj disertaciji – polazeći od logike da se, ponajprije, koristi ono što je dostupnije i jednostavnije, a samim time lakše, jeftinije i brže, daje korisne učinke, a tek kad se ove mogućnosti iscrpe, pristupa intenziviranju – Turković na naše područje uvesti diobu metoda ekonomske analize na ekstenzivne (jednostavne i jeftine, ne pretpostavljaju samo stručnost analitičara, ne traže dodatna ulaganja, a brzo daju rezultate) i intenzivne (kompliciranije i skuplje, zahtijevaju skupe stručnjake, rad dugo traje i sporije daje rezultate) metode, primat u uvođenju diobe analize prema kriteriju intenziteta pripada – Popoviću.<sup>26</sup>

---

<sup>24</sup> Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb.

<sup>25</sup> Ibid.

<sup>26</sup> Ibid.



### 2.3.3. Stalna i povremena analiza

Tintor govori da se ova podjela bazira na trajanju i kontinuitetu analize. Pri tome pod povremenom analizom podrazumijeva se samo ona koja se javlja sporadično, po potrebi ili preciznije koja se javlja bez uočenih pravilnosti u pogledu vremena i sadržaja. Pod stalnom analizom se međutim uočava takva pravilnost pa se pored redovne, kontinuirane analize pod stalnom analizom podrazumijeva i periodička mjesečna ili kvartalna analiza.<sup>27</sup>

Kako se sam proces reprodukcije odvija kontinuirano, naravno da je logično očekivati i stalno analitičko sagledavanje i utjecanje na taj proces. Podrazumijeva se stalnost u vremenu i u sadržaju. Povremena analiza bit će uzrokovana različitim potrebama kako u pogledu produbljanju spoznaje sadržaja koji je predmetom stalne analize, tako i utvrđivanje novih sadržaja.<sup>28</sup>

### 2.3.4. Kratkoročna i dugoročna analiza

Dioba na analizu za kraće i analizu za duže razdoblje, odnosno diobu s aspekta vremena obuhvaćanja analitičke građe iliti njezina vremenskog obuhvat, još je jedna dioba koju, u nas, uvodi Popović.

Podjela na analizu za kraće i duže vremensko razdoblje polazi od dužine razdoblja u kojemu se vrši ispitivanje. Analiza za kraće vremensko razdoblje se vrši: a) u toku poslovanja tj. za pojedina kraća razdoblja u okviru poslovne godine, i b) po završetku poslovne godine. U ova slučaja ova je analiza vrlo operativna i ima preventivan karakter jer joj je cilj racionalizacija tekućeg poslovanja. Naime, kada se radi o analizi u toku poslovne godine onda se kao normala za ispitivanje uzima plan i nastoji se održati planirani zadatak i planirane proporcije pa čak i postići i bolja rješenja. Kada se radi o analizi po završetku poslovne godine onda se, koristeći iskustva iz protekle godine, teži k racionalizaciji planskih zadataka i proporcija za narednu godinu.<sup>29</sup>

---

<sup>27</sup> Tintor, J. (2009): Poslovna analiza: operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia: Poslovni dnevnik, Zagreb

<sup>28</sup> Tintor, J. (1983): Uvod u ekonomsku analizu poslovanja OUR-a, Zagreb

<sup>29</sup> Popović, Ž. (1979): Ekonomska analiza poslovanja, Zagreb

#### 2.4. Poslovna analiza kao faza projekta

Poslovni analitičar u IT poduzeću je neizostavan član svakoga tima iz razloga jer spaja kupca (klijenta) s radnim timom u kojem se nalazi. Uloga poslovnog analitičara u IT poduzeću je kreirati proizvod da zadovolji najveće preferencije korisnika. Kako bi se takav rezultat postigao, potrebne su određene kompetencije i osobine koje poslovni analitičar treba imati. Kompetencije predstavljaju dinamičnu kombinaciju kognitivnih i metakognitivnih vještina, znanja i razumijevanja, međuljudskih, intelektualnih i praktičnih vještina te etičkih vrijednosti.<sup>30</sup> U ovoj cjelini će se opisati kompetencije i osobine potrebne za uspješnog poslovnog analitičara.

Za ulogu poslovnog analitičara u timu, veliku važnost se pridaje karakteristikama osobe, a prije svega se pridaje važnost o znanju iz određenog područja. Poslovni analitičar u IT poduzeću i u poduzeću koje se bavi financijama neće imati jednak opis posla, zato su bitna znanja i vještine iz određenih područja poslovanja kako bi poslovni analitičar što efikasnije i bolje obavljao svoj posao, a samim time i klijenta učinio zadovoljnijim.<sup>31</sup>

Komunikacijske vještine i osobine se odnose na način na koji se razmišlja i ponaša prema ljudima koji nas okružuju. Lakše je osobi koja ima dobre komunikacijske vještine naučiti neka nova tehnička znanja, dok osobama koje imaju već tehničko znanje je teže naučiti komunikacijskim vještinama. Jedan od razloga je što se komunikacijske vještine uče i razvijaju godinama, a nekim ljudima je takva vještina jednostavno urođena. Upravo dobre komunikacijske vještine su bitne za poslovnog analitičara. Uloga poslovnog analitičara zahtijevaju posebne tehnike poslovne analize i čine fokus operativnog znanja koje je potrebno za uspješnu provedbu očekivanja i želja.<sup>32</sup>

Bihevioralne vještine i osobine koje bi poslovni analitičar trebao imati su: razvijena komunikacija, uvažavanje mišljenja, inicijacija, društvena svijest i odgovornost, analitičke vještine i kritičko razmišljanje, vodstvo, pregovaračke sposobnosti, samouvjerenost i slično.<sup>33</sup>

---

<sup>30</sup> Agencija za znanost i visoko obrazovanje: Pojmovnik, dostupno na: <https://www.azvo.hr/hr/pojmovnik/78-kompetencije>, pristupljeno: 5.6.2020.

<sup>31</sup> Kupper, H.U.(1995): Controlling, Stuttgart,Schaffer/Poeschel

<sup>32</sup> Ibid.

<sup>33</sup> Ibid.

Poslovna znanja potrebna za poslovnog analitičara: financije i ekonomija, bankarstvo, programiranje, razvoj poslovnih slučajeva i slično.<sup>34</sup>

Tehnike potrebne za poslovnog analitičara: Upravljanje projektima, analiza strategije, tehnike istraživanja, modeliranje i slično.<sup>35</sup>

U nastavku ovoga rada će se pričati detaljnije o vještinama potrebnim za poslovnog analitičara.<sup>36</sup>

#### 2.4.1. Bihevioralne vještine i osobine

Komunikacija obuhvaća široki spektar vještina kao što su slušanje, izgradnja odnosa, utjecanje i izgradnja suosjećajnosti. Bit posla poslovnog analitičara podrazumijeva prikupljanje i analiziranje podataka tj. informacija zbog shvaćanja problema s kojim se klijent susreće. Izgradnja odnosa s klijentom predstavlja važan element komunikacijskih vještina i svodi se na vještinu adekvatnog snalaženja s drugim ljudima, na društvenom i profesionalnom nivou. Poslovni analitičar treba izgraditi dubinski i iskren odnos s klijentom da bi mogli dijeliti informacije, da dijele ideje i da razgovaraju o mogućim promjenama. Cijeli taj proces je mnogo jednostavniji ako klijenti vjeruju poslovnom analitičaru.<sup>37</sup>

Priroda posla poslovnoga analitičara je takva da je timski rad najčešći oblik rada jer je poslovni analitičar u stalnoj komunikaciji ili s klijentima ili sa svojim timom. Od poslovnog analitičara se očekuje da prikuplja podatke i informacije od raznih izvora te delegiranje zadataka unutar tima radi postizanja ciljeva projekta.<sup>38</sup>

Poslovni analitičar ne bi smio prihvaćati stvari onakve kakve jesu i ne donositi nagle zaključke. Kritičko razmišljanje i analitičke sposobnosti znače ne pristati na očito. Potrebno je što

---

<sup>34</sup> Kupper, H.U.(1995): Controlling, Stuttgart,Schaffer/Poeschel

<sup>35</sup> Ibid.

<sup>36</sup> Ibid.

<sup>37</sup> Paul, D., Yeates, D.,Cadle, J., (2010): Business analysis, Swindon,BISL

<sup>38</sup> Ibid.

temeljitiše istraživanje dok se stvari ne razjasne u potpunosti kako ne bi došlo pogrešaka u procesu.<sup>39</sup>

#### 2.4.2. Poslovna znanja

Kako bi se uspostavio kredibilitet s klijentima, važno je dobro razumijevanje poslovnog područja u kojem se radi. Poslovni analitičari trebaju biti stručnjaci za određene poslovne segmente ili mogu biti generalisti. Ne postoji pravilan odgovor da se odredi da li je bolje biti specijalist ili generalist. Oba su korisna, u zavisnosti od organizacijskog konteksta. Bitna je procjena koliko dobro kompetencije poslovnog analitičara ispunjavaju potrebe stvarne situacije i da se prepoznaju nedostaci i prednosti.<sup>40</sup>

#### 2.4.3. Tehnike

Upravljanje projektima uključuje različite procese kao npr. upravljanje prostorom, integracijama, troškovima, vremenom, ljudskim resursima, kvalitetom, rizicima i sl. Rijetko koji poslovni analitičar će imati sve navedene vještine, ali kada je tim mali, poslovni analitičar će ponekad morati preuzeti ulogu voditelja projekta. Kod velikih projekata će se često zapošljavati stručnjak u vođenju projekata. Važno je da analitičar shvati sve faze projekta kako bi mogao pristupiti što kvalitetnijem planiranju upravljanja projektima.<sup>41</sup>

Za razvoj karijere bilo kojeg stručnjaka, razvoj kompetencija je najznačajniji aspekt. Poslovni analitičari se jedino mogu razvijati i uspjeti ako se nudi različiti spektar kompetencija koje omogućavaju da dokažu kako prepoznaju, razvijaju i analiziraju opcije za dodavanje vrijednosti organizacijama.<sup>42</sup>

---

<sup>39</sup> Paul, D., Yeates, D., Cadle, J., (2010): Business analysis, Swindon, BISL

<sup>40</sup> Ibid.

<sup>41</sup> Ibid.

<sup>42</sup> Ibid.

### 3. MODELIRANJE BAZE PODATAKA

#### 3.1. Objašnjenje pojma modeliranje podataka

##### 3.1.1. Modeliranje podataka

Modeliranje podataka je postupak pronalaženja kategorija podataka i odnosa među njima.. Opisi podataka, odnosno modeli podataka, mogu pripadati logičkoj ili fizičkoj razini. Opis podataka koji se razlikuje po interpretaciji podataka i razini apstrakcije opisan je različitim formalnim modelima podataka <sup>43</sup>

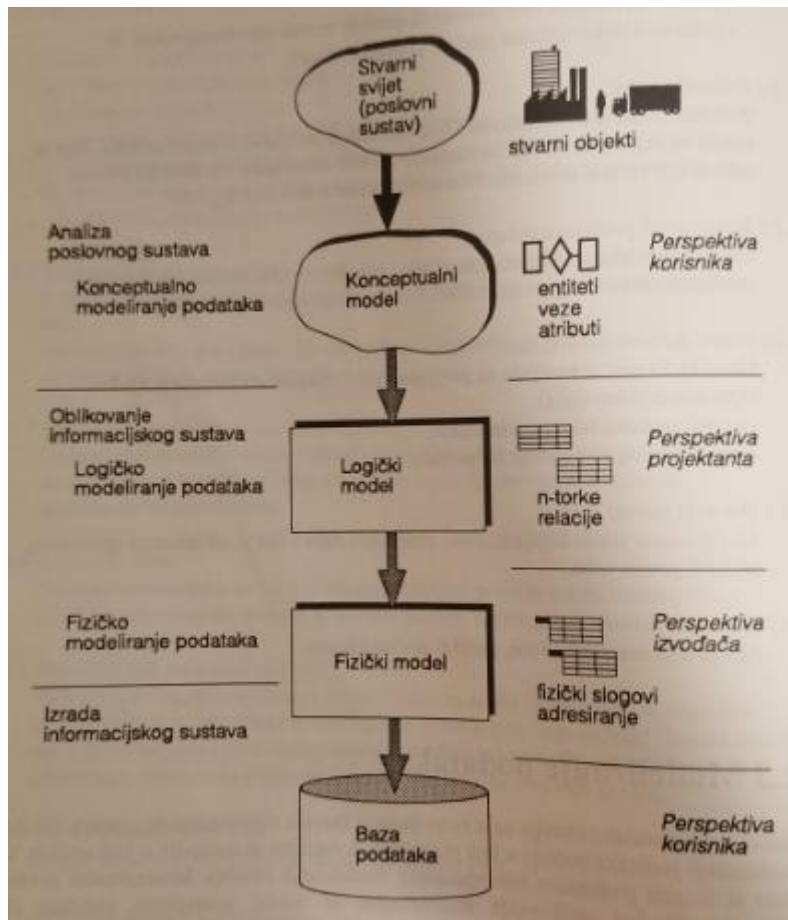
Razina apstrakcije je podjeljena na konceptualnu, logičku i fizičku apstrakcije. Pripadajući modeli se nazivaju konceptualni, logički i fizički model. Entitet, veza entiteta, atribut, vrijednost atributa su neki od pojmova koji se mogu pronaći u konceptualnome modelu. Logički slog, polje, veza između slogova su pojmovi koji se vežu s logičkim modelom dok fizički slog, adresiranje slogova i grupiranje slogova se povezuje s fizičkim modelom.

Varga ističe kako se razvoj informacijskog sustava odvija u svim fazama modeliranje podataka. U fazi planiranja se dešava grubo modeliranje, a u fazi analize se detaljno nastavlja modeliranje. Konceptualni se model postupkom logičkog modeliranja pretvara u logički model u fazi oblikovanja, a fizički model nastaje postupkom fizičkog modeliranja. Baza podataka je rezultat fizičko modela. Ona se formira u fazi izrade informacijskog sustava, a detaljno provjerava i uvodi u rad u fazi uvođenja. Postupak modeliranja podataka prikazan je i na slici 2.<sup>44</sup>

---

<sup>43</sup> Srića, V. (1988): Uvod u sistemski inženjering. Informator, Zagreb

<sup>44</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb



Slika 2. Od stvarnog svijeta do baze podataka, Varga (1994.), Baze podataka, str. 40

Logička struktura baze podataka određuje model podataka, koji čini osnovu za implementiranje i projektiranje baze. Sustav za upravljanje bazom zahtjeva od podataka da u bazi moraju biti logički organizirani. Sadašnji sustavi za upravljanje bazom podržavaju neke od ovih modela:

- Relacijski model.
- Mrežni model.
- Hijerarhijski model
- Objektni model

Mrežni i hijerarhijski model bili su u upotrebi sredinom 20. stoljeća. Od 1980. godine pa sve do danas prevladava relacijski model.<sup>45</sup>

### 3.2. Objašnjenje pojma baze podataka

Prema Vargi, kolekcija podataka strukturiranih u skladu s fizičkim, logičkim i konceptualnim modelom podataka informacijskog sustava naziva se baza podataka.<sup>46</sup>

Date definira baze podataka kao skup integriranih i operativnih podataka.<sup>47</sup>

Martin u svojoj knjizi definira baze podataka kao skup međuovisnih podataka koji su spremljeni bez zahilosti te koji služe jednoj ili više aplikacija na adekvatan način i gdje su podaci neovisni od programa kojima se obrađuju.<sup>48</sup>

Maleković i Rabuzin definiraju baze podataka kao kolekcija podataka, ograničenja i operacija koja reprezentira neke aspekte realnog svijeta. Dio realnog svijeta, koji je od interesa za razmatranje, naziva se aplikacijskom domenom, a reprezentiranje aplikacijske domene predstavlja zapravo njeno modeliranje. Jedan od mogućih modela aplikacijske domene je model pomoću baze podataka. Dakle, modeliranjem aplikacijske domene AD dolazi se do baze podataka BP koja je model za AD.<sup>49</sup>

---

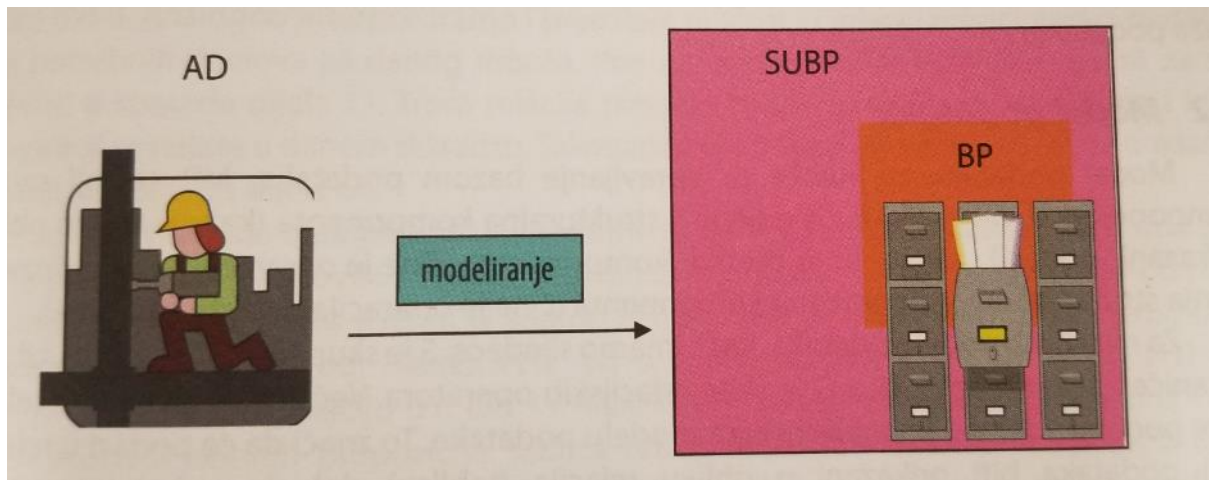
<sup>45</sup> Manger, R. (2010): Osnove projektiranja baza podataka, dostupno na: [https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310_polaznik.pdf), pristupljeno 1.7.2020.

