

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Antanas Gilaitis

**Informacijos paieškos sistemos elektroninėje
bibliotekoje (eLABa) projektavimas ir tyrimas**

Magistro darbas

Darbo vadovas
prof. E. Bareiša

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Antanas Gilaitis

**Informacijos paieškos sistemos elektroninėje
bibliotekoje (eLABa) projektavimas ir tyrimas**

Magistro darbas

Recenzentas

2009-05-22

doc. dr. A. Lenkevičius

Vadovas

prof. E. Bareiša
2009-05-25

Atliko

2009-05-22

IFM-3/2 gr. stud.
Antanas Gilaitis

TURINYS

1. ĮVADAS	8
2. ANALITINĖ DALIS	9
2.1. Problema.....	9
2.2. Egzistuojantys sprendimai.....	9
2.2.1. Stokholmo technikos universiteto sprendimas.....	9
2.2.2. FEDORA realizacija.....	11
2.2.3. Egzistuojančių sistemų apibendrinimas.....	12
2.3. Projektavimo metodologija.....	12
2.3.1. Indekso kūrimas ir redagavimas.....	13
2.3.2. Paieškos vykdymas.....	13
2.3.3. Rezultatų pateikimas.....	13
2.3.4. Paieškos istorijos peržiūrėjimas.....	13
2.4. Projektavimo technologija.....	13
2.4.1. Apache Lucene paieškos variklis.....	13
2.4.2. Zebra paieškos variklis.....	14
2.4.3. JBoss Seam Framework.....	15
2.4.4. Apache Axis.....	15
2.5. Įgyvendinimo problemos.....	15
2.5.1. Indekso dydis.....	15
2.5.2. Specifinių raidžių atpažinimas.....	16
2.6. Išvados.....	16
3. PROJEKTINĖ DALIS	17
3.1. Sistemos reikalavimai.....	17
3.1.1. Sistemos tikslai.....	17
3.1.2. Vartotojai.....	17
3.1.3. Diegimo aplinka.....	17
3.1.4. Veiklos kontekstas.....	18
3.1.5. Sistemos panaudos atvejai.....	19
3.1.6. Funkciniai reikalavimai.....	20
3.1.7. Nefunkciniai reikalavimai.....	20
3.2. Sistemos architektūra.....	21
3.2.1. Sistemos veikimo principas.....	21
3.2.2. Sistemos sudėtis.....	21
3.2.3. Duomenų bazė.....	23
3.2.4. Duomenų srautų diagramos (DSD).....	25
3.3. Testavimas.....	26
3.3.1. Testavimo tikslai ir apimtis.....	26
3.3.2. Sąsajos testavimas.....	26
3.3.3. Testavimo strategijos.....	27
3.3.4. Vienetų testavimas.....	27
3.3.5. Integravimo testavimas.....	29
3.3.6. Priėmimo testavimas.....	29
3.3.7. Testavimo rezultatų kaupimas.....	29
3.3.8. Testavimo išvados.....	29
3.4. Kokybės analizė.....	30
3.4.1. Kokybės vertinimo tikslai.....	30

3.4.2.	Kokybės vertinimo kriterijai	30
3.4.3.	Kokybės užtikrinimo ir vertinimo procesas	31
3.4.4.	Kokybės vertinimas	32
3.4.5.	Kokybės vertinimo rezultatai	33
3.4.6.	Išvados	34
4.	TIRIAMOJI EKSPERIMENTINĖ DALIS	35
4.1.	Tikslas	35
4.2.	Tyrimo metodika	35
4.3.	Rezultatai.....	35
4.4.	Išvados	40
5.	IŠVADOS	41
6.	LITERATŪRA.....	42
7.	TERMINŲ IR SANTRUMPŲ ŽODYNAS	43
8.	PRIEDAI	44
1.	PRIEDAS. SISTEMOS ĮDIEGIMO PATVIRTINIMO AKTAS.....	44

PAVEIKSLAI

1 Pav. Paieškos sistemos lango pavyzdys.....	9
2 Pav. Paieškos rezultatų langas	10
3 Pav. Dokumento detalios informacijos langas.....	10
4 Pav. FEDORA paieškos langas	11
5 Pav. FEDORA rezultatų langas	11
6 Pav. Konteksto diagrama	18
7 Pav. Panaudos atvejų diagrama	19
8 Pav. Sistemos veikimo principas	21
9 Pav. Sistemos struktūra.....	22
10 Pav. Bendroji indeksavimo posistemės veikimo schema	22
11 Pav. Bendroji paieškos posistemės veikimo schema	23
12 pav. ER diagrama.....	23
13 pav. Indeksavimo DSD	25
14 pav. Paieškos DSD.....	25
15 Pav. Paieškos laiko priklausomybės nuo indekso dydžio diagrama.....	36
16 Pav. Indekso dydžio priklausomybės nuo dokumentų dydžio diagrama.....	37
17 Pav. Indeksavimo laiko priklausomybės nuo dokumentų kiekio diagrama.....	38
18 Pav. Paieškos laiko priklausomybės nuo dokumentų kiekio,paieškai naudojant optimizuotą ir neoptimizuotą indeksą, diagrama	39

LENTELĖS

Lentele Nr. 1 Duomenų bazės lentelės „USER“ specifikacija	24
Lentele Nr. 2 Duomenų bazės lentelės „RESULTS“ specifikacija	24
Lentele Nr. 3 Duomenų bazės lentelės „HISTORY“ specifikacija	24
Lentele Nr. 4 Vienetų testavimo rezultatai	28
Lentele Nr. 5 Kokybės vertinimo rezultatai.....	34
Lentele Nr. 6 Paieškos laiko priklausomybės nuo indekso dydžio eksperimento rezultatai	35
Lentele Nr. 7 Indekso dydžio priklausomybės nuo saugomų dokumentų dydžio eksperimento rezultatai.....	36
Lentele Nr. 8 Dokumentų indeksavimo laiko priklausomybės nuo indeksuojamų dokumentų kiekio eksperimento rezultatai	37
Lentele Nr. 9 Paieškos laiko priklausomybės nuo dokumentų kiekio, paieškai naudojant optimizuotą ir neoptimizuotą indeksą, eksperimento rezultatai	39

Information search engine in e-library (eLABa) design and analysis

SUMMARY

This system is invented to help people search for the e. documents in FEDORA repository. The system has following features: indexing of Fedora FOXML records, including the text contents of data streams and search in the index. There is possibility to registered users to save results in the history page, and to do repeated search if you want. Also you can go directly to e. document after you have made search.

1. ĮVADAS

Atskiras Kauno technologijos universiteto (KTU) Informatikos fakulteto Informacinių technologijų diegimo centro (ITDC) Bibliotekų programinės įrangos skyrius (BPIS) atlieka keletą pagrindinių funkcijų ir yra atsakingas už:

- Universiteto bibliotekose esančias IT, jų diegimą ir priežiūrą
- Projektų vykdymą kuriant Lietuvos virtualią biblioteką (LVB)

Vienas tokių projektų yra Lietuvos mokslo ir studijų elektroninių dokumentų informacinė sistema (eLABa) skirta Lietuvos mokslo institucijoms, kurių autoriai kuria mokslines publikacijas ir kurias reikia įkelti, saugoti, kataloguoti ir pateikti skaitytojams bei kitoms sistemoms.

eLABa atlieka elektroninių (toliau e.) dokumentų leidybos, pateikimo į e. bibliotekos visateksčių dokumentų duomenų bazės (VDDDB) talpyklas, metaduomenų užpildymo ir tikrinimo, visų duomenų saugojimo, indeksavimo (teksto ir metaduomenų) paieškai ir duomenų pateikimo skaitytojams funkcijas.

Kuriamas produktas nėra skirtas pardavimui. Jis skirtas mokslo institucijų darbuotojams, mokslininkams bei visiems žmonėms atlikti paieškai tarp mokslo publikacijų, straipsnių, elektroninių tezių ir disertacijų, mokslo ataskaitų, žurnalų ir kitų e. leidinių.

Padidėjusi informacijos apie mokslinius tyrimus sklaida ir pasiekiamumas per eLABa sistemą skatins inicijuoti naujus mokslinius tyrimus jau atliktų pagrindu, kurti naujas ar patobulintas teorijas, technologijas ar produktus. Taip pat tai leis mokslinių tyrimų rezultatus plačiau pritaikyti praktikoje ir naudoti verslo sektoriuje, kurti naujus ar patobulintus produktus, pagrįstus mokslinių tyrimų rezultatais, naujomis technologijomis ir inovacijomis.[2]

2. ANALITINĖ DALIS

2.1. Problema

Sukurti informacijos paieškos ir pateikimo posistemę, apimančią į eLABa įkeltų duomenų paieškos indeksų kūrimą ir tvarkymą, patogios informacijos paieškos ir rezultatų pateikimo aplinkos formavimą bei įvairių elektroninių (virtualių) paslaugų, susijusių su ieškomais dokumentais, teikimą.

2.2. Egzistuojantys sprendimai

2.2.1. Stokholmo technikos universiteto sprendimas

Pasaulyje jau yra sukurta panašių sistemų. Kaip pavyzdį, galime paimti jau sukurta Stokholmo technikos universiteto elektroninių tezių ir disertacijų paieškos sistemą. Jos bendras vaizdas pavaizduotas 1 paveikslėlyje [6]

The screenshot shows the KTH Biblioteket search interface. At the top, there are navigation links: 'Sök', 'Innehåll', and 'Kontakt'. The main header includes the KTH logo and the text 'KTH Biblioteket'. Below this, the page title is 'Publications from KTH'. The search form is titled 'Start Page/Search' and includes a section for 'Publication type' with checkboxes for 'Articles (0)', 'Theses (3744)', 'Master theses / Undergraduate theses', 'Reports (4)', 'Chapters (0)', and 'Other (0)'. The form also has input fields for 'Author' (last and first name), 'Title' (or a word from the title), 'Series' (number), 'Year' (YYYY), and 'ISBN' (include hyphens). A 'Department' dropdown menu is set to 'Select department:'. The 'Sort by' section has dropdowns for 'Author', 'Ascending', and '20' (number of displayed hits per page). There are 'Search' and 'Reset' buttons at the bottom. A 'Get Acrobat Reader' logo is visible on the left sidebar.

1 Pav. Paieškos sistemos lango pavyzdys

Tai yra pagrindinis paieškos langas kur įvedę užklausą ir pasirinkę kokiuose laukuose ieškoti gauname rezultatų langą. Paieškos rezultatų langą įvedus užklausą „elektronis“ matome 2 paveikslėlyje:

Publikationer

KTH / Biblioteket / Publikationer

Startsida/sök
Avhandlingar
Examensarbeten / Uppsatser
Rapport

Get Acrobat Reader

Publikationer från KTH

Sökresultat

Sökfråga: **electronic** (Titel)

Antal träffar: 28, visar 1 - 20 [nästa sida](#)

Författare	Titel	Publikationstyp	År	Fulltext
An, Wei	Electronic speckle pattern interferometry applied to shape and deformation measurement	Licentiatavhandling 2000		Endast tryckt
Andersson, David	From the Electronic Structure of Point Defects to Functional Properties of Metals and Ceramics	Doktorsavhandling	2007	PDF [1070 kb]
Armiento, Rickard	The many-electron energy in density functional theory: from exchange-correlation functional design to applied	Doktorsavhandling	2005	PDF [908 kb]

Aktyvi nuoroda

2 Pav. Paieškos rezultatų langas

Kaip matome išvestas gautų rezultatų sąrašas kur paspaudus aktyvią nuorodą galima nueiti į platesnį to įrašo aprašymą ir jei mus ta informacija tenkina galima parsisiųsti elektroninį dokumentą paspaudus nuorodą. (3 pav.)

