

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Рачунарска техника и софтверско инжењерство

Ниво студија: Основне академске студије

Предмет: Пројектовање информационих система и база података

Број индекса: 578/2015

Стефан Д. Николић

Развој информационог система за управљање документацијом

Дипломски рад

Комисија за преглед и одбрану:

1. **Др Милан Ерић – ментор**

2. _____

3. _____

Датум одбране: _____

Оцена: _____



Универзитет у Крагујевцу
Факултет инжењерских наука



Основне студије: Рачунарска техника и софтверско инжењерство

Име и презиме: Стефан Николић

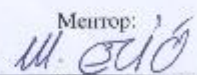
Број индекса: 578/2015

ПРИЈАВА ЗАВРШНОГ (ДИПЛОМСКОГ) РАДА

Тема рада: *Развој информационог система за управљање документацијом*

Задатак: Задатак дипломског рада је пројектовање и имплементација информационог система за управљање документацијом чија ће улога бити да омогући кориснику да креира документе, управља њима, прати их, модификује и складишти. Кроз дипломски рад приказати комплетан поступак пројектовања информационог система са објашњењем кључних појмова. Кроз основну улогу, циљеве, задатке и функцију објаснити систем за управљање документацијом. Закључним разматрањима описати интеграција система и могућности имплементација.

Крагујевац, 16.04.2019

Ментор: 
Проф. др Милош Д. Јерић

Изјава о самосталној изради рада

Изјављујем да сам овај дипломски рад урадио самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

Стефан Николић, 578/2015

Садржај

1. Увод.....	6
2. Опис задатка дипломског рада.....	8
3. Поступак развоја информационог система.....	9
3.1. Функционално моделирање.....	10
3.1.1. Функционална декомпозиција	11
3.1.2. Дефинисање захтева корисника.....	14
3.1.3. Технички предуслови.....	16
3.2. Информационо моделирање.....	18
3.2.1. Дефинисање детаљних захтева	19
3.2.2. Креирање ЕР дијаграма.....	29
3.2.3. Креирање атрибута.....	32
3.2.4. Дефинисање пословних правила	34
3.3. Апликативно моделирање	40
3.3.1. Дефинисање физичког дизајна	40
3.3.2. Генерисање шеме базе података.....	44
3.3.3. Израда апликације.....	47
3.4. Имплементација.....	51
3.4.1. Увођење	51
3.4.2. Тестирање	53
3.4.3. Одржавање.....	54
4. Закључак	56
5. Литература	57

Abstract

Development procedure of document management information system is described in this thesis. Basic terms related to the information system are defined at the beginning, and afterwards all stages of information system development are described, which are: functional modeling, information modeling, application modeling and implementation. Functional decomposition, user requirements defining and technical prerequisites are described within functional modeling. Detailed requirements defining, ER diagrams and attributes creating and business rules defining are performed within information modeling. Physical design defining, database schema generating and application production are done within application modeling. Initiation, testing and maintenance are performed within implementation.

Key words: Data, information, information technologies, information system

Резиме

У овом раду је описан поступак развоја информационог система за управљање документацијом. На почетку су дефинисани основни појмови везани за информациони систем, а након тога су описане све фазе развоја информационог система: функционално моделирање, информационо моделирање, апликативно моделирање и имплементација. У оквиру функционалног моделирања је описана функционална декомпозиција, дефинисани су захтеви корисника и технички предуслови. У оквиру информационог моделирања је извршено дефинисање детаљних захтева, креирање ЕР дијаграма и атрибута и дефинисање пословних правила. У оквиру апликативног моделирања је урађено дефинисање физичког дизајна, генерисање шеме базе података и израда апликације. У оквиру имплементације је извршено увођење, тестирање и одржавање.

Кључне речи: Податак, информација, информационе технологије, информациони систем

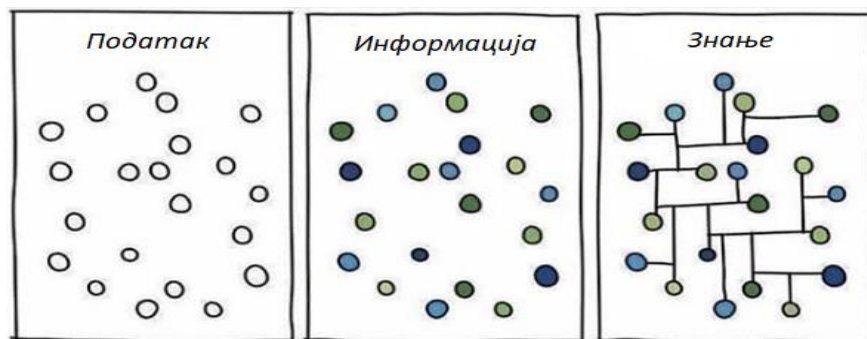
1. Увод

Кроз историју, највећи помаци на пољу технологије, изазвани индустријализацијом, називани су „индустријским револуцијама“. Испрва, средином 18. века, ти помаци су се односили на развој механизације, којом је индустрија заменила пољопривреду као основа економске структуре друштва. Цео век касније, долази до настанка нових извора енергије, струје, нафте и гаса. Тада су се створили услови за даљи развој индустрије, нарочито индустрије челика, као и интензивне употребе електричне енергије. Ситуација се променила крајем 60-их година 20. века, када долази до успона електронике, тј. распрострањене употребе транзистора и микропроцесора, али и успона телекомуникација и рачунара, који су довели до широке примене информационих технологија. Може се рећи да се у садашњости одвија још једна индустријска револуција, проузрокована експанзијом Интернета, која има за циљ да повеже сва производна средства ради њихове интеракције у реалном времену. То је прва револуција која није базирана на проналаску новог извора енергије, већ на процесу дигитализације.

Као што се може видети, информационе технологије већ дуже време имају значајну улогу у развоју индустрије. Информационе технологије означавају употребу рачунара за складиштење, достављање, обраду и пренос података или информација, али и друге активности као што су пројектовање рачунарских мрежа и информационих система, инжењеринг хардвера, дизајнирање софтвера и база података и слично.

Потребно је разумети разлику између податка и информације. Податак је једноставна и сирова чињеница која нема значење и може бити број, реч, симбол итд. Податак се тумачи од стране човека или машине да би добио значење. Информација је обрађен податак или скуп података који поседује одређен контекст, значај и сврху. Додељује смисао податку и повећава његову поузданост. Када се податак трансформише у информацију, он нема бескорисних детаља и може бити искоришћен од стране људи или машина на различите начине. Организовани систем информација чини знање и може се користити за добијање нових значења и података, као и решавање неког проблема.

На слици 1 може се видети илустрован пример разлика између податка, информације и знања¹.



Слика 1. Разлике између податка, информације и знања

¹ Data to Wisdom Via Information, Knowledge & Insight. (20. јун 2019). Преузето 16. септембра 2019, са <https://www.theifactory.com/news/gaining-wisdom-from-data/>

Сада, када је дефинисано које су разлике између податка и информације, може се дефинисати појам информационог система. „Информациони систем је систем у коме се везе између објеката и везе система са околином остварују разменом информација“². Представља интегрисани скуп компоненти чији је основни задатак прикупљање, обрада, дистрибуција и архивирање информација. Сачињавају га улазни подаци и излазне информације, хардвер и софтвер рачунара, базе података, телекомуникациони системи и технологије, процеси и извршиоци.

На слици 2 је илустрован информациони систем као модел реалног система³.



Слика 2. Информациони систем као модел реалног система

Сваки информациони систем има свој животни циклус. Животни циклус представља фазе и активности које треба обавити током развоја информационог система, као што су планирање развоја, анализа захтева, пројектовање (дизајн) система, имплементација, примена и подршка.

² Полишчук, Ј. (септембар 2007). *Пројектовање информационих система*. Подгорица, Црна Гора

³ Ерић, М. (2018). *Пројектовање информационих система и база података* [PDF].

Преузето 09. октобра 2018, са

https://drive.google.com/drive/folders/1gvcBjQopMezrtq2pYCpPx8Tb_4m7lPD0?fbclid=IwAR3t9dWwdlILsdD4RxCr6hBHj-w2cWkQzX-rhMkeammn_CMHtwkLk4gbs10

2. Опис задатка дипломског рада

Потребно је пројектовати и имплементирати информациони систем који ће омогућити корисницима да креирају потребне документе, као што су захтеви, извештаји, дописи, предлози и слично. Након тога корисници прослеђују наведене документе надлежним особама, које их обрађују или прослеђују даље, притом мењајући статус документима, тако да корисник може у сваком тренутку имати увид у тренутни статус документа, као и надлежну особу којој је прослеђен или која га тренутно обрађује. Када документ стигне до свог коначног одредишта, након обраде, може бити архивиран или враћен кориснику на дораду.

На слици 3 је приказан ток докумената у овом информационом систему ⁴.



Слика 3. Ток докумената у информационом систему

Рад се неће бавити креирањем садржаја документа, претпоставља се да корисник поседује документ у електронској форми, већ дистрибуцијом документа и праћењем његовог статуса. Корисник не мора да води рачуна коме прослеђује одређени документ, већ ће бити одређено којим надлежним особама се прослеђују одређени типови докумената. Надлежне особе могу да обраде документ или да га проследе другој надлежној особи на обраду, као и да га врате кориснику ради модификације. Након обраде документа, исти се архивира, без обзира на коначан статус документа, тј. да ли је прихваћен или одбијен, и ту се ток документа завршава. У свакој фази тока документа поставља се статус и особа којој је упућен документ, те на тај начин корисник може „пратити“ своје документе.

Корисник информационог система може бити било који запослени у једној установи, нпр. факултету, а надлежне особе су такође корисници, али са већим одговорностима и овлашћењима унутар информационог система.

⁴ Document Routing and Approval. (б.д.). Преузето 16. септембра 2019, са <https://www.integrify.com/document-routing-and-approval/>

3. Поступак развоја информационог система

„Развојем информационог система треба дефинисати што објективнију слику реалног света, његових бивших и садашњих стања, као подлогу за процену будућег понашања и наравно, подлогу за даљи развој и примену информатичких технологија“⁵.

Као погодан начин за разумевање, описивање система и комуникацију између пројектаната и корисника показао се поступак моделирања. Приликом моделирања, умањује се сложеност система елиминисањем детаља, док се за графичку презентацију модела користе геометријски облици и линије.

Стандарди који подржавају моделирање су IDEF0 и IDEF1X (енгл. *Integrated Definition*). IDEF0 стандар се користи за функционално моделирање и омогућава извршење функционалне декомпозиције, стварање документације, бољу комуникацију између чланова пројектног тима и корисника и слично. Стандард IDEF1X користи се за информационо моделирање, тј. апстрактно представљање реалног система помоћу скупа ентитета, веза између објеката и атрибута. Циљ моделирања је развој технологије за интеграцију мрежа различитих конфигурација. Техника IDEF моделирања је основа за извођење поступка реинжењеринг пословних процеса.

Може се рећи да развој информационог система обухвата четири фазе:

- Функционално моделирање,
- Информационо моделирање,
- Апликативно моделирање и
- Имплементација.

На слици 4 налази се стабло активности поступка развоја информационог система⁵.



Слика 4. Стабло активности поступка развоја информационог система

⁵ Вељовић, А. (б.д.). *Развој информационог система и базе података*.

3.1. Функционално моделирање

Код функционалног моделирања пословни систем се декомпонује и планирају се потребни ресурси за реализацију. Користи се IDEF0 техника за моделирање, ради анализе система по нивоима и интеграције активности. Функционални модел састоји се од хијерархије дијаграма, који постепено приказују све више детаља о функцијама и њиховим везама са осталим деловима система.

IDEF0 функционално моделирање се користи као упутсво за опис сложених пословних процеса, омогућава брзе промене у организацији система и, најважније, користи се као прототипски приступ функционалном моделирању, где се на једноставан начин проверавају алтернативне идеје.

Функционално моделирање обухвата три подређене активности:

- Функционална декомпозиција,
- Дефинисање захтева корисника и
- Технички предуслови.

На слици 5 се налази стабло активности за активност функционалног моделирања ⁵.



Слика 5. Стабло активности за функционално моделирање

3.1.1. Функционална декомпозиција

Код функционалне декомпозиције највише пажње придаје се захтевима менаџмента, јер се услед промена тржишта, мењају концепт, принципи, архитектура и др. Функционална декомпозиција изводи се кроз три подређене активности:

- Дефинисање граница система,
- Дефинисање стабла активности и
- Верификација стабла активности.

Дефинисање граница система

Утврђивањем граница система треба јасно дефинисати циљеве који дају одговоре на следећа питања:

- Зашто се процес моделира?
- Шта ће процес приказати?
- Шта ће корисник модела направити са њим?
- Чему служи модел?

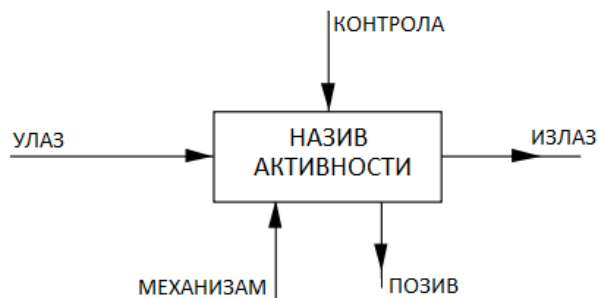
Поред тога, треба идентификовати задатке сваког запосленог и односе између задатака. За реализацију ових активности користи се IDEF0 техника, чију синтаксу чине правоугаоници, који представљају активности, затим стрелице, које представљају токове или преносе података, и правила.

Дефинисање граница система заправо представља дефинисање контекстног дијаграма, који представља највиши ниво апстракције и који се декомпозицијом преводи у нижи ниво. Одређен је једним правоугаником који представља границе модела. Унутар и ван њега теку информације путем стрелица. Однос између правоуганика и стрелица је одређен помоћу стране правоуганика на који је стрелица наслоњена.

Улази представљају информације које се користе ради дефинисања излаза. Контроле дефинишу како, када и да ли ће се активност извести и какви ће бити излази. Излази су информације добијене активношћу. Механизми су извори који изводе активност, а позиви су посебни случајеви механизма и они дају детаљнији приказ изведен у некој другој активности у истом или неком другом моделу.

Софтверски алат коришћен у раду за креирање дијаграма пословних процеса је AllFusion Process Modeler.

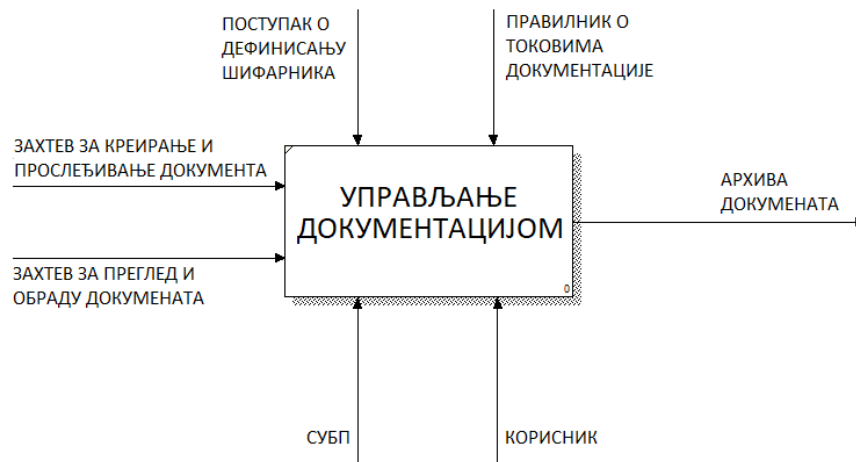
На слици 6 приказан је модел контекстног дијаграма ⁵.