<sup>46</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>47</sup> Date, C. J. (1990): An Introduction to Data Base System, vol. 1., Addison – Wesley

<sup>48</sup> Martin, J. (1977): Computer Data-base Organization, 2<sup>nd</sup> ed. Prentice-Hall

<sup>49</sup> Maleković M., Rabuzin K. (2016): Uvod u baze podataka, Mini-Print-Logo, Varaždin



Slika 3. Modeliranje aplikacijske domene, Maleković, 2016. Str.1

### 3.3. Konceptualno modeliranje podataka

Varga govori da od specifikacije informacijskih zahtjeva koje nastaje konceptualno modeliranje. Rezultat toga je izrađeni konceptualni model podataka. Konceptualni model podataka je konzistentan, neredundantan i cjelovit opisi podataka u informacijskom sustavu.<sup>50</sup>

#### 3.3.1. Model entiteti-veze

Model entitet-veze promatra svijet kroz entitete, njihov odnos, te atribute entiteta ili njihova odnosa. U promatranju svijeta koriste se razni postupci. Jedan od njih je postupak apstrakcije. Apstrakcija je označava prepoznavanje glavnog ili općeg, a odbacivanje sporednog ili nebitnog. Klasifikacija, generalizacija i agregacija su najbitniji procesi u postupku modeliranja podataka.<sup>51</sup>

Entiteti se klasificiraju, opisuju i grupiraju u klase, razrede tj. tipove prema obilježjima koji bi im mogli biti zajednički. Takva vrsta apstrakcije naziva se klasifikacija. Tako se npr. imena

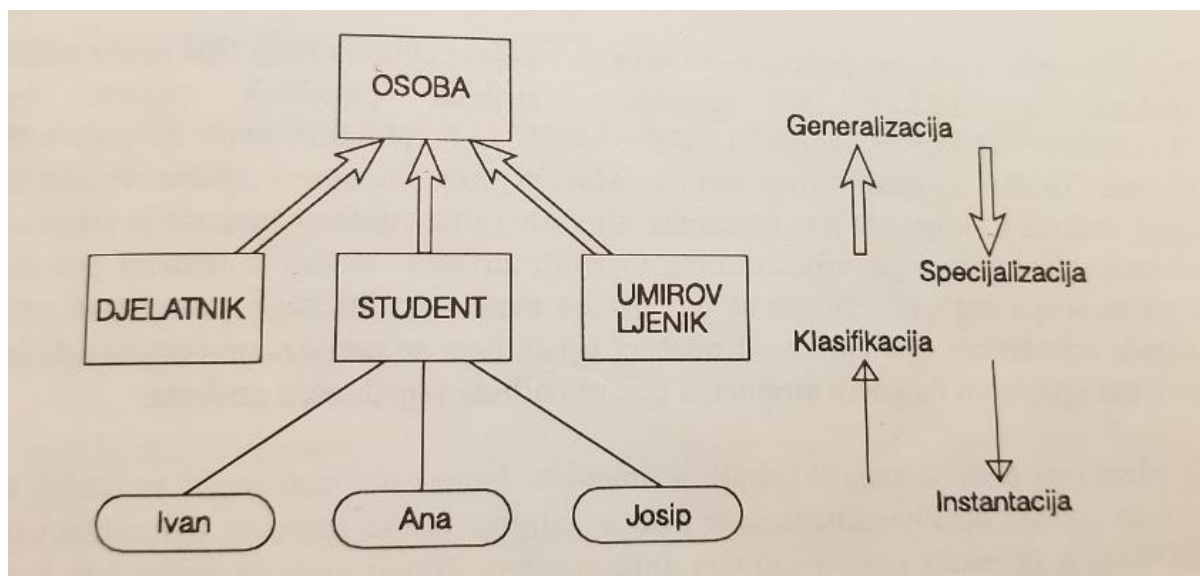
<sup>50</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>51</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb



studenta mogu klasificirati u tip entiteta student. Instancija je reverzibilni postupak, od tipa entiteta prema njegovim pojavama.<sup>52</sup>

Proces gdje se tipovi entiteta niže razine poopćuju tipom entiteta više razine naziva se generalizacija. Npr. tipovi entiteta djelatnik, student, i umirovljenik generaliziraju se tipom entitet osoba. Obrnut postupak naziva se specijalizacijom. U ovom primjeru osoba se specijalizira na djelatnika, studenta ili umirovljenika.<sup>53</sup>



Slika 4. Klasifikacija i generalizacija, Varga (1994.) str. 45

Stvaranje novog pojma na temelju odnosa postojećih pojmova naziva se agregacija.

Definicija entiteta prema Vargi glasi da je entite opipljiv ili figurativni predmet ili događaj o kojemu se u informacijskom sustavu prikupljaju podaci. Kuća, osoba, automobil i slično su primjeri entiteta. Svi ovi entiteti imaju svoje atribute odnosno obilježja. Atribut je obilježje entiteta ili veze između entiteta. Veza označava međusobni odnos između dva ili više. Broj entiteta koji sudjeluju u vezi naziva se stupanj veze.<sup>54</sup>

<sup>52</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>53</sup> Ibid.

<sup>54</sup> Ibid.

Matematičkim pojmom pridruživanja elemenata jednog skupa elementima drugog skupa može se objasniti veza.. Za svrhu uvodi se pojam: jednoznačnog, uvjetnog i višeznačnog pridruživanja.<sup>55</sup>

Koncept ključa se koristi za identifikaciju pojave entiteta. Skup atributa čije vrijednosti omogućuju jednoznačnu identifikaciju svake pojave tipa entiteta naziva se ključem. Postoje više vrsti ključeva koje se koriste. Najčešće korišteni ključevi su primarni i strani ključ. Polje ili skup polja s kojima se kontolira način kako se informacije povezuju s drugim tablicama naziva se primarni ključ. Samo jedan primarni ključ je moguć u svakoj tablici<sup>56</sup> Strani ključ je polje u sekundarnoj tablici s kojom se vrši povezivanje i ne mora imati jedinstvenu vrijednost.<sup>57</sup>

### 3.3.2. Izrada modela entitet-veze

Prepoznavanje informacijskih zahtjeva smatra se korakom u analizi poslovnog sustava koji rezultira izradom modela entiteti-veze. Informacijski zahtjevi zapravo opisuju poslovna pravila koja vrijede unutar poslovnog sustava. Njih iskazuju sami korisnici informacijskog sustava u narativnom obliku, čime daju definiciju informacijskih zahtjeva. Na temelju njih analitičar izrađuje formalnu specifikaciju informacijskih zahtjeva.<sup>58</sup>

Model entiteti-veze prikazuju entitete i njihovu strukturu, a za prikaz pravila obavljanja i pokretanja procesa koriste se tehnike za opis procesa i događaja. Sam model entiteti-veze koncentrira se na identifikaciju entiteta informacijskog sustava i identifikaciju odnosa (veza) među entitetima.<sup>59</sup>

---

<sup>55</sup> Blanuša, D. (1965): Viša matematika. Tehnička knjiga, Zagreb

<sup>56</sup> Microsoft: Dodavanje i izmjena primarnog ključa tablice u programu Access, dostupno na: <https://support.office.com/hr-hr/article/dodavanje-i-izmjena-primarnog-klju%C4%8Da-tablice-u-programu-access-07b4a84b-0063-4d56-8b00-65f2975e4379>, pristupljeno 1.7.2020

<sup>57</sup> Varga, M. (1994.): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>58</sup> Teorey, T. J., Yang, D., Fry, J. P. (1986): A Logical Desing Methodology for Relational Databases Using the Extended Entity – relationship model. ACM Computing Surveys, vol.18, no. 2.

<sup>59</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

Postupak izrade modela entiteti-veze:<sup>60</sup>

- Prikupiti i analizirati informacijske zahtjeve,
- Izraditi model entiteti-veze:
  - Utvrditi entitete
  - Utvrditi vezu među entitetima
  - Utvrditi ključeve
  - Utvrditi ograničenja unosa, brisanja i promjene ključeva
  - Utvrditi attribute entiteta
- Konsolidirati model entiteti-veze

### 3.4. Logičko modeliranje podataka

Logičko oblikovanje baze podataka temelji se na zavisnostima baza podataka. Riječ je o zavisnostima među atributima relacijske sheme. Ove zavisnosti predstavljaju uvjete integriteta, tj. ograničavaju moguća stanja relacija. Prisustvo određenih, nepoželjnih zavisnosti u relacijskoj shemi zahtijeva dekompoziciju dane relacijske sheme što je temelj za postupak poznat kao normalizacija relacijske sheme.<sup>61</sup>

#### 3.4.1. Relacijski model podataka

Codd je teoretski razradio relacijski model. Njegova osnovna kvaliteta je jednostavnost, kako u praktičnom radu tako i u razumijevanju. Matematički temelji omogućavaju precizno definiranje modela.<sup>62</sup>

Iz matematičkih teorija skupova, relacijski model je preuzeo osnovne koncepte. Ti koncepti su domena, relacija i atribut.<sup>63</sup>

---

<sup>60</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>61</sup> Maleković M., Rabuzin K. (2016): Uvod u baze podataka, Mini-Print-Logo, Varaždin

<sup>62</sup> Codd, E. F. (1970.): A Relation Model of Data for Large Shared Data Banks. Communications of ACM, 13.

<sup>63</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

Imenovani skup vrijednosti naziva se domenom (npr. Naziv\_mjeseca\_u\_godini: Svibanj, Lipanj, Srpanj... itd. ili npr. domena broj\_dana\_u\_mjesecu = {1..31})<sup>64</sup>

Imenovani stupac relacije naziva se atributom. Skup različitih relacijskih shema jest relacijska shema baze podataka. Dvije su relacijske sheme različite ako su im skupovi atributa različiti.<sup>65</sup>

### 3.4.2. Normalizacija

Skup podataka koji je neredundantan i sadrži semantički integritet predstavlja dobro oblikovanu relacijsku bazu podataka tj. ne prikazuje anomalije pri unosu, brisanju ili promjeni podataka. Ako se podatak memorira na više mjesta, onda se govori o redundanciji.<sup>66</sup>

Dobro oblikovana relacijska baza podataka treba biti u prikladnoj normalnoj formi. Normalne forme definiraju ograničenja podataka u relacijama. Time se osigurava konzistentnost relacija i njihovo jednostavno korištenje, a izbjegavaju moguće anomalije pri unosu, brisanju ili promjeni. Normalna forma definira se na relacijskoj shemi, a primjenjuje na relaciji koju shema opisuje. Normalna forma može se proširiti na čitavu relacijsku shemu baze podataka.<sup>67</sup>

---

<sup>64</sup> Freeservers: Relacijski modeli podataka, dostupno na: <http://tecajevi.freeservers.com/isrelac.htm> – pristupljeno 09.09.2020

<sup>65</sup> Varga, M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb

<sup>66</sup> Ibid.

<sup>67</sup> Ibid.

## 4. PROGRAMSKO INŽENJERSTVO KAO TEMELJ USPJEŠNOG RAZVOJA SOFTWARE-A

### 4.1. Uvod u programsko inženjerstvo

Elektronička računala su postala neizostavni dio naše svakodnevnice zbog uzbudljivih, zabavnih i praktičnih primjena. Primjeri takvih primjena su igranje igrice na mrežnim platformama, bankarske transakcije, pretraživanje digitalnih sadržaja i slično. Sve to ne bi bilo moguće bez programa predviđenih za to. Kreiranje programa naziva se programiranje. Programiranje je pisanje linija koda s uputama računalo kako i što činiti u određenim situacijama.<sup>68</sup>

Programiranje je vrlo dinamična tehnička disciplina, u kojoj se stalno događaju promjene. Promjene u smislu preslagivanja i izmjena. Relativna mladost je razlog dinamike ovoga područja. Ideje automatske obrade podataka javljaju se krajem 19. stoljeća. Područje doživljava procvat u drugoj polovici 20. stoljeća.<sup>69</sup>

### 4.2. Objasnjene pojma algoritma

Algoritam je konačan skup jednoznačno definiranih, logički povezanih koraka (postupaka). Ima jasno definirane ulazne i izlazne veličine te mora biti primjenjiv za rješavanje određenog tipa problema, ne samo jednog konkretnog problema.<sup>70</sup>

Algoritme se može zapisivati na više načina. Zapisi algoritma mogu biti:<sup>71</sup>

- u obliku teksta,
- pseudokodom,
- s pomoću dijagrama toka (grafički),
- zapis u programskom jeziku.

---

<sup>68</sup> Kalafatić Z., Pošćić A., Šegvić S., Šribar J. (2016): Python za znatiželjne: sasvim drukčiji pogled na programiranje, 1. izd., Element, Zagreb

<sup>69</sup> ibid

<sup>70</sup> Kučak, D. (2013): Izrada web-aplikacija: priručnik, Algebra učilište, Zagreb

<sup>71</sup> Ibid.

Svojstva algoritma su:<sup>72</sup>

- algoritam za problem na koji se primjenjuje mora davati točan izlaz na sve njegove dozvoljene ulaze,
- algoritam ne smije biti beskonačan – do rješenja problema mora se doći u konačnom broju koraka,
- svaki algoritam mora imati točno definirano što su mu ulazne veličine, a što izlazne veličine,
- naredbe u algoritmu moraju biti jasne, jednoznačno definirane i ne smiju biti dvosmislene.

### 4.3. Modeliranje poslužitelja (Backend as a Service - BaaS)

#### 4.3.1. Backend as a Service (Baas)

Modeliranje poslužitelja je koncept umrežavanja mobilnih i web aplikacija s pozadinskim uslugama temeljenim na oblaku.<sup>73</sup> Prvo će biti objašnjen pojam „na oblaku“. Cloud ili računalstvo u oblaku koristi Internet kao platformu za rad. Podatkovna veza je temelj za takav oblik računalstva. Ukoliko je omogućen pristup internetu, pristup svojim podacima, aplikacijama i dokumentima će biti moguć. Postoji mogućnost povećanja i smanjenja kapaciteta clouda, pa samim time to ima financijski aspekt u poduzeću radi smanjena troškova.<sup>74</sup>

Tri glavne vrste računalstva u oblaku uključuju Infrastructure as a Service (IaaS), Platform as a Service (Paas) i Software as a Service (SaaS). Svaka vrsta računalstva u oblaku pruža različite

---

<sup>72</sup> Kučak, D. (2013): Izrada web-aplikacija: priručnik, Algebra učilište, Zagreb

<sup>73</sup> Monroe, M. (2013): The Gospel of MBaaS (Part 1 of 2), InfoQ, dostupno na: <https://www.infoq.com/news/2013/05/MBaaS-Anypresence>, pristupljeno: 24.06.2020.

