

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Paulius Grabauskas

**Lanksčios optinio požymių atpažinimo sistemos
projektavimas ir tyrimas**

Magistro darbas

Darbo vadovas
prof. Eduardas Bareiša

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Paulius Grabauskas

**Lanksčios optinio požymių atpažinimo sistemos
projektavimas ir tyrimas**

Magistro darbas

Recenzentas

doc. Aleksas Riškus

2010-05-26

Darbo vadovas

prof. Eduardas Bareiša

2010-05-26

Atliko

IFM-4/2 gr. stud.

Paulius Grabauskas

2010-05-26

Kaunas, 2010

Turinys

1.	ĮVADAS	9
2.	ANALITINĖ DALIS	10
2.1.	Problema	10
2.2.	Dokumentų klasifikavimas	11
2.2.1.	Įvadas	11
2.2.2.	Klasifikavimo užduotis	11
2.2.3.	Užklausa prieš profilius.....	12
2.2.4.	Klasifikavimo algoritmų taksonomija.....	13
2.3.	Susijusios technologijos.....	13
2.3.1.	Teksto šablonų sulginimas (angl. string pattern matching).....	13
2.3.1.1.	Įvadas	13
2.3.1.2.	Vieno šablono algoritmai	14
2.3.1.3.	Algoritmai naudojantys begalinį šablonų skaičių.....	15
2.3.2.	Teksto grupavimas (angl. text clustering).....	17
2.3.3.	Informacijos išrinkimas (angl. information retrieval – IR)	19
2.4.	Optinio simbolių atpažinimo sistemos.....	23
2.5.	Projektavimo metodika	24
2.5.1.	Įvadas	24
2.5.2.	Taikymo sritis	24
2.5.2.1.	Projekto tikslas ir adresatas (kas naudosis)	24
2.5.2.2.	Problemos sprendimas pasaulyje.....	24
2.5.2.3.	Situacijos Lietuvoje įvertinimas	24
2.5.3.	Produkto apibūdinimas	25
2.5.3.1.	Programų sistemos funkcijos.....	25
2.5.3.2.	Sistemos kontekstas.....	26
2.5.3.3.	Vartotojo charakteristikos	26
2.5.3.4.	Vartotojo problemos.....	27
2.5.3.5.	Vartotojo tikslai	27
2.5.3.6.	Bendri apribojimai.....	27
2.6.	Projektavimo metodologijos ir technologijų analizė.....	27
2.6.1.	Įvadas	27
2.6.2.	Maisto produktų ženklavimas.....	28
2.7.	Egzistuojantys sprendimai	31
2.7.1.	Programa „Food Additives“.....	31
2.7.2.	Programa „Pocket Food Additives“	32
2.7.3.	Programa „SveikasVaikas mobile“	33
2.8.	Analizės išvados	35
3.	PROJEKTO REIKALAVIMŲ SPECIFIKAVIMO DALIS	36
3.1.	Projekto varovai (angl. project drivers)	36
3.1.1.	Sistemos paskirtis.....	36
3.1.1.1.	Projekto kūrimo pagrindas (pagrindimas)	36
3.1.1.2.	Sistemos tikslai (paskirtis).....	36
3.1.2.	Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys	36
3.1.3.	Vartotojai	37
3.2.	Projekto apribojimai	37
3.2.1.	Apribojimai sprendimui	37
3.2.2.	Diegimo aplinka.....	37
3.3.	Funkciniai reikalavimai	38
3.3.1.	Veiklos sudėtis (angl. the scope of the work)	38
3.3.1.1.	Veiklos kontekstas.....	38
3.3.1.2.	Veiklos padalinimas	38
3.3.2.	Sistemos sudėtis (angl. the scope of the product)	39
3.3.2.1.	Sistemos ribos.....	39
3.3.2.2.	Panaudojimo atvejų sąrašas	39
3.3.2.3.	Funkciniai reikalavimai	42
3.3.2.4.	Reikalavimai duomenims	44
3.4.	Nefunkciniai reikalavimai.....	44
3.4.1.	Reikalavimai sistemos išvaizdai (angl. look and feel)	44
3.4.2.	Reikalavimai panaudojamumui (angl. usability)	44
3.4.3.	Reikalavimai vykdymo charakteristikoms (angl. performance)	44

3.4.4.	Reikalavimai veikimo sąlygoms (angl. operational).....	44
3.4.5.	Reikalavimai sistemos priežiūrai (angl. maintainability and portability).....	44
3.5.	Projekto išeiga (angl. project issues)	45
3.5.1.	Atviri klausimai (problemos)	45
3.5.2.	Egzistuojantys sprendimai (angl. off-the-shelf solutions).....	45
4.	PROJEKTO ARCHITEKTŪROS SPECIFIKACIJOS DALIS	46
4.1.	Įvadas.....	46
4.1.1.	Skyriaus „Architektūros specifikacija“ paskirtis.....	46
4.1.2.	Apžvalga	46
4.2.	Architektūros pateikimas	46
4.3.	Architektūros tikslai ir apribojimai	46
4.4.	Panaudojimo atvejų vaizdas.....	47
4.5.	Sistemos statinis vaizdas.....	48
4.5.1.	Apžvalga	48
4.5.2.	Paketų detalizavimas.....	48
4.5.2.1.	Paketas TextRecognitionService	48
4.5.2.2.	Paketas EAnalyser	49
4.5.2.3.	Paketas SharedClasses.....	50
4.6.	Išdėstymo (angl. deployment) vaizdas.....	51
4.7.	Duomenų vaizdas.....	52
5.	TIRIAMOJI EKSPERIMENTINĖ DALIS	54
5.1.	Tyrimo metodika	54
5.2.	Kokybės vertinimas	54
5.3.	Tobulinimo galimybės	58
5.4.	Patobulinimų realizavimas.....	59
5.5.	Eksperimentas.....	62
6.	IŠVADOS	66
7.	LITERATŪROS SĄRAŠAS	67
8.	TERMINŲ IR SANTRUMPŲ ŽODYNAS	69
9.	PRIEDAI.....	70
9.1.	PRIEDAS 1: Sistemos dinaminis vaizdas.....	70
9.1.1.	Sąveikos diagramos	70
9.1.1.1.	Sąveikos diagrama „Pateikti nuotrauką iš fotoaparato“	70
9.1.1.2.	Sąveikos diagrama „Pateikti nuotrauką iš failo“	70
9.1.1.3.	Sąveikos diagrama „Apkirpti nuotrauką“	70
9.1.1.4.	Sąveikos diagrama „Atšaukti nuotraukos apkirpimą“	71
9.1.1.5.	Sąveikos diagrama „Analizuoti“	71
9.1.1.6.	Sąveikos diagrama „Atvaizduoti rezultatus“	71
9.1.2.	Sekų diagramos.....	72
9.1.2.1.	Sekų diagrama „Pateikti nuotrauką iš fotoaparato“	72
9.1.2.2.	Sekų diagrama „Pateikti nuotrauką iš failo“	72
9.1.2.3.	Sekų diagrama „Apkirpti nuotrauką“	73
9.1.2.4.	Sekų diagrama „Atšaukti nuotraukos apkirpimą“	73
9.1.2.5.	Sekų diagrama „Analizuoti“	74
9.1.2.6.	Sekų diagrama „Atvaizduoti rezultatus“	74
9.1.3.	Bendradarbiavimo diagrama „Analizuoti“	75
9.1.4.	Veiklos diagramos	75
9.1.4.1.	Veiklos diagrama „Pateikti nuotrauką iš fotoaparato“	75
9.1.4.2.	Veiklos diagrama „Pateikti nuotrauką iš failo“	76
9.1.4.3.	Veiklos diagrama „Apkirpti nuotrauką“	76
9.1.4.4.	Veiklos diagrama „Atšaukti nuotraukos apkirpimą“	77
9.1.4.5.	Veiklos diagrama „Analizuoti“	77
9.1.4.6.	Veiklos diagrama „Atvaizduoti rezultatus“	78
9.1.5.	Būsenų diagrama.....	78

Paveikslai

1 pav. Teksto grupavimas	17
2 pav. Panašumo nuokrypio nustatymas	21
3 pav. Išvadų tinklo modelis	22
4 pav. Kliento ir serverio programinės įrangos apsikeitimo duomenimis diagrama	26
5 pav. Ekologinio žemės ūkio logotipas	29
6 pav. Sveikas vaikas logotipas	30
7 pav. Trys kandidatai į ES ekologiško gaminio logotipą.....	30
8 pav. Programos „Food Additives“ langas	31
9 pav. Programos „Pocket Food Additives“ langas	32
10 pav. Visų priedų sąrašas.....	33
11 pav. Meniu	33
12 pav. Paieška	34
13 pav. Informacija apie priedą	34
14 pav. Veiklos kontekstas	38
15 pav. Panaudojimo atvejų diagrama	39
16 pav. Panaudojimo atvejų diagrama	47
17 pav. Sistemos paketų ryšiai	48
18 pav. Paketo TextRecognitionService klasių diagrama	49
19 pav. Paketo EAnalyser klasių diagrama.....	50
20 pav. Paketo SharedClasses klasių diagrama.....	51
21 pav. Sistemos išdėstymo vaizdas	51
22 pav. Maisto produkto „Orbit Spearmint“ etiketės nuotrauka	55
23 pav. Maisto produkto „Orbit WHITE“ etiketės nuotrauka	55
24 pav. Maisto produkto „Paukščių pienas“ etiketės nuotrauka	56
25 pav. Maisto produkto „DVARO fermentinis sūris“ etiketės nuotrauka	56
26 pav. Maisto produkto „Paukštienos sultinys ECO+“ etiketės nuotrauka	56
27 pav. Maisto produkto „Maggi Kostilija“ etiketės nuotrauka.....	56
28 pav. Maisto produkto „Knor Kanapuljong“ etiketės nuotrauka	57
29 pav. Maisto produkto „Maggi Kanapuljong“ etiketės nuotrauka.....	57
30 pav. Kriterijaus „Korektiškumas“ vertinimo suminės eilutės duomenų diagrama	58
31 pav. Paketo Fotrac klasių diagrama	60
32 pav. Įskiepių klasių diagrama	60
33 pav. Sistemos paketų ryšiai po naujo paketo integracijos.....	61
34 pav. Klasė Preservative.....	61
35 pav. Eksperimento rezultatų diagrama.....	65

Lentelės

1 lentelė. Statistiniai operatoriai ir loginės algebros statistinių aproksimacijų operatoriai	22
2 lentelė. Optinio teksto atpažinimo sistemų palyginimas	23
3 lentelė. Programų palyginimas.....	34
4 lentelė. Veiklos įvykių sąrašas.....	38
5 lentelė. Panaudojimo atvejo „1. Pateikti nuotrauką iš fotoaparato“ aprašymas	39
6 lentelė. Panaudojimo atvejo „2. Pateikti nuotrauką iš failo“ aprašymas	40
7 lentelė. Panaudojimo atvejo „3. Apkirpti nuotrauką“ aprašymas	40
8 lentelė. Panaudojimo atvejo „4. Atšaukti nuotraukos apkirpimą“ aprašymas.....	40
9 lentelė. Panaudojimo atvejo „5. Analizuoti“ aprašymas	41
10 lentelė. Panaudojimo atvejo „6. Atvaizduoti rezultatus“ aprašymas.....	41
11 lentelė. Funkciniai reikalavimai panaudojimo atvejui „1. Pateikti nuotrauką iš fotoaparato“	42
12 lentelė. Funkciniai reikalavimai panaudojimo atvejui „2. Pateikti nuotrauką iš failo“	42
13 lentelė. Funkciniai reikalavimai panaudojimo atvejui „3. Apkirpti nuotrauką“.....	42
14 lentelė. Funkciniai reikalavimai panaudojimo atvejui „4. Atšaukti nuotraukos apkirpimą“.....	43
15 lentelė. Funkciniai reikalavimai panaudojimo atvejui „5. Analizuoti“	43
16 lentelė. Funkciniai reikalavimai panaudojimo atvejui „6. Atvaizduoti rezultatus“.....	43
17 lentelė. Produkto „EAnalyser“ vertinimas	54
18 lentelė. Kriterijaus „Korektiškumas“ vertinimas	57
19 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant TesseactOcrPlugin įskiepi ir vieną požymio atpažinimo taisyklę	63
20 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant CuneiFormPlugin įskiepi ir vieną požymio atpažinimo taisyklę	63
21 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant TesseactOcrPlugin įskiepi ir dvi požymio atpažinimo taisykles.....	64
22 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant CuneiFormPlugin įskiepi ir dvi požymio atpažinimo taisykles.....	64

SUMMARY

Development and research of flexible optical attribute recognition system

These master theses consist of three main logical sections, which include the analysis of analytical problem technologies and design methodology; the specification of project requirements and project implementation; the research of system quality; the implementation of suggested improvements and finally the design and realization of experiment.

The first part includes the study of document classification problem, relative technologies and optical symbol recognition systems. In the analysis phase the project itself is introduced. It is the system that presents the information about additives found in food from a given food label photo. Initially, it is important to gather up the requirements and relative technologies for the project implementation.

The second part includes the specification of requirements for the project and project implementation. It was decided to use Microsoft .NET technology and client – server architecture.

The third part includes the tests of system quality. The most important criterion of the system quality is correctness, which shows how accurately the system finds the additives in food label. Using testing data system gives only 49% of correctness, which means it finds only half of all product label additives. For this reason there were improvements suggested and later on they were implemented. During the experiment with enhanced system version the percent of correctness grew up to the 88%, which completely justify the expectations.

SANTRAUKA

Lanksčios optinio požymių atpažinimo sistemos projektavimas ir tyrimas

Šis magistrinis darbas susideda iš trijų pagrindinių loginių dalių: analitinės problemos technologijų bei projektavimo metodologijos analizės; projekto reikalavimų specifikuojimo bei projekto realizavimo; sistemos kokybės tyrimo, siūlomų patobulinimų įgyvendinimo bei eksperimento.

Pirmoje dalyje nagrinėjama dokumentų klasifikavimo problema ir susijusios technologijos. Taip pat nagrinėjamos optinio simbolių atpažinimo sistemos. Pristatomas projektas: sistema iš maisto produkto sudėties etiketės nuotraukos pateikia informaciją apie maisto produkte naudojamus maisto priedus. Iškeliami reikalavimai projektui. Pasirenkamos tinkamiausios technologijos projekto įgyvendinimui.

Antroje dalyje specifikuojami reikalavimai projektui ir projektas įgyvendinamas. Naudojama Microsoft .NET technologija bei serverio – kliento architektūra.

Trečioje dalyje tiriama sistemos kokybė. Didžiausias dėmesys skiriamas kokybės kriterijui „Korektiškumas“, kuris parodo kaip tiksliai sistema aptinka maisto produkto sudėtyje esančius priedus. Sistema su duotais testiniais duomenimis aptinka tikrai 49% priedų. Todėl pasiūlomi ir įgyvendinami sistemos patobulinimai. Tuomet atliekamas eksperimentas, kurio metu ištiriama patobulintos sistemos kokybė. Įgyvendinus patobulinimus, sistema su duotais testiniais duomenimis aptinka 88% priedų, o tai yra labai geras rezultatas.

1. ĮVADAS

Stebint kaip sparčiai auga poreikis valgyti sveiką maistą (tai labai gerai parodo didelis internetinio maisto priedų žinyno „Sveikas Vaikas“ [17] lankytojų aktyvumas), buvo nuspręsta sukurti produktą, kuris padėtų renkantis sveikus maisto produktus.

Maisto produktų gamintojai privalo pateikti informaciją apie maisto produkto sudėtį. Ši informacija pateikiama maisto produkto sudėties etiketėje. Šiais laikais dažnas maisto produktas turi įvairių chemiškai ir kitais būdais išgaunamų kenksmingų žmogui priedų. Tokie priedai naudojami norint pagerinti maisto produkto skonį, išvaizdą, kvapą, pratęsti galiojimo terminą. Maisto priedų įvairovė skaičiuojama šimtais, todėl eilinis maisto produkto vartotojas tikrai nežino reikiamos informacijos apie juos. Situaciją pablogina tai, kad skirtingi gamintojai priedus maisto produkto etiketėse nurodo nevienodai. Vieni juos žymi kodais (pasaulinės sveikatos organizacijos „Codex Alimentarius“ standartas [25]), kiti tiesiog užrašo jų pavadinimus.

Pagrindinis kuriamos sistemos uždavinys – išgauti tekstą iš maisto produkto etiketės, jame surasti maisto produkte naudojamus priedus ir pateikti informaciją apie jų pavojingumą.

Informacijos apie maisto produkte naudojamus priedus išgavimas iš maisto produkto sudėties teksto yra aiškus klasifikavimo pavyzdys. Kadangi informacija pateikiama maisto produkto etiketėje, tai ją reikia perkelti į kompiuterį, o tai yra optinio simbolių atpažinimo (angl. optical character recognition) uždavinys.

Vienas svarbiausių informacinių technologijų uždavinys – pateikti teisingą informaciją patogiai ir laiku. Todėl kuriama sistema turi pasižymėti mobilumu, patogumu naudotis ir paprastumu.

2. ANALITINĖ DALIS

2.1. Problema

Šiandieniniame technologijų pasaulyje darosi vis sunkiau tvarkyti duomenis, todėl vis dažniau susiduriame su tokiais teiginiais kaip „skęstame duomenyse ir dūstame dėl informacijos trūkumo“, „informacijos gausa bukina“. Taigi, informacijos yra pakankamai, jos netgi yra per daug, tačiau trūksta priemonių, kad būtų galima atsirinkti tikslinę informaciją. Organizacijose, valstybinėse institucijose dar ir šiandien operuojama popierinėmis dokumentų formomis. Neretai reikalingai informacijai surasti tenka peržvelgi šūsnis popierinių dokumentų. Toks paieškos procesas trunka ilgai ir yra neefektyvus.

Šiai problemai išspęsti reikia universalaus metodo, kuris skaitmenizuoja popierinį dokumentą ir suranda reikiamą informaciją jame. Skaitmenizavimas – tai informacijos pavertimas į skaitmeninį formatą. Taigi, problema susideda iš dviejų dalių:

1. Popierinio dokumento skaitmenizavimo – optinio simbolių atpažinimo (angl. optical character recognition);
2. Reikiamos informacijos paieškos skaitmenizuotame dokumente – tam tikrų požymių (ieškomos informacijos) atpažinimo duotame dokumento tekste.

Pirmai problemos daliai spręsti yra naudojamos optinio ženklų atpažinimo sistemos. O antrą problemos dalį galima spręsti naudojant teksto klasifikavimo (angl. text classification) metodus. Teksto klasifikavimo tikslas – natūralios kalbos tekstui priskirti tam tikras apibrėžtas kategorijas. Pavyzdžiui, naujienos yra skirstomos į tam tikras kategorijas (temas): mokslas, sportas, verslas ir pan. Teksto klasifikavimas dar vadinamas teksto kategorizavimu (angl. text categorization), dokumentų klasifikavimu (angl. document classification), dokumentų kategorizavimu (angl. document categorization). Yra išskiriami du teksto klasifikavimo būdai: prižiūrimas klasifikavimas (angl. supervised classification) ir neprižiūrimas klasifikavimas (angl. unsupervised classification).

Šiame darbe bus aprašoma sistema, kuri pateiks informaciją apie mūsų kasdieniniame gyvenime naudojamus maisto produktus. Ne paslaptis, kad maisto produktų gamintojai norėdami pagerinti maisto produkto skonį, išvaizdą, kvapą bei prailginti galiojimo terminą į maisto produktą prideda priedų. Vieni maisto priedai yra nekenksmingi, tačiau kai kurie yra uždrausti tam tikrose šalyse, nes yra pavojingi žmogaus sveikatai. Todėl vartotojui yra svarbu pasirinkti tą maisto produktą, kuris yra mažiau pavojingas jo sveikatai. Tačiau maisto produkto etiketėje informacijos apie maisto priedus pateikiama nedaug – tai gali būti priedo kodas arba pavadinimas, kuris eiliniam maisto produkto vartotojui nesuteikia reikiamos informacijos. Sukurta sistema skaitmenizuos maisto produkto etiketėje esantį tekstą, tekste

suras informaciją apie maisto produkte naudojamus priedus ir pateiks reikiamą informaciją apie juos maisto produktų vartotojui.

2.2. Dokumentų klasifikavimas

2.2.1. Įvadas

[1] Duotas dokumentų rinkinys D ir klasių rinkinys C . Klasifikatorius klasei c_i yra funkcija $f'_i : D \rightarrow \{0, 1\}$. Ši funkcija yra aproksimuota nežinoma funkcija $f_i : D \rightarrow \{0, 1\}$, kuri išreiškia dokumentų ryšį su klase c_i . Tuomet dokumentų klasifikavimo užduotis gali būti aprašyta taip: kai duotas dokumentų klasių rinkinys C ir pavyzdiniai dokumentai kiekvienai klasei, reikia sukonstruoti klasifikatorių, kuris duotam dokumentui d randa klasę arba klases, kurioms dokumentas d labiausiai tinka.

Užrašymas aukščiau yra įvadas į šiuolaikinį automatinį dokumentų klasifikavimą.

2.2.2. Klasifikavimo užduotis

Dokumentų klasifikavimas gali būti naudojamas daugelyje sričių [1]:

1. Elektroninio pašto filtravimas

Sistemos, filtruojančios žmogaus gaunamus laiškus, kad atsikratyti apgaulingų laiškų arba, kad suskirstyti gaunamus laiškus į skirtingas klases.

2. Pašto nukreipimas (angl. mail routing)

Didelės įmonės dabartiniu metu automatizuoja savo dokumentų apdorojimą naudodamos darbo srautų (angl. work-flow) valdymo sistemas, kurios leidžia kompanijoje cirkuliuoti ne originaliems dokumentams, o elektroniniams dokumentų atvaizdams (kopijoms). Šios technologijos esmė yra dokumento nukreipimas į reikiamą darbo srautą. Šiame procese dalyvauja paviršutiniška žmogaus dokumento turinio interpretacija, kuri užima daug laiko ir gali būti klaidinga.

3. Naujienų stebėjimas

Žiniomis paremtose įmonėse, tokiose kaip vertybinių popierių biržos, didelė dalis darbuotojų skenuoja laikraščius ar iš kitų informacijos šaltinių renka informaciją, kuri yra susijusi su vidaus arba tarptautine rinkos ekonomija, arba su konkrečiomis įmonėmis esančiomis vertybinių popierių rinkoje. Rezultatai siunčiami darbuotojui, kuriam ši informacija reikalinga. Šis skubus darbas galėtų būti automatizuotas.

4. *Specialios kabelinės programos, kurios atsižvelgia į gyventojų interesus*

Spaudos agentūros stengiasi suteikti vis labiau individualizuotas paslaugas, kur kiekvienas klientas iš didelio naujienų srauto gauna tik tai naujienas, kurios yra susijusios su juo, pagal jo profilį (angl. profile).

5. *Turinio klasifikavimas*

Didelio informacijos kiekio bankai (tokie, kaip Europos patentų biuras) tradiciškai naudoja dokumentų išankstinę klasifikaciją (angl. pre-classification), kuri padeda demaskuoti netinkamus dokumentus. Dokumentai rankiniu būdu išskaidomi į didelę semantinę hierarchiją arba suindeksuojami naudojant žodynus. Šis procesas yra brangus ir gali būti klaidingas, be to, sudėtinga įgyvendinti pokyčius žodyne. Modernios žiniatinklio paieškos sistemos naudoja automatinę išankstinę puslapių klasifikaciją.

Visos aukščiau paminėtos sritys turi panašumų. Vietoj to, kad naudotų klasikinės užklausas, kurios išreiškia trumpalaikės informacijos poreikį, jos visos naudoja profilius, kuriais išreiškiamas ilgalaikis tam tikros informacijos poreikis.

2.2.3. Užklausos prieš profilius

[1] Vartotojo, sunkiai suprantančio savo informacijos poreikį, sudaryta kelių raktinių žodžių funkcija yra labai menka jo poreikio specifikacija. Taigi, pagal užklausą vis gaudamas atgalinį ryšį, vartotojas turi suformuluoti daug užklausų, kad išsiaiškintų savo informacijos poreikį.

Tam, kad būtų pasiektas aukštas tikslumas, daugeliu atvejų turi būti naudojamos kelių paprastesnių užklausų disjunkcijos ir konjunkcijos. Konstruoti tokias detalias užklausas rankomis yra labai komplikauta (ir brangu), jeigu tai daroma reguliariai ir pakartotinai. Taigi, yra poreikis rentabilios (angl. cost-effective) technologijos automatiniam profilių išgavimui. Tokia technologija naudojama automatiniame teksto dokumentų klasifikavime.

Raktinių žodžių išgavimas iš duoto dokumento ir jų normalizavimas veda į raktinių žodžių rinkinį – dokumento profilį. Akivaizdu, kad semantiškai susietų dokumentų klasė gali būti panašiai charakterizuojama pagal raktinį (bet pasvertą) žodžių rinkinį sukuriant klasės profilį.

Yra naudojamos klasifikavimo sistemos, kurios paremtos rankiniu (eksperto arba vartotojo) profilių konstravimu. Šios sistemos gali pasiekti aukštą tikslumą ir mažą klaidų tikimybę. Bet rankinis profilių konstravimas yra brangus procesas, kuris yra netinkamas kasdieniniam naudojimui arba sparčiai besivystantiems klasifikavimams. Čia būtų galima naudoti automatinio mokymosi technikas, kad išgauti tokius profilius automatiškai.

2.2.4. Klasifikavimo algoritmų taksonomija

[1] Didelis kiekis automatinio mokymosi algoritmų, kurie buvo naudojami dirbtinio intelekto bendruomenėje yra pritaikomi informacijos išrinkimo kontekste. Gerai žinomi prižiūrimo (angl. supervised) mokymosi klasifikavimo algoritmai, kurie gali išmokyti klasifikuoti profilius automatiškai iš pavyzdinių dokumentų. Mokymosi terminologijoje kiekvienas dokumentas turi požymius, pagal kuriuos jis gali būti suklasifikuotas. Klasifikavimo algoritmai gali suklasifikuoti viską, kas turi požymius.

Galima išskirti tris automatinio klasifikavimo algoritmų klases:

1. Taisyklėmis paremti (angl. rule-based) klasifikatoriai

Šie algoritmai mokosi išgaudami taisykles pagal iš anksto suklasifikuotus dokumentus. Geras pavyzdys yra Ripper algoritmas. Taip pat sprendimų taisyklės (angl. decision rules), kurios yra tam tikra sprendimo medžių (angl. decision trees) forma bei kiti algoritmai paremti teiginių (angl. propositional) logika.

2. Linijiniai (angl. linear) klasifikatoriai

Šie algoritmai, kiekvienai klasei paskaičiuoja klasės profilį – svorių vektorių. Kiekvienas vektoriaus svoris atitinka tam tikrą požymį. Svorio reikšmė yra paremta požymio dažnumu dokumente ir kitomis padarytomis išvadomis. Taigi, mokinant šiuos algoritmus sudaromi svorių vektoriai, o atpažįstant naujų dokumentų klases, dokumentuose ieškoma panašių vektorių. Linijiniai algoritmai: Rocchio, Naive Bayesian, Vector Support Machines, Winnow.

3. Pavyzdžiais paremti (angl. example-based) klasifikatoriai

Šie klasifikatoriai klasifikuoja naują dokumentą mokymo rinkinyje (angl. training set) ieškodami k dokumentų arčiausiai klasifikuojamo ir remdamiesi tam tikra daugumos balsavimo forma artimiausių kaimynų klasėse. Pavyzdžiais paremti algoritmai: K-Nearest Neighbor.

2.3. Susijusios technologijos

2.3.1. Teksto šablonų sulyginimas (angl. string pattern matching)

2.3.1.1. Įvadas

Informatikoje šablonų sulyginimas yra automatinės duomenų analizės technika (paprastai atliekama naudojant kompiuterį). Šios technikos pagalba nežinomo objekto grupė būdingų savybių yra palyginama su žinomų objektų savybių grupėmis, tam kad suklasifikuoti nežinomą objektą.

Tokie šablonai įprastai siejasi su sekomis arba medžių struktūromis. Šablonų sulyginimas naudojamas tam, kad patikrinti ar lyginami dalykai yra reikiamos struktūros, taip pat surasti reikiamą struktūrą arba pakeisti sulygintą struktūrą kažkuo kitu.

Teksto šablonų sulyginimo algoritmai, dar vadinami teksto paieškos algoritmais. Tai yra svarbi klasė teksto algoritmų, kurie tekste ieško vieno arba kelių šablonų vietos.