Слика 6. Модел контекстног дијаграма

Што се граница система за управљање документацијом тиче, као улази постојаће захтеви за креирање нових докумената и прослеђивање докумената надлежним особама, као и захтеви за преглед докумената који су кориснику, који спада у надлежне особе, упућени и обраду тих докумената. Излаз система ће бити архива докумената, без обзира на њихов статус (прихваћени или одбијени). Као контроле система користиће се правилник о токовима документације и поступак о дефинисању шифарника појмова и особа. Систем за управљање базом података (СУБП) и сам корисник система ће бити механизми.

На слици 7 се налази контекстни дијаграм информационог система за управљање документацијом.



Слика 7. Контекстни дијаграм система за управљање документацијом

Дефинисање стабла активности

Након дефинисања граница система потребно је дефинисати стабло активности, којим се успостављају хијерархијске везе између активности, а које се реализује применом методе решавања проблема одозго-надоле. Тиме се сложена активност декомпонује на подређене активности, након чега се те једноставне, подређене активности решавају.

Дакле, корен стабла представља полазну активност, док чворови без потомака представљају активности које се једноставно решавају. Помоћу стабла је приказана хијерархија активности система, али везе између активности нису приказане.

На слици 8 приказано је стабло активности за активност „Управљање документацијом“.



Слика 8. Стабло активности система за управљање документацијом

На основу анализе информационог система, дошло се до закључка да се активност „Управљање документацијом“ може поделити на подређене активности:

- Унос документа, која обухвата креирање предмета документа, одабир фајла, категоризацију и верзионисање и, на крају, прослеђивање документа;
- Дистрибуцију докумената, у коју спадају обрада документа, закључавање или откључавање документа и праћење статуса документа;
- Архивирање докумената, која подразумева складиштење документа, евиденцију свих докумената корисника и претраживање докумената.

Верификација стабла активности

Дефинисање стабла активности је веома важна активност, те је потребно верификовати исто. Верификацију обавља верификационо тело које контролише развој система и притом благовремено отклања настале грешке. Путем верификације у израду информационог система укључује се и клијент за чије потребе се систем и развија.

У случају овог информационог система за управљање документацијом, верификационо тело представљају поједини запослени на факултету, укључујући професоре, сараднике и ненаставно особље. Оцена верификационог тела је да је комплетна функционална декомпозиција одрађена исправно, укључујући и стабло активности, и да се може наставити са радом.

3.1.2. Дефинисање захтева корисника

Упознавање пројектанта са захтевима и потребама корисника представља једну од кључних активности приликом развоја информационог система, како би систем задовољио потребе корисника, а пројектант могао да доноси правилне закључке. Дефинисање захтева корисника обухвата:

- Дефинисање захтева из докумената,
- Дефинисање захтева интервјуом,
- Дефинисање матрице односа и
- Анализа захтева корисника.

Дефинисање захтева из докумената

Прикупљање информација из постојеће документације подразумева прикупљање докумената у којима су представљени неки организациони прописи и интерни стандарди, као и разне процедуре и упутства. Ту спадају улазни и излазни документи, извештаји, прописи и др. Уколико не постоје документи са процедурама и прописима треба их написати, јер побољшавају постојећу организацију рада. За прикупљену документацију потребно је одговорити на нека од следећих питања:

- одакле потичу подаци и како су добијени;
- које су вредности минимума и максимума података у пракси;
- да ли се уносе тачни и комплетни подаци;
- како се и колико често користе;
- колико су тачни излазни подаци у садашњости;
- како се може побољшати поступак рада и др.

Од доступних докумената за информациони систем за управљање документацијом, на располагању су правилник о токовима документације и корисничко упутство за апликацију за управљање документацијом која је претходно коришћена.

Правилник о токовима документације садржи све потребне путање једног документа, од „извора“ до „понора“, и као такав је од велике важности за развој система.

Из корисничког упутства апликације се може закључити да је претходно коришћена десктоп апликација која је развијана прилично застарелим технологијама, али поседује већину потребних карактеристика које се захтевају и од информационог система који се развија, па ће бити искоришћена као пример како систем треба да функционише.

Дефинисање захтева интервјуом

Приликом интервјуа могуће је сазнати потребе, циљеве и проблеме из угла непосредних извршиоца, тј. корисника. Интервју је једна од кључних активности приликом развоја система, где руководиоци могу да одреде правац у коме ће се систем даље развијати.

Најпре је потребно припремити се за извођење интервјуа и то припремом листе руководиоца са којима ће се разговарати, тема за разговор, временског распореда, општих питања, записника итд. Циљ ових активности је развој препорука за будуће акције, идентификација проблема у постојећем систему, разјашњење одговорности и уопштено разумевање пословања. Нека од општих питања за интервјуисање су:

- које су надлежности и одговорности;
- који су циљеви и какве се промене могу очекивати убудуће;
- који су највећи проблеми и која су решења тих проблема;
- на који начин се систем може побољшати и др.

Након интервјуа потребно је написати резиме, анализирати проблеме и повезати их са процесима пословања и дефинисати закључке на основу којих ће се систем даље развијати.

Интервју је обављен са професором на факултету за који је развија информациони систем, као и запосленим у служби општих и правних послова факултета. Приликом интервјуа, између осталог, дошло се до информација о надлежностима и одговорностима запослених на факултету, што се тиче управљања документацијом, а затим се дошло до закључка да је највећи проблем немогућност праћења статуса документа, тј. запослени немају информацију о процесу обраде њихових докумената, стога је циљ направити информациони систем чија је главна карактеристика, поред креирања документа, праћење статуса истог, а један од начина побољшања тренутног система је и прелазак на новије технологије израде система.

Дефинисање матрице односа

Матрица односа представља матрицу веза улазних и излазних информација са активностима из стабла активности. Сваком објекту се додају операције помоћу којих активност може да користи тај објекат, користећи тзв. CRUD матрицу. Операције које се могу извршити над објектима су: креирање (**Create**), претраживање (**Retrieve**), ажурирање (**Update**) и брисање (**Delete**).

У оквиру претходно дефинисаних активности дефинишу се објекти Корисник, Документ, Надлежна особа, Радно место, Одељење, Тип документа и Група надлежних особа.

Табела 1 представља матрицу односа за активност „Управљање документацијом“, на којој се може видети које операције активности могу да изврше над појединим објектима.

Табела 1. Матрица односа активности „Управљање документацијом“

Назив активности	Назив објекта	C R U D
УНОС ДОКУМЕНТА	Документ	C R U D
	Корисник	R
	Радно место	R
	Одељење	R
	Тип документа	R
ДИСТРИБУЦИЈА ДОКУМЕНТА	Документ	R U
	Корисник	R
	Надлежна особа	R
	Група надлежних особа	R
	Тип документа	R
АРХИВИРАЊЕ ДОКУМЕНТА	Документ	R U
	Корисник	R
	Тип документа	R

Анализа захтева корисника

Последња активност у оквиру дефинисања захтева корисника је анализа захтева, коју спроводи верификационо тело, које том приликом одобрава наставак развоја система или указује на грешке начињене у претходним активностима.

Као и код активности „Функционална декомпозиција“, оцена верификационог тела је да су сви захтеви узети у обзир, како захтеви добијени кроз интервју, тако и они који су добијени кроз обраду постојеће документације. Матрица односа је, такође, обухватила све активности и објекте, стога се може наставити са развојем система.

3.1.3. Технички предуслови

Технички предуслови подразумевају рачунарски систем са дефинисаним системом хардвера, софтвера и документације, који ће бити потребан за развој информационог система. Систем хардвера чине радна и масовна меморија, улазне и излазне јединице и централна процесорска јединица. Систем софтвера чине оперативни систем, језички процесори (интерпретери и компајлери), апликативни софтвер и базе података. Систем документације чине документације за хардвер и софтвер и остала документација.

Активност „Технички предуслови” изводи се кроз следеће активности:

- Дефинисање архитектуре система,
- Кадровске потребе и,
- Динамика реализације и трошкови.

Дефинисање архитектуре система

Потребно је усвојити основне претпоставке за поступак реинжењеринга пословних процеса, који треба базирати на новијим методама, техникама и технологијама, као и принципима дистрибуиране обраде, коришћења база података и др. Опрему треба бирати тако да се имају у виду техничко – технолошки предуслови, који су сагледавани према структури архитектуре референтног модела:

- квалитетан комуникациони систем,
- висок степен компатибилности рачунарске опреме,
- отвореност мрежне архитектуре,
- модуларност опреме крајњих корисника,
- ефикасност система управљања подацима и
- коришћење софтверских производа за развој апликације.

С обзиром да ће апликација која ће се развијати у оквиру информационог система бити израђена веб технологијама и приступаће јој се помоћу веб претраживача, што се комуникационог система тиче, довољан је само приступ интернету.

Рачунарска опрема која ће се користити за развој система укључује само персоналне рачунаре и сервере повезане на интернет, тако да ће бити могуће приступити апликацији на серверу са било ког рачунара унутар мреже.

Како се веб претраживачи користе за приступ апликацији, није потребно поседовати рачунар високих перформанси, већ ће велика већина рачунара без икаквих проблема покретати апликацију, али се на серверу мора обезбедити да, уколико се више корисника конектује на сервер у исто време, не дође до преоптерећења.

Подацима ће се управљати системом за управљање базама података (СУБП), и то, MySQL вишекорисничким системом, док ће се за развој апликације користити неке од најновијих технологија за развој веб апликација.

Кадровске потребе

У кадровске потребе спадају потребан кадар за реализацију пројекта развоја информационог система и потребна обука за коришћење информационах технологија.

Сматра се да је потребно имати минимум потребног кадра у који спадају руководиоца, водећи пројектанти за моделирање процеса и података, софтверских решења и база података, затим систем инжењер и референт документације. Такође потребно је континуирано образовање кадра и аутоматизација њиховог рада.

Код информационог система за управљање документацијом улоге из потребног кадра за развој система биће подељене између пројектанта система (аутора рада), професора и запослених у служби за правне и опште послове.

Динамика реализације и трошкови

Динамика реализације развоја информационог система за управљање документацијом усвојена је договором између пројектанта (аутора рада) и професора.

Трошкови реализације се најчешће деле на:

- трошкове развоја апликација,
- трошкове техничко – технолошких ресурса и
- трошкови експлоатације.

Свака од група трошкова се, такође, састоји из посебних трошкова које је потребно узети у обзир, као што су обука пројектног тима, стручна помоћ при развоју, рачунарска и пратећа опрема, потрошња електричне енергије, одржавање опреме и сл.

С обзиром да је потребно развити информациони систем за управљање документацијом у оквиру дипломског рада, трошкови се не разматрају.

3.2. Информационо моделирање

Информационо моделирање се изводи на основу претходно урађене активности функционалног моделирања. Критичне функције функционалног моделирања представљају главне пројекте који се развијају у оквиру активности информационог моделирања. Информациони модел приказује везе између података у систему.

Информационо моделирање омогућава дефинисање системске документације, бољу комуникацију, јасну слику о пословним правилима и „логичку” слику базе података и обухвата следеће четири подређене активности:

- Дефинисање детаљних захтева,
- Креирање „ЕР“ дијаграма,
- Креирање атрибута и
- Дефинисање пословних правила.

На слици 9 приказано је стабло активности информационог моделирања ⁵.



Слика 9. Стабло активности информационог моделирања

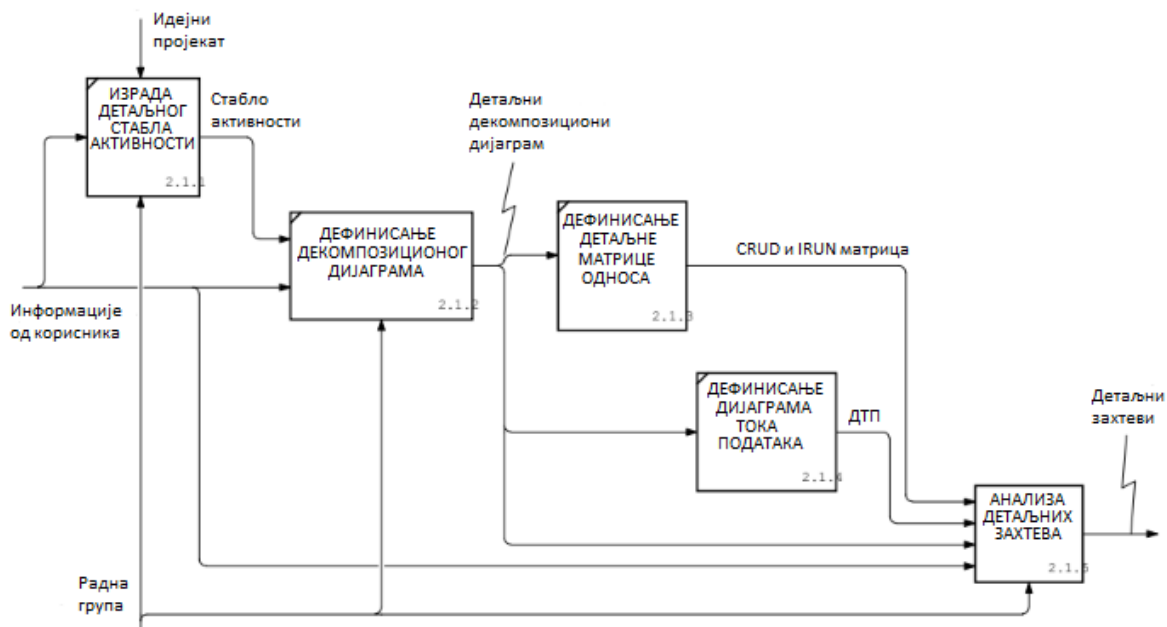
3.2.1. Дефинисање детаљних захтева

На основу претходно урађене активности “Функционално моделирање” биће изведена и активност дефинисања детаљних захтева, која представља детаљно продубљивање функција за које се спроводи реинжењеринг пословних процеса. Изводи се кроз следеће активности:

- Израда детаљног стабла активности,
- Дефинисање декомпозиционог дијаграма,
- Дефинисање детаљних матрица односа,
- Дефинисање дијаграма тока података и
- Анализа детаљних захтева.

За дефинисање токова информација између подређених активности користиће се декомпозициони дијаграми.

На слици 10 је приказан декомпозициони дијаграм за активност „Дефинисање детаљних захтева“, на коме је дефинисан ток информација, почев од улазног документа „Идејни пројекат“, преко набројаних активности, до излазног документа „Детаљни захтеви“ ⁵.