Sök Innehåll Kontakt

KTH Biblioteket

Publikationer

KTH / Biblioteket / Publikationer

Publikationer från KTH

Full beskrivning

Filer: **Beskrivning** **Filstorlek** **Format**

[Fulltext](#) 1070Kb PDF (kräver [Acrobat Reader](#))

Författare: Andersson, David

Titel: From the Electronic Structure of Point Defects to Functional Properties of Metals and Ceramics

Institution: KTH, Materialvetenskap

URI: urn:nbn:se:kth:diva-4309

Publikationstyp: Doktorsavhandling

Språk: Engelska [en] med sammanfattning på Engelska [en]

Nyckelord: first principles, ab initio, density functional theory, Calphad, point defects, diffusion, solid electrolytes, oxygen storage materials

Disputation: 2007-03-30, F2, Lindstedtsvägen 26, Stockholm, 10:00, Engelska

Opponent: Burton, Benjamin, Dr, National Institute of Standards and Technology, USA

Abstract [en]: Point defects are an inherent part of crystalline materials and they influence important physical and chemical properties, such as diffusion, hardness, catalytic activity and phase stability. Increased understanding of point defects enables us to tailor the defect-related properties to the application at hand. Modeling and simulation have a prominent role in acquiring this knowledge. In this thesis

Nuoroda dokumento parsisiuntimui

3 Pav. Dokumento detalios informacijos langas

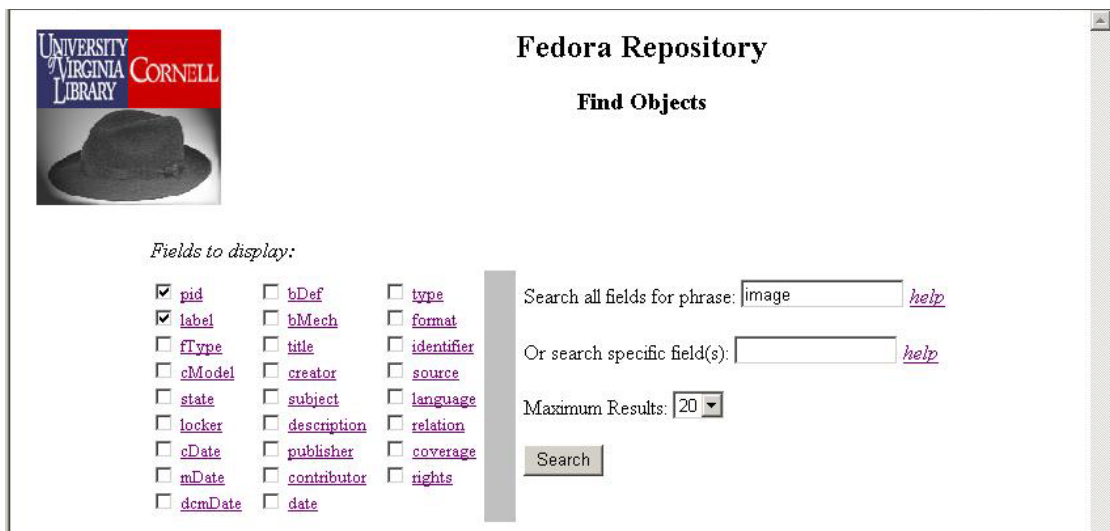
Panašiai veiks ir eLABa paieškos sistema. Tik joje bus daugiau funkcijų tokių kaip paieškos istorijos registravimas, bei paieškos rezultatų išsaugojimas.

2.2.2. FEDORA realizacija

Paieškai FEDORA talpykloje atlikti būtų galima naudoti ir jų pačių sukurtą paieškos sistemą, bet ji yra visiškai nepatogi. Ji taip pat kaip ir daugelis panašių paieškos sistemų informacijos ieško tik tarp dokumento metaduomenų.

Paieškai vykdyti į paieškos langą (4 pav.) reikia įvesti ieškomą tekstą bei pažymėti kuriuos metaduomenų laukus išvesti. Tai paprastam vartotojui nėra visiškai suprantama ir patogiu, kadangi jis jau turi iš anksto žinoti ką koks lauko pavadinimas reiškia ir kokia informacija į jį vedama.

Norint atlikti sudėtingesnę užklausą reikia pačiam įrašyti nemažai specifinių išraiškų, kas yra labai nepatogu. Pavyzdžiui, norint ieškoti dokumentų sukurtų kažkokio autoriaus kurio pavardėje yra raidė „n“ ir jie sukurti 2005 metais, reiktų suformuluoti tokią užklausą: cDate:2005 creator~*n*, o tai nėra patogu.



4 Pav. FEDORA paieškos langas

Atlikus paiešką, į ekraną išvedami rezultatai. (5 pav.) Aiškiai matome, kad rezultatai nėra pateikiami vaizdžiai. [3]

pid	label
demo:5	Image of Coliseum in Rome
demo:29	Image of Coliseum in Rome
demo:7	Image of Architectural Drawings for Pavillion III, University of Virginia
demo:6	Pavillion III, IVA Image Collection - University of Virginia
demo:11	Exhibit Intro: Architectural drawings, Pavillion III, IVA Image Collection - University of Virginia

5 Pav. FEDORA rezultatų langas

2.2.3. Egzistuojančių sistemų apibendrinimas

Išanalizavus ir įvertinus panašias paieškos sistemas galime išskirti dvi pagrindines teikiamas funkcijas:

- Paieškos vykdymas
- Rezultatų pateikimas

Iš analizuotų paieškos sistemų realizacijų pavyzdžių galime daryti išvadą, kad nei viena iš jų nėra visiškai patogi vartojimui ir nėra pilnai tenkinamas funkcionalumas. Jos nėra patogios vykdyti paieškai, kadangi nėra iki galo realizuotos, bei turi per mažai funkcijų. Taip pat turime pabrėžti, kad paieška vykdoma tik tarp dokumento metaduomenų t.y. autoriaus, pavadinimo, išleidimo metų, antraštės ir pan. o mums reikia paieškos ir visame dokumente.

2.3. Projektavimo metodologija

Kuriamai paieškos sistemai be pagrindinių teikiamų funkcijų galime išskirti dar kelias svarbias funkcijas tam kad galėtume vykdyti pilnojo teksto (full text) paiešką t.y. paieškos indekso kūrimas ir redagavimas. Tada sistemos funkcionalumas atrodytu taip:

- Paieškos vykdymas
- Rezultatų pateikimas
- Indekso kūrimas ir redagavimas
- Paieškos istorijos peržiūrėjimas

Siūlomas sprendimas indeksavimui ir paieškai panaudoti Lucene arba Zebra paieškos variklius. Juos pritaikius galima indeksuoti FEDORA FOXML formato įrašus bei dokumentus, bei atlikti pilnojo teksto paiešką.

Per SOAP (Simple Access Object Protocol) protokolą, bus komunikuojama su Fedora sistema, iš kurios atitinkamai bus surinkta reikiama informacija apie talpinamus e. dokumentus – ir jie bus suindeksuojami. Sukurtoji sistema indeksuos ne tik minėtų formatų e. dokumentų turinį tekstiniame formate, bet taip pat ir jų metaduomenis.

Kuriamas sistema veiks JBoss konteineryje. [5] Tai yra atviro kodo Java taikomųjų programų konteineris.

Bus sudaryta žiniatinklio sąsaja, per kurią bus galima vykdyti paiešką, bei administruoti indeksą. Registruotų vartotojų paieškos istorija bus išsaugoma tam, kad būtų galima atlikti pakartotinę paiešką po tam tikro laiko.

Kad pasinaudotų sistema paprastam vartotojui tereikia turėti asmeninį kompiuterį su prijungtu interneto ryšiu ir jis iš karto galės naudotis sistemos teikiamomis paslaugomis.

2.3.1. Indekso kūrimas ir redagavimas

Administratoriaus teises turintis vartotojas gali daryti įvairius veiksmus su indeksu. Jis gali pašalinti duomenis iš indekso įvedęs atitinkamo objekto identifikatorių (PID), taip pašalinti visus duomenis iš indekso t.y. sukurti tuščią indeksą, taip pat perindeksuoti pasirinktą objektą įvedęs atitinkamo objekto unikalų identifikatorių (PID) arba iš karto visus objektus.

2.3.2. Paieškos vykdymas

Tai yra sistemos pagrindinė funkcija. Vartotojai turi įvesti užklauso duomenis norėdami atlikti kažkokią paiešką FEDORA talpykloje. Vartotojas gali nustatyti filtrus kokius rezultatus rodyti, o kokių nerodyti, kaip pvz.: nurodyti metų intervalą. Sistema turėtų išsaugoti paieškos pagrindinius parametrus, tam kad vartotojui nereiktų jų iš naujo įvedinėti atliekant paiešką su kitais užklauso žodžiais.

2.3.3. Rezultatų pateikimas

Atlikęs paiešką vartotojas gali peržiūrėti paieškos rezultatus. Vartotojas gali pasirinkti kaip išrikiuoti rezultatus pagal metus ar pagal antraštę. Vartotojas gali nueiti į patį VDDDB objektą ir paržiūrėti jo turinį.

2.3.4. Paieškos istorijos peržiūrėjimas

Vartotojas gali peržiūrėti buvusių užklauso istoriją. Jis gali ištrinti nereikalingus įrašus iš istorijos, bei gali pakartotinai įvykdyti paieškos užklauso.

2.4. Projektavimo technologija

Pagrindinės Java programavimo kalba [4] paremtos naudojamos technologijos yra šios:

- JBoss Seam Framework
- Apache Lucene, Zebra paieškos variklių bibliotekos
- Apache Axis.

2.4.1. Apache Lucene paieškos variklis

Apache Lucene yra aukštos kokybės pilno teksto (full-text) paieškos variklio biblioteka parašyta JAVA programavimo kalboje. [1]

Lucene privalumai:

- Prioritetinė paieška t.y. tinkamiausi rezultatai gražinami pirmiausiai.
- Galimos įvairios užklauso.

- Paieška laukuose pvz.: autorius, pavadinimas, turinys.
- Rikiavimas pagal bet kuri lauką.
- Leidžia lygiagrečiai vykdyti atnaujinimą ir paiešką.

Apache lucene paieškos variklis yra palaikomas Linux, Solaris bei Windows operacinių sistemų. Sukurtas indeksas sudaro nuo dvidešimt iki trisdešimt procentų indeksuojamo teksto dydžio.

Paieškos indeksas sudarytas iš segmentų t.y. mažesnių indekso dalių. Segmentuose informacija yra išsaugojama kaip bitų seka. Paieškos indekse gali būti suindeksuota iki keturių milijardų dokumentų.

Paieškos užklausos gali būti sudaromos labai įvairios nuo elementaraus žodžio iki sudėtingiausiu išraiškų. Pavyzdžiui: „Kaunas“, „kompiuteris“ ar „kompiuterijos naujovės pasaulyje“. Taip pat atpažįstami ir daugelyje paieškos sistemų naudojami simboliai „*“ ir „?“. Žvaigždutė reiškia vieno ar kelių skirtingų simbolių paiešką. Jei žvaigždutė padėta ieškomo žodžio gale. Tarkim „Kaun*“, tai reikš, kad bus ieškoma žodžių „Kaunas“, „Kauno“, „Kaunui“ ir t.t. Klaustuko simbolis reiškia, kad bus ieškoma vieno skirtingo simbolio į ta vietą kur jis yra pavyzdžiui, „Kaun?“ – bus ieškoma žodžių „Kauną“, „Kauno“ ir t.t.

Paieškos užklausos taip pat gali būti sudėtinės panaudojant logines operacijas tokias kaip „AND“, ar „OR“. Tarkim reikia surasti visus universitetus išskyrus Kauno. Šiuo atveju turėtume tokią užklausą su minuso „-“ ženklu „universitetas - Kaunas“. Ir atitinkamai norėdami surasti visus Kauno universitetus turėtume vykdyti užklausą su pluso „+“ ženklu.

2.4.2. Zebra paieškos variklis

Zebra yra aukštos kokybės pilno teksto (full-text) paieškos variklis. Jam duomenys gali būti pateikiami XML arba MARC formatu. [11]

Zebra privalumai:

- Įvairus indeksuojamų duomenų formatus (XML, SGML, Text, ISO2709 (MARC)).
- „Miglota paieška“ (automatinis klaidų ištaisymas).
- Paieška laukuose pvz.: autorius, pavadinimas, turinys.
- Išraiškų ieškojimas.

Zebra paieškos variklis turi panašias savybes kaip ir Lucene paieškos variklis. Jis taip gali suindeksuoti labai didelius dokumentų kiekius, bei vykdyti žaibišką paiešką. Zebra paieškos variklyje yra realizuota galimybė prieš gaunant rezultatus juos surikiuoti.

2.4.3. JBoss Seam Framework

JBoss Seam yra atviro kodo Java programavimo kalba paremtas kūrimo šablonas, skirtas kurti sudėtingas WEB taikomas programas JBoss Seam yra integruota Java Server Faces (JSF) ir EJB. JSF yra viena naujausių technologijų. Tai technologija kuri supaprastina vartotojo sąsajos kūrimą ir jos susiejimą su serverio programine įranga. Programuotojams ji suteikia galimybę greitai sudaryti žiniatinklio taikomas programas panaudojant pakartotinio panaudojimo komponentus puslapyje, apjungiant šiuos komponentus su konkrečios programinės įrangos duomenų šaltiniais bei suteikia galimybę serveryje apdoroti vartotojo naršyklėje sugeneruotus įvykius. Ši technologija apima daug kitų ištobulintų technologijų: JSP (Java Server Pages), Servlet ir kitas. [7]

2.4.4. Apache Axis

Apache Axis yra SOAP (Simple Object Access Protocol) protokolo realizacija. SOAP protokolas yra unifikuotas ir yra priimtas W3 konsorciumo (W3C) standartas.

