

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

Renata Baronienė

**Biomedicininį duomenų analizės sistemos modelis**

Magistro darbas

Vadovas doc. dr. E. Paliulis

Šiauliai, 2013

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

TVIRTINU

IT katedros vedėjas doc. dr. M. Bernotas  
2013-05-\_\_

## **Biomedicininų duomenų analizės sistemos modelis**

Informatikos inžinerijos magistro darbas

### **Autorius**

ITM-11 gr. studentė  
2013 m. gegužės 30 d.

R. Baronienė

### **Vadovas**

IT katedros docentas  
2013 m. gegužės \_\_ d.

E. Paliulis

### **Recenzantai:**

IT katedros docentas  
2013 m. gegužės \_\_ d.

M. Bernotas

IT katedros lektorė  
2013 m. gegužės \_\_ d.

S. Ramanauskaitė



TVIRTINU

Informacinių technologijų  
katedros vedėjas

doc. M. Bernotas

2013 m. \_\_\_\_\_ vasario \_\_\_\_\_ mėn. \_26\_ d.

## Magistro darbo užduotis

Studentei **RENATAI BARONIENEI**

**Darbo tema: BIOMEDICININIŲ DUOMENŲ ANALIZĖS SISTEMOS MODELIS**  
(*angl. MODEL OF BIOMEDICAL DATA ANALYSIS SYSTEM*)

Patvirtinta 2013 m. \_\_\_\_ kovo \_\_\_\_ mėn. \_\_29\_\_ d. potvarkiu Nr. \_\_\_\_\_060-03-9\_\_\_\_\_

1. Darbe atlikti šias užduotis:
  - 1.1. Naujausių, šiuo metu rinkoje siūlomų duomenų bazių valdymo sistemų (DBVS) teorinė analizė.
    - 1.1.1. Ištirti ir palyginti labiausiai paplitusias DBVS pagal pasirinktus kriterijus.
  - 1.2. Duomenų gavybos (DG) technologijų analizė.
    - 1.2.1. DG algoritmų analizė ir palyginimas.
    - 1.2.2. DG apdorojimo įrankių analizė ir palyginimas.
  - 1.3. Ligų savybių, bruožų ir diagnostikos analizė.
    - 1.3.1. Simptomų ir reikalingų tyrimų analizė.
    - 1.3.2. Ligos diagnozavimo proceso analizė.
  - 1.4. Sukurti biomedicininį duomenų analizės sistemos projektą.
    - 1.4.1. Atlikti reikalavimų specifikacijas.
    - 1.4.2. Sukurti ligos diagnozavimo proceso algoritmą.
    - 1.4.3. Suprojektuoti duomenų bazę, biomedicininį duomenų kaupimui.
    - 1.4.4. Parinkti duomenų gavybos algoritmą biomedicininį duomenų analizei.
  - 1.5. Pasirinktais įrankiais realizuoti suprojektuotą modelį.
  - 1.6. Atlikti eksperimentą ir praktinį modelio patikrinimą.
2. Aiškinamojo rašto turinys turi atitikti Informacinių technologijų katedros studentų magistro darbų metodinius nurodymus.
3. Magistro darbas pateikiamas su įrišta darbo užduotimi, įdėtu vadovo atsiliepimu ir kompaktine plokštele.
4. Kompaktinėje plokštelėje įrašyti visus su sukurtu algoritmu ir atliktais tyrimais susijusius failus ir magistro darbo aiškinamąjį raštą.
5. Gynimo metu pateikiama pristatymo pateiktis ir liudijimas, kad magistro darbas yra įkeltas į Lietuvos ETD informacinę sistemą.
6. Apginto katedroje baigiamojo darbo pristatymo paskutinioji diena – 2013 m. gegužės 23 d.

**Baigiamojo darbo vadovas**

2013 m. vasario 26 d.

\_\_\_\_\_

(parašas)

**doc. dr. E. Paliulis**

**Studentė**

2013 m. vasario 26 d.