Tarkime, Σ yra alfabetas (baigtinis rinkinys). Paprastai ir šablonas, ir tekstas, kuriame ieškoma, yra susiję alfabeto Σ elementais. Σ gali būti įprastinė žmogaus abėcėlė (pavyzdžiui, raidės nuo A iki Ž lietuvių kalboje). Kiti taikymai gali naudoti dvejetainį alfabetą ($\Sigma = \{0,1\}$) ir panašiai.

Įvairūs algoritmai gali būti klasifikuojami pagal šablonų skaičių, kurį jie naudoja.

2.3.1.2. Vieno šablono algoritmai

Naïve teksto paieškos algoritmas

Naïve algoritmas dar vadinamas brutaliomis jėgomis (angl. brute-force) metodu. Tai paprasčiausias ir mažiausiai efektyvus būdas patikrinti, ar tekste yra ieškomas teksto fragmentas. Naudojant šį metodą reikia patikrinti kiekvieną vietą simbolis po simbolio, kur galėtų būti teksto fragmentas. Taigi, pirmiausiai tikrinama, ar pirmas teksto simbolis sutampa su teksto fragmento pirmu simboliu. Jeigu ne, tuomet tikrinama, ar antrasis teksto simbolis sutampa su pirmuoju teksto fragmento simboliu. Jeigu ne, tuomet tikrinama, ar trečiasis teksto simbolis sutampa su pirmuoju teksto fragmento simboliu ir t.t. Pirmam teksto fragmento simboliui sutapus, tikrinamas antrasis simbolis ir t.t. Paprastu atveju reikia ieškoti tikrai vieno ar dviejų simbolių kiekvienai neteisingai pozicijai, norint patikrinti, ar tikrinama pozicija yra bloga. Taigi, vidutiniu atveju, tai užtrunka $O(n+m)$ žingsnių, kur n yra teksto ilgis, o m yra teksto fragmento ilgis. Bet blogiausiu atveju (pavyzdžiui, ieškant teksto fragmento „aaaab“ tekste „aaaaaaaaab“) tai užtrunka $O(m \cdot n)$ žingsnių.

Knuth-Morris-Pratt teksto paieškos algoritmas

Knuth-Morris-Pratt teksto paieškos algoritmą 1977 metais išrado Donald Knuth, Vaughan Pratt ir James H. Morris. [3] Pagrindinis principas, kuriant šį algoritmą, buvo toks: fragmentas yra žinomas iš anksto ir ši informacija turi būti panaudota. Pavyzdžiui, jei fragmento pirmasis simbolis toliau nesikartoja, o fragmento ir sekos lyginimo metu sutapo $(j-1)$ simbolių, ir j -asis nesutapo, tai faktiškai yra žinoma daug daugiau – sekoje nuo tos vietos, kur buvo pradėta tikrinti, galima praleisti $(j-1)$ simbolių ir pereiti prie j -ojo simbolio lyginimo, nes nė vienas iš šių praleistų simbolių tikrai nesutaps su pirmuoju fragmento simboliu. Šios idėjos įgyvendinimas ir sudaro nagrinėjamo algoritmo turinį. Be to, remiantis įvairiais papildomais pastebėjimais galima pasiekti, kad rodyklės ir reikšmės niekad nemažėtų.

[3] Nors lyginimų skaičiumi šis algoritmas daug efektyvesnis už brutalią jėgą algoritmą, tačiau jo reikšmingumas nėra labai didelis. Taikymuose ne taip jau dažnai pasitaiko pasikartojančio teksto pasikartojančių fragmentų paieška. Šis metodas turi kitą privalumą – sekoje simboliai peržiūrimi nuosekliai ir niekada negrįžtama atgal. Todėl tokį algoritmą yra patogiau realizuoti situacijose, kai simboliai įvedami vienas po kito, iš šių simbolių yra išrenkami reikalingi fragmentai, tačiau nereikia naudoti papildomos atminties anksčiau buvusiems sekos simboliams įsiminti.

Knuth-Morris-Pratt algoritmas naudoja ne daugiau kaip $O(m+n)$ simbolių lyginimų, kur n yra teksto ilgis, o m yra teksto fragmento ilgis.

Boyer-Moore teksto paieškos algoritmas

Boyer-Moore algoritmas yra ypač efektyvus teksto paieškos algoritmas. Jis naudojamas kaip etalonas praktinėje teksto paieškos literatūroje. Šį algoritmą 1977 metais sukūrė Bob Boyer ir J. Strother Moore [2]. Naudojant šį algoritmą nereikia apdoroti kiekvieno simbolio ieškomam tekste, nes kai kuriuos simbolius algoritmas tiesiog praleidžia. Algoritmas greitėja ilgėjant tekstui, kuriame ieškoma teksto fragmento. Algoritmas yra efektyvus todėl, kad po kiekvieno nesėkmingo bandymo, kai ieškoma atitikimo tarp teksto ir teksto fragmento, gaunama informacija, kuri naudojama tų teksto pozicijų išsiaiškinimui, kuriuose teksto fragmento negali būti.

[4] Boyer-Moore algoritmo idėja gali būti pagrįsta tokiu pavyzdžiu: jei teksto fragmento 10100111 8-a, 7-a ir 6-a pozicijos sutampa su teksto simboliais, bet 5-a pozicija nesutampa, iš karto teksto fragmento analizė perkeliama per 7 pozicijas ir tikrinama nuo 15-os pozicijos. Šitokį sprendimą galima pagrįsti tuo, kad teksto fragmento 6, 7, 8 pozicijos (t.y. fragmento dalis 111) niekur daugiau teksto fragmente nesikartoja.

Boyer-Moore paieškos algoritmas blogiausiu atveju naudoja $O(m+n)$ simbolių lyginimų. Jam reikia apie $O(n/m)$ lyginimų, jei alfabetas nėra labai mažas, o teksto fragmentas nėra labai ilgas. Pastaroji savybė pagrįsta pastebėjimu, kad jei simbolių alfabete yra daug, o fragmentas ne toks jau ilgas, daugelio simbolių arba nebus fragmente iš viso, arba jie pasikartos ne daugiau kaip vieną kartą.

2.3.1.3. Algoritmai naudojantys begalinį šablonų skaičių

Paprastai naudojant begalinio šablonų skaičiaus algoritmus, visi šablonai negali būti išvardyti. Todėl dažniausiai tokie šablonai pateikiami kaip reguliariosios išraiškos.

Reguliariosios išraiškos

Reguliariosios išraiškos yra kilusios iš automatų teorijos ir iš formalių kalbų teorijos, kurios abi yra teorinės informatikos šakos. Šios šakos nagrinėja skaičiavimo (automatų)

modelius ir būdus kaip aprašyti ir klasifikuoti formalias kalbas. 1950 metais matematikas Stephen Cole Kleene aprašė šiuos modelius naudodamas savo matematinę notaciją, kurią pavadino reguliariaisiais rinkiniais [5].

Reguliariosios išraiškos, tai glausta ir lanksti priemonė teksto šablonų sulyginimui, tokių kaip tam tikri simboliai, žodžiai arba teksto fragmentai. Reguliariosios išraiškos yra rašomos formalia kalba, kuri gali būti interpretuojama naudojant reguliarių išraiškų procesorių. Reguliarių išraiškų procesorius – tai programa, kuri veikia kaip sintaksinių analizatorių (angl. parser) generatorius arba tiria tekstą ir identifikuoja teksto vietas, kurios atitinka pateiktą specifikaciją.

Sekantis pavyzdys parodo kelias specifikacijas, kurios gali būti išreikštos kaip reguliariosios išraiškos:

- Simbolių seka „arba“ bet kokiame kontekste. Pavyzdžiui, „**arba**“, „**arbata**“ arba „**karbamidas**“;
- Žodis „arba“, kai jis yra kaip atskiras žodis;
- Žodis „arba“, kai jis eina po žodžių „didelis“ arba „vidutinis“;
- Vienas arba daugiau skaitmenų, kurie neprivalomai gali turėti taško simbolį ir du ar daugiau skaitmenų (pvz. „52“ arba „687.45“).

Reguliariosios išraiškos gali būti daug sudėtingesnės nei šiuose pavyzdžiuose.

Reguliariųjų išraiškų pagrindai

Reguliariosios išraiškos dažnai vadinamos šablonais, kurie aprašo teksto fragmentų rinkinius. Šablonai paprastai naudojami glaustai aprašyti teksto fragmentų rinkinį, neišvardijant visų fragmentų. Pavyzdžiui, fragmentų rinkinys „Tomas“, „Domas“ ir „Jonas“ gali būti užrašyti šablonu „(T|D|J)o(m|n)as“. Taigi, šis šablonas atitinka kiekvieną iš trijų fragmentų.

Loginis operatorius „ir“

Vertikalus brūkšnyas atskiria alternatyvas. Pavyzdžiui, „Tomas|Domas“ atitinka „Tomas“ arba „Domas“.

Grupavimas

Kaip ir matematikoje, skliaustai nurodo veiksmų eilę. Pirmiausiai atliekami veiksmai skliaustuose. Pavyzdžiui, „Tomas|Domas“ ir „(T|D)omas“ atitinka tą patį rinkinį „Tomas“ arba „Domas“.

Kvantavimas (angl. quantification)

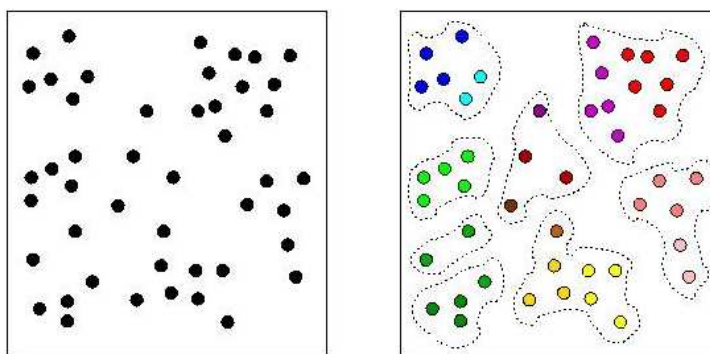
Kvantorius po elemento (tokio kaip simbolis arba simbolių grupė) nusako kaip dažnai tas elementas gali būti atkartojamas. Dažniausiai naudojami kvantoriai yra klaustukas, žvaigždutė ir pliuso ženklas:

- Klaustukas „?“ parodo, kad gali būti nulis arba vienas prieš klaustuką einantis elementas. Pavyzdžiui, „Klo?nas“ atitinka „Klonas“ ir „Klounas“.
- Žvaigždutė „*“ parodo nulis arba daugiau prieš žvaigždutę esančių elementų. Pavyzdžiui, „ab*c“ atitinka „ac“, „abc“, „abbc“, „abbbc“ ir t.t.
- Pliuso ženklas „+“ parodo vienas arba daugiau prieš pliusą esančių elementų. Pavyzdžiui, „ab+c“ atitinka „abc“, „abbc“, „abbbc“ ir t.t., bet ne „ac“.

Šios konstrukcijos gali būti kombinuojamos, kad suformuotų sudėtingas išraiškas. Formuojama labai panašiai kaip ir matematikoje konstruojant aritmetines išraiškas iš skaičių ir operatorių +, −, × ir ÷. Pavyzdžiui, abu šablonai „(T|D)omas“ ir „(D|T)om?s“ yra teisingi rinkiniui „Tomas“ ir „Domas“.

2.3.2. Teksto grupavimas (angl. text clustering)

Pats paprasčiausias šios technologijos apibrėžimas galėtų būti: tai objektų, kurių nariai tam tikra prasme panašūs, apjungimo į grupes procesas [6]. Objektai suskirstomi į grupes taip, kad skirtumai klasterių viduje būtų kuo mažesni, o tarp klasterių — kuo didesni. Teksto grupavimo – apjungimo į klasterius pavyzdys, pateiktas 1 paveikslėlyje.



1 pav. Teksto grupavimas

Klasterizavimo algoritmai skirstomi į dvi dideles klases [7]: hierarchiniai ir nehierarchiniai. Hierarchiniai metodai nusako klasterių tarpusavio hierarchiją, t.y. visi objektai laikomi vienu dideliu klasteriu, kurį sudaro mažesni klasteriai, šiuos savo ruožtu dar mažesni ir t.t. Tik nustatius bendrą visų klasterių tarpusavio priklausomybių struktūrą sprendžiama, koks klasterių skaičius yra optimalus. Hierarchiniai metodai dar skaidomi į dvi grupes: skaidymo (vienintelis klasteris nuosekliai skaidomas į dalis) ir jungimo (kiekvieną

žingsniu naujas objektas yra jungiamas prie esamo klasterio). Šios klasės metodus geriausia taikyti tuomet, kai imtis nedidelė. Nehierarchiniai metodai (pvz., k-vidurkių) paprastai taikomi tada, kai iš anksto žinomas (pasirenkamas) klasterių skaičius ir norima tiriama objektus klasterizuoti. Šios klasės metodai taikomi esant imčiai su nemažiau kaip šimtu elementų.

Pagrindinis bet kurio klasterizavimo metodo kriterijus yra atstumas. Objektai, kurie yra arti vienas kito priklauso tam pačiam klasteriui, o kurie toli vienas nuo kito – skirtingiems klasteriams.

Atstumui tarp klasterių vertinti yra naudojami šie metodai [7]:

- Paprastasis – artimiausio kaimyno (angl. single linkage) – skaičiuoja atstumą tarp artimiausių taškų.
- Pilnutinis – tolimiausio kaimyno (angl. complete linkage) – skaičiuoja atstumą tarp labiausiai nutolusių taškų.
- Vidurkių (angl. average linkage) - skaičiuoja visų galimų klasterių porų vidutinį atstumą.
- Ward – minimizuoja dispersiją klasterio viduje.
- Centrų (angl. centroid) – priskiriamas klasteriui tas objektas, kurio centras yra arčiausiai. Centras yra visų objektų koordinatų klasteryje vidurkis.

Nėra vienareikšmio atsakymo į klausimą, kuris jungimo metodas tinkamiausias, todėl kiekvienu atveju geriausia taikyti bent keletą jų.

Klasterizavimą apima šie penki etapai [8]:

- Klasterizavimui iš duomenų aibės pasirinkti tik dominančius objektus;
- Nuspręsti, pagal kokius požymius bus apjungiami objektai į klasterius;
- Pasirinkti kiekybinį matą, pagal kurį bus matuojamas objektų panašumas atsižvelgiant į objektų tipą, savybes;
- Vienu ar kitu metodu suskirstyti objektus į klasterius;
- Peržiūrėti gautus rezultatus. Jei gauname kelis labai didelius klasterius arba daug labai mažų klasterių, tai gali reikšti, kad duomenys buvo tiesiog tolygiai paskirstyti. Taip pat šiame etape reiktų užduoti tokius klausimus: Ar gauti duomenų klasteriai yra prasmingi? Ar algoritmas surado teisingus klasterius? Nors atsakyti į šiuos klausimus pakankamai sudėtinga, nes klasterizavimas yra subjektyvus, labai priklausantis nuo tyrėjo, tačiau remiantis įvairiais statistiniais metodais galima gauti tikslesnius atsakymus.

Teksto grupavimas gali būti panaudojamas daugelyje sričių:

- *Marketinge*: ieškant vartotojų grupių didelėje duomenų bazėje, kurioje saugoma informacija apie pačius klientus, jų pirkimo įrašai;
- *Biologijoje*: augalų ir gyvūnų klasifikavimas pagal jų savybes;
- *Bibliotekose*: knygų užsakymas;
- *Draudime*: automobilio draudimo poliso turėtojų su didelėmis draudiminėmis išmokomis identifikavimas; apgavysčių tyrimas;
- *Miesto planavime*: namų grupių išskyrimas pagal tipą, vertę ir geografinę vietovę;
- *Žemės drebėjimų tyrimuose*: stebėtų žemės drebėjimų epicentrų grupavimas išskiriant pavojingas zonas;
- *Žiniatinklyje*: dokumentų klasifikavimas, blogo duomenų grupavimas siekiant atrasti panašumų.

Būtų galima išskirti šiuos klasterizavimo technologijos trūkumus:

- Dabartiniai klasterizavimo metodai turi nemažai specifiškumo, neapima visų reikalavimų, arba juos traktuoja skirtingai.
- Gali kilti problemų turint didelius dimensijų arba duomenų kiekius dėl laiko našies sudėtingumo.
- Metodo efektyvumas priklauso nuo atstumo mato tarp klasterių pasirinkimo. (Matas – tai atstumo skaičiavimo formulė. Pagal nutylėjimą skaičiuojamas Euklido atstumo kvadratas).
- Jeigu nėra aiškų atstumo matas, jis parenkamas tyrėjo, kas nėra paprasta, ypač daugiamatėje erdvėje.
- Tiems patiems duomenims taikant skirtingus klasterinės analizės metodus, galima gauti skirtingus rezultatus.
- Klasterizavimo algoritmo rezultatai gali būti interpretuojami skirtingai arba šališkai paties tyrėjo.

2.3.3. Informacijos išrinkimas (angl. information retrieval – IR)

[9] Informacijos išrinkimas – tai technologija, susijusi su dokumentų, informacijos dokumentuose, metaduomenų, susijusių su dokumentais, paieška. Taip pat ši technologija apima ir paiešką reliacinėse duomenų bazėse, žiniatinklyje. Informacijos išrinkimas

naudojamas tiek struktūrizuotai informacijai apdoroti, pvz. duomenų bazės įrašai, tiek nestructūrizuotai, pvz. paveiksliukai, muzikos ir video failai.

[10] Ankstesnės IR sistemos buvo paremtos logine algebra. Šios sistemos leisdavo vartotojui ieškoti informacijos apibrėžiant paiešką sudėtingomis loginėmis konjunkcijos, disjunkcijos ir loginio neigimo išraiškomis. Tokios loginės algebros sistemos turėjo keletą trūkumų: nebuvo jokio dokumentų kategorizavimo, vartotojams buvo pernelyg sudėtinga suformuoti tinkamą paieškos užklausą. Nors buvo įrodyta, kad loginės algebros sistemos ne tokios efektyvios kaip kategorizavimu paremtos informacijos išgavimo sistemos, tačiau dauguma patyrusių vartotojų vis dar naudoja senąsias IR sistemas, nes turi daugiau galimybių kontroliuoti informacijos išgavimo procesą. Vis tik paprasti vartotojai labiau linkę naudoti kategorizavimu paremtas IR sistemas, kurios kiekvienam dokumentui priskiria skaitinį įvertį ir pagal jį suskirsto dokumentus į kategorijas. Šiam procesui aprašyti pasiūlyti šie modeliai: vektoriaus erdvės, tikimybinis ir išvadų tinklo modelis.

Vektoriaus erdvės modelis (angl. vector space model)

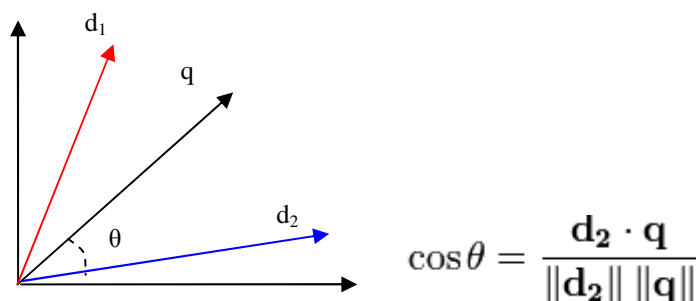
Vektoriaus erdvės modelyje tekstas yra atvaizduojamas terminų vektoriumi. Terminų apibrėžimas priklauso nuo pritaikymo. Paprastai terminai – tai vienetiniai žodžiai, raktiniai žodžiai ar ilgesnės frazės. Jei terminu pasirenkamas žodis, vektoriaus erdvės dimensijų skaičius atitiks žodžių kiekį žodyne. Tuomet bet kuris tekstas gali būti atvaizduojamas vektoriumi šioje daugiamatėje erdvėje. Jei terminas priklauso tekstui, tuomet jis įgyja nenulinę tekstinio vektoriaus reikšmę dimensijoje, susijusioje su tuo terminu. Kadangi bet kuris tekstas turi ribotą rinkinį terminų (o žodyne gali būti milijonai terminų), todėl daugelis tekstinių vektorių yra labai išsibarstę. Dauguma vektoriais paremtų sistemų operuoja su teigiamu vektoriaus erdvės kvadrantu, t.y. nei vienam terminui nepriskiriama neigiama reikšmė.

[11] Vektorių erdvės modelio procedūra gali būti suskirstyta į tris etapus:

1. Dokumento indeksavimas, kur iš dokumento išgaunamas tekstas yra susiejamas su terminais;
2. Indeksuotų terminų įvertinimas tam, kad pagreitinti vartotojui svarbaus dokumento paiešką;
3. Dokumento kategorizavimas atsižvelgiant į užklausą remiantis panašumo matavimais.

[10] Turint užklausą ir norint dokumentui priskirti skaitinį įvertį, modelis nustato panašumą tarp užklauso vektoriaus (kadangi užklausa taip pat yra tekstas ir gali būti konvertuota į vektorių) ir dokumento vektoriaus. Panašumas nustatomas lyginant nuokrypio

kampus tarp minėtų vektorių. Praktikoje yra lengviau apskaičiuoti kampo kosinusą nei patį kampą. Kosinuso nulinė reikšmė reiškia, kad užklausa ir dokumento vektoriai yra ortogonalūs (t.y. užklausa terminas neegzistuoja nagrinėjamame dokumente), ir atvirkščiai, jei kosinuso reikšmė lygi vienetui. 2 paveikslėlyje pateikiamas užklausa q ir dokumentų d_1 bei d_2 atvaizdavimas vektorių erdvėje ir kosinuso kampo θ apskaičiavimas.



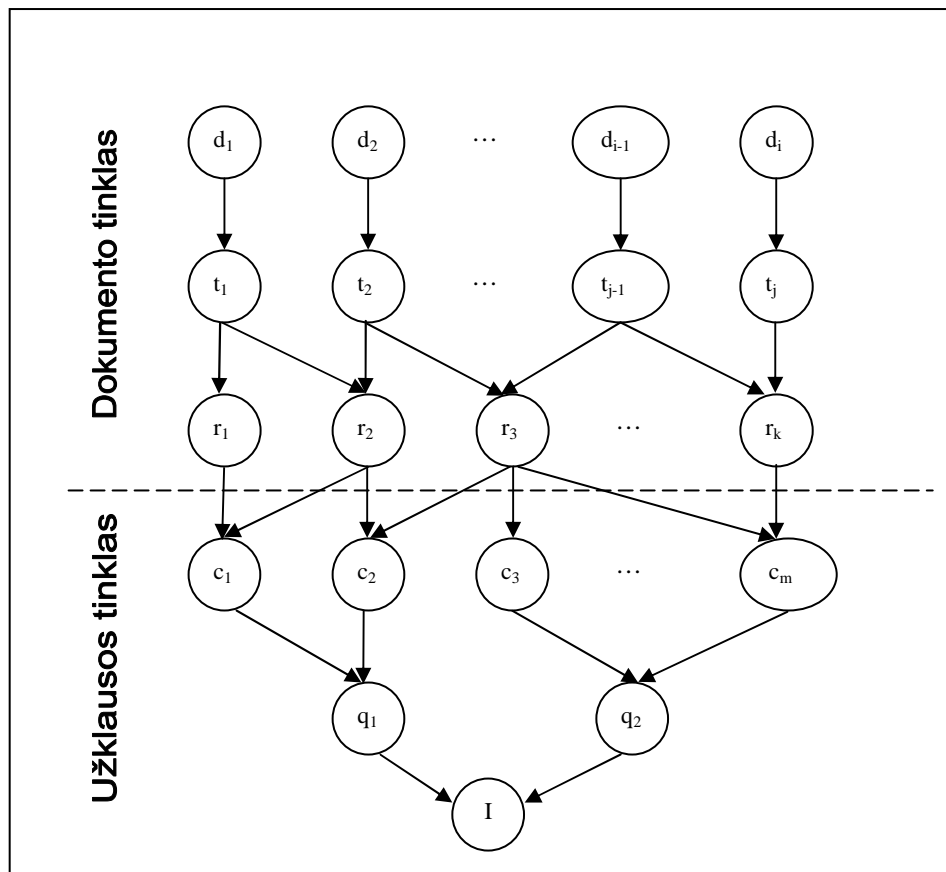
2 pav. Panašumo nuokrypio nustatymas

Tikimybinis modelis (angl. probabilistic model)

Šios šeimos IR modeliai paremti pagrindiniu principu: dokumentų rinkinys turi būti kategorizuotas mažėjančia tikimybe pagal jų svarbumą užklausiai. Tai vadinamasis tikimybinis kategorizavimo principas. Tikimybinis IR modelis apskaičiuoja dokumento svarbumo tikimybę užklausiai ir tai yra pagrindinė šio modelio esmė. Šio modelio pirminę idėją pasiūlė Maron ir Kuhns 1980 metais. Nuo tada buvo pasiūlyta daugybė šio modelio interpretacijų, kiekvienas jų paremtas vis kitokia tikimybės apskaičiavimo metodika.

Išvadų tinklo modelis (angl. inference network model)

[12] Šiame modelyje dokumento paieška yra modeliuojama kaip išvadų formavimo procesas išvadų tinkle. Daugelis metodų, naudojamų IR sistemose, gali būti realizuotos naudojant būtent šį modelį. Jis pasižymi tuo, kad turint daugybę šaltinių įrodymų, atlieka kategorizavimą kombinuodamas pateiktus įrodymus. Iš esmės šis modelis atitinka Bajeso tinklo modelį, naudojamą dokumentų, dokumento turinio ir užklausa modeliavimui. Išvadų tinklo modelis susideda iš dviejų potinklų: dokumento tinklo, gauto indeksavimo ir rezultatų išgavimo metu, bei užklausa tinklo, sudaryto iš užklausa teksto. Išvadų tinklo modelis pateiktas 3 paveikslėlyje.



3 pav. Išvadų tinklo modelis

Dokumentų tinklas vaizduoja dokumento terminų rinkinį ir susideda iš terminų viršūnių bei dokumento konceptų viršūnių. Dokumento viršūnės dalyvauja rezultato kategorizavime, o priežastinės linijos, jungiančios viršūnes, nurodo, kad tam tikras dokumento turinys yra atvaizduojamas konceptu. Kiekviena linija turi sąlygos tikimybę arba svorį, tam, kad nurodyti ryšio stiprumą. Viršūnės vertinimas atliekamas remiantis tėvų viršūnėmis ir sąlygos tikimybėmis.

Užklauso tinklas vaizduoja pateiktą užklausą ir susideda iš viršūnių karkaso, kuris atspindi reikalingus konceptus (užklauso konceptų viršūnes) ir operatorius (užklauso operatorių viršūnes), sujungtus atvirkštinėje medžio struktūroje. Užklauso tinkle yra paskutinė viršūnė, kuri vaizduoja vartotojo informacijos poreikį. Šiame tinkle naudojami statistiniai operatoriai ir loginės algebros statistinių aproksimacijų operatoriai (1 lentelė).

1 lentelė. Statistiniai operatoriai ir loginės algebros statistinių aproksimacijų operatoriai

Operatorius	Paaiškinimas
#and	IR terminai
#or	ARBA terminai
#not	Termino neigimas
#sum	Tikimybių suma
#wsum	Tikimybių sumos svoris
#max	Tikimybių maksimumas

2.4. Optinio simbolių atpažinimo sistemos

Optinis simbolių atpažinimas, sutrumpintai vadinamas OCR (angl. optical character recognition), yra nuskenuotuose paveikslukuose esančio žmogaus arba spausdinto teksto vertimas į elektroninį tekstą. OCR plačiai naudojamas knygų ir dokumentų pavertimui į jų elektronines versijas, kad kompiuterizuoti dokumentų saugojimo sistemas įstaigose arba publikuoti tekstams internete. OCR leidžias redaguoti tekstą, ieškoti informacijos jame, saugoti jį kompaktiškiau, pritaikyti tokias technologijas kaip mašininis vertimas, teksto mašininis skaitymas (angl. text-to-speech), teksto klasifikavimas.

Projekte taip pat bus naudojama optinio simbolių atpažinimo sistema. Tinkamos sistemos pasirinkimui buvo sudaryta keletas pasirinktų OCR sistemų palyginimo lentelė (lentelė 1). Lentelėje pateikti kriterijai ir įvertintas konkrečios sistemos funkcionalumas.