<sup>74</sup> Sales force: Why Move To The Cloud? 10 Benefits Of Cloud Computing, dostupno na: <https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html>, pristupljeno 24.06.2020.

razine kontrole, fleksibilnosti i upravljanja tako da se može odabrati pravi skup usluga za svoje potrebe.<sup>75</sup>

Backend, koji se još naziva programiranje na strani poslužitelja, događa se na serveru i u bazi podataka. To je mehanizacija koja djeluje iza scene kako bi omogućila korisnicima komunikaciju na strani klijenta (frontend).<sup>76</sup>

#### 4.4. Modeliranje sučelja (frontend)

Frontend web development je praksa pretvaranja podataka u grafičko sučelje kroz korištenje HTML-a, CSS-a, i JavaScript-a tako da korisnik može vidjeti i imati interakciju s podacima.<sup>77</sup>

Frontend ili kako se još naziva GUI (Graphical User Interface) je sučelje koje posjetitelji na web stranici ili aplikaciji mogu vidjeti i koristiti. Frontend se uglavnom koristi za prikazivanje različitih vrsta sadržaja i omogućavanje korisnikova inputa da budu dostupni backendu. Sučelje uključuje tekst, grafiku, video i ostale medije.<sup>78</sup>

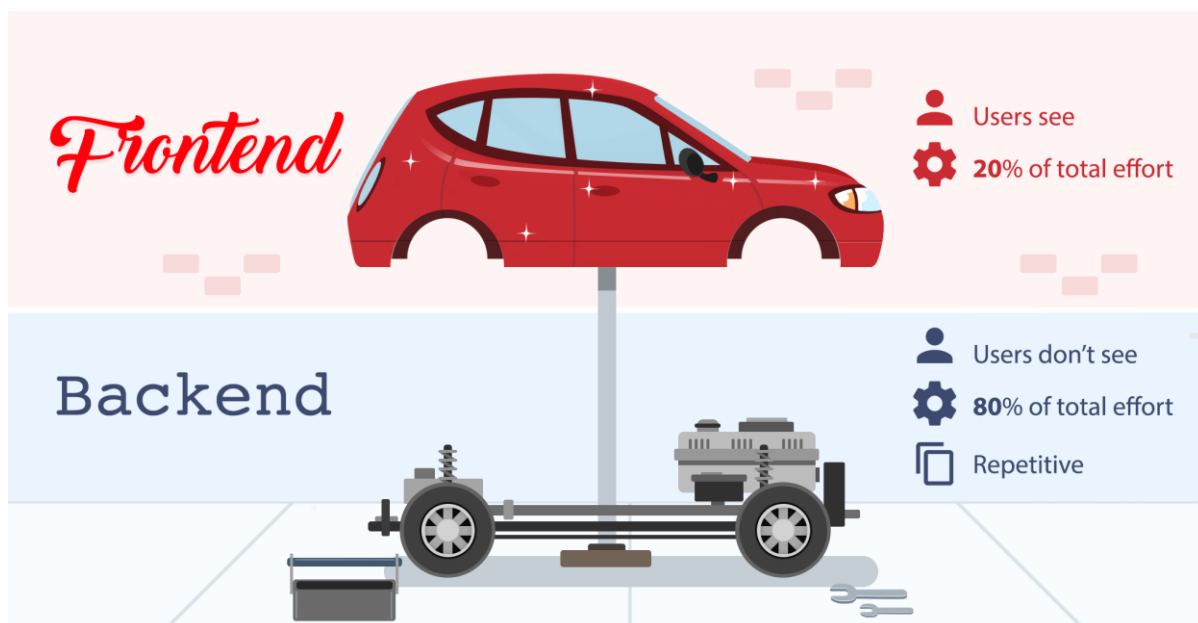
---

<sup>75</sup> Amazone web shop: What is cloud computing?, dostupno na: <https://aws.amazon.com/what-is-cloud-computing/>, pristupljeno 24.06.2020.

<sup>76</sup> Sagara Idea Lab (2019): The fundamentals of front end and back end development, dostupno na: <https://medium.com/@sagarajkt/the-fundamentals-of-front-end-and-back-end-development-5973ac0910cf>, pristupljeno 24.06.2020.

<sup>77</sup> Codesido, I. (2009): What is front-end development?, dostupno na: <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>, pristupljeno 24.06.2020

<sup>78</sup> RYTE: Frontend, dostupno na: <https://en.ryte.com/wiki/Frontend>, pristupljeno 24.06.2020.



Slika 5. Ilustracija na prikaz razlike između frontenda i backenda, slika preuzeta s <https://blog.back4app.com/wp-content/uploads/2019/07/backend-vs-frontend-1.png> [1.7.2020.]

HTML jezik je temelj svake web-stranice. HTML je jednostavan opisni jezik kojim se opisuje raspored sadržaja na stranici. HTML kratica dolazi od HyperTextMarkup Language, a HTML datoteke su obične tekstualne datoteke. Kada se u preglednik upiše adresa neke web-stranice, tada se od servera dobiva HTML datoteku. Ta se datoteka čita, njen kod interpretira i preglednik prikazuje sadržaj na ekranu. Postoji čitav niz HTML elemenata kojima se može prikazati sadržaj, no najvažniji element svake web-stranice je link. Linkovima se povezuje stranica s nekim drugim stranicama i tako se stvara kompaktna cjelina. Oni su ono što povezuje cijeli web i da nekim čudom nisu izmišljeni, adresu svake stranice koju se želi posjetiti moralo bi se upisivati u adresnu traku preglednika. HTML nije programski jezik, već opisni, jer se njime opisuje izgled i sadržaj web-stranice.<sup>79</sup>

CSS (Cascadin Style Sheets) su se pojavili 1996. godine i predstavili su potpuno novi način pogled na web stranice i prikaz sadržaja. CSS je omogućio složeniji dizajn i njegov jednostavan način održavanja na web stranicama. Ugrubo gledano, CSS na jednostavan način omogućuje da se razdvoji izgled web stranice od njenog sadržaja.<sup>80</sup>

<sup>79</sup> Abrus, L. (2003): Izrada weba: abeceda za webmastere: kompletan vodič. Bug: SysPrint, Zagreb

<sup>80</sup> Ibid.



Frontend i backend igraju važnu ulogu u razvoju web stranice iako imaju značajne razlike. Frontend govori o vizualnim aspektima web-stranice koju korisnik može vidjeti. S druge strane, sve što se dešava u pozadini (traženje određene stranice, poziv na server itd.) može se pripisati backendu. Backend bi se mogao opisati kao pokretač za frontend.<sup>81</sup>

---

<sup>81</sup> Sagara Idea Lab (2019): The fundamentals of front end and back end development, dostupno na: <https://medium.com/@sagarajkt/the-fundamentals-of-front-end-and-back-end-development-5973ac0910cf>, pristupljeno 24.06.2020.

## 5. OSIGURANJE KVALITETE PROGRAMSKE PODRŠKE

Prije nego što se započne tema o osiguranju kvalitete, treba definirati programsku podršku. Prema Slaveku, sve metode, alati i procedure razvoja imaju samo jedan cilj: proizvesti kvalitetnu programsku podršku – softver. Programska podrška nevidljiva je, samo su funkcije, kontrole, podatci i korisnička sučelja dostupni u obliku tekstova ili dijagrama. Programska podrška nematerijalna je i zbog toga teško ju je mjeriti.<sup>82</sup>

Teško je upravljati nečim što se može izmjeriti. Manjak pouzdanih mjerila proizvoda i procesa otežava kontrolu, a samim time i planiranje. Kvaliteta isporučene programske podrške ne može se jamčiti. Zamjena je niska i proizvodnost nije kvantificirana, a troškovi procjene i razvoja isporuke obično su nerealni.<sup>83</sup>

### 5.1. Specifičnost programske podrške i kvalitete

#### 5.1.1. Što je kvaliteta programske podrške?

Da bi ponudili odgovor na navedeno pitanje, prvo treba odgovoriti što je kvaliteta, a zatim što je programska podrška.

Prema Webster's Unabridged Dictionary kvaliteta je bilo koja karakteristika ili karakteristike koje mogu učiniti objekt dobrim ili lošim, hvale vrijednim ili vrijednim osude.<sup>84</sup>

Ako se kvalitetu definira s tradicionalnoga tehnološkoga stajališta, tada su prikladna neka razmatranja M. J. Jurana.

Osnovni temeljni blok na kojem se izgrađuje sposobnost za uporabu jest karakteristika kvalitete. Bilo koje obilježje (svojstvo, atribut, itd.) proizvoda, materijala ili procesa koji su potrebni da bi se postigla sposobnost za uporabu jest karakteristika kvalitete.<sup>85</sup>

---

<sup>82</sup> Slavek, N. (2016): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

<sup>83</sup> Ibid.

<sup>84</sup> Merriam webster: Quality, dostupno na: <https://www.merriam-webster.com/dictionary/quality#examples>, pristupljeno 1.7.2020

<sup>85</sup> Juran Joseph, M. (2010): Juran's Quality Handbook: The Complete Guide to Performance Excellence, VI-Edition, McGraw-Hill

## 5.2. Životni ciklus programske podrške

Razvoj programske podrške može biti vrlo složen i uvijek postoje različiti načini rješavanja razvoja. Inženjerstvo programske podrške disciplina je koja uključuje tri glavna elementa – metode, alate i procedure. Ono omogućava menadžerima kontrolu procesa razvoja, a praktičarima daje temelje za izgradnju visoko kvalitetne programske podrške. Procedure inženjerstva programske podrške omogućuju racionalan i pravovremen razvoj programske podrške. One definiraju redoslijed primjene metoda, dokumentaciju za isporuku, kontrole koje pomažu osiguranju kvalitete i koordinaciju promjena. Sustava, slijedni proces razvoja programske podrške naziva se životni ciklus programske podrške. Modeli procesa razvoja mogu se definirati na trima razinama: U (Universal) razini, W (Worldly) razini i A (Atomic) razini.<sup>86</sup>

### 5.2.1. Vodopadni model

Vodopadni model, iako najstariji, još je uvijek najpoznatiji i najčešće korišteni U-model procesa razvoja programske podrške. Razvoj započinje uspostavljanjem zahtjeva za sustav. Sljedeći je korak analiza zahtjeva za sustav i programsku podršku. Razvojni inženjer odnosno analitičar mora razumjeti domenu informacija, potrebne funkcije, performanse i sučelja programske podrške. Prema analizi zahtjeva izrađuje se dokument specifikacije zahtjeva za programsku podršku. Za izradu toga dokumenta može poslužiti norma ANSI/IEEE Std. 830, Specifikacija zahtjeva za programsku podršku.<sup>87</sup>

Dizajn programske podrške svodi se na arhitekturu programske podrške, strukturu podataka i procedure. Sljedeća je etapa kodiranje, pri čemu se dizajn prevodi u računalni kod. Nakon kodiranja započinje se testiranje. Prethodno je potrebno odrediti strategiju testiranja. Strategija testiranja programske podrške treba obuhvatiti najniže razine programske podrške kako bi se utvrdilo da je i najmanji odsječak koda točno razvijen. Isto tako treba i obuhvatiti i najviše razine kako bi se odredile glavne zadaće sustava prema zahtjevima korisnika.

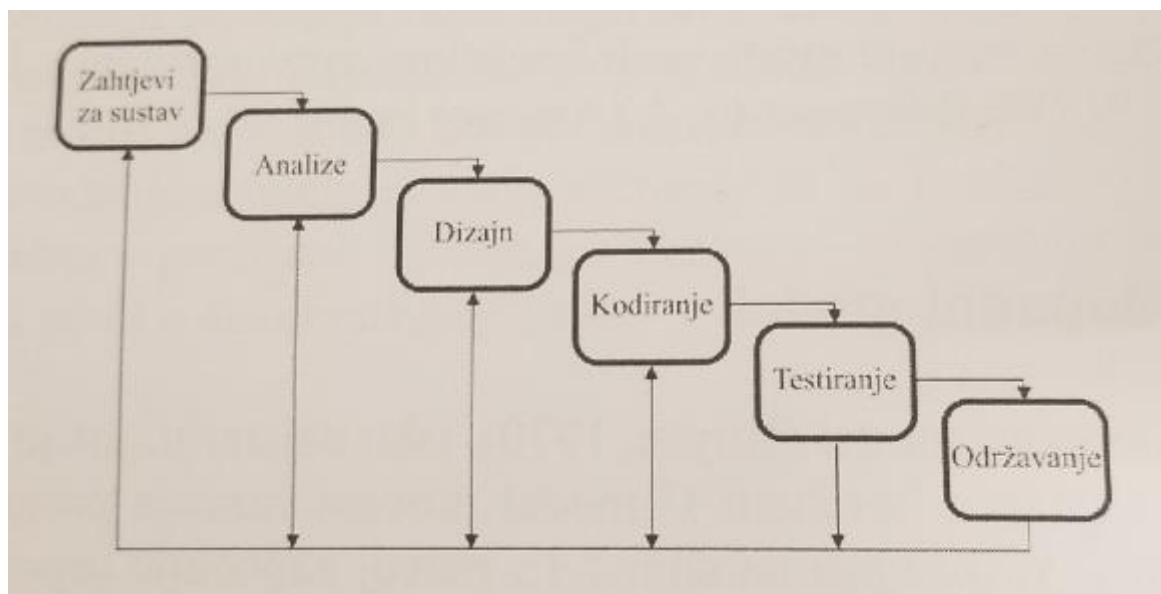
---

<sup>86</sup> Slavek, N. (2016): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

<sup>87</sup> Slavek, N. (2016): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

Testiranje programske podrške element je sustava verifikacije i validacije, čime se verifikacijom osigurava da programska podrška točno implementira određenu zadaću, dok se validacijom osigurava da programska podrška udovoljava svim korisničkim zahtjevima.<sup>88</sup>

Zadnja je etapa održavanje programske podrške nakon isporuke korisniku. Programska podrška mijenjat će se ako korisnik traži neke nove funkcije, ako se tijekom korištenja pronađu pogreške, ili se programska podrška mora prilagođavati novom korisničkom okolišu.<sup>89</sup>



Slika 6. Vodopadni model, Ninoslav Slavek, str. 86

Prednosti ovog modela su što omogućava departmentalizaciju i kontrolu. Raspored može biti točno definiran kao i zadani rokovi za svaku fazu razvoja projekta. Jednostavan za korištenje i lagan za shvatit. Lagano je upravljati s njime. Faze i procesi se završavaju jedan po jedan (sljedeća faza ne može krenuti dok prijašnja nije završena). Jasno definiranje i delegiranje zadataka.<sup>90</sup>

<sup>88</sup> Slavek, N. (2016): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

<sup>89</sup> Royce, W.W. (1970): Managing the development of large software systems, dostupno na: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf> – pristupljeno 1.7.2020.