\_\_\_\_\_

(parašas)

**R. Baronienė**

## SANTRAUKA

### **Biomedicininų duomenų analizės sistemos modelis**

Šiuolaikinėms sistemoms svarbu ne tik pasyvus duomenų saugojimas, bet ir jų intelektualus apdorojimas ir net sprendimų priėmimas. Biomedicininės sistemos buvimas leistų operatyviau priimti reikiamus sprendimus sprendžiant įvairias biomedicininės problemas.

Darbo tikslas – panaudojant duomenų gavybos metodikas iširti galimybę diagnozuoti ligą, turint paciento nusiskundimų ir atliktų tyrimų duomenis. Suprojektuotas biomedicininų duomenų analizės sistemos modelis, į kurį įeina duomenų bazės modelis – loginė DB schema, tinkanti universaliu atveju, duomenų gavybos metodų ir įrankių parinkimas, diegimas ir panaudojimas. Pradžioje atliekama duomenų bazių valdymo sistemų, duomenų gavybos metodų ir įrankių teorinė analizė.

Sukurtas biomedicininų duomenų analizės sistemos modelis sugebantis įvertinti ir diagnozuoti ligą pagal paciento nusiskundimus ir tyrimus. Biomedicininei duomenų analizei parinktas duomenų gavybos klasterinės analizės metodas. Atliktas eksperimentas ir praktinis modelio patikrinimas.

Sukurto biomedicininų duomenų analizės sistemos modelio pagrindu galima kurti intelektualias biomedicininės sprendimų priėmimo sistemas gydytojams ir kitiems medicinos darbuotojams kaip pagalbinę priemonę ligų diagnostikai.

**Raktiniai žodžiai:** Duomenų bazių valdymo sistemos, duomenų gavyba, duomenų bazė, biomedicininų duomenų analizė.

## SUMMARY

### **Biomedical Data Analysis System Model**

Current systems is not just a passive data storage, but also the treatment of intellectuals and even decision-making. Biomedical Systems presence would promptly take the necessary decisions to solve various biomedical problems.

A purpose of this Master's Thesis is exploring a possibility to diagnose diseases, using data-mining techniques while having available patient complaint and patient examination data. Biomedical data analysis system model is designed. It includes database model - logical database scheme suitable for universal cases, data-mining methods and tools selection, installation and use. First of all theoretical analysis is carried out for database management systems, data-mining methods and tools.

Biomedical data analysis system model is designed to enable disease diagnosis, while using patient complaint and patient examination data. Data-mining cluster analysis method is used to analyse biomedical data. An experiment is conducted and practical model is tested.

Intellectual biomedical decision-making systems can be created on the basis of this designed biomedical data analysis system model. They can be used to help doctors and other medical staff in diagnosing diseases.

**Keywords:** database management systems, data mining, database, biomedical data analysis.

## SANTRUMPŲ IR ŽYMĖJIMŲ SĄRAŠAS

RDBVS – reliacinė duomenų bazių valdymo sistema.

DBVS – duomenų bazių valdymo sistema.

DB – duomenų bazė.

GUI – grafinė vartotojo sąsaja.

DG – duomenų gavyba.

SQL – Structured Query Language – struktūrizuota užklausų kalba.

OLE DB – standartas, duomenų gavybos proceso specifikavimo dokumentas.

ADO – programavimo įrankis.

DICOM – medicininių duomenų standartas, skirtas vaizdinių duomenų apsikeitimui tarp sistemų.

Ontologija – sąvokų visumos specifikavimas, vaizdinis perteikimas.

XML – duomenų struktūrų ir jų turinio aprašomoji kalba.

ODBC – standartizuota programavimo sąsaja prisijungimui prie duomenų bazių.

JDBC – standartizuota Java kalbos sąsaja su reliacinėmis duomenų bazėmis.

API – aplikacijų programavimo sąsaja.

ODM – Oracle Data Mining.

PHP – programavimo kalba skirta internetinių puslapių kūrimui.

ASP – Microsoft kompanijos sukurta technologija, skirta programavimui WWW tinkle.

Biomedicina – apibendrinta mokslų sritis, mokslo klasifikacijoje apimanti biologijos mokslus, nagrinėjančius gyvąsias sistemas ir medicinos mokslus.

Patogenezė – ligos raida.