2 lentelė. Optinio teksto atpažinimo sistemų palyginimas

Savybė, funkcionalumas	ABBYY FineReader Engine	GOOCR	Tesseract OCR	CuneiForm
Palaiko lietuvių kalbą	+	-	+	+
Naudoja žodyną	+	-	+	+
Galima apmokyti	-	-	+	-
Veikia su .NET technologija	+	-	+	+
Paverstų paveikslukų ištiesinimas (angl. deskew)	+	-	-	+
Atviro kodo (angl. open source)	-	+	+	+
Nemokama	-	+	+	+
Toliau vystoma	+	+	+	-

ABBYY FineReader Engine [13] yra pasaulyje pripažintas vienas iš geriausių teksto atpažinimo variklių, tačiau šis variklis yra mokamas. Produkto tinklapyje net nerodoma šio variklio kaina. Dėl šio produkto kainos reikia tartis atskirai, o tai reiškia, kad kaina yra labai didelė. Todėl šis produktas netinka, nors ir turi kitas reikiamas savybes. GOOCR [14] yra nemokamas teksto atpažinimo variklis, tačiau jis veikia gana prastai, nepalaiko lietuvių kalbos ir neatitinka kitų reikiamų savybių. Abu teksto atpažinimo varikliai Tesseract OCR [15] ir CuneiForm [16] gana neblogai susidoroja su teksto atpažinimu, yra nemokami, palaiko lietuvių kalbą ir veikia .NET aplinkoje. Kadangi CuneiForm toliau nėra vystomas, tai projekte pasirinktas naudoti Tesseract OCR.

2.5. Projektavimo metodika

2.5.1. Įvadas

Bus kuriama programa skirta mobiliajam įrenginiui, analizuojanti pateiktą maisto produkto etiketės nuotrauką, kurioje bus atpažįstamas tekstas. Programa atpažintame tekste suras informaciją apie maisto produkte naudojamus priedus [17], kurie pažymėti kodais prasidedančiais raide E bei pateiks informaciją apie juos. Maisto produkto etiketės nuotrauką bus galima pateikti dviem būdais: pasinaudojant mobiliajame įrenginyje įmontuotu fotoaparatu (pagrindinis būdas) arba pateikiant nuotrauką failo pavidalu (papildomas būdas). Nuotraukos analizė vyks ne pačiame mobiliajame įrenginyje, o paprastame kompiuteryje (serveryje), todėl papildomai bus kuriama ir serverio programinė įranga. Informacija apie priedus taip pat bus laikoma serveryje. Kliento ir serverio programų bendravimui bus reikalingas Interneto ryšys. Tai yra unikalus projektas, todėl analogiškų produktų nėra sukurta.

2.5.2. Taikymo sritis

2.5.2.1. Projekto tikslas ir adresatas (kas naudosis)

Projekto tikslas yra programinė įranga, skirta mobiliajam įrenginiui, kurios pagalba mobiliojo įrenginio vartotojas galės gauti naudingos informacijos apie maisto produkte naudojamus priedus. Tai jam padės apsispręsti, ar verta vartoti maisto produktą. Programa naudosis vartotojai, turintys mobiliuosius įrenginius ir besirūpinantys tuo ką valgo.

2.5.2.2. Problemos sprendimas pasaulyje

Kadangi kuriamas projektas yra unikalus, tai analogiško sprendimo pasaulyje nėra. Galima rasti nebent dalinai poreikius atitinkančiu produktų. Pavyzdžiui, produktas scanR [18], skirtas teksto atpažinimui mobiliajame įrenginyje. Produktas scanR, pasinaudodamas mobiliojo įrenginio fotoaparatu, gauna teksto nuotrauką, kurią siunčia į serverį, o serveris grąžina atpažintą tekstą PDF [19] formatu. scanR veikia iPhone, Blackberry ir Nokia S60 mobiliųjų įrenginių platformose. Taigi, produktas scanR į kuriamą produktą yra panašus tuo, kad teksto atpažinimui naudoja serverį. Kitas dalinai poreikius atitinkantis produktas yra programa „Pocket Food Additives“ [20], kuri labai paprasta, pateikia maisto priedų sąrašą ir leidžia tame sąrašė ieškoti priedų.

2.5.2.3. Situacijos Lietuvoje įvertinimas

Kaip ir minėta anksčiau, analogiškų sprendimų nėra. Tačiau yra programinė įranga, kuri pateikia informaciją apie maisto produktų priedus. Tai UAB „TeleSoftas“ sukurta programa „SveikasVaikas mobile“ [21]. Programa yra katalogas, kuriame sudėta informacija

apie maisto produktuose naudojamus priedus. Galima atlikti paiešką pagal priedo kodą. Informacija yra gana skurdoka: priedo kategorija (priedo įtaka žmogui) ir priedo pavadinimas. Produktas veikia Java Micro Editon (J2ME) [22] platformą palaikančiuose įrenginiuose. Kuriamas produktas bus daug pranašesnis, nes pateiks reikiamą informaciją išanalizavęs maisto produkto etiketės nuotrauką, nereikės atlikti paieškos ir įvedinėti maisto produkto priedų kodų. Taip pat produktas pateiks kur kas daugiau informacijos apie priedą, nei kategorija ir pavadinimas.

2.5.3. Produkto apibūdinimas

2.5.3.1. Programų sistemos funkcijos

Kuriama programinė įranga pateiks informaciją apie maisto produkte naudojamus priedus analizuodama maisto produkto etiketės nuotrauką. Programa veiks mobiliajame įrenginyje, todėl vartotojas šį programinį produktą visuomet turės su savimi ir galės naudoti kada tik jam prireiks. Programa analizuos mobiliojo įrenginio fotoaparatu padarytą maisto produkto etiketės nuotrauką, atpažins tekstą etiketėje ir pateiks informaciją apie maisto produkte naudojamus priedus (pagal kodus etiketėje, pvz. E621, E400 ir t.t.).

Detalesnis aukščiau paminėtų programos funkcijų aprašymas:

1. Nuotraukos užkrovimas iš mobiliojo įrenginio fotoaparato

Programoje reikiamos produkto etiketės nuotrauką bus galima gauti iš mobiliajame įrenginyje esančio fotoaparato, kurio funkcionalumas bus pasiekiamas tiesiai iš programinės įrangos.

2. Nuotraukos užkrovimas iš failo

Programa reikiamos produkto etiketės nuotrauką bus galima užkrauti iš mobiliajame įrenginyje išsaugoto arba įkelto failo.

3. Reikiamos nuotraukos dalies iškirpimas

Bus galima iškirpti tiksliai nuotraukos analizei reikiamą nuotraukos dalį, t.y. nuotraukos dalį turinčią priedų kodus. Tokiu būdu bus tiksliau atpažįstamas nuotraukoje esantis tekstas ir pats atpažinimo procesas užtruks gerokai trumpiau.

4. Vaizdus rezultatų pateikimas

Informacija apie maisto produkte rastus priedus bus pateikiama vaizdžiai. Priedai bus suskirstyti į kategorijas pagal įtaką žmogaus organizmui. Kiekviena

kategorija turės savo paveikslėlį, bus pateikiama ir bendra produkto išvada. Todėl iš karto matysis, ar verta tokį maisto produktą vartoti.

5. *Detalus rezultatų pateikimas*

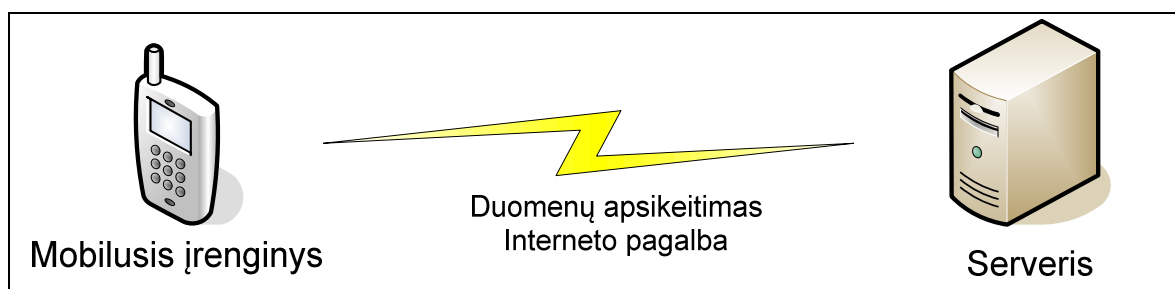
Atlikus nuotraukos analizę, bus pateikiamas rastų priedų sąrašas, kuriame matysime priedo kategorijos paveikslėlį, kodą ir pavadinimą. Galėsime pasirinkti norimą priedą iš sąrašo ir pamatyti detalią informaciją apie jį.

6. *Patogus programos valdymas*

Planuojama, kad programa veiks prisilietimais valdomame mobiliajame įrenginyje. Todėl pagrindinės programos funkcijos turės būti pasiekiamos per ganėtinai didelius grafinės vartotojo sąsajos elementus, kad nereikėtų naudoti specialaus įrankio, skirto mobiliojo įrenginio valdymui prisilietimais (angl. stylus).

2.5.3.2. Sistemos kontekstas

Kuriamas produktas bus sudarytas iš dviejų dalių: mobiliojo įrenginio kliento programinės įrangos ir serverio programinės įrangos. Kliento ir serverio programos viena su kita bendraus Interneto pagalba. Žemiau (4 pav.) pateikiama apsikeitimo duomenimis diagrama.



4 pav. Kliento ir serverio programinės įrangos apsikeitimo duomenimis diagrama

2.5.3.3. Vartotojo charakteristikos

Programine įranga galės naudotis bet kuris vartotojas, kuris moka naudotis standartinėmis mobiliojo įrenginio programomis, žino kaip mobiliojo įrenginio pagalba daromos nuotraukos, moka dirbti su mobiliojo įrenginio failais. Programinės įrangos sąsaja bus paprasta ir intuityvi.

2.5.3.4. Vartotojo problemos

Dažnas maisto produkto vartotojas pirkdamas maisto produktą nesuvokia ar maisto produktą yra sveika naudoti, ar maisto produkto geriau vengti. Ant maisto produkto etiketės yra nurodyti maisto produkte naudojamų priedų kodai. Deja tikrai nedaugelis maisto produktų vartotojų žino jų reikšmes, o ir pačių priedų rūšys skaičiuojamos šimtais. Tokio kiekio informacijos neatsimins net ir tuo besidomintis vartotojas. Todėl vartotojui būtų gerai visuomet su savimi turėti pagalbininką, kuris padėtų maisto produkto tinkamumo vartoti klausimais.

2.5.3.5. Vartotojo tikslai

Kiekvieno maisto produkto vartotojo tikslas turėtų būti galimybė žinoti, kokį poveikį žmogaus organizmui gali padaryti vartojamas maisto produktas. Deja maisto produktų gamintojai reikiamą informaciją paslepia po maisto produktų priedų kodais, kurie paprastam vartotojui yra nesuprantami. Todėl dažnas vartotojas rinkdamasis maisto produktą neturi galimybės pasirinkti analogiško, bet mažiau kenksmingo maisto produkto. Pagal maisto produkto etiketėje esančių priedų kodų kiekį taip pat negalima susidaryti nuomonės, nes kai kurie Priedai nėra pavojingi žmogaus organizmui. Taigi, kuriama programinė įranga leis vartotojui greitai, aiškiai ir intuityviai bet kurioje vietoje gauti reikiamą informaciją apie maisto produkte naudojamus priedus.

2.5.3.6. Bendri apribojimai

Kliento programinė įranga bus kuriama mobiliam įrenginiui veikiančiam Windows Mobile [23] aplinkoje, nes šio tipo įrenginiai yra pakankamai galingi susidoroti su projekto keliamais reikalavimais. Vis dėl to šie įrenginiai dar nėra tokie galingi, kad atstotų paprastą kompiuterį. Todėl pagrindiniai skaičiavimai bus atliekami kompiuteryje (serveryje). Taigi, duomenų keitimuisi tarp kliento programinės įrangos ir serverio bus reikalingas Interneto ryšys.

2.6. Projektavimo metodologijos ir technologijų analizė

2.6.1. Įvadas

Maisto skoniui, išvaizdai ir kvapui pagerinti į maistą dedami įvairūs priedai. Kai kurie maisto priedai naudojami daugelį amžių: maisto marinavimui naudojamas actas, druska, sieros dioksido panaudojimas kai kuriuose vynuose. Tam, kad šiuos priedus būtų galima reguliuoti ir informuoti vartotojus, kiekvienas priedas turi unikalų numerį. Iš pat pradžių Europoje pripažinti ir nekenksmingi priedai buvo žymimi E numeriais. Tačiau Codex Alimentarius komisija [25] (1963 m. įkurtas Pasaulinės sveikatos organizacijos organas, kuris

rūpinasi vartotojų sveikata ir tarptautine maisto prekyba [27]) pritaikė šiuos numerius tarptautiniu mastu ir jais žymimi visi maisto priedai, nesvarbu, leistini vartoti, ar ne.

Jeigu maisto priedas žymimas raide E, tai reiškia, kad jis atitinka saugos reikalavimus ir jį leidžiama vartoti visoje ES [26]. Šis leidimas stebimas, peržiūrimas ir koreguojamas atsižvelgiant į naujus mokslinius faktus. Bet kai kurios šalys, nepriklausančios Europai, naudoja tik numerius be prefikso E, nepriklausomai, ar priedas patvirtintas Europoje, ar ne. Pavyzdžiui, acto rūgštis užrašoma kaip E260 ant produktų, parduodamų Europoje, bet kai kuriose kitose šalyse tiesiog užrašomas skaičius 260.

Skirtingi E numerio intervalai žymi skirtingas priedų grupes [26], [27]:

- *Antioksidantai* E300 – E399: prailgina maisto produktų vartojimo trukmę, neleisdami riebalams, augaliniam aliejui ir tam tikriems vitaminams jungtis su deguonimi ore.
- *Dažikliai* E100 – E199: kartais naudojami, kai apdorojant ar sandėliuojant maisto produktus dingsta jų natūrali spalva arba kai produktui norima suteikti pastovų atspalvį.
- *Emulsikliai, stabilizatoriai, stingdančios medžiagos ir tirštikliai* E400 – E599. Emulsikliai padeda sumaišyti sudėtinės dalis, kurios paprastai nesusimaišo. Stabilizatoriai neleidžia sumaišytoms sudedamosioms dalims vėl atsiskirti.
- *Konservantai* E200 – E299: neleidžia maistui gesti. Konservantai naudojami daugelyje maisto produktų, kurių galiojimo laikas ilgas, jeigu nenaudojamas kitas konservavimo būdas, pavyzdžiui, užšaldymas, konservavimas arba džiovinimas.
- *Saldikliai* E900 – E999: dažnai naudojami vietoj cukraus tokiuose produktuose, kaip putojantys nealkoholiniai gėrimai, jogurtai ir kramtomoji guma.

2.6.2. Maisto produktų ženklėjimas

Maisto produktų etiketės – nepaprastai svarbi ir vartotojų pasirinkimą lemianti informavimo priemonė. Aiškesnis ženklėjimas padėtų vartotojams pasirinkti sveikesnį maistą, kadangi būtų aiškiau matyti, ką perkame. Etiketėje turi būti pateikiama ši informacija [32]:

- sudedamųjų dalių sąrašas;
- tinkamumo vartoti terminas arba „Tinka vartoti iki... (data)“;
- specialios laikymo arba vartojimo sąlygos, pavyzdžiui, „laikyti sausoje vėsioje vietoje“, „galima užšaldyti“;
- produktų grynavis (neto) kiekis kilogramais ar litrais;

- gamintojo pavadinimas ir adresas;
- kita informacija, nurodyta teisės aktuose.

Informacija etiketėse turi būti pateikta lietuvių kalba. Šiuos duomenis siūloma pateikti ne smulkesniu kaip 3 mm dydžio šriftu, kad nereiktų vargti ieškant dominančios informacijos [33].

Produktai, pagaminti ar turintys genetiškai modifikuotų organizmų, turi būti pažymėti užrašu „Pagaminti iš genetiškai modifikuotų...“, „Genetiškai modifikuoti organizmai“, „Produkto sudėtyje yra genetiškai modifikuotų organizmų“. Trumpiniai GM ar GMO draudžiami.

Visi teiginiai apie ypatingas maisto savybes, pavyzdžiui, „mažai riebalų“, „daug skaidulinių medžiagų“ ar „padeda sumažinti cholesterolio kiekį“, turi būti pagrįsti ir įrodyti moksliniais tyrimais [32].

Vartotojas gali rinktis, kokius produktus vartoti. Visai neseniai sertifikuoti ekologiški, nepavojingi ir naudingi sveikatai maisto produktai pradėti žymėti atitinkamais logotipais.

„Ekologinis žemės ūkis“ logotipu (5 pav.) žymimi ekologiški produktai. Ar produktas ekologiškas, nustatoma remiantis Tarybos reglamentais (EB) Nr. 834/2007 ir Nr. 889/2008, ir Lietuvos įstatymais. Ekologiškas produktas visada bus paženklintas etikete, kurioje pateikiama pati svarbiausia informacija: ekologiškų produktų sertifikavimo ženklas, identifikuotas gamintojas, sertifikavimo darbus atlikusi sertifikacijos įstaiga [30].



5 pav. Ekologinio žemės ūkio logotipas

Viešoji įstaiga „Sveiko vaiko institutas“ [31] ne pelno siekianti organizacija, kurios pagrindinis tikslas – padidinti natūralesnių maisto produktų pasirinkimą Lietuvoje. Norėdamas gauti šios organizacijos ženklą (6 pav.), gamintojas turi nurodyti produktą, kuris atitinka šiuos kriterijus:

- be cheminių maisto priedų, žymimų E numeriais;
- be GMO (genetiškai modifikuotų organizmų);
- kurio gamyba ir žaliavos yra kruopščiai atrenkamos ir prižiūrimos.

Prekinis ženklas suteikiamas tik patikrintiems produktams, jų sudėtį ištyrus Nacionalinėje maisto ir veterinarijos rizikos vertinimo instituto (NMVRVI) laboratorijoje arba Kauno Technologijos universiteto Maisto institute.



6 pav. Sveikas vaikas logotipas

Šiuo metu vyksta ES ekologiško gaminio logotipo konkursas. Virš 3400 studentų iš visų 27 ES valstybių narių dalyvavo ES ekologiško gaminio logotipo konkurse [29]. Jau atrinkti trys logotipo kandidatai (7 pav.) iš kurių bus išrinktas laimėtojas. Naujasis logotipas bus pradėtas naudoti nuo 2010-ųjų liepos mėnesio. Jis bus klijuojamas ant fasuotų ekologiškų gaminių, kurie atitinka Bendrijos reikalavimus, nustatytus ekologiniam ūkininkavimui ir gamybai [30].



7 pav. Trys kandidatai į ES ekologiško gaminio logotipą

2.7. Egzistuojantys sprendimai

Esant tokiai gausybei maisto priedų, vartotojas turėtų teisę žinoti, ar jo pasirinktas maisto produktas sveikas ar ne. Tačiau jų kenksmingumas užkoduotas po E kodais, kurių visų prisiminti tiesiog neįmanoma. Internete galima rasti maisto priedų sąrašų, tačiau programinės įrangos pasirinkimas šia tema yra skurdus.

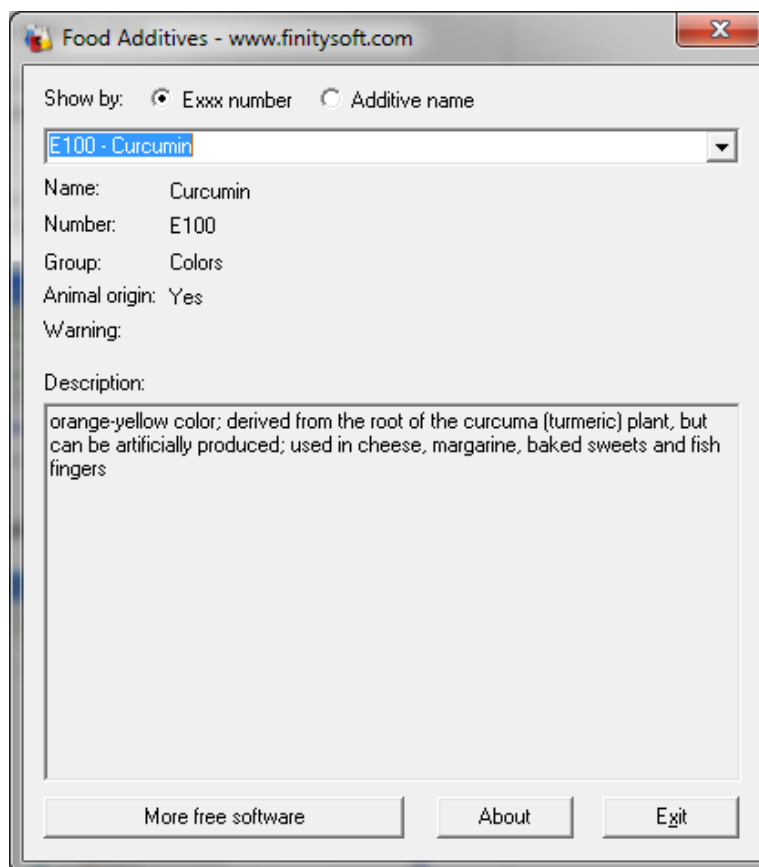
2.7.1. Programa „Food Additives“

Tai nemokama kompanijos FinitySoft [28] programa veikianti Windows operacinėse sistemose.

Pagrindinės funkcijos:

- Paieška (renkantis iš sąrašo arba įvedant tekstą):
 - pagal E kodą;
 - pagal priedo pavadinimą;
- Suteikiama informacija:
 - Priedo pilnas pavadinimas ir E kodas;
 - Apie priedų grupę, kuriai priklauso priedas;
 - Ar priedas yra gyvūninės kilmės;
 - Įspėjimą, jei priedas sukelia tam tikras ligas ar yra uždraustas vartoti.

Pagrindinis ir vienintelis programos langas pateiktas 8 paveikslėlyje.



8 pav. Programos „Food Additives“ langas

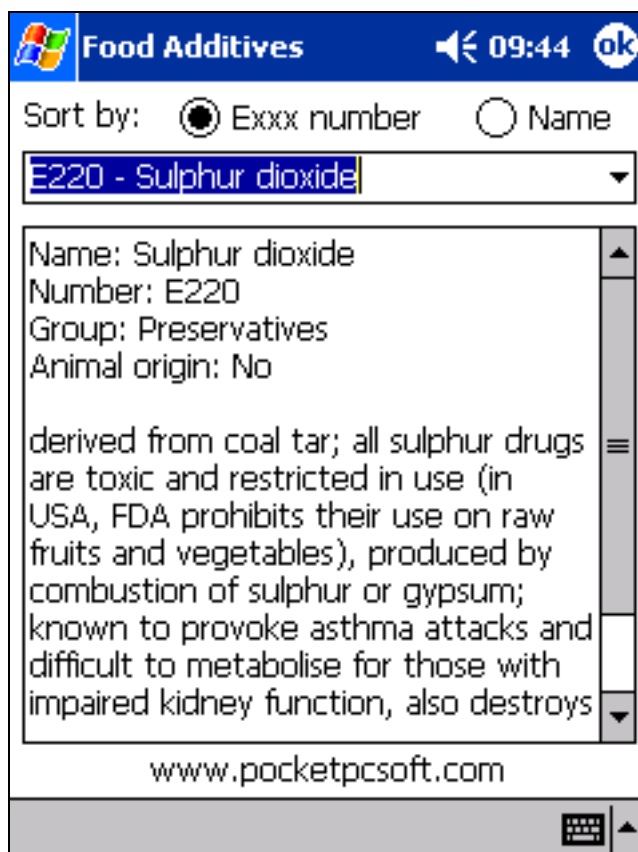
2.7.2. Programa „Pocket Food Additives“

Tai nemokama kompanijos PocketPCSoft [20] programa veikianti Windows Mobile operacinėse sistemose.

Pagrindinės funkcijos:

- Paieška (renkantis iš sąrašo arba įvedant tekstą):
 - pagal E kodą;
 - pagal priedo pavadinimą;
- Suteikiama informacija:
 - Priedo pilnas pavadinimas ir E kodas;
 - Apie priedų grupę, kuriai priklauso priedas;
 - Ar priedas yra gyvūninės kilmės;
 - Įspėjimą, jei priedas sukelia tam tikras ligas ar yra uždraustas vartoti.

Pagrindinis ir vienintelis programos langas pateiktas 9 paveikslėlyje.



9 pav. Programos „Pocket Food Additives“ langas

2.7.3. Programa „SveikasVaikas mobile“

Tai nemokama lietuviškos kompanijos TeleSoftas [21] programa, kuri pritaikyta veikti J2ME platformą palaikančiuose mobiliuosiuose įrenginiuose.

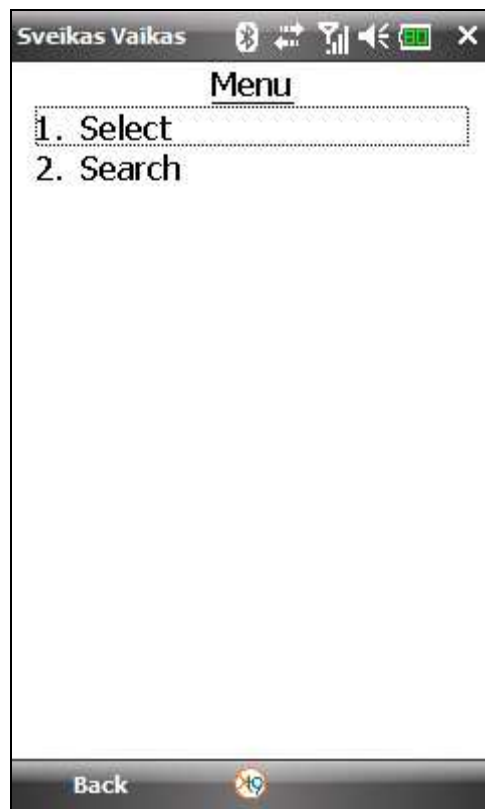
Pagrindinės funkcijos:

- Paieška (tiek renkantis iš sąrašo, tiek vedant tekstą):
 - pagal E kodą;
- Suteikiama informacija:
 - Priedo pilnas pavadinimas;
 - Priedo kenksmingumas atvaizduojamas ikona ir užrašomas žodžiais.

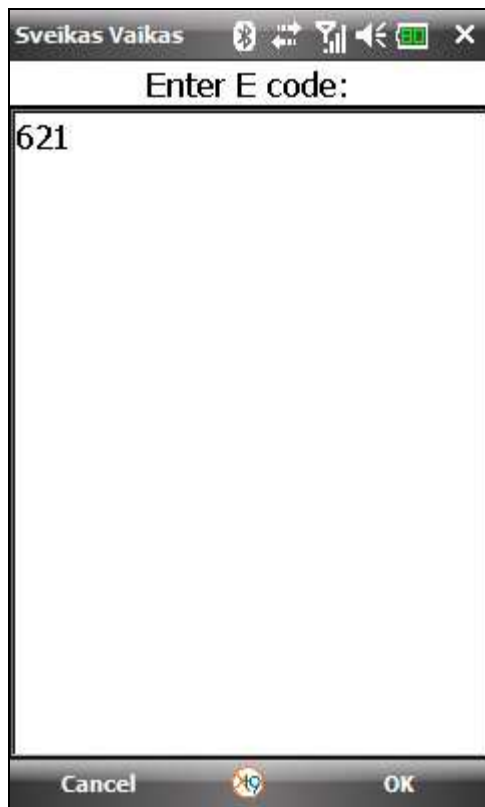
Programos „SveikasVaikas mobile“ langai pateikti 10 – 13 paveikslėliuose.



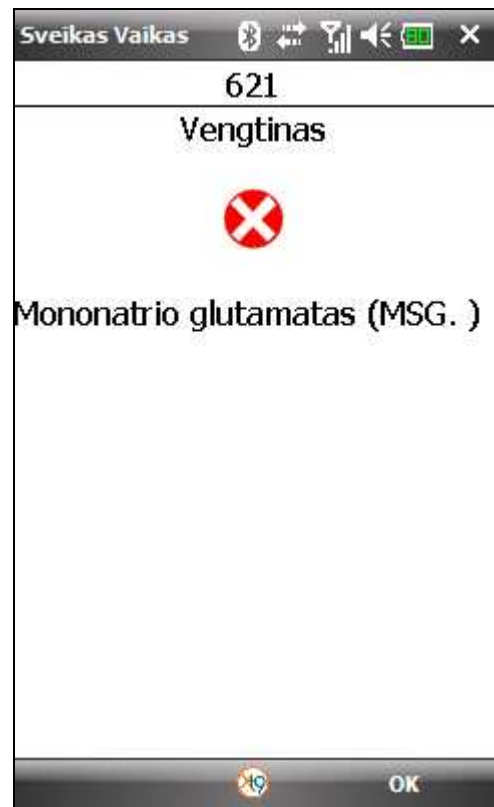
10 pav. Visų priedų sąrašas



11 pav. Meniu



12 pav. Paieška



13 pav. Informacija apie priedą

Lentelėje 3 yra pateiktas aukščiau aprašytų programų palyginimas. Lentelėje pateikti kriterijai ir įvertintas konkrečios programos funkcionalumas.