<sup>90</sup> Tutorials point: Waterfall model, dostupno na: - [https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm), pristupljeno 24.06.2020

Nedostaci ovoga modela su da ovaj model ne dozvoljava provjeru kada dođe u fazu testiranja jer se jako teško vratiti nazad i promijeniti nešto što nije dobro dokumentirano ili osmišljeno u konceptu razrade. Ako se pronađe pogreška tijekom razvoja, proces je potrebno obnoviti od točke gdje je pogreška pronađena.<sup>91</sup>

### 5.2.2. Prototip

Model prototipa daje veću fleksibilnost od vodopadnog modela jer omogućava izravno uključivanje korisnika u vrlo ranim etapama razvoja sustava. Na taj način omogućava se validacija zahtjeva za programsku podršku i dizajn.<sup>92</sup>

Na temelju zahtjeva korisnika moguće je brzo izgraditi prvi prototip koji korisnik može analizirati i dati svoje prijedloge za usavršavanje. Proces se ponavlja sve do zadovoljenja korisnika.<sup>93</sup>

Ovisno o cilju koji je potrebno postići prototipom, razlikuje se: istraživački prototip za potrebe razjašnjenja zahtjeva za programsku podršku, eksperimentalni prototip za ispitivanje predloženog rješenja, evolucijski prototip za prilagodbu sustava promjenama zahtjeva. Nadogradnja je prototipnoga pristupa spiralni model.<sup>94</sup>

### 5.2.3. Spiralni model

Radi prevladavanja nedostataka vodopadnog modela Boehm je predložio spiralni model koji također pripada u U-modele.<sup>95</sup>

---

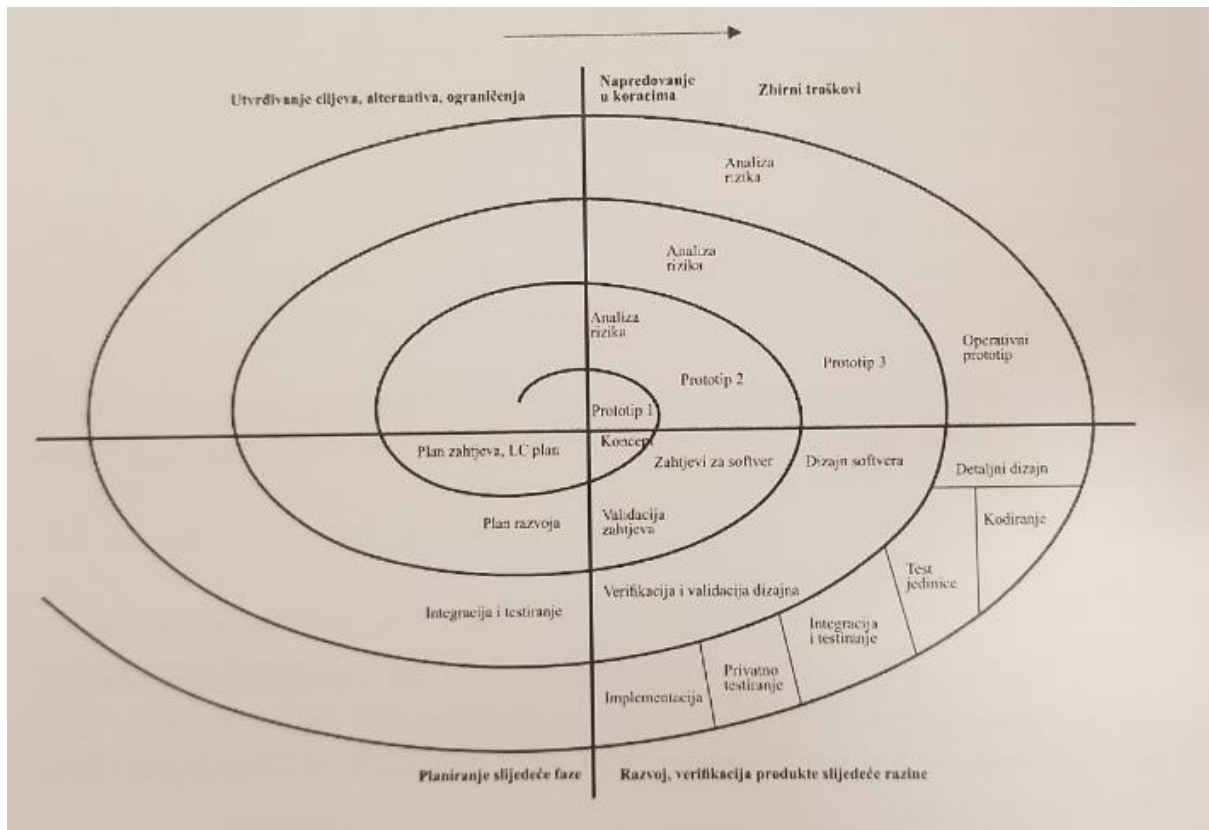
<sup>91</sup> Tutorials point: Waterfall model, dostupno na: - [https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm), pristupljeno 24.06.2020

<sup>92</sup> Slavek, N. (2016.): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

<sup>93</sup> Ibid.

<sup>94</sup> Ibid.

<sup>95</sup> B.W. Boehm, (1988): Asprial model of software development and enhancement, IEEE Comp



Slika 7. *Spiralni model, Ninoslav Slavek, str. 88*

U spiralnom modelu pozornost se pridaje rizicima projekta, problemima troškova i naglašava se korištenje prototipom.

Zrakasta ili radijalna dimenzija predstavlja zbirne troškove koji nastaju provedbom koraka na određeni datum. Angularna dimenzija predstavlja napredak u kompletiranju svakoga ciklusa spirale.

Svaki ciklus obuhvaća napredak istim redoslijedom koraka za svaki dio proizvoda i za svaku razinu elaboracije, od općega koncepta sustava, kodiranja do pojedinačnog programa.

Svaki ciklus spirale započinje identifikacijom:

- Ciljeva svakoga dijela proizvoda
- Alternativa implementacije tih dijelova proizvoda
- Ograničenja koja se nameću aplikacijom pojedine alternative

Sljedeći je korak vrjednovanje alternative u odnosu na ciljeve i ograničenja. Taj će proces identificirati područja nesigurnosti koja su značajan izvor rizika za projekt.

Nakon evaluacije rizika sljedeći su koraci kao u vodopadnome modelu (koncept, zahtjevi za programsku podršku, prethodni dizajn itd.). Svaka razina specifikacije programske podrške validira se i priprema plan za sljedeći ciklus.

Ako je korisnik u potpunosti zadovoljan tim operativnim prototipom, onda je on ujedno i krajnji proizvod. Ako korisnik još uvijek nije u potpunosti zadovoljan, tada operativni prototip postaje detaljna specifikacija zahtjeva za programsku podršku, a razvoj se nastavlja prema vodopadnom modelu, uključujući detaljan dizajn, kodiranje, testiranje i implementaciju.<sup>96</sup>

### 5.3. Testiranje programske podrške

Programeri nisu savršena bića i pogreške se događaju. Čak i ako program ne sadrži pogreške, možda će programeri koristiti tuđi kod koji sadrži pogreške. Štoviše, promjena koda u jednoj klasi ili sučelju može dovesti do toga da kod u drugoj klasi ili sučelju koji je radio prestane raditi. To se zove regresija. Pažljivo testiranje je važna faza svakog razvojnog procesa.<sup>97</sup>

Općenito, testiranje je analitički postupak za mjerenje i osiguranje kvalitete programske podrške i procesa razvoja. Kao analitička mjera za osiguranje kvalitete testiranje identifikacijom pogrešaka i njihovim otklanjanjem pomaže u povećanju kvalitete programske podrške.<sup>98</sup>

U procesu razvoja programske podrške, koristi se različite metode testiranja da se prouči i ocijeni rad pojedinih softvera ili aplikacije. Fokus će biti na 5 tipova metode testiranja.<sup>99</sup>

To su:

- Testiranje „crna kutija“
- Testiranje „bijela kutija“
- Testiranje „siva kutija“

---

<sup>96</sup> B.W. Boehm, (1988.): Aspiral model of software development and enhancement, IEEE Comp.

<sup>97</sup> Slavek, N. (2016.): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Studio HS internet, Osijek

<sup>98</sup> Ibid.

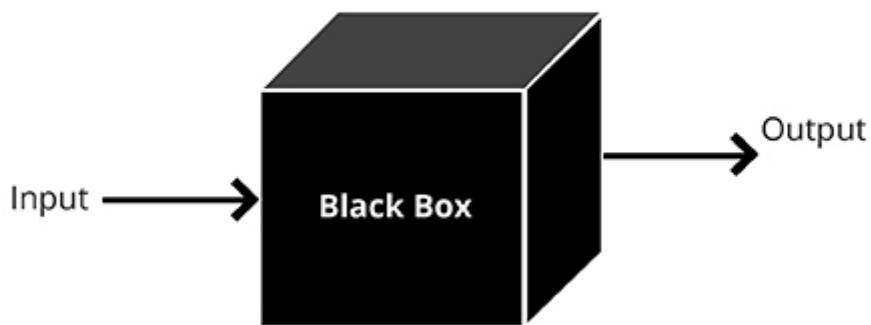
<sup>99</sup> The Automation Resources (2018): Software testing, dostupno na: <https://testautomationresources.com/software-testing-basics/software-testing-methods/>, pristupljeno 1.7.2020.

- Agilno testiranje
- Ad-hoc testiranje

### 5.3.1. Testiranje „crna kutija“

Testiranje „crna kutija“ (Black box testing) je metoda testiranja u kojem tester ne može vidjeti unutarnju implementaciju koda i programa. Ispitivača samo zanima radi li programska podrška ono što bi trebala raditi, ali ne i kako bi trebala raditi. Dok se vrši testiranje, tester je samo fokusiran na inpute i outpute bez ikakvih saznanja o internim procesima samoga sustava. Svrha ove metode testiranja je da se provjeri funkcionalnost samoga softvera i da se bude sigurno da softver radi sukladno zahtjevima i da zadovoljava korisnikove zahtjeve.<sup>100</sup>

## BLACK BOX TESTING APPROACH



Slika 8. Model testiranja „crna kutija“, slika preuzeta s <https://testautomationresources.com/software-testing-basics/software-testing-methods/> [1.7.2020.]

Prednosti ovakve metode testiranja je da znanje o programiranju i programskim jezicima nije potrebno, stoga svatko može provoditi ovu vrstu testiranja. Testni scenariji se mogu napisati prema funkcionalnoj specifikaciji prije početne implementacije. Metoda je veoma korisna kada se koristi na velikim sistemima.

---

<sup>100</sup> The Automation Resources (2018): Software testing, dostupno na: <https://testautomationresources.com/software-testing-basics/software-testing-methods/>, pristupljeno 1.7.2020.

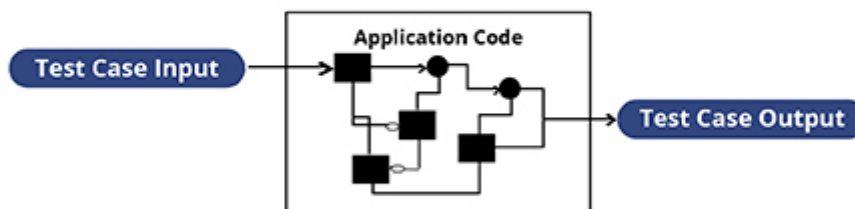


Nedostaci ove metode su da je ograničena pokrivenost testnim scenarijima. Ukoliko specifikacija sustava nije dovoljno jasna, teško je dizajnirati testne slučajeve. Mogućnost neučinkovitog testiranja jer tester ima ograničeno znanje o sustavu.<sup>101</sup>

### 5.3.2. Testiranje „bijela kutija“

Testiranje „bijela kutija“ (White box testing) ispituje unutarnju strukturu programa. Unutarnja struktura sustava kao i vještina programiranja se upotrebljava kako bi se dizajnirali testni slučajevi. Provodi se detaljna provjera strukture programskog koda i interne logike sustava. Za razliku od testiranja „crna kutija“, u ovoj metodi testiranja tester mora imati određeno znanje kako bi mogao provoditi takvu vrstu testiranja tj. mora znati kako je softver implementiran, kako radi itd. Tester odabere odgovarajuće inpute za pronalaženje putanje prolaza kroz kod te ocjenjuje odgovarajući output.<sup>102</sup>

## WHITE BOX TESTING APPROACH



Slika 9. Prikaz testiranja po modelu „bijela kutija“, preuzeto s <https://medium.com/@ocfranz/importance-of-white-box-testing-in-embedded-systems-dc475280bd3c>, [1.7.2020]

<sup>101</sup> The Automation Resources (2018): Software testing, dostupno na: <https://testautomationresources.com/software-testing-basics/software-testing-methods/>, pristupljeno 1.7.2020.

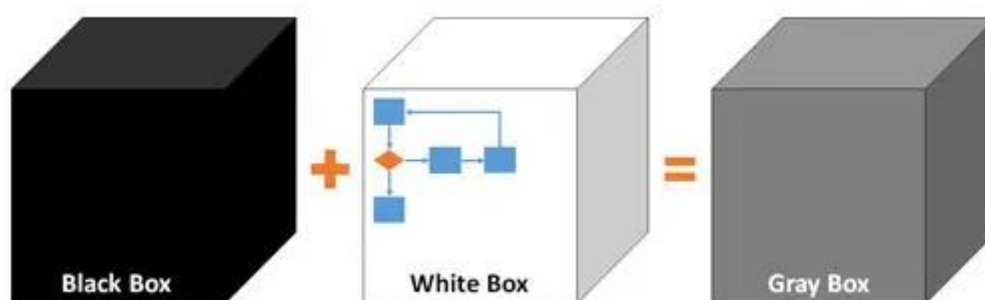
<sup>102</sup> Software Testing Fundamentals: White box testing, dostupno na: <http://softwaretestingfundamentals.com/white-box-testing/>, pristupljeno 1.7.2020

Prednosti ove metode testiranja su da tester koji ima napredno znanje o programiranju može lako odlučiti koja vrsta podataka može pomoći u testiranju sustava. Temeljitiije je testiranje i pomaže pri optimizaciji programskog koda.

Nedostaci ove metode su da je potreban tester koji ima znanje u kodiranje, što znači da tester više košta. Ponekad je nemoguće provjeriti svaki kutak koda i da li on funkcionira pa se zna često desiti da dosta koda bude ne testirano. Teško je za održavanje jer su potrebni specijalizirani alati.<sup>103</sup>

### 5.3.3. Testiranje „siva kutija“

Testiranje „siva kutija“ je kombinacija metode testiranje „bijela kutija“ i testiranja „crna kutija“. U ovoj metodi, tester ima parcijalno znanje o internoj strukturi. Ova vrste testiranja se vrši iz perspektive korisnika.<sup>104</sup>



Slika 10. Prikaz testiranja „siva kutija“, preuzeto s <http://softwaretestingfundamentals.com/gray-box-testing/> [1.7.2020.]

Prednosti ove vrste testiranja su da se uzima ono najbolje iz testiranja „crna kutija“ i testiranja „bijela kutija“, tester „siva kutija“ se oslanjaju na funkcionalnu specifikaciju i definiciju sučelja, tester je upoznat s podacima u sustavu pa sukladno tome je u mogućnosti dizajnirati dobre testne scenarije, testovi se rade iz perspektive korisnika, a ne dizajnera.

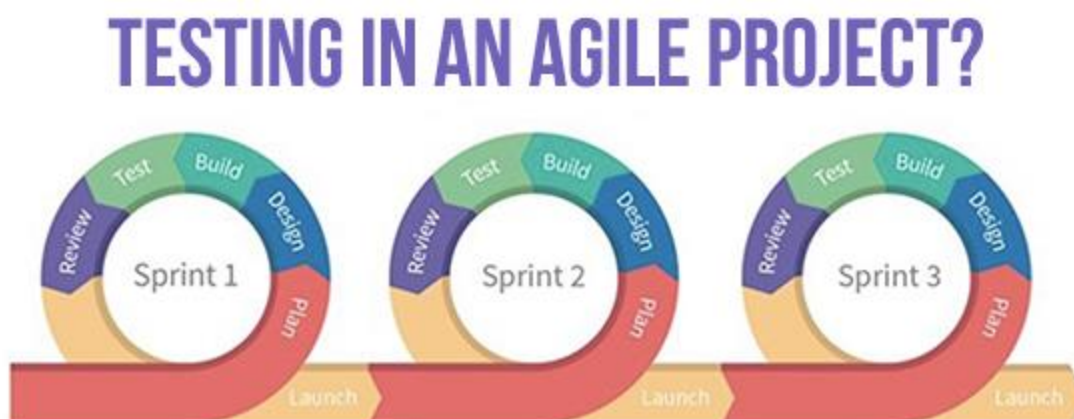
<sup>103</sup> Software Testing Fundamentals: White box testing, dostupno na: <http://softwaretestingfundamentals.com/white-box-testing/>, pristupljeno 1.7.2020

<sup>104</sup> Software testing fundamentals: Gray box testing, dostupno na: <http://softwaretestingfundamentals.com/gray-box-testing/>, pristupljeno 6.7.2020

Nedostaci ove vrste testiranja su da se isti testovi mogu ponavljati, mnogo testnih puteva će ostati ne testirano.<sup>105</sup>

#### 5.3.4. Agilno testiranje

Agilno testiranje znači testiranje softvera na greške ili neke druge probleme u što bržem mogućem roku kako bi se dobio što brža povratna informacija i kako bi se uklonili nedostaci.<sup>106</sup>



Slika 11. Prikaz agilnog testiranja, preuzeto s <http://softwaretestingfundamentals.com/agile-testing/> [1.7.2020.]

