

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Optimizavimo paieškos sistemoms principų taikymas sudėtinguose tinklalapiuose

Search engine optimization for complex websites

Bakalauro darbas

Atliko: Evaldas Morkvėnas (parašas)

Darbo vadovas: prof. Saulius Minkevičius (parašas)

Darbo recenzentas: dr. Algirdas Lančinskas (parašas)

Vilnius – 2016

SANTRAUKA

Vystant bet koki verslą internete vienas esminių iššūkių yra pritraukti kuo daugiau lankytojų. Siekiant, kad tinklalapis būtų gausiai lankomas, reikia, kad vartotojai žinotų apie jį, o nežinant – prirėkus lengvai surastų. Todėl vienas svarbiausių žingsnių, kuriant gausiai lankomą interneto tinklalapį, yra kruopščiai suplanuotas ir sėkmingai atliktas bei nuolat tobulinamas tinklalapio optimizavimas paieškos sistemoms.

Šiame darbe siekiama išanalizuoti pagrindinius Google paieškos sistemos veikimo principus bei komponentus. Taip pat darbe siekiama išanalizuoti bei pateikti informaciją apie patį optimizavimo paieškos sistemoms procesą, pateikti esminius optimizavimo paieškos sistemoms metodus, išsiaiškinti dažniausiai daromas klaidas optimizuojant sudėtingus tinklalapius ir pateikti jų sprendimo būdus.

Sudėtingais tinklalapiais darbe laikomi tinklalapiai, turintys daugiau nei šimtą puslapių, kurių turinys dinamiškai keičiasi ar yra generuojamas lankytojų. Siekiant labiau įsigilinti į sudėtingų tinklalapių problematiką ir atskleisti darbo temą iš praktinės pusės buvo suprojektuotas ir realizuotas sudėtingo tinklalapio prototipas, paremtas AngularJS karkasu. Prototipui buvo pritaikyti darbe analizuoti optimizavimo paieškos sistemoms metodai.

Raktiniai žodžiai: sudėtingi tinklalapiai, optimizavimas paieškos sistemoms, SEO, SPA, JavaScript kalbos karkasas, optimizavimas.

SUMMARY

One of the biggest challenges while trying to create a successful business website is to attract as much visitors as possible. That is when search engine optimization (SEO) comes in place. SEO is a critical step in order to make the website easily founded. That is exactly why well planned and successfully executed SEO strategy is one of the key factors of running online business.

This thesis aims to analyze the core principles and components of Google search engine. Also it aims to analyze and break down the process of search engine optimization into steps, provide essential optimization methods, find out most common issues related SEO and provide possible solutions.

Term complex websites in this thesis is used to describe websites which contain more than a hundred pages, with dynamically changed content or content generated by website users. In order to look at this topic from a more practical perspective, author implemented the prototype of complex website based on AngularJS framework and applied optimization methods analyzed in thesis.

Keywords: complex websites, search engine optimization, SEO, SPA, JavaScript framework, optimization.

TURINYS

ĮVADAS.....	6
1. GOOGLE PAIEŠKOS SISTEMA	8
1.1. Komponentai.....	8
1.2. Organiniai paieškos rezultatai.....	10
1.3. Neorganiniai paieškos rezultatai	10
2. OPTIMIZAVIMO PAIEŠKOS SISTEMOMS PROCESAS	12
2.1. Verslo analizė	12
2.2. Planavimas	13
2.3. Realizacija.....	15
2.4. Rezultatų stebėjimas	17
2.5. Įsivertinimas.....	18
2.6. Priežiūra	18
3. SUDĖTINGI TINKLALAPIAI	19
3.1. Apibrėžimas	19
3.2. Rinkmenų suspaudimas serveryje.....	19
3.3. Dinaminės nuorodos	19
3.4. Nuorodų reikšmingumas.....	20
3.5. Vidinės nuorodos	21
3.6. Tinklalo struktūra.....	21
3.7. Plagijuotas turinys.....	23
4. OPTIMIZAVIMO PAIEŠKOS SISTEMOMS PRINCIPŲ PRITAIKYMAS AngularJS KARKASUI.....	24
4.1. Asinchroninis JavaScript ir XML programavimas (AJAX)	24
4.2. SPA samprata.....	24
4.3. AngularJS karkasas	25
4.4. Kuriamos sistemos aprašymas	25
4.5. Reikalavimai sistemai	25
4.6. Naudojamos technologijos.....	26
4.7. Realųjų vartotojų atskirimas nuo paieškos sistemos robotų	26
4.8. Optimizavimas Open graph protokolui.....	29
4.9. Svetainės žemėlapių generavimas	30
4.10. Navigacijos optimizavimas.....	31

4.11. Žymės „<title>“ optimizavimas.....	31
4.12. Žymės „<meta>“ optimizavimas	31
4.13. Sistemos prototipas	32
REZULTATAI IR IŠVADOS.....	35
ŠALTINIAI.....	37
SANTRUPOS	39

IVADAS

XXI amžiuje tikriausiai jau niekam nebereikia pasakoti apie interneto svarbą ir jo paplitimą. Technologijoms besivystant į priekį, internetas tampa pasiekiamas vis didesnei auditorijai, kurios nebelemia nei socialinė padėtis, nei vartotojų amžius. Be interneto gyvenimą sunkiai įsivaizduoja daugelis tiek jaunesniosios, tiek vyresniosios kartos vartotojų, nors galbūt jų interneto naudojimo tikslai ir įpročiai kardinaliai skiriasi.

Internetu teikiamos paslaugos bei pardavimai neretai sudaro didesnę dalį įmonių pajamų, negu prekės ir paslaugos, kurias įsigyjame ne virtualioje erdvėje. Besikuriančios naujos įmonės neįsivaizduoja savo įvaizdžio kūrimo be internetinio tinklalapio. Vis dažniau pirmesnė dienos šviesą išvysta internetinė parduotuvės versija negu realioji. Norėdami atidaryti parduotuvę ar konkrečios paslaugos įmonę, į kurią kasdien užeitų daugybė potencialių klientų, reikėtų atsižvelgti į daugybę įvairių aspektų: fizinę vietą, kuri būtų patogi bei lengvai pastebima klientams; interjerą, kuris atkreiptų dėmesį, malonų klientų aptarnavimą, reklamos kampaniją bei daugelį kitų dalykų. Norint atidaryti gausiai lankomą internetinę parduotuvę ar tinklalapį, kuriame teiktume paslaugas, reikėtų atsižvelgti į kiek kitokius aspektus. Norint, kad internetinė parduotuvė būtų gausiai lankoma, reikia, kad vartotojai žinotų apie ją, o nežinant – prireikus lengvai surastų. Statistiniai duomenys teigia, kad net daugiau nei 90 procentų interneto lankytojų į tokius tinklalapius patenka naudodami paieškos sistemas [Bia11]. Tai reiškia, kad vienas svarbiausių žingsnių kuriant gausiai lankomą interneto tinklalapį yra kruopščiai suplanuotas ir sėkmingai atliktas bei nuolat tobulinamas tinklalapio optimizavimas paieškos sistemoms. Pagrindinis tikslas, kurio siekiama optimizuojant svetainę paieškos sistemoms yra, kad vartotojas per paieškos sistemą ieškodamas prekės, paslaugos ar informacijos, kurią pateikia mūsų tinklalapis, nuorodą į mūsų internetinę parduotuvę pamatytų kuo aukštesnėje paieškos sistemos rezultatų vietoje. Tačiau tie patys minėtieji statistikos šaltiniai teigia, kad net 75 procentų vartotojų domina tik pirmojo rezultatų puslapio duomenys [Bia11]. Žinant tai, kad internetinių svetainių skaičius 2016 metais jau perkopė 1 mlrd. ribą, o didžiausią rinkos dalį užimančios paieškos sistemos Google viename rezultatų puslapyje pateikiama 10–20 tinklalapių adresų, galima įsivaizduoti kokios milžiniškos konkurencijos sulauktume norint, kad mūsų kuriama internetinė parduotuvė atsidurtų pirmoje Google paieškos rezultatų vietoje, pagal tokį raktažodį, kuris laikomas konkurencingu [Int16]. Išties optimizavimo paieškos sistemoms procesas yra neatsiejama sėkmingos internetinės svetainės dalis, reikalaujanti nemažai kantrybės ir pastangų, todėl verta apie jį pagalvoti iš pradžių, dar prieš pradėdant kurti tinklalapį.

Sudėtingais tinklalapiais šiame darbe laikomi tinklalapiai, turintys daugiau nei šimtą puslapių ir kurių turinys dinamiškai keičiasi ar yra generuojamas lankytojų. Tokie tinklalapiai

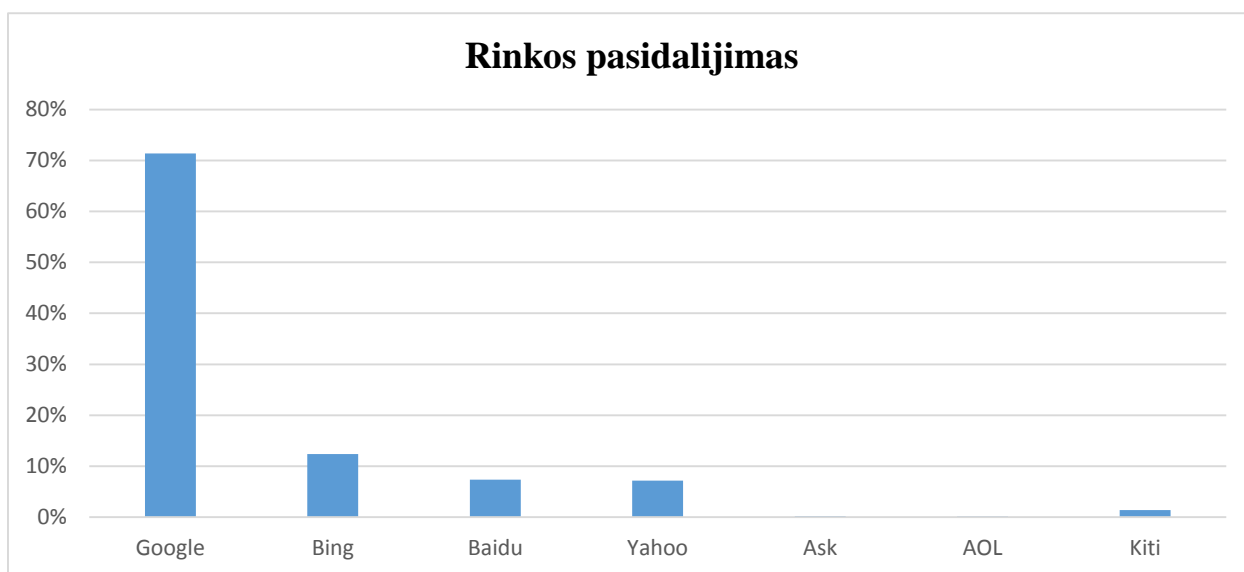
paprastai yra komercinės paskirties, todėl ši tema yra itin naudinga siekiant pritraukti didesnę vartotojų srautą, bei padidinti pelną.

Bakalaurinio darbo tikslas yra išanalizuoti optimizavimo paieškos sistemoms principus bei išsiaiškinti ar jie yra pritaikomi kuriant SPA (angl. *Single Page Application*) tipo tinklalapius, sukuriant sudėtingo tinklalapio prototipą, paremtą AngularJS karkasu. Darbo uždaviniai:

1. Išanalizavus pristatyti pagrindinius Google paieškos sistemos veikimo principus, bei komponentus.
2. Išanalizuoti optimizavimo paieškos sistemoms procesą, išskaidant jį į atskiras dalis.
3. Pateikti esminius tinklalapių optimizavimo paieškos sistemoms metodus.
4. Išsiaiškinti specifinius optimizavimo paieškos sistemoms aspektus, į kuriuos reikėtų atsižvelgti optimizuojant sudėtingus tinklalapius.
5. Išanalizavus galimas sudėtingų tinklalapių struktūras, parinkti ir pagrįsti tinkamiausią struktūrą sudėtingam tinklalapiui.
6. Realizuoti SPA tipo sudėtingo tinklalapio prototipą, paremtą AngularJS karkasu.
7. Realizuotam prototipui pritaikyti optimizavimo paieškos sistemos principus.