Apache Axis SOAP realizacija yra naudojama indeksavime komunikavimui su Fedora talpykla. [10]

SOAP yra protokolas skirtas struktūrizuotos informacijos mainams decentralizuotoje, paskirstytoje aplinkoje. Tai XML (Extensible Markup Language) žymų kalba paremtas protokolas kuris sudarytas iš trijų pagrindinių dalių:

- Voko, kuris apibrėžia šabloną kuriame nurodyta iš ko yra sudaryta žinutė ir kaip reikia ją apdoroti
- Kodavimo taisyklių rinkinio, nurodančio taisykles kaip interpretuoti programos apibrėžtus tipus
- Susitarimo apibrėžiančio procedūrų iššaukimą ir jų sugeneruotų rezultatų parsisuntimą iš nutolusio objekto [8]

2.5. Įgyvendinimo problemos

Bandant įgyvendinti paieškos sistemą tenka susidurti su keletu projektavimo ir įgyvendinimo problemų.

2.5.1. Indekso dydis

Optimaliai paieškai atlikti turi būti kuriamas indeksas. Jei FEDORA saugykloje saugomų objektų yra nedaug tai ir indeksas neužims daug vietos. Sistemos resursai priklauso nuo to kokio dydžio indeksas bus kuriamas. Indeksavimo sistemos redaguojamas indeksas įprastai yra saugomas toje pačioje sistemoje, kurioje vyksta pats indeksavimas, todėl esant

dideliam suindeksuotų dokumentų kiekiui šis indeksas gali stipriai išaugti ir tam tikrose situacijose užimti nemažai disko vietos. Akivaizdu, kad pilno teksto indeksui sukurti prireiks nemažo serverio. Dėl šio resurso problemų neturėtų iškilti, kadangi sistemą gali aptarnauti du galingi SUN serveriai:

Sun V40Z serveris [9]

Sun E2900 serveris [9]

2.5.2. Specifinių raidžių atpažinimas

Kadangi e. dokumentai bus saugomi įvairiomis kalbomis, reikės realizuoti ir paiešką ne vien lotyniškais raidėmis t.y. reikės padaryti, kad būtų atpažįstamos tokios raidės kaip „š“, „ž“, „ė“ ir kt.

2.6. Išvados

- Paieškos sistema leis lengvai susirasti bet kokį dokumentą esanti FEDORA saugykloje.
- Tai leis dar efektyviau panaudoti elektroninėje bibliotekoje saugomus dokumentus.
- Paieškos sistema veiks internete, todėl ją lengvai bus galima pasiekti iš bet kurios vietos kur yra internetas.
- Bus galima vykdyti paiešką įvairiomis kalbomis.

3. PROJEK TINĖ DALIS

Šiame skyriuje pateikiamos esminės sukurtos sistemos specifikacijos dalys: sistemai keliami reikalavimai, sistemos architektūra, su sistemos realizacija susijusių ypatumų aprašymai, sistemos testavimas ir diegimas.

3.1. Sistemos reikalavimai

3.1.1. Sistemos tikslai

Sukūrus ir įdiegus eLABa bus sudarytos palankios sąlygos, naudojant informacines ir komunikacines technologijas, sparčiau skleisti naujausias mokslo žinias ir mokomąją medžiagą, gerinti mokslinių tyrimų ir studijų kokybę, plėtoti nuotoline ir visą gyvenimą trunkančias studijas, propaguoti Lietuvos mokslininkų bei mokslo ir studijų institucijų veiklą ir pasiekimus, ugdyti žinių visuomenę ir plėtoti žinių ekonomiką, sudaryti prielaidas suformuoti elektroninę Lietuvos akademinę biblioteką. Mano kuriamas posistemis turėtų užtikrinti sukauptos informacijos paiešką, kad ji būtų efektyviai panaudota iš pradžių tarp mokslo institucijų, o vėliau ir visuomenėje. [2]

3.1.2. Vartotojai

Sistemoje numatomos dvi vartotojų grupės:

1) Vartotojai – tai žmonės, kurie naudojami sukurta sistema. Tai daugiausia turėtų būti akademinės visuomenės nariai.

2) Administratoriai – tai asmenys administruojantys sistemą. Administratoriaus teises turintys vartotojai galės vykdyti visas paprastų vartotojų funkcijas, bet papildomai turės dvi funkcijas t.y. indekso kūrimas ir redagavimas.

Sistemos vartotojai / administratoriai naudodamiesi sistema gali:

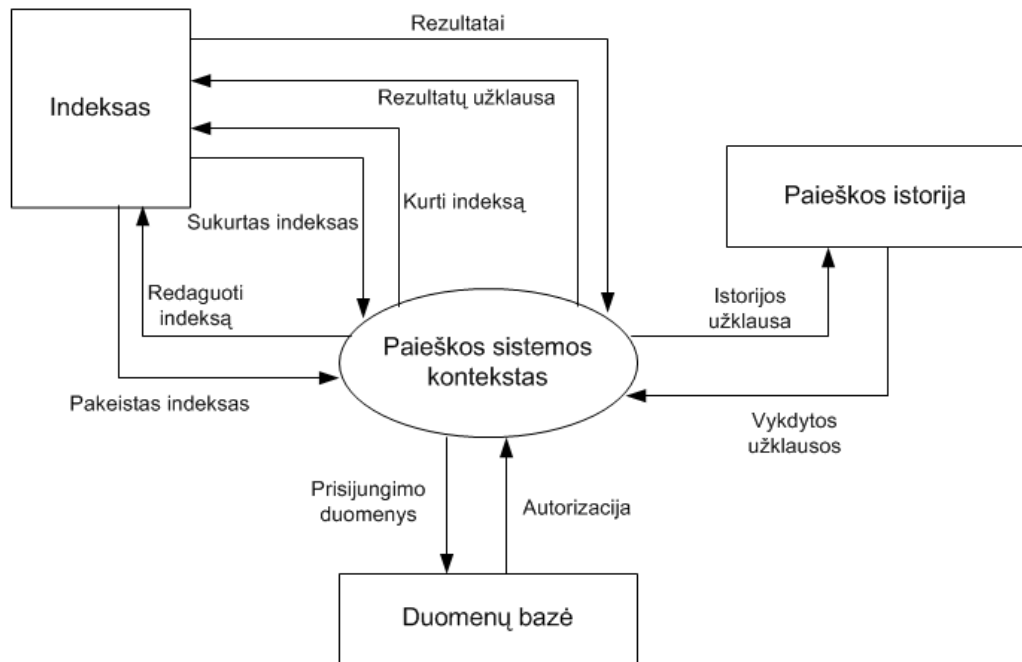
1. Prisijungti / atsijungti
2. Atlikti paiešką
3. Peržiūrėti rezultatus
4. Peržiūrėti istoriją
5. Kurti / redaguoti indeksą

3.1.3. Diegimo aplinka

Kuriama sistema veiks JBoss konteineryje kažkuriame iš įmonės serverių. Tai yra atviro kodo Java taikomųjų programų konteineris. Norėdami pasinaudoti programa vartotojai turės jungtis prie serverio naudodami interneto naršyklę ir taip gaus numatytas paslaugas.

3.1.4. Veiklos kontekstas

Šiame skyriuje pateikiama paieškos sistemos veiklos konteksto diagrama vaizduojanti paieškos sistemos sąveiką su duomenų saugyklomis. (6 pav.)



6 Pav. Konteksto diagrama

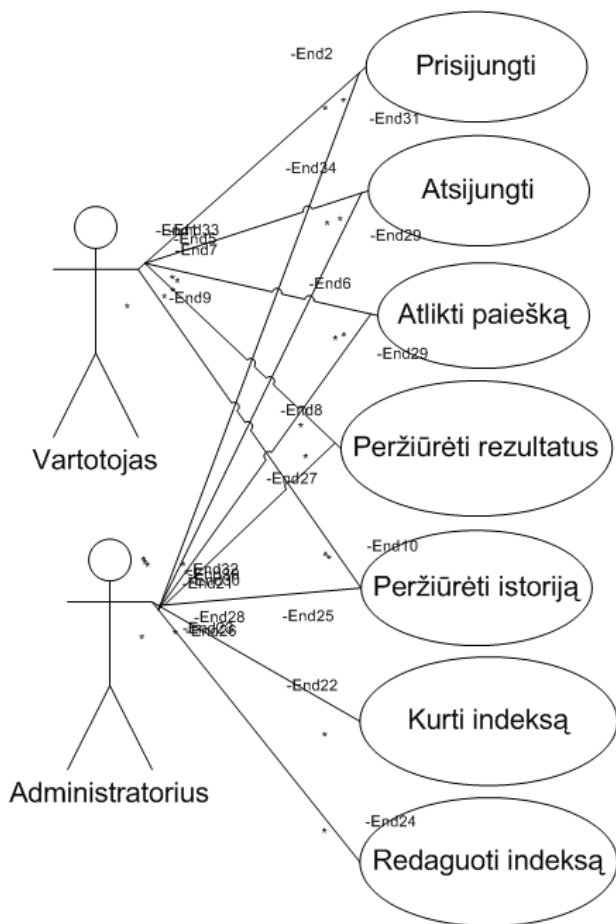
Sistemoje numatomi duomenų srautai pateikiami 1 lentelėje. Čia „in“ žymimas įeinantis srautas, o „out“ – išeinantis srautas.

1 Lentelė. Veiklos padalinimas

Eil nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Vartotojas prisijungia prie sistemos	Prisijungimo duomenys (in)
2	Sistemos administratorius sukuria indeksą	Indeksas (out)
3	Sistemos administratorius redaguoja indeksą	Įrašo PID (in) Pakeistas indeksas (out)
5	Vartotojas vykdo paiešką	Paieškos užklausa (in) Rezultatai (out)
4	Vartotojas nori peržiūrėti istoriją	Istorijos užklausa (in) Rezultatų istorija (out)
6	Vartotojas nori peržiūrėti rezultatus	Rezultatai (out)

3.1.5. Sistemos panaudos atvejai

Sistemos panaudos atvejai pavaizduoti panaudos atvejų diagramoje.



7 Pav. Panaudos atvejų diagrama

Aptarsime kiekvieną panaudos atvejį.

1. **Prisijungti / atsijungti** – registruotas vartotojas gali prisijungti prie sistemos, kad galėtų peržiūrėti paieškos istoriją. Pabaigęs darbą jis turėtų atsijungti nuo sistemos.
2. **Atlikti paiešką** – tai yra pagrindinė sistemos funkcija. Kiekvienas sistemos vartotojas gali vykdyti paiešką visateksčių dokumentų duomenų bazėje pagal tam tikrą sritį ar tiesiog pagal kažkokį žodį ar žodžių junginius.
3. **Peržiūrėti rezultatus** – įvykdžius paiešką, paieškos rezultatai arba pranešimas apie rezultatų nebuvimą yra išvedamas į ekraną. Vartotojas gali pasirinkti kaip išrikiuoti rezultatus pagal metus, antraštę ar pan. Vartotojas gali nueiti į patį VDDB objektą ir peržiūrėti jo turinį.
4. **Peržiūrėti istoriją** – registruoti vartotojai gali peržiūrėti visų anksčiau vykdytų užklausų istoriją ir pakartotinai ją įvykdyti. Neregistruoti vartotojai gali peržiūrėti tik esamos sesijos vykdytų užklausų istoriją ir pakartotinai ją įvykdyti.

5. **Kurti / redaguoti indeksą** – tai yra tik vartotojų, turinčių administratoriaus teises funkcija. Administratorius gali pašalinti duomenis iš indekso įvedęs atitinkamo objekto identifikatorių (PID). Administratorius gali pašalinti visus duomenis iš indekso t.y. sukurti tuščią indeksą. Administratorius gali perindeksuoti pasirinktą objektą įvedęs atitinkamo objekto unikalų identifikatorių (PID). Administratorius gali perindeksuoti visus FEDORA saugykloje esančius objektus.

3.1.6. Funkciniai reikalavimai

Funkciniai reikalavimai suformuluoti detalizuojant esamus panaudos atvejus.