## TURINYS

<b>IVADAS</b> .....	<b>8</b>
<b>I. BIOMEDICININIŲ DUOMENŲ ANALIZĖS MODELIO KŪRIMO PROBLEMATIKA</b> .....	<b>10</b>
<b>II. MODELIO REIKALAVIMŲ ANALIZĖ</b> .....	<b>11</b>
<b>III. MODELIO TYRIMO METODIKOS APRAŠYMAS</b> .....	<b>12</b>
3.1. DBVS ANALIZĖ .....	12
3.2. DUOMENŲ GAVYBA.....	16
3.3. DUOMENŲ GAVYBOS ALGORITMAI .....	16
3.4. INTEGRUOJAMI DUOMENŲ GAVYBOS ĮRANKIAI .....	18
3.5. BIOMEDICINOS SAMPRATA.....	19
3.6. DUOMENŲ GAVYBOS PANAUDOJIMO ATVEJAI .....	20
3.7. KITŲ AUTORIŲ ATLIKTI DBVS PALYGINIMAI .....	21
3.8. ANALITINĖS DALIES IŠVADOS .....	22
3.9. SISTEMOS PROJEKTAVIMAS.....	22
<b>IV. MODELIO TYRIMO REZULTATŲ APRAŠYMAS</b> .....	<b>38</b>
4.1. DBVS CHARAKTERISTIKŲ PALYGINIMAS .....	38
4.2. DUOMENŲ GAVYBOS ALGORITMŲ PALYGINIMAS .....	45
4.3. DUOMENŲ GAVYBOS ĮRANKIŲ PALYGINIMAS .....	46
4.4. DUOMENŲ GAVYBOS REALIZACIJA .....	46
4.5. KLASTERINĖ DUOMENŲ ANALIZĖ.....	48
4.6. ANALIZĖS REZULTATŲ PALYGINIMAS .....	53
<b>IŠVADOS</b> .....	<b>55</b>
<b>LITERATŪRA</b> .....	<b>56</b>

## ĮVADAS

Šiuo metu labai aktuali problema yra duomenų kaupimas ir spartus jų apdorojimas, nes duomenų kiekiai ir jų apimtys nuolat didėja. Žmogus jau tampa nepajėgus išanalizuoti tokią duomenų gausą. Tokiame dideliame kiekyje informacijos gali būti ir strategiškai svarbi ir niekinė informacija. Minėtos problemos paskatino atsirasti aukštos kokybės taikomiesiems paketams, programavimo įrankiams, duomenų analizės priemonėms, kurios padeda nepasimesti ir apdoroti informacijos gausą. Žinoma, tai kartu padidino ir vartotojų prieinamumą prie pažangiausių technologijų.

Viena iš tokių technologijų yra duomenų gavyba. Duomenų gavyba (*angl. Data Mining*) – tai procesas, naudojantis įvairius duomenų analizės įrankius, kurie padeda atrasti tokias duomenų struktūras ir ryšius, kurie būtų panaudojami realioms išvadoms ir sistemos rezultatams apibrėžti. Ši technologija sėkmingai taikoma tiek versle, medicinoje ir kitose gyvenimo srityse, kur reikia apdoroti labai didelius informacijos kiekius [1, 2, 3].

Priešasčių, kodėl yra reikalinga kompiuterizuoti medicinos procesus, yra labai daug. Remiantis pasauline patirtimi būtų galima teigti, kad medicinos procesų kompiuterizavimas turi teigiamą poveikį tiek žiūrint iš administracijos, tiek iš medicinos personalo perspektyvų. Duomenų patikimumas medicinoje yra svarbus, kadangi neteisingas klaidingų duomenų interpretavimas gali būti netgi gyvybiškai pavojingas. Struktūrizuotas duomenų pavidalas – dar viena priežastis, kodėl reikalinga kompiuterizuoti mediciną. Be to, duomenys sukaupti duomenų bazių valdymo sistemose gali būti perkelti kitas sistemas ar netgi pritaikyti specializuotoms statistinės analizės taikomosioms programoms [10].