3 lentelė. Programų palyginimas

Savybė, funkcionalumas	Food Additives	Pocket Food Additives	SveikasVaikas mobile
Patogi paieška	+	+	+/-
Galimybė atlikti paiešką pagal E kodą ir maisto priedo pavadinimą	+	+	+/-
Vaizdus paieškos rezultatų pateikimas	-	-	+
Pateikta informacija yra išsami ir pakankama	+	+	-
Veikimas mobiliajame įrenginyje	-	+	+
Galimybė OCR priemonių pagalba iš maisto produkto etiketės išskirti visus maisto priedus, pažymėtus E kodu.	-	-	-
Galimybė iš maisto priedų sąrašo nustatyti kenksmingiausią.	-	-	-
Galimybė patogiai valdyti programą rankų pirštais	-	-	-

2.8. Analizės išvados

Išanalizavus klasifikavimo problemą, su ja susijusias technologijas ir projekto problematiką, buvo nuspręsta naudoti teksto šablonų sulyginimo (angl. string pattern matching) technologiją – reguliariąsias išraiškas (angl. regular expressions). Reguliariosios išraiškos pasirinktos todėl, kad jomis galima lanksčiai aprašyti tekste esančius požymius, analizuojamos tekstinės informacijos kiekis nebus didelis, todėl teksto analizės greitis nėra svarbus.

Teksto atpažinimui iš paveikslėlio pasirinktas Tesseract OCR [15] optinio simbolių atpažinimo variklis, nes jis gerai susidoroja su teksto atpažinimu, yra nemokamas, palaiko lietuvių kalbą ir veikia su .NET technologija, kuri bus naudojama projekte.

3. PROJEKTO REIKALAVIMŲ SPECIFIKAVIMO DALIS

3.1. Projekto varovai (angl. project drivers)

3.1.1. Sistemos paskirtis

3.1.1.1. Projekto kūrimo pagrindas (pagrindimas)

Daugelis maisto produkto vartotojų pirkdami maisto produktus nesuvokia ar tam tikrą maisto produktą yra sveika naudoti, ar maisto produkto geriau vengti. Ant maisto produkto etiketės yra nurodyti maisto produkte naudojamų priedų kodai. Deja tikrai nedaugelis maisto produktų vartotojų žino jų reikšmes, o ir pačių priedų rūšys skaičiuojamos šimtais. Tokio kiekio informacijos neatsimins net ir tuo besidomintis vartotojas. Todėl vartotojui būtų gerai visuomet su savimi turėti pagalbininką, kuris padėtų maisto produkto tinkamumo vartoti klausimais.

Kiekvieno maisto produkto vartotojo tikslas turėtų būti galimybė žinoti, kokį poveikį žmogaus organizmui gali padaryti vartojamas maisto produktas. Deja, maisto produktų gamintojai reikiamą informaciją paslepia po maisto produktų priedų kodais, kurie paprastam vartotojui yra nesuprantami. Todėl dažnas vartotojas rinkdamasis maisto produktą, neturi galimybės pasirinkti analogiško, bet mažiau kenksmingo maisto produkto. Pagal maisto produkto etiketėje esančių priedų kodų kiekį taip pat negalima susidaryti nuomonės, nes kai kurie priedai nėra pavojingi žmogaus organizmui.

3.1.1.2. Sistemos tikslai (paskirtis)

Kuriama programinė turi vieną pagrindinį tikslą – maisto produkto vartotojui padėti sužinoti daugiau reikiamos informacijos apie maisto produkte naudojamus priedus. Šį tikslą vartotojas turėtų pasiekti paprastu, visur prieinamu, greitai reikiamus atsakymus duodančiu būdu. Iš to išplaukia ir sekantis sistemos tikslas – programinė įranga turi būti mobili. Tikslams pasiekti pasirinktas paprastas, bet labai lankstus būdas. Programinė įranga analizuos maisto produkto etiketės nuotrauką, t.y. joje atpažins tekstą, tekste ras maisto produkte naudojamų priedų kodus ir pagal rastus kodus vartotojui pateiks reikiamą informaciją.

3.1.2. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys

Sistemos užsakovas yra Kauno technologijos universiteto (KTU), programų inžinerijos katedros vedėjas prof. Eduardas Bareiša. Užsakovo organizacija yra Kauno Technologijos Universitetas, programų inžinerijos katedra.

Programų inžinerijos katedra kuruoja:

- Informatikos bakalauro studijų Programavimo technologijų specializaciją,

- Programų sistemų inžinerijos magistro studijas,
- Vienlusčių sistemų magistro studijas.

Pagrindinės mokslinių tyrimų kryptys:

- Elektroninės aparatūros automatizuotas projektavimas ir testavimas;
- Taikomosios srities aktyvių komponentų tyrimas;
- Distancinių studijų tyrimas ir diegimas Lietuvoje.

Sistemos kūrėjas yra KTU „Programų sistemų inžinerijos“ magistrantūros studentas Paulius Grabauskas.

3.1.3. Vartotojai

Programine įranga galės naudotis bet kuris vartotojas, kuris moka naudotis standartinėmis mobiliojo įrenginio programomis, žino kaip mobiliojo įrenginio pagalba daromos nuotraukos, moka dirbti su mobiliojo įrenginio failais. Programinės įrangos sąsaja bus paprasta ir intuityvi.

3.2. Projekto apribojimai

3.2.1. Apribojimai sprendimui

Kliento programinė įranga turės veikti Windows Mobile Professional 6.1 aplinkoje. Mobilusis įrenginys, kuriame veiks kliento programinė įranga turės turėti WQVGA (240 x 400) ekrano raišką, 128MB operatyviosios atminties bei fotoaparatai turinti autofokusą ir minimalią 2MP raišką. Kliento programinei įrangai taip pat reikalingas Interneto ryšys, todėl ir mobilus įrenginys turi palaikyti Interneto ryšį. Kliento programinė įranga bus programuojama naudojantis .NET Compact Framework programinio kodo bibliotekomis.

Serverio programinė įranga veiks Windows operacinėje sistemoje, kurioje įdiegtas IIS (angl. Internet Information Services) serveris bei ASP.NET programinio kodo bibliotekos. Duomenys apie priedus bus saugomi serveryje, XML failuose.

3.2.2. Diegimo aplinka

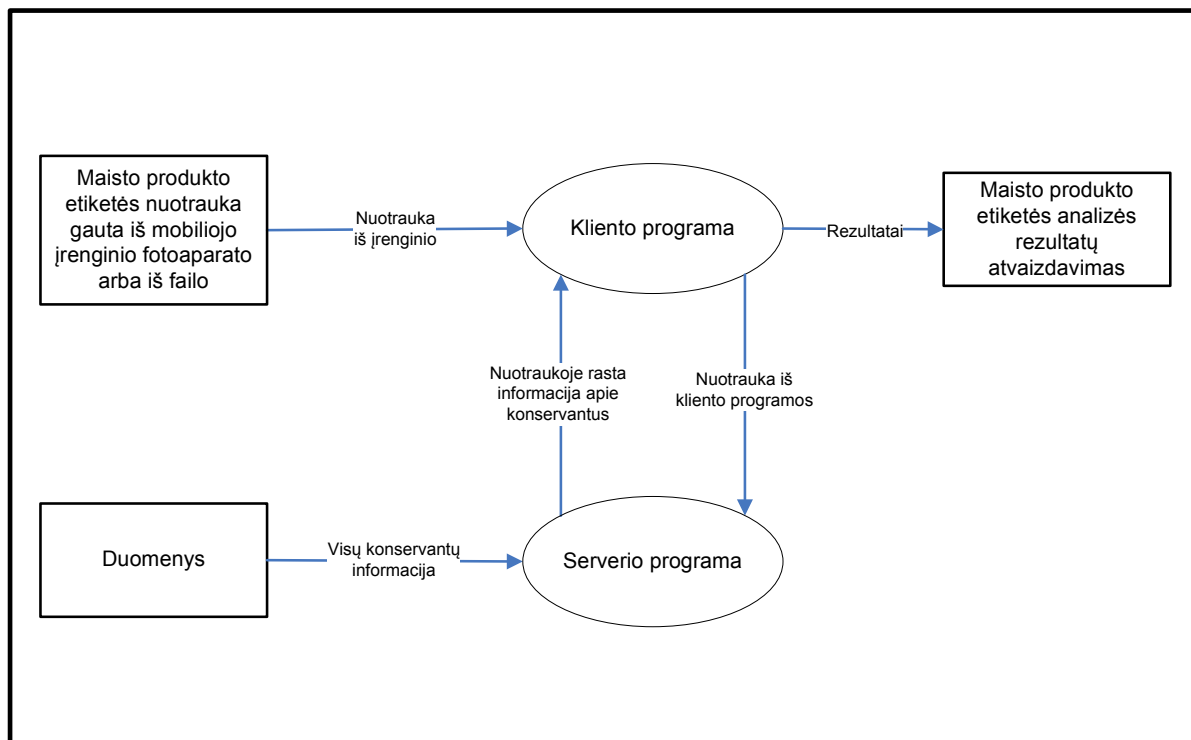
Kliento programinė įranga bus diegiama mobiliajame įrenginyje, kuris veikia Windows Mobile Professional 6.1 aplinkoje bei turi įdiegtas .NET Compact Framework 3.5 bibliotekas. Serverio programinė įranga bus diegiama kompiuteryje, kuriame reikia Windows operacinė sistema palaikanti IIS 5.1 arba naujesnę serverio programinę įrangą, bei turinti įdiegtas ASP.NET 3.5 programinio kodo bibliotekas.

3.3. Funkciniai reikalavimai

3.3.1. Veiklos sudėtis (angl. the scope of the work)

3.3.1.1. Veiklos kontekstas

Sistemos veiklos kontekstas pavaizduotas 14 paveikslėlyje.



14 pav. Veiklos kontekstas

3.3.1.2. Veiklos padalinimas

Lentelėje 4 pateikiamas kliento programos veiklos įvykių sąrašas.

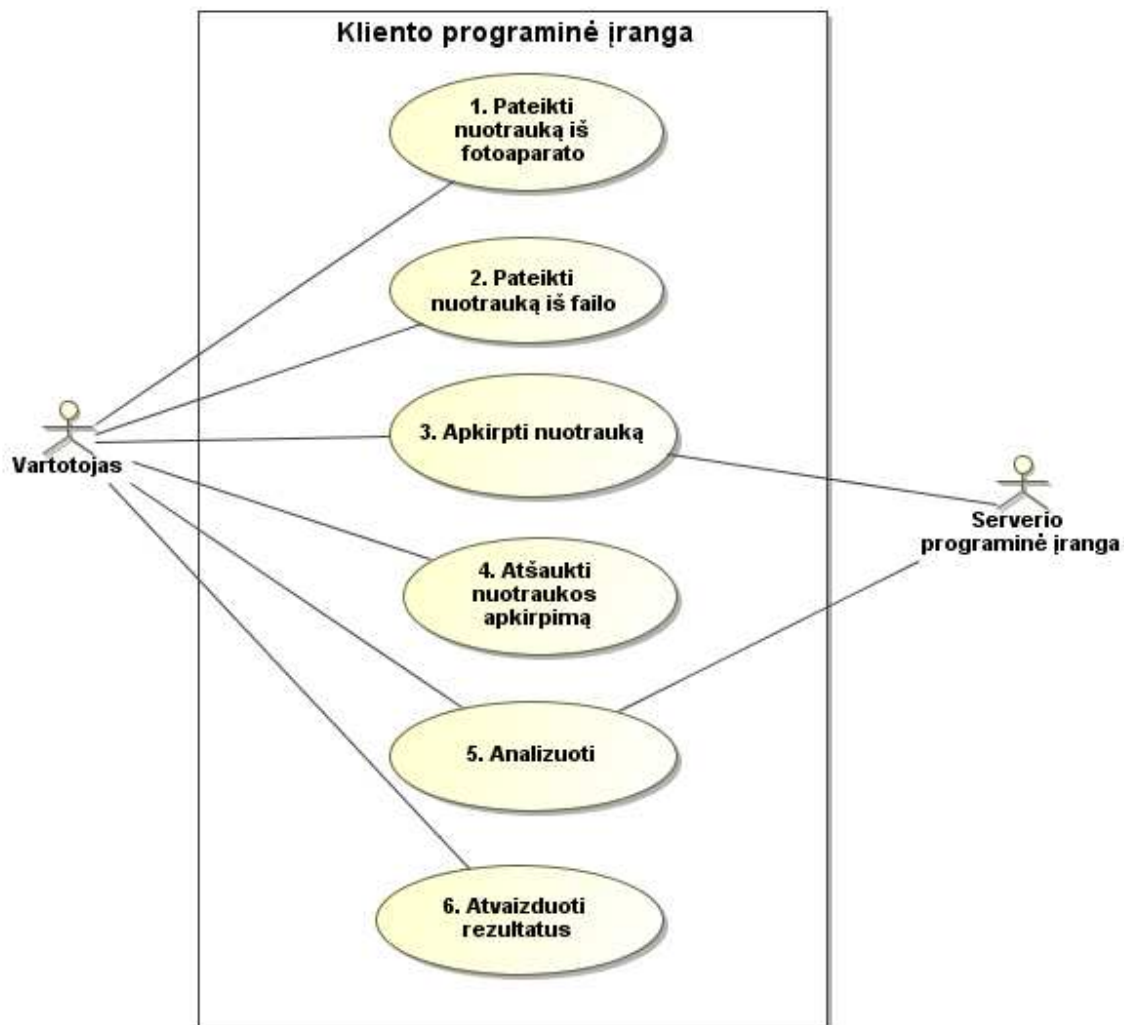
4 lentelė. Veiklos įvykių sąrašas

Eil. nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1.	Kliento programai pateikta produkto etiketės nuotrauka	Nuotrauka iš įrenginio (in)
2.	Serverio programai pateikta produkto etiketės nuotrauka	Nuotrauka iš kliento programos (out)
3.	Serverio kliento programa pateikti duomenys apie maisto produkto etiketės naudojamus priedus	Nuotraukoje rasta informacija apie priedus (in)
4.	Kliento programa pateikia rezultatus	Rezultatai (out)

3.3.2. Sistemos sudėtis (angl. the scope of the product)

3.3.2.1. Sistemos ribos

Sistemos ribos atvaizduotos panaudojimo atvejų diagrama, kuri pateikta 15 pav.



15 pav. Panaudojimo atvejų diagrama

3.3.2.2. Panaudojimo atvejų sąrašas

5 lentelė. Panaudojimo atvejo „1. Pateikti nuotrauką iš fotoaparato“ aprašymas

1. Panaudojimo atvejis: Pateikti nuotrauką iš fotoaparato	
Aprašas	Apima veiklą, kurios metu vartotojas pasinaudodamas mobiliojo įrenginio fotoaparatu kliento programai pateikia maisto produkto etiketės nuotrauką.
Aktoriai	Vartotojas.
Prieš-sąlygos	Maisto produkto etiketė.
Sužadinimo sąlyga	Programoje vartotojas vykdo fotoaparato įjungimo veiksmą.
Po-sąlygą	Į programą įkelta maisto produkto etiketės nuotrauka.
Pagrindinis scenarijus	Vartotojas programoje pasirenka nuotraukos darymo veiksmą, nufotografuoja maisto produkto etiketę ir patvirtina nuotraukos padarymą. Programos lange rodoma padaryta nuotrauka.
Alternatyvūs scenarijai	Nėra.

6 lentelė. Panaudojimo atvejo „2. Pateikti nuotrauką iš failo“ aprašymas

2. Panaudojimo atvejis: Pateikti nuotrauką iš failo	
Aprašas	Apima veiklą, kurios metu vartotojas pasinaudodamas mobiliojo įrenginio failų sistema kliento programai pateikia maisto produkto etiketės nuotrauką.
Aktoriai	Vartotojas.
Prieš-sąlygos	Maisto produkto etiketės nuotrauka mobiliojo įrenginio faile.
Sužadinimo sąlyga	Programoje vartotojas vykdo nuotraukos failo parinkimo veiksmą.
Po-sąlyga	I programą įkelta maisto produkto etiketės nuotrauka.
Pagrindinis scenarijus	Vartotojas programoje pasirenka nuotraukos failo parinkimo veiksmą, suranda ir parenka nuotraukos failą. Programos lange rodoma parinkta nuotrauka.
Alternatyvūs scenarijai	Nėra.

7 lentelė. Panaudojimo atvejo „3. Apkirpti nuotrauką“ aprašymas

3. Panaudojimo atvejis: Apkirpti nuotrauką	
Aprašas	Apima veiklą, kurios metu vartotojas kliento programoje gali apkirpti įkeltą maisto produkto etiketės nuotrauką.
Aktoriai	Vartotojas.
Prieš-sąlygos	I programą įkelta maisto produkto etiketės nuotrauka.
Sužadinimo sąlyga	Programoje vartotojas pažymi nuotraukos vietą, kuri bus apkirpta ir vykdo nuotraukos apkirpimo veiksmą.
Po-sąlyga	Įsimenamos apkerpamos nuotraukos dalies koordinatės, programoje atvaizduojama tiksliai apkirpimui paruošta dalis. Vykdamas panaudojimo atvejį „5. Analizuoti“, nuotrauka bus apkirpta serveryje.
Pagrindinis scenarijus	Vartotojas programoje pažymi apkerpamą nuotraukos dalį ir pasirenka nuotraukos apkirpimo veiksmą. Programos lange, rodoma apkirpta nuotraukos dalis.
Alternatyvūs scenarijai	Nėra.

8 lentelė. Panaudojimo atvejo „4. Atšaukti nuotraukos apkirpimą“ aprašymas

4. Panaudojimo atvejis: Atšaukti nuotraukos apkirpimą	
Aprašas	Apima veiklą, kurios metu vartotojas kliento programoje gali sugrąžinti apkirptą nuotrauką į pradinę padėtį, t.y. atstatyti nuotraukos vaizdą prieš apkirpimą.
Aktoriai	Vartotojas.
Prieš-sąlygos	Apkirpta maisto produkto etiketės nuotrauka.
Sužadinimo sąlyga	Programoje vartotojas vykdo nuotraukos apkirpimo atšaukimo veiksmą.
Po-sąlyga	Pašalinamos nuotraukos apkirpimo koordinatės, programoje atvaizduojama visa nuotrauka.
Pagrindinis scenarijus	Vartotojas programoje pasirenka nuotraukos apkirpimo atšaukimo veiksmą. Programos lange, rodoma visa nuotrauka.
Alternatyvūs scenarijai	Nėra.

9 lentelė. Panaudojimo atvejo „5. Analizuoti“ aprašymas

5. Panaudojimo atvejis: Analizuoti	
Aprašas	Apima veiklą, kurios metu serveriui perduodama maisto produkto etikės nuotrauka bei nuotraukos apkirpimo duomenys (jeigu nuotrauka turėtų būti apkirpta prieš analizę). Toliau analizuojama pateikta nuotrauka, kad būtų išgauti nuotraukoje esantys maisto produkto priedų kodai ir kliento programai gražinami rezultatai.
Aktoriai	Serverio programinė įranga.
Prieš-sąlygos	Maisto produkto etikės nuotrauka.
Sužadinimo sąlyga	Programoje vartotojas vykdo maisto produkto etikės analizės veiksmą.
Po-sąlyga	Kliento programai serveris pateikia informaciją apie maisto produkto etiketėje rastus priedus.
Pagrindinis scenarijus	Vartotojas programoje pasirenka analizės veiksmą. Tuomet kliento programinė įranga serveriui siunčia maisto produkto etikės nuotrauką ir kitus reikiamus duomenis. Serveris išanalizavęs nuotrauką, kliento programai gražina informaciją apie maisto produkto etikės nuotraukoje rastus priedus.
Alternatyvūs scenarijai	Nėra.

10 lentelė. Panaudojimo atvejo „6. Atvaizduoti rezultatus“ aprašymas

6. Panaudojimo atvejis: Atvaizduoti rezultatus	
Aprašas	Apima veiklą, kurios metu kliento programinė įranga atvaizduoja iš serverio programinės įrangos gautus rezultatus.
Aktoriai	Serverio programinė įranga.
Prieš-sąlygos	Pateikti duomenys apie maisto produkto etiketėje rastus priedus.
Sužadinimo sąlyga	Serverio programinė įranga pateikia informaciją apie maisto produkto etiketėje rastus priedus kliento programinei įrangai.
Po-sąlyga	Kliento programoje atvaizduojami duomenys apie maisto produkto etiketėje rastus priedus.
Pagrindinis scenarijus	Serverio programinė įranga kliento programai atsiunčia informaciją apie maisto produkto etiketėje rastus priedus. Kliento programinė įranga atvaizduoja rastų priedų sąrašą.
Alternatyvūs scenarijai	Serverio programinė įranga kliento programai atsiunčia informaciją apie maisto produkto etiketėje rastus priedus. Kliento programinė įranga atvaizduoja rastų priedų sąrašą. Vartotojas išsirenka konkretų priedą iš sąrašo ir peržiūri detalią informaciją apie priedą.

3.3.2.3. Funkciniai reikalavimai

11 lentelė. Funkciniai reikalavimai panaudojimo atvejui „1. Pateikti nuotrauką iš fotoaparato“

1. Panaudojimo atvejis: Pateikti nuotrauką iš fotoaparato	
Reikalavimo pavadinimas	Reikalavimo aprašymas
1.1 FR.CameraPhoto.LoadCamera	Galima įjungti mobiliojo įrenginio fotoaparata.
1.2 FR.CameraPhoto.TakePhoto	Galima gauti nuotrauką iš mobiliojo įrenginio fotoaparato.
1.3 FR.CameraPhoto.LoadPhoto	Galima užkrauti padarytą nuotrauką į kliento programą.
1.4 FR.CameraPhoto.ShowPhoto	Galima atvaizduoti nuotrauką kliento programos lange.

12 lentelė. Funkciniai reikalavimai panaudojimo atvejui „2. Pateikti nuotrauką iš failo“

2. Panaudojimo atvejis: Pateikti nuotrauką iš failo	
Reikalavimo pavadinimas	Reikalavimo aprašymas
2.1 FR.FilePhoto.ChoosePhoto	Galima pasirinkti nuotraukos failą iš mobiliojo įrenginio failų sistemos.
2.2 FR.FilePhoto.LoadFile	Galima užkrauti pasirinktos nuotraukos failą į kliento programą.
2.2 FR.FilePhoto.ShowPhoto	Galima atvaizduoti užkrautą nuotrauką kliento programos lange.

13 lentelė. Funkciniai reikalavimai panaudojimo atvejui „3. Apkirpti nuotrauką“

3. Panaudojimo atvejis: Apkirpti nuotrauką	
Reikalavimo pavadinimas	Reikalavimo aprašymas
3.1 FR.CropPhoto. SelectRectangle	Galima pažymėti nuotraukos dalį.
3.2 FR.CropPhoto.GetCoordinates	Galima gauti pažymėtos nuotraukos dalies koordinates.
3.3 FR.CropPhoto. LoadRectanglePhotoPart	Galima atvaizduoti pažymėtą nuotraukos dalį kliento programos lange.
3.4 FR.CropPhoto.SetCoordinates	Galima serveriui nusiųsti iškerpamos nuotraukos dalies koordinates.
3.5 FR.CropPhoto. CropUploadedPhoto	Galima apkirpti į serverį nusiųstą nuotrauką.

14 lentelė. Funkciniai reikalavimai panaudojimo atvejui „4. Atšaukti nuotraukos apkirpimą“

4. Panaudojimo atvejis: Atšaukti nuotraukos apkirpimą	
Reikalavimo pavadinimas	Reikalavimo aprašymas
4.1 FR.UncropPhoto.DeleteCoordinates	Galima pašalinti įsimintas nuotraukos pažymėjimo koordinates.
4.2 FR.UncropPhoto.LoadPhoto	Galima užkrauti visą nuotrauką.

15 lentelė. Funkciniai reikalavimai panaudojimo atvejui „5. Analizuoti“

5. Panaudojimo atvejis: Analizuoti	
Reikalavimo pavadinimas	Reikalavimo aprašymas
5.1 FR.Analyse.Connect	Galima prisijungti prie serverio programinės įrangos.
5.2 FR.Analyse.UploadPhoto	Galima serveriui nusiųsti maisto produkto etiketės nuotrauką.
5.3 FR.Analyse.AnalysePhoto.GetText	Galima gauti nusiųstos į serverį nuotraukos tekstą.
5.3 FR.Analyse.AnalysePhoto.FindPreservativesCodes	Galima rasti išgautame iš nuotraukos tekste esančius maito produkto priedų kodus.
5.4 FR.Analyse.AnalysePhoto.GetPreservativesInformation	Pagal iš nuotraukos išgautus maito produkto priedų kodus, galima gauti priedų informaciją.
5.5 FR.Analyse.AnalysePhoto.UploadPreservativesInformation	Informaciją apie maito produkto etiketėje rastus priedus, galima nusiųsti į kliento programinę įrangą.

16 lentelė. Funkciniai reikalavimai panaudojimo atvejui „6. Atvaizduoti rezultatus“

6. Panaudojimo atvejis: Atvaizduoti rezultatus	
Reikalavimo pavadinimas	Reikalavimo aprašymas
6.1 FR.Results.ShowList	Galima atvaizduoti maisto produkto etiketėje rastų priedų sąrašą.
6.2 FR. Results.ShowMainConclusion	Galima atvaizduoti bendrą maisto produkto išvadą, kuri parenkama pagal rastą labiausiai žmogaus organizmui kenkiantį priedą.
6.3 FR. Results.ShowDetails	Galima atvaizduoti iš sąrašo pasirinkto priedo detalią informaciją.

3.3.2.4. Reikalavimai duomenims

Programai pateikiamos nuotraukos maksimalus dydis pikseliais 1600x1200. Nuotraukoje esantis teksto vaizdas turi būti aiškus, sufokusuotas, minimalus šrifto dydis 8 punktai. Pageidautina, kad nuotraukoje esantis tekstas būtų juodos spalvos. Tikrai tenkinant šiuos reikalavimus, programa korektiškai atpažins tekstą.

Informacija apie priedus bus laikoma XML failų pavidalu. Duomenys apie maisto produkte naudojamus priedus iš serverio programos į kliento programą taip pat bus perduodami XML formatu.

3.4. Nefunkciniai reikalavimai

3.4.1. Reikalavimai sistemos išvaizdai (angl. look and feel)

Bendri reikalavimai vartotojo sąsajai:

- paprastas (nesudėtingas) panaudojimas;
- prieinamumas, kad vartotojas nesivaržytų naudodamas sistemą;
- neįkyri sąsaja (pavyzdžiui, nereikalaujanti pastoviai ką nors kelis kartus patvirtinti);

3.4.2. Reikalavimai panaudojamumui (angl. usability)

Paprastas panaudojimas be kokio mobiliojo įrenginio vartotojo be specialaus apsimokymo.

3.4.3. Reikalavimai vykdymo charakteristikoms (angl. performance)

Paprastų programos veiksmų vykdymas neturėtų užtrukti ilgiau nei kelios sekundės. Maisto produkto etiketės nuotraukos analizė neturėtų užtrukti daugiau nei 40 sekundžių. Esant geram Interneto ryšiui (512Kb/s), nuotraukos nusiuntimas į serverį neturėtų užtrukti daugiau nei 20 sekundžių.

3.4.4. Reikalavimai veikimo sąlygoms (angl. operational)

Programos veikimui būtinas Interneto ryšys.

3.4.5. Reikalavimai sistemos priežiūrai (angl. maintainability and portability)

Daugeliu atvejų kliento programinė įranga nereikalauja jokios priežiūros. Bet per konfigūracijos failą bus galima keisti serverio programinės įrangos adresą bei serverio atsako laiką (angl. timeout) .

Serverio programinei įrangos priežiūra apsiribos duomenų apie maisto produktų priedus atnaujinimu.

3.5. Projekto išeiga (angl. project issues)

3.5.1. Atviri klausimai (problemos)

Neaišku kaip reikės įgyvendinti nuotraukos siuntimo į serverį progreso stebėjimą, nes standartinėmis .NET Compact Framework 3.5 priemonėmis to stebėti nepavyks. Reikės sugalvoti sprendimą šiai problemai spręsti.