Pateiksime esminius sistemai taikomus funkcinius reikalavimus:

- Sistema turi informuoti vartotoją apie galimą laikiną prisijungimo prie sistemos sutrikimą. Daugelio vartotojų pakartotinis nesėkmingas prisijungimas prie sistemos gali reikšti, kad serveris kur patalpinta duomenų bazė neveikia, arba yra kitokia sistemos sutrikimai.
- Sistema turi pranešti apie sąmoningą ar nesąmoningą vartotojo atsijungimą nuo sistemos. Dėl per ilgo neveiklumo vartotojas gali būti atjungtas nuo sistemos taip apsaugant jo asmeninius duomenis.
- Sistema vykdydama paiešką turi būtina gražinti kažkokius rezultatus, o jeigu jų nėra būtina apie tai informuoti vartotoją.
- Sistema turi atvaizduoti gautus rezultatus, o jeigu jų nėra apie tai turi būti pranešta vartotojui.
- Sistema turi atvaizduoti rezultatų paieškos istoriją. Jeigu ji tuščia apie tai turi būti pranešta.
- Sistema turi pranešti apie sėkmingą ar nesėkmingą indekso sukūrimą ar redagavimą. Netinkamo formato duomenys negali būti suindeksuoti. Sistema informuoja sistemos administratorių apie nesuindeksuojamą dokumentą išvesdama jo PID į ekraną.

3.1.7. Nefunkciniai reikalavimai

Nefunkciniai sistemos reikalavimai apima reikalavimus vartotojo sąsajai, sistemos saugumui, panaudojamumui, vykdymo charakteristikoms, duomenims, sistemos palaikymui ir pan.

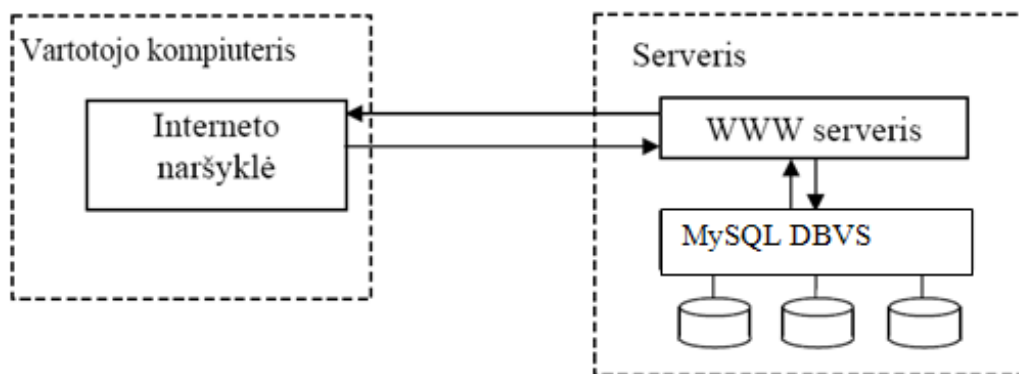
Pateiksime esminius sistemai taikomus nefunkcinius reikalavimus:

- Vartotojo sąsaja turėtų būti neperkrauta, paprasto dizaino, meniu turėtų būti aiškus ir tinkamai išdėstytas.
- Sistema turi būti paprasta naudotis vartotojui, susipažinusiam su bazine naudojimosi instrukcija.
- Sistema neturi perduoti asmeninių vartotojo duomenų kitiems, su sistemos administravimu nesusijusiems, asmenims.
- Labai svarbu, kad paieškos užklausos būtų vykdomos greitai. Ir taip pat greitai ir vaizdžiai pateikiami paieškos rezultatai.
- Vartotojo sąsajos elementai turi būti sugrupuoti pagal paskirtį ir, jei reikia, išdėlioti atskirose skiltyse. Sąsajos elementų grupavimas pagal paskirtį suteikia galimybę greičiau juos aptikti ir greičiau atlikti reikiamus pakeitimus.
- Kaip ir dauguma sistemų, ši sistema kuriama su nuostata, kad ateityje ji bus tobulinama ir bus išplėsta atsižvelgiant į vartotojų pageidavimus.

3.2. Sistemos architektūra

3.2.1. Sistemos veikimo principas

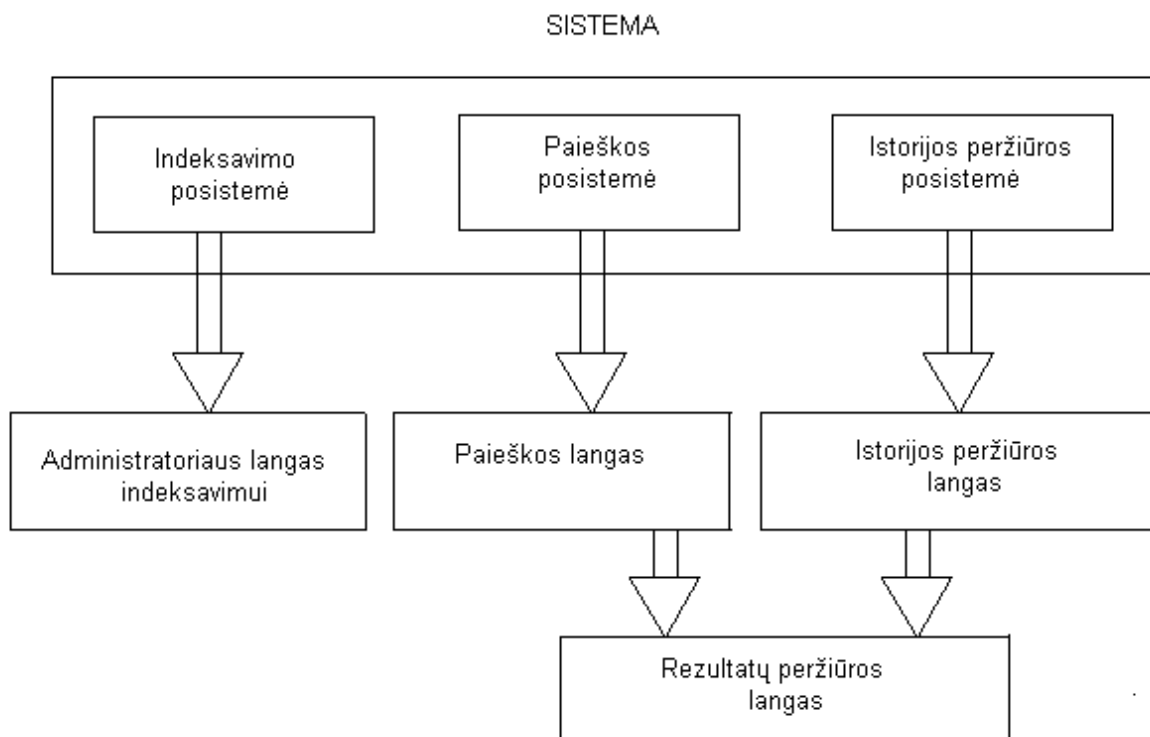
Sistemos veikimas yra paremtas kliento ir serverio principu. Norėdamas pasinaudoti sistema vartotojas jungiasi prie serverio, kuriame įdiegta eLABa paieškos sistema, naudodamasis interneto naršykle. (8 pav.)



8 Pav. Sistemos veikimo principas

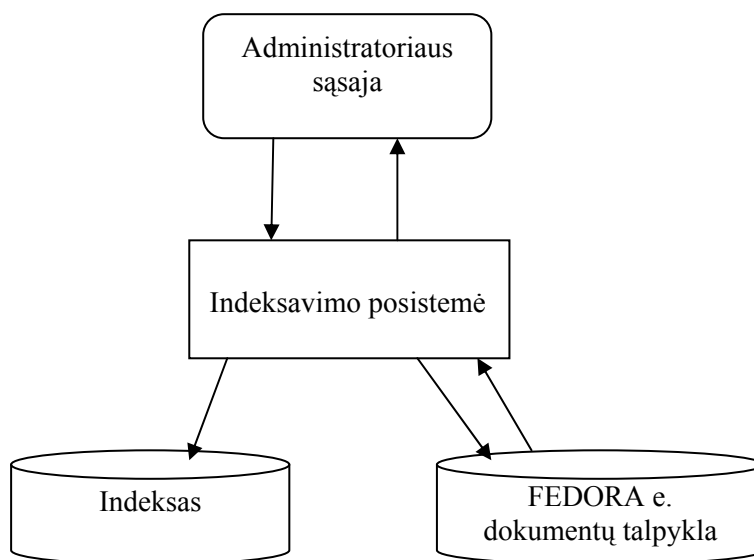
3.2.2. Sistemos sudėtis

Struktūriškai sistema susideda iš keleto dalių – indeksavimo, paieškos ir rezultatų atvaizdavimo, bei istorijos kaupimo bei atvaizdavimo.



9 Pav. Sistemos struktūra

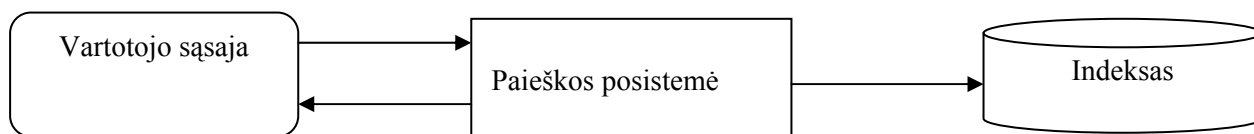
Indeksavimo posistemė yra turbūt pati svarbiausia šioje sistemoje, kadangi nuo gero ar blogo dokumentų suindeksavimo tiesiogiai priklauso paieškos laikas ir kokybė. Šios posistemės bendroji veikimo schema pateikta žemiau.



10 Pav. Bendroji indeksavimo posistemės veikimo schema

Kaip matome iš paveikslėlio indeksavimo posistemė yra valdoma per administratoriaus sąsają. Galime pastebėti, kad indeksavimo posistemė gali būti atskirta nuo talpyklos ir veikti visiškai kitoje tarnybinėje stotyje, bet paprastai indeksavimas yra vykdomas toje pačioje tarnybinėje stotyje kur yra saugomi indeksuojami objektai.

Toliau panagrinėkime paieškos posistemę, kurios bendroji veikimo schema atvaizduota 11 pav.

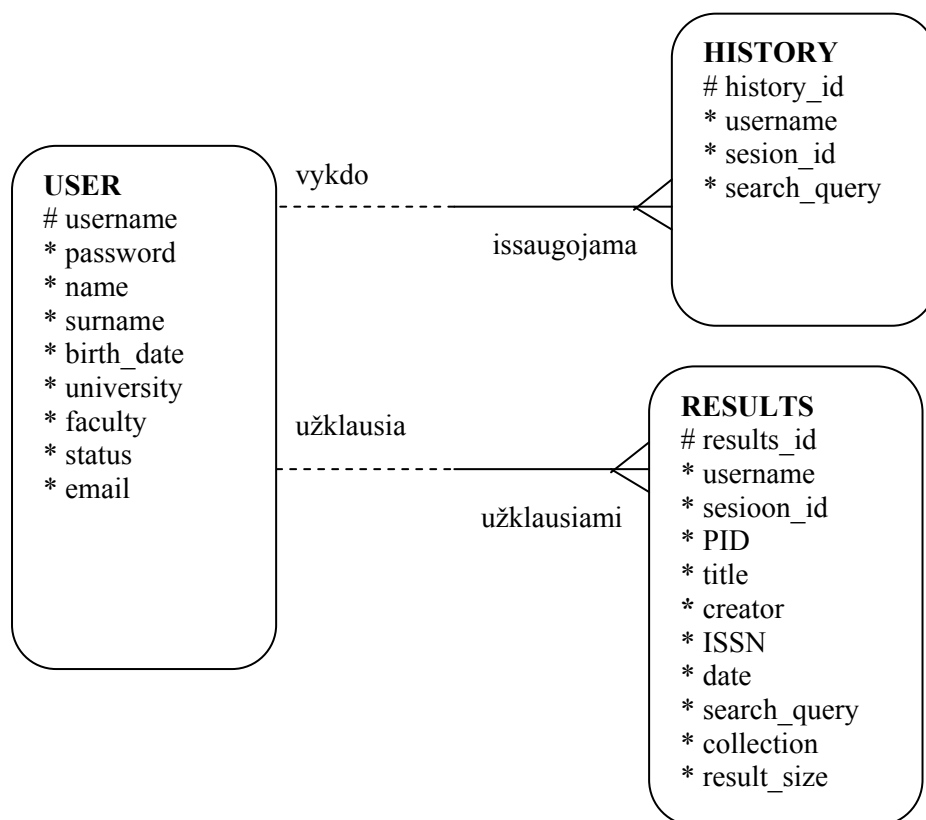


11 Pav. Bendroji paieškos posistemės veikimo schema

Paieškos posistemė bus atsakinga už paieškos vykdymą ir rezultatų grąžinimą. Gauti rezultatai bus atvaizduojami rezultatų peržiūros lange. Kaip matome schemoje, paieškos posistemė naudoja indeksavimo sistemos sukurtą indeksą paieškai atlikti.