Слика 10. Декомпозициони дијаграм дефинисања детаљних захтева

Израда детаљног стабла активности

Активност израде детаљног стабла активности се изводи за изабрану критичну функцију, приликом које се декомпозиција врши до нивоа примитивних процеса. Улаз у ову активност је идејни пројекат на основу добијених информација од корисника, а излаз је детаљно стабло активности за сваку критичну функцију. Као што је претходно поменуто, корен стабла представља полазну активност, док чворови без потомака представљају примитивне активности које се једноставно решавају.

У активности „Дефинисање стабла активности“ у оквиру функционалног моделирања приказано је стабло активности за полазну активност „Управљање документацијом“. Овде ће бити израђена стабла активности за њене подређене активности.

На слици 11 приказано је стабло активности за активност „Унос документа“.



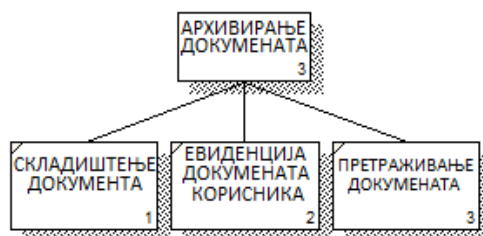
Слика 11. Стабло активности уноса документа

На слици 12 је приказано стабло активности за активност „Дистрибуција докумената“.



Слика 12. Стабло активности дистрибуције докумената

На слици 13 је приказано стабло активности за активност „Архивирање докумената“.



Слика 13. Стабло активности архивирања докумената

Дефинисање декомпозиционог дијаграма

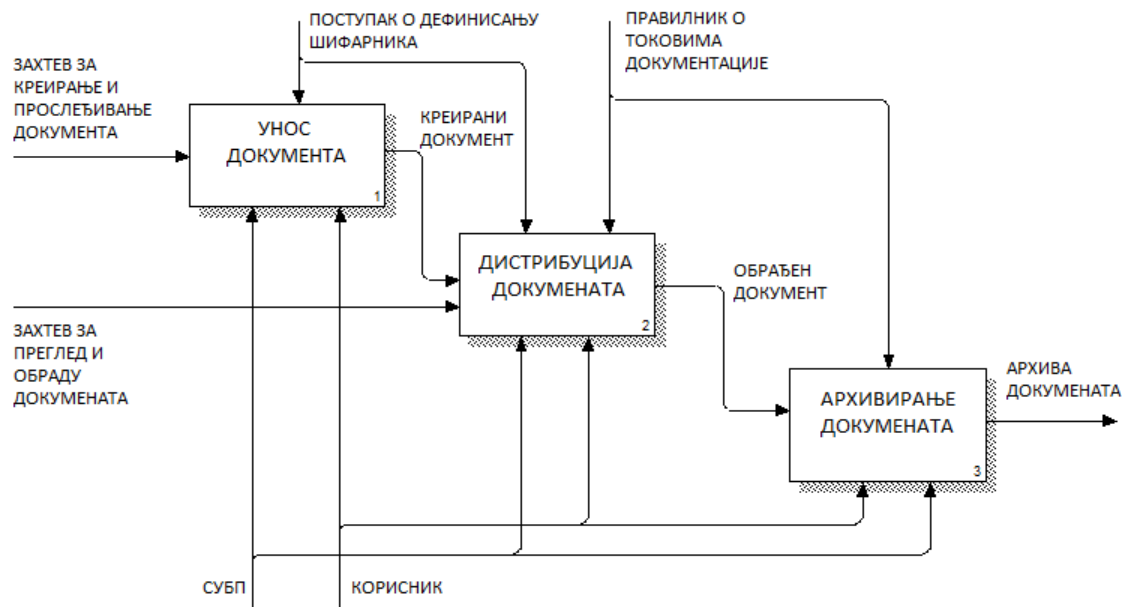
У оквиру ове активности успостављају се хоризонталне везе између повезаних подактивности. Улази у ову активност су детаљна стабла активности и информације од корисника, а излази су детаљни декомпозициони дијаграми.

Декомпозициони дијаграм представља спајање неповезаних активности са стабла активности, а са претходно дефинисаног контекстног дијаграма пренесене су граничне стрелице, које је такође потребно повезати са активностима.

Активност „Управљање документацијом“ се декомпонује на три подређене активности:

- Унос документа, чији је улаз захтев за креирање и прослеђивање документа, излаз је креирани документ, контрола је поступак о дефинисању шифарника појмова и особа, а механизми су СУБП и сам корисник.
- Дистрибуција докумената, чији су улази креирани документи и захтев за преглед и обраду докумената, излаз је обрађени документ спреман за даљу дистрибуцију или архивирање, контроле су поступак о дефинисању шифарника појмова и особа и правилник о токовима документације, а механизми СУБП и корисник.
- Архивирање докумената, чији је улаз обрађени документ, излаз је архива свих докумената, контрола је правилник о токовима документације, а механизми СУБП и корисник.

Слика 14 представља декомпозициони дијаграм система за управљање документацијом.



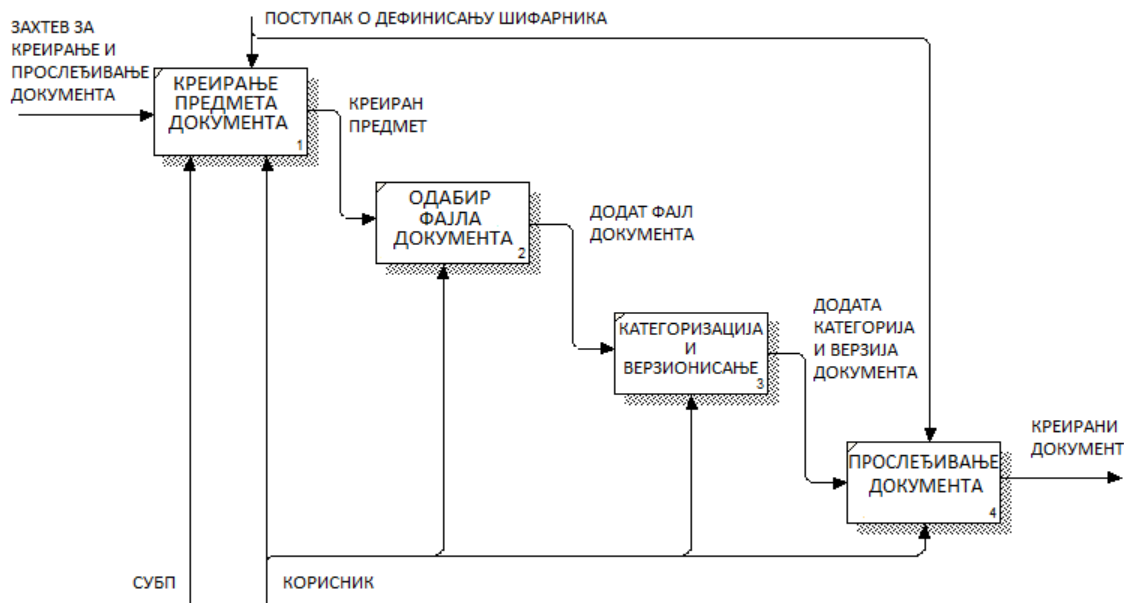
Слика 14. Декомпозициони дијаграм система

Претходни дијаграм представља декомпозициони дијаграм целог информационог система, односно његове полазне активности. Такође, представља нижи ниво апстракције у односу на контекстни дијаграм. Дијаграми који ће бити представљени на наредним сликама представљаће још нижи ниво апстракције, где ће бити креирани декомпозициони дијаграми свих подактивности полазне активности. Граничне стрелице ће бити пренесене са претходно дефинисаног декомпозиционог дијаграма полазне активности.

Подактивност „Унос документа“ декомпонована је на четири једноставне активности:

- Креирање предмета документа, чији је улаз захтев за креирање и прослеђивање документа, излаз је креиран предмет, контрола је поступак о дефинисању шифарника појмова и особа, а механизми СУБП и корисник.
- Одабир фајла документа, чији је улаз креирани предмет, излаз је предмет са додатим фајлом документа, у електронском облику или унос скенирањем папирних докумената, а механизам је корисник.
- Категоризација и верзионисање, чији је улаз предмет са додатим фајлом документа, излаз је предмет са додатом категоријом документа и почетном верзијом, а механизам је корисник.
- Прослеђивање документа, чији је улаз предмет са додатом категоријом и верзијом, излаз је креирани документ, контрола је поступак о дефинисању шифарника појмова и особа, а механизам, такође, корисник.

На слици 15 приказан је декомпозициони дијаграм за активност „Унос документа“.

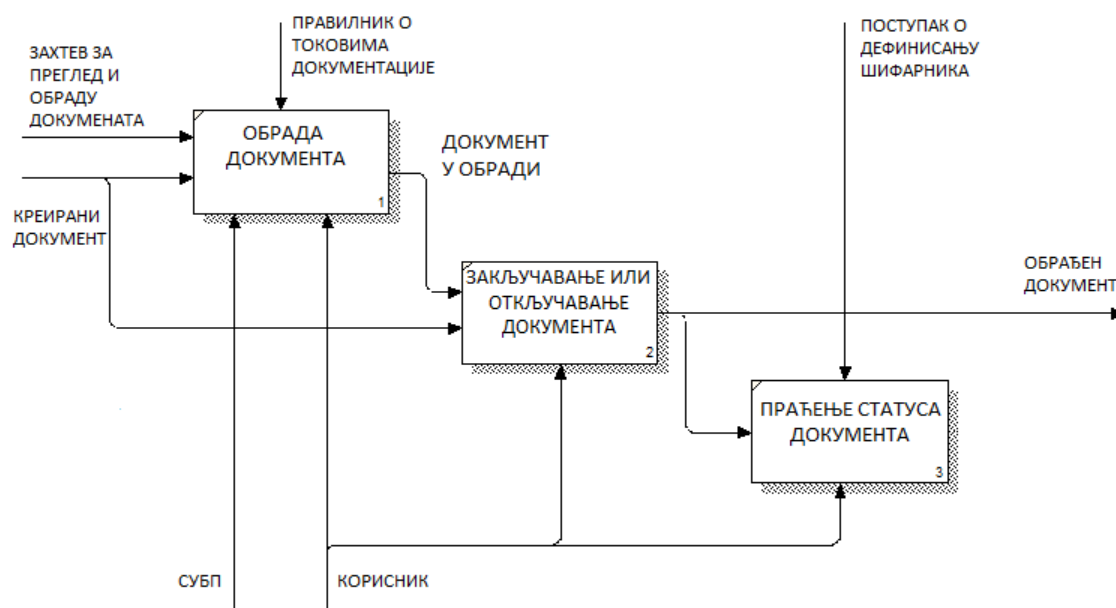


Слика 15. Декомпозициони дијаграм активности „Унос документа“

Друга подактивност је „Дистрибуција докумената“ која обухвата три подређене активности:

- Обрада документа, чији су улази креирани документ и захтев за преглед и обраду документа, приликом ког надлежна особа може да „реши“ документ који јој је прослеђен, да проследи документ другој надлежној особи на обраду или да врати документ кориснику на модификацију, излаз активности је обрађен документ, контрола је правилник о токовима документације, а механизми су СУБП и корисник.
- Закључавање или откључавање документа, у зависности од тога да ли је документ у процесу обраде или је враћен кориснику на модификацију. Када надлежна особа започне обраду документа који јој је упућен, корисник не може више мењати документ, осим у случају када му је документ враћен. Улази су креиран документ или документ у процесу обраде од стране надлежне особе, излаз је обрађен документ, а механизам је корисник.
- Праћење статуса документа од стране корисника, чији је улаз обрађени документ, чији тренутни статус и надлежну особу којој је упућен корисник може пратити, контрола је поступак о дефинисању шифарника појмова и особа, а механизам сам корисник.

На слици 16 се налази декомпозициони дијаграм за активност „Дистрибуција докумената“.

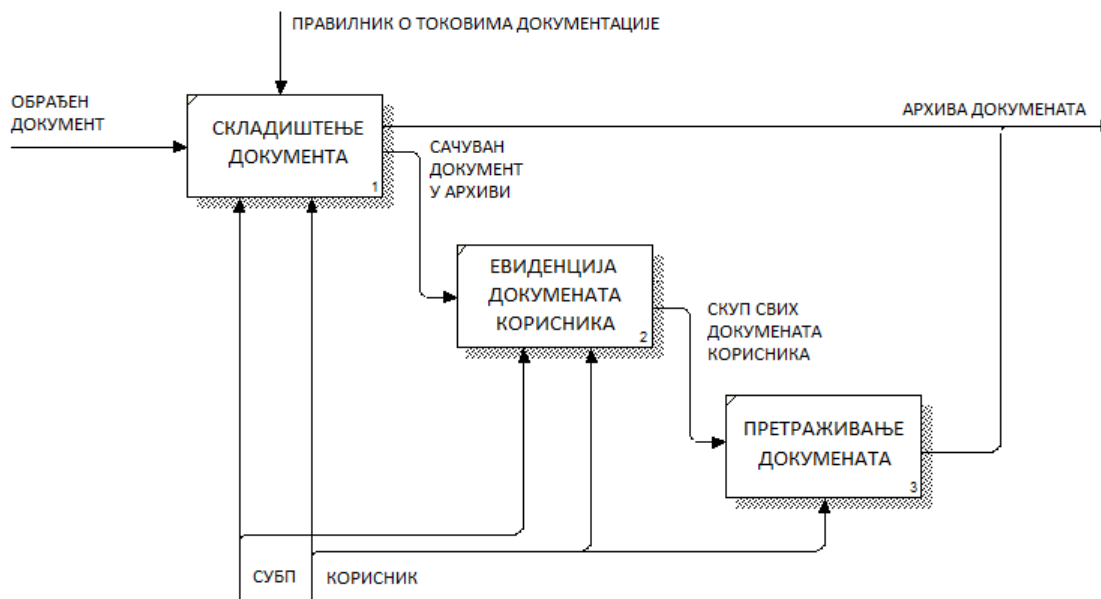


Слика 16. Декомпозициони дијаграм активности „Дистрибуција докумената“

Последња подактивност је „Архивирање докумената“ и обухвата три подређене активности:

- Складиштење докумената, које се извршава након реализоване обраде документа и која подразумева чување документа у трајној меморији, чији је улаз сам обрађен документ, излаз је сачуван документ у архиви, контрола је правилник о токовима документације, а механизми су СУБП и корисник.
- Евиденција докумената корисника, где корисник има увид у све своје документе, архивираних и активних, чији су улази документи сачувани у архиви, излаз је скуп свих докумената корисника, архивираних и активних, а механизми су СУБП и сам корисник.
- Претраживање докумената корисника по параметрима докумената, укључујући датум израде, тип или статус документа, надлежну особу и слично, чији је улаз скуп свих докумената корисника, излаз је филтрирана архива докумената, а механизам је корисник.

На слици 17 приказан је декомпозициони дијаграм за активност „Архивирање докумената“.