3.5.2. Egzistuojantys sprendimai (angl. off-the-shelf solutions)

Teksto atpažinimui iš nuotraukos gali būti panaudotas Tesseract 2.04 [34] OCR varikliukas. Tai nemokamas atviro kodo (angl. open source) projektas, kuris geriausiai iš nemokamų OCR variklių atpažįsta tekstą. Kadangi pats varikliukas nėra pritaikytas .NET aplinkai, tai norint jį naudoti .NET aplinkoje bus reikalingas kitas atviro kodo produktas Tessnet2 [35].

Nuotraukų apdorojimui mobiliajame įrenginyje gali būti panaudotos OpenNETCF 1.4 [36] programinio kodo bibliotekos, kurios šiek tiek kompensuoja reikiamą funkcionalumo darbui su paveiksliais nebuvimą .NET Compact Framework 3.5 programinio kodo bibliotekose.

Nuotraukų apdorojimui kompiuteryje (serveryje) gali būti panaudotos AForge.NET 2.1 [37] programinio kodo bibliotekos, kurios turi labai naudingų funkcijų, pvz. kreivai nufotografuoto (pasukto kampu) teksto ištiesinimas.

4. PROJEKTO ARCHITEKTŪROS SPECIFIKACIJOS DALIS

4.1. Įvadas

4.1.1. Skyriaus „Architektūros specifikacija“ paskirtis

Architektūros specifikacijos dokumentas aprašo architektūrą kliento ir serverio programinės įrangos, skirtos informacijos pateikimui apie maisto produkte naudojamus priedus iš maisto produkto etiketės nuotraukos. Dokumentą naudos sistemos užsakovas, kad patikrintų ar planuojama architektūra atitinka jo lūkesčius, taip pat sistemos kūrėjai, įrankio programavimui bei testuotojai, kad patikrintų sistemos atitikimą specifikacijai.

4.1.2. Apžvalga

Šis skyrius apima kliento ir serverio programinės įrangos, skirtos informacijos pateikimui apie maisto produkte naudojamus priedus iš maisto produkto etiketės nuotraukos architektūrą. Dokumente pateikiami architektūros tikslai bei apribojimai įrankiui, įrankio panaudojimo atvejai, įrankio statiniai bei dinaminiai vaizdai, duomenų bazės vaizdas bei architektūros įtaka įrankio kokybei.

4.2. Architektūros pateikimas

Architektūra pateikiama per panaudos atvejų, statinį bei dinaminį sistemos vaizdus, taip pat sistemos komponentų išdėstymo vaizdą ir duomenų vaizdą.

Panaudojimo atvejų vaizdą sudaro panaudojimo atvejų diagrama bei jos aprašymas. Statinį sistemos vaizdą sudaro bendra sistemos klasių diagrama, sistemos paketų aprašymas bei klasių diagramos kiekvienam paketui. Dinaminiam sistemos vaizdai aprašyti panaudotos sąveikos, sekų, bendradarbiavimo, veiklos bei būsenų diagramos. Sistemos komponentų išdėstymo vaizdas parodo kaip bus išdėstyti sistemos komponentai. O duomenų vaizdas parodo duomenų bazės struktūrą.

4.3. Architektūros tikslai ir apribojimai

Architektūros projektavimui naudojami keli įrankiai: Microsoft Visio Professional 2003 ir No Magic MagicDraw Enterprise 16.6. Programinė įranga bus kuriama C# kalba, naudojant Microsoft Visual Studio 2008 programinės įrangos kūrimo aplinką (IDE). Duomenų saugojimui bus naudojami XML failai. Programinės įrangos išėties tekstai bus saugojami versijų kontrolės sistemoje Subversion (SVN). Programinė bus kuriama pagal Agile procesą, kuris daugiau vertina:

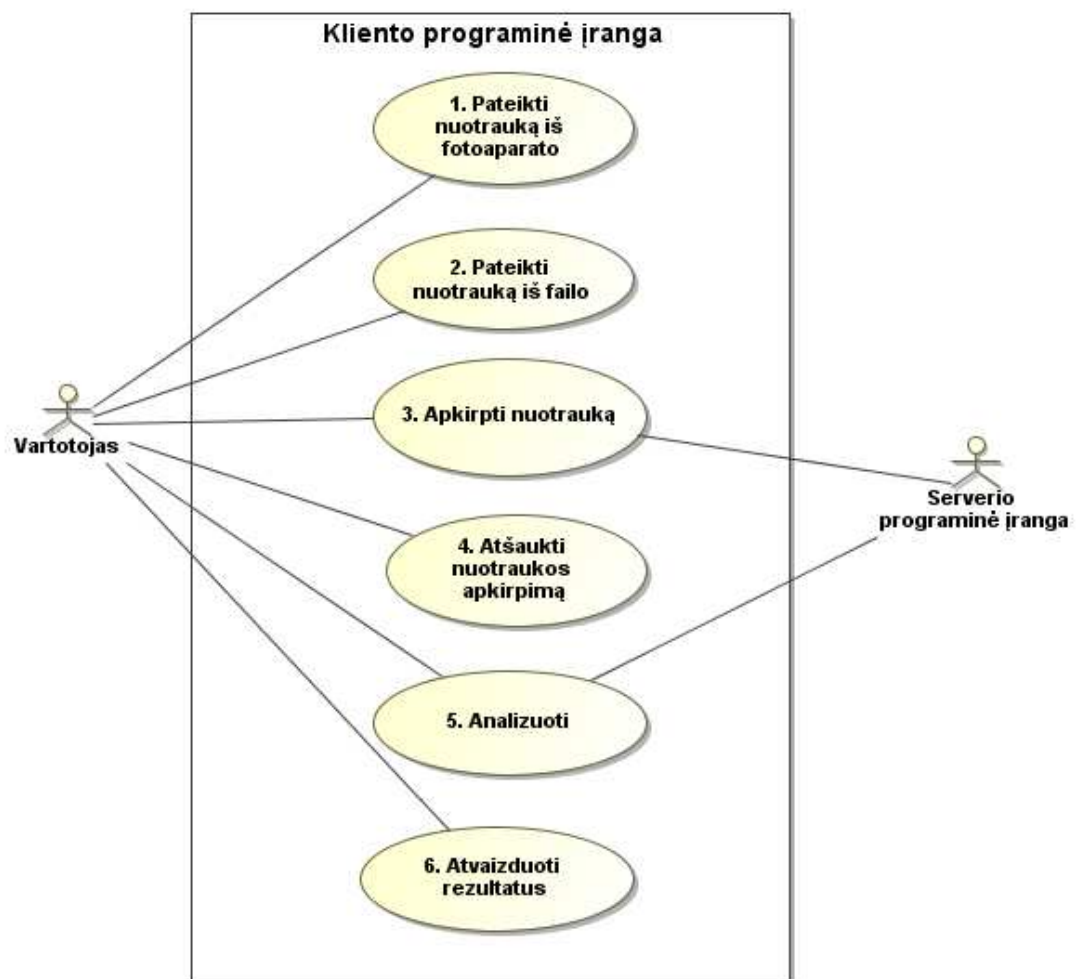
- Individus ir bendravimą nei procesus ir įrankius;
- Veikiančią programinę įrangą nei išsamią dokumentaciją;
- Bendradarbiavimą su klientais nei kontrakto pasirašymą;
- Reakcija į pakeitimus nei plano vykdymą.

4.4. Panaudojimo atvejų vaizdas

Programinė įranga bus naudojama šiais aspektais:

- Turės galimybę gauti ir užkrauti į programą maisto produkto etiketės nuotrauką iš mobiliojo įrenginio fotoaparato;
- Turės galimybę užkrauti maisto produkto etiketės nuotrauką iš mobiliojo įrenginio failų sistemos;
- Programinėje įrangoje užkrautą nuotrauką bus galima apkirpti, kad liktų tiksliai priedų atpažinimui maisto produkto etiketėje reikalingas tekstas.
- Atšaukti nuotraukos apkirpimą, jeigu apkirpimas buvo atliktas neteisingai.
- Analizuos pateiktą maisto produkto etiketės nuotrauką ir pateiks informaciją apie maisto produkto etiketėje rastus priedus;
- Atvaizduos rezultatų sąrašą bei detaliai kiekvieną rezultatą.

Programinės įrangos panaudojimo atvejų vaizdas pateiktas 16 paveikslėlyje.



16 pav. Panaudojimo atvejų diagrama

Programinės įrangos panaudojimo atvejai detalčiau išnagrinėti skyriuje „3. PROJEKTO REIKALAVIMŲ SPECIFIKAVIMO DALIS“.

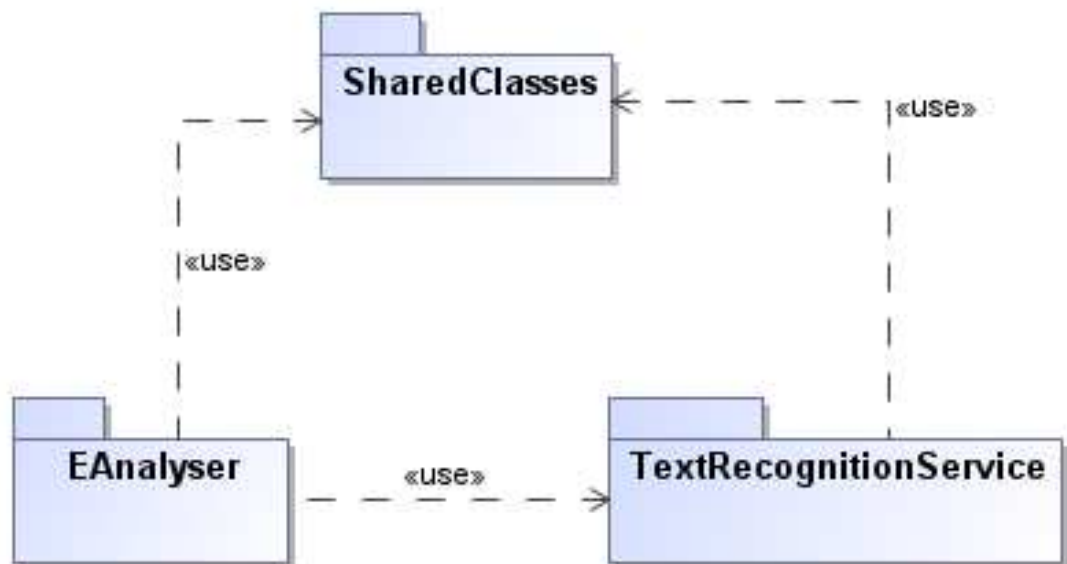
4.5. Sistemos statinis vaizdas

4.5.1. Apžvalga

Pagal skyriuje „3.3. Funkciniai reikalavimai“ pateiktus funkcinis reikalavimus buvo nuspręsta sistemą skaidyti į 3 paketus:

- EAnalyser – paketas atsakingas už kliento programinės įrangos funkcionalumą;
- TextRecognitionService – paketas atsakingas už serverio programinės įrangos funkcionalumą;
- SharedClasses – paketas atsakingas už bendrą kliento ir serverio programinės įrangos funkcionalumą bei bendras struktūras.

Paveikslėlyje 17 pavaizduoti sistemos paketų ryšiai.



17 pav. Sistemos paketų ryšiai

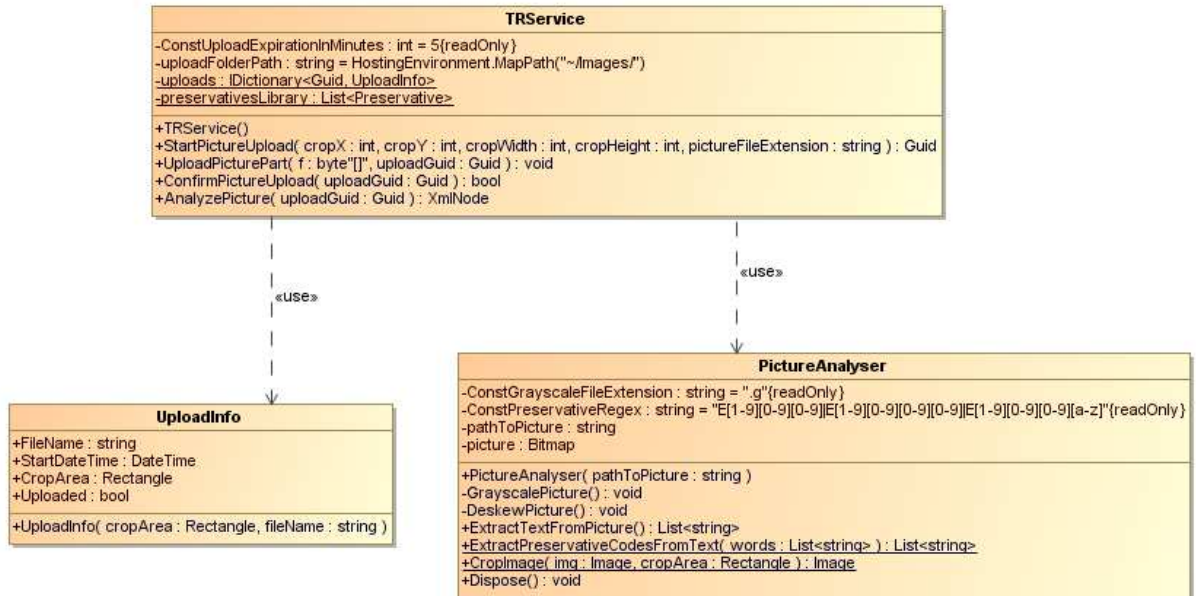
4.5.2. Paketų detalizavimas

4.5.2.1. Paketas TextRecognitionService

Pakete TextRecognitionService esančios klasės reikalingos serverio programinės įrangos funkcionalumui. Šiame pakete bus realizuotos šios klasės:

- TRService – serverio programinės įrangos klasės, skirtos serverio bendravimui su kliento programine įranga web servisų pagalba;
- PictureAnalyser – klasė, skirta darbui su pateikta nuotrauka ir jos analizavimui;
- UploadInfo – tai struktūrinė klasė, sauganti į serverį įkeliamos nuotraukos informaciją.

Paketo TextRecognitionService klasių diagrama pavaizduota 18 paveikslėlyje.



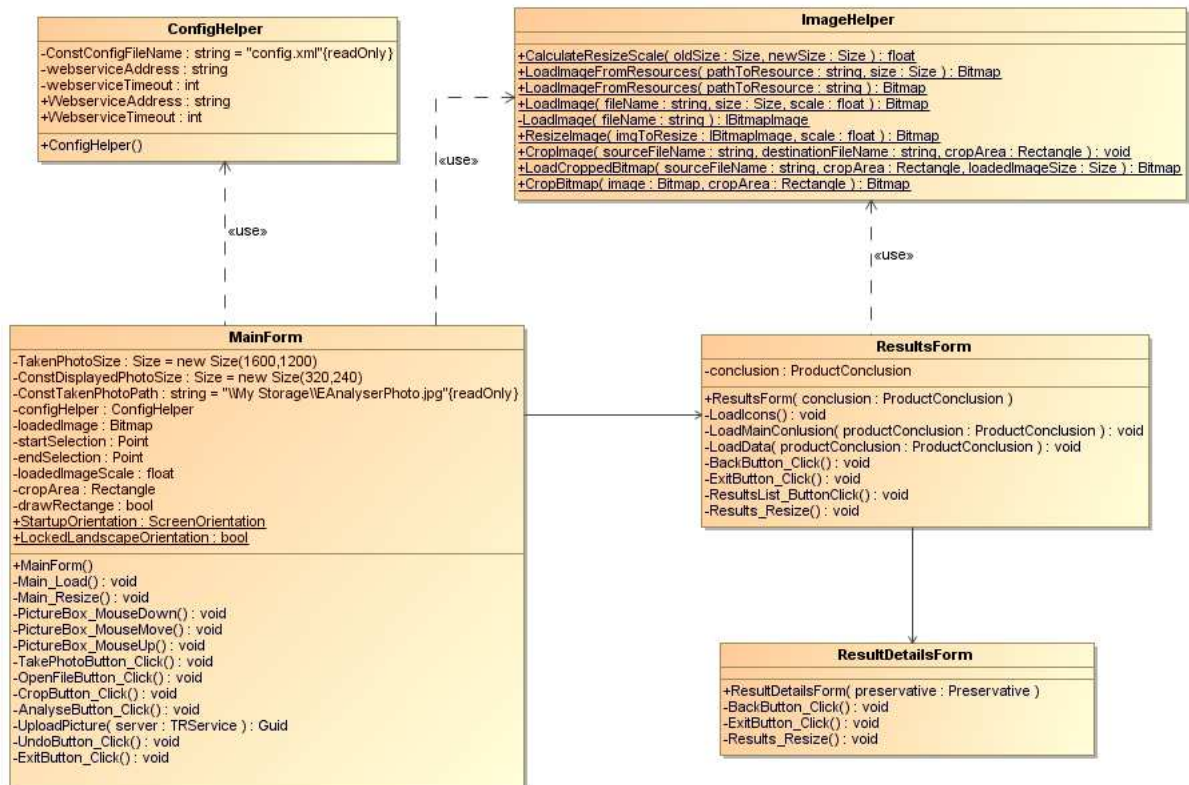
18 pav. Paketo TextRecognitionService klasių diagrama

4.5.2.2. Paketas EAnalyser

Pakete EAnalyser esančios klasės reikalingos kliento programinės įrangos funkcionalumui. Šiame pakete bus realizuotos šios klasės:

- ConfigHelper – pagalbinė klasė, skirta dirbti su kliento programinės įrangos konfigūracijos failu;
- ImageHelper – pagalbinė klasė, skirta dirbti su paveiksliais kliento programinėje įrangoje;
- MainForm – kliento programinės įrangos pagrindinio lango klasė, kurioje bus realizuotas nuotraukos gavimas iš mobiliojo įrenginio fotoaparato arba failo, nuotraukos apkirpimas, nuotraukos apkirpimo atšaukimas bei kreipimasis į serverio programinę įrangą;
- ResultsForm – kliento programinės įrangos rezultatų sąrašo atvaizdavimo klasė;
- ResultDetailsForm – kliento programinės įrangos detalaus rezultato atvaizdavimo klasė.

Paketo EAnalyser klasių diagrama pavaizduota 19 paveikslėlyje.



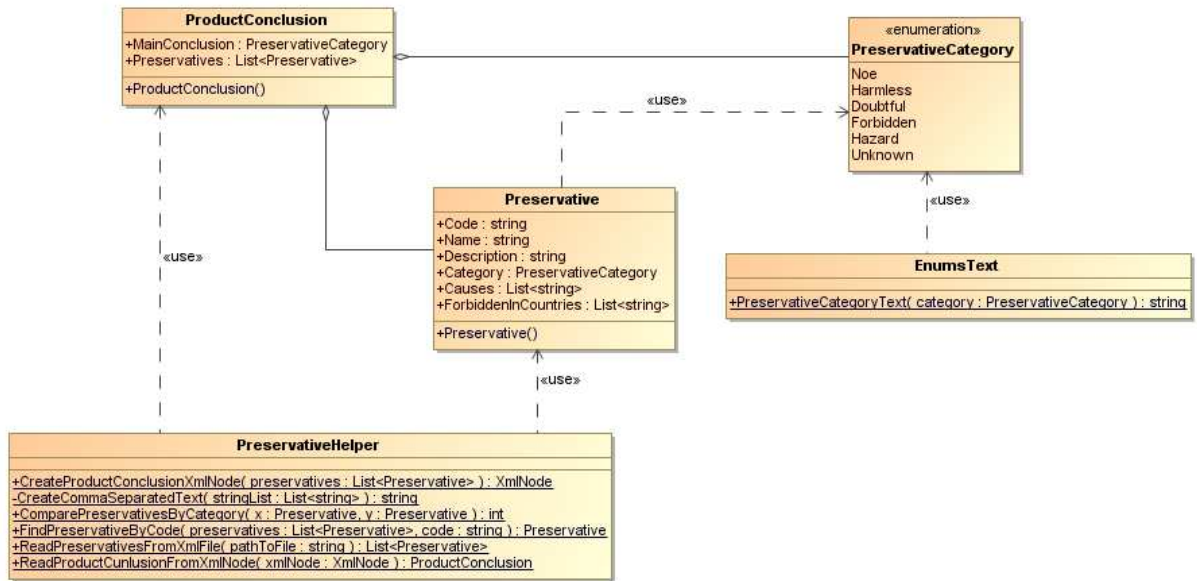
19 pav. Paketo EAnalyser klasių diagrama

4.5.2.3. Paketas SharedClasses

Pakete SharedClasses esančios klasės, tai bendros klasės naudojamos ir serverio programinės įrangos, ir kliento programinės įrangos. Šiame pakete bus realizuotos šios klasės:

- PreservativeHelper – pagalbinė klasė, skirta darbui su priedų informacija;
- ProductConclusion – struktūrinė klasė sauganti informaciją maisto produkte rastus priedus bei bendros produkto išvados informaciją;
- Preservative – struktūrinė klasė sauganti informaciją apie maisto produkto priedą;
- PreservativeCategory – tai sąrašo struktūra (angl. enumeration), skirta maisto produkto priedų kategorijų išvardinimui;
- EnumsText – tai klasė, skirta paversti sąrašo struktūrą (angl. enumeration) į tekstą.

Paketo SharedClasses klasių diagrama pavaizduota 20 paveikslėlyje.

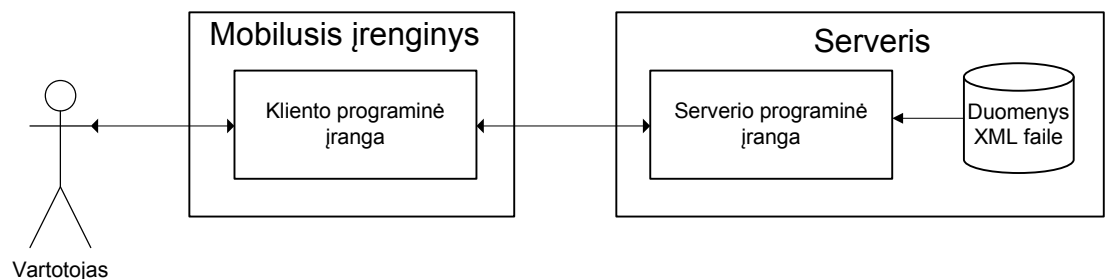


20 pav. Paketo SharedClasses klasių diagrama

4.6. Išdėstymo (angl. deployment) vaizdas

Kuriama sistema turės dvi dalis: kliento programinę įrangą ir serverio programinę įrangą. Kliento programinę įrangą veiks mobiliajame įrenginyje su Windows Mobile operacine sistema, kurios minimalūs reikalavimai yra nurodyti skyriuje „3.2.1. Apribojimams sprendimui“. Serverio programinę įrangą veiks bet kokiam kompiuteryje, kuriame veikia Windows operacinė sistema palaikanti IIS serverio versijas nuo 5.1. Programinę įrangą bus kuriama naudojant Microsoft .NET technologijas. Kadangi programinei įrangai duomenis reikės tikrai skaityti, pasirinktas duomenų saugojimo būdas XML faile.

Sistemos išdėstymo vaizdas pateiktas 21 paveikslėlyje.



21 pav. Sistemos išdėstymo vaizdas

4.7. Duomenų vaizdas

Kaip ir minėta anksčiau, programinė įranga duomenis tikrai skaitys, todėl duomenims saugoti pasirinktas XML failas. XML duomenų struktūroje bus saugoma tokia informacija:

- Priedo kodas;
- Priedo pavadinimas;
- Priedo kategorija. Kategorija priskiriama pagal priedo įtaką žmogui. Projekte naudojamos šios kategorijos:
 - Nekenksmingas – priedo poveikis yra nenustatytas arba nustatytas kaip nekenksmingas;
 - Abejotinas – kai kurių nurodytų šaltinių teigimu, šie priedai gali sukelti negalavimus, tačiau nėra neginčijamų įrodymų, kad tokios pasekmės kyla visais atvejais. Atsargumo priemonių rekomenduojama imtis, jei šiuos priedus vartos maži vaikai (ypatingai iki 3 metų amžiaus), jautrūs ar turintys polinkį į konkrečias ligas suaugusieji;
 - Vengtinas – yra mokslinių tyrimų, įrodančių kad didesnės priedo koncentracijos arba junginiai su kitais priedais gali pakenkti.
 - Uždrausta kai kuriose šalyse – priedas yra uždraustas arba neleidžiamas naudoti vienoje iš šių šalių: JAV, Kanada, Australija ir Naujoji Zelandija.
 - Neatpažintas – jeigu pagal rastą priedo kodą, nebuvo rasta informacija apie priedą.
- Priedo aprašymas;
- Kokius negalavimus gali sukelti priedo naudojimas dideliais kiekiais;
- Jeigu priedo kategorija yra „Uždrausta kai kuriose šalyse“, nurodomos šalys, kuriose priedas yra uždraustas.

Duomenų failo pavyzdys:

```
<preservatives>
  <preservative code="E154" name="Rudasis FK" category="forbidden">
    <description>Sintetinių azo dažų mišinys. Gali sukelti nepakantumą žmonėms,
    netoleruojantiems salicilatų. Be to, yra histamino išlaisvintojas ir gali
    suintensyvinti astmos simptomus. Kartu su benzoatais sukelia vaikų
    hiperaktyvumą. Vienas iš mišinio komponentų gali būti
    mutageniškas.</description>
    <causes>alergija, astma, mutageniškumas</causes>
    <forbidden>JAV, Kanada, Australija, Naujoji Zelandija</forbidden>
  </preservative>
  ...
  </preservative>
  <preservative code="E503" name="Amonio karbonatas, Amonio rūgštusis karbonatas"
  category="harmless">
    <description>Gaminamas iš amonio sulfato ir kalcio karbonato, natūralus
    mineralas.</description>
    <causes>vidurių pūtimas</causes>
    <forbidden></forbidden>
  </preservative>
</preservatives>
```

Panaši XML struktūra bus naudojama ir serverio rezultatų pateikimui kliento programinei įrangai:

```
<preservatives productConclusion="">
  <preservative code="" name="" category="">
    <description></description>
    <causes></causes>
    <forbidden></forbidden>
  </preservative>
  ...
  <preservative code="" name="" category="">
    <description></description>
    <causes></causes>
    <forbidden></forbidden>
  </preservative>
</preservatives>
```

Taigi, rezultatų XML struktūra beveik nesiskiria nuo duomenų XML struktūros, pridėtas tikrai papildomas atributas `productConclusion`, kuris nurodo bendrą produkto išvadą.

5. TIRIAMOJI EKSPERIMENTINĖ DALIS

5.1. Tyrimo metodika

Sukurtai sistemai vertinti buvo pasirinkti šie pagrindiniai vertinimo kriterijai:

1. Paprastumas – tai, ar programinė įranga nėra apkrauta nereikalingomis funkcijomis;
2. Patikimumas – tai programinės įrangos savybė „nenulūžti“ neįprastose situacijose;
3. Intuityvumas – tai, ar vartotojas lengvai perpranta programinės įrangos vartotojo sąsają;
4. Naudojimo patogumas – tai, ar vartotojui patogiu naudotis produktu;
5. Korektiškumas – tai, ar programinė įranga teisingai atlieka savo funkcijas.

Kiekvienas vertinimo kriterijus vertinamas 10 balų skalėje. Bendras sistemos įvertinimas gaunamas sudėjus visų vertinimo kriterijų balus.

5.2. Kokybės vertinimas

Kokybei įvertinti buvo sudaryta lentelė (17 lentelė), kurioje pateikiami tokie stulpeliai:

1. Kriterijus – kokybės vertinimo kriterijaus pavadinimas;
2. Pastabos – pastabos, kuriomis remiantis vertinamas kriterijus;
3. Vertinimas – kriterijaus vertinimas 10 balų skalėje.

Produkto kokybę įvertinome taip:

17 lentelė. Produkto „EAnalyser“ vertinimas

Kriterijus	Pastabos	Vertinimas
Paprastumas	Programoje visos reikiamos funkcijos išdėstytos vienoje vietoje, todėl programa paprasta naudotis. Programoje nėra nereikalingų funkcijų. Beveik visos programos funkcijos gali būti valdomos rankų pirštų pagalba.	9
Patikimumas	Veikia stabiliai.	10
Intuityvumas	Vartotojo sąsają nesunku perprasti.	9
Naudojimo patogumas	Funkcijos, kurias galima ir kurių negalima vykdyti einamuoju momentu yra išskirtos skirtingoms spalvomis, todėl iš karto aišku ką galima daryti, o ko negalima. Iš kiekvieno programos lango yra galimybė grįžti atgal arba išeiti iš programinės įrangos. Jeigu programinė įranga atlieka ilgesnį laiko tarpą truncančius skaičiavimus, tuomet vartotojui rodomi pranešimai, kurie informuoja apie tai kas yra vykdoma. Todėl vartotojui laukti rezultatų nėra nuobodu.	10

Kriterijus	Pastabos	Vertinimas
Korektiškumas	Programa neaptinka maisto priedų, kurie yra aprašyti kitaip nei kodais Pavyzdžiui, sudėtyje paminėtas maisto priedo pavadinimas „mononatrio glutamatas“, o ne maisto priedo kodas E621. Taip pat programa neaptinka ir kai kurių kodais aprašytų maisto priedų.	5
Viso:		43 iš 50

Daugiausiai pastangų reikėjo įdėti vertinant kriterijų „Korektiškumas“. Vertinimui buvo parinktos aštuonių maisto produktų etikečių nuotraukos (22 – 29 pav.).