3.2.3. Duomenų bazė

Duomenų saugojimui sistemoje naudojama MySQL duomenų bazė. Duomenų bazės esybių ryšių diagrama pateikta 12 pav.



12 pav. ER diagrama

Duomenų bazės lentelių specifikacijos su lentelių laukais, duomenų tipais ir laukų aprašymais pateikiamos toliau.

Duomenų bazės lentelė „USER“ naudojama saugoti duomenims apie sistemos vartotojus. Joje saugojami pagrindiniai duomenys identifikuojantys ir apibūdinantys vartotoją.

Lentelė Nr. 1 Duomenų bazės lentelės „USER“ specifikacija

USER		
Laukas	Tipas	Aprašymas
username	VARCHAR(11)	Vartotojo prisijungimo vardas
password	VARCHAR(100)	Vartotojo prisijungimo slaptažodis
name	VARCHAR(100)	Vartotojo vardas
surname	VARCHAR(100)	Vartotojo pavardė
birth_date	BIGINT(25)	Gimimo data
university	VARCHAR(100)	Universitetas
faculty	VARCHAR(100)	Fakultetas
status	VARCHAR(100)	Vartotojo statusas
email	VARCHAR(100)	Elektroninio pašto adresas

Duomenų bazės lentelėje „RESULTS“ saugojami paieškos rezultatai. Įvykdžius sėkmingą paiešką ir gavus kažkokius rezultatus, jie yra išsaugomi šioje lentelėje.

Lentelė Nr. 2 Duomenų bazės lentelės „RESULTS“ specifikacija

RESULTS		
Laukas	Tipas	Aprašymas
results_id	INTEGER	Identifikatorius
username	VARCHAR(11)	Vartotojo prisijungimo vardas
sesijon_id	VARCHAR(100)	Sesijos identifikatorius
PID	VARCHAR(1000)	Unikalus dokumento identifikatorius
title	VARCHAR(1000)	Dokumento antraštė
creator	VARCHAR(1000)	Autorius
ISSN	BIGINT(1000)	Sisteminis dokumento numeris
date	VARCHAR(1000)	Dokumento sukūrimo data
search_query	VARCHAR(1000)	Paieškos užklausa
collection	VARCHAR(1000)	Kolekcija kuriai priklauso darbas
result_size	INTEGER	Gautų rezultatų kiekis

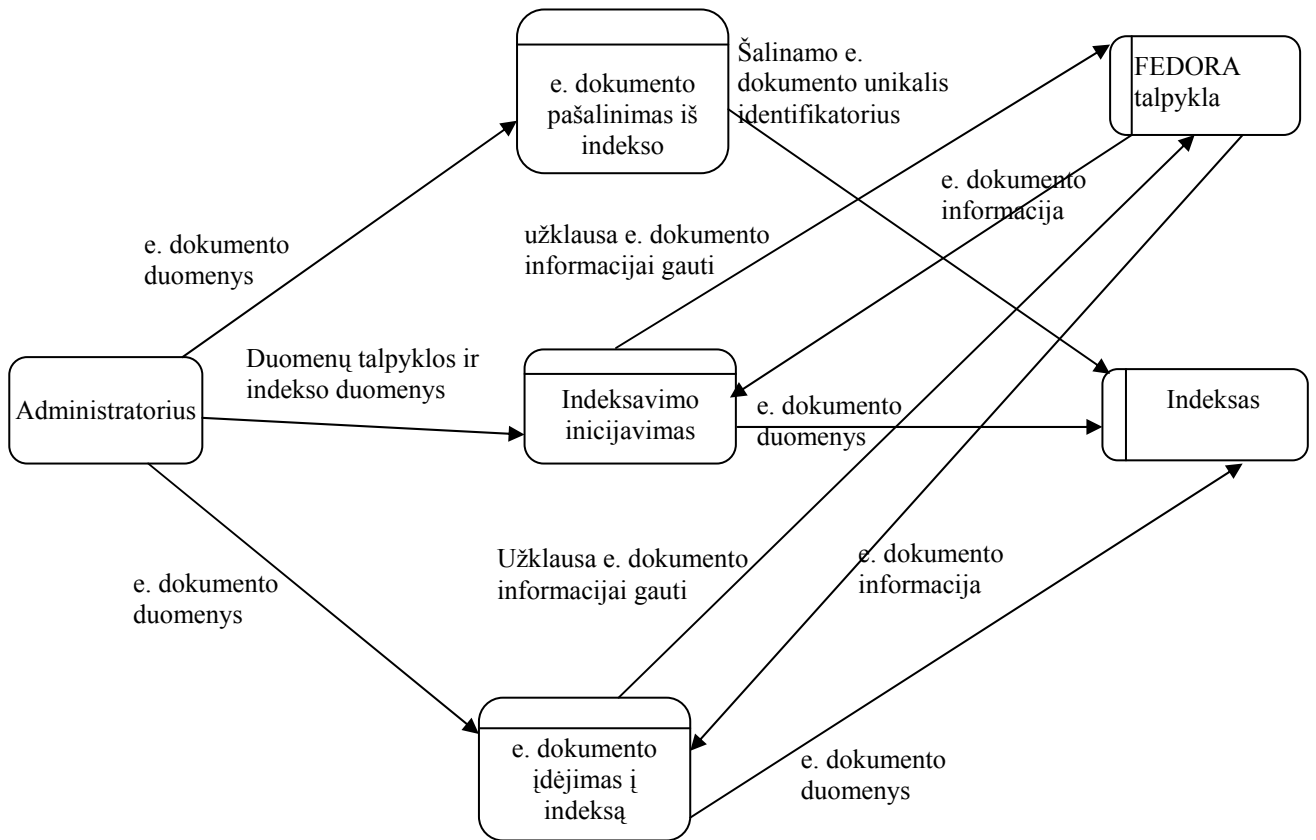
Duomenų bazės lentelėje „HISTORY“ saugojama paieškos istorija. Įvykdžius sėkmingą paiešką ir gavus kažkokius rezultatus, paieškos užklausa yra išsaugoma šioje lentelėje.

Lentelė Nr. 3 Duomenų bazės lentelės „HISTORY“ specifikacija

HISTORY		
Laukas	Tipas	Aprašymas
history_id	INTEGER	Identifikatorius
username	VARCHAR(11)	Vartotojo prisijungimo vardas
sesijon_id	VARCHAR(100)	Sesijos identifikatorius
search_query	VARCHAR(1000)	Paieškos užklausa

3.2.4. Duomenų srautų diagramos (DSD)

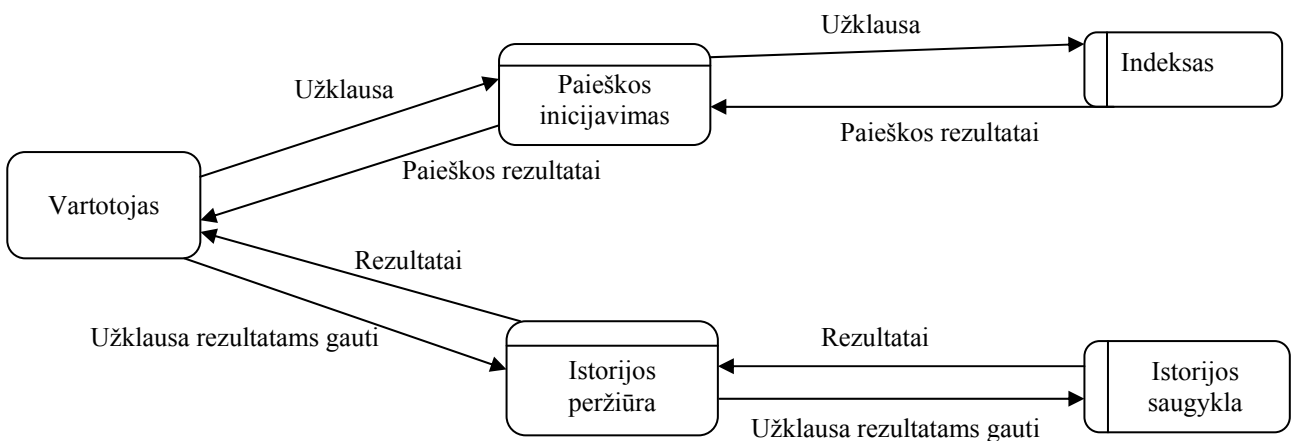
Indeksavimo DSD:



13 pav. Indeksavimo DSD

Iš diagramos (13 pav.) matyti, kad administratorius inicijuoja e. dokumento įdėjimą į indeksą, nurodydamas dokumento unikalų identifikatorių (PID), taip pat e. dokumento pašalinimą nurodydamas PID, bei visų e. dokumentų indeksavimą.

Paieškos DSD:



14 pav. Paieškos DSD

14 paveikslėlyje yra pavaizduota paieškos duomenų srautų diagrama. Vartotojas pateikia paieškos užklausą ir gauna rezultatus, bei gali pateikti užklausą pakartotiniam rezultatų peržiūrėjimui. Pateikęs užklausą, vartotojas gauna rezultatus, kurie yra išvedami ir parodomi rezultatų lange.

3.3. Testavimas

3.3.1. Testavimo tikslai ir apimtis

Testavimo metu siekiama patikrinti sistemos atitikimą specifikacijai, aptikti galimas programavimo bei logikos klaidas, patikrinti atskirų sistemos dalių funkcionavimą. Taip pat atliekama vartotojo sąsajos patikra ir įvertinimas. Šis procesas skirtas ne tik surasti ir ištaisyti klaidas, bet ir įvertinti programinės įrangos patikimumą, nustatyti vartotojo palankumą/pripažinimą.

Testavimui naudojamos kelių tipų strategijos. Vienetų testavimui taikoma „baltos dėžės“ (angl. „white box“) testavimo strategija, kuria siekiama patikrinti kodo padengimą bei aptikti galimas vidines logikos klaidas. Taip pat naudojama „juodos dėžės“ (angl. „black box“) testavimo strategija, kuria bandoma aptikti galimas funkcines klaidas.

„Juodos dėžės“ testavimo strategija taikoma ir integracijos testavime, siekiant patikrinti apjungtų sistemos dalių funkcionavimą. Testavimo metu taip pat vertinama vartotojo sąsaja, jos patogumas, priimtinumas bei kitos savybės.

3.3.2. Sąsajos testavimas

Sistema yra realizuota JBoss Seam Framework pagrindu, todėl vartotojo sąsajos testavimas nėra paprastas procesas. Šis procesas priklauso nuo programavimo stiliaus, todėl programuojant taikoma sluoksninė metodika, kuri palengvina vartotojo sąsajos testavimą. Sluoksninės metodikos idėja yra realizuoti vartotojo sąsajos komponentus tokiu būdu, kad grafines vartotojo sąsajos kodas būtų atskirtas nuo funkcionalumą realizuojančios dalies ir grafinėje dalyje paliktas tik labai plonas specifinis sluoksnis.

Automatizuotai testuoti grafinę vartotojo sąsaja nėra lengva, ypač jei ji yra realizuota didelę specifiką įnešančių tarpinių technologijų (konkrečiu atveju JBoss Seam Framework) pagrindu. Todėl grafinių komponentų testavimas atliekamas rankiniu būdu – paprasčiausiai atvaizduojant komponentą ir stebint jį ekrane prie įvairių įmanomų vaizduojamų duomenų.

Atskirta grafinių komponentų funkcionalumo dalis yra patalpinta atskirose klasėse, todėl čia pritaikyta vienetų testavimo (unit test) strategija komponentų funkcionalumui testuoti.

Kadangi rankiniu būdu testuojant grafinę vartotojo sąsają sistemai paduodamos įvairios reikšmės ir stebimas sistemos elgesys, tai tuo pačiu kai kur dalinai testuojamas ir sistemos funkcionalumas. Tačiau čia pagrindinis akcentas yra vartotojo sąsajos komponentų languose pateikiamų klaidos pranešimų korektiškumas, nes funkcionalumo testavimas vykdomas atskirai.

Pateiksime dažniausiai pasitaikančių sąsajos klaidų, kurios buvo ištestuotos sąrašą:

- Klaidingai naudojama sąsaja – pakviestas komponentas iškviečia kitą komponentą ir daro klaidas jo naudojamoje sąsajoje, pvz.: parametrai neteisinga tvarka.
- Sąsajos neteisingas interpretavimas – kviečiantis komponentas remiasi prielaida apie kiekvieno komponento elgesį, kuris yra neteisingas.
- Synchronizacijos klaidos – pakviestas ir kviečiantysis komponentai dirba skirtingais greičiais ir nebegaliojanti informacija vis dar būna prieinama.