Слика 17. Декомпозициони дијаграм активности „Архивирање докумената“

Дефинисање детаљних матрица односа

Претходно је била дефинисана матрица односа која означава на који начин активности могу користити одређене објекте (CRUD матрица). Она се може проширити повезивањем објеката са атрибутима, чиме настаје IRUN матрица. Повезаност објеката са атрибутима може бити: убацивање атрибута (**I**nsert), претраживање атрибута (**R**etrieve), ажурирање атрибута (**U**ppdate) и додела нула вредности (**N**ull field). Дакле, IRUN матрица дефинише коришћење атрибута у одређеном објекту за одређену активност.

Функционални модел је повезан са информационим моделом преко речника података. Индиректном синхронизацијом могуће је синхронизовање једног модела података са више модела процеса. У случају да се појави потреба за новим објектом или атрибутом, или да се измени постојећи, може се изменити модел процеса, а затим надоградити речник података, као веза са моделом података.

Поступак „инверзног инжењерства“, када од готовог софтверског производа настаје његов модел података који се убацује у декомпозициони дијаграм за касније моделирање података је један од важнијих поступака у развоју информационих система.

У табели 2 биће приказан извештај у облику комбиноване CRUD и IRUN матрице, где се за са једне стране налазе претходно дефинисане активности и објекти, као и операције које активност може извршити над тим објектима, док се са друге стране налазе атрибути сваког објекта и операције којима су повезани са објектима.

Табела 2. CRUD и IRUN матрица за активност „Управљање документацијом“

Назив активности	Назив објекта	C R U D	Назив атрибута	I R U N
УНОС ДОКУМЕНТА	Корисник	R	Шифра корисника	R
			Име	I U
			Презиме	I U
			Звање	I U N
			ЈМБГ	I
			Мејл	I
			Лозинка	I U
			Шифра радног места	R U
			Шифра одељења	R U N
	Радно место	R	Шифра радног места	R
			Назив радног места	I U
	Одељење	R	Шифра одељења	R
			Назив одељења	I U
ДИСТРИБУЦИЈА ДОКУМЕНАТА	Надлежна особа	R	Шифра надлежне особе	R
			Назив позиције надлежне особе	I U
			Шифра корисника	I U
	Група надлежних особа	R	Шифра групе надлежних особа	R
			Назив групе надлежних особа	I U
АРХИВИРАЊЕ ДОКУМЕНАТА	Документ	R U	Шифра документа	R
			Датум креирања документа	I
			Назив фајла документа	I U N
			Тренутни статус документа	R U
			Шифра типа документа	I U
			Шифра надлежне особе	R U
			Шифра корисника	R
	Тип документа	R	Шифра типа документа	R
			Назив типа документа	I U
			Шифра надлежне особе	I U

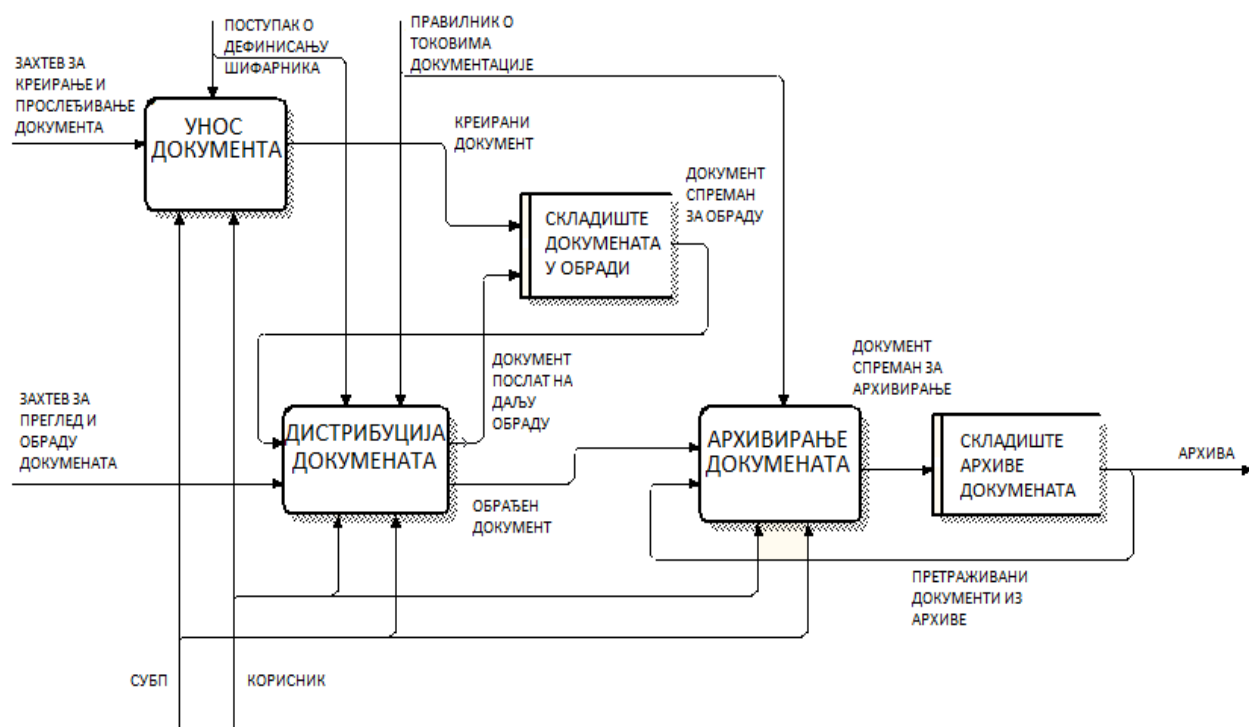
Дефинисање дијаграма тока података

Графички опис који повезује процесе, токове података и складишта података назива се дијаграм тока података, чија је намена фокусирање на проблем токова податак између процеса и анализа складишта података ради оптимизације расположивости и времена претраживања.

Дијаграм тока података се графички описује помоћу:

- процеса, који представљају низ операција којима се улазни подаци трансформишу у излазне;
- тока података, који представљају пут којим протичу подаци и који показују између којих елемената се одвија ток података;
- складишта података, који служе за чување података;
- спољних објеката, који извршавају повезивање са објектима ван посматраног система и представљају изворе и поноре података.

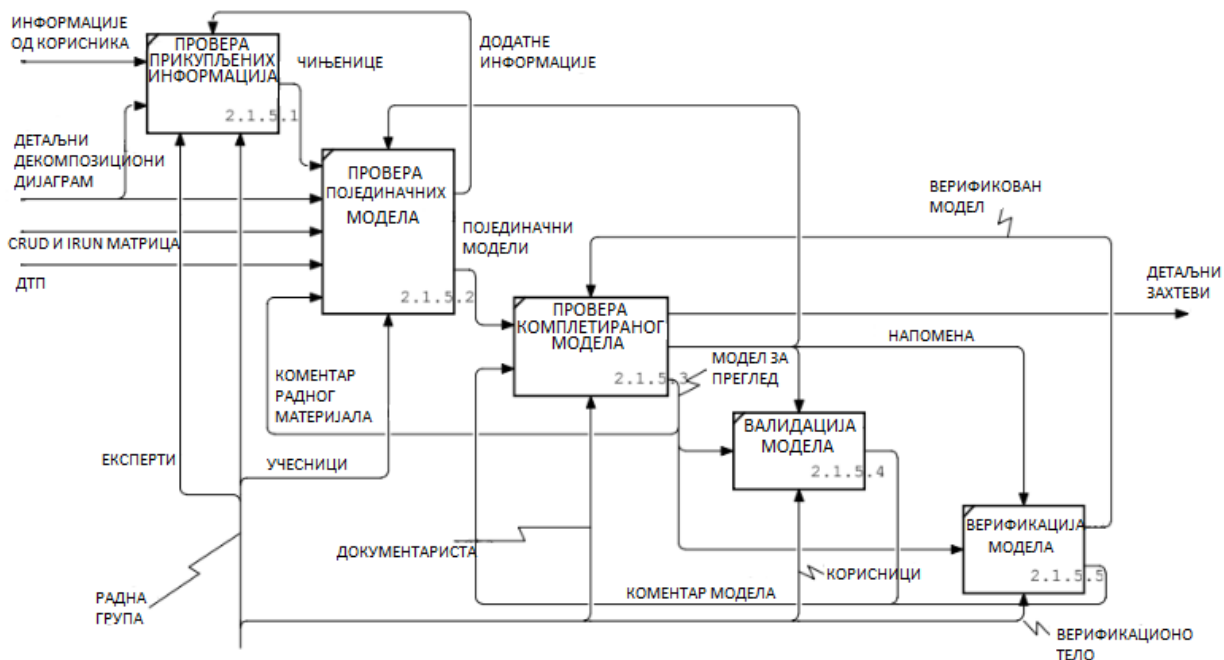
На слици 18 се може видети дијаграм тока података за полазну активност „Управљање документацијом“. На њему се може видети да се улазни подаци након прве активности смештају у складиште докумената у обради, након чега следи обрада и дистрибуција докумената, а на крају, након последње активности, документи се смештају у складиште архиве докумената, из ког се добијају излазни подаци.



Слика 18. Дијаграм тока података за активност „Управљање документацијом“

Анализа детаљних захтева

Анализа детаљних захтева представља преглед свих, до сада извршених, активности. На слици 19 представљен је редослед корака провере детаљних захтева у форми дијаграма ⁵.



Слика 19. Редослед корака провере детаљних корака

„Провера прикупљених информација“ представља проверу постављања модела реинжењеринга, тј. провера да ли се у међувремену десила нека измена.

Улаз у систем представља захтев корисника да креира нови предмет документа и проследи га надлежној особи или захтев надлежне особе да прегледа или обради документа која су њој упућена. Излаз из система представља архиву свих документа. Документи се обрађују посредством правилника о токовима документације и поступка дефинисања шифарника појмова и особа, а уз помоћ система за управљање базом података (СУБП) и корисника.

Претходно наведене информације се поклапају претходним захтевима корисника, те је прва активност спроведена.

„Провера појединачних модела“ представља анализу нижег нивоа апстракције система, тј. анализа детаљних декомпозиционих дијаграма, матрица односа и дијаграма тока података. Претходно наведени дијаграма и матрице се такође уклапају у наведене захтеве корисника, стога је провера појединачних модела такође успешно спроведена.

Након тога следи активност „Провера комплетираног модела“ која представља проверу декомпозиционог дијаграма целог система, односно полазне активности. С обзиром да су појединачни модели исправни, може се закључити и да је комплетирани модел такође исправан, те је и трећа активност спроведена.

Активност „Валидација модела“ спроводе корисници, чији коментари представљају излаз ове активности и њихова је оцена да је модел система исправан.

Последњу активност „Верификација модела“ спроводи верификационо тело, које врши проверу модела. Верификационо тело је, такође, потврдило је да је модел система исправан и да је комплетна динамика система дефинисана.

3.2.2. Креирање ЕР дијаграма

Креирање ЕР дијаграма подразумева моделирање података, односно дефинисање дијаграма ентитета. За његово креирање се користи модел података, којим се дефинишу ентитети (објекти). Сви ентитети имају своје атрибуте и међусобно су повезани везама.

Ова активност обухвата следеће подактивности:

- Идентификација кандидата за ентитете,
- Идентификација веза,
- Дефинисање ЕР модела и
- Верификација ЕР дијаграма.

Идентификација кандидата за ентитете

Ентитет представља објекат посматрања, односно све што се може једнозначно идентификовати, а самим тим и изоловати из околине и описати. Ентитет може бити особа, ствар, догађај или појам од интереса и описује се помоћу својих атрибута. Могу се описати на основу пасивне и активне улоге везе, на основу атрибута на документима и на основу интереса посматрања.

Кандидати за ентитете у овом информационом систему биће објекти који су претходно наведени приликом дефинисања матрице односа (CRUD матрице) и то:

- Корисник,
- Радно место,
- Одељење,
- Надлежна особа,
- Група надлежних особа,
- Документ и
- Тип документа.

Идентификација веза

У овој активности први пут се дефинишу везе између ентитета. Веза представља међусобну повезаност ентитета и користи се за дефинисање зависности између ентитета, односно описивање начина повезивања ентитета. Везе се означавају „глаголским фразама“ и идентификују се као повезивање ентитета на основу интереса, дефинисање зависности ентитета (зависни и независни ентитети, дефинисање значења зависности (референцијални интегритет) итд.

Везе у информационом систему за управљање документацијом, на основу претходно наведених кандидата за ентитете могу бити:

- „Ради“ – повезује ентитете „Корисник“ и „Радно место“;
- „Припада“ – повезује ентитете „Корисник“ и „Одељење“;
- „Је“ – повезује ентитете „Корисник“ и „Надлежна особа“;
- „Је члан“ – повезује ентитете „Корисник“ и „Група надлежних особа“;
- „Креира“ – повезује ентитете „Корисник“ и „Документ“;
- „Је типа“ – повезује ентитете „Документ“ и „Тип документа“;
- „Упућен“ – повезује ентитете „Документ“ и „Надлежна особа“;
- „Задужен за“ – повезује ентитете „Надлежна особа“ и „Тип документа“.

Дефинисање ЕР модела

Поступак дефинисања ЕР (entitet-relacija) модела обухвата:

- дефинисање независних ентитета,
- дефинисање зависних ентитета и
- дефинисање веза.

Софтверски алат коришћен за моделирање података је „CA Erwin Data Modeler“.

Независни ентитети поседују барем једну особину која их може једнозначно идентификовати, тј. не зависе од других ентитета. У овом случају, независни ентитети су „Радно место“, „Одељење“, „Група надлежних особа“ и „Тип документа“.

Зависни ентитети су они ентитети чија идентификација и постојање зависе од других ентитета². Могу бити карактеристични, асоцијативни, пројектни и ентитети категорије.

Карактеристични ентитет (слаб ентитет) представља ентитет који се идентификује преко независног ентитета и у овом случају то су „Корисник“, „Документ“ и „Надлежна особа“.

Асоцијативни ентитети су ентитети који представљају везу више ентитета. Пројектни ентитети имају сличне особине као асоцијативни, али немају своје атрибуте, док ентитети категорије представљају поткатогије ентитета. За сада се ни један од ентитета не може сврстати у неку од ових група.

Веза представља асоцијацију између два или више ентитета, тј. однос међу њима. Ентитет од кога полази веза зове се „родитељ“, а ентитет ка коме иде веза је „дете“. Веза између ентитета има особину кардиналности, тј. означава количину ентитета који су укључени у везу. Везе могу бити идентификујуће, неидентификујуће, неодређујуће и везе категорије.

Код идентификујуће везе кључ ентитета „родитељ“ представља део идентитета ентитета „дете“. Ни једна веза претходно дефинисана не спада у ову групу веза.