1. GOOGLE PAIEŠKOS SISTEMA

2016 metų balandžio mėnesio duomenimis paieškos sistemų rinkoje ryškia lydere išlieka Google paieškos sistema, užimanti net 71,35 procentų rinkos dalies (1 pav.). Tuo tarpu artimiausias konkurentas Bing užima vos 12,37 procentų rinkos dalies [Net16]. Esant tokiam rinkos pasiskirstymui, dažniausiai atliekant optimizavimą paieškos sistemoms orientuojamasi būtent į Google paieškos sistemą. Dėl šios priežasties darbo autorius pateikiamuose pavyzdžiuose orientuosis į rinkos lyderę, nors pagrindiniai optimizavimo principai tinka visoms paieškos sistemoms.



1 pav. Rinkos pasidalijimas [Net16]

1.1. Komponentai

1.1.1. Robotas

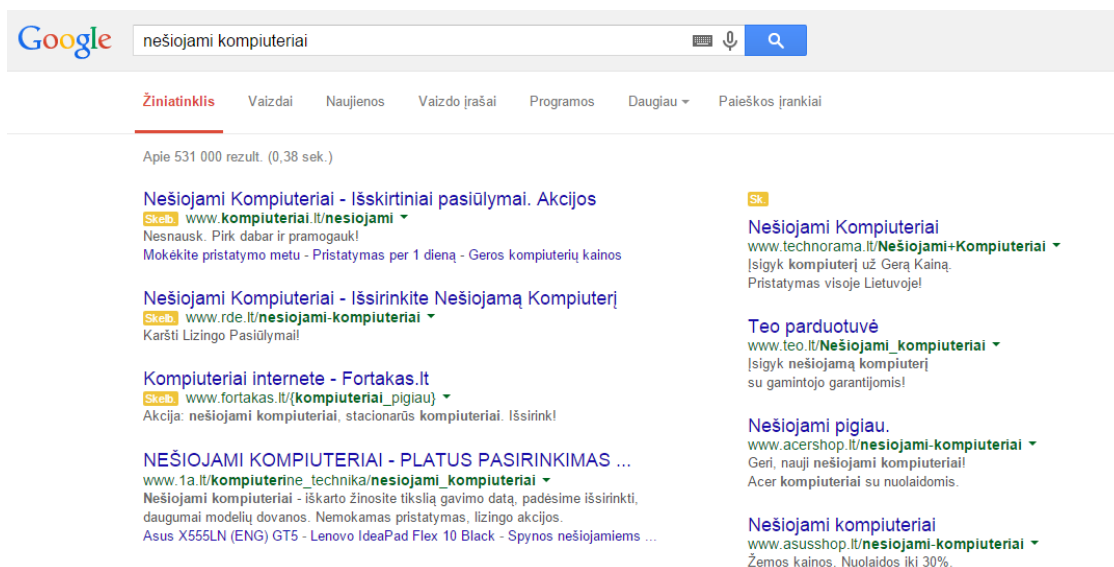
Robotas (angl. *Spider, robot, bot, crawler*) – robotas, voras, botas, visais šiais pavadinimais vadinamas komponentas yra automatizuota programinė įranga, kurią turi kiekviena paieškos sistema. Robotas yra skirtas tam, kad keliautų per tinklalapius ir surinktų kiek įmanomą daugiau informacijos apie kiekvieną tinklalapį. Juos kaip ir įprastus vartotojus galima sekti serverio išklotinėse arba naudojant analitinę programinę įrangą. Robotai keliaudami pasiekia ne tik puslapių rinkmenas, bet ir daugelį kitų, įskaitant „robots.txt“, „sitemap.xml“. Paieškos sistemos robotai keliauja per tinklalapius atsitiktine tvarka. Nuo tinklalapio šviežumo, bei turinio dydžio priklauso kaip dažnai tinklalapis aplankomas robotų. Tame pačiame tinklalapyje robotai gali apsilankyti ne vieną kartą. Po pirmo apsilankymo tinklalapyje, robotai po kurio laiko grįžta patikrinti ar neatsirado turinio pasikeitimų.

1.1.2. Indeksatorius

Indeksatorius (angl. *Indexer*) – komponentas atliekantis indeksavimo procesą. Robotų surinktą informaciją apdoroja indeksatorius, kuris yra atsakingas už tai, kad duomenų bazėje iš surinktų duomenų būtų saugomi svarbiausi ir tiksliausiai konkretų tinklalapį apibūdinantys duomenys. Šio proceso pagalba paieškos sistema prireikus gali itin greitai pasiekti duomenis pagal tikslų paieškos kriterijų. Nors robotai stengiasi aplankyti kiek įmanoma daugiau tinklalapių, tačiau ne visi tampa suindeksuotais. Indeksavimas yra stipriai susijęs su konkrečios paieškos sistemos variklio algoritmais, todėl formulė, pagal kurią jis vykdomas, yra itin slapta ir tuo pačiu yra vienas iš labiausiai intriguojančių optimizavimo paieškos sistemoms aspektų. Indeksavimo procesas nėra visiškai automatinis. Rankiniu būdu perimamas valdymas, kai norima iš paieškos rezultatų pašalinti potencialiai kenksmingus tinklalapius, šlamštą (angl. *Spam*) arba norint pakelti tinklalapio reitingą priklausomai nuo jo prekinio ženklo vertės ar kitų ypatybių.

1.1.3. Paieškos sąsaja

Paieškos sąsaja – tai paieškos sistemos dalis, per kurią vartotojai pateikia užklausas ir patenka į paieškos sistemos rezultatų puslapius. Google paieškos sistema leidžia naudotis įvairiais filtrais ir „susiaurinti“ vartotojo paiešką pagal datą, kalbą, vietą, rezultatų tipą ir daug kitų parametrų. Rezultatų lange vartotojui pateikiami tiek organiniai, tiek neorganiniai rezultatai (plačiau 2.2, 2.3 skyriuose). Rezultatų paieškos lange (2 pav.) įvedus raktažodį „nešiojami kompiuteriai“ kairėje pusėje pateikiami trys neorganiniai (remėjų) rezultatai, po jų einantys visi kiti – organiniai. Dešinėje pusėje pateikiami neorganiniai rezultatai. Google neorganinius rezultatus pažymi geltona etikete „Skelb.“, o visą neorganinių rezultatų sekciją etikete „Sk.“. Taip leisdami vartotojui suprasti pateikiamų rezultatų tipą.



2 pav. Google paieškos rezultatų langas

1.2. Organiniai paieškos rezultatai

Paieškos rezultatai skirstomi į du tipus: organinius ir neorganinius. Organiniai rezultatai yra tie, kuriuos Google paieškos sistema pateikia pagal savo kokybės indeksą. Būtent todėl reikalingas optimizavimas paieškos sistemoms, kadangi dėl jo paieškos sistemos geriau supranta koks turinys yra pateikiamas tinklalapyje, kokia yra jo kokybė ir, ar šis turinys yra tas, kurio vartotojas ieško. Nepaisant daugybės faktorių iš tinklalapio pusės yra ir tokių, kurie nuo paties tinklalapio ir nepriklauso, pavyzdžiui vieta. Google rodo lokalizuotus paieškos rezultatus, tai reiškia, kad paieškos rezultatai tais pačiais raktažodžiais gali skirtis, kai paieška atliekama iš skirtingų regionų [Tak11]. Pirmoje lentelėje pateikiami duomenys tyrimo, kuriame buvo tiriamas vartotojų elgesys paieškos sistemose, akivaizdžiai matoma, kad vartotojai labiau pasitiki organiniais rezultatais, nes pirmoje pozicijoje esanti neorganinė nuoroda sulaukė vos 50 procentų peržiūrų, kai tuo tarpu tiek pat peržiūrų sulaukė organinių rezultatų nuorodos esančios 6-7 pozicijose [Jer10]. Tik kas dešimtas vartotojas pastebi neorganines nuorodas, esančias žemiau 4 pozicijos.

1 lentelė. Tyrimo rezultatai [Jer10]

Organiniai rezultatai		Neorganiniai rezultatai	
Reitingas	Procentai peržiūrėjo	Pozicija	Procentai peržiūrėjo
1	100%	1	50%
2	100%	2	40%
3	100%	3	30%
4	85%	4	20%
5	60%	5	10%
6	50%	6	10%
7	50%	7	10%
8	30%	8	10%
9	30%		
10	20%		

1.3. Neorganiniai paieškos rezultatai

Neorganiniais arba rėmėjų rezultatais vadinami paieškos rezultatai, už kuriuos tinklalapių savininkai moka pinigus paieškos sistemoms už realius paspaudimus. Dauguma paieškos sistemų, tarp jų ir Google, suteikia galimybę tinklalapių savininkams reklamuotis jų sistemoje. Tiesa, Google į reklamas žiūri labai atsargiai. Vartotojas reklamą pamatys tik tada, jeigu paieškos sistema, pagal vartotojo įvestą raktažodį nuspręs, kad konkrečiame skelbime gali būti informacijos, kurios jis ieško. Jeigu tokių rezultatų sistema neranda, reklamos vartotojui ji išvis nepateikia. Rezultatai pateikiami atsižvelgiant į Google kokybės indeksą, kuriuo įvertinamas

tinklalapis, todėl nebūtinai sumokėjus didesnę pinigų sumą, rezultatas atsiras aukštesnėje pozicijoje. Tiesa, statistikos duomenimis 70-80 procentų vartotojų ignoruoja rėmėjų nuorodas ir dėmesį sutelkia į organinius rezultatus, todėl prieš investuojant reikėtų gerai apgalvoti savo strategiją, nes galbūt didesnis dėmesio skyrimas optimizavimo paieškos sistemoms procesams atlikti atneš daugiau naudos, negu mokamos vietos neorganinių rezultatų sąrašė [Bar12].

2. OPTIMIZAVIMO PAIEŠKOS SISTEMOMS PROCESAS

Visą optimizavimą paieškos sistemoms galima suskirstyti į šešis pagrindinius žingsnius: **verslo analizę, planavimą, realizaciją, rezultatų stebėjimą, įšivertinimą bei priežiūrą.** Optimizavimo paieškos sistemoms procesas dažnai vyksta iteracijomis, nes norint pasiekti norimų rezultatų dažnai nepakanka atlikti minėtuosius žingsnius po vieną kartą [Eng15]. Autorius šiame darbe aptars visus šešis žingsnius, tačiau didžiausias dėmesys skiriamas realizacijos daliai, kurioje atliekami konkretūs techniniai žingsniai, susiję su internetinio puslapio struktūros bei kitomis modifikacijomis.

2.1. Verslo analizė

Verslo analizė yra pirma optimizavimo paieškos sistemoms proceso dalis. Ją taip pat galima išskaidyti į skirtingas dalis: rinkos tyrimą, konkurentų analizę, dabartinės situacijos įvertinimą bei raktažodžių tyrimą.