3.3.3. Testavimo strategijos

Taikomos testavimo strategijos:

- **Vienetų testavimas** – strategija, apimanti vienetų, kurie yra mažiausios sistemos dalys, testavimą. Konkrečiu atveju šie vienetai bus JAVA klasės ir jų metodai.
- **Integravimo testavimas** – strategija taikoma prieš bendrą sistemos testavimą. Konkrečiu atveju orientuota į susietų Java klasių/modulių tarpusavio sąveikos testavimą.
- **Priėmimo testavimas** – testavimo strategija, skirta įsitikinti, jog kuriama sistema atitinka užsakovo reikalavimus. Testavimas vykdomas dalyvaujant užsakovui ir/arba sistemos vartotojams.
- **Aukšto lygio testavimas** – užsakovo aplinkoje vykdomas testavimas, kurio metu siekiama nustatyti, ar korektiškai veikia sistema specifinėje užsakovo aplinkoje.

3.3.4. Vienetų testavimas

Sistema realizuota JBoss Seam Framework technologijos pagrindu, todėl kaip vienetus čia galima traktuoti JBoss Seam komponentus. Tačiau šie komponentai iš principo yra grafinės sąsajos komponentai, apjungti su susijusiu funkcionalumu. Kaip jau buvo minėta, programavimas buvo vykdomas siekiant kuo labiau atskirti grafinę sąsają nuo susieto funkcionalumo, kad komponentų funkcijoms galima būtų taikyti vienetų testavimo strategiją.

Pagrindiniai testuojami vienetai yra statinės ir ne statinės klasės, bei atitinkamai jų funkcijos ir metodai. Programuojant buvo siekiama kur įmanoma ir prasminga bendrojo pobūdžio funkcionalumą realizuoti kaip statines funkcijas (iš kurių yra sudarytos statinės klasės), tokiu būdu supaprastinant sistemos struktūrą ir palengvinant testavimo procedūrą. Tokios klasės testuojamos paprasčiausiai iškviečiant jų funkcijas ir lyginant jų gražinamus rezultatus. Testuojant ne statines klases, sukuriami jų egzemplioriai, o vidiniams kintamiesiems suteikiamos įvairios reikšmės (tarp jų ir ribinės). Testuojant abiem atvejais perduodamiems per metodus ar funkcijas parametrus bus suteikiamos įvairios galimos reikšmės.

Testavimui buvo naudojamos JUnit karkaso priemonės. Testavimo metu buvo sudaryti automatizuoti testai, testuojantys atskirus vienetus, kurių testavimo rezultatai pateikti toliau:

Lentelė Nr. 4 Vienetų testavimo rezultatai

Testuojamo vieneto (metodo) pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
Login(username , password)	Prisijungimo metodo testavimas	5	Ne	Buvo išbandyta ar prie sistemos gali prisijungti neregistruoti vartotojai.
Logout()	Atsijungimo nuo sistemos metodo testavimas	2	Ne	Buvo išbandyta ar pabaigiama vartotojo sesija atsijungus.
updateIndex(PI D)	Naujo objekto įtraukimo į indeksą testavimas	13	Taip	Buvo išbandyta ar objektas (dokumentas) sėkmingai suindeksuojamas.
deleteIndex(PI D)	Objekto pašalinimo iš indekso testavimas	4	NE	Buvo išbandyta ar objektas (dokumentas) sėkmingai pašalinamas iš indekso.
createIndex()	Visų dokumentų perindeksavimo metodo testavimas	3	Ne	Buvo išbandyta ar visi objektai iš naujo suindeksuojami.
deleteIndex()	Indekso ištrynimo metodo testavimas	5	Ne	Buvo išbandyta ar pašalinami visi objektai iš indekso.
writeResults()	Rezultatų įrašymo į duomenų bazę metodo testavimas	3	Taip	Buvo išbandyta ar gauti rezultatai teisingai surašomi į duomenų bazės lentelę.
writeHistory()	Paieškos istorijos įrašymo į duomenų bazę metodo testavimas	7	Taip	Buvo išbandyta ar paieškos istorija teisingai išsaugoma duomenų bazėje.
deleteHistory(sesion_id)	Paieškos istorijos pašalinimo iš duomenų bazės	4	Ne	Buvo išbandyta ar galima pašalinti norimą paieškos istorijos įrašą.

	testavimas			
--	------------	--	--	--

3.3.5. Integravimo testavimas

Šiame testavimo metode, testavimas atliekamas su atskirais sistemos komponentais, kurie po vienetų testavimo yra apjungiami į didesnius komponentus. Naudojamas „iš apačios į viršų“ (angl. „bottom-top“) integravimo testavimas, kuomet testavimas pradedamas nuo žemiausio lygio komponentų iki aukščiausio lygio komponentų. Taip atskiri komponentai integruojami į lygius, kol sukomplektuojama pilna sistema. Integravimo testavimas yra „juodos dėžės“ tipo su testais, kurie generuojami pagal pateiktas specifikacijas.

Pirmiausiai, buvo sujungta ir ištestuota indeksavimo ir paieškos komponentai. Vėliau, buvo prijungta ir ištestuota istorijos saugojimo ir peržiūros komponentai.

3.3.6. Priėmimo testavimas

Šiame etape sukurta programinė įranga testuojama paties vartotojo (užsakovo) arba jam dalyvaujant. Testavimas atliekamas peržiūrint kiekvieną programinės įrangos panaudos atvejį. Tokiu būdu nustatoma, ar pateikta programinė įranga atitinka vartotojo poreikiams, ar išpildyti visi specifikacijoje pateikti reikalavimai. Testavimui į pagalbą gali būti pasitelktas „juodos dėžės“ metodas. Kadangi visa programinė įranga jau bus sukurta, šiuo atveju bus atliekamas visos sistemos testavimas. Jei programinė įranga specifikacijos neatitinka, tuomet turi būti pildoma klaidų ataskaita.

3.3.7. Testavimo rezultatų kaupimas

Testavimo metu momentiniai testavimo rezultatai bus pateikiami testuotojui ekrane bei registruojami testavimo pranešimų žurnale (angl. „log“).

3.3.8. Testavimo išvados

Testavimas yra atskiras šios programų sistemos kūrimo etapas. Jis labai svarbus tiek vertinant tiek ir užtikrinant sistemos kokybę. Testavimo metu ieškoma klaidų programų sistemos kode, sistemos funkcionavime. Testavimo metu taip pat tikrinamas sistemos funkcionavimas įvairiomis sąlygomis, pavyzdžiui esant įvairiam serverio apkrautumui ar resursų poreikiui, testuojama vartotojo sąsaja bei atliekami kiti sistemos testavimai.

Atlikę testavimus ir palyginę testavimo rezultatus, galime teigti, kad pagal susidarytus testus programa veikia gerai. Tačiau kaip jau visiems yra žinoma, nėra tokio testo, kuris pašalintų visas sistemos klaidas. Testavimas buvo vykdomas keliais etapais kaip ir buvo

suplanuota: iš pirmo buvo testuojamos atskiros sistemos dalys, o paskui buvo testuojamos jau apjungtos sistemos komponentės.

3.4. Kokybės analizė

Šioje dalyje pateikiami programų sistemos kokybės vertinimo tikslai ir kriterijai, kokybės užtikrinimo ir vertinimo procesas, vertinimo rezultatai ir išvados.

3.4.1. Kokybės vertinimo tikslai

Programų sistemos kokybės vertinimui keliami tikslai:

- ✓ užtikrinti, kad sistema atitiktų specifikaciją;
- ✓ sumažinti klaidų skaičių sistemoje;
- ✓ įvertinti sistemos priimtinumą vartotojui.
- ✓ įvertinti sistemos funkcionavimą įvairiomis sąlygomis;
- ✓ įvertinti sistemos vystymo ir palaikymo galimybes;

3.4.2. Kokybės vertinimo kriterijai

Norint įgyvendinti kokybės vertinimo tikslus bei įvertinti ir užtikrinti kuriamos programų sistemos kokybę, jos vertinimui parinkti tokie kriterijai:

- **Korektiškumas** – laipsnis, kuriuo programa atitinka specifikaciją. Ar yra visos numatytos funkcijos.
- **Efektivumas** – kaip efektyviai programa panaudoja išteklius. Sistemos veikimas esant didelėms apkrovoms, kai yra didelis resursų naudojimas ar su sistema vienu metu dirbančių vartotojų skaičius.
- **Testuojamumas** – kaip greitai galima nustatyti, ar sistema veikia teisingai ar ne.
- **Pakartotinis panaudojimas** – ar galima sistemos komponentus panaudoti kuriant kitas sistemas arba naujus sistemos modulius.
- **Saugumas** – kaip išpildyti sistemai keliami saugumo reikalavimai, tokie kaip autorizacija, resursų prieigos nustatymas ir pan.
- **Patikimumas** – laipsnis kuriuo programa atlieka jai patikėtas funkcijas ir neatlieka jai nebūdingų funkcijų. Tai gali būti laiko tarpas tarp klaidų pasirodymų arba sistemos „lūžimų“.
- **Sąveikos galimybės** – kaip lengva programinę sistemą susieti su kitomis programinėmis sistemomis (čia įeina import, export ir t.t.).

- ***Panaudojamumas*** – ar lengva ir patogiu su programa dirbti. Ar vartotojai lengvai ir greitai perpranta sistemą.
- ***Palaikomumas*** – pastangos reikalingos aptikti ir ištaisyti programos defektus. Čia taip pat vertinamos pastangos, kurios reikalingos, norint įdiegti programos pakeitimus (atnaujinimas pagal kintančius reikalavimus).

3.4.3. Kokybės užtikrinimo ir vertinimo procesas

Kokybės užtikrinimo ir vertinimo procesą sudaro tam tikri etapai, kurie buvo atliekami vykdant projektą. Toliau pateikiami šių etapų aprašymai.

Sistemos specifikacijų peržiūros

Sistemos specifikacijų peržiūros metu buvo stengiamasi aptikti galimas sistemos logikos, funkcionalumo, architektūros bei kitas klaidas. Šios peržiūros metu buvo taisomi neatitikimai.

Šiame etape buvo peržiūrimos šios sistemos specifikacijos:

- ✓ reikalavimų specifikacija
- ✓ architektūros specifikacija
- ✓ detalios architektūros specifikacija
- ✓ vartotojo dokumentacijos
- ✓ testavimo plano peržiūra

Testavimas

Testavimas yra atskiras šios programų sistemos kūrimo etapas. Jis labai svarbus tiek vertinant tiek ir užtikrinant sistemos kokybę. Testavimo metu ieškoma klaidų programų sistemos kode, sistemos funkcionavime. Testavimo metu taip pat tikrinamas sistemos funkcionavimas įvairiomis sąlygomis, pavyzdžiui esant įvairiam serverio apkrautumui ar resursų poreikiui, testuojama vartotojo sąsają bei atliekami kiti sistemos testavimai. Sistemos testavimui yra naudojami įvairūs sistemos testavimo metodai, pavyzdžiui, vienetų testavimas, integracinis testavimas ir kiti.

Taigi bendru atveju testavimas leidžia ne tik įvertinti įvairias sistemos kokybės charakteristikas, bet ir dalinai užtikrinti aukštesnę programų sistemos kokybę.

Klaidų taisymas

Kaip ir testavimas, šis etapas yra atskiras programų sistemos kūrimo etapas ir turi didelę įtaką sistemos kokybės užtikrinimui ir įvertinimui. Jis leidžia įvertinti sistemos palaikomumą,

defektų pašalinimo sąnaudas, o taip pat gali įtakoti ir kitus sistemos kokybę nusakančius kriterijus.

3.4.4. Kokybės vertinimas

Vertinant sistemą pagal pasirinktus kriterijus, bus naudojama skaitinė įvertinimo išraiška. Galimas balų skaičius kiekvienam kriterijui – nuo 0 iki 10.

Korektiškumas

Testavimo metu buvo testuojamos ir tikrinamos sistemos funkcijos ir jų veikimas. Buvo tikrinama, ar sistemos funkcijos atitinka savo specifikaciją. Atlikus testavimus, buvo įsitikinta, jog sistemos funkcijos atitinka specifikaciją.

Galutinis įvertinimas: 10.

Efektyvumas

Testavimo metu buvo bandomas sistemos veikimas esant didesniems nei vidutiniai resursų naudojimo poreikiams, pavyzdžiui, keletu vartotojų paieškos vykdymas vienu metu. Atlikus testavimą, buvo įsitikinta, jog sistema veikia efektyviai.

Galutinis įvertinimas: 9.

Testuojamumas

Buvo vertinamos darbo ir laiko sąnaudos, kurių reikia, norint ištestuoti sistemos veikimą, įvertinti, ar sistema veikia taip kaip turėtų, ar nėra klaidų duomenų apdorojime ir kitose funkcijose. Galima pasakyti, jog sistema bendru atveju yra pakankamai nesunkiai testuojama.