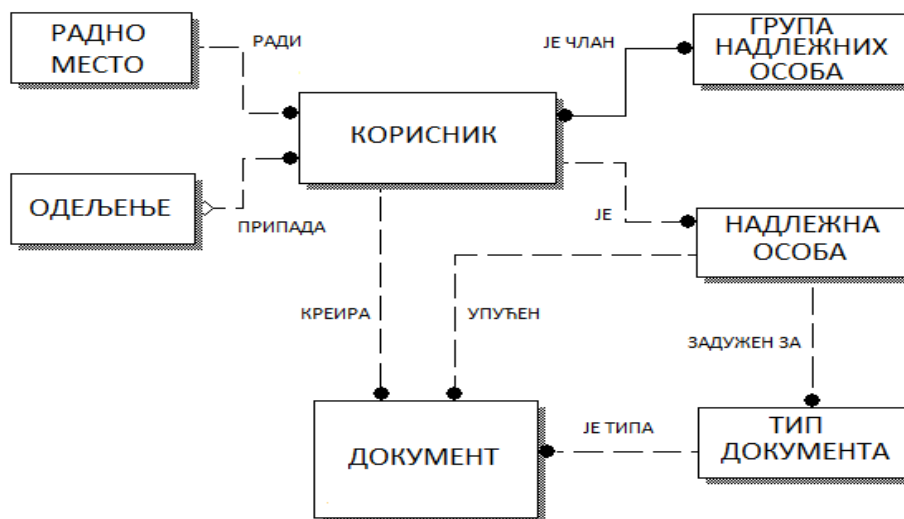
Ако се ентитет „дете“ може јединствено идентификовати, онда се таква веза назива неидентификујућа. Може бити обавезна и необавезна, у зависности од тога да ли је код „детета“ обавезан унос кључа „родитеља“ или није. У овом систему све везе, осим везе „Је члан“, се могу дефинисати као неидентификујуће.

Везе категорије су дефинисане за хијерархију надређених и подређених ентитета и таквих веза нема у систему за управљање документацијом, а неодређујуће везе говоре о томе да се један ентитет може придружити већем броју ентитета и обрнуто. Веза „Је члан“ је пример неодређујуће везе.

Верификација ЕР дијаграма

Верификација ЕР дијаграма представља општу проверу ентитета и веза и обавезно се изводи са корисницима. Дијаграм представља резултате анализе докумената и података добијених у разговору са корисницима.

На слици 20 налази се верификовани ЕР дијаграм информационог система за управљање документацијом, одобрен од стране корисника.



Слика 20. ЕР дијаграм система

3.2.3. Креирање атрибута

Атрибутима се описују карактеристична својства ентитета. Активност „Креирање атрибута“ се дели на следеће подактивности:

- Дефинисање листе кандидата за атрибуте,
- Дефинисање кључева,
- Поступак нормализације и
- Дефинисање атрибута.

Дефинисање листе кандидата за атрибуте

Основна правила за дефинисање листе кандидата за атрибуте су:

- Сви ентитети имају произвољан број атрибута,
- Један атрибут припада само једном ентитету,
- Сваки примерак ентитет има вредности за све атрибуте,
- Примерак ентитета има само једну вредност за један атрибут,
- Сваки атрибут мора имати једно доследно значење.

У претходно представљеној IRUN матрици односа налазе се кандидати за атрибуте свих ентитета, тако да колону атрибута у тој матрици можемо сматрати листом кандидата.

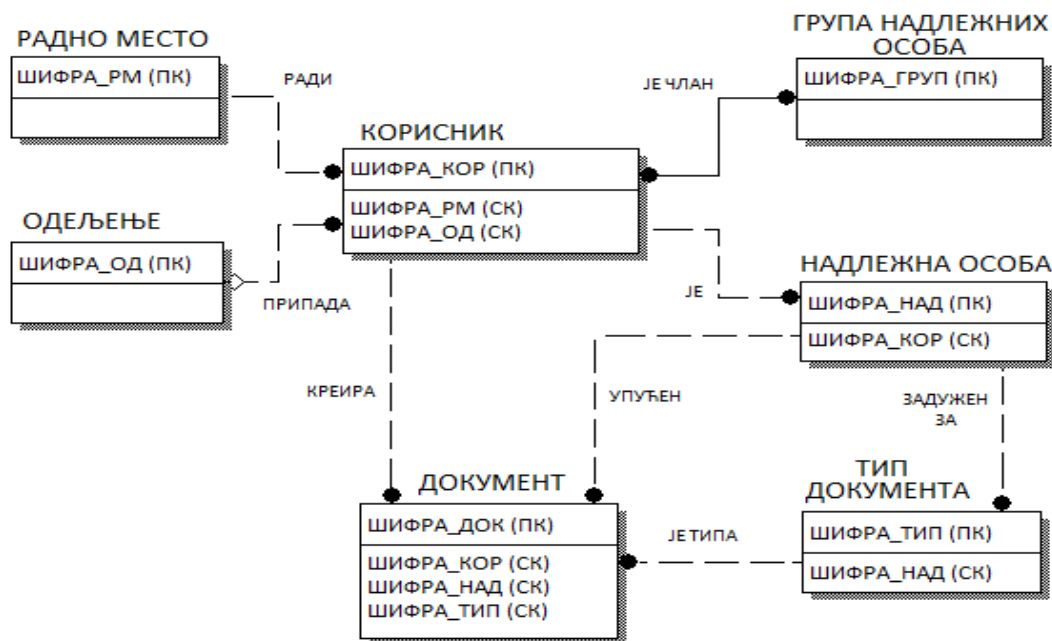
Дефинисање кључева

За сваки ентитет потребно је дефинисати атрибут или групу атрибута чије вредности ће јединствено идентификовати тај ентитет. Тај атрибут или група атрибута назива се примарни кључ ентитета, а најчешће се за примарни кључ узима шифра тог ентитета. Из претходног следи да атрибут који представља примарни кључ ентитета мора бити обавезан, тј. не сме да недостаје. У овом случају за кључеве свих ентитета биће одабране шифре тих ентитета, јер су оне јединствене.

Поред примарних кључева постоје и алтернативни кључеви, који су такође јединствени за ентитет, али не спадају у примарни кључ (нпр. ЈМБГ код корисника).

Поред набројаних, постоје и пренесени (страни) кључеви, који у датом ентитету не представљају кључ, али представљају кључ неког другог ентитета. Они служе за повезивање ентитета „родитељ“ са ентитетом „дете“.

На слици 21 приказан је ЕР дијаграм са додатим кључевима.



Слика 21. ЕР дијаграм са додатим кључевима

Поступак нормализације

Нормализација подразумева отклањање редувансе података, тј. представља трансформацију ентитета тако да су сви његови атрибути функцијски зависни од кључа, а међусобно независни. Нормализација се изводи кроз три нормалне форме.

Прва нормална форма је испуњена ако сваки атрибут ентитета има само једну вредност и једно значење. Према листи кандидата за атрибуте, сви атрибути испуњавају поменуте услове, па следи да је прва нормална форма испуњена.

Друга нормална форма захтева да прва нормална форма буде испуњена и да сваки атрибут који није кључ потпуно зависи од примарног кључа, тј. није могуће пронаћи атрибут ако се зна само део кључа ентитета. С обзиром да су у овом моделу система сви кључеви простог типа, може се рећи да је и друга нормална форма испуњена.

Трећа нормална форма је задовољена ако сваки атрибут који није кључ или део кључа зависи од читавог кључа и не служи ничему другом, тј. не може се извести или израчунати из неког другог атрибута. С обзиром да су сви наведени атрибути описни и нису изведени из других, закључак је да је и трећа нормална форма испуњена.

Поред наведене три нормалне форме постоје још четврта и пета нормална форма, као и Боусе-Codd-ова нормална форма, али оне неће бити разматране у случају овог информационог система.

Дефинисање атрибута

За дефинисање атрибута потребно је извршити идентификацију, алокацију и ревизију атрибута. Атрибути су идентификовани на основу захтева корисника и пословне документације. Алокација подразумева пребацивање атрибута из једног ентитета у други, али у овом случају није било потребе алоцирати ни један атрибут. Приликом ревизије могуће је елиминисати вишеструке вредности неког атрибута у једном појављивању ентитета, али у овом систему сви атрибути могу имати само једну вредност за једно појављивање ентитета.

За сваки атрибут се могу дефинисати скуп вредности, правила дозвољених вредности, недостајуће вредности (NULL) или међусобно искључујуће вредности.

3.2.4. Дефинисање пословних правила

Активност „Дефинисање пословних правила“ обухвата следеће подактивности:

- Дефинисање кардиналности веза,
- Дефинисање референцијалног интегритета и
- Идентификација пословног домена.

Дефинисање пословних правила повезано је са структурним динамичким правилима интегритета која обухватају:

- ограничења којима се дефинишу стања базе података,
- операције коју могу да наруше ограничења и
- акције које треба извршити ако дође до нарушавања ограничења.

Постоје структурна ограничења (интегритет ентитета, референцијални интегритет), ограничења над стандардним доменом (тип података, дужина података), ограничења над вредношћу домена (оператори поређења, опсег дозвољених вредности, листа константи из одговарајућег домена, „NOT NULL“ поља) и ограничења на кардиналности (код „родитељског“ ентитета „нула, један или више“, „један или више“, „нула или један“, конкретне вредности, а код ентитета „дете“ тотално или парцијално учешће).

Операције могу бити убацивање слога (креирање ентитета или везе, додавање вредности ентитету или вези), измена слога (измена вредности која није кључ или измена вредности у свим ентитетима, ако је део кључа измењен) или брисање слога (брисање ентитета, везе или свих ентитета типа „дете“, ако је обрисан ентитет типа „родитељ“).

Акције могу бити следеће: одбијање операције, прослеђивање операције на везни ентитет, убацивање „претпостављеног“ објекта уместо недостајућег објекта, убацивање „NULL“ вредности уместо недостајућег објекта и одсуство акције.

Дефинисање кардиналности веза

Везе имају особину која се назива кардиналност пресликавања којом се дефинишу пресликавања „родитељ - дете“ и „дете – родитељ“, тј. одређивање количине инстанци ентитета „родитеља“ и „детета“ у вези.

Пресликавање „родитељ – дете“ означава са колико се инстанци „детета“ повезује свака инстанца „родитељ“ и може бити:

- нула, један или више;
- један или више;
- нула или један;
- тачан број.

Кардиналност пресликавања „дете – родитељ“ може бити:

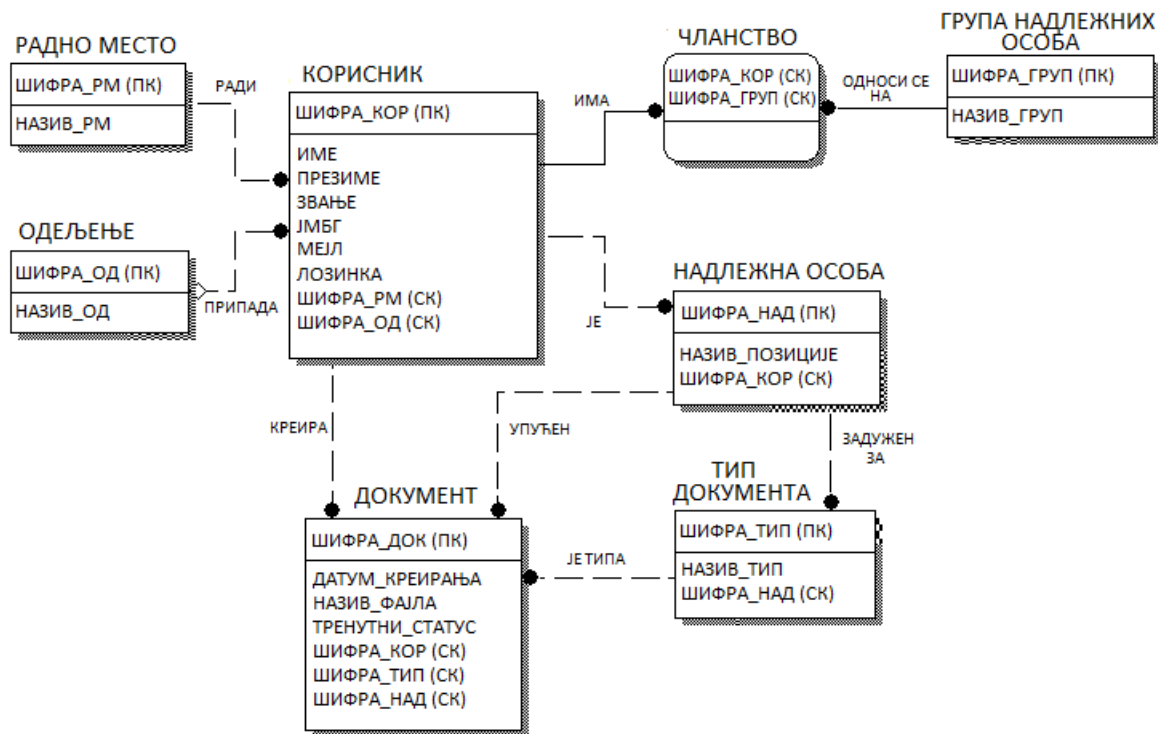
- кардиналност са дозвољеном „NULL“ вредношћу страног кључа;
- кардиналност где није дозвољена „NULL“ вредност страног кључа.

Већина веза у ЕР дијаграму информационог система за управљање документацијом је неидентификујућег типа, што значи да се ентитет „дете“ може идентификовати без знања везе са ентитетом „родитељ“. У зависности од тога да ли је веза са „родитељем“ обавезна, ентитет „дете“ може имати егзистенцијалну зависност. У овом случају кључ „родитеља“ прелази у ентитет „дете“ као страни кључ. Примери таквих веза у овом систему су „Радно место - Корисник“, „Одељење - Корисник“ (ентитет „Корисник“ није егзистенцијално завистан од ентитета „Одељење“), „Корисник – Надлежна особа“, „Корисник – Документ“, „Надлежна особа – Документ“, „Надлежна особа – Тип документа“ и „Тип документа – Документ“.

Осим неидентификујућих веза, у ЕР дијаграму овог информационог система налази се још и неодређујућа веза типа „више према више“ и то веза „Корисник – Група надлежних особа“. Код овог типа везе примерак једног ентитета везује се са нула, једним или више примерака другог ентитета и обрнуто. Основно правило које се мора испоштовати у овом случају је да се веза „више према више“ мора претворити у везу типа „нула или један према више“, а то се постиже креирањем асоцијативног ентитета који представља везу два ентитета и који је егзистенцијално – идентификационо завистан од њих. Примарни кључ новог ентитета је сложени кључ који чине примарни кључеви два ентитета, а поред кључа нови ентитет може имати и друге атрибуте, као и сви остали ентитети.

Следећи корак је дефинисање пуног атрибутивног модела који представља претходно дефинисани ЕР дијаграм са ентитетима у којима се, поред кључева, налазе и остали атрибуту. Такође, потребно је додати и нове ентитете који су настали као веза типа „више према више“ између ентитета и дефинисати везе нових ентитета са осталим ентитетима и њихове атрибуте.

На слици 22 може се видети пуни атрибутивни модел за информациони систем за управљање документацијом.