2.1.1. Rinkos tyrimas

Rinkos tyrimo metu reikia atsakyti į tokius klausimus, kaip: kuo ypatingas yra mūsų verslas, (pvz.: el. parduotuvė), kuriais išskirtiniais bruožais pasižymi mūsų parduotuvė, ką pranašesnio ji gali pasiūlyti už konkurentų parduotuves, kokių daugiausiai prekių tikimės parduoti, kurias prekes yra svarbiausia parduoti, kokiai auditorijai tikimasi parduoti savo prekes, iš kokio regiono tikimės sulaukti savo klientų, kokią dalį biudžeto galime skirti optimizacijai, kokių rezultatų tikimės, kaip nustatysime ar konkrečios iteracijos metu rezultatai yra teigiami ar neigiami bei daugeli kitų su verslo modeliu susijusių klausimų. Visi šie atsakymai reikalingi tam, kad prieš pradėdant atlikti konkrečius optimizavimo veiksmus jau tiksliai žinotume kokių rezultatų tikimės ir kurioms el. parduotuvės dalims turėsime skirti daugiausiai dėmesio, kad pasiektume norimus rezultatus.

2.1.2. Konkurentų analizė

Atlikus rinkos tyrimą, jau žinosime savo verslo stiprybes bei silpnybes, todėl laikas konkurentų analizei, kurios metu bandoma išsiaiškinti, kurie konkurentai, ir svarbiausia kodėl, yra Google ir kitų paieškos sistemų rezultatų viršūnėse. Taip pat stebima kurie tinklalapiai pateikia nuorodas į juos, koks turinys publikuojamas, į kokius raktažodžius taikomasi bei kiek lankytojų sulaukia minėtieji konkurentai. Trumpai tariant, kuo daugiau žinosime konkurentų stiprybių bei naudojamų strategijų, tuo greičiau suprasime, kuria linkme turėtume judėti su savo projektu, kad pasiektume norimų rezultatų.

2.1.3. Dabartinės situacijos vertinimas

Dabartinės situacijos vertinime bandoma išsiaiškinti kaip tinklalapis, kurį norime optimizuoti, šiuo metu yra vertinamas paieškos sistemų. Šioje dalyje bandoma atsakyti į tuos pačius klausimus kaip ir konkurentų analizėje, tik šį kartą, tiriamasis objektas yra tinklalapis, su kuriuo dirbsime. Šioje dalyje papildomai analizuojami paieškos sistemų reitingai skiriami mūsų tinklalapiui, tinklalapio jungiamoji struktūra, turinio dubliavimasis bei išorinės nuorodos, nukreipiančios į jį.

2.1.4. Raktažodžių tyrimas

Raktažodžių tyrimo metu identifikuojami ir susisteminami raktažodžiai, pagal kuriuos norime sulaukti lankytojų srauto. Egzistuojantiems tinklalapiams raktažodžių tyrimas turėtų identifikuoti, kurie jau naudojami raktažodžiai atneša naudą, t.y. pagal kuriuos raktažodžius sulaukiame lankytojų srauto ir į kuriuos jau esamus raktažodžius vartotojai nelinkę kreipti dėmesio. Raktažodžių tyrimas taikomas ne tik pačiam turiniui, kuris atvaizduojamas tinklalapyje, bet ir tokioms tinklalapio dalims kaip domeno vardas, nuorodų pavadinimai, vidinės nuorodos, direktorijų, kategorijų pavadinimai ir daugybei kitų elementų. Taip pat sprendžiama ar optimizacijos metu orientuosimės į populiariausius raktažodžius, tuos, pagal kuriuos, šiuo metu sulaukiama didžiausia dalis lankytojų srauto, bet kurie yra itin konkurencingi, o galbūt didesnę dėmesį skirsime ne tokiems abstraktiems, labiau sukonkretintiems raktažodžiais, tokiu atveju tikėdamiesi, kad didesnis lankytojų srautas ateis būtent dėl to turinio, kurį siūlo mūsų tinklalapis [Mal07].

Visų keturių verslo analizės dalių rezultatas turėtų būti dokumentas, kuriame atsispindėtų kiekvienos dalies tyrimo rezultatai, skirti naudoti vėlesniuose optimizavimo žingsniuose.

2.2. Planavimas

Planavimo dalyje atsakoma į kertinius klausimus, kurie išskyla turint verslo analizės rezultatus. Šioje dalyje turėsime apspręsti turinio organizavimo, išorinių nuorodų, socialinių tinklų, paieškos sistemų marketingo bei techninę strategijas.

2.2.1. Turinio organizavimo strategija

Turinio organizavimo strategijoje turime apspręsti visus aspektus, kurie susiję su turinio kūrimu, modifikavimu, platinimu bei archyvavimu. Šios strategijos metu turime nuspręsti kaip turinys bus pateikiamas vartotojams, ar naudosis blogus, naujienas, atsiliepimus, kitas medijos priemones. Taip pat suskirstomos tinklalapio dalys į tas, kurias norime, kad aplankytų paieškos

sistemų robotai, suindeksuotų ir taip leistų joms atsidurti paieškos rezultatuose bei į tas, kurių nenorėsime matyti paieškos rezultatuose.

2.2.2. Išorinių nuorodų strategija

Išorinių nuorodų strategijos planavimas yra itin svarbus galutiniam optimizavimo paieškos sistemoms rezultatui. Šioje dalyje reikia nuspręsti kokios išorinės nuorodos atves į mūsų tinklalapį. Ar skirsime dalį biudžeto tam, kad nuoroda į mūsų tinklalapį atsirastų tam tikruose, gerą reputaciją turinčiuose, tinklalapiuose. Taip pat, kaip išnaudosime žiniasklaidą, blogus, komentarus bei kitus šaltinius tam, kad per juos mūsų tinklalapį pasiektų kuo daugiau lankytojų.

2.2.3. Socialinių tinklų strategija

Žinome, kokią didžiulę auditoriją turi tokie socialiniai tinklai kaip Facebook, Twitter arba LinkedIn. Todėl socialinių tinklų strategijos metu reikėtų nuspręsti ar norėsime savo klientų auditoriją plėsti socialinių tinklų pagalba. Jeigu taip, pasirinkti socialinius tinklus, platformas, kurie labiausiai tiktų pagal savo pobūdį, tematiką ar kitus ypatumus.

2.2.4. Techninė strategija

Techninės strategijos metu sprendžiami tinklalapio realizacijos techniniai klausimai. Tam, kad sėkmingai būtų galima atlikti optimizavimą paieškos sistemoms, turime žinoti ir suderinti keletą techninių dalykų, tokių kaip: naršyklių palaikymas, talpinimas, duomenų srautas, bei kiti su tinklalapio palaikymu susiję klausimai. Daug kas priklauso nuo programinės įrangos, kuria paremtas tinklalapis. Jeigu naudojame turinio valdymo sistemą (TVS) reikia įsitikinti, kad ji yra lanksti ir be vargo leis atlikti tokias operacijas kaip tinklalapio adreso keitimą, klaidos pranešimų valdymą ar tinklalapio struktūros keitimą. Tinklalapio veikimo greitis itin svarbus, kai kalbame apie sudėtingus tinklalapius, su tūkstančiais puslapių ir dokumentų. Lėtai įkeliami tinklalapiai švaisto lankytojų laiką ir, tuo pačiu, neigiamai veikia koncentraciją. Jeigu tinklalapis kraunamas labai lėtai paieškos sistemų robotai gali nerinkti informacijos apie visus tinklalapio puslapius. Techninės strategijos metu taip pat turi būti suplanuota kokie įrankiai bus naudojami lankytojų srauto analizei.

Galutinis planavimo žingsnio rezultatas yra optimizavimo planas, kuriame yra išdėstyti galimi vidiniai ir išoriniai optimizacijos variantai. Šis planas taip pat nusako žingsnius ir procedūras, kurių bus privaloma laikytis pridedant ar modifikuojant turinį po optimizavimo atlikimo. Žinant, kad paieškos sistemų algoritmai nuolat keičiasi planas turėtų būti peržiūrimas bent kartą per keletą mėnesių.

2.3. Realizacija

Atlikus verslo analizę ir turint optimizavimo planą pradedama realizacijos dalis. Šioje dalyje panaudojami visi rezultatai ir informacija surinkta iš prieš tai buvusių analizių ir tyrimų. Realizacija skirstoma į dvi dalis: vidinį optimizavimą ir išorinį optimizavimą. Jos dydis bei sudėtingumas priklauso nuo konkretaus projekto dydžio. Realizacijos fazė nėra įvykdoma per vieną iteraciją. Per didelis pakeitimų skaičius gali apsunkinti rezultatų stebėjimą ir tokiu atveju būtų sunku susekti, kurie pakeitimai atnešė naudos, o kurie ne.

2.3.1. Vidinis optimizavimas

Vidinio optimizavimo metu visi atliekami pakeitimai yra tinklalapio viduje. Šiuos pakeitimus galima suskirstyti į du tipus: pakeitimai konkrečiam puslapiui, pavyzdžiui tinkamų raktažodžių parinkimas ir pakeitimai visam tinklalapiui, kuriuos taikome visai tinklalapio struktūrai, o ne konkrečiam tinklalapio puslapiui.

2.3.1.1. Domeno vardo pasirinkimas

Kuriant naują tinklalapį pirmasis žingsnis yra jo domeno vardo pasirinkimas. Raktažodžio buvimas domeno varde buvo vienas iš reitingavimo faktorių, todėl ilgą laiką buvo naudojama strategija, kai buvo stengiamasi, kad domeno vardas atitiktų konkretų raktažodį, į kurį taikosi tinklalapio turinys. Pavyzdžiui, pagal šią strategiją pardavinėjant iPhone dėklus mūsų parduotuvės domeno vardas galėtų būti „iphonedeklai.lt“. Tačiau 2012 metų rugsėji Google išleido EMD atnaujinimą, kurio esmė buvo neleisti prastos kokybės turinio tinklalapiams pasirodyti aukštose rezultatų pozicijose vien dėl to, kad raktažodis, pagal kurį buvo atlikta paieška buvo rastas domeno varde [Sch12]. Dėl šios priežasties nebėra būtina stengtis galvojant domeno vardą į jį įkomponuoti raktažodį. tinklalapis, kurio domeno vardas yra firmos pavadinimas ar prekinis ženklas turi visus šansus būti aukščiausiose rezultatų pozicijose.

2.3.1.2. Raktažodžiai „<title>“ žymėje

Vienas iš svarbiausių pakeitimų, kurie atliekami konkretiems puslapiams yra būtent teisingas „<title>“ žymės naudojimas. Šioje žymėje esantis turinys bus rodomas paieškos sistemų rezultatuose, kaip pavadinimas ir tuo pačiu nuoroda į tinklalapį. Reikia atsiminti, kad „<title>“ žymėje esanti informacija tiek vartotojams, tiek paieškos sistemoms nusako konkretaus puslapio temą. Idealiu atveju visų tinklalapio puslapių pavadinimai turėtų būti unikalūs. Juose esantys raktažodžiai turėtų apibūdinti būtent tame puslapyje esančią informaciją. Reikėtų vengti naudoti tik tinklalapio pavadinimo ar firmos vardo, nes tokia informacija būtų per daug abstrakti. Tinkamą

ir veiksmingą „<title>“ žymę galėtų sudaryti tinklalapio arba verslo pavadinimas, lokacija bei pagrindinės informacijos esančios tame puslapyje apibūdinimas. Šios informacijos ilgis neturėtų viršyti 70 simbolių, priešingu atveju Google paieškos sistema savo rezultatuose šį tekstą sutrumpins ir gale padės daugtaškį [Joy11].

2.3.1.3. Raktažodžiai „<meta>“ žymėje

Pirmoji „<meta>“ žymė su atributu „keywords“ yra raktažodžiams „<meta name=“keywords“ content=“raktazodziai“/>“. Šioje žymėje surašomi raktažodžiai, trumpai ir aiškiai apibūdinantys puslapyje atvaizduota turinį.

Tuo tarpu „description“ atributas „<meta name=“description“ content=“aprasymas“/>“ yra kiek svarbesnis, nes jis dažnai naudojamas kaip trumpas aprašymas paieškos rezultatuose po nuoroda. Tačiau ne visais atvejais nuorodos aprašymas bus „description“ atributo turinys, Google dažnai į aprašymą įdeda tekstą iš paties puslapio.