Prednosti agilnog testiranja su ušteda vremena i novaca, manje dokumentacije, povratne informacije od kupca, dnevnim sastancima je lakše utvrditi moguće pogreške.

Nedostaci agilnog testiranja je da sam proces testiranja traje dugo, kada se otkrije jedna greška u kodu, moguće je da se pojave neke druge koje su povezane s tom greškom, a teško ih je uočiti.<sup>107</sup>

---

<sup>105</sup> Tutorials Point: Software testing methods, dostupno na: [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_methods.htm](https://www.tutorialspoint.com/software_testing/software_testing_methods.htm), pristupljeno 6.7.2020.

<sup>106</sup> Ibid.

<sup>107</sup> Tutorials Point: Software testing methods, dostupno na: [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_methods.htm](https://www.tutorialspoint.com/software_testing/software_testing_methods.htm), pristupljeno 6.7.2020.

### 5.3.5. Ad hoc testiranje

Metoda testiranja u kojoj se testiranje vrši bez dokumentacije i planiranja naziva se Ad hoc testiranje. Testiranje se provodi proizvoljno i improvizacijski. Ovom vrstom testiranja je moguće pronaći greške koje se ne bi pronašle automatski testiranjem. Uspješnost ovakvog testiranja ovisi o kreativnosti i upornosti testera, a ponekad i o sreći.<sup>108</sup>

---

<sup>108</sup> Software testing: Ad hoc Testing, dostupno na: <http://softwaretestingfundamentals.com/ad-hoc-testing/>, pristupljeno 6.7.2020

## 6. PRIMJER PROJEKTA - STUDIJ SLUČAJA

### 6.1. Uvod

U ovome poglavlju bit će prikazano i opisano nastajanje projekta u sklopu IT poduzeća. Projekt je rađen u poduzeću KING ICT-u u sklopu „Winter Academy“ prakse. Cilj ovog projekta je bilo osmisliti internu poslovnu aplikaciju koja će omogućiti zaposlenicima KING-a da objavljuju oglase ako žele nešto prodati. U velikim poduzećima ima mnogo zaposlenih koji dolaze iz svih krajeva Hrvatske pa tako postoje različite kulture i običaje. Neki od zaposlenika imaju obiteljska privatna gospodarstva (OPG) te proizvode različite proizvode. Od jabuka, mandarina iz doline Neretve pa sve do slatkih medova ili vina s poluotoka Pelješca. Zaposlenici bi rado unaprijedili prodajne kanale, a jedan od mogućih prodajnih kanala je da ga plasiraju proizvod unutar svoje firme. Većina se komunikacije zaposlenih odvija putem elektroničke pošte, Skype business platforme, Slacka i drugih platformi. Vi, kao prodavač, ne biste htjeli miješati poslovnu elektroničku poštu s elektroničkom poštom koji imaju veze s vašim proizvodom, ako se prodaje. Cilj ovog diplomskog rada je pobliže objasniti razvoj interne poslovne aplikacije u IT poduzeću. Web aplikacija je namijenjena svim zaposlenicima unutar poduzeća preko koje bi prodavači mogli objavljivati svoje oglase s različitim proizvodima gdje bi svim zainteresiranim kupcima, unutar firme, bile dostupne sve informacije na jednome mjestu. Ovom web aplikacijom rješava se problem miješanja privatne elektroničke pošte i onih vezanih za proizvode. Zainteresiranom kupcu se pružaju informacije o samom proizvodu te je olakšana komunikacija putem same platforme. Naziv projekta je „Market Place“.

### 6.2. Primjer funkcionalne specifikacije

Poslovni analitičar je zadužen za izradu i pisanje funkcionalne specifikacije. Funkcionalna specifikacija je temelj svakoga projekta. Prilikom izrade poslovne aplikacije, poslovni analitičar je u komunikaciji s klijentom te zajedno s timom programera, osobom koja je zadužena za izradu baze podataka i testerom dogovaraju i izrađuju funkcionalnu specifikaciju. Funkcionalna specifikacija se mora poštovati od strane programera jer kada se ona napiše, predaje se klijentu na potpis. Nakon potpisa kreće se u izradu projekta. Projekt ne smije značajno odstupati od specifikacije. Nije poželjno da programeri uzmu stvari u svoje ruke i ignoriraju specifikaciju jer na kraju poslovni analitičar odgovara za cijeli projekt. Na kraju, ako je klijent potpisao specifikaciju, očekuje se da će mu se takvo što i isporučiti.

U nastavku će biti prikazane slike kako je izgledala funkcionalna specifikacija s projekta:

## Market Place

Funkcionalna specifikacija

Verzija 1.0

### VERIFIKACIJA I ODOBRENJE

	Ime i prezime	Potpis	Datum
Za Isporučitelja:			
Za Naručitelja:			

Slika 12. Prikaz naslove strane funkcionalne specifikacije (Izrada autora)

## 1. NAMJENA I OPSEG

Namjena dokumenta je opisati procese koji su potrebni za izradu interne aplikacije „Market Place“. „Market Place“ je web aplikacija kojom bi se mogli koristiti zaposlenici M SAN grupacije. Glavni problem koji će ova aplikacija riješiti je smanjene nepotrebne komunikacije putem maila između zaposlenika unutar grupacije o prodaji/kupnji određenih proizvoda. Umjesto dosadašnjeg oglašavanja putem maila, ova aplikacija omogućuje prodavačima oglašavanje svojih proizvoda. Kupci u svakom trenutku mogu pristupiti željenoj aplikaciji i vidjeti koje proizvode nude kolege unutar grupacije.

U dokumentu su navedene sve funkcionalnosti unutar kojih se nalaze glavni i alternativni scenariji te scenariji iznimki. Procesi pristupa aplikaciji, objave oglasa i pretraživanja oglasa se opisuju kroz podatkovne skupove, poslovna pravila, poruke i ekrane aplikacije.

*Slika 13. Prikaz namjene i opsega (Izrada autora)*

## 2. AKTERI

Akteri ove funkcionalnosti su korisnici sustava tj. zaposlenici KING ICT-a.

### 3. F001 – PRISTUPI APLIKACIJI I AŽURIRAJ PROFIL

#### 3.1 OPIS

Korisnik otvara aplikaciju te se prijavljuje u sustav koristeći domenski račun (@king-ict.hr) i lozinke koja mu je dodijeljena. Korisniku će se nuditi mogućnost ažuriranja ažuriranja svog profil i promjene podatke. S obzirom da se korisnik prijavljuje s domenskim korisničkim imenom; ime, prezime i e-mail će biti automatski popunjeni. Ostali podaci koje je potrebno popuniti su broj telefona, IBAN, naziv zgrade, kat, tvrtku te mjesto sjedenja u firmi.

#### 3.2 STANJE SUSTAVA PRIJE IZVOĐENJA

Aplikacija je u funkciji.

Korisnik se nalazi na početnoj stranici za prijavu.

#### 3.3 STANJE SUSTAVA NAKON IZVOĐENJA

Korisnik je uspješno prijavljen u sustav.

Korisnik je uspješno ažurirao profil.

#### 3.4 SC001 – PRIJAVI SE

1. Korisnik unosi svoj e-mail i lozinku te odabire opciju „Prihvati“
2. Sustav provjerava ispravnost unesenih podataka ([PSK001](#), [PPO01](#))
3. Sustav prijavljuje korisnika na aplikaciju i otvara početnu stranicu

Alternativni scenariji:

**A1: Korisnik je u prethodnoj prijavi odabrao opciju za pamćenje njegove prijave u aplikaciju.** Sustav popunjava polja za prijavu korisnikovim podacima. Korisnik odabire opciju za prijavu u aplikaciju. Scenarij se nastavlja u 3. koraku.

Scenariji iznimki:

**1a: Korisnik je unio pogrešnu e-mail adresu.** Sustav prikazuje korisniku poruku ([P001](#)).

**1b: Korisnik je unio pogrešnu lozinku.** Sustav prikazuje korisniku poruku ([P001](#)).

**1c: Korisnik nije unio lozinku.** Sustav prikazuje korisniku poruku ([P001](#)).

**1d: Ne postoji korisnik s unesenim e-mailom.** Sustav prikazuje korisniku poruku ([P002](#)).

Slika 14. Prikaz funkcionalnosti i scenarija (F001 – Pristupi aplikaciji i ažuriraj profil) (Izrada autora)

Opis funkcionalnosti govori što sustav mora napraviti kako bi korisnik izvršio željenu radnju. Ovdje je primjer kako će se korisnik ulogirati u sustav. Korisnik se prijavljuje u sustav svojom elektroničkom poštom i lozinkom koje je dobio od poduzeća. U specifikaciji mora biti



navedeno stanje sustava prije izvođenja i stanje sustava nakon izvođenja. Stanje sustava nakon izvođenja nudi scenarije što se treba dogoditi ako korisnik odabere određenu funkciju. U ovom primjeru jasno se vidi da se korisniku nude dvije opcije prilikom prijave u sustav. Prva opcija je da se prijavi u sustav, a druga opcija je da ažurira svoj profil.

„SC001 – Prijavi se“ opisuje postupak kako se prijaviti u sustav. Alternativni scenariji je scenariji koji je moguć da se dogodi prilikom nekog scenarija, ali nije primaran scenariji. Ovdje je primjer da kada se korisnik prijavljuje u sustav da se ponudi opcija da aplikacija zapamti njegovo korisničko ime i lozinku kako se ne bi morala svakog puta ponovo upisivati. Scenariji iznimki govori koje su mogući scenariji da se korisnik neuspješno prijavi u sustav.

### 3.5 SC002 – UNESI ILI AŽURIRAJ KORISNIČKE PODATKE

1. Korisnik odabire mogućnosti pregleda ili ažuriranja profila
2. Sustav otvara novi ekran gdje se unose podaci
3. Korisnik unosi podatke i odabire opciju za spremanje
4. Sustav provjerava jesu li uneseni svi obvezni podaci ([PSK002](#))
5. Korisnik odabire opciju za prihvaćanje
6. Sustav sprema unesene podatke ([P005](#))

Alternativni scenarij:

**A1:** Korisnik odabire opciju povratka na početnu stranicu odabirom loga. Sustav vraća korisnika na početnu stranicu ([EK004](#)). Scenarij završava.

Scenarij iznimke:

**3a:** Korisnik nije unio sve obvezne podatke. Sustav prikazuje korisniku poruku ([P003](#)). Scenarij se nastavlja u 3. koraku.

**3a:** Korisnik odustaje od ažuriranja svog profila. Sustav prikazuje korisniku poruku ([P004](#)). Ukoliko korisnik potvrdi odustajanje, scenarij završava.

### 3.6 POSLOVNA PRAVILA

Oznaka	Opis poslovnog pravila
PP001	Korisnik mora imati aktivan domenski račun

Slika 15. Prikaz scenarija (Izrada autora)

Ovdje se vidljiv drugi scenarij i postupak koji je potreban da se on dogodi.

Poslovna pravila su uvjeti koji moraju biti zadovoljeni kako bi sustav funkcionirao. Ovdje je primjer toga da korisnik mora imati aktivan domenski račun jer ako nema, ostali scenariji nisu u mogućnosti da se dogode.

#### 4.9 POSLOVNA PRAVILA

Oznaka	Opis poslovnog pravila
PP002	Prilikom otvaranja početne stranice aplikacije, oglasi moraju biti poredani prema datumu objave tj. najnoviji oglasi se prikazuju prije starijih oglasa
PP003	Duljina trajanja oglasa iznosi 30 dana. Sustav automatski dodaje 30 dana na datum objave te izračunava do kad će oglas biti aktivan te će biti prikazano kod oglasa. Datum se računa po formuli „Datum predaje oglasa + 30 dana- Datum isteka oglasa“
PP004	Prilikom odabira određene kategorije, ta kategorija mora biti povezana sa svim svojim proizvodima unutar kategorije.
PP005	Prilikom pretraživanja po ključnim riječima, korisnik mora ispravno unijeti riječ koju želi pretražiti. Ukoliko to ne učini ispravno, sustav mu neće prikazati željene rezultate.
PP006	Prilikom pretraživanja po ključnim riječima, pretraživanje je moguće samo po nazivu oglasa i po proizvodu.
PP007	Prilikom predaje novog oglasa, korisnik može maksimalno postaviti četiri slike.
PP008	Od datuma objave, oglas je aktivan 30 dana te se oglas se automatski premješta u „istekli oglasi“
PP009	Prilikom komentiranja oglasa, datum objave oglasa mora biti vidljiv

Slika 16. Primjer poslovnih pravila (Izrada autora)

#### 6.2 PSK002 – AŽURIRAJ PROFIL

Naziv podatka	Tip podatka	Obavezan (DA/NE)	Opis/napomena
Ime	Tekstualni	DA	Ime korisnika. Podaci se povlače iz sustava i nije ih moguće ažurirati
Prezime	Tekstualni	DA	Prezime korisnika. Podaci se povlače iz sustava i nije ih moguće ažurirati
E-mail	Alfanumerički	DA	Domenska e-mail adresa. Podaci se povlače iz sustava i nije ih moguće ažurirati
Broj telefona	Alfanumerički, maksimalno znakova 12	DA	Broj osobnog kontakta, službeni ili privatni broj mobitela. Zapisati u obliku: xxx/xxx-xxxx
IBAN	Alfanumerički, maksimalno 21 znak	NE	Broj računa ukoliko netko želi izvršiti uplatu online

Market Place

12

Slika 17. Primjer podatkovnih skupova (Izrada autora)

Podatkovni skupovi su skupovi koji se klasificira podatak. Vidi se da je „E-mail“ naziv podataka te tip podataka je alfanumerički. Sustav dozvoljava unos svih znakova. U nazivu podataka „Broj telefona“ je tip podataka alfanumerički s maksimalno 12 znakova. Sustav neće dozvoliti upis više od 12 znakova.

## 7. PORUKE U APLIKACIJI

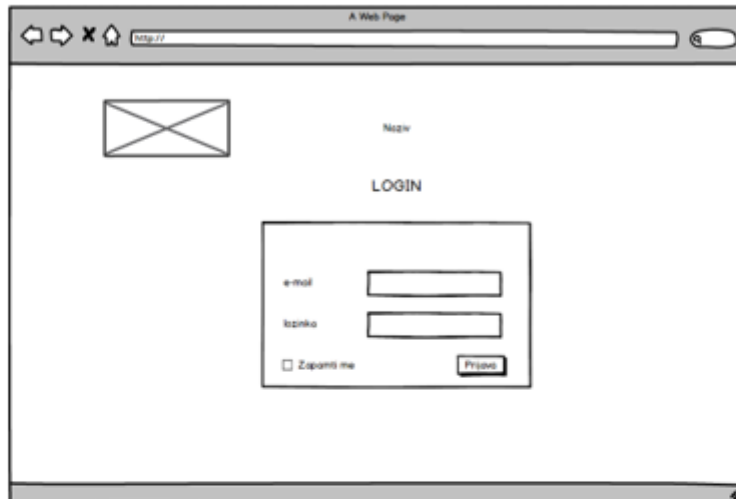
Oznaka	Opis poslovnog pravila
P001	Korisničko ime ili lozinka nisu ispravni.
P002	Neispravna e-mail adresa! Molimo kontaktirati IT podršku.
P003	Obavezan unos!
P004	Jeste li sigurni da želite odustati od ažuriranja profila?
P005	Podaci su spremjeni!
P006	Oglas je uspješno objavljen!
P007	Vaš oglas je spremljen i vidljiv u spremjenim oglasima.
P008	Jeste li sigurni da želite odustati od objave oglasa?
P009	Trenutno nema oglasa u odabranoj kategoriji ili je ključna riječ neispravno unesena.
P010	Došlo je do greške, pokušajte ponovo.
P011	Jeste li sigurni da želite obrisati ovaj oglas?
P012	Oglas je uspješno obrisani!