Слика 22. Пуни атрибутивни модел

Дефинисање референцијалног интегритета

Референцијални интегритет се односи на постојање пренесеног (страног) кључа у неком ентитету и обезбеђује коректно повезивање ентитета. Ентитет у коме се дефинише примарни кључ назива се „родитељ“, а ентитет где се налази пренесени кључ „дете“ назива се „дете“. Интегритетом ентитета обезбеђује се да два ентитета не могу имати исти примарни кључ и да кључ не може имати „NULL“ вредност. Референцијални интегритет обезбеђује да унесена вредност атрибута одговара вредности атрибута који је примарни кључ другог ентитета и описује понашање модела када долази до нарушавања кардиналности веза. Дефинише се посебно за ентитет „родитељ“ и ентитет „дете“ и то за сваку операцију (убацивање, ажурирање, брисање).

Након креирања новог ентитета „Чланство“, који је производ везе између ентитета „Корисник“ и ентитета „Група надлежних особа“, створиле су се и две везе идентификујућег типа и то „Корисник – Чланство“ и „Група надлежних особа – Чланство“. У том случају, примарни кључеви „родитеља“ постају делови сложеног кључа „детета“, па ентитет „Чланство“ има егзистенцијалну зависност према поменутом два ентитета. Потребно је дефинисати референцијални интегритет за све операције могу могу нарушити кардиналност везе, за ентитете који су повезани. У овом случају, прво се испитује веза „Корисник – Чланство“, а затим и „Група надлежних особа – Чланство“.

Референцијални интегритет за везу „Корисник – Чланство“ је дефинисан при извођењу операција:

- Убацавање: Ентитет „Корисник“ се може унети без ограничења, а ентитет „Чланство“ се не може унети без ентитета „Корисник“;
- Ажурирање: Каскадно за ентитет „Корисник“, тј. аутоматски се ажурира „Чланство“ кад се измени „Корисник“, а „Чланство“ се не може ажурирати ако не постоји „Корисник“;
- Брисање: Каскадно за „Корисника“, тј. бришу се сви ентитети „Чланство“ за обрисаног „Корисника“, а „Чланство“ се може обрисати без ограничења.

Дефинисан је и референцијални интегритет за везу „Група надлежних особа – Чланство“ за операције:

- Убацавање: Ентитет „Група надлежних особа“ се уноси без ограничења, а „Чланство“ не може бити унето без „Групе надлежних особа“;
- Ажурирање: Каскадно се ажурира „Чланство“ када се измени „Група надлежних особа“, а „Чланство“ се не може мењати без „Групе надлежних особа“;
- Брисање: Не може се обрисати „Група надлежних особа“ ако је „родитељ“ ентитету „Чланство“, а „Чланство“ се може брисати без ограничења.

Ако се ради о вези неидентификујућег типа, кључ ентитета „родитељ“ се налази у описном делу ентитета „дете“ као пренесени (страни) кључ. Егзистенцијална зависност ентитета „дете“ од ентитета „родитељ“ зависи од тога да ли је веза обавезна или не, тј. да ли дозвољава „NULL“ вредности за страни кључ.

У овом систему постоји само једна неидентификујућа веза у којој не постоји егзистенцијална зависност ентитета и то је веза „Одељење – Корисник“. Референцијални интегритет за ту везу је при извођењу операција је следећи:

- Убацавање: Оба ентитета се могу уносити без ограничења, али ако се уноси „Корисник“ без „Одељења“, онда се вредност страног кључа поставља на „NULL“;
- Ажурирање: Каскадно се ажурира „Корисник“ када се измени „Одељење“, а „Корисник“ се може ажурирати без ограничења;
- Брисање: Није дозвољено брисање „Одељења“ ако постоје „Корисници“ који имају кључ „Одељења“ као страни кључ, а „Корисник“ се може брисати без ограничења.

Преостале везе у овом информационом систему за управљање документацијом су неидентификујућег типа, али са егзистенцијалном зависношћу ентитета, што значи да није дозвољена недостајућа (NULL) вредност пренесеног кључа. Као и код претходног типа, примарни кључ ентитета „родитељ“ постаје страни кључ ентитета „дете“, али не као део примарног кључа. Везе које су преостале за дефинисање референцијалног интегритета су: „Радно место – Корисник“, „Корисник – Надлежна особа“, „Корисник – Документ“, „Надлежна особа – Документ“, „Надлежна особа – Тип документа“ и последња је „Тип документа – Документ“.

За везу „Радно место – Корисник“ референцијални интегритет за операције дефинисан је на следећи начин:

- Убацивање: „Радно место“ се може уносити без ограничења, а „Корисник“ се не може унети без „Радног места“;
- Ажурирање: Каскадно се ажурира „Корисник“ ако се измени „Радно место“, а „Корисник“ се може изменити без ограничења, али без брисања страног кључа који представља „Радно место“;
- Брисање: „Радно место“ се не може обрисати, ако постоји „Корисник“ са страним кључем „Радног места“, а „Корисник“ се може брисати без ограничења.

Веза „Корисник – Надлежна особа“ поседује следећи референцијални интегритет при извођењу операција:

- Убацивање: „Корисник“ се убацивати без ограничења, а „Надлежна особа“ се не може убацити без „Корисника“;
- Ажурирање: Каскадно се ажурира „Надлежна особа“ када се измени „Корисник“, а „Надлежна особа“ се такође може ажурирати без ограничења, али се не може уклонити страни кључ „Корисника“;
- Брисање: „Корисник“ се не може обрисати, ако постоји као страни кључ код неке „Надлежне особе“, а „Надлежна особа“ се може брисати без ограничења.

Референцијални интегритет везе „Корисник – Документ“ при извођењу операција дефинисан је као:

- Убацивање: „Корисник“ се може унети без ограничења, а „Документ“ се не може унети без „Корисника“;
- Ажурирање: Каскадно се ажурира „Документ“ приликом измене „Корисника“, а „Документ“ се може ажурирати без ограничења, али без брисања страног кључа који представља „Корисника“;
- Брисање: Када се обриже „Корисник“ каскадно се бришу сви његови „Документи“, а „Документ“ се може брисати без ограничења.

Веза „Надлежна особа – Тип документа“ поседује следећи референцијални интегритет за операције:

- Убацивање: „Надлежна особа“ се уноси без ограничења, а „Тип документа“ се не може унети без дефинисања „Надлежне особе“;
- Ажурирање: Када се измени „Надлежна особа“ каскадно се ажурира „Тип документа“, а на „Типу документа“ се могу мењати све описне вредности, осим брисања „Надлежне особе“;
- Брисање: Не може се обрисати „Надлежна особа“ ако постоји „Тип документа“ за који је она надлежна, а „Тип документа“ се може брисати без ограничења.

Ентитет „Документ“ између осталих поседује и кључ ентитета „Надлежна особа“ као страни кључ. Међутим, он означава надлежну особу којој је тренутно упућен поменути документ и не дефинише се експлицитно приликом креирања документа, већ се на почетку преузима из ентитета „Тип документа“, а током животног циклуса документа се може мењати. Референцијални интегритет везе „Надлежна особа – Документ“ за извођење операције је дефинисан на следећи начин:

- Убацивање: „Надлежна особа“ се може уносити без ограничења, а ентитет „Документ“ приликом уношења добија кључ „Надлежне особе“ преко ентитета „Тип документа“;
- Ажурирање: Каскадно се ажурира ентитет „Документ“ када се измени „Надлежна особа“, „Документ“ се може ажурирати без ограничења, а „Надлежна особа“ може ажурирати „Документ“, ако га проследи другој „Надлежној особи“;
- Брисање: „Надлежна особа“ се не може обрисати, ако постоји „Документ“ који јој је упућен, а „Документ“ се може брисати без ограничења.

Последња веза је „Тип документа – Документ“ и њен референцијални интегритет за операције је дефинисан као:

- Убацивање: „Тип документа“ се уноси без ограничења, а „Документ“ није могуће унети, ако се не одабере „Тип документа“;
- Ажурирање: Када се измени „Тип документа“ каскадно се ажурирају сви „Документи“ тог типа, а „Документ“ се може ажурирати без ограничења, али није дозвољено брисање „Типа документа“;
- Брисање: Није могуће брисање „Типа документа“ ако постоји као страни кључ у неком „Документу“, а „Документ“ се може брисати без ограничења.

Идентификација пословног домена

Сви атрибути узимају вредности из одређеног скупа могућих вредности који се назива домен и може се дефинисати као базни и типски домен. Под базним доменима се подразумевају типови података као што су карактер, цео или реалан број, низ карактера и слично и постоје операције које су дефинисане над овим доменима. Типски домен се дефинише преко свог имена, базног домена или ограничења која су дефинисана у домену. Постоје проста ограничења (релациони оператори, скуп дозвољених вредности, опсег дозвољених вредности, „NOT NULL“ вредности) и сложена ограничења (проста ограничења повезана логичким операторима).

Све шифре у систему ће бити целобројног типа, називи ће бити текстуалног типа са максималном дужином од 50 карактера, код „Корисника“ ће поља за име, презиме, звање, ЈМБГ, мејл и лозинку бити текстуалног типа са максимално дужином од 30 карактера, а код „Документа“ поља за тренутни статус и назив фајла, такође, текстуалног типа са максимално 20 карактера, а поље за датум креирања датумског типа.

3.3. Апликативно моделирање

Апликативно моделирање треба да омогући физичко креирање ефикасне базе података и да помогне приликом развоја апликације и одабира начина приступа подацима. Изводи се кроз следеће подактивности:

- Дефинисање физичког дизајна,
- Генерисање шеме базе података и
- Израда апликације.

На слици 23 налази се стабло активности за апликативно моделирање ⁵.



Слика 23. Стабло активности апликативног моделирања

3.3.1. Дефинисање физичког дизајна

Дефинисањем физичког дизајна преводи се претходно дефинисани логички модел у физички, тј. ентитети се преводe у табеле, атрибути у колоне, а дефинишу се и одговарајућа ограничења. Потребно је извршити следеће подактивности:

- Селектовање СУБП,
- Дефинисање табела и колоне,
- Дефинисање индекса и
- Дефинисање начина управљања подацима.

Селектовање СУБП

У овој активности потребно је дефинисати који ће систем за управљање базама података (СУБП) бити коришћен, где ће се креирати физичка шема и где ће се одредити опције које се користе за генерисање колона. За потребе апликације која ће се користити у информационом систему за управљање документацијом биће коришћен MySQL систем за управљање базама података, који ради као сервер и пружа интерфејс за приступ базама података.

Подаци се представљају типовима података који могу бити нумерички (целобројни, реални и логички), знаковни или структурирани. Структурирани типови су скупови вредности чија структура има одређени смисао и дефинишу се на следећи начин: ентитети постају табеле, атрибути колоне, инстанце редови, а у пресеку реда и колоне налази се поље. Више повезаних табела чини релациону базу података.

Системи за управљање базама података имају две основне функције. Прва је складиштење и одржавање података, а друга је контролисан приступ складиштеним подацима и приказивање података. Дакле, СУБП треба да омогући додавање и брисање редова, мењање података у бази података, претраживање базе података и издвајање конкретних података из базе.

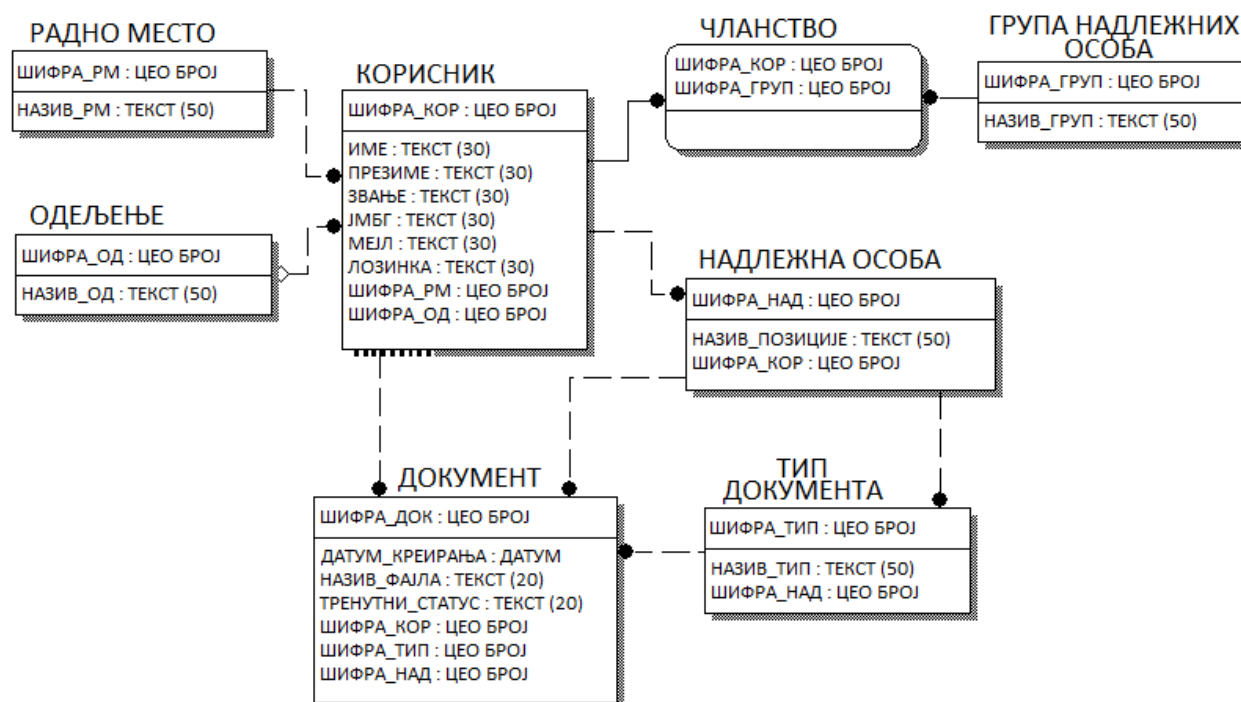
Системом за управљање базама података управља се помоћу SQL упитног језика који омогућује корисницима на веома брзо добију одговор на постављена питања. Користи се за дефинисање база података (креирање, додавање и брисање табела, погледа и индекса), претраживање података (издвајање редова из табела), манипулацију подацима (убацивање, измена и брисање редова и поља) и управљање подацима (додавање и одузимање привилегија, закључавање табела, трансакције итд.).

Дефинисање табела и колона

Потребно је дефинисати које ће све табеле постојати у бази података и које ће колоне те табеле имати, а за то ће се користити претходно дефинисани пуни атрибутивни модел система базиран на ЕР дијаграму. Дакле, табеле добијају називе на основу назива ентитета, а називи атрибута постају називи колона. Потребно је дефинисати које особине ће имати колоне, тј. подаци у њима. Особине које се дефинишу су тип података, домен, валидација и подразумевана вредност.

Што се тиче типова података и величина података, претходно је дефинисано да ће све шифре у систему бити целобројног типа (INTEGER), називи радног места, одељења, позиције надлежне особе, групе надлежних особа и типа документа бити знаковног типа, тј. текст (VARCHAR) максималне дужине 50 карактера, назив фајла и статус документа ће бити текст максималне дужине 20 карактера, а име, презиме, звање, ЈМБГ, мејл и лозинка ће такође бити текст, али максималне дужине 30 или 50 карактера. Датум креирања документа ће бити датумског (DATE) типа.