Taigi, duomenų bazių valdymo sistema (DBVS) yra neatsiejama bet kokios informacinės sistemos dalis. Duomenų bazėse (DB) saugoma esminė organizacijos ar įmonės veiklos informacija. Dauguma rinkoje siūlomų ir įmonėse naudojamų DBVS veikia reliacinio modelio pagrindu [9]. Galima teigti, kad reliacinis modelis vis dar iš lieka populiariausias, nors yra sukurta naujesnių duomenų saugojimo modelių.

Analizuojant literatūrą nepavyko rasti išsamesnių DBVS tinkamumo biomedicininėms sistemoms įvertinimo. DBVS pasiūla didelė, todėl kartais sunku apsispręsti, kurią reikėtų pasirinkti. Todėl charakteristikų tyrimas ir palyginimai gali padėti apsispręsti pasirenkant reikiamą DBVS konkrečiu jos panaudojimo atveju.

**Tikslas:** sukurti universalų biomedicininį duomenų analizės sistemos modelį skirtą gydytojams ar medicinos personalui.

### **Uždaviniai:**

1. Atlikti populiariausių reliacinių, reliacinių-objektnių duomenų bazių valdymo sistemų teorinę analizę.
2. Atlikti duomenų gavybos metodų bei technologijų, duomenų gavybos apdorojimo įrankių analizę.
3. Išanalizuoti ligos savybes, bruožus ir diagnostikos procesą.
4. Suprojektuoti reikalavimų modelį, kuriame atspindėtų sistemos paskirtis, projekto dalyviai, vartotojai, panaudojimo atvejai, reikalavimai, projekto išėiga, sprendžiami uždaviniai bei įvairūs sistemos apribojimai.
5. Suprojektuoti duomenų bazę biomedicininį duomenų kaupimui.
6. Biomedicininį duomenų analizei parinkti duomenų gavybos algoritmą.
7. Atlikti eksperimentą ir praktinį modelio patikrinimą.

**Objektas:** duomenų bazių, duomenų gavybos įrankių ir metodų naudojimas, biomedicininį duomenų apdorojimo ir analizės palengvinimui.



**Nauda:** palengvinamas gydytojų, tyrinėtojų ir analitikų darbas, greičiau apdorjami duomenys, sparčiau gaunamas rezultatas. Tai leistų operatyviau priimti reikiamus sprendimus sprendžiant įvairias biomedicines problemas.

**Kuriamo modelio ribos:** tiriamos reliacinės, objektinės-reliacinės duomenų bazių valdymo sistemos, pagrindinės jų funkcijos ir galimybės, turimi duomenų tipai. Tyrimo eksperimente analizuojami tik skaitmeninio ir simbolinio tipo duomenys. Nenagrinėjami vaizdiniai ar garsiniai tyrimų duomenys. Modelio realizacija ir eksperimentas bus atliekamas tik su konkrečiomis, po palyginimų nustatytais tinkamiausiomis priemonėmis (DBVS, DG įrankiu, DG metodikomis ir t.t.)

**Darbo aprobavimas:**

E. Paliulis, R. Baronienė. *Reliacinių duomenų bazių valdymo sistemų charakteristikų analizė*. Technologijos fakulteto konferencija „Studentų moksliniai darbai“. Šiaulių universitetas. 2012.

E. Paliulis, R. Baronienė. *Duomenų bazių valdymo sistemų tinkamumo biomedicininėms sistemoms įvertinimas* // Jaunųjų mokslininkų darbai. ISSN 1648-8776. 2012, 5(38), p. 91-96.

## I. BIOMEDICININIŲ DUOMENŲ ANALIZĖS MODELIO KŪRIMO PROBLEMATIKA

Ligos diagnozavimas yra sudėtingas ir didžiulės atsakomybės reikalaujantis procesas. Kad būtų patvirtinta diagnozė, reikalingi tam tikri kriterijai, todėl tirdami pacientą, gydytojai ieško kuo daugiau duomenų, patvirtinančių ligą. Tuo tikslu atliekami įvairūs tyrimai – laboratoriniai tyrimai, neurologiniai tyrimai, magnetinio rezonanso, tomografijos tyrimai, ir pan. Diagnozuoti ligą nėra paprasta, todėl laboratoriniai tyrimai gali papildomai padėti. Diagnozės negalima patvirtinti remiantis tik apžiūra ar remiantis tik paciento nusiskundimais. Negalima diagnozės nustatyti remiantis ir vien tik tyrimo atsakymais. Žinoma, visi tyrimai padeda nustatyti galutinę diagnozę, tačiau kiekvienas jų turi būti vertinamas atsakingai, patyrusio specialisto.