„Robots“ atributu „<meta name=“robots“ content=“index/follow“/>“ paieškos sistemoms nusakoma kaip elgtis su puslapiais. Žymės turinys susideda iš dviejų dalių: „index/noindex“ ir „follow/nofollow“. Pirma dalimi paieškos sistemoms leidžiama arba draudžiama, kad tinklalapis būtų rodomas paieškos rezultatuose. Antroje dalyje nusakoma kaip paieškos sistemos turėtų elgtis su nuorodomis puslapyje: leisti ar neleisti pasitikėti nuorodomis vedančiomis į kitus puslapius [Par13].

2.3.1.4. Nuotraukos ir paveikslėliai

Paveikslėliuose esantis tekstas yra neįskaitomas paieškos sistemų robotams, todėl nereikėtų pateikti jokio svarbaus teksto paveikslėlių formatu. Atributas „alt“ yra pavaizduojamas, kai dėl tam tikrų priežasčių naršyklė negali atvaizduoti paveikslėlio arba užvedus ant jo pelę. Šio atributo reikšmė turėtų būti paveikslėlio pavadinimas arba tiesiog tai, kas atspindi paveikslėlyje esančią informaciją (3 pav.). Šis atributas daro didžiausią įtaką paveikslėlio pasirodymui Google paveikslėliu paieškoje.

```

```

3 pav. Taisyklingai naudojamas paveikslėlio „alt“ atributas

Taip pat svarbus yra ir paveikslėlio pavadinimas. Dažnai pasitaikanti klaida yra, kai pamirštama pakeisti originalaus paveikslėlio pavadinimą į tokį, kuris jį apibūdintų. Reikėtų kiek įmanoma sumažinti paveikslėlių dydį, nes didžioji dalis vartotojų nelaukia, jei tinklalapis yra kraunamas ilgiau nei 3 sekundes [Hay13]. Optimaliausi naudojimui paveikslėlių tipai yra:

- JPEG formato paveikslėliai pasižymi geriausiu kokybės ir dydžio santykiu

- GIF formato paveikslėliai yra tinkamiausi tinklalapio dekoracijoms
- PNG yra JPEG formato atitikmuo, pasižymintis geresne kokybe ir didesne užimama vieta

2.3.1.5. Svetainės žemėlapis

Svetainės žemėlapis yra XML dokumentas, kuriame saugoma informacija apie visus svetainei priklausančius adresus, kuriuos norima indeksuoti. Apie kiekvieną adresą papildomai galima saugoti atnaujinimo dažnį, prioritetą arba paskutinio pasikeitimo datą (4 pav.). Svetainės žemėlapis skirtas tam, kad palengvintų robotų darbą apkeliaujant tinklalapį. Sukūrus svetainės žemėlapi, jo vieta robotams nurodoma įrašant adresą į „robots.txt” rinkmeną [Kil13].

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.google.com/schemas/sitemap/0.90">
  <url>
    <loc>http://www.ismaniparduotuve.lt</loc>
    <lastmod>2015-06-27T23:55:42+01:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>
</urlset>
```

4 pav. Svetainės žemėlapis iš vieno įrašo

2.3.2. Išorinis optimizavimas

Priešingai nei vidinio, išorinio optimizavimo nėra galimybės visiškai kontroliuoti. Išorinis optimizavimas yra susijęs su visais faktoriais svetainės išorėje, kurie atveda lankytojų srautą.

Pagrindinis išorinio optimizavimo veiksnys yra nuorodos iš teigiamai paieškos sistemų vertinamų tinklalapių į optimizuojama tinklalapį. Kuo didesnis kiekis tokių nuorodų, tuo tinklalapis vertinamas geriau. Išlaikyti aukštą reitingą padeda naujai atsirandančios nuorodos į optimizuojamą tinklalapį, kurios byloja, kad tinklalapis yra aktualus ir lankomas. Dažnai nuorodos patalpintos klientų ar panašios tematikos tinklalapiuose yra itin aukštos vertės, todėl tai yra vienas dažniausiai pasitaikančių išorinių nuorodų kūrimo būdų.

2.4. Rezultatų stebėjimas

Rezultatų stebėjimo fazėje stebima kaip keičiasi tokie rodikliai kaip paieškos sistemų robotų apsilankymai, išorinių nuorodų skaičius, paieškos sistemų reitingai, tinklalapio lankomumo statistika ir kiti. Šios fazės galutinis rezultatas yra suvestinė, kurioje yra visi minėtieji statistiniai rodikliai. Paprastai suvestinė pateikiama kaip mėnesinė ataskaita.

2.5. Įsivertinimas

Įsivertinimo fazėje naudojama ataskaita, prieš tai gauta rezultatų stebėjimo dalyje. Gauti rezultatai yra lyginami su užsibrėžtais tikslais. Ši dalis skirta tam, kad būtų galima stebėti kaip keičiasi optimizavimo plano progresas. Dažnai įsivertinimo dalis atskleidžia nemažai problemų. Smulkesnės problemos sprendžiamos priežiūros dalyje, tuo tarpu didesnėms problemoms spręsti gali tekti pašalinti kai kuriuos sprendimo būdus ir ieškoti naujų ar grįžti prie verslo analizės.

2.6. Priežiūra

Priežiūros fazėje atsiduriame, kai pagrindinis darbas realizacijos bei kitose dalyse yra atliktas. Pagrindinis dėmesys dabar bus skiriamas tinklalapio priežiūrai ir nedidelių problemų sprendimui, kurie dažnu atveju sutaps su kai kuriomis realizacijos dalimis.

3. SUDĖTINGI TINKLALAPIAI

3.1. Apibrėžimas

Sudėtingais puslapiais laikomi tinklalapiai, turintys daugiau nei šimtą puslapių, kurių turinys dinamiškai keičiasi ar yra generuojamas lankytojų. Sudėtingi tinklalapiai dėl savo struktūros ir architektūrinių sprendimų stipriai apsunkina optimizavimą paieškos sistemoms. Tokio tipo tinklalapio pavyzdys gali būti bet kuri elektroninė parduotuvė, turinti daugybę prekių, kurios pastoviai atnaujinamos ir papildomos.

Realizacijos dalyje buvo pateikti bendri aspektai, į kuriuos svarbu atsižvelgti nepriklausomai nuo svetainės sudėtingumo. Šiame skyriuje bus atsižvelgiama į konkrečias problemas, kurias dažniausiai iškyla optimizuojant sudėtingus tinklalapius.

3.2. Rinkmenų suspaudimas serveryje

Serverį optimizuoti galima įgalinant HTTP suspaudimą. Tokiu atveju serveris suspaudžia rinkmenas naudodamas GZIP algoritmą (5 pav.), o interneto naršyklė išpakuoja suspaustas rinkmenas, kai vartotojas atidaro puslapį. Šis pakeitimas gali sumažinti iš serverio siunčiamų rinkmenų dydį apie 50 procentų ar net daugiau. Google paieškos sistemos naudojamas GoogleBot robotas palaiko GZIP algoritmą, todėl HTTP suspaudimo įgalinimas ne tik leidžia vartotojui greičiau pasiekti tinklalapį, bet taip pat sumažina laiką, per kurį Google robotas apkeliauja tinklalapį.