На слици 24 биће приказан физички модел података са дефинисаним типовима и величинама података за информациони систем за управљање документацијом.



Слика 24. Физички модел података система

Што се тиче домена, може се видети да подаци у систему припадају домену бројева, домену знакова или домену датума. Неки од података могу имати недостајућу вредност (NULL) и то су звање или шифра одељења код „Корисника“ и назив фајла код „Документа“.

Валидациона правила, односно ограничења вредности, дефинишу дозвољене вредности у колони. За дефинисање дозвољених вредности могу се користити оператори поређења, листа константи из одговарајућег домена или као опсег дозвољених вредности. На пример, у табели „Документ“ датум креирања не може бити различит од данашњег дана, а тренутни статус може имати вредности: „Прослеђен“, „У обради“ и „Завршен“.

Подразумеване вредности се аутоматски додељују за изабрану колону и то су данашњи датум за датум креирања, статус „Прослеђен“ за тренутни статус и шифра надлежне особе која је наведена за изабрани тип документа у табели „Документ“.

Поред наведеног, могуће је дефинисати и скупове прекомпајлираних SQL израза, тзв. тригере, који се извршавају када се појави одређени догађај. У случају овог система, користиће се тригери референцијалног интегритета, који се користе за одржавање интегритета између две повезане табеле. Одржавање интегритета подразумева одговоре на операције које су дефинисане у одељку „Дефинисање референцијалног интегритета“. Тригери се дефинишу на основу врсте правила референцијалног интегритета (RESTRICT, CASCADE...), врсте везе (идентификујућа, неидентификујућа...) и врсте улоге ентитета („родитељ“ или „дете“). Поред наведених, могу се креирати тригери или процедуре који извршавају неке одређене задатке, као што су нека израчунавања и слично. Процедуре су сличне тригерима, али се не извршавају зависно од неког догађаја, већ по позиву.

Дефинисање индекса

Зарад бржег претраживања табела користе се посебни типови табела, тзв. индекси које чувају адресе редова у табелама. Индексне табеле везане за примарне кључеве табела се креирају аутоматски, а могуће је креирати индексе и по другим колонама табела. У систему за управљање документацијом најчешће се претражује табела „Документ“, па ће остале индексне табеле бити креиране само над овом табелом и то над свим колонама.

Дефинисање начина управљања подацима

Једна од најбитнијих функција организације података је дефинисање начина управљања подацима која обухвата складиштење, поновно приступање и контролу управљања подацима. Управљање подацима подразумева:

- интегритет базе података,
- трансакциону обраду података,
- сигурност података,
- закључавање података и
- опоравак базе података.

Интегритет базе података омогућава тачност, коректност и конзистентност података и дефинише правила интегритета за одређивање услова које подаци треба да задовоље, када се врши провера и које акције се предузимају у случају незадовољавања услова. Интегритет базе је дефинисан интегритетом на серверу базе података и садржи речник података (мета-база података), интегритет домена (дефинисање скупа дозвољених вредности), интегритет табеле (сваки ред у табели мора бити јединствен), референцијални интегритет (интегритет релација), аутореференцијални интегритет (у истој табели су дефинисани спољни и оригинални кључ) и пословна правила (корисничка правила и ограничења).

Трансакциона обрада података се односи на извршење неке логичке јединице рада, тзв. трансакције, а основни циљ базе података је ефикасно обрада трансакција. Активности над базом података је атомски, тј. или су активности успешно извршене или је база података остала непромењена. Трансакције се деле на DML трансакције (повезивање више DML трансакција) и DDL трансакције (изводи се само једна наредба).

Сигурности података представља заштиту података од неовлашћеног коришћења. Корисник који је креирао табелу једини може да је користи, осим ако не дозволи приступ и другим корисницима.

Закључавање података представља механизам за регулисање конкуретног приступа подацима у бази података и може бити заједничко (истовремено заједничко закључавање за обезбеђивање упита над конзистентним подацима) или искључиво (за омогућавање ексклузивног уношења промена података).

Опоравак базе података представља повратак базе података у стање пре софтверског или хардверског отказа система. Процес опоравка базе података се изводи помоћу периодичног копирања базе на екстерну меморију, записивање промена у журнал и када дође до отказа система ради изводи се опорављање базе података (преко копије базе ако је дошло до оштећења базе података или преко журнала ако је база доведена у неконзистентно стање.

3.3.2. Генерисање шеме базе података

Шема базе података се генерише на основу претходно дефинисаног физичког дизајна, а чине је табеле, колоне и релације. Процес генерисања шеме базе података може се извести директним (генерисање шеме базе података из логичког модела) или инверзним (добивање физичке и логичке шеме података из постојеће базе података) инжењерингом. Генерисање се састоји од следећих активности:

- Креирање табела,
- Креирање индекса,
- Генерисање пословних правила и
- Верификација шеме базе података.

Креирање табела

Креирање табела се изводи „CREATE TABLE“ наредбом, при чему се креира празна табела са називом и колонама из физичког модела података. Приликом креирања, поред назива табеле наводе се називи колоне, тип података и величина података за сваку колону. Након тога се подаци могу уносити у табеле.

Креирање индекса

Индекси се креирају наредбом „CREATE INDEX“ за одређену табелу над једном или више колона. Индекс садржи по једну ставку за сваку индексирану колону и у њој чува физичке адресе редова који имају дату вредности у индексираној колони. Индекси омогућавају смањивање времена приступа тако што директно приступају редовима. Приликом креирања индекса, поред имена, дефинишу се и називи табеле и колоне које се индексирају. Препоручљиво је креирање јединствених индекса (да свака вредности у колони буде јединствена) ради побољшања перформанси и омогућавања јединствености података.

Генерисање пословних правила

Пословна правила се огледају у дефинисању ограничења за табеле и колоне. Сврха ограничења је дефинисање ранга валидних вредности, па се свака наредба проверава важећим ограничењима која морају бити задовољена да би наредба успела. Ограничењима се дефинишу домени колона, правила интегритета и референцијалног интегритета табела и додатна правила пословања. Дефинише се наредбом „CONSTRAINT“, након које се уносе име ограничења и врста ограничења за изабрану табелу или колону, као и њихови називи. Могу бити ограничења над примарним или пренесеним кључем, ограничења јединствене вредности (UNIQUE), ограничења за недозвољену недостајућу вредност (NOT NULL) или ограничење за проверу валидности неке вредности (CHECK).

Верификација шеме базе података

Шема базе података се верификује за претходно дефинисани физички модел података и пословна правила. У наредним табелама (табеле 3, 4, 5 и 6) биће приказане SQL наредбе за креирање свих табела са колонама, индексима и ограничењима.

Табела 3. Креирање табела „Корисник“ и „Чланство“ и индекса

Корисник	Чланство
<pre>CREATE TABLE `KORISNIK` (`sifra_kor` INTEGER, `ime` VARCHAR(30), `prezime` VARCHAR(30), `zvanje` VARCHAR(30), `jmbg` VARCHAR(30), `mejl` VARCHAR(30), `lozinka` VARCHAR(30), `sifra_rm` INTEGER, `sifra_od` INTEGER, CONSTRAINT `KORISNIK_PK` PRIMARY KEY (`sifra_kor`), CONSTRAINT `radi` FOREIGN KEY (`sifra_rm`) REFERENCES `RADNO_MESTO` (`sifra_rm`), CONSTRAINT `pripada` FOREIGN KEY (`sifra_od`) REFERENCES `ODELJENJE` (`sifra_od`)); CREATE UNIQUE INDEX `IND_KOR_PK` ON `KORISNIK` (`sifra_kor`);</pre>	<pre>CREATE TABLE `CLANSTVO` (`sifra_kor` INTEGER, `sifra_grup` INTEGER, CONSTRAINT `CLANSTVO_PK` PRIMARY KEY (`sifra_kor`, `sifra_grup`), CONSTRAINT `ima` FOREIGN KEY (`sifra_kor`) REFERENCES `KORISNIK` (`sifra_kor`), CONSTRAINT `odnosi_se_na` FOREIGN KEY (`sifra_grup`) REFERENCES `GRUPA_NADLEZNIH_OSOBA` (`sifra_grup`)); CREATE UNIQUE INDEX `IND_CLAN_PK` ON `CLANSTVO` (`sifra_kor`, `sifra_grup`);</pre>

Табела 4. Креирање табела „Радно место“ и „Одељење“ и индекса

Радно место	Одељење
<pre>CREATE TABLE `RADNO_MESTO` (`sifra_rm` INTEGER, `naziv_rm` VARCHAR(50), CONSTRAINT `RADNO_MESTO_PK` PRIMARY KEY (`sifra_rm`)); CREATE UNIQUE INDEX `IND_RM_PK` ON `RADNO_MESTO` (`sifra_rm`);</pre>	<pre>CREATE TABLE `ODELJENJE` (`sifra_od` INTEGER, `naziv_od` VARCHAR(50), CONSTRAINT `ODELJENJE_PK` PRIMARY KEY (`sifra_od`)); CREATE UNIQUE INDEX `IND_OD_PK` ON `ODELJENJE` (`sifra_od`);</pre>

Табела 5. Креирање табела „Документ“ и „Тип документа“ и индекса

Документ	Тип документа
<pre>CREATE TABLE `DOKUMENT` (`sifra_dok` INTEGER, `datum_kreiranja` DATE, `naziv_fajla` VARCHAR(20), `trenutni_status` VARCHAR(20), `sifra_nad` INTEGER, `sifra_kor` INTEGER, `sifra_tip` INTEGER, CONSTRAINT `DOKUMENT_PK` PRIMARY KEY (`sifra_dok`), CONSTRAINT `upucen` FOREIGN KEY (`sifra_nad`) REFERENCES `NADLEZNA_OSOBA` (`sifra_nad`), CONSTRAINT `kreira` FOREIGN KEY (`sifra_kor`) REFERENCES `KORISNIK` (`sifra_kor`), CONSTRAINT `je_tipa` FOREIGN KEY (`sifra_tip`) REFERENCES `TIP_DOKUMENTA` (`sifra_tip`)); CREATE UNIQUE INDEX `IND_DOK_PK` ON `DOKUMENT` (`sifra_dok`); CREATE INDEX `IND_DOK_DAT` ON `DOKUMENT` (`datum_kreiranja`); CREATE INDEX `IND_DOK_STAT` ON `DOKUMENT` (`trenutni_status`); CREATE INDEX `IND_DOK_NAD` ON `DOKUMENT` (`sifra_nad`); CREATE INDEX `IND_DOK_KOR` ON `DOKUMENT` (`sifra_kor`); CREATE INDEX `IND_DOK_TIP` ON `DOKUMENT` (`sifra_tip`);</pre>	<pre>CREATE TABLE `TIP_DOKUMENTA` (`sifra_tip` INTEGER, `naziv_tip` VARCHAR(50), `sifra_nad` INTEGER, CONSTRAINT `TIP_DOKUMENTA_PK` PRIMARY KEY (`sifra_tip`), CONSTRAINT `zaduzen_za` FOREIGN KEY (`sifra_nad`) REFERENCES `NADLEZNA_OSOBA` (`sifra_nad`)); CREATE UNIQUE INDEX `IND_TIP_PK` ON `TIP_DOKUMENTA` (`sifra_tip`);</pre>

Табела 6. Креирање табела „Група надлежних особа“ и „Надлежна особа“ и индекса

Група надлежних особа	Надлежна особа
<pre>CREATE TABLE `GRUPA_NADLEZNIH_OSOBA` (`sifra_grup` INTEGER, `naziv_grup` VARCHAR(50), CONSTRAINT `GRUPA_NO_PK` PRIMARY KEY (`sifra_grup`)); CREATE UNIQUE INDEX `IND_GRUP_PK` ON `GRUPA_NADLEZNIH_OSOBA` (`sifra_grup`);</pre>	<pre>CREATE TABLE `NADLEZNA_OSOBA` (`sifra_nad` INTEGER, `naziv_pozicije` VARCHAR(50), `sifra_kor` INTEGER, CONSTRAINT `NADLEZNA_OSOBA_PK` PRIMARY KEY (`sifra_nad`), CONSTRAINT `je` FOREIGN KEY (`sifra_kor`) REFERENCES `KORISNIK`(`sifra_kor`)); CREATE UNIQUE INDEX `IND_NAD_PK` ON `NADLEZNA_OSOBA` (`sifra_nad`);</pre>

3.3.3. Израда апликације

Апликација се израђује на основу претходно дефинисане шеме базе података и захтева корисника. Детаљан опис форме изводи се кроз следеће активности:

- Дефинисање менија,
- Дефинисање изгледа форме,
- Дефинисање упита и
- Дефинисање извештаја.

Израда апликације је везана за клијент - сервер архитектуру, где се на серверу налазе базе података и СУБП, а на клијентској страни се дефинише апликација.

Клијентска апликација поседује интерфејс са кориснику (GUI) и преко ње корисници уносе, претражују и анализирају податке. Обухвата унос података и директну трансакциону обраду коришћењем форми и израду упита и израду извештаја алатима за развој апликације. Унос података преко форми треба да омогући аутоматску проверу интегритета. Директна трансакциона обрада подразумева операције за додавање и ажурирање података у бази. Упити и извештаји се користе за дефинисање извештаја са више колона, записа и графичких извештаја.

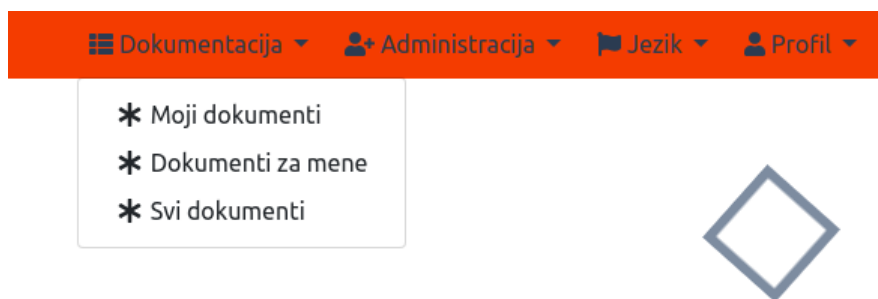
Што се тиче алата за развој апликације, у овом случају биће коришћени програмски језик Јава (за део апликације који се извршава на серверу) и Ангулар радно окружење (за део апликације који се извршава код клијента). Јава је програмски језик опште намене, објектно – оријентисан, дизајниран тако да се апликације израђене у њему могу покретати на свим платформама. Ангулар је радно окружење (енгл. Framework) отвореног кода за израду веб апликација и базиран је на Тајпскрипт (енгл. TypeScript) програмском језику.