Wrigley France, S.N.C., 68600 Biesheim, France (Prantsusmaa/Francija/Prancūzija).
Suhkruvaba mūndimaitselfine nārīmiskumm magusainetega. Koostis: magusaine E420, nārīmiskummialus, magusaine ksūlītōol (15 %), niiskusesāilitaja E422, lōhna-ja maitseained, magusained E421, E965, E951; emulgaator soja letsitiin, magusaine E950, antioksidant E320. **Sisaldab fenūūlalaniini allikat. Ūleliigne tarbimine vōib pōhjustada kōhulahtisust.** Toitumisalane teave 100g kohta: energiasaldus 640 kJ/154 kcal, valgud 0g, sūšivesikud 64g (millest suhkrud 0g, poliūoolid 64g, tārklis 0g), rasv 0g (millest kūllastunud rasvhapped 0g), kiudained 0g, naatrium 0g.
Košlājāmā gumija bez cukura ar piparmētru garšu. Ar saldinātājiem. Sastāvs: saldinātājs E420, gumijas bāze, saldinātājs ksilitols (15 %), mitrumuzturētājs E422, aromatizētāji, saldinātāji E421, E965, E951; emulgators sojas lecitīns, saldinātājs E950, antioksidants E320. **Satur fenilalanina avotu. Pārmērīga lietošana var izraisīt diareju.** Uzturvērtība uz 100g: enerģētiskā vērtība 640 kJ/154 kcal, olbaltumvielas 0g, ogļhidrāti 64g (no kuriem cukuri 0g, polioli 64g, ciete 0g), tauki 0g (no kuriem piesātinātās taukskābes 0g), šķiedrvielas 0g, soda 0g.
Becukrē mētū skonio kramtomoji guma su saldikļiais. Sudētis: saldiklis E420, gumos pagrindas, saldiklis ksilitols (15 %), drēgme īšlaikanti medžiaga E422, kvapiosios medžiagos, saldikļiai E421, E965, E951; emulsiklis soju lecitinas, saldiklis E950, antioksidatorius E320. **Turi fenilalanino šaltinī. Vartojimas dideliāis kiekiais gali laisvinti vidurius.** 100g maistingumas : enerģinē vērtē 640 kJ/154 kcal, baltymū 0g, angliavandeniū 64g (īš jū: cukrū 0g, polioliū 64g, krakmolo 0g); riebalū 0g (īš jū: sočiūjū riebalū rūgščiū 0g); skaiduliniū medžiagū 0g, natrio 0g.
1,9g x 14 = 27g

22 pav. Maisto produkto „Orbit Spearmint“ etiketės nuotrauka

WRIGLEY FRANCE S.N.C., 68600 BIESHEIM, FRANCE. (PRANTSUSMAA, FRANCIJA, PRANCŪZIJA).
SUHKRUVABA MŪNDI-JA MENTŪOLIMAITSELFINE NĀRĪMISKUMM MAGUSAINETEGA. KOOSTIS: MAGUSAINĒ E420, NĀRĪMISKUMMIĀLUS, MAGUSAINĒ E967, NIISKUSESĀILITAJA E422, LŌHNA-JA MAITSEAINED, MAGUSAINED E421, E965, E951, E950; HAPPESE REGULAATOR E500, EMULGAATOR SOJA LETSITIIN, ANTIOKSĪDANT E320. **SISALDAB FENŪŪLALANIINI ALLIKAT. ŪLELIIGNE TARBIMINE VŌIB PŌHJUSTADA KŌHULAHTISUST.** TOITUMISALANE TEAVE 100g KOHTA: ENERGIASILDUS 640 kJ/ 154 kcal, VALGUD 0g, SŪSIVESIKUD 64 g (MILLEST SUHKRUD 0g, POLŪŪOLID 64g, TĀRKLIS 0g), RASV 0g (MILLEST KŪLLASTUNUD RASVHAPPED 0g), KIUDAINED 0g, NAATRIUM <0,1g. NĀRI PEALE SŌŌMIST JA JOOMIST. MITMEKŪLGSELT JA TASAKAALUSTATULT TOITUMINE NING TERVISLIK ELUSTIIL ON TĀHTSAD.
KOŠLĀJĀMĀ GUMIJA BEZ CUKURA AR SALDINĀTĀJIEM, AR PIPARMĒTRU UN MENTOLA GARŠU. SASTĀVS: SALDINĀTĀJS E420, GUMIJAS BĀZE, SALDINĀTĀJS E967, MITRUMUZTURĒTĀJS E422, AROMATIZĒTĀJI, SALDINĀTĀJI E421, E965, E951, E950; SKĀBUMA REGULĒTĀJS E500, EMULGATORS SOJAS LECITĪNS, ANTIOKSIDANTS E320. **SATUR FENILALANĪNA AVOTU. PĀRMĒRĪGA LIETOŠANA VAR IZRAISĪT DIAREJU.** UZTURVĒRTĪBA UZ 100g: ENERĢĒTISKĀ VĒRTĪBA 640 kJ/154 kcal, OLBALTUMVIELAS 0g, OGĻHIDRĀTI 64g (NO KURIEM CUKURI 0g, POLIOLI 64g, CIETE 0g), TAUKI 0g (NO KURIEM PIESĀTINĀTĀS TĀUKSKĀBĒS 0g), ŠĶIEDRVIELAS 0g, SODA <0,1g. **KOŠLĀJIET PĒC ĒŠĀNAS UN DZERŠĀNAS. DAUDZVEIDĪGS, SABALANSĒTS UZTURS UN VESELĪGS DŽĪVESVEIDS IR NOZĪMĪGS.**
BEČUKRĒ MĒTŪ IR MENTOLĪO SKONĪO KRĀMTOMOJI GUMA SU SALDIKLĪAIS. SUDĒTIS: SALDIKLIS E420, GUMOS PAGRINDAS, SALDIKLIS E967, DRĒGME ĪŠLAIKANTI MEDŽIAGA E422, KVAPIOSIOS MEDŽIAGOS, SALDIKLĪAI E421, E965, E951, E950; RŪGŠTINGUMA REGULĪŪJANTI MEDŽIAGA E500, EMULSIKLIS SOJŪ LECITINAS, ANTIOKSIDATORIUS E320. **TURI FENILALANĪNO ŠALTIŅ. VARTOJĪMAS DIDELĪAIS KIEKĪAIS GALĪ LAISVINTĪ VIDURIUS.** 100g MAISTĪNGUMAS: ENERĢINĒ VĒRTĒ 640kJ/154 kcal, BALTĪMŪ 0g, ANGLĪAVANDENĪŪ 64g (ĪŠ JŪ: CUKRAUS 0g, POLIOLIŪ 64g, KRĀKMOLO 0g), RĪEBALŪ 0g (ĪŠ JŪ SOČĪJŪ RĪEBALŪ RŪGŠČĪŪ 0g), SKĀIDULINIŪ MEDŽIAGŪ 0g, NATRIO <0,1g. **KRAMTYTI PO VALĢIOAR ĢĒRĪMO. SVARBU ĶVAIRĪ SUBALANSUŪTĀ DIETĀIR SVEIKAS ĢYVENĪMO BŪDAS. 14 TABS – 27g**

23 pav. Maisto produkto „Orbit WHITE“ etiketės nuotrauka

SALDAINIAI

LT/Saldainiai PAUKŠČIŲ PIENAS vanilinio skonio

Sudėtis: cukrus, gliukozės sirupas, trinta kakava, sviestas, augaliniai riebalai, nenugriebtas sutirštintas pienas su cukrumi, kiaušinio baltymo milteliai (putų stabilizatorius E1505), tirštklis agar-agaras, rūgštingumo reguliavimo medžiaga citrinų rūgštis, emulsiklis sojų lecitinas, kvapiosios medžiagos, konservantas kalio sorbatas.

Produkte gali būti žemės riešutų, kitų riešutų pėdsakų.
100 g - 438 kcal (1983 kJ), 3,0 g - baltymų, 25 g - riebalų, 57,3 g - angliavandenių.

LV/ Konfektes PAUKŠČIŲ PIENAS ar vaniļas garšu

Sastāvs: cukurs, gliukozes sirups, rīvēts kakao, sviests, augu eļļa, iebiezināts pilnpiens ar cukuru, olas baltuma pulveris (putu stabilizētājs E1505), biezinātājs agar-agar, skābuma regulētājs citronskābe, emulgators soju lecitīns, aromātvielas, konservants kālija sorbāts. Produkts var saturēt zemes riekstu, citu riekstu daļiņas.

100 g - 438 kcal (1983 kJ), 3,0 g - olbaltumu, 25 g - tauku, 57,3 g - ogļhidrātu.

DE/Konfekt PAUKŠČIŲ PIENAS mit Vanille-Geschmack

Zusammensetzung: Zucker, Glukosesirup, geriebener Kakao, Butter, Pflanzenfette, unentrahmte angedickte Milch mit Zucker, Eiweißpulver (Schaumstabilisator E1505), Verdickungsmittel (Agar-Agar), Säureregulator (Zitronensäure), Emulgator (Sojalecithin), Duftstoffe, Konservierungsstoff (Kaliumsorbat). Das Produkt darf Spuren von Erdnüssen, anderen Nüssen enthalten.

100 g - 438 kcal (1983kJ), 3,0 g - Eiweiß, 25 g - Fett, 57,3 g - Kohlenhydrate.

PL/ PAUKŠČIŲ PIENAS o smaku waniliowym

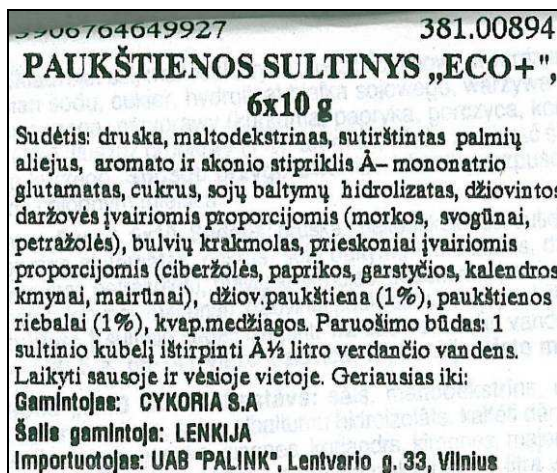
Skład: cukier, syrop glukozowy, miazga kakaowa, masło, tłuszcz roślinny, mleko zagęszczone słodzone, białko jaja w proszku (stabilizator piany: 1505), środek zagęszczający: agar-agar, regulator kwasowości: kwas cytrynowy, emulgator: lecytyna sojowa, substancje zapachowe, substancja konserwująca: sorbinian potasu. Produkt może zawierać nieznaczne ilości arachisu, innych orzechów.

100 g - 438 kcal (1983 kJ). białko: 3,0 g, tłuszcz: 25 g, węglowodany: 57,3 g.

24 pav. Maisto produkto „Paukščių pienas“ etiketės nuotrauka



25 pav. Maisto produkto „DVARO fermentinis sūris“ etiketės nuotrauka



26 pav. Maisto produkto „Paukštienos sultinys ECO+“ etiketės nuotrauka



27 pav. Maisto produkto „Maggi Kostilija“ etiketės nuotrauka

Sudėtis: druska, augaliniai riebalai, skonio ir kvapo stiprikliai (mononatrio glutamatas, dinatrio guanilatas ir inozinatas), krakmolai, vištienos riebalai (3%), kvapiosios medžiagos (tarp jų garstyčių), ciberžolės, džiovintos morkos, cukrus, mielių ekstraktas, petražolių lapai, salierų sėklos, rūgštingumą reguliuojanti medžiaga (citrinos rūgštis), džiovintos vištienos ekstraktas, dažiklis (amoniakinė karamelė).

28 pav. Maisto produkto „Knor Kanapuljong“ etiketės nuotrauka

Sudėtis: druska (49%), maltodekstrinas, kukurūzų krakmolai, aromato ir skonio stiprikliai (mononatrio glutamatas, dinatrio inozinatas, dinatrio guanilatas), gliukozė, augaliniai riebalai, cukrus, kvapiosios medžiagos, ciberžolė, pipirų ekstraktas, vištienos milteliai (0,15%), petražolės. *Produkto sudėtyje gali būti gliukeno, kiaušinių, sojų, salierų ir žuvies pėdsakų.*
Sastāv: sāls (49%), maltodekstrīns, ciete, garšas pastiprinātāji (nātrija glutamāts, dinātrija inozināts, dinātrija guanilāts), gliukoze, augu tauki, cukurs, aromatizētāji, tumeriks, piparu ekstrakts, vīstas gaļas pulveris (0,15%), pētersīji. *Var saturēt glutēnu, olas, soju, selerijas un zivis minimālos daudzumos.*
Koostisosad: keedusool (maksimaalselt 49%), maltodekstriin, tärklis, lõhna- ja maitsetugevdajad (naatriumvesinikglutamaat, dinaatrium-5-inosinaat, dinaatrium-5-guanilaat), glükooos, taimne rasv, suhkur, lõhna- ja maitseained, kurkuma, pipraekstrakt, kanaliha pulber (0,15%), petersell. *Toode võib sisaldada väheses koguses gluteeni, muna, sojat, sellerit ja kala.*

29 pav. Maisto produkto „Maggi Kanapuljong“ etiketės nuotrauka

Kad įvertinti kaip korektiškai yra atpažįstami maisto priedai šiose nuotraukose, buvo sudaryta lentelė (18 lentelė), kurioje pateikiami tokie stulpeliai:

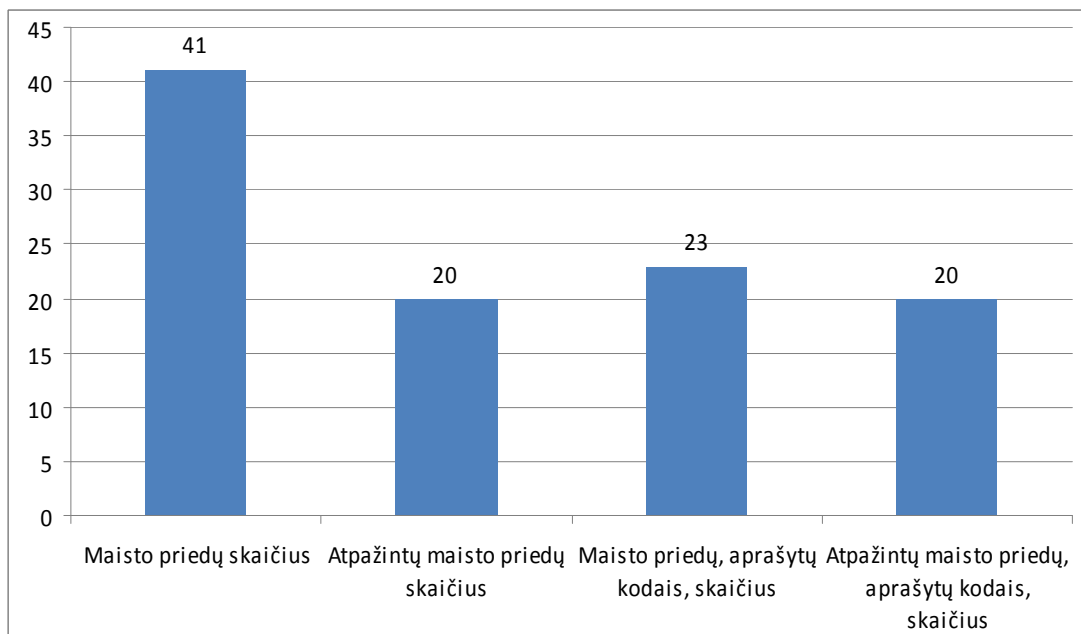
1. Etiketės pav. numeris – nagrinėjamos etiketės paveiksluko numeris;
2. Maisto priedų skaičius – visų maisto priedų esančių etiketėje skaičius;
3. Maisto priedų aprašytų kodais skaičius – etiketėje esantis maisto priedų aprašytų kodais (pvz. E621) skaičius;
4. Atpažintų maisto priedų skaičius – kiek priedų programa atpažino;
5. Atpažintų maisto priedų, aprašytų kodais, skaičius – kiek priedų, aprašytų kodais, programa atpažino.

Kriterijų „Korektiškumas“ vertinome taip:

18 lentelė. Kriterijaus „Korektiškumas“ vertinimas

Etiketės pav. numeris	Maisto priedų skaičius	Maisto priedų, aprašytų kodais, skaičius	Atpažintų maisto priedų skaičius	Atpažintų maisto priedų, aprašytų kodais, skaičius
22	9	7	7	7
23	10	9	7	7
24	5	1	1	1
25	3	3	3	3
26	2	0	0	0
27	4	3	2	2
28	5	0	0	0
29	3	0	0	0
Viso:	41	23	20	20

Dėl aiškumo, iš 18 lentelės suminės eilutės duomenų buvo nubraižyta diagrama:



30 pav. Kriterijaus „Korektiškumas“ vertinimo suminės eilutės duomenų diagrama

Kaip matome, programa veikia nekorektiškai, t.y. visiškai neatpažįsta maisto priedų, kurie aprašyti ne kodais. Taip pat ne visada atpažįstami ir maisto priedų kodai (20 iš 23). Tai įtakoja ir maisto produkto etiketės kokybę, ir etiketės nuotraukos kokybę, ir naudojama teksto atpažinimo sistema). Kadangi teisingai atpažinta pusė maisto priedų (20 iš 41), tai kriterijaus „Korektiškumas“ vertinimas yra 5 iš 10.

Gautas bendras įvertinimas 43 iš 50 yra gana aukštas, tačiau iš visų kriterijų labai išsiskiria kriterijus „Korektiškumas“, kurio vertinimas yra mažas (5 iš 10). Todėl korektiškumą reikia tobulinti.

5.3. Tobulinimo galimybės

Skyriuje „5.2. Kokybės vertinimas“ nustatėme, kad kriterijaus „Korektiškumas“ įvertinimas yra silpnai (5 iš 10), o visų kitų kriterijų įvertinimas yra labai gerai (9-10 iš 10). Todėl programų sistemą reikia tobulinti kriterijaus „Korektiškumas“ atžvilgiu. Priežastys, dėl kurių gaunami nekorektiški rezultatai:

1. Programų sistema suprojektuota atpažinti tikrai maisto priedus, kurie aprašyti kodais (pvz. E621, E160a);
2. Nuotraukos arba maisto produkto etiketės kokybė;
3. Teksto atpažinimo sistemos kokybė.

Buvo sugalvoti du programų sistemos patobulinimai, kurie padėtų panaikinti arba sumažinti aukščiau paminėtų priežasčių poveikį:

1. Padaryti sistemą nepriklausomą nuo teksto atpažinimo sistemos, t.y. sistemą reiktų patobulinti taip, kad joje būtų galima naudoti bet koki optinį teksto atpažinimo variklį, kuris galimai geriau atpažins tekstą;
2. Patobulinti sistemą taip, kad būtų galima universaliai aprašyti taisykles, pagal kurias maisto produkto etiketės tekste būtų ieškoma maisto priedų (požymių).

5.4. Patobulinimų realizavimas

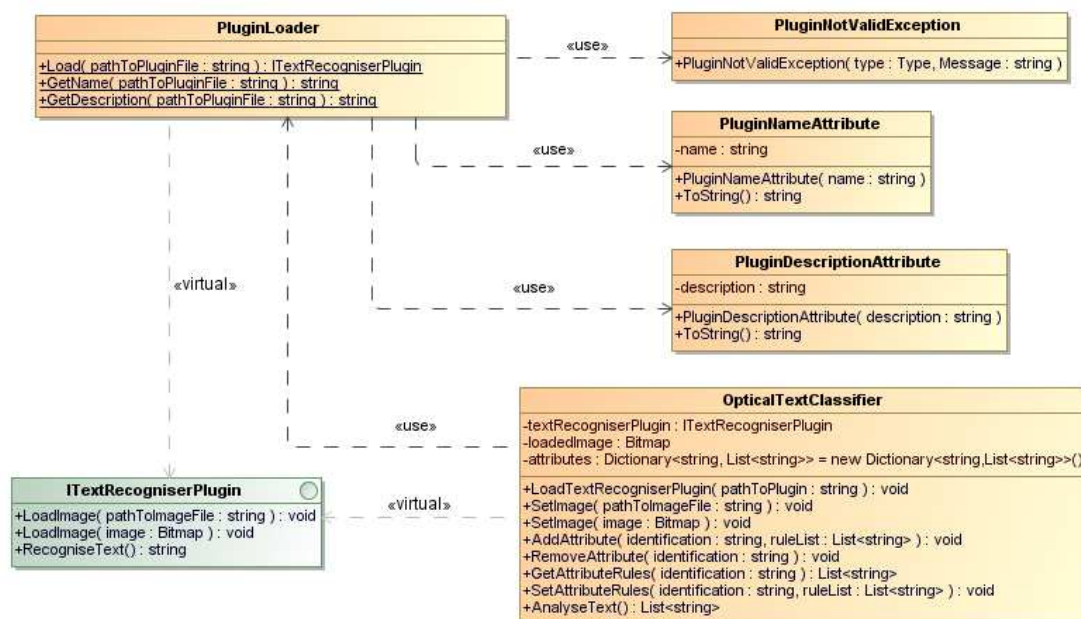
Patobulinimų, aprašytų skyriuje „5.3. Tobulinimo galimybės“, realizavimui buvo sukurtas paketas Fotrac (angl. Flexible optical text recognizer and classifier). Šis paketas atsakingas už du dalykus:

1. Įskiepių (angl. plugins) sąsają (angl. interface), skirtą įvairių optinio teksto atpažinimo variklių naudojimui šiame ir kituose paketuose bei funkcionalumą, skirtą darbui su įskiepiais, t.y. tinkamų naudoti įskiepių identifikavimas bei jų užkrovimas į sistemą.
2. Funkcionalumą, kuris pagal duotus požymius (sudarytus iš reguliariosiomis išraiškomis aprašytų taisyklių) ieško reikiamos informacijos tekste. Tekstas yra gaunamas iš pateikto paveiksliuko naudojant teksto atpažinimo sistemos įskiepi.

Pakete buvo realizuotos šios klasės ir sąsajos:

- `ITextRecogniserPlugin` – sąsaja, skirta optinio teksto atpažinimo sistemų įskiepių kūrimui bei naudojimui.
- `PluginLoader` – klasė, skirta `ITextRecogniserPlugin` sąsajos įskiepių užkrovimui į sistemą.
- `PluginNameAttribute` – atributo, realizuojančio teksto atpažinimo sistemos įskiepi naudojančią sąsają `ITextRecogniserPlugin`, pavadinimo aprašymui. Tuo pačiu šis atributas skirtas ir papildomam įskiepio tipo identifikavimui.
- `PluginDescriptionAttribute` – atributo, realizuojančio teksto atpažinimo sistemos įskiepi naudojančią sąsają `ITextRecogniserPlugin`, aprašui. Tuo pačiu šis atributas skirtas ir papildomam įskiepio tipo identifikavimui.
- `PluginNotValidException` – klaidos aprašymui, jeigu aptinkama, kad duotas įskiepis nėra sąsajos `ITextRecogniserPlugin` įskiepis.
- `OpticalTextClassifier` – klasė, kuri naudoja sąsajos `ITextRecogniserPlugin` įskiepius tam, kad iš duoto paveiksliuko gautų tekstą. Tuomet šiame tekste, pagal duotus požymius (sudarytus iš reguliariosiomis išraiškomis aprašytų taisyklių) ieško informacijos ir grąžina rastų požymių sąrašą.

Paketo Fotrac klasių diagrama pavaizduota 31 paveikslėlyje.

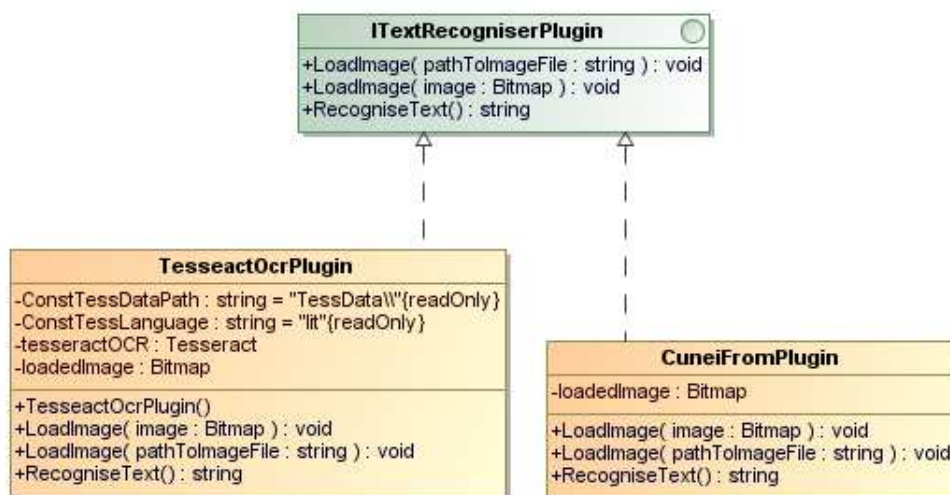


31 pav. Paketo Fotrac klasių diagrama

Taip pat buvo sukurti dviejų optinio teksto atpažinimo sistemų įskiepai:

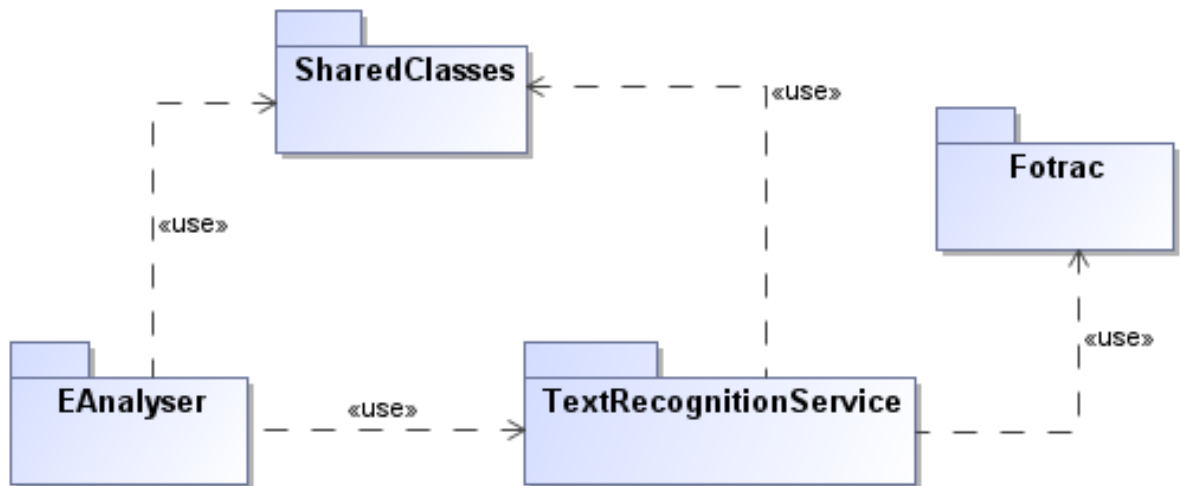
- TesseactOcrPlugin – teksto atpažinimo sistemos Tesseract OCR [15] įskiepis;
- CuneiFormPlugin – teksto atpažinimo sistemos CuneiForm [16] įskiepis.

Įskiepių klasių diagrama pavaizduota 32 paveikslėlyje.



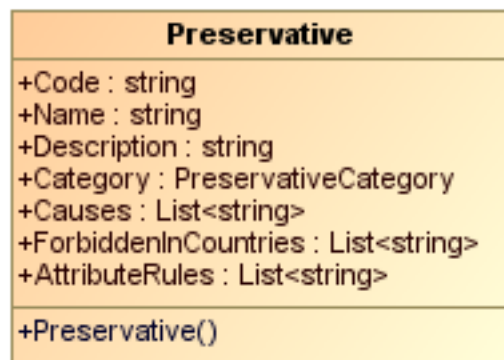
32 pav. Įskiepių klasių diagrama

Naujai sukurtas funkcionalumas buvo integruotas į sukurtą sistemą. Kadangi buvo sukurtas naujas paketas, tai pasikeitė ir sistemos paketų ryšiai, kurie pavaizduoti 33 paveikslėlyje.