```
HTTP/1.1 200 OK
Date: Tue, 01 May 2012 13:42:53 GMT
Server: Apache
X-Pingback: http://www.
X-Powered-By: W3 Total Cache/0.9.2.5b
Last-Modified: Tue, 01 May 2012 13:42:54 GMT
Vary: Cookie,User-Agent,Accept-Encoding
Expires: Tue, 01 May 2012 14:42:54 GMT
Pragma: public
Cache-Control: max-age=3600, public, must-revalidc
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
```

5 pav. Tinklalapio HTTP suspaudimas GZIP algoritmu

3.3. Dinaminės nuorodos

Raktažodžiai puslapio nuorodoje įtaka daro ne tik paieškos sistemų rezultatams, bet ir skaitomumui. Pavyzdžiui, situacijoje, kai vartotojas ieškodamas kameros Canon Powershot SD400 paieškos rezultatuose mato tris adresus:

1. Amazon.com <http://www.amazon.com/gp/product/B0007TJ5OG/102-8372974-4064145?v=glance&n=502394&m=ATVPDKIKX0DER&n=3031001&s=photo&v>

2. Canon.com

<http://consumer.usa.canon.com/ir/controller?act=ModelDetailAct&fcateoryid=145&modelid=11158>

3. DPreview.com - <http://www.dpreview.com/reviews/canonsd400>

Pirmame ir antrame pavyzdžiuose vartotojas negali suprasti į kokio turinio puslapį jis pateks paspaudęs nuorodą, nes šios nuorodos yra dinamiškai sugeneruotos, todėl jose atsispindi įvairūs identifikatoriai. Tuo tarpu, matydamas trečią pavyzdį (6 pav.), vartotojas nesunkiai supranta, kad nuorodos turinys greičiausiai yra Cannon SD400 apžvalga [Moz15]. Dėl šios priežasties nuorodą reikėtų generuoti būtent taip, kad jos pavadinime atsispindėtų turinys ir raktažodžiai. Šis aspektas yra ypač svarbus, kai kalbame apie tinklalapius, kuriuose turinys yra kuriamas vartotojų bei nuolat kintantis.

[Canon PowerShot SD400 Digital ELPH / Digital IXUS 50 ...](http://www.dpreview.com/reviews/canonsd400)

www.dpreview.com/reviews/canonsd400 ▾ Išversti šį puslapį

Announced in February 2005, the SD400 (which goes by the names of Digital IXUS 50 in Europe and IXY DIGITAL 55 in Japan) is the latest in a long line of ...

6 pav. Google paieškos rezultatas nr. 3

3.4. Nuorodų reikšmingumas

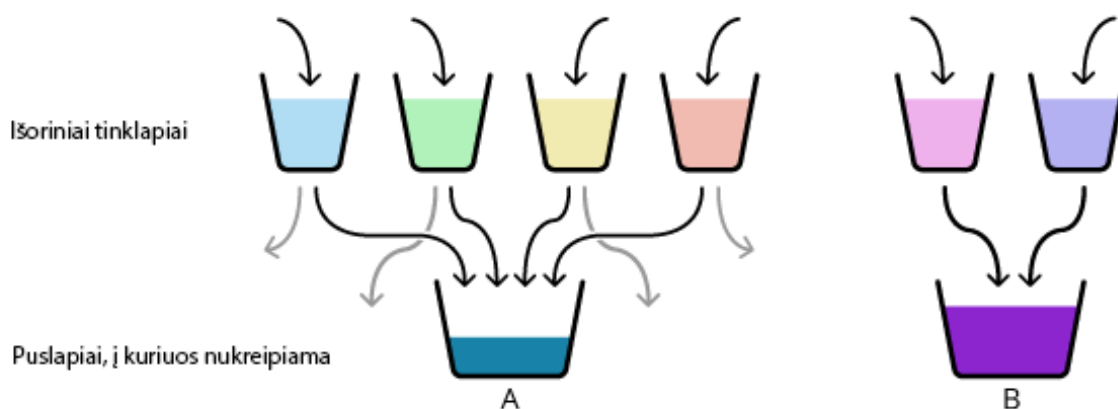
Nuorodos reikšmingumas yra optimizavimo terminas, apibūdinantis kiek reikšminga yra nuoroda iš vieno tinklalapio į kitą. Pavyzdžiui nuoroda iš žinomo ir gerai paieškos sistemų vertinamo tinklalapio bus vertinama kaip labai reikšminga, tuo tarpu, kai nuoroda iš itin retai lankomo tinklalapio turės itin mažą reikšmingumą. Tačiau šiuo aspektu tinklalapiai negali piktnaudžiauti, nes 2012 metais išleistas Google Penguin atnaujinimas baudžia tinklalapius, turinčius daug prastai vertinamų išorinių nuorodų, mažindamas jų reitingą.

Faktoriai darantys įtaką nuorodos reikšmingumui:

- Nuorodos vieta puslapyje (nuorodos esančios tinklalapio viršuje yra reikšmingesnės už nuorodas esančias apačioje)
- Nuorodos vieta kontekste (nuoroda pateikta puslapio konteksto tekste yra reikšmingesnė už nuorodą atskirai pateiktą šalutinėse puslapio dalyse)
- Tinklalapio, iš kurio yra nuoroda, reitingas.

Įsivaizduojant situaciją, kad yra du tinklalapiai A ir B, turinčius vienodus reitingus, tačiau tinklapis A turi vieną nuorodą iš neblogai vertinamo tinklalapio. Tokioje situacijoje tinklapis A paieškos rezultatuose atsiras aukštesnėje pozicijoje, nei tinklapis B. Tačiau galima analizuoti sudėtingesnę atvejį: 7 pav. pateikiamas panašus pavyzdys su tinklalapiais A ir B. Šiame pavyzdyje į tinklalapį A atveda keturios išorinės nuorodos, o į B dvi, tačiau paieškos rezultatuose

tinklalapis B atsiras aukštesnėje pozicijoje. Taip atsitinka todėl, kad puslapiuose, kuriuose yra nuoroda į tinklalapį A taip pat yra nuorodų į kitus tinklalapius, jos pažymėtos pilka spalva. Tuo tarpu puslapiai, nukreipiantys į tinklalapį B turi po vieną nuorodą, todėl jų reikšmingumas yra didesnis.



7 pav. Nuorodų reikšmingumo pasidalijimas [Dem13]

3.5. Vidinės nuorodos

Vidinės nuorodos yra ypatingai svarbios norint sukurti patogią navigaciją tinklalapyje tiek vartotojams, tiek paieškos sistemoms. Vienas iš populiariausių būdų kaip galima organizuoti vidines nuorodas yra kategorijos ir subkategorijos. Pavyzdžiui jeigu kalbama apie elektroninę parduotuvę su daugybe prekių tai kategorijos ir subkategorijos galėtų būti pateiktos tokiu būdu:

Avalynė – Krepšinio – Gamintojo pavadinimas

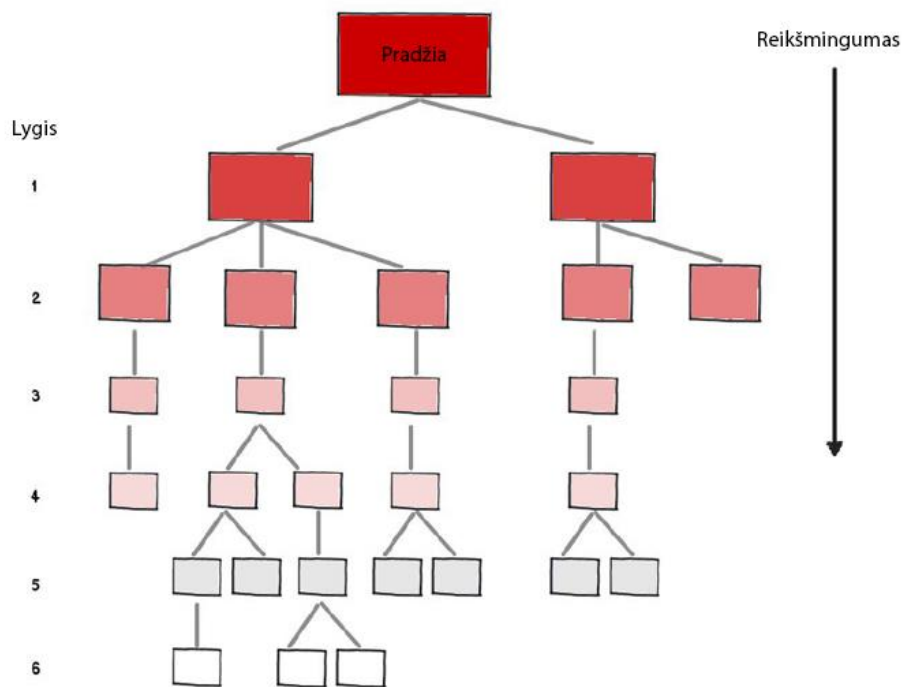
Apranga – Marškinėliai – Gamintojo pavadinimas

Tokių būdu elementariai organizuojant prekes pagal tipą nesunkiai galima sukurti navigaciją, kuria naudodamasis vartotojas atėjęs į parduotuvę per kelis paspaudimus atsidurs puslapyje, kuriame matys visas jį dominančias prekes. Tokia navigacija dažniausiai pateikiama išsiskleidžiančio meniu pagalba ir yra pakankamai intuityvi ir suprantama vartotojams [Jer10].

3.6. Tinklalapio struktūra

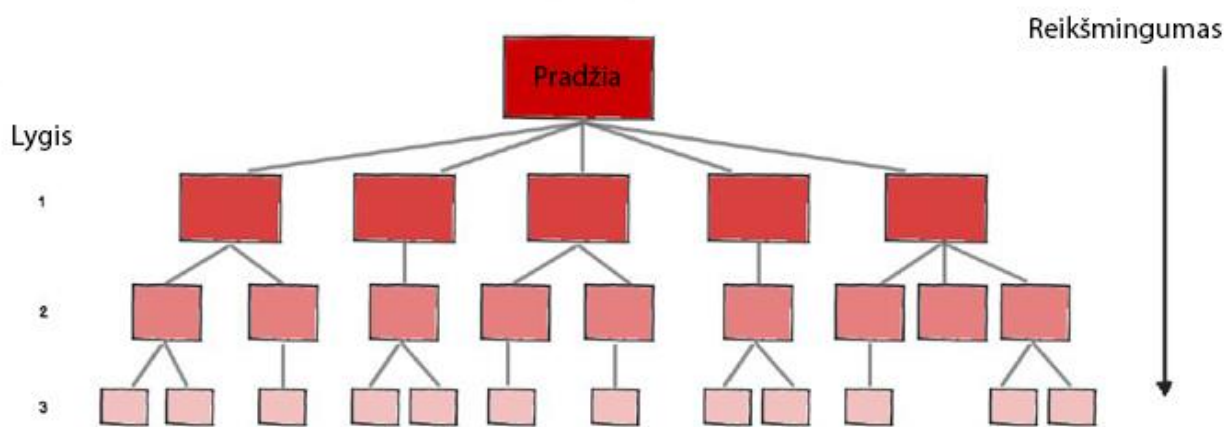
Tinklalapio struktūra yra svarbus faktorius ne tik optimizavimui paieškos sistemoms, bet ir tinklalapio patogumui, bei plečiamumui. Planuojant tinklalapio struktūrą, itin svarbu kiek įmanoma labiau daryti ją „orientuotą į plotį“, tokios struktūros esmė, kad vartotojui prireiktų kuo mažiau žingsnių pasiekti norimą puslapį, o protingai išdėsčius puslapius optimaliai išnaudoti nuorodų reikšmingumą iš aukštesnio puslapio.

8 pav. pateikta į gylį orientuota struktūra. Puslapio reikšmingumas krenta žemyn didėjant puslapio lygiui. Pavyzdžiui jeigu vartotojui reikia 5 paspaudimų, kad jis atsidurtų norimo produkto puslapyje, tai tokio puslapio reikšmingumas einant per lygius bus gerokai sumažėjęs.



8 pav. Tinklalapio struktūra, „orientuota į gylį“ [Jan12]

Tuo tarpu 9 pav. pateikta struktūra yra „orientuota į plotį“. Šios struktūros esmė yra nesukurti papildomų nereikalingų sluoksnių, kurie padidintų lygių skaičių nuo pradinio puslapio iki to, kurį nori pasiekti vartotojas. Dažniausiai tokia struktūra pasiekama skaidant tinklalapį į kategorijas bei subkategorijas (žr. 4.6 poskyrį), tuo pačiu tai yra puikus pagrindas optimizavimo paieškos sistemoms pakeitimams, kurių gali prireikti plečiantis tinklalapiui. „Orientuota į plotį“ struktūrą taip pat labai svarbi, nes paieškos sistemos nustoja indeksuoti puslapius, kai pasiekia tam tikrą lygį, todėl sukūrus struktūrą su labai daug lygių yra tikimybė, kad paskutinio lygio puslapių paieškos sistemos nepateiks savo rezultatuose.



9 pav. Tinklalapio struktūra, orientuota į plotį [Jan12]

3.7. Plagijuotas turinys

Dažnai pasitaikanti problema sudėtinguose tinklalapiuose, o ypač elektroninėse parduotuvėse yra plagijuotas turinys. Parduotuvėse siūlomų prekių aprašymai beveik visada būna pateikti iš gamintojų tinklalapių. Paieškos sistemos randa plagijuotą turinį, todėl tokius puslapius vertina neigimai.

Šią problemą galima spręsti dviem būdais. Jeigu įmanoma stengtis pateikti autentiškus aprašymus, tačiau toks sprendimo būdas dažniausiai kainuoja per daug laiko ir resursų. Paprastesnis sprendimas būtų tokius puslapius pažymėti specialiu parametru „ref=noindex“. Šiuo parametru pažymėtas turinys nėra indeksuojamas, todėl taip pažymėti galima visą pastraipą, kurioje yra neautentiškas turinys.

4. OPTIMIZAVIMO PAIEŠKOS SISTEMOMS PRINCIPŲ PRITAIKYMAS AngularJS KARKASUI

4.1. Asinchroninis JavaScript ir XML programavimas (AJAX)

AJAX (asinchroninis JavaScript ir XML programavimas) – tinklalapių kūrimo metodika, naudojama interaktyviems ir itin spartiems tinklalapiams kurti. Šią technologijų kombinaciją sudaro:

- HTML/XHTML bei CSS informacijos vaizdavimui.
- Dokumento objektinis modelis (DOM) bei JavaScript kalba dinamiškam turinio atvaizdavimui.
- XML, XSLT ir XMLHttpRequest objektas asinchroniniam duomenų apsikeitimui su serveriu.

Paprastai tinklalapių interaktyvumas sukuriamas siunčiant HTTP užklausas, tačiau tai nėra efektyvu, kadangi šiuo būdu dažnu atveju yra švaistomi resursai. Keliaujant per įprastą tinklalapį dažniausiai keičiasi pagrindinės informacijos sekcija, tačiau tokios sekcijos kaip tinklalapio navigacija ar kitos statinės dalys nesikeičia, tačiau vistiek yra atnaujinamos su kiekviena užklausa. Tinklalapiai naudojantys AJAX technologijas gali išsiųsti užklausas į serverį ir gauta atsakymą, JavaScript kalbos dėka, panaudoti atnaujinant tik reikiamą puslapio sekciją. Tokiu būdu taupomi tinklo bei serverio procesoriaus resursai, nes dalis logikos perkeliama į kliento naršyklę.

4.2. SPA samprata

SPA (angl. *Single Page Application*) – tinklalapis, kuris reikiamus resursus atsisiuočia pirmą kartą užsikraudamas naršyklėje. Priešingai nei standartiniai interneto tinklalapiai, sudarytas iš vieno pagrindinio puslapio, kurio atskiros dalys gali būti nepriklausomai atnaujinamos AJAX pagalba taip, kad visas tinklalapis nepersikrautų. Būtent ši savybė padaro SPA tipo tinklalapius itin interaktyvius ir patrauklius vartotojams [Mi12].

Privalumai:

- Greitesnis tinklalapio veikimas. Kadangi pagrindiniai resursai į naršyklę atsiunčiami pirmą kartą užkraunant puslapį paprastai duomenys iš serverio gaunami JSON formatu.
- Patraukli ir interaktyvi vartotojo sąsaja.
- Žymiai sumažinamas kodo dubliavimas.

Trūkumai:

- Kliento naršyklėje JavaScript kalbos palaikymas yra privalomas. Nors dauguma standartinių naršyklių palaiko JavaScript, tačiau be jo klientas negalėtų naudotis tinklalapiu.
- Saugumas. Kadangi JavaScript kalbos kodas yra vykdomas kliento naršyklėje, todėl gali būti pakeistas lokaliai. Visa prieigos kontrolė, duomenų validacija, bei kiti su saugumu susiję aspektai turi būti antrą kartą patikrinami serveryje.
- Optimizavimas paieškos sistemoms. Paieškos sistemos nuolat tobulina savo robotus, tačiau dauguma jų vis dar nesugeba teisingai skaityti JavaScript kalba parašytų tinklalapių, todėl optimizavimui atlikti būtini specifiniai metodai.

4.3. AngularJS karkasas

AngularJS – vienas populiariausių šių laikų atvirojo kodo JavaScript karkasų, sukurtas Google korporacijos. Iš kitų JavaScript kalbos karkasų išsiskiriantis tuo, kad gali būti naudojamas ne tik kaip papildomas komponentas vartotojo sąsajai patobulinti, tačiau kaip technologija, naudojama pilnai vartotojo pusei (angl. *Client side*) realizuoti.

Šiuo karkasu kuriami MVC (angl. *Model-View-Controller*) architektūros tinklalapiai. Tai reiškia, kad sistema yra išskaidoma į atskiras dalis, kad funkcionalumas ir poreikiai nepersidengtų. Pagrindinė MVC idėja yra atskirti tris pagrindines sistemos dalis: duomenis (modelį), verslo logiką (kontrolerį) bei duomenų atvaizdavimo lygmenį (atvaizdą). Atvaizdas panaudoja duomenis gautus iš modelio ir pateikia juos vartotojui [Ba13]. Kai vartotojas sąveikauja su sistema įvesdamas duomenis ar kitais būdais kontroleris atsakingas už tai, kad duomenys modelyje būtų atitinkamai pakeisti. Modeliui pasikeitus atvaizdas yra „perpiešiamas“ ir pateikiamas vartotojui.

4.4. Kuriamos sistemos aprašymas

Kuriama sistema yra dažnai pasitaikantis sudėtingo tinklalapio pavyzdys: SPA sistema, su daugiau nei šimtu puslapių, kurioje turinį kuria vartotojai bei turinio kiekis yra pastoviai didėjantis.

Kuriamos sistemos tema yra internetinė meno kūrinių galerija, kurioje užsiregistravę menininkai gali pasidalinti savo kūryba įkeldami savo darbus į sistemą. Įkelti meno kūriniai yra vieši ir matomi neregistruotiems vartotojams. Visas turinys sistemoje yra sukuriamas vartotojų ir saugomas duomenų bazėje.

4.5. Reikalavimai sistemai

Sistemos prototipo sukūrimui buvo pasirinkti esminiai reikalavimai, kurie bus įgyvendinti: Funkciniai reikalavimai (FR):

FR1. Naudotojas turi galimybę užsiregistruoti sistemoje.

FR2. Naudotojas turi galimybę į sistemą įkelti meno kūrinį.

FR3. Naudotojas turi galimybę matyti visų sistemoje užsiregistravusių autorių sąrašą.

FR4. Naudotojas turi galimybę matyti visus kūrinius, įkeltus į sistemą.

Nefunkciniai reikalavimai (NFR):

NFR1. Visi sistemoje vaizduojami duomenys turi būti matomi paieškos sistemų robotams.

NFR2. Sistemos puslapiai, kuriuose vaizduojami menininkų kūriniai turi būti pasiekiami per išorines nuorodas.

NFR3. Sistema automatiškai sugeneruoja tinklalapio žemėlapi (angl. *Sitemap*).

NFR4. Sistema palaiko Open Graph protokolą

4.6. Naudojamos technologijos

Serverio pusė:

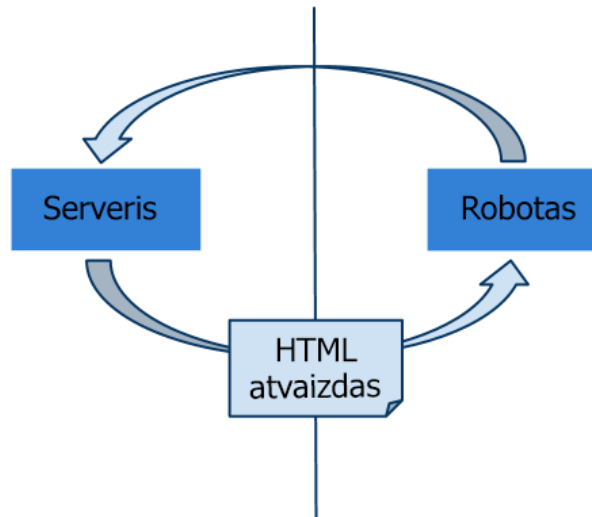
- ASP.NET Web API – sistemos dalis, pateikianti duomenis į kliento naršyklę.
- Entity Framework – objektinio - reliacinio susiejimo (ORM) technologija, atsakinga už prisijungimą prie duomenų bazės ir verslo objektų generavimą.
- Microsoft Unity – priklausomybės injekcijos karkasas
- Node.js – JavaScript paremta platforma, skirta dideliems ir spartiems tinklo projektams kurti.
- Express.js – vienas populiariausių MVC karkasų skirtų Node.js platformai.

Kliento pusė:

- AngularJS – JavaScript kalbos karkasas, kurio kuriamas SPA tinklalapis.
- Sky Forms – CSS karkasas, kuriuo paremtas tinklalapio dizainas.

4.7. Realių vartotojų atskirimas nuo paieškos sistemos robotų

Google paieškos sistema yra sukūrusi standartą Ajax Crawling, kuriuo naudojasi dauguma populiariausių paieškos sistemų [Gog15]. Šis standartas nusako kokius pakeitimus reikėtų atlikti tinklalapyje, naudojančiame Ajax technologijas, kad paieškos sistemos robotai matytų juose dinamiškai pateikiamą turinį. Pav 10. pateikiama Ajax Crawling veikimo schemos žingsniai.



10 pav. „Ajax Crawling“ schema [Aja14]

Ajax crawling schema veikia šešiais etapais:

1. Paieškos sistemos robotas indeksuodamas puslapį, kuris pažymėtas „<meta>“ atributu „<meta name="fragment" content="!">“ prie nuorodos prideda konstantą „?_escaped_fragment_="“, taip gaudamas vadinamą „negražią nuorodą“ (angl. *Ugly URL*).
2. Robotas į serverį siunčia pakeistos nuorodos užklausą.
3. Serveris gavęs „negražią nuorodą“ pašalina joje esančią konstantą.
4. Serveris sukuria HTML momentinį atvaizdą sutvarkytai nuorodai.
5. Serveris grąžina robotui parengtą HTML momentinį atvaizdą.
6. Robotas nuskaito gauto atvaizdo turinį, pradėdamas indeksuoti kitą puslapį kartoja veiksmus nuo pirmo žingsnio.

4.7.1. Serverio paruošimas

Šiam sprendimui įgyvendinti sistemos prototipe naudojamas atviro kodo projektas Prerender, kuris bus naudojamas kartu su Node.js serveriu. Serveris perims visas užklausas, kuriose aptiks konstantą „?_escaped_fragment_="“ ir persiųs šias užklausas Prerender komponentui, kuris grąžins pilnai užkrauto puslapio vaizdą.

Komponentas iš Github saugyklos atsisiunčiamas komanda:

„git clone https://github.com/prerender/prerender.git“

Visos reikalingos bibliotekos, nuo kurių priklausys naudojami komponentai išdėstytos „packages.json“ rinkmenoje, jos įdiegiamos komanda:

„npm install“.

Atsisiuntus komponentą ir įdiegus visas reikalingas bibliotekas, serveris paleidžiamas komanda „**node server.js**“, po kurios tinklo prievadu 3000, klausomasi užklausų (11 pav.).