*Slika 18. Prikaz poruka u aplikaciji (Izrada autora)*

Sustav u nekim scenarijima može vratiti neku poruku. Sve poruke je potrebno napisati i navesti gdje bi se one mogle pojaviti.

## 8. EKRANI

### 8.1 EK001 LOGIN



A Web Page

http://

Naziv

LOGIN

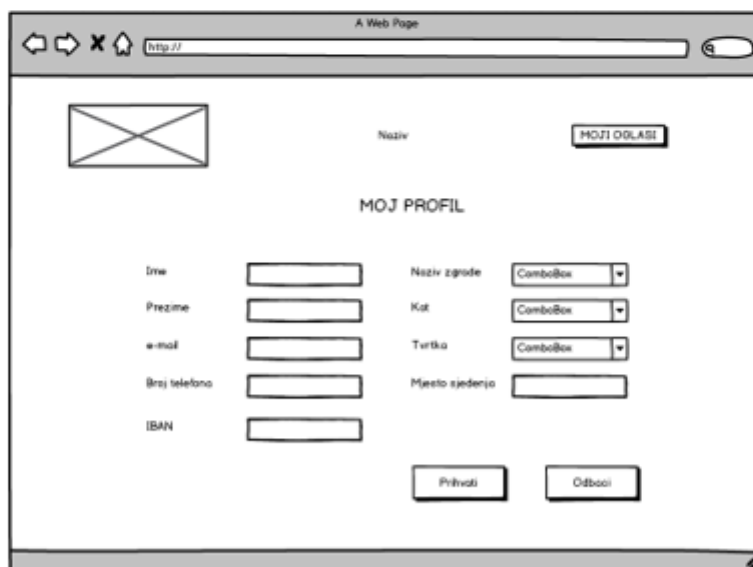
e-mail

lozinka

Zapamti me

Prijava

### 8.2 EK002 MOJ PROFIL



A Web Page

http://

Naziv

MOJI ODLASCI

MOJ PROFIL

Ime

Prezime

e-mail

Broj telefona

IBAN

Naziv zgrade

Kat

Turka

Mesto sjedenja

Privatni

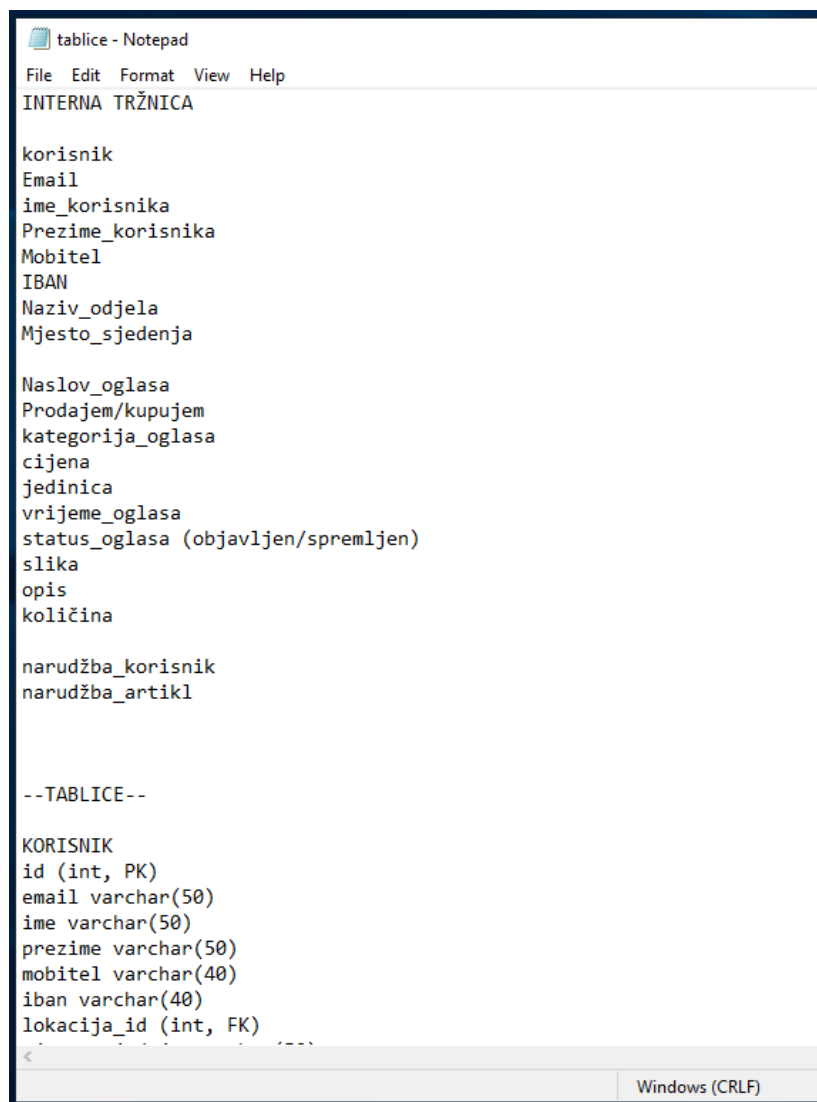
Odbaci

Slika 19. Prikaz skice ekrana (Izrada autora)

Prilikom izrade specifikacije, potrebno je izraditi skicu ekrana kako bi se vidjelo gdje bi se što nalazilo kada bi aplikacija bila gotova. Slika 19. prikazuje prijavu u sustav i ažuriranje profila.

### 6.3. Primjer baze podataka

Nakon što poslovni analitičar predstavi timu zahtjeve koje korisnik očekuje, kreće se s predlaganjem ideja kako bi mogla aplikacija izgledati i koje bi funkcionalnosti mogla imati. Ključna stvar za izradu baze podataka je da se točno definiraju objekti i njegovi atributi u konceptualnom dijelu izrade baze podataka.



```
tablice - Notepad
File Edit Format View Help
INTERNA TRŽNICA

korisnik
Email
ime_korisnika
Prezime_korisnika
Mobitel
IBAN
Naziv_odjela
Mjesto_sjedenja

Naslov_oglasa
Prodajem/kupujem
kategorija_oglasa
cijena
jedinica
vrijeme_oglasa
status_oglasa (objavljen/spremljen)
slika
opis
količina

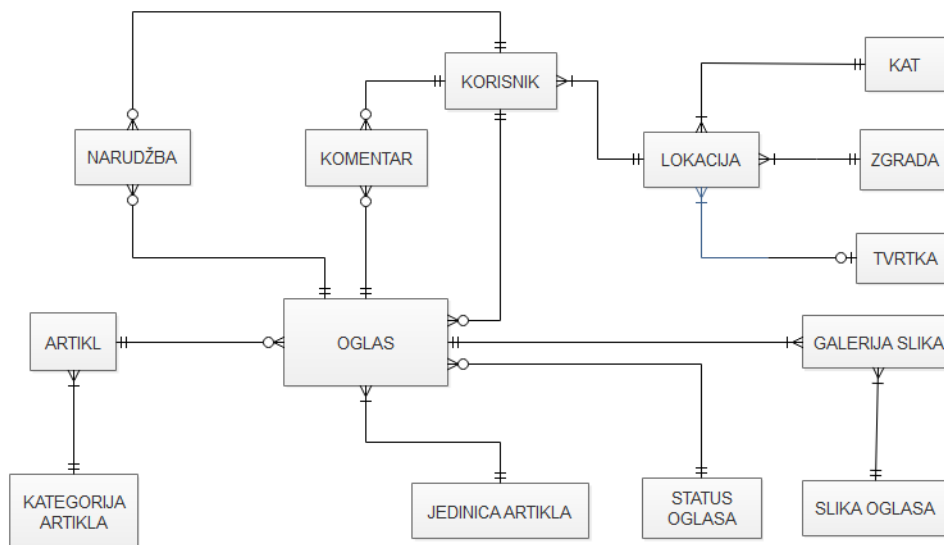
narudžba_korisnik
narudžba_artikl

--TABLICE--

KORISNIK
id (int, PK)
email varchar(50)
ime varchar(50)
prezime varchar(50)
mobitel varchar(40)
iban varchar(40)
lokacija_id (int, FK)
```

Slika 20. Prikaz konceptualnog modela baze podataka (Izrada autora)

Na slici 20. vidi se da je „korisnik“ objekt i njegove attribute: e-mail, ime\_korisnika, prezime\_korisnika, mobitel, IBAN itd.

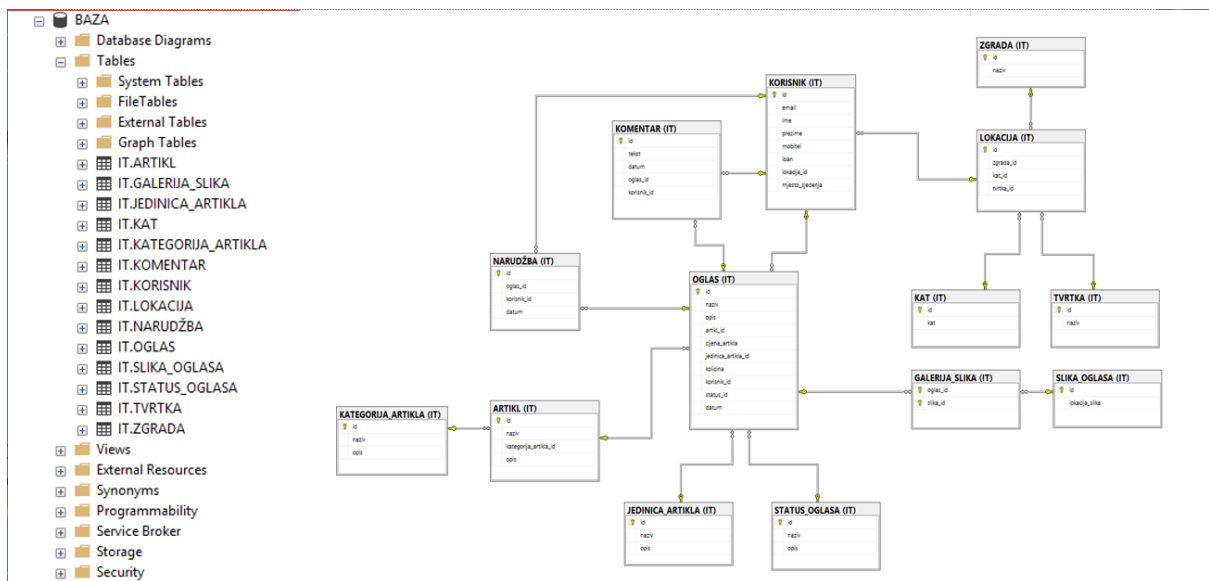


Slika 21. Prikaz ER Modela (Izrada autora)

Nakon konceptualne izrade baze podataka, definiranje objekata i atributa, određivanja tipova podataka, osoba koja izrađuje baze podataka mora napraviti ER Model tj. prikaz ER Modela baze podataka.

Interpretacija ER Modela na primjeru povezanosti „Oglas“ i „Korisnik“:

Korisnik može objaviti jedan, nijedan ili više oglasa, određeni oglas se može odnositi na jednog i samo jednog korisnika.



Slika 22. Prikaz ER Modela s objektima i atributima (Izrada autora)

U ovom, detaljnijem prikazu, ER Modela jasno se može vidjeti primarne ključeve. U tablici „Oglas“ polje „Artikl“ je ono koje jednoznačno definira redak te može biti primarni ključ jer ne postoje dvije ili više osobe s istim ID-om artikla.

```
CREATE TABLE [IT].[KORISNIK](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [email] [varchar](50) NOT NULL,
    [ime] [varchar](50) NOT NULL,
    [prezime] [varchar](50) NOT NULL,
    [mobitel] [varchar](15) NOT NULL,
    [iban] [varchar](25) NULL,
    [lokacija_id] [int] NOT NULL,
    [mjesto_sjedenja] [varchar](50) NULL,
    CONSTRAINT [PK_korisnik_id] PRIMARY KEY CLUSTERED
)
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Slika 23. Prikaz linija koda za izradu baze podataka kod objekta „KORISNIK“ (Izrada autora)



## 6.4. Primjer programskog inženjerstva

U ovome poglavlju će biti prikazane linije koda za ovaj projekt i prikaz stvarnog ekrana.

```
/**
 * Create custom private Route for authenticated part of app.
 *
 * @param {*} param0
 */
const PrivateRoute = ({ component: Component, ...rest }) => (
  <Route {...rest} render={(props) => (
    Auth.isAuthenticated === true
      ? <Component {...props} />
      : <Redirect to={{ pathname: "/login", state: { from: props.location } }} />
  )} />
)

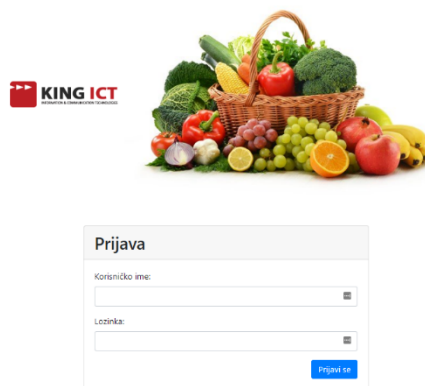
configureAxios();

class App extends Component {
  render() {
    return (
      <div className="App">
        <Switch>
          <Route path="/login" component={Login} />
          <PrivateRoute path="/" component={Marketplace} />
        </Switch>
      </div>
    );
  }
}

export default App;
```

Slika 24. Prikaz linija koda za autentifikaciju u aplikaciju (Izrada autora)

Ovdje su prikazane linije koda za autentifikaciju u aplikaciju. Prvi dio koda daje naredbu za autentifikaciju korisnika u sustav. Ako je korisnik uspješno autentificiran, sustav mu otvara početnu stranicu aplikacija, inače sustav preusmjerava na ponovnu autentifikaciju.



Prijava

Korisničko ime:

Lozinka:

[Prijavi se](#)

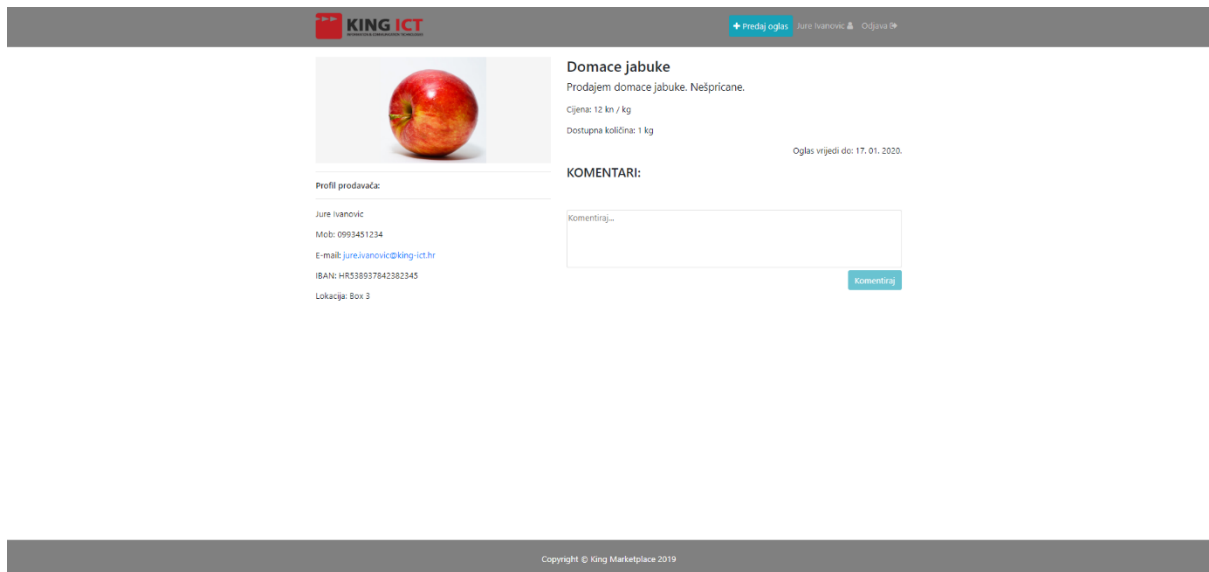
Slika 25. Prikaz prijave u aplikaciju (Izrada autora)