33 pav. Sistemos paketų ryšiai po naujo paketo integracijos

Integruojant naują funkcionalumą reikėjo pakeisti paketo SharedClasses (aprašytas skyriuje „4.5.2.3. Paketas SharedClasses“) klasę Preservative, kuri saugo informaciją apie maisto produkto priedą. Buvo pridėta nauja savybė *AttributeRules*, kurioje saugomos maisto priedą apibūdinančio požymio taisyklės, aprašytos reguliariosiomis išraiškomis. Klasė su nauja savybe, atvaizduota 34 paveikslėlyje.



34 pav. Klasė Preservative

Tai pat reikėjo pakeisti kitos paketo SharedClasses klasės PreservativeHelper metodą *ReadPreservativesFromXmlFile*, kuris užkrauna maisto priedų informaciją iš duomenų failo. Šis metodas pakeistas taip, kad iš duomenų failo, kuriame saugoma informacija apie maisto priedus, dar papildomai imtų maisto priedą apibūdinančio požymio taisyklės. Jeigu informacijos apie maisto priedą apibūdinantį požymį nėra, tai užkraunant duomenis sukuriamas požymis, kuris turi taisyklę ieškoti maisto priedo pagal jo kodą. Pavyzdžiui, maisto priedas E621 natrio glutamatas duomenų faile neturi taisyklėmis aprašyti požymio, tai užkraunant duomenis automatiškai sukuriama taisyklė E621, pagal kurią bus ieškoma šio požymio tekste.

Sekantis paketas, kurį reikėjo keisti buvo TextRecognitionService (aprašytas skyriuje „4.5.2.1. Paketas TextRecognitionService“). Teko pakeisti klasės TRService metodą *AnalyzePicture*, taip, kad jis naudotų naujai sukurtą paketo Fotrac klasę OpticalTextClassifier ir vieną iš dviejų įskiepių TesseractOcrPlugin arba CuneiFormPlugin. Klasė PictureAnalyser tapo nereikalinga, nes ją pakeitė lankstesnė klasė OpticalTextClassifier.

Galiausiai buvo keičiama duomenų failo struktūra (aprašyta skyriuje „4.7. Duomenų vaizdas“), kad prie maisto priedo informacijos dar būtų saugomos ir maisto priedą nusakančio požymio taisyklės. Naujo duomenų failo struktūra (nauja dalis paryškinta):

```
<preservatives>
  <preservative code="..." name="" category="...">
    <description>...</description>
    <causes>..., ...</causes>
    <forbidden>..., ...</forbidden>
    <attributeRules>
      <attributeRule>...</attributeRule>
      ...
      <attributeRule>...</attributeRule>
    </attributeRules>
  </preservative>
  ...
  <preservative code="..." name="" category="...">
    <description>...</description>
    <causes>..., ...</causes>
    <forbidden>..., ...</forbidden>
    <attributeRules>
      <attributeRule>...</attributeRule>
      ...
      <attributeRule>...</attributeRule>
    </attributeRules>
  </preservative>
</preservatives>
```

Be abejo, buvo papildyti ir maisto priedų informacijos duomenys, pridėdant maisto priedus nusakančius požymius.

5.5. Eksperimentas

Eksperimentas buvo atliekamas siekiant išsiaiškinti kiek pagerėjo sistemos kokybės kriterijaus „Korektiškumas“ vertinimas atlikus pakeitimus sistemoje (skyrius „5.4. Patobulinimų realizavimas“). Eksperimente vertinimas buvo atliekamas taip pat kaip ir prieš tai vertinant kriterijų „Korektiškumas“ (skyrius „5.2. Kokybės vertinimas“).

Pradžioje buvo atliktas vertinimas naudojant teksto atpažinimo sistemos Tesseract OCR įskiepi TesseractOcrPlugin ir kiekvieno maisto priedo atpažinimo požymį aprašant viena taisykle – maisto priedas tekste ieškomas pagal jo kodą. Pvz. E621 natrio glutamato taisyklė: „E621“. Kaip ir reikėjo tikėtis, eksperimento rezultatai (19 lentelė) visiškai sutapo su rezultatais (18 lentelė) prieš programinės įrangos patobulinimą, nes buvo naudojamos tos pačios taisyklės ir teksto atpažinimo variklis kaip prieš patobulinimus.

19 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant TesseractOcrPlugin įskiepi ir vieną požymio atpažinimo taisyklę

Etiketės pav. numeris	Maisto priedų skaičius	Maisto priedų, aprašytų kodais, skaičius	Atpažintų maisto priedų skaičius	Atpažintų maisto priedų, aprašytų kodais, skaičius
22	9	7	7	7
23	10	9	7	7
24	5	1	1	1
25	3	3	3	3
26	2	0	0	0
27	4	3	2	2
28	5	0	0	0
29	3	0	0	0
Viso:	41	23	20	20

Tuomet buvo patikrintas kitos teksto atpažinimo sistemos CuneiForm įskiepis CuneiFormPlugin, naudojant tą pačią vieną atpažinimo požymio taisyklę. Buvo atpažinti visi 23 maisto priedai, kurie buvo aprašyti kodais (20 lentelė). Tai reiškia, kad šis teksto atpažinimo variklis geriau susidoroja su teksto atpažinimu (Tesseract OCR atpažino 20 iš 23).

20 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant CuneiFormPlugin įskiepi ir vieną požymio atpažinimo taisyklę

Etiketės pav. numeris	Maisto priedų skaičius	Maisto priedų, aprašytų kodais, skaičius	Atpažintų maisto priedų skaičius	Atpažintų maisto priedų, aprašytų kodais, skaičius
22	9	7	7	7
23	10	9	9	9
24	5	1	1	1
25	3	3	3	3
26	2	0	0	0
27	4	3	3	3
28	5	0	0	0
29	3	0	0	0
Viso:	41	23	23	23

Sekančiame etape vėl buvo naudojamas įskiepis TeseactOcrPlugin, tiktai dabar kiekvienas maisto priedo atpažinimo požymis aprašomas dvejomis taisyklėmis:

1. Maisto priedas tekste ieškomas pagal jo kodą;
2. Maisto priedas tekste ieškomas pagal jo pavadinimą.

Pvz. E621 natrio glutamato taisyklės: „E621“ ir „natrio glutamatas“. Kadangi dabar požymiai atpažįstami lanksčiau, tai sistema pradėjo atpažinti ir maisto priedus aprašytus pavadinimais (21 lentelė), tačiau dėl netinkamai atpažinto teksto buvo atpažinti ne visi požymiai (32 iš 41).

21 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant TeseactOcrPlugin įskiepi ir dvi požymio atpažinimo taisykles

Etiketės pav. numeris	Maisto priedų skaičius	Maisto priedų, aprašytų kodais, skaičius	Atpažintų maisto priedų skaičius	Atpažintų maisto priedų, aprašytų kodais, skaičius
22	9	7	9	7
23	10	9	7	7
24	5	1	5	1
25	3	3	3	3
26	2	0	1	0
27	4	3	2	2
28	5	0	2	0
29	3	0	3	0
Viso:	41	23	32	20

Galiausiai su dvejomis požymio atpažinimo taisyklėmis buvo patikrintas įskiepis CuneiFormPlugin. Buvo gautas geriausias rezultatas (22 lentelė), atpažinti 36 maisto priedai iš 41.

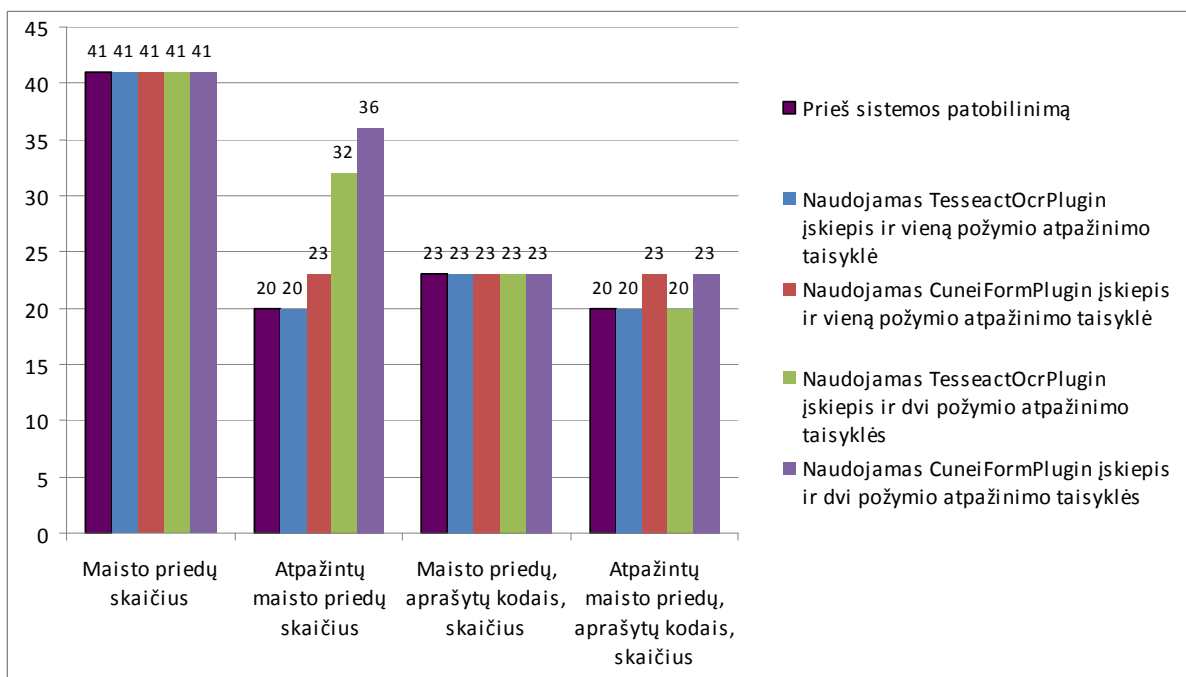
22 lentelė. Kriterijaus „Korektiškumas“ vertinimas naudojant CuneiFormPlugin įskiepi ir dvi požymio atpažinimo taisykles

Etiketės pav. numeris	Maisto priedų skaičius	Maisto priedų, aprašytų kodais, skaičius	Atpažintų maisto priedų skaičius	Atpažintų maisto priedų, aprašytų kodais, skaičius
22	9	7	8	7
23	10	9	10	9
24	5	1	5	1
25	3	3	3	3
26	2	0	2	0
27	4	3	3	3
28	5	0	2	0
29	3	0	3	0
Viso:	41	23	36	23

Atidžiau panagrinėjus kodėl naudojant abu teksto atpažinimo variklius buvo aptikti ne visi maisto priedai, paaiškėjo, kad naudojant Tesseract OCR teksto atpažinimo variklį 8 maisto priedai neaptikti, dėl netinkamai atpažinto teksto ir 1 dėl nestandartinio maisto priedo užrašymo. CuneiForm teksto atpažinimo variklio rezultatai geresni, bet panašūs: 4 maisto priedai neaptikti dėl netinkamai atpažinto teksto ir 1 dėl nestandartinio maisto priedo užrašymo.

Atvejį, kai maisto priedas buvo užrašytas nestandartiškai galima išspręsti šiam priedui pakoregavus taisyklę. Nestandartiškai užrašytas tekstas: „dinatrio guanilatas ir inozinatas“. Standartiškai užrašytas tekstas: „dinatrio guanilatas ir dinatrio inozinatas“. Kad būtų atpažintas maisto priedas dinatrio inozinatas, reikia taisyklę „dinatrio inozinatas“ pakeisti į „dinatrio.* inozinatas“.

Apibendrinant visus eksperimento rezultatus buvo nubraižyta 18 - 22 lentelių suminių eilučių duomenų diagrama:



35 pav. Eksperimento rezultatų diagrama

Iš diagramos aiškiai matome, kad naudojant dvi požymio atpažinimo taisykles ir naują teksto atpažinimo variklį kriterijaus korektiškumas vertinimas smarkiai padidėjo, atpažinti 36 iš 41 maisto priedo (88%), lyginant su rezultatais prieš sistemos patobulinimą 20 iš 41 (49%). Taigi, pagerėjo ir bendra sistemos kokybė.

6. IŠVADOS

- Buvo atlikta dokumentų klasifikavimo ir susijusių technologijų analizė. Nuspręsta naudoti teksto šablonų sulyginimo (angl. string pattern matching) technologiją – reguliariąsias išraiškas (angl. regular expressions). Reguliariosios išraiškos pasirinktos todėl, kad jomis galima lanksčiai aprašyti tekste esančius požymius, o analizuojamos tekstinės informacijos kiekis nėra didelis, todėl teksto analizės greitis nėra svarbus.
- Išanalizavus projektui keliamus reikalavimus, buvo pasirinkta serverio ir kliento architektūra, ir naudojama Microsoft .NET technologija. Serverio dalyje atliekami sudėtingiausi skaičiavimai: optinis teksto atpažinimas ir teksto analizė, o kliento dalis skirta duomenų pateikimui bei rezultatų atvaizdavimui. Kliento dalis veikia Windows Mobile aplinkoje.
- Projekto programinės įrangos kokybės analizė parodė, kad programinė įranga atpažįsta tikrai dalį pateikiamų duomenų (maisto produkto sudėties etiketės nuotraukoje esančių maisto priedų). Atpažįstamų duomenų kiekis su duotais testiniais duomenimis buvo 49%. Todėl buvo pasiūlyti ir įgyvendinti sistemos patobulinimai. Sukurtas atskirtas komponentas, kurio pagalba naudojant įskiepių (angl. plugins) technologiją galima naudoti įvairius optinio teksto atpažinimo variklius bei lanksčiai analizuoti atpažintą tekstą. Šis komponentas buvo integruotas į projekto programinę įrangą.
- Buvo atliktas sistemos patobulinimų analizės eksperimentas ir ištirta, kad naudojant geresnį optinio teksto atpažinimo variklį bei teisingai aprašius tekste ieškomų duomenų (maisto priedų) požymius, atpažįstamų duomenų kiekis su duotais testiniais duomenimis padidėjo iki 88%. Tai yra labai geras rezultatas. Ateityje jį būtų galima dar pagerinti panaudojus tobulesnį optinio teksto atpažinimo variklį.

7. LITERATŪROS ŠARAŠAS

1. **C. H. A. Koster**, IR2 – full-text information retrieval, Lecture Notes, p. 13-16, <http://www.cs.ru.nl/~kees/ir2/papers/h03.pdf> [žiūrėta 2010-04-25].
2. **R. S. Boyer, J. S. Moore**, A fast string searching algorithm, Comm. ACM 20: 762–772 [publikuota 1977].
3. **Algimantas Juozapavičius**, Duomenų struktūros ir algoritmai, Vilniaus universiteto leidykla, p. 69-70 [publikuota 1997].
http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D_20090618_182844-59741/DS.005.0.01.ETD [žiūrėta 2010-05-10].
4. **Ilma Ivanauskaitė**, Mokslinio žurnalo tinklalapio kūrimas, baigiamasis magistro darbas, http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D_20090618_182844-59741/DS.005.0.01.ETD [žiūrėta 2010-05-10].
5. **S. C. Kleene**, Representation of Events in Nerve Nets and Finite Automata, Princeton University Press, p. 3–42 [publikuota 1956].
6. **Fung, Glenn**, A Comprehensive Overview of Basic Clustering Algorithms, p. 4, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.5037&rep=rep1&type=pdf>, [žiūrėta 2010-05-11].
7. **Nethra Sambamoorthi**, Hierarchical Cluster Analysis. Some Basics and Algorithms, p. 4-8, http://www.crmportals.com/hierarchical_cluster_analysis.pdf, [žiūrėta 2010-05-11].
8. **Paul Thacker**, Cluster Analysis, <http://csurs1.csr.uky.edu/~pthacker/cluster.html>, [žiūrėta 2010-05-11].
9. **Ed Greengrass**, Information Retrieval: A Survey, <http://www.cs.umbc.edu/research/cadip/readings/IR.report.120600.book.pdf>, p. 5-8, [publikuota 2000-11-30].
10. **Amit Singhal**, Modern Information Retrieval: A Brief Overview, <http://singhal.info/ieee2001.pdf>, [žiūrėta 2010-05-13].
11. Ižanga į vektorių erdvės modelį, <http://eivind.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html>, [žiūrėta 2010-05-13].
12. Informacija apie išvadų tinklo modelį, <http://www.eecs.qmul.ac.uk/~andrew/pubs/sigir02/html/node5.htm>, [žiūrėta 2010-05-13].
13. Optinio simbolių atpažinimo variklis ABBYY FineReader Engine, http://www.abbyy.com/ocr_sdk_windows/, [žiūrėta 2009-08-03].
14. Optinio simbolių atpažinimo variklis GOCR, <http://jocr.sourceforge.net>, [žiūrėta 2009-08-03].
15. Optinio simbolių atpažinimo variklis Tesseract OCR, <http://code.google.com/p/tesseract-ocr/>, [žiūrėta 2009-08-04].
16. Optinio simbolių atpažinimo variklis CuneiForm, <http://www.cuneiform.ru/eng/index.html>, [žiūrėta 2009-08-04].
17. Sveikas vaikas – maisto priedų žinynas, <http://www.sveikasvaikas.lt>, [žiūrėta 2009-08-12].
18. Programinis produktas scanR, <http://scanr.com>, [žiūrėta 2009-02-04].
19. Informacija apie PDF formatą, http://en.wikipedia.org/wiki/Portable_Document_Format, [žiūrėta 2009-09-02].
20. Programinis produktas Pocket Food Additives, <http://www.pocketpcsoft.com>, [žiūrėta 2009-09-17].
21. Programinis produktas SveikasVaikas mobile, <http://www.telesoftas.com/sveikas>, [žiūrėta 2009-09-03].
22. Informacija apie J2ME platformą, http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition,

- [žiūrėta 2009-09-05].
23. Informacija apie Windows Mobile platformą, <http://www.microsoft.com/windowsmobile>, [žiūrėta 2009-09-10].
 24. Volere šablonai, <http://www.volere.co.uk>, [žiūrėta 2009-09-10].
 25. Komisijos „Codex Alimentarius“ svetainė, <http://www.codexalimentarius.net>, [žiūrėta 2009-09-15].
 26. Lankstinukas „Kaip teisingai perskaityti etiketę?“, http://ec.europa.eu/food/food/labellingnutrition/foodlabelling/publications/leafletLT3_12112007_web.pdf, [žiūrėta 2009-09-15].
 27. **R. Masiliūnas**, E maisto priedai, <http://www.olimpiados.lt/content/view/1001/1>, [publikuota 2009-03-31].
 28. Programinis produktas Food Additives, <http://www.finitysoft.com>, [žiūrėta 2009-09-17].
 29. Ekologiško gaminio logotipo konkursas, http://ec.europa.eu/agriculture/organic/logo/index_lt.htm, [žiūrėta 2009-09-18].
 30. Viešosios įstaigos „Ekoagros“ svetainė, <http://www.ekoagros.lt>, [žiūrėta 2009-09-19].
 31. Viešosios įstaigos „Sveiko vaiko institutas“ svetainė, skirta maisto produktui suteikti sveiko maisto logotipą, <http://istirtas.lt>, [žiūrėta 2009-09-20].
 32. Straipsnis „Maisto produktų ženklavimas“, <http://www.ecc.lt/zinokteises/?LT=maisto-produktu-zenklinimas>, [žiūrėta 2009-09-20].
 33. Straipsnis „Esame tai, ką valgome, net jei nežinome ką!“, http://ec.europa.eu/news/agriculture/080207_1_lt.htm, [žiūrėta 2009-09-20].
 34. Atviro kodo programinis produktas Tesseract, <http://code.google.com/p/tesseract-ocr>, [žiūrėta 2009-11-05].
 35. Atviro kodo programinis produktas Tessnet2, <http://www.pixel-technology.com/freeware/tessnet2>, [žiūrėta 2009-11-05].
 36. Nemokamos programinio kodo bibliotekos OpenNETCF, <http://www.opennetcf.com>, [žiūrėta 2009-11-05].
 37. Nemokamos programinio kodo bibliotekos AForge.NET, <http://code.google.com/p/aforge>, [žiūrėta 2009-11-05].

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

Terminas / Santrumpa	Aprašymas
Skaitmenizavimas	Informacijos pavertimas į skaitmeninį formatą.
PDF	Atviras failo formatas, skirtas technologiškai neutraliam dvimačiam dokumentui atvaizduoti (angl. Portable Document Format)
Klasterizavimas	Grupavimas
J2ME	Java platforma, skirta mobiliesiems įrenginiams (angl. Java Micro Editon)
ES	Europos sąjunga
GM	Genetiškai modifikuotas
GMO	Genetiškai modifikuotas organizmas
NMVRVI	Nacionalinis maisto ir veterinarijos rizikos vertinimo institutas
OCR	Optinis simbolių atpažinimas (angl. optical character recognition)
WQVGA	Ekrano rezoliucijos pavadinimas reiškiantis, kad ekrano rezoliucija yra 240 x 400 taškų.
MB	Megabaitas (angl. megabyte)
Kb	Kilobitas (angl. kilobit)
MP	Megapiksėlis (angl. megapixel)
IIS	Interneto serveris veikiantis Windows aplinkoje (angl. Internet Information Services)
.NET	Microsoft sukurta sistema (angl. framework), skirta programinės įrangos kūrimui ir naudojimui Windows aplinkoje.
ASP.NET	Microsoft sukurta sistema (angl. Framework), skirta Interneto programinės įrangos kūrimui bei naudojimui IIS serveriuose.
XML	Duomenų struktūrų bei jų turinio aprašomoji kalba (angl. eXtensible Markup Language)
IDE	Integruota kūrimo aplinka (angl. integrated development environment)
SVN	Versijų kontrolės sistema „Subversion“.
FOTRC	Lankstaus optinio teksto atpažinimo ir klasifikavimo (angl. Flexible optical text recognizer and classifier) programinio kodo biblioteka.

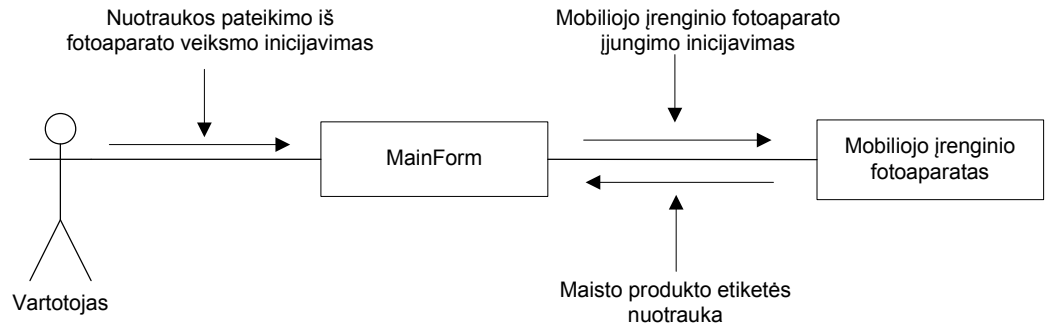
9. PRIEDAI

9.1. PRIEDAS 1: Sistemos dinaminis vaizdas

9.1.1. Sąveikos diagramos

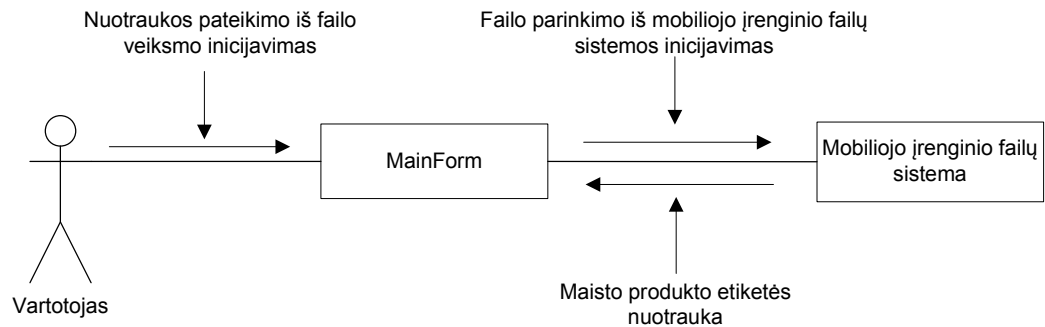
Sąveikos diagramos pateiktos 1– 6 paveikslėliuose.