Kaip pagalbininkas gydytojams gali ateiti kompiuteris – įdiegus atitinkamus algoritmus, jis pagreitina ligos diagnozavimą, o svarbiausia – leidžia atlikti analizes ir naudoti šiuos duomenis tiek gydymo kokybės užtikrinimo, tiek mokslo tikslais. Savaimė suprantama, žmogaus-gydytojo darbo kompiuteris pakeisti negali, o ir ligos diagnozavimas kažin ar kada nors būtų patikėtas tegul ir labai galingam kompiuteriui, bet kaip pagalbininkas gydytojui jis darosi vis vertingesnis.

Siekiant užsibrėžtų tikslų svarbiausia problema – universalus biomedicininės sistemos modelio projektavimas ir kūrimas, kuris susideda iš:

- DB projekto – loginė DB schema universaliai tinkanti įvairių biomedicininų tyrimų, nusiskundimų parametrų, ligų parametrų suvedimui. Suprojektuoti DB loginę schemą, kuri tiktų universaliu atveju, suvedinėjant įvairius biomedicininus duomenis.
- Duomenų gavybos metodų ir įrankių parinkimo – diegimas ir jų panaudojimas nustatant ligą pagal įvestus tyrimo ir nusiskundimo parametrus. Parinkti ir konfigūruoti duomenų gavybos įrankius ir metodus, kad jie padėtų nustatyti ligas.

Antroji problema, tai DBVS pasirinkimas biomedicininei sistemai. Rinkoje egzistuoja daugiau nei penkiasdešimt įvairių DBVS. Kyla natūralus klausimas, kuri geriau tinka biomedicininėms sistemoms, kurią DBVS geriau pasirinkti kuriant platformą.

Trečioji problema – duomenų gavyba. Reikalingos informacijos išgavimas iš DB esančių biomedicininų duomenų. Šiuolaikinėms sistemoms svarbu ne tik pasyvus duomenų saugojimas, bet ir jų intelektualus apdorojimas ir net sprendimų priėmimas. Duomenų gavybos algoritmų yra daug, vėl kyla klausimas, kuris geriau tiktų biomedicininėms sistemoms, kurį metodą geriau pasirinkti kuriant tokį modelį.

Biomedicininų sistemų išskirtinumas pastebimas, ypač jei kalbama apie universalesnes biomedicininės sistemas. Biomedicininėse sistemose duomenys pasižymi kompleksišku ir didele tipų įvairove, nuo paprastų skaičių, simbolių iki nuotraukų, vaizdų ir net video įrašų, ligos bruožai yra specifiniai, todėl ligos diagnozavimo procesas yra labai svarbus, taip pat kaip ir programinės įrangos, kaip pagalbinės priemonės parinkimas.

Biomedicininų duomenų analizės sistemos buvimas leistų gydytojui greičiau priimti sprendimą. Žinoma, sprendimas liks gydytojo kompetencijoje, tai būtų daugiau patarimoji sistema.

Būtent tokios sistemos projektavimo galimybės ir pats modelis nagrinėjamas šiame darbe. Panašių, konkrečių modelių mokslinėje literatūroje nepavyko aptikti.

## II. MODELIO REIKALAVIMŲ ANALIZĖ

Projektuojant sistemą, padėsiančią gydytojui diagnozuoti pacientui ligą, yra svarbu suprojektuoti loginę duomenų bazės schemą universaliai tinkančią įvairių biomedicininį tyrimų, nusiskundimų parametrų, ligų parametrų, biomedicininį duomenų suvedimui. Ligos diagnozavimo procesas yra sudėtingas dėl savo ypatumų, tokių kaip įvairūs skirtingų ligų bruožai, ligų požymiai ir simptomai. Kad būtų lengviau suprasti, ligos paieškai aprašyti, simptomams ir tyrimams charakterizuoti pateikta ontologija.