```
C:\SEO\prerender\prerender>node server.js
2016-04-26T15:50:48.056Z starting worker thread #0
2016-04-26T15:50:48.089Z starting worker thread #1
2016-04-26T15:50:48.970Z starting phantom...
2016-04-26T15:50:49.007Z starting phantom...
2016-04-26T15:50:49.266Z Server running on port 3000
2016-04-26T15:50:49.266Z Server running on port 3000
2016-04-26T15:50:51.615Z started phantom
2016-04-26T15:50:51.621Z started phantom
```

11 pav. Serverio paleidimas

4.7.2. Kliento paruošimas

Paruošus serverį, kuris lauktų užklausų iš paieškos sistemų robotų SPA aplikacijoje reikalinga konfigūracija, kuria paieškos sistemų robotų užklausos būtų atskiriamos nuo tikrų lankytojų užklausų ir nukreipiamos į Prerender serverį.

MVC karkasas Express.js, skirtas Node.js platformai, su kuriuo yra suderinamas Prerender komponentas į SPA tinklalapį įdiegiamas komandine eilute:

„npm install -g express“.

12 pav. pateikiamas Express.js karkaso sukuriamas rinkmenos „www“ fragmentas, kuriame nustatomas tinklo prievadas (angl. *Port*), kuriuo bus pasiekamas SPA tinklalapis. Pagal nutylėjimą tinklo prievadas yra 3000, tačiau jau veikiantis Prerender serveris yra pasiekiamas tuo pačiu tinklo prievadu 3000, todėl SPA aplikacijos numatytasis tinklo prievadas yra pakeičiamas į 8080, kuris dažniausiai yra naudojamas tinklo servisams.

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('../app');
var debug = require('debug')('SPA:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */

var port = normalizePort(process.env.PORT || '8080');
app.set('port', port);
```

12 pav. SPA serverio konfigūracijos rinkmenos „www“ fragmentas

Kitas kliento konfigūravimo žingsnis yra nusakyti, Express.js karkasui SPA aplikacijos priklausomybę nuo Prerender serviso. 13 pav. pateikiamas „app.js“ rinkmenoje esančios konfigūracijos fragmentas, kuriuo nusakoma, kad veikiantis Prerender servisas yra pasiekiamas konkrečiu adresu bei tinklo prievadu, kuris buvo nustatytas ankstesniuose žingsniuose.

```
var routes = require('./routes/index');
var users = require('./routes/users');
var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(require('prerender-node').set(prerenderServiceUrl, 'http://localhost:3000'));
```

13 pav. SPA kliento rinkmenos „app.js“ fragmentas

Tinklalapio „index.html“ rinkmenoje pridamas „<meta>“ atributas „<meta name="fragment" content="!">“, kuriuo paieškos sistemos robotams pažymima, kad puslapyje yra naudojamos Ajax technologijos.

Šios konfigūracijos rezultatas – visos užklauskos, kuriose bus randama konstanta „?_escaped_fragment_“, bus peradresuojamos į Prerender servisą, kuris paieškos sistemų robotams gražins momentinį puslapio vaizdą.

4.8. Optimizavimas Open graph protokolui

Dalis didžiųjų socialinių tinklų, tokių kaip Facebook, Twitter ar Google+ palaiko Open graph protokolą. Šio protokolo pagalba galima internetiniam puslapiui suteikti „Socialinio objekto“ statusą [Fac14]. Puslapis turintis šį statusą yra atitinkamai rodomas socialinių tinklų paieškos sistemose.