```
import React, { Component } from 'react';
import { Link } from 'react-router-dom';

class OglasKartica extends Component {
  render() {
    return (
      <div className="col-lg-4 col-md-6 mb-4">
        <div className="card h-100">
          <Link to={`/oglas/${this.props.oglas.idOglas}`}>
            <img className="card-img-top slikica" src={`data:image/jpg;base64,` + this.props.oglas.slika.slika} alt="" />
          </Link>
          <div className="card-body">
            <h6>
              <Link className="tekst-boja text-hover" to={`/oglas/${this.props.oglas.idOglas}`}>
                {this.props.oglas.naziv}
              </Link>
            </h6>
            <h5>{this.props.oglas.cijena} kn / {this.props.oglas.nazivJediniceArtikla}</h5>
          </div>
        </div>
      </div>
    );
  }
}

export default OglasKartica;
```

Slika 26. Prikaz linije koda za izgled kartice (Izrada autora)

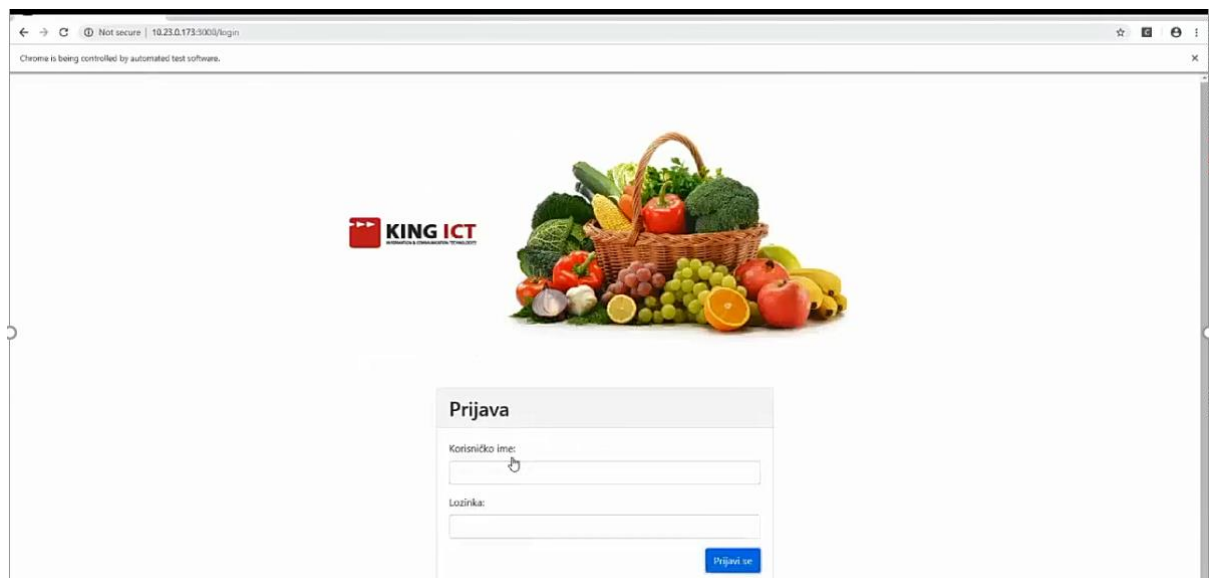


Slika 27. Prikaz izgleda kartice (Izrada autora)

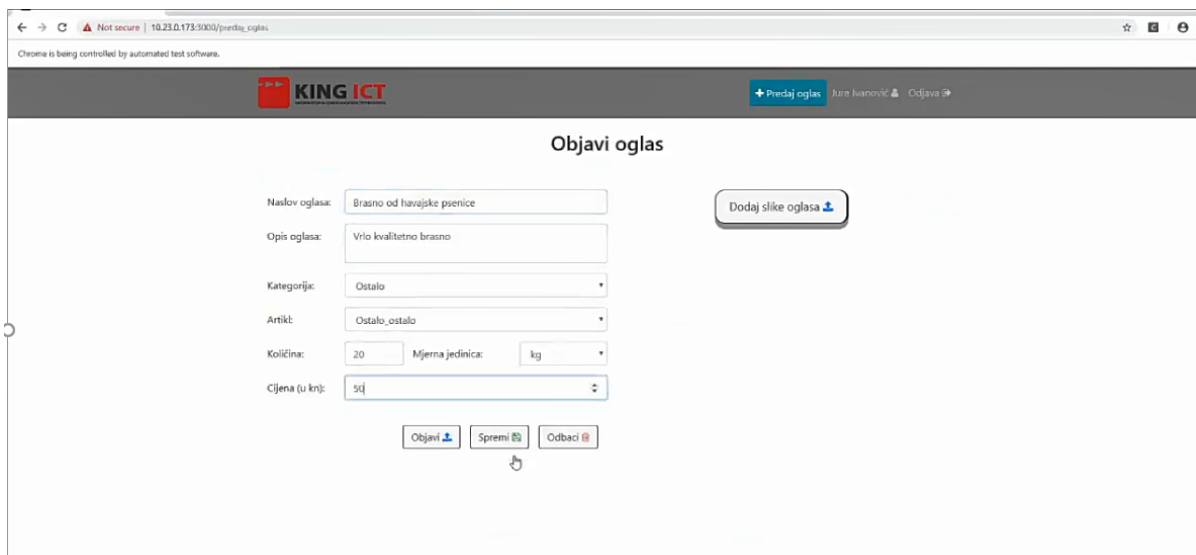
## 6.5. Primjer testiranja

Na ovome projektu testiranje se vršilo „ad hoc“ metodom te povremenim automatskim testiranjem.

Bit će prikazano u koracima kako se testirala funkcionalnost predaje oglasa u par slika.

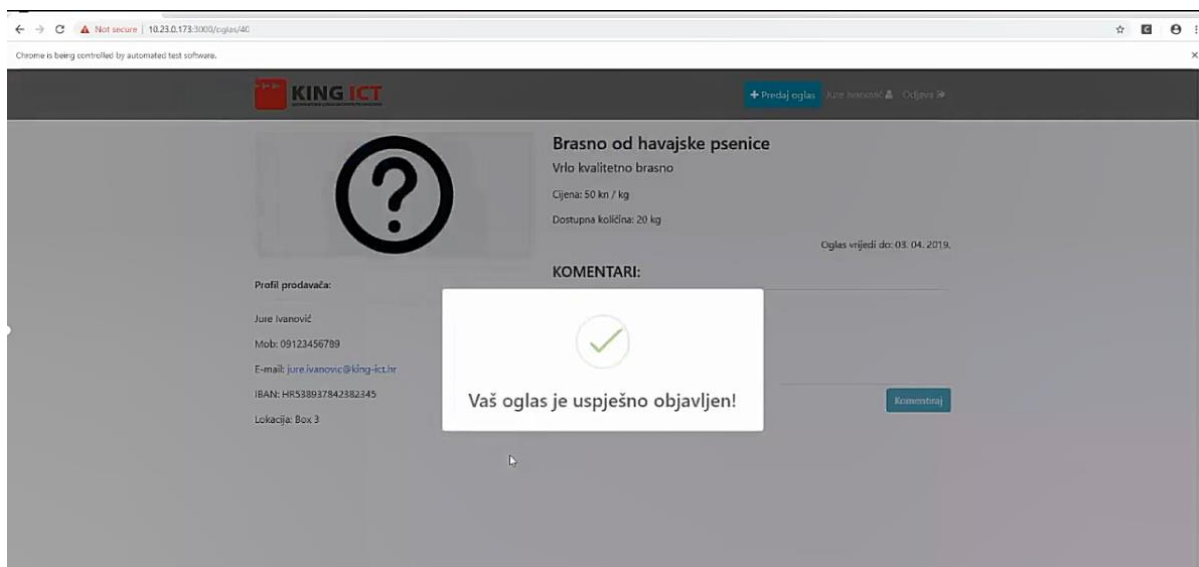


Slika 28. Prikaz testiranja – prijava u sustav (Izrada autora)



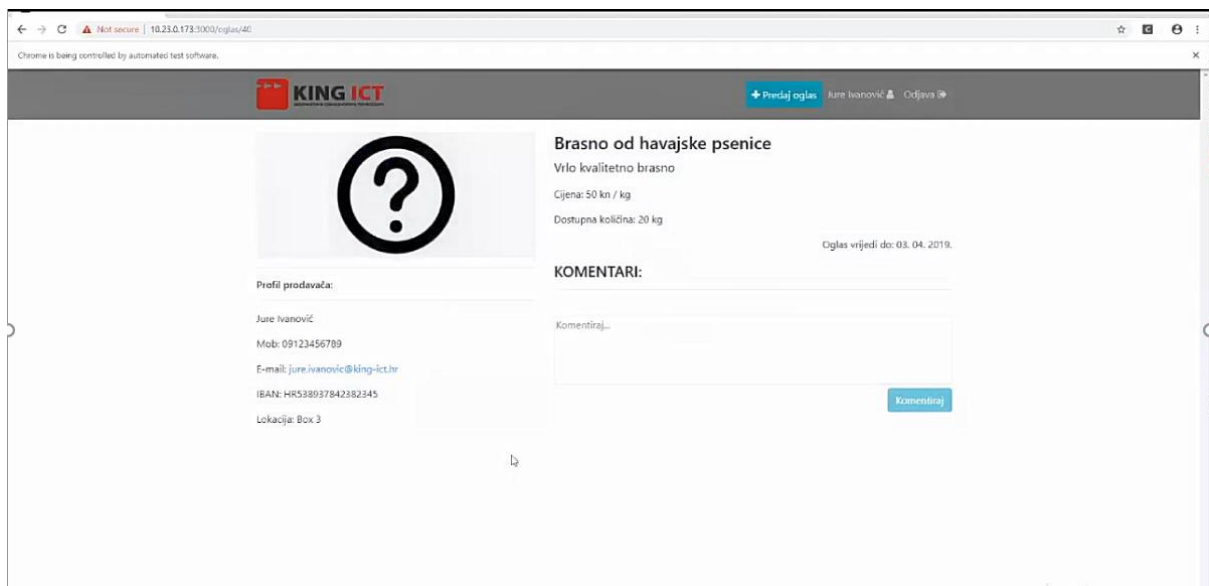
Slika 29. Prikaz popunjavanja sučelja (Izrada autora)

Nakon uspješne autentifikacije u sustav, sljedeći korak je bio da prilikom predaje oglasa, upišu se u sučelja potrebne informacije kako bi se oglas uspješno objavio. Nakon upisa svega u sučelja, pritišće se tipka „objavi“.



Slika 30. Prikaz uspješno objavljenog oglasa (Izrada autora)

Sustav vraća poruku da je oglas uspješno objavljen.



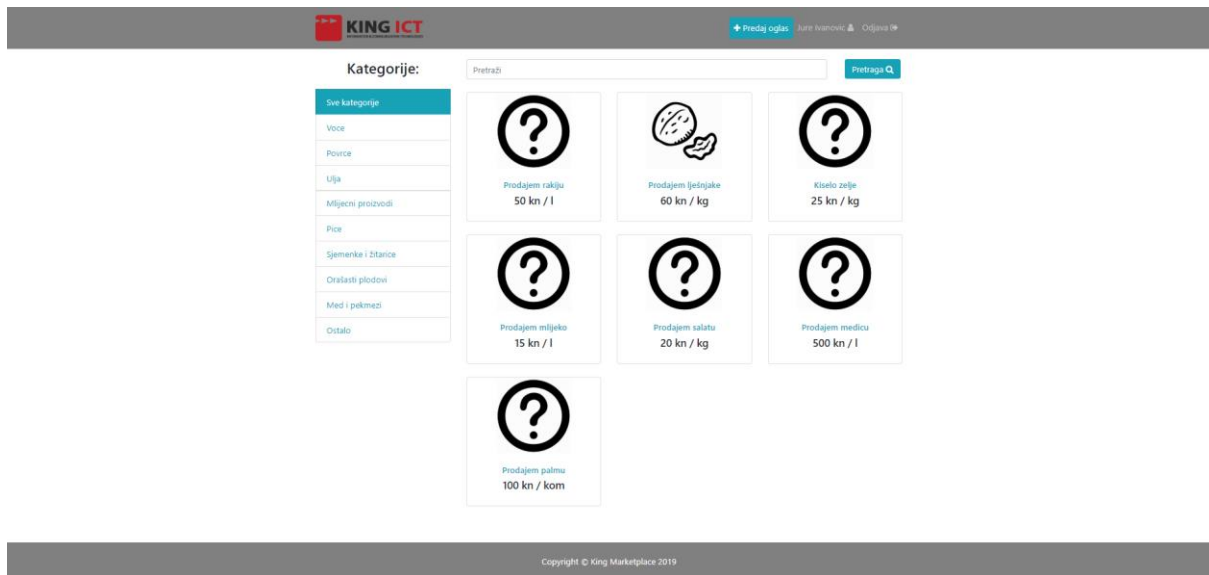
Slika 31. Prikaz objavljenog oglasa (Izrada autora)

Svrha ove funkcionalnosti je bila da se korisnik autentificira u sustav, upiše sve navedene informacije u sučelja te da preda oglas. Ovim testom se potvrđuje da je funkcionalnost ispravno napravljena.

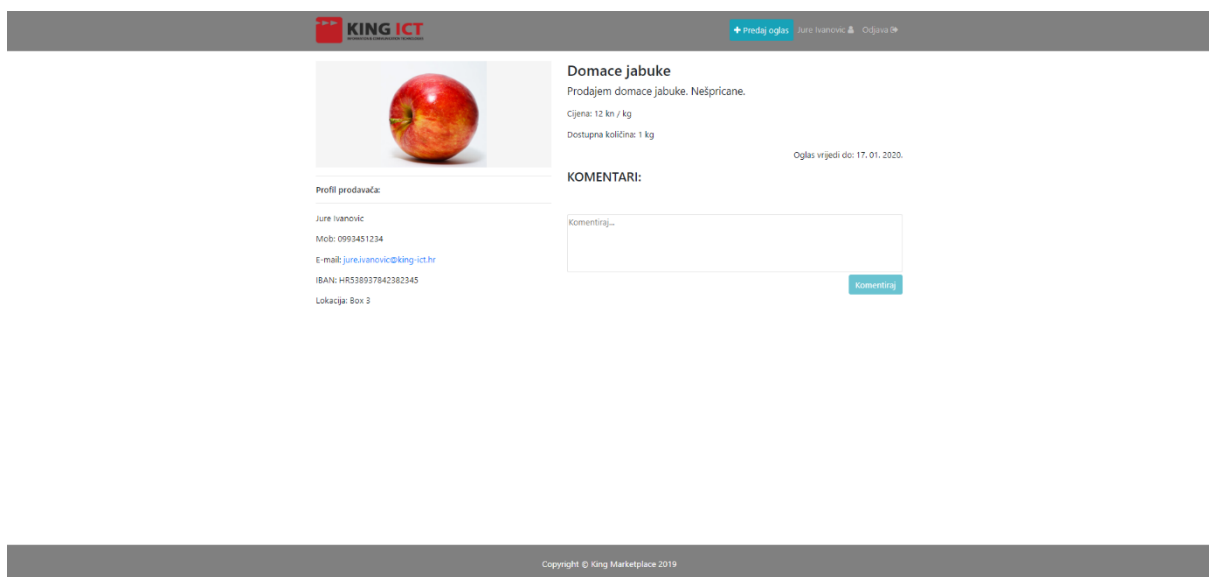
## 6.6. Prikaz aplikacije



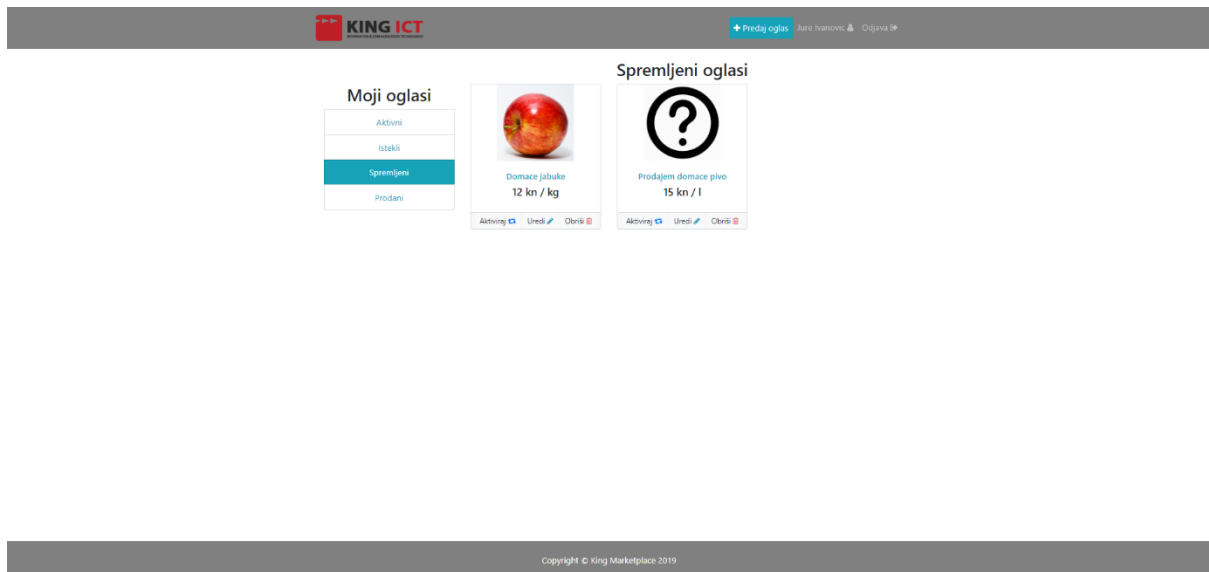
Slika 32. Prikaz stranice za autorizaciju u sustav (Izrada autora)



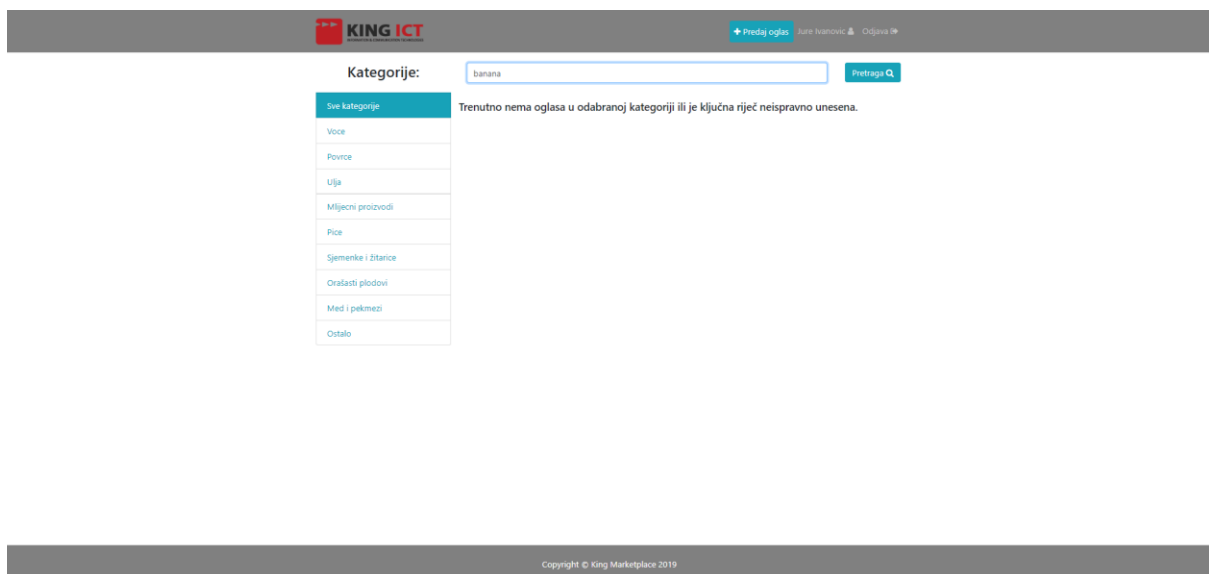
Slika 33. Prikaz početne stranice aplikacije (Izrada autora)



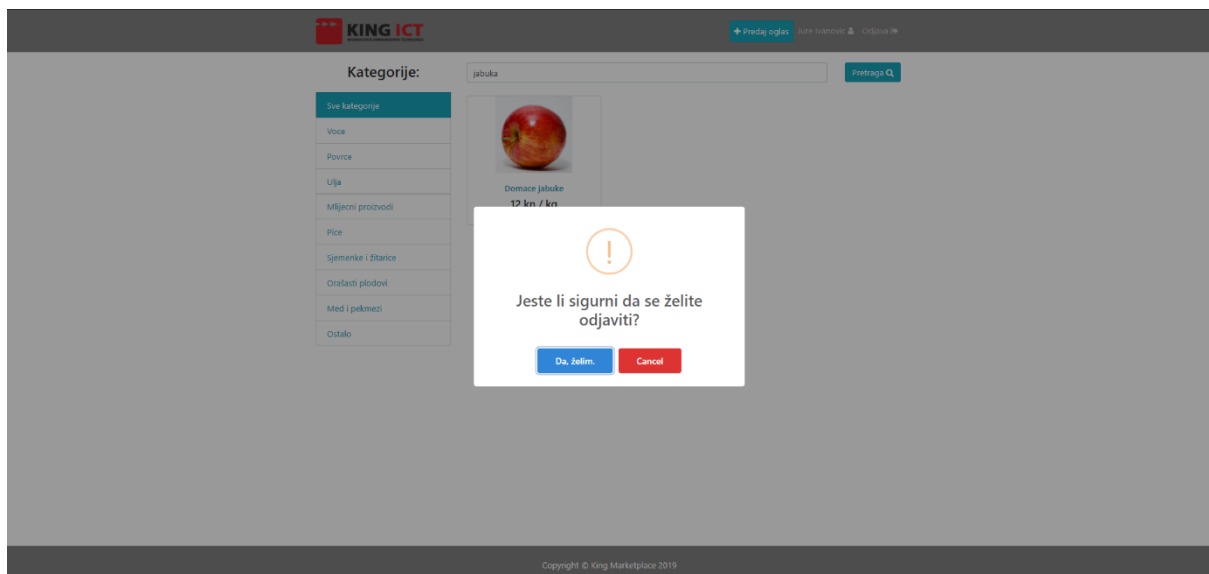
Slika 34. Prikaz oglasa (Izrada autora)



Slika 35.: Prikaz spremljenih oglasa (Izrada autora)



Slika 36. Prikaz pretraživanja po ključnim riječima (Izrada autora)



Slika 37. Prikaz odjave iz sustava (Izrada autora)



## 7. ZAKLJUČAK

U današnje moderno, informacijski napredno doba, aplikacije su sastavni dio poslovnoga života. Poslovni život ne bi bio isti bez aplikacija koje svakodnevno pomažu u poslovnom svijetu (npr. Excel). Razvoj poslovanja i napredak poduzeća bio bi značajnije usporen bez njih. Aplikacije se razvijaju s ciljem povećanja efikasnosti zaposlenika, poboljšanja i olakšavanja rada zaposlenih što prikazuje aplikacija koja je predmet ovoga rada.

Da bi aplikacija bila uspješno napravljena potrebna je analiza. Analiza je ključni dio svakog projekta. Ukoliko se napravi loša analiza, cijeli projekt izrade aplikacije može ispasti nezadovoljavajuć za korisnika. Loše vođen projekt i prethodna analiza mogu dovesti do lošeg kredibiliteta u poslovnom svijetu. Zato je važno da se definiraju zadaci, kako će oni biti obavljeni, tko će ih obavljati i u kojem vremenskom razdoblju, kako bi planiranje i realizacija projekta bili što efikasniji. Zbog loše analize postoji mogućnost da dođe do nesporazuma s klijentom te je u tom slučaju cijeli tim primoran ponovo razvijati aplikaciju. Takve situacije dovode do stresa, frustracije između zaposlenih i na kraju, financijskih gubitaka za poduzeće. Najbitnije pitanje koje se postavlja prije izrade aplikacije je: „Koji problem se rješava ovom aplikacijom?“. Kada se dobije odgovor na ovo pitanje, može se krenuti s daljnjim razvojem.

U daljnjem razvoju, ključne su tehničke vještine zaposlenih, koje podrazumijevaju znanje u izradi baze podataka i programiranje. Izrada aplikacije započinje kreiranjem baze podataka koja je od velikog značaja jer se iz nje povlače svi podaci, dok programiranje podrazumijeva izgled sučelja na bazi podataka. Ukoliko je baza loše dizajnirana i potrebne su promjene na njoj u fazi izrade aplikacije, takvo nešto može dovesti do velikih poteškoća u daljnjoj fazi projekta jer i najmanja promjena na bazi podataka može dovesti do velikih promjena u linijama koda tijekom programiranja.

Pisanje linija koda mora biti organizirano i smješteno u razrede kako bi se svi mogući dodaci ili promjene u liniji koda mogle što brže i jednostavnije zamijeniti ili dodati. Posljednja faza je testiranje aplikacije kako bi se otklonile neželjene greške i samim time bi se unaprijedio rad aplikacije. Uzevši u obzir sve faze razvoja aplikacije, može se zaključiti da je u procesu izrade aplikacije najbitnije planiranje i razrada plana. Također, svatko unutar tima ima odgovornost kako bi aplikacija bila funkcionalna. Ukoliko jedan dio faze zakaže, čitava aplikacija bi mogla biti neuspješna.

## LITERATURA

1. Abrus, L. (2003): Izrada weba: abeceda za webmastere: kompletan vodič Zagreb: Bug: SysPrint, Zagreb
2. Batarelo, A. Ž. (2008): Uvod u poslovnu analizu, TEB poslovno savjetovanje, Zagreb