Renkantis DBVS biomedicininį duomenų saugojimui ir apdorojimui svarbi duomenų tipų įvairovė ir galimybė naudotis specializuotais duomenų gavybos įrankiais. Tokios sistemos išskirtinės tuo, kad medicininiai duomenys pasižymi kompleksiskumu, labai didele tipų įvairove, ligos bruožai yra specifiniai, todėl ligos diagnozavimo procesas yra labai svarbus, taip pat kaip ir programinės įrangos kaip pagalbinės priemonės parinkimas.

Duomenų gavybos metodų ir įrankių parinkimas, diegimas ir jų panaudojimas nustatant ligą pagal įvestus tyrimo ir nusiskundimo parametrus. Duomenų gavybos algoritmo metodo parinkimas reikalingos informacijos išgavimui iš duomenų bazėje esančių biomedicininį duomenų. Programinės įrangos pasirinkimui reikalinga atlikti išsamią DBVS ir DG įrankių bei metodų analizę.

Darbo tikslas yra panaudojant programinę įrangą, įgyvendinti galimybę gydytojui ligos nustatymo galimybę, remiantis paciento nusiskundimais ir turimais tyrimais. DB skirta informacijos apie paciento sveikatos būklę, ligas, kaupimui ir apdorojimui ir pritaikoma medicininėse įstaigose kad palengvinti medikų darbą, nustatant pacientui ligą. Praktinė prasme, darbo tikslas yra panaudojant programinę įrangą, įgyvendinti galimybę analizuoti medicininis duomenis.

Sistemos paskirtis – registruoti pacientus, pacientų nusiskundimus, simptomus, kitus duomenis, susijusius su paciento tyrimais. Duomenų analizavimas pasitelkiant duomenų gavybos įrankius, siekiant išsiaiškinti duomenų gavybos įrankio pritaikomumą medicininiam duomenims. Sistema svarbi modernizuojant medicinos sistemą, didinant jos efektyvumą ir gerinant medicinos paslaugų kokybę.

### III. MODELIO TYRIMO METODIKOS APRAŠYMAS

Sistemos projektavimas bus vykdomas trim etapais.

Pirmajame etape renkama literatūra, susijusi su biomedicina, biomediciniais duomenimis, duomenų gavybos ir jų apdorojimo įrankiais. Išanalizuotos ir aprašytos duomenų bazių valdymo sistemos: MS SQL Server, Oracle, MySQL, PostgreSQL, DB2, Firebird bei Informix. Remiantis šaltinių pateiktais duomenimis [60, 61, 62] pasirinktos populiariausios reliacinės bei objektinės-reliacinės DBVS. Reliacinis modelis yra plačiai naudojamas dėl savo lankstumo ir taikymo paprastumo, modelis duomenis apdoroja ir pateikia lentelių pavidalu. Failinės DB sistemos atmestos nes neužtikrina tinkamos prieigos prie duomenų bazės, saugumo, skirtingų vartotojų lygių naudojimo, tai pat problematiška realizuoti sudėtingus sąryšius. DB turi palaikyti kliento-serverio architektūrą.

Apžvelgti moksliniai tyrimai duomenų gavybos metodų taikyme biomedicinos srityje.

Sekančiame etape detaliam sudaromas reikalavimų modelis, kuriame atspindėtų sistemos paskirtis, projekto dalyviai, vartotojai, panaudojimo atvejai, reikalavimai, projekto išeiga, sprendžiami uždaviniai bei įvairūs sistemos apribojimai. Pagal sudarytus reikalavimus nubraižytos kuriamos sistemos panaudojimo atvejų diagramos, suprojektuota DB loginė schema bei esybių ryšių diagrama. Ligos paieškai aprašyti, simptomams ir tyrimams charakterizuoti sudaryta ir pateikta ontologija.

Paskutiniame etape atliktas eksperimentas, jo analizė ir detalus aprašymas, taip pat pateiktos eksperimento išvados.