9.1.1.1. Sąveikos diagrama „Pateikti nuotrauką iš fotoaparato“



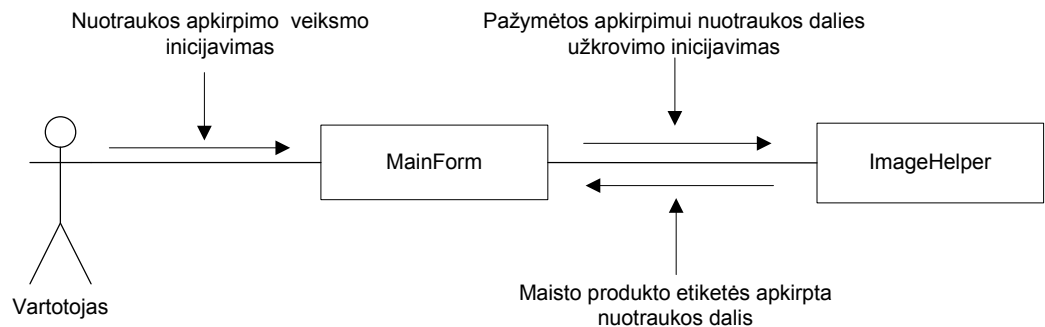
1 pav. Sąveikos diagrama „Pateikti nuotrauką iš fotoaparato“

9.1.1.2. Sąveikos diagrama „Pateikti nuotrauką iš failo“



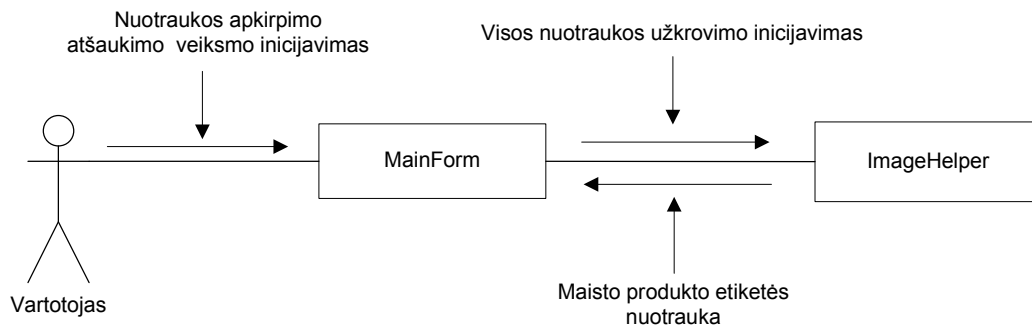
2 pav. Sąveikos diagrama „Pateikti nuotrauką iš failo“

9.1.1.3. Sąveikos diagrama „Apkirpti nuotrauką“



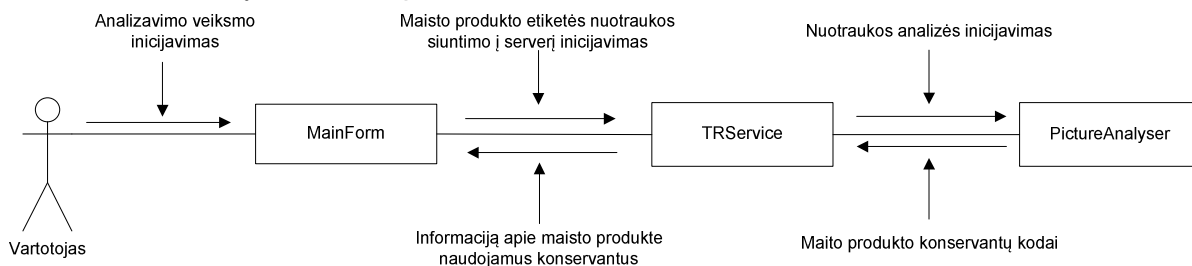
3 pav. Sąveikos diagrama „Apkirpti nuotrauką“

9.1.1.4. Sąveikos diagrama „Atšaukti nuotraukos apkirpimą“



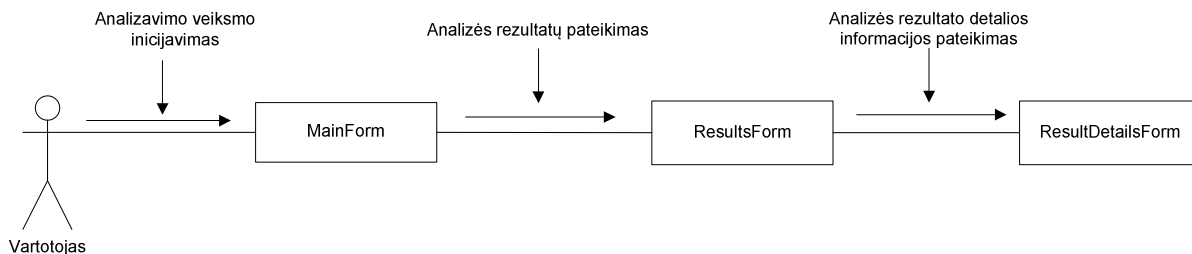
4 pav. Sąveikos diagrama „Atšaukti nuotraukos apkirpimą“

9.1.1.5. Sąveikos diagrama „Analizuoti“



5 pav. Sąveikos diagrama „Analizuoti“

9.1.1.6. Sąveikos diagrama „Atvaizduoti rezultatus“

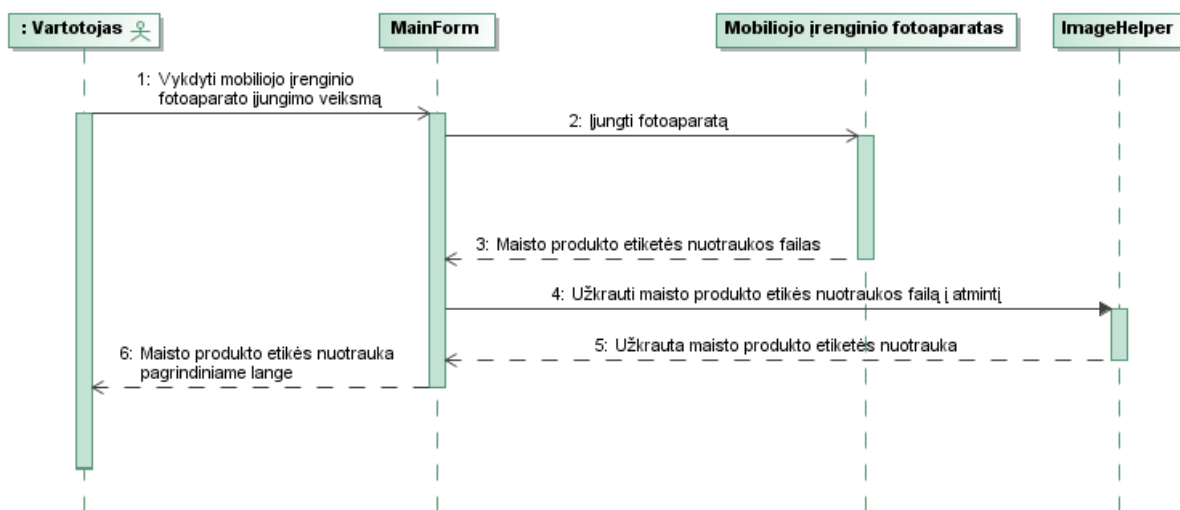


6 pav. Sąveikos diagrama „Atvaizduoti rezultatus“

9.1.2. Sekų diagramos

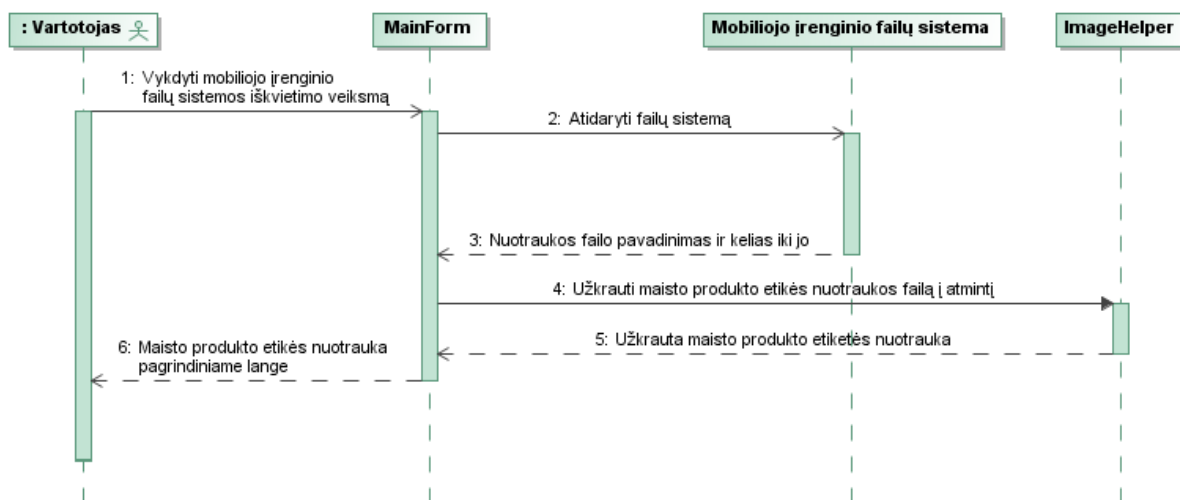
Sekų diagramos pateiktos 7 – 12 paveikslėliuose.

9.1.2.1. Sekų diagrama „Pateikti nuotrauką iš fotoaparato“



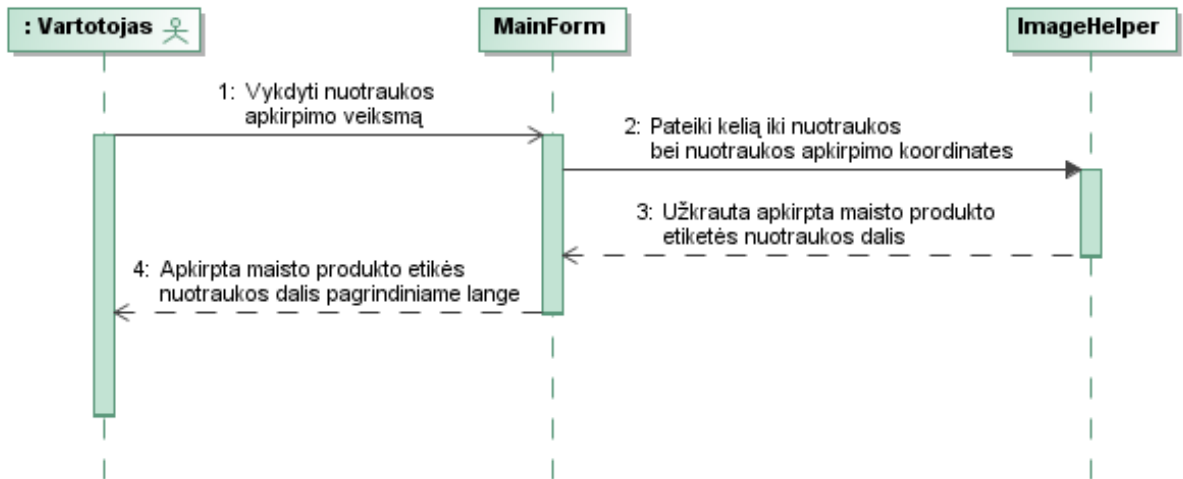
7 pav. Sekų diagrama „Pateikti nuotrauką iš fotoaparato“

9.1.2.2. Sekų diagrama „Pateikti nuotrauką iš failo“



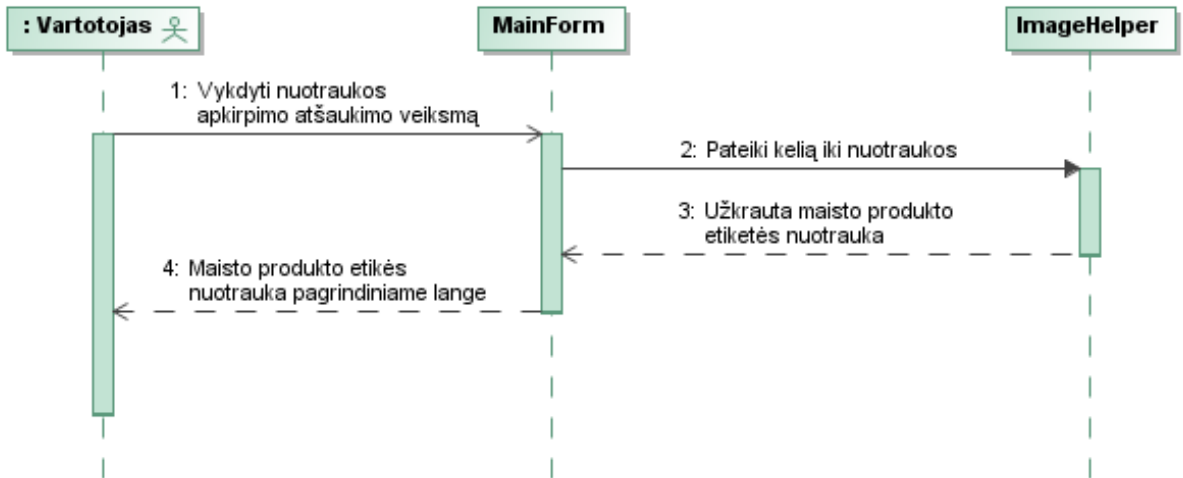
8 pav. Sekų diagrama „Pateikti nuotrauką iš failo“

9.1.2.3. Sekų diagrama „Apkirpti nuotrauką“



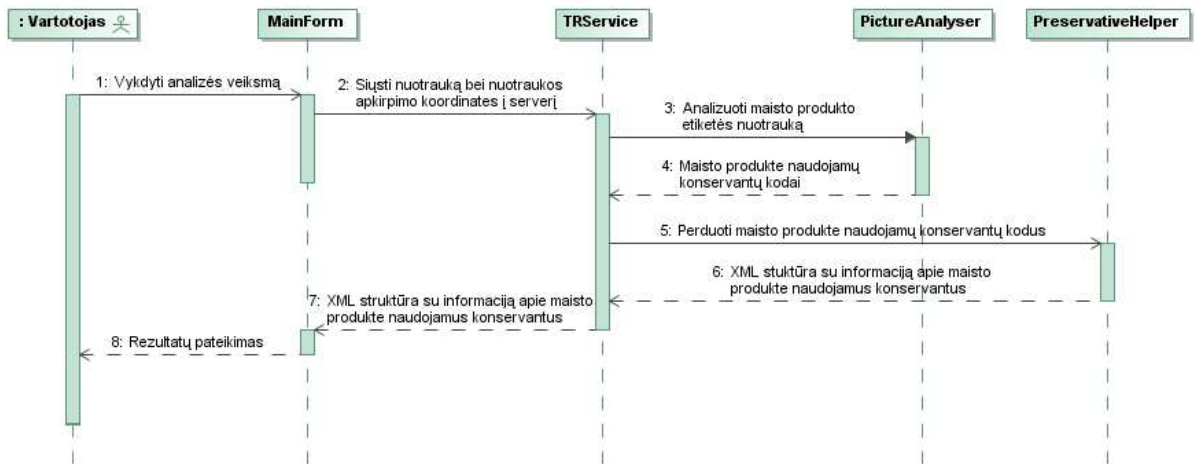
9 pav. Sekų diagrama „Apkirpti nuotrauką“

9.1.2.4. Sekų diagrama „Atšaukti nuotraukos apkirpimą“



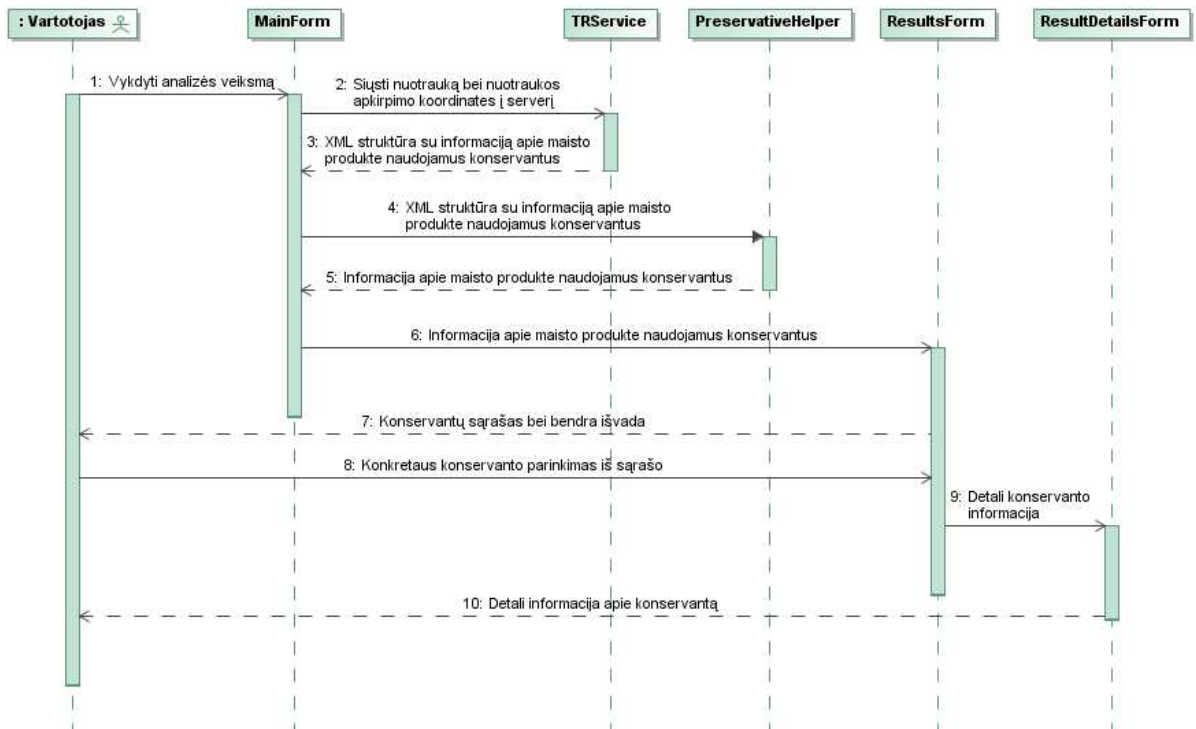
10 pav. Sekų diagrama „Atšaukti nuotraukos apkirpimą“

9.1.2.5. Sekų diagrama „Analizuoti“



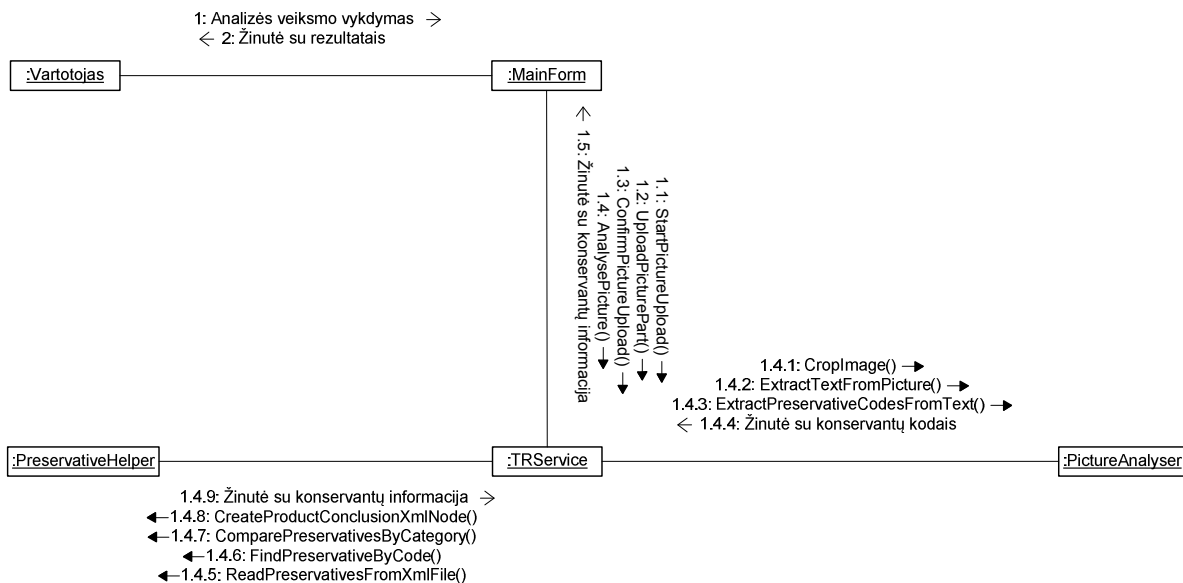
11 pav. Sekų diagrama „Analizuoti“

9.1.2.6. Sekų diagrama „Atvaizduoti rezultatus“



12 pav. Sekų diagrama „Atvaizduoti rezultatus“

9.1.3. Bendradarbiavimo diagrama „Analizuoti“

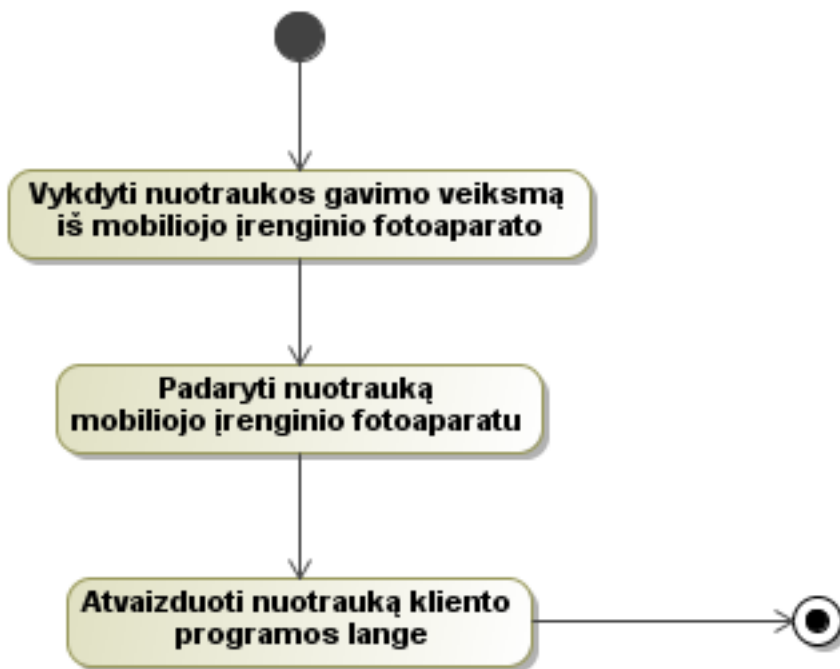


13 pav. Bendradarbiavimo diagrama „Analizuoti“

9.1.4. Veiklos diagramos

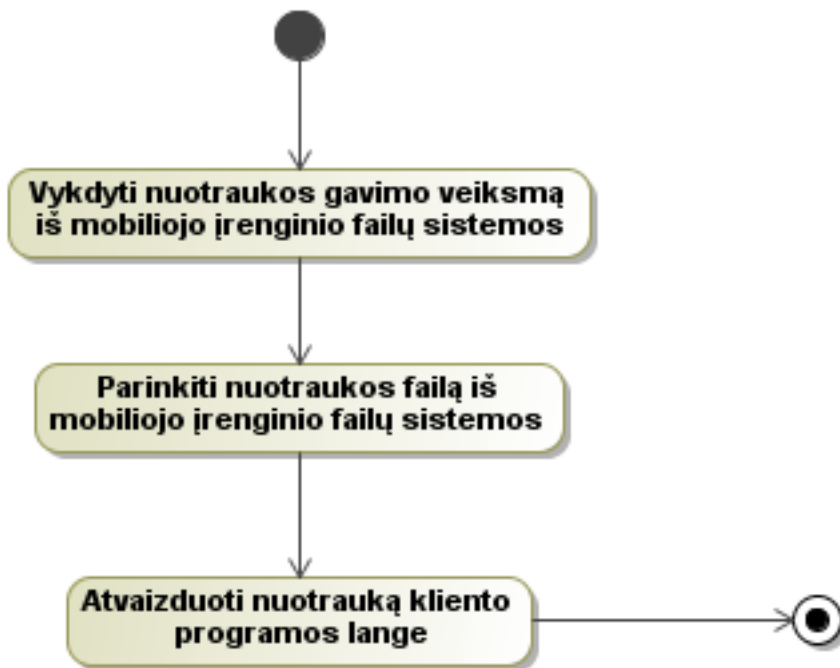
Veiklos diagramos pateiktos 14 – 19 paveikslėliuose.

9.1.4.1. Veiklos diagrama „Pateikti nuotrauką iš fotoaparato“



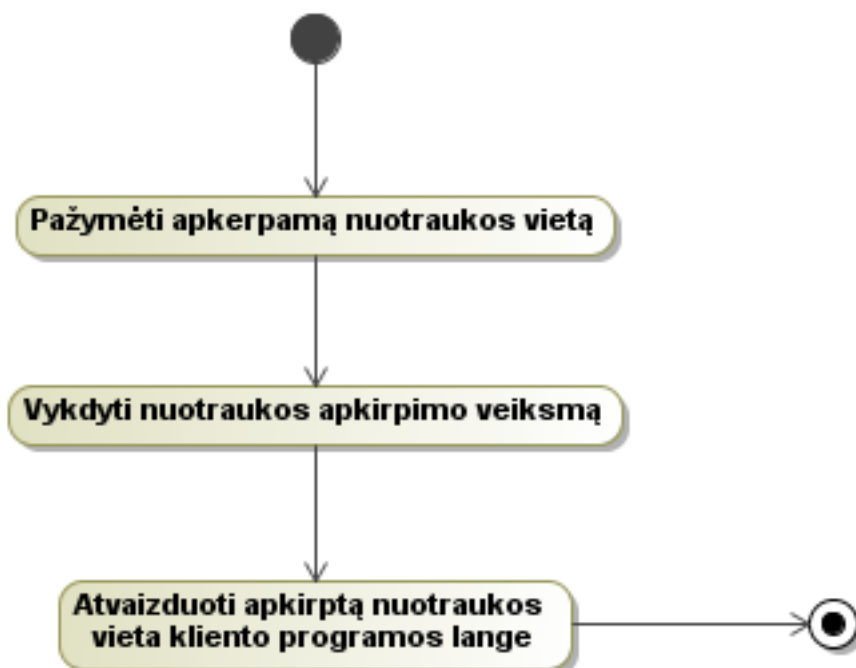
14 pav. Veiklos diagrama „Pateikti nuotrauką iš fotoaparato“

9.1.4.2. Veiklos diagrama „Pateikti nuotrauką iš failo“



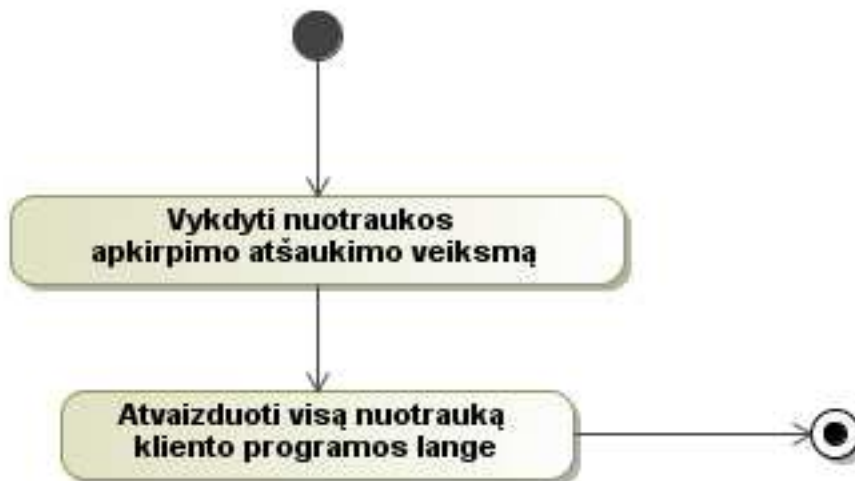
15 pav. Veiklos diagrama „Pateikti nuotrauką iš failo“

9.1.4.3. Veiklos diagrama „Apkirpti nuotrauką“



16 pav. Veiklos diagrama „Apkirpti nuotrauką“

9.1.4.4. Veiklos diagrama „Atšaukti nuotraukos apkirpimą“



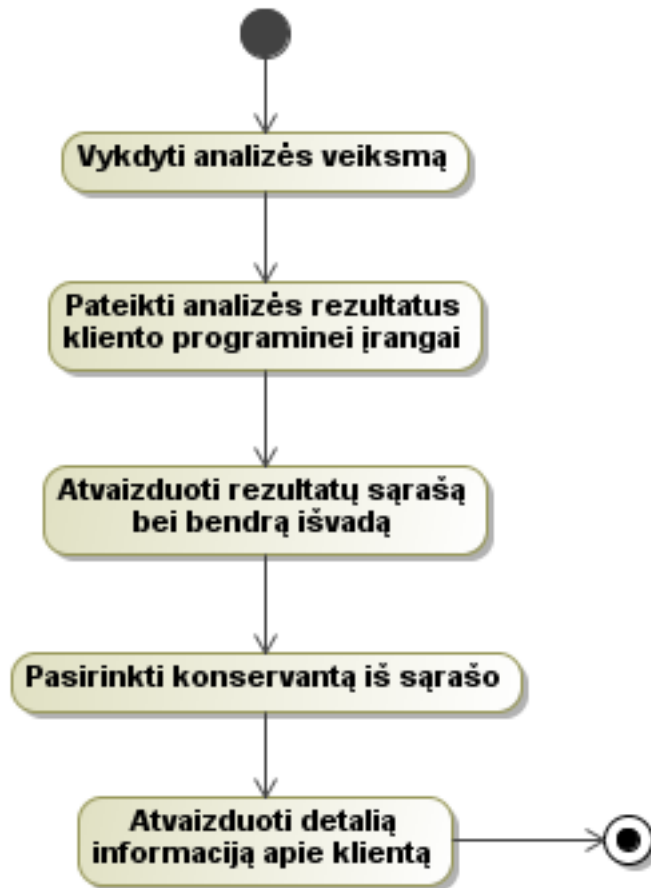
17 pav. Veiklos diagrama „Atšaukti nuotraukos apkirpimą“

9.1.4.5. Veiklos diagrama „Analizuoti“



18 pav. Veiklos diagrama „Analizuoti“

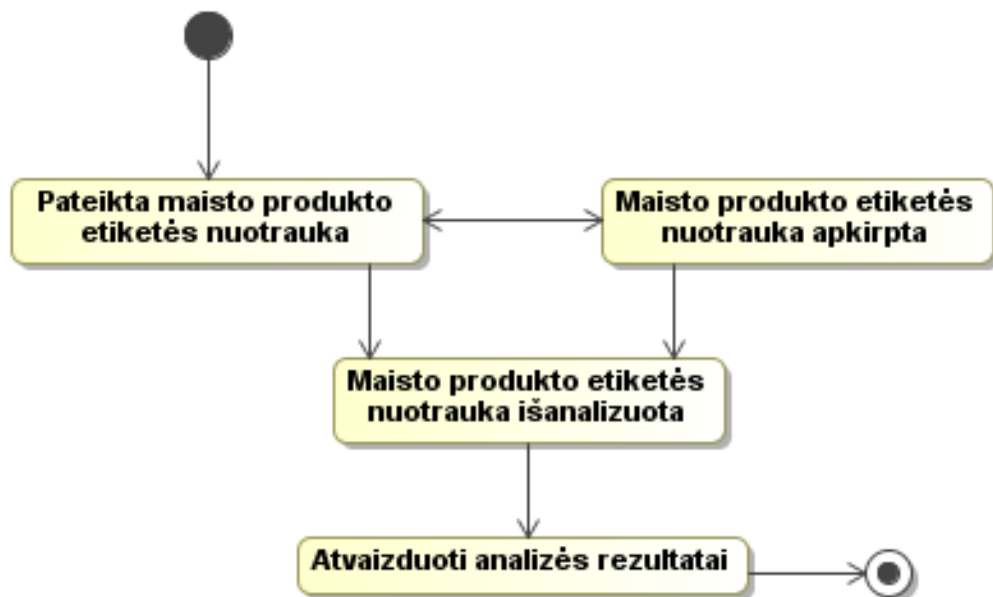
9.1.4.6. Veiklos diagrama „Atvaizduoti rezultatus“



19 pav. Veiklos diagrama „Atvaizduoti rezultatus“

9.1.5. Būsenų diagrama

Programinės įrangos būsenų diagrama pateikta 20 paveikslėlyje.



20 pav. Būsenų diagrama