Galutinis įvertinimas: 8.

Pakartotinis panaudojimas

Buvo vertinama sistemos modulių realizacija ir galimybė juos pritaikyti kuriant kitas sistemas arba naujus modulius. Kadangi sistema realizuota naudojant objektinį programavimą, tai jos modulių panaudojimas kituose darbuose yra galimas.

Galutinis įvertinimas: 9.

Saugumas

Buvo tikrinamas sistemai keliamų saugumo reikalavimų realizavimas. Šiuo atveju buvo tikrinama sistemos autorizacijos mechanizmo realizacija bei resursų prieigos ir kontrolės reikalavimų išpildymai.

Galutinis įvertinimas: 10.

Patikimumas

Buvo žiūrima, kaip ilgai sistema gali veikti „nelūžusi“. Šiuo atveju sistema buvo stebima viso testavimo bei validavimo metu. Per tą laikotarpį sistema nei karto „nenulūžo“. Testavimo metu taip pat buvo tikrinama, ar sistema atlieka visas numatytas funkcijas bei neatlieka nebūdingų arba nepageidaujamų funkcijų. Atlikus sistemos testavimus, buvo įsitikinta, jog sistema neatlieka jai nebūdingų funkcijų, o visas numatytas funkcijas atlieka teisingai.

Galutinis įvertinimas: 10.

Sąveikos galimybės

Vertinant sistemą pagal šį kriterijų, buvo tiriama galimybė susieti sistemą su kitomis programų sistemomis. Šiuo atveju buvo vertinama galimybė importuoti ir eksportuoti duomenis kitais formatais ir pan.

Galutinis įvertinimas: 9.

Panaudojamumas

Buvo vertinama, kaip lengva naudotis sistema ir jos teikiamomis funkcijomis. Vertinimo metu buvo analizuojami tokie sistemos funkcijų veikimo aspektai:

- Ar vartotojai supranta sistemos funkcijas ir kaip jomis naudotis?
- Ar vartotojams aiškus sistemos veikimas, procesai ir jų rezultatai?
- Ar pakankamai aiški ir suprantama vartotojo pagalba ir kt.?

Galutinis įvertinimas: 8.

Palaikomumas

Buvo vertinama, kiek laiko vidutiniškai reikia, norint ištaisyti sistemoje aptiktą defektą. Defektus galima suskirstyti į tam tikras grupes pagal jų daromą žalą, nepatogumus vartotojams, klaidingą veikimą, ištaisymo sąnaudas, svarbą ir kitus kriterijus. Priklausomai nuo defektų grupės, jų pašalinimui yra skiriamas tam tikras laikas. Bendru atveju defektų šalinimas sistemoje yra pakankamai gerai ir efektyviai realizuojamas.

Atnaujinimų diegimas sistemoje priklauso nuo diegiamų pakeitimų masto.

Galutinis įvertinimas: 9.

3.4.5. Kokybės vertinimo rezultatai

Kokybės vertinimo rezultatų lentelė:

Lentelė Nr. 5 Kokybės vertinimo rezultatai

Nr.	Kriterijus	Įvertinimas
1	Korektiškumas	10
2	Efektyvumas	9
3	Testuojamumas	8
4	Pakartotinis panaudojimas	9
5	Saugumas	10
6	Patikimumas	10
7	Sąveikos galimybės	9
8	Panaudojamumas	8
9	Palaikomumas	9
Bendras įvertinimas		9,0

3.4.6. Išvados

Kaip matyti iš kokybės įvertinimų, sistema realizuota gerai ir pakankamai kokybiškai. Sistema atitinka specifikaciją, vartotojo poreikius, yra pakankamai nesunkiai palaikoma ir vystoma, o jos komponentai gali būti naudojami kuriant kitas sistemas.

4. TIRIAMOJI EKSPERIMENTINĖ DALIS

4.1. Tikslas

Eksperimento tikslas – išsiaiškinti kuris paieškos variklis „Lucene“ ar „Zebra“ yra tinkamesnis naudoti paieškos sistemoje, bei pabandyti optimizuoti jo veikimą.

4.2. Tyrimo metodika

Eksperimentui atlikti parenkami tokie eksperimentiniai atvejai:

1. Paieškos laiko priklausomybė nuo indekso dydžio – vykdoma paieška indekse sudarytam iš skirtingo kiekio dokumentų.
2. Indekso dydžio priklausomybė nuo saugykloje saugomų dokumentų dydžio.
3. Dokumentų indeksavimo laiko priklausomybė nuo indeksuojamų dokumentų kiekio.

Eksperimentui atlikti sukuriama specialios programos, kurios realizuoja atskirus eksperimentinius atvejus. Šios programos vykdymo metu fiksuoja sistemos veikimo greitį ir rezultatus su skirtingais paieškos varikliais.

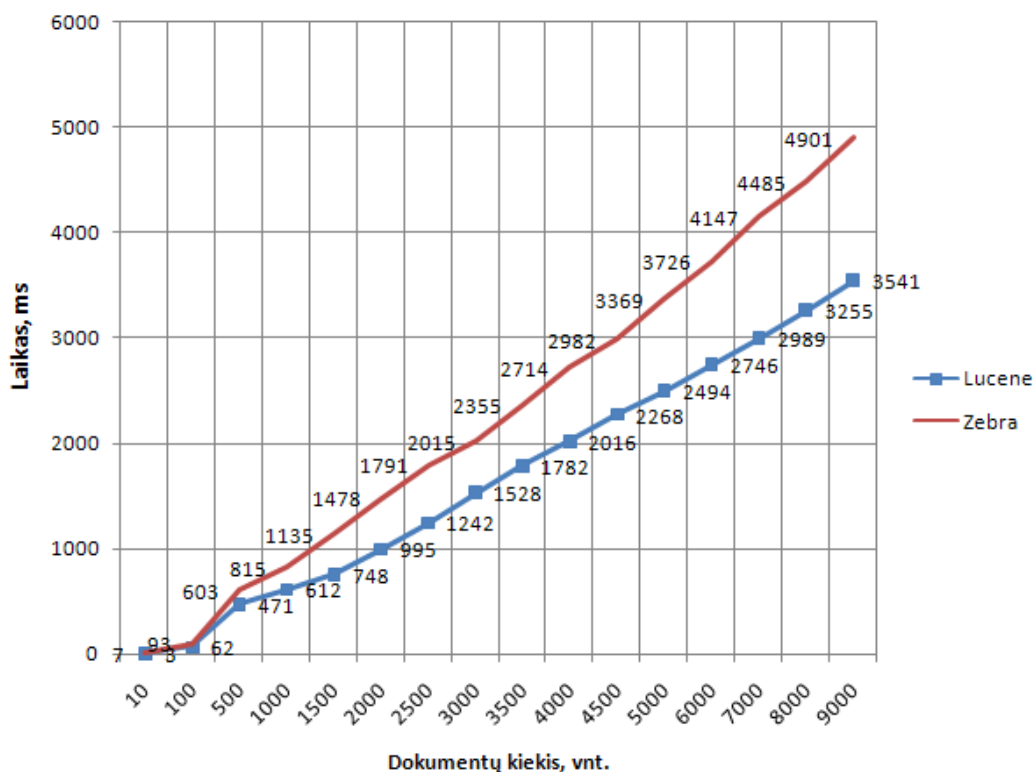
4.3. Rezultatai

- Paieškos laiko priklausomybė nuo indekso dydžio

Lentelė Nr. 6 Paieškos laiko priklausomybės nuo indekso dydžio eksperimento rezultatai

Eksperimentas	Suindeksuotų dokumentų kiekis (vnt.)	Paieškos laikas, naudojant Lucene paieškos variklį (ms)	Paieškos laikas, naudojant Zebra paieškos variklį (ms)
1	10	3	7
2	100	62	93
3	500	471	603
4	1000	612	815
5	1500	748	1135
6	2000	995	1478
7	2500	1242	1791
8	3000	1528	2015
9	3500	1782	2355
10	4000	2016	2714
11	4500	2268	2982
12	5000	2494	3369
13	6000	2746	3726
14	7000	2989	4147
15	8000	3255	4485
16	9000	3541	4901

Paieškos laiko priklausomybė nuo indekso dydžio



15 Pav. Paieškos laiko priklausomybės nuo indekso dydžio diagrama

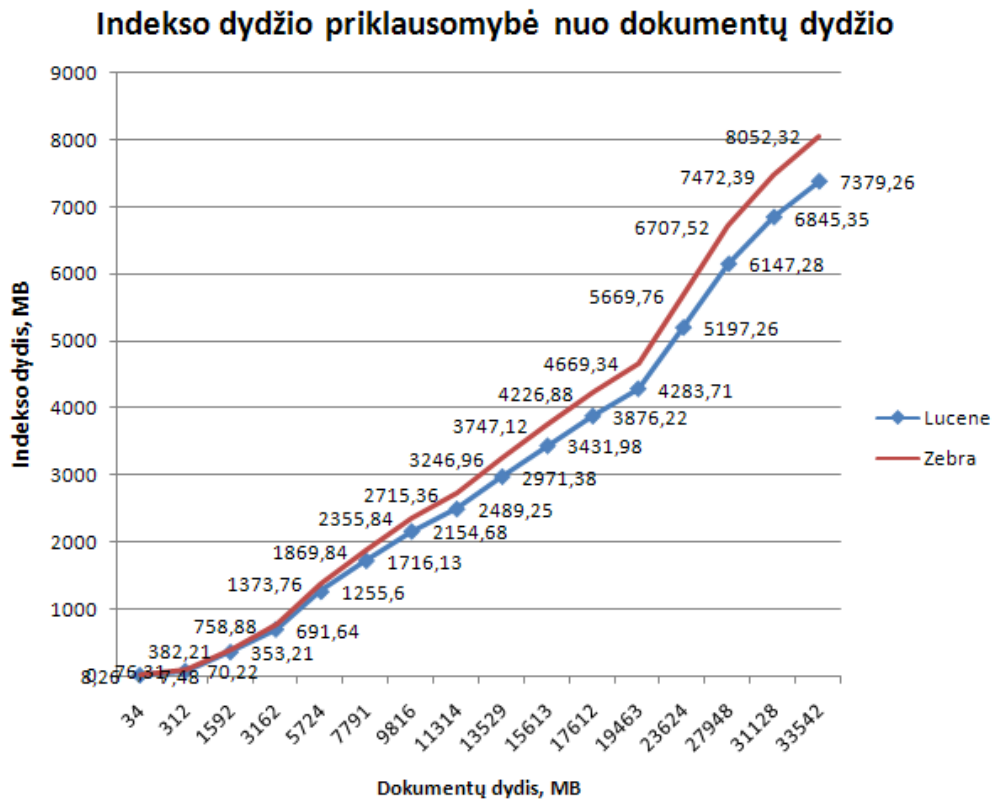
Kaip matome iš eksperimento rezultatų lentelės ir diagramos, indeksavimui ir paieškai naudojant Lucene paieškos variklį, paieška vykdoma vidutiniškai 1,37 karto greičiau, nei tokia pati paieška naudojant Zebra paieškos variklį.