Kūrinių peržiūros puslapiuose naudojamas fragmentas (14 pav.), kuriuo nusakomi šio puslapio meta duomenys (turinio pavadinimas, reprezentacinio paveikslėlio adresas, tinklalapio pavadinimas bei aprašymas). Socialinių tinklų, palaikančių Open graph protokolą, paieškos robotai indeksuodami šį puslapį atsižvelgs į pateiktus atributus ir atitinkamai atvaizduos turinį.

```
<meta property="og:title" content="{{vm.title}}" />
<meta property="og:image" content="{{vm.imageSrc}}" />
<meta property="og:site_name" content="Meno galerija" />
<meta property="og:description" content="{{vm.description}}" />
```

14 pav. „Open graph“ protokolo sintaksės pavyzdys

4.9. Svetainės žemėlapio generavimas

Svetainės žemėlapio protokolo formatas yra sudarytas iš XML žymų. Rinkmena turi būti išsaugoma UTF-8 koduota ir patenkinti šias sąlygas [Sit15]:

- Dokumentas atitinkamai prasideda ir baigiasi žymėmis „<urlset>“ ir „</urlset>“.
- Šakninėje žymoje apibrėžiama protokolo standarto vardų erdvė (angl. *Namespace*).
- Kiekvienai nuorodai sukuriama žymė „<url>“.
- Kiekviena „<url>“ žymė turi turėti vaikinę žymę „<loc>“.

NFR3. įgyvendinamas sukuriant svetainės žemėlapiui generuoti skirtą atskirą kontrolerio klasę „SitemapController.cs“. Taip pat sukuriamas žymės modelis „SitemapNode“, turintis ypatybes, kuriomis atitinkamai nusakomas nuorodos atnaujinimo dažnumas, paskutinio atnaujinimo data, prioritetas ir nuorodos adresas. Kontrolerio klasėje sukuriamas metodas „GetSitemapNodes()“, gražinantis „SitemapNode“ objektų sąrašą, kurio aprašomi visi sistemos puslapiai. 15 pav. pateiktame kodo fragmente vykdomas ciklas, kuriuo iš duomenų bazės gaunami visi sistemos vartotojų įkelti kūriniai.

```
foreach (var product in _unit.ProductsRepository.GetAll())
{
    nodes.Add(
        new SitemapNode()
        {
            Url = string.Concat(_products, "/", product.Id),
            Frequency = SitemapFrequency.Weekly,
            Priority = 0.8,
            LastModified = product.Created
        });
}
return nodes;
```

15 pav. „GetSitemapNodes()“ metodo fragmentas

Sugeneruotos žymės perduodamos kaip parametras į metodą „GetSitemapDocument“ (16 pav.), kuris iš šių suformuoja svetainės žemėlapi XML dokumentą.

```
foreach (SitemapNode sitemapNode in sitemapNodes)
{
    XElement urlElement = new XElement(
        xmlns + "url",
        new XElement(xmlns + "loc", Uri.EscapeUriString(sitemapNode.Url)),
        sitemapNode.LastModified == null ? null : new XElement(
            xmlns + "lastmod",
            sitemapNode.LastModified.Value.ToLocalTime().ToString("yyyy-MM-ddTHH:mm:sszz")),
        sitemapNode.Frequency == null ? null : new XElement(
            xmlns + "changefreq",
            sitemapNode.Frequency.Value.ToString().ToLowerInvariant()),
        sitemapNode.Priority == null ? null : new XElement(
            xmlns + "priority",
            sitemapNode.Priority.Value.ToString("F1", CultureInfo.InvariantCulture)));
    root.Add(urlElement);
}
XDocument document = new XDocument(root);
return document.ToString();
```

16 pav. „GetSitemapDocument()“ metodas

4.10. Navigacijos optimizavimas

AngularJS karkasas pagal nutylėjimą navigacijai tarp puslapių naudoja „#!” simbolius. Norint pašalinti šiuos simbolius prie kiekvienos nuorodos reikalingas pakeitimas tinklalapio „states.js“ rinkmenoje. (17 pav.). Šiuo pakeitimu nusakoma, kad „locationProvider“ servisas, kuris yra atsakingas už tinklalapio navigacijos sistemą naudotų HTML5 kalbos režimą. Minėtuojau

```
$locationProvider.html5Mode({
  enabled: true,
  requireBase: false
});
```

17 pav. AngularJS konfigūracijos fragmentas

režimu veikdamas „locationProvider“ servisas navigacijai naudoja HTML5 kalbos „History” programavimo sąsają, kuri leidžia keisti naršyklės dabartinę nuorodą neperkraudant puslapio, todėl karkasui nebereikia prie kiekvienos nuorodos pridėti simbolių „#!”.

4.11. Žymės „<title>“ optimizavimas

Žyme „<title>“ optimizuoti sukurtas atskiras servisas pavadinimu „pageTitle“. Visos karkaso kontrolių klasės gali pasiekti šį servisą ir funkcijos „setTitle“ pagalba (18 pav.) nustatyti dabartinio puslapio antraštę.

```
.service('pageTitle', function() {
  var title = 'Galerija';
  return {
    title: function() { return title; },
    setTitle: function(newTitle) { title = newTitle; }
  };
});
```

18 pav. Serviso atsakingo už „<title>“ žymės priskyrimą įgyvendinimas

Norint panaudoti serviso pateikiamą reikšmę į tinklalapio pagrindinį šabloną pridama HTML kalbos „<title>“ žymė (19 pav.). Jos reikšmė nustatoma iškviečiant „pageTitle“ serviso funkciją „title“, kuri grąžina reikšmę konkrečiam puslapiui.

```
<title ng-bind="pageTitle.title()"></title>
```

19 pav. „<title>“ žymės priskyrimas panaudojant reikšmę iš serviso

4.12. Žymės „<meta>“ optimizavimas

Žyme „<meta>“, sukurtas taip pat atskiras „metaInformation“ servisas (20 pav.). Karkaso kontrolių klasės gali priskirti dabartinio puslapio aprašymą funkcija „setMetaDescription“ bei raktažodžius, kurie gali būti priskiriami kelis kartus į funkciją „appendMetaKeywords“ perdavus dabartinio puslapio turinį apibūdinančius raktažodžius.

```

.service('MetaInformation', function () {
  var metaDescription = '';
  var metaKeywords = '';
  return {
    metaDescription: function () { return metaDescription; },
    metaKeywords: function () { return metaKeywords; },
    reset: function () {
      metaDescription = '';
      metaKeywords = '';
    },
    setMetaDescription: function (newMetaDescription) {
      metaDescription = newMetaDescription;
    },
    appendMetaKeywords: function (newKeywords) {
      for (var key in newKeywords) {
        if (metaKeywords === '') {
          metaKeywords += newKeywords[key].name;
        } else {
          metaKeywords += ', ' + newKeywords[key].name;
        }
      }
    }
  }
}

```

20 pav. „MetaInformation“ serviso fragmentas

Aprašymo ir raktazodžių reikšmės iš serviso panaudojamos į pagrindinį tinklalapio šabloną įdedant „<meta>“ žymes (21 pav.), kuriu reikšmės priskiriamos atitinkamai kviečiant serviso funkcijas „metaDescription“ ir „metaKeywords“.

```

<meta name="description" content="{{ MetaInformation.metaDescription() }}">
<meta name="keywords" content="{{ MetaInformation.metaKeywords() }}">

```

21 pav. „<meta>“ žymių priskyrimas

4.13. Sistemos prototipas

4.13.1. Naudotojo registracija ir prisijungimas

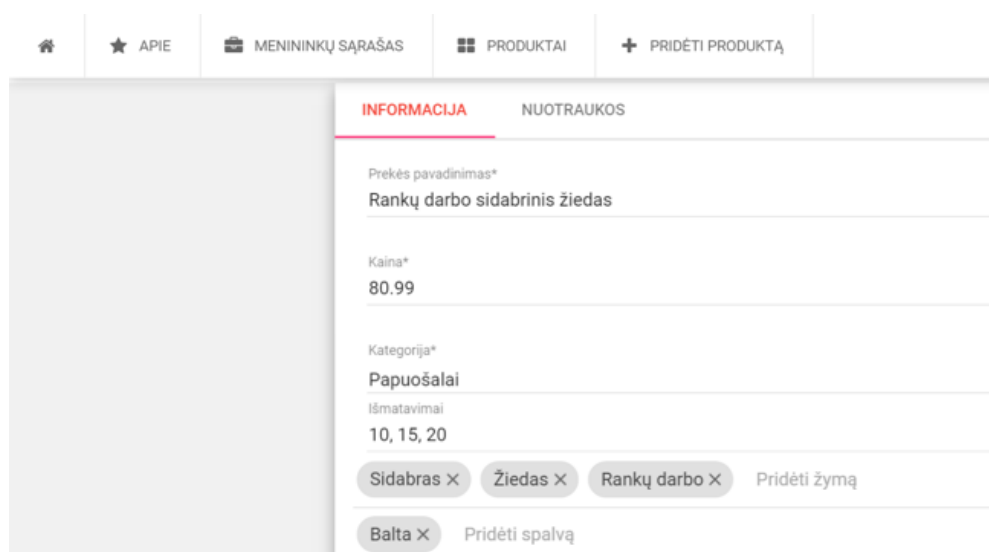
FR1. Buvo įgyvendintas sukuriant registracijos formą, kuri prieinama visiems vartotojams (22 pav.).

22 pav. Registracijos formos fragmentas

Registracijos formoje įvedami duomenys, kurie bus naudojami prisijungimui (prisijungimo vardas, slaptažodis) bei duomenys, kurie pateikiami paieškos sistemoms, pagal kuriuos būtų galima surasti autorių, bei jo darbus. Vartotojui sėkmingai užpildžius registracijos formą duomenys yra išsaugomi duomenų bazėje. Pagal FR2. vartotojas su savo duomenimis gali prisijungti sistemoje.

4.13.2. Naujo kūrinio įkėlimas į sistemą

FR2. įgyvendintas sukuriant kūrinio formą, kuri yra prieinama tik registruotiems sistemos vartotojams. Užsiregistravęs ir prisijungęs prie sistemos vartotojas norėdamas įkelti kūrinį į sistemą navigacijos juostoje pasirenka mygtuką „Pridėti produktą“, po kurio paspaudimo atidaroma kūrinio įkėlimo forma. (23 pav.).



23 pav. Naujo kūrinio įkėlimo formos fragmentas

4.13.3. Sistemoje užsiregistravusių autorių sąrašas

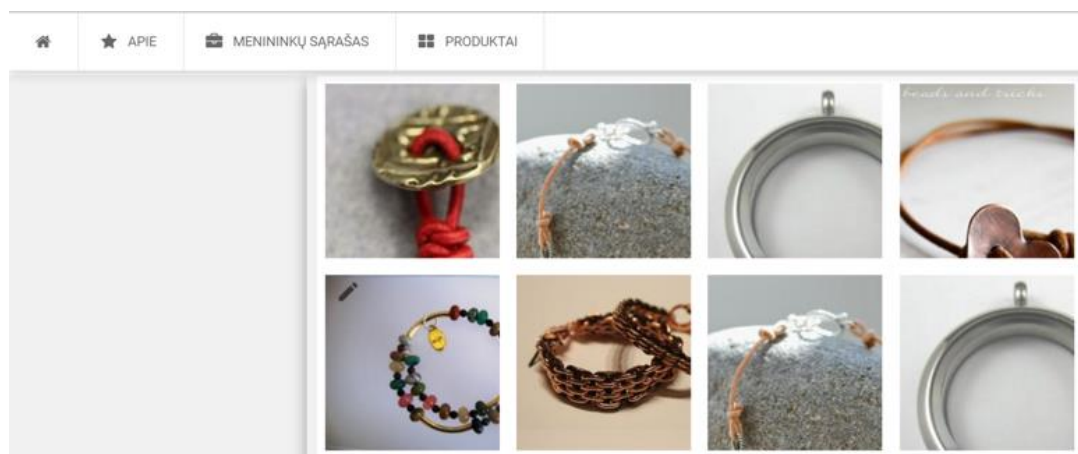
FR3. įgyvendintas sukuriant atskirą puslapį, kuriame rodomi visi užsiregistravę autoriai bei jų kontaktiniai duomenys (24 pav.).