3. Birolla, H. i dr. (1988): Osnove informatike, Informator, Zagreb
4. Blanuša, D. (1965): Viša matematika, Tehnička knjiga, Zagreb
5. Boehm, B.W. (1988): Asprial model of software development and enhancement, IEEE Comp
6. Braut, R. (1966): Ekonomska analiza poslovanja poduzeća, I.dio, Visoka privredna Škola u Zagrebu, Zagreb
7. Codd, E. F. (1970): A Relation Model of Data for Large Shared Data Banks. Communications of ACM, 13.
8. Date, C. J. (1990): An Introduction to Data Base System, vol. 1. Addison – Wesley
9. Juran Joseph M., (2010): Juran's Quality Handbook: The Complete Guide to Performance Excellence, VI-Edition, McGraww-Hill
10. Kalafatić Z., Pošćić A., Šegvić S., Šribar J. (2016): Python za znatiželjne : sasvim drukčiji pogled na programiranje, 1.izd., Element, Zagreb
11. Kim, K. A., Nofsinger, J. R., Mohr, D. J. (2009): Corporate Governance , Prentice Hall, New Jersey
12. Klaić, B. (1974): Veliki rječnik stranih riječi, izraza i kratica, Zora, Zagreb
13. Krajčević, F. (1975): Analiza poslovanja organizacije udruženog rada, Zagreb
14. Krleža, M. (1996): Opća enciklopedija Leksigrafskog zavoda, I. svezak, Jugoslavenski leksikografski zavod, Zagreb, str. 116.
15. Kučak, D. (2013): Izrada web-aplikacija: priručnik, Algebra učilište, Zagreb
16. Kupper, H.U.(1995): Controlling, Stuttgart,Schaffer/Poeschel
17. Langefors B., Sundgren B. (1975): Information Systems Architecture, Petrocelli/Charter
18. Maleković M. (2016): Uvod u baze podataka, Varaždin: Fakultet organizacije i informatike: Sveučilište u Zagrebu, Mini-Print-Logo, Varaždin
19. Maleković M. (2017): Teorija i primjena baza podataka, Fakultet organizacije i informatike, Varaždin

20. Maleković M., Rabuzin K. (2016): Uvod u baze podataka; Varaždin: Fakultet organizacije i informatike; Zagreb: Sveučilište u Zagrebu, Varaždin : Mini-Print-Logo
21. Manger, R. (2010): Osnove projektiranja baza podataka, dostupno na: [https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310_polaznik.pdf) [1.7.2020.]
22. Martin J. (1977): Computer Data-base Organization, 2<sup>nd</sup> ed. Prentice-Hall
23. Paul, D., Yeates, D., Cadle, J., (2010): Business analysis, Swindon, BISL
24. Popović, Ž. (1979): Ekonomska analiza poslovanja, Zagreb
25. Royce, W.W. (1970): Managing the development of large software systems, dostupno na: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf> [1.7.2020.]
26. Schumpeter J.A. (1975): Povijest ekonomske analize, Informator, Zagreb
27. Slavek N. (2016): Osiguranje kvalitete programske podrške, Sveučilište Josipa Jurja Strossmayera, Elektrotehnički fakultete, Osijek: Studio HS internet, Osijek
28. Srića, V. (1988): Uvod u sistemski inženjering. Informator, Zagreb
29. Teorey, T. J., Yang, D., Fry, J. P. (1986.): A Logical Desing Methodology for Relational Databases Using the Extended Entity – relationship model. ACM Computing Surveys, vol.18, no. 2.
30. Tintor, J. (1983): Uvod u ekonomsku analizu poslovanja OUR-a, Zagreb
31. Tintor, J. (2009): Poslovna analiza : operativno poslovanje, poslovna načela, financiranje, kreditna sposobnost, proizvodnja, snaga zarađivanja, ljudski potencijali, Masmedia : Poslovni dnevnik, Zagreb.
32. Tipurić, D. (2007): Nadzorni odbor i korporativno upravljanje, Sinergija nakladništvo d.o.o., Zagreb
33. Tkalac, S.(1993): Relacijski model podataka, Društvo za razvoj informacijske pismenosti, Zagreb
34. Varga M. (1994): Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb.
35. Yakov F. (2015): Programiranje Java; Prijevod 2. izd.; Stega tisak, Zagreb

## POPIS SLIKA I TABLICA

### Popis slika

Slika 1. Shema povratne veze planiranja, kontrole i analize .....	7
Slika 2. Od stvarnog svijeta do baze podataka .....	16
Slika 3. Modeliranje aplikacijske domene.....	18
Slika 4. Klasifikacija i generalizacija.....	19
Slika 5. Ilustracija na prikaz razlike između frontenda i backenda .....	26
Slika 6. Vodopadni model .....	30
Slika 7. Spiralni model.....	32
Slika 8. Model testiranja „crna kutija“ .....	34
Slika 9. Prikaz testiranja po modelu „bijela kutija“ .....	35
Slika 10. Prikaz testiranja „siva kutija“ .....	36
Slika 11. Prikaz agilnog testiranja.....	37
Slika 12. Prikaz naslove strane funkcionalne specifikacije.....	40
Slika 13. Prikaz namjene i opsega .....	41
Slika 14. Prikaz funkcionalnosti i scenarija.....	42
Slika 15. Prikaz scenarija .....	44
Slika 16. Primjer poslovnih pravila.....	45
Slika 17. Primjer podatkovnih skupova.....	45
Slika 18. Prikaz poruka u aplikaciji .....	46
Slika 19. Prikaz skice ekrana.....	47
Slika 20. Prikaz konceptualnog modela baze podataka.....	48
Slika 21. Prikaz ER Modela .....	49
Slika 22. Prikaz ER Modela s objektima i atributima.....	50
Slika 23. Prikaz linija koda za izradu baze podataka kod objekta „KORISNIK“ .....	50
Slika 24. Prikaz linija koda za autentifikaciju u aplikaciju .....	51
Slika 25. Prikaz prijave u aplikaciju .....	52
Slika 26. Prikaz linije koda za izgled kartice .....	52
Slika 27. Prikaz izgleda kartice .....	53
Slika 28. Prikaz testiranja – prijava u sustav .....	53
Slika 29. Prikaz popunjavanja sučelja.....	54
Slika 30. Prikaz uspješno objavljenog oglasa .....	54

Slika 31. Prikaz objavljenog oglasa.....	55
Slika 32. Prikaz stranice za autorizaciju u sustav .....	55
Slika 33. Prikaz početne stranice aplikacije .....	56
Slika 34. Prikaz oglasa .....	56
Slika 35.: Prikaz spremljenih oglasa .....	57
Slika 36. Prikaz pretraživanja po ključnim riječima.....	57
Slika 37. Prikaz odjave iz sustava .....	58