Дефинисање менија

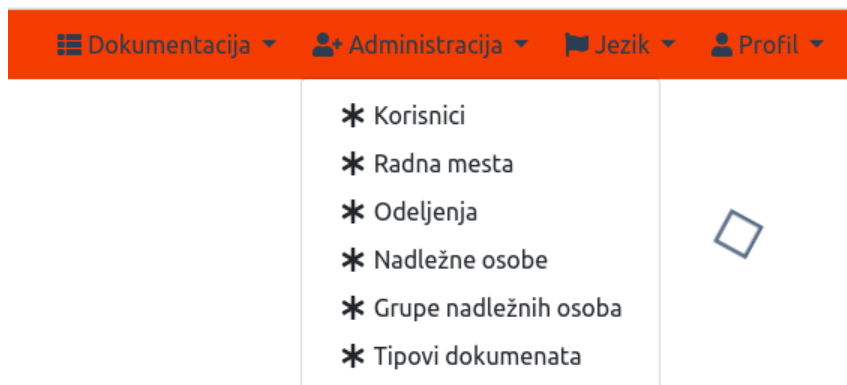
Менији се користе за бржу и лакшу навигацију кроз веб апликацију и треба да поштују одређена правила структурирања, како би указивали на могући редослед позивања операција. Нека од правила структурирања која један мени треба да поштује су: да садржи концизан наслов, да раздваја листе опција на целине, да омогући напуштање менија без избора било које опције, да користи адекватне именице за опис опција и слично.

У апликацији информационог система за управљање документацијом мени ће се увек налазити на врху стране, а избор опција ће зависити од привилегија корисника, тј. обичан корисник ће моћи да види само своје документе, надлежна особа ће поред својих моћи да види и документе који су упућени њој, а надлежна особа са одређеним статусом (нпр. декан факултета) ће, поред претходно наведених, моћи да види документе свих корисника. Администратор система ће поред докумената моћи да види и остале опције (кориснике, радна места, одељења...) и да их креира, модификује и брише.

На сликама 25 и 26 биће приказан мени са отвореним падајућим менијима за избор докумената и шифарника.



Слика 25. Мени са отвореним падајућим менијем за избор докумената



Слика 26. Мени са отвореним падајућим менијем за избор шифарника

Дефинисање изгледа форме

Форме се користе за приказивање података из база података. Као и менији, форме морају поштовати правила структурирања како би била прегледнија и лакша за коришћење. Неке од карактеристика које форме треба да испоштују су: да имају наслов, да буду симетричне и балансиране, да кориснику прикажу само потребне информације, да се колоне имају називе, да се омогући кориснику додавање, измена и брисање редова и слично.

На сликама 27 и 28 налазе се форме за приказивање свих докумената и унос или измену докумената.

Datum kreiranja	Tip dokumenta	Trenutni status	Komentar	Datum obrade	Trenutno nadležan	Korisnik
15.10.2019.	Zahtev	PRIJAVLJEN			Sef katedre	Stefan Nikolic
16.10.2019.	Molba	U OBRADI		21.10.2019.	Prodekan	Stefan Nikolic

Prikazuje se 1 - 2 of 2 elemenata.

Слика 27. Форма за приказ свих докумената

Kreirajte ili izmenite Dokument

Datum kreiranja

Naziv fajla

Tip dokumenta

Otkazi Sačuvaj

Слика 28. Форма за унос и измену докумената

Дефинисање упита

За позивање и добијање жељених података из базе података користе се SQL упити (QUERY) који представљају сет заједничких команди и функција. Упити се могу извршавати над једном или више табела, а извршавају се коришћењем „SELECT“ наредбе. Такође, постоје операције за одржавање база података, као што су уношење (наредба „INSERT“), ажурирање (наредба „UPDATE“) и брисање (наредба „DELETE“). Поред наведених постоје и операције за измену самих табела у бази, као што су брисање табеле (наредба „DROP TABLE“) и измена табеле (наредба „ALTER TABLE“).

За дефинисање упита у инфомационом систему за управљање документацијом биће коришћен пример корисника са шифром 25 који има највише привилегије, тј. надлежна је особа и може приступати свим документима корисника. Упити за добијање докумената (личних, упућених и свих) су:

- `SELECT * FROM DOKUMENT WHERE SIFRA_KOR = 25;`
- `SELECT * FROM DOKUMENT WHERE SIFRA_NAD = 25;`
- `SELECT * FROM DOKUMENT;`

Потребно је дефинисати и упите за уношење, ажурирање и брисање докумената. Прво ће, као пример, бити приказан упит за уношење новог документа, а након тога упити за ажурирање и брисање тог документа. Упити су:

- `INSERT INTO DOKUMENT (SIFRA_DOK, DATUM_KREIRANJA, NAZIV_FAJLA, TRENUTNI_STATUS, SIFRA_NAD, SIFRA_KOR, SIFRA_TIP) VALUES (35, CURDATE( ), 'FAJL.TXT', 'PRIJAVLJEN', 25, 11, 2);`
- `UPDATE DOKUMENT SET TRENUTNI_STATUS = 'U OBRADI', DATUM_OBRADE = CURDATE( ) WHERE SIFRA_DOK = 35;`
- `DELETE FROM DOKUMENT WHERE SIFRA_DOK = 35;`

Дефинисање извештаја

Дефинисање извештаја представља добру праксу приликом развоја информационог система, јер кориснику даје могућност да своје податке добије у електронској форми, коју касније може пренети на папир. Поред приказивања самих података из базе података, на извештајима је могуће приказати податке настале као резултат разних израчунавања, као и разне дијаграме који графички приказују податке.

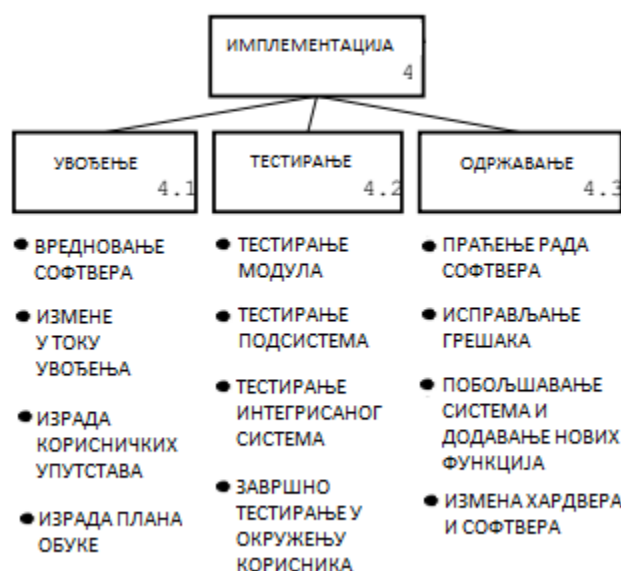
У овом случају, поред извештаја свих докумената једног корисника, могуће је приказати и извештаје попут просечног времена обраде документа за надлежну особу, графички приказ броја докумената на годишњем нивоу и слично.

3.4. Имплементација

Током активности „Имплементација“ потребно је, уз помоћ консултанта, радити на увођењу, тестирању и одржавању информационог система и корисничког софтвера. Изводи се на основу претходно реализаног апликативног моделирања и захтева корисника, а коначан производ је кориснички софтвер. Дакле, имплементација се изводи кроз следеће активности:

- Увођење,
- Тестирање и
- Одржавање.

На слици 29 приказано је стабло активности за активност „Имплементација“⁵.



Слика 29. Стабло активности за активност „Имплементација“

3.4.1. Увођење

Активности „Увођење“ изводи се кроз следеће активности:

- Вредновање софтвера,
- Измене у току увођења,
- Израда корисничких упутстава и
- Израда плана обуке.

Вредновање софтвера

За вредновање софтвера потребно је дефинисати основне захтеве за функције и перформансе софтвера и дефинисати ограничења. Постиже се помоћу фактора и метрике квалитета, а као резултат вредновања добија се оцена. Захтеви вредновања се исказују преко функционалних захтева, садржаја коришћења, броја корисника итд.

Постоје дефинисани стандарди који се поштују приликом вредновања софтвера и могу се посматрати кроз следеће процесе:

- Анализа захтева вредновања (дефинисање захтева кроз интервју, анкету или упитник);
- Спецификација вредновања (избор метрике и карактеристика, дефинисање ранга и критеријума оцењивања);
- Пројектовање вредновања (спецификација и уградња модула вредновања);
- Примена вредновања (мерење, оцењивање и извештавање мерења);
- Извештавање о вредновању (анализа резултата и генерисање извештаја).

Измене у току увођења

Приликом увођења могу се уочити неке неправилности које нису уочене у току развоја информационог система, а могуће је наићи и на примедбе корисника, па је потребно извршити неке измене. Свака измена би требала бити документована у табелама базе података или у електронској документацији.

Израда корисничких упутстава

Корисничка упутства представљају општа упутства за коришћење апликације на свим оперативним системима. Поред папирне форме, пожељно је да постоји и у електронском облику или у облику интернет документације. Корисничка упутства треба да имају следеће карактеристике:

- неопходни су јасни и концизни изрази,
- без граматичких и словних грешака,
- процедуре описане логичким редом,
- могућност једноставног избора итд.

Израда плана обуке

Након претходно наведеног, потребно је изградити план обуке корисника. Потребно је узети у обзир рачунарску описмењеност корисника, јер ће од ње зависити дужина обуке. План се креира по приоритетима увођења модула или подсистема.

3.4.2. Тестирање

Тестирање подразумева оцењивање одлика и перформанси система и ревидирање (корекцију) система како би се прилагодио кориснику. Провера се обавља кроз валидацију и верификацију. Валидација служи за проверу да ли производ задовољава потребе корисника, а верификација за проверу да ли је производ направљен правилно. Постоје функционално тестирање (тестирање функционалне спецификације – техника црне кутије) и структурно тестирање производа (провера програмских структура, токова података и др. – техника беле кутије). Тестирање се изводи кроз следеће активности:

- Тестирање модула,
- Тестирање подсистема,
- Тестирање интегрисаног система и
- Завршно тестирање у окружењу корисника.

Тестирање модула

На почетку је потребно проверити све спецификације и конструкције везане за модул, јер је низ независних компоненти повезан са модулом. Претходно је потребно тестирати и сваку од тих независних компоненти појединачно.

Тестирање подсистема

Проблеми се најчешће појављују приликом тестирања подсистема, нарочито ако је више пројектаната укључено у пројекат, због различитих интерпретација спецификација. Потребно је уклопити те спецификације, а то се постиже спајањем модула, дефинисањем интерних интерфејса, тестирањем рада групе модула и тестирањем везе са екстерним интерфејсима.

Тестирање интегрисаног система

Након тестирања подсистема, потребно је спојити их у једну целину. Приликом тестирања интегрисаног система, потребно је да корисник обезбеди реалне податке за употребу, који ће се користити за тестирање. Тада се тестира цео систем, тј. његова функционалност, перформансе, рестарт, опоравак и функционисање.

Завршно тестирање у окружењу корисника

Завршно тестирање подразумева дефинисање захтева спецификација и испуњење услова преузимања. Ако постоји само један корисник назива се алфа – тестирање, а ако постоји више корисника, тј. ако се производ дистрибуира на више места или више појединаца, онда се назива бета – тестирање.

3.4.3. Одржавање

У току одржавања све мане и неправилности система излазе на видело због лоше реализованих претходних фаза. Основни проблеми су непридржавање стандарда, лоша документација, непостојање тестова, некоректно одржавање итд. Активност „Одржавање“ чине следеће активности:

- Праћење рада,
- Исправљање грешака,
- Побољшање система и додавање нових функција и
- Измена хардвера и софтвера.

Праћење рада

Праћење треба изводити континуирано, јер у случају да је потребно извести замену, информације стечене у току праћења помажу у одређивању врсте промене. Могу се пратити време употребе (у сатима), урађени послови и трансакције, информације о времену, учитавања, регистровање недостатака и др. Ако се прати време и примете варијације у времену извршавања трансакције, могуће је на време извршити корекцију и спречити отказ система.

Исправљање грешака

Исправљање грешака се изводи уз помоћ претходно описаног праћења рада, а поред исправљања прати се и одступање софтвера и врши прилагођавање кориснику и задатку.

Побољшање система и додавање нових функција

Активност побољшања система и додавања нових функција је углавном незаобилазан део имплементације, јер корисници у току коришћења увиде нове могућности, па се јављају нови захтеви за побољшање система. Ако су претходне фазе правилно спроведене, побољшавање система не би требало да буде проблем.

Измена хардвера и софтвера

Хардверске измене се углавном односе на надградњу мреже у клијент – сервер архитектури. Измена може бити у топологији мреже, у хардверу и софтверу за умрежени клијент – сервер систем, у апликационом посреднику за рад са базом података и у активности у систему клијент – сервер.

Софтверске измене се углавном односе на базе података и апликације које се темеље на њима. Разлози за промене могу бити прелазак на неки новији хардвер, промена у захтевима корисника, развој апликације, промена услова и правила пословања итд.

4. Закључак

Праћењем и правилном реализацијом свих претходно наведених активности могуће је креирати стабилан и поуздан информациони систем, који се лако може надоградити. На примеру информационог система за управљање документацијом приказано је које све фазе развоја мора извршити један информациони систем.

Информациони систем пројектован у раду обухвата само неке од основних функција управљања документацијом. Један од начина унапређења система био би омогућавање кориснику да креира садржај документа, према стандардима креирања документације, јер се у тренутном систему сматра да корисник већ поседује документ у електронском облику. Такође, кориснику се може омогућити ковертовање фајлова, отварање фајлова директно из апликације, комуникацију између корисника и надлежних особа и слично. Поред наведеног, може се унапредити безбедност и заштита система и омогућити поверљива размена докумената.

Имплементација оваквог система није тешка, јер се базира на клијент – сервер архитектури, где се апликација са базом података налази на серверу, па кориснику није потребно ништа осим рачунара који има приступ интернету да би користио апликацију. Корисници имају различит ниво приступа апликацији, у зависности од улоге који имају у систему. Апликација је прегледна и веома лака за коришћење, тако да сваки корисник може, за веома кратко време, да овлада коришћењем апликације.

Управљање документацијом је једна од активности са којом се све компаније и институције сусрећу и у многима од њих се јављају проблеми нагомилане или изгубљене документације. Коришћењем информационог система за управљање документацијом решава се већина тих проблема и добијају се неке могућности које раније нису постојале, као што је праћење статуса докумената, концепт сличан праћењу пошиљки послатих курирским службама. Поред тога, олакшано је претраживање докумената, као и само прослеђивање докумената, које корисник, овим путем, може послати са свог рачунара, без потребе за одласком до особе надлежне за документацију.

5. Литература

1. *Data to Wisdom Via Information, Knowledge & Insight*. (20. јун 2019). Преузето 16. септембра 2019, са <https://www.theifactory.com/news/gaining-wisdom-from-data/>
2. Полишчук, Ј. (септембар 2007). *Пројектовање информационих система*. Електротехнички факултет Подгорица, Црна Гора
3. Ерић, М. (2018). *Пројектовање информационих система и база података* [PDF]. Преузето 09. октобра 2018, са https://drive.google.com/drive/folders/1gvcBjQopMezrtq2pYCpPx8Tb_4m7lPD0?fbclid=IwAR3t9dWwdlILsdD4RxCr6hBHj-w2cWkQzX-rhMkeammn_CMHtwkLk4gbs10
4. *Document Routing and Approval*. (б.д.). Преузето 16. септембра 2019, са <https://www.integrify.com/document-routing-and-approval/>
5. Вељовић, А. (б.д.). *Развој информационих система и базе података*.