Vardas	Pavardė	El.paštas	Tinklapis
Mindaugas	Matulionis	mindaugas@gmail.com	www.minecraft.lt
Jolita	Studentiene	jolita@gmail.com	www.deviantart.com/jolita
Tomas	Pavardavicius	toma@gmail.com	www.tomostudija.lt
Evaldas	Morkvenas	efomoro@gmail.com	www.arthouse.lt
Tauras	Jogela	tauras@gmail.com	www.deviantart.com/taurasj

24 pav. Sistemoje užsiregistravusių autorių sąrašo puslapio fragmentas

4.13.4. Į sistemą įkeltų kūrinių sąrašas

FR4. įgyvendintas sukuriant atskirą puslapį, kuriame parodomos visų autorių įkeltų kūrinių nuotraukos (25 pav.). Paspaudus ant nuotraukos atidaromas peržiūros langas su kūrinio aprašymu bei nuoroda į autoriaus puslapį.



25 pav. Į sistemą įkeltų kūrinių sąrašo puslapio fragmentas

REZULTATAI IR IŠVADOS

Rezultatai:

1. Darbe išanalizuoti ir pristatyti pagrindiniai Google paieškos sistemos veikimo principai bei komponentai: robotas, indeksatorius ir paieškos sąsaja.
2. Visas iteracinis optimizacijos procesas išskaidytas į šešias dalis. Šios dalys išanalizuotos atskirai po vieną ir išskaidytos į dar smulkesnius žingsnius. Paašškinta kiekvienos dalies paskirtis ir reikšmė.
3. Pateikti konkretūs optimizavimo paieškos sistemoms metodai, su realiais pavyzdžiais, kuriais optimizuojamos pagrindinės HTML kalbos žymės, tinklalapio žemėlapis bei turinys.
4. Išanalizuoti specifiniai optimizavimo paieškos sistemoms aspektai, kuriais paašškinama dinaminių nuorodų, vidinių nuorodų ir tinklalapio struktūros svarba optimizuojant sudėtingus tinklalapius.
5. Išanalizavus sudėtingų tinklalapių struktūras orientuotas į gylį ir į plotį, atsižvelgiant į jų atnešamą naudą optimizacijos paieškos sistemoms tikslais, galima teigti, kad tinkamiausia tinklalapio struktūra optimizavimo paieškos sistemoms yra orientuota į plotį.
6. Realizuotas SPA tipo tinklalapio prototipas, paremtas AngularJS karkasu, atitinkantis darbe apibrėžto sudėtingo tinklalapio sąvoką. Prototipui įgyvendintas pagrindinis kuriamos sistemos funkcionalumas: naudotojo registracija ir prisijungimas, naujo kūrinio įkėlimas į sistemą, autorių sąrašas bei į sistema įkeltų kūrinių sąrašas.
7. Realizuotam prototipui pritaikyti darbe išanalizuoti optimizavimo metodai - sukurta struktūra „orientuota į plotį“, turinys optimizuotas Open Graph protokolui, įgyvendintas automatinis tinklalapio žemėlapio generavimo komponentas, tinklalapio navigacija, palaikanti HTML5 kalbos režimą bei sukurti komponentai optimizuojantys pagrindines HTML žymes.

Apibendrinant šį darbą gautos išvados:

1. Optimizavimo procesas yra vykdomas iteracijomis, todėl norint pasiekti geriausių rezultatų neužteks optimizavimo žingsnius atlikti vieną kartą. Šiam tikslui pasiekti geriausia būtų palaipsniui taikyti optimizavimo metodus ir stebėti jų pasiekiamus rezultatus. Tokiu būdu bus galima pagrįstai įvertinti konkrečių metodų veiksmingumą, priešingu atveju būtų sunku atskirti, kuris metodas atnešęs daugiausia naudos.

2. Palankiausia tinklalapio struktūra optimizavimui paieškos sistemoms yra „orientuota į plotį“. Šios struktūros tikslas yra pasiekti, kad vartotojui prireiktų kuo mažiau žingsnių patekti į norimą puslapį, tokiu būdu neeikvojamas puslapio reikšmingumas patenkant į jį nuorodomis.
3. Optimizavimo metodų pritaikymas sudėtingiems tinklalapiams reikalauja daugiau techninių žinių, kadangi reikia tinkamai išanalizuoti sistemos veikimo principus ir atitinkamai atlikti pakeitimus tačiau paprastai reikalauja mažiau techninio darbo ir laiko sąnaudų, kadangi dalį optimizacijos metodų galima automatizuoti, pvz.: tinklalapio žemėlapių generavimą.
4. Nors ir paieškos sistemų robotai sparčiai tobulėja, visgi sudėtingi SPA tipo tinklalapiai, paremti JavaScript kalbos karkasais, vis dar reikalauja papildomų žingsnių, tokių kaip momentinio HTML atvaizdo generavimas, kurie pagerintų jų skaitomumą paieškos sistemų robotams.
5. SPA tipo tinklalapiai, kuriuose yra atvaizduojamas dinamiškai besikeičiantis turinys, yra tinkami realizuojant sudėtingus tinklalapius bei norint juos optimizuoti paieškos sistemoms, kadangi taip pat dinamiškai galima keisti ir organizuoti informaciją, kurią surinks paieškos sistemų robotai.
6. Sukurtas prototipas galėtų būti atskaitos taškas, kuriant sudėtingo tipo tinklalapius, nes jame realizuoti esminiai SEO principai, kuriais optimizuojamos pagrindinės HTML kalbos žymės, tinklalapio žemėlapis bei turinys.

ŠALTINIAI

- [Mi12] Michael S. Mikowski, Josh C. Powell. Single Page Web Applications. 2012. [žiūrėta 2016-05-24]. Prieiga per internetą: <<http://deals.manningpublications.com/spa.pdf>>
- [Ba13] Viknes Balasubramanee, Chathuri Wimalasena. Twitter Bootstrap and AngularJs. 2013. [žiūrėta 2016-05-18]. Prieiga per internetą: <<https://www.computer.org/csdl/proceedings/cluster/2013/9999/00/06702640.pdf>>
- [Eng15] Eric Enge, Stephan Spencer, Jessie Stricchiola. The Art of SEO: Mastering Search Engine Optimization 3rd Edition, 2015. [žiūrėta 2016-05-18].
- [Jer10] J. I. Jerkovic. SEO Warrior. O'Reilly Media, Inc., Sebastopol, 2010, 496 p. [žiūrėta 2016-05-18].
- [Bia11] Daniel Bianchini. 10 Stats to Justify SEO. 2011. [žiūrėta 2016-05-15]. Prieiga per internetą: <<http://www.searchenginejournal.com/10-stats-to-justify-seo/36762/>>
- [Int16] Total number of Websites. 2016. [žiūrėta 2016-05-10]. Prieiga per internetą: <<http://www.internetlivestats.com/total-number-of-websites/>>
- [Net16] Desktop Search Engine Market Share. 2016. [žiūrėta 2016-05-02]. Prieiga per internetą: <<http://www.netmarketshare.com>>
- [Bar12] Vinny La Barbera. 8 SEO Stats That Are Hard to Ignore 2012. [žiūrėta 2016-04-12]. Prieiga per internetą: <<http://www.imforza.com/blog/8-seo-stats-that-are-hard-to-ignore/>>
- [Sch12] Barry Schwartz. Google Issues „Weather Report” Of Crack Down On Low Quality Exact Match Domains. 2012. [žiūrėta 2016-04-12]. Prieiga per internetą: <<http://searchengineland.com/low-quality-exact-match-domains>>
- [Joy11] Alex Joy. How to Optimize Title Tags for SEO. 2011. [žiūrėta 2016-04-12]. Prieiga per internetą: <<http://webdesignzo.com/how-to-optimize-title-tags-for-seo/>>
- [Dem13] Boris Demaria. The Flow of Link Juice. 2013. [žiūrėta 2016-04-11]. Prieiga per internetą: <<http://blog.woorank.com/2013/05/the-flow-of-link-juice/>>
- [Jan12] Anne Jansa. Why You Lose If You Skip Website Planning. 2012. [žiūrėta 2016-04-01]. Prieiga per internetą: <<http://adpearance.com/blog/why-you-lose-if-you-skip-website-planning>>
- [Moz15] Rand Fishkin. What is a URL? 2015. [žiūrėta 2016-04-22]. Prieiga per internetą: <<http://moz.com/learn/seo/url>>
- [Hay13] Mark Hayes. 10 Must Know Image Optimization Tips. 2013. [žiūrėta 2016-04-01]. Prieiga per internetą: <<http://www.shopify.com/blog/7412852-10-must-know-image-optimization-tips>>
- [Par13] Aashna Parikh, Sanjay Deshmukh. Search Engine Optimization. 2013. [žiūrėta 2016-03-26]. Prieiga per internetą: <<http://www.ijert.org/view-pdf/6593/search-engine-optimization>>

- [Tak11] Frank Takes. Search Engine Optimization Overview. 2011. [žiūrėta 2016-03-26]. Prieiga per internetą: <<http://liacs.leidenuniv.nl/~takesfw/pdf/seo.pdf>>
- [Mal07] Ross A. Malaga. The Value of search Engine Optimization. 2011. [žiūrėta 2016-03-24].
- [Kil13] Chris Kilbourn. The Ultimate Guide to SEO for E-commerce Websites. 2013. [žiūrėta 2016-03-20]. Prieiga per internetą: <<http://kissmetrics.com/seo-for-ecommerce-websites/>>
- [Gog15] A proposal for making AJAX crawlable. 2009. [žiūrėta 2016-03-18]. Prieiga per internetą: <<https://webmasters.googleblog.com/2009/10/proposal-for-making-ajax-crawlable.html>>
- [Li12] Yunjia Li, Mike Wald and Gary Wills. Let Google Index Your Media Fragments. 2012. [žiūrėta 2016-03-17]. Prieiga per internetą: <<http://eprints.soton.ac.uk/336529/1/devel2012.pdf>>
- [Fac14] Facebook Developers. The Open Graph. How it works? 2014. [žiūrėta 2016-03-15]. Prieiga per internetą: <<https://developers.facebook.com/docs/sharing/opengraph>>
- [Aja14] AJAX Crawling. 2014. [žiūrėta 2016-03-02]. Prieiga per internetą: <<https://developers.google.com/webmasters/ajax-crawling/docs/getting-started>>
- [Sit15] Sitemaps XML protocol. 2015. [žiūrėta 2016-03-02]. Prieiga per internetą: <<http://www.sitemaps.org/protocol.html>>

SANTRUPOS

SEO – (angl. *Search Engine Optimization*). Optimizavimas paieškos sistemoms

SPA – (angl. *Single Page Application*). Vieno puslapio tinklalapis, kurio atskiros dalys gali būti nepriklausomai atnaujinamos AJAX pagalba.

TVS – Turinio valdymo sistema. Programinė įranga, skirta supaprastinti informacinių sistemų turinio valdymą.

XML – (angl. *Extensible Markup Language*). Bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba

AJAX – Asinchroninis JavaScript ir XML programavimas, tinklalapių kūrimo metodika, naudojama interaktyviems ir itin spartiems tinklalapiams kurti.

DOM – (angl. *Document Object Model*). Dokumento objektinis modelis, aprašantis dokumento struktūrą bei būdus, kuriais dokumentas gali būti pasiekiamas ar manipuluojamas.

XSLT – (angl. *Extensible Stylesheet Language (XSL) transformations*). Kalba, kuria aprašoma XML dokumento transformaciją į HTML dokumentą arba į kitokios struktūros XML dokumentą.

MVC – (angl. *Model-View-Controller*). Architektūros šablonas, kurio tikslas atskirti vaizdą ir sąveiką nuo sistemos duomenų.