- **Indekso dydžio priklausomybė nuo saugykloje saugomų dokumentų dydžio.**

Lentelė Nr. 7 Indekso dydžio priklausomybės nuo saugomų dokumentų dydžio eksperimento rezultatai

Eksperimentas	Indeksuojamų dokumentų dydis (MB)	Indekso dydis, naudojant Lucene paieškos variklį (MB)	Indekso dydis, naudojant Zebra paieškos variklį (MB)
1	34	7,48	8,26
2	312	70,22	76,31
3	1592	353,21	382,21
4	3162	691,64	758,88
5	5724	1255,6	1373,76
6	7791	1716,13	1869,84
7	9816	2154,68	2355,84
8	11314	2489,25	2715,36
9	13529	2971,38	3246,96
10	15613	3431,98	3747,12
11	17612	3876,22	4226,88

12	19463	4283,71	4669,34
13	23624	5197,26	5669,76
14	27948	6147,28	6707,52
15	31128	6845,35	7472,39
16	33542	7379,26	8052,32



16 Pav. Indekso dydžio priklausomybės nuo dokumentų dydžio diagrama

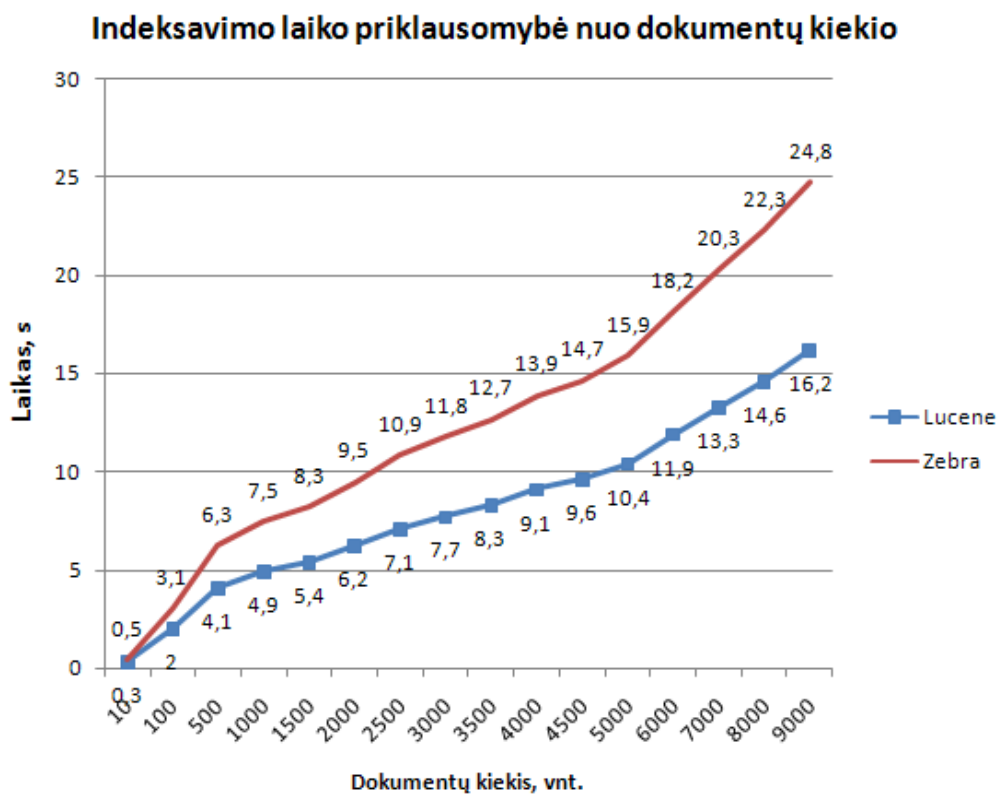
Kaip matome iš eksperimento rezultatų lentelės ir diagramos, indeksavimui ir paieškai naudojant Lucene paieškos variklį, paieškos indeksas užima vidutiniškai 1,09 karto mažiau, vietos nei naudojant Zebra paieškos variklį.

- **Dokumentų indeksavimo laiko priklausomybė nuo indeksuojamų dokumentų kiekio.**

Lentelė Nr. 8 Dokumentų indeksavimo laiko priklausomybės nuo indeksuojamų dokumentų kiekio eksperimento rezultatai

Eksperimentas	Suindeksuotų dokumentų kiekis (vnt.)	Indeksavimo laikas, naudojant Lucene paieškos variklį (s)	Indeksavimo laikas, naudojant Zebra paieškos variklį (s)
1	10	0,3	0,5
2	100	2	3,1
3	500	4,1	6,3
4	1000	4,9	7,5
5	1500	5,4	8,3

6	2000	6,2	9,5
7	2500	7,1	10,9
8	3000	7,7	11,8
9	3500	8,3	12,7
10	4000	9,1	13,9
11	4500	9,6	14,7
12	5000	10,4	15,9
13	6000	11,9	18,2
14	7000	13,3	20,3
15	8000	14,6	22,3
16	9000	16,2	24,8



17 Pav. Indeksavimo laiko priklausomybės nuo dokumentų kiekio diagrama

Kaip matome iš eksperimento rezultatų lentelės ir diagramos, indeksavimui ir paieškai naudojant Lucene paieškos variklį, indekso sudarymo laikas vidutiniškai 1,5 karto geresnis, nei naudojant Zebra paieškos variklį.

Atlikę šiuos bandymus galime daryti išvadą, kad geriau naudoti Lucene paieškos variklį. Toliau bandysime optimizuoti šio paieškos variklio darbą.

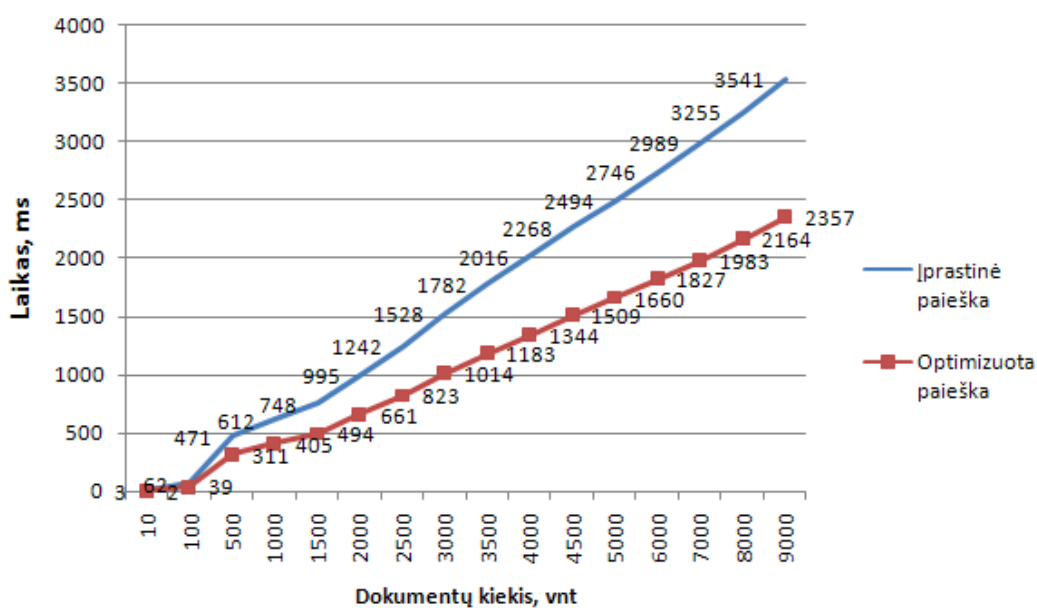
Vykiant eksperimentus buvo pastebėta, kad į e. dokumento metaduomenis įtraukus papildomus raktinius žodžius iš e. dokumento turinio, paieškos greitis padidėja, o indekso dydis pasikeičia labai nežymiai. Atlikus paieškos laiko priklausomybės nuo dokumentų kiekio priklausomybės tyrimą į metaduomenis įtraukiant raktinius paieškos žodžius ir palyginus juos

su anksčiau gautais rezultatais, buvo gauti tokie eksperimento rezultatai. Jie atvaizduoti 9 lentelėje bei diagramoje. (18 pav.)

Lentelė Nr. 9 Paieškos laiko priklausomybės nuo dokumentų kiekio, paieškai naudojant optimizuotą ir neoptimizuotą indeksą, eksperimento rezultatai

Eksperimentas	Suindeksuotų dokumentų kiekis (vnt.)	Paieškos laikas, naudojant Lucene paieškos variklį (ms)	Paieškos laikas, naudojant Lucene paieškos variklį, įtraukus ieškomus žodžius į metaduomenis (ms)
1	10	3	2
2	100	62	39
3	500	471	311
4	1000	612	405
5	1500	748	494
6	2000	995	661
7	2500	1242	823
8	3000	1528	1014
9	3500	1782	1183
10	4000	2016	1344
11	4500	2268	1509
12	5000	2494	1660
13	6000	2746	1827
14	7000	2989	1983
15	8000	3255	2164
16	9000	3541	2357

Optimizuotos ir neoptimizuotos paieškos palyginimas



18 Pav. Paieškos laiko priklausomybės nuo dokumentų kiekio, paieškai naudojant optimizuotą ir neoptimizuotą indeksą, diagrama

Kaip matome iš eksperimento rezultatų lentelės ir diagramos, paieškai naudojant optimizuotą indeksą, paieška įvykdoma vidutiniškai 1,5 karto greičiau nei ieškant neoptimizuotame indekse.

4.4. Išvados

Susumavę eksperimento rezultatus galime teigti, kad paieškos sistemoje geriau naudoti Lucene paieškos variklį, nes paieška vykdoma vidutiniškai 1,37 karto greičiau, indeksavimas vidutiniškai 1,5 karto greičiau, o indeksas užima vidutiniškai 1,09 karto mažiau vietos lyginant su Zebra paieškos varikliu. Taip pat buvo nustatyta, kad į e. dokumento metaduomenis įtraukus paieškos raktinius žodžius, tuo pačiu optimizuojant sudaromą indeksą, paieška vykdoma vidutiniškai 1,5 karto greičiau.

5. IŠVADOS

- Darbe buvo išanalizuota keletas paieškos sistemų, bei įvardinti jų privalumai bei trūkumai. Palygintos sistemos kūrimo technologijos.
- Eksperimentiškai buvo tiriama kokią paieškos variklį naudoti sistemoje. Buvo prieita išvados, kad šiai sistemai tinkamiausias yra Lucene paieškos variklis.
- Eksperimentiškai buvo nustatytas galimas paieškos vykdymo optimizavimas.
- Suprojektuota ir realizuota paieškos sistema, leidžianti vykdyti paiešką tarp elektroninių dokumentų. Ji ypatinga tuo, kad vykdo paiešką ir dokumento tekste, o ne tik jo metaduomenyse.

6. LITERATŪRA

- [1] Apache Lucene - Query Parser Syntax [Žiūrėta 2007 11 22], prieiga internete: <<http://lucene.apache.org/java/docs/>>
- [2] eLABa specifikacija. Kaunas, 2006 m.
- [3] Fedora Search Interface Documentation [Žiūrėta 2007 11 22], prieiga internete: <<http://www.fedora.info/download/2.1.1/userdocs/server/webservices/search/index.html>>
- [4] JAVA technologies [Žiūrėta 2007 11 22], prieiga internete: <<http://java.sun.com/>>
- [5] JBoss Application Server [Žiūrėta 2007 11 22], prieiga internete: <<http://labs.jboss.com/jbossas/docs/index.html>>
- [6] Publications from KTH [Žiūrėta 2007 11 22], prieiga internete: <<http://www.diva-portal.org/kth/index.xsql?lang=en>>
- [7] Seam - Contextual Components. A Framework for Enterprise Java [Žiūrėta 2007 11 22], prieiga internete: <<http://docs.jboss.com/seam/2.0.0.GA/reference/en/html/index.html>>
- [8] Simple Object Access Protocol - SOAP [Žiūrėta 2007 11 22], prieiga internete: <<http://webdesign.about.com/library/weekly/aa031802a.htm>>
- [9] Sun Servers [Žiūrėta 2007 11 22], prieiga internete: <<http://www.sun.com/servers/>>
- [10] Web Services - Axis [Žiūrėta 2007 11 22], prieiga internete: <<http://ws.apache.org/axis/>>
- [11] Zebra - User's Guide and Reference [Žiūrėta 2007 11 22], prieiga internete: <<http://www.indexdata.dk/zebra/doc/>>

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

Elektroninis dokumentas (e. dokumentas) – dokumentas saugantis tekstine, grafine ar kito pobūdžio informacija elektroniniame pavidale.

eLABa – Lietuvos mokslo ir studijų elektroninių dokumentų informacinė sistema. Tai informacinė sistema, kuri sudaro prielaidas suformuoti elektroninę Lietuvos akademinę biblioteką.

FEDORA – e. dokumentų talpyklų programinė įranga.

FEDORA objektas – FEDORA talpykloje esantis e. dokumentas ir jo metaduomenys.

PID – unikalus identifikatorius FEDORA sistemoje, pagal kurį galima surasti bet kurį e. dokumentą.

Pilnojo teksto paieška (full-text search) – paieškos tipas kuomet paieška yra atliekama paieškos objektų turinyje.

XML – išplėstinė žymių kalba – duomenų formatas struktūrizuotų dokumentų keitimuisi tinkle (*eXtensible Markup Language*).

8. PRIEDAI

1. PRIEDAS. SISTEMOS ĮDIEGIMO PATVIRTINIMO AKTAS

SISTEMOS ĮDIEGIMO PATVIRTINIMO AKTAS

2009 m. gegužės mėn. 19 d.

Kaunas

Kauno Technologijos universiteto Informatikos fakulteto Informacinių technologijų diegimo centro Bibliotekų programinės įrangos skyrius, atstovaujamas skyriaus vedėjo Viliaus Kučiuko, patvirtina, kad Antanas Gilaitis sėkmingai įdiegė ir perdavė informacijos paieškos sistemą elektroninėje bibliotekoje (eLABa).

Užsakovas

Vilius Kučiukas
Kauno Technologijos universitetas
Informatikos fakultetas
Informacinių technologijų diegimo centras
Bibliotekų programinės įrangos skyrius

(parašas)