#### 3.1. DBVS analizė

##### 3.1.1. *Microsoft SQL Server*

Microsoft SQL Server yra reliacinė duomenų bazių valdymo sistema, kurią sukūrė kompanija „Microsoft“. Naujausia versija – Microsoft SQL Server 2012.

Microsoft SQL Server yra platinama pagal nuosavybinę licenciją. Microsoft SQL Server yra reliacinė duomenų bazių valdymo sistema. Microsoft SQL Server gali funkcionuoti Windows Server 2003 Service Pack 2, Windows Server 2008, Windows Vista, Windows Vista Service Pack 1, Windows XP Service Pack 2, Windows XP Service Pack 3 platformose [6, 9, 28].

„SQL Server 2008“ palaiko šias išorines programavimo priemones: VB, C++, C#, XML.

MS SQL Server yra galimos tokios XML savybės:

- HTTP pasiekiamumas (rašant SQL sakinių naršyklės adresų eilutėje);
- Formuoti XML galima SELECT sakinio pagalba (FOR XML);
- XML views (nurodoma kaip sąryšiniai duomenys turi būti atvaizduoti XML);
- XPath queries (nurodoma kaip duomenys turi būti paimti iš DB);
- OpenXML (nurodo, kaip duomenys turi būti atvaizduoti XML ir suteikia galimybę XML failo duomenis atvaizduoti duomenų bazės lentelėse);
- OLE DB ir ADO pasiekiamumas (naudojama sąsajai su sąryšinėmis DB formuojant XML iš DB ir surašant XML duomenis į DB).

Microsoft SQL Server SQL kalbų standartai - Transact-SQL(T-SQL) ir ANSI SQL [9, 14, 21].

##### 3.1.2. *Duomenų gavybos įrankis MS SQL 2008 Data Mining Add-Ins for MS Excel 2007*

Šis duomenų gavybos įrankis nemokamas, tačiau norint juo naudotis reikia turėti įdiegtą ne tik MS Excel 2007, bet ir MS SQL Server 2005 ar 2008 su Analysis Services;

Patogus dėl savo naudojimo paprastumo;

Įrankis turi savyje daug duomenų gavybos metodų:

- prognozavimas,
- klasifikavimas,

- loginė regresija,
- klasterizavimas Naive Baye's algoritmas,
- asociacijų taisyklės nustatymas,
- prekių krepšelio uždavinys,
- lentelės užpildymas pagal pavyzdį [6, 7].

### 3.1.3. Oracle

Oracle korporacija pasaulyje yra didžiausia verslo programinės įrangos gamintoja, tiekianti verslo programinę įrangą didžiausioms pasaulio kompanijoms. Oracle produktų ir paslaugų spektrą sudaro duomenų bazės, jų programavimo įrankiai bei aplikacijų priemonės, konsultacijos bei kitos paslaugos.

Paskutinė išleista versija Oracle Database 11g Release 2 Standard Edition, Standard Edition One, ir Enterprise Edition. Oracle DBVS platinama pagal nuosavybinę licenciją. Licencija privaloma visiems faktiniams vartotojams arba procesoriams.

Oracle duomenų bazėse yra įdiegtas objektinis-reliacinis duomenų valdymo modelis, kuris leidžia naudotis plačiausiai pasaulyje paplitusiomis reliacinės duomenų bazių priemonėmis ir standartais, bei neribotai plėsti jį naudojant duomenų bazėje saugomus objektus [13]. Oracle objektai palaiko daugelį objektinio modelio savybių, tokių kaip savybės ir metodai, paveldimumas, inkapsuliacija, kompozicija ir perdengimas. Šios savybės leidžia sukurti įvairialypius, funkcionaliūs bei integruotus duomenų valdymo sprendimus.

Oracle veikia per DB administravimo priemonę Oracle SQL Developer. Oracle DBVS serveris veikia nutolusiame kompiuteryje, užtikrina duomenų saugojimą, vartotojų užklausų aptarnavimą, duomenų integralumą ir saugumą. Kiekvieną kartą jungiantis sukuriama atskira sesija, kuriai reikalingas slaptažodis.

Oracle palaikomi programavimo įrankiai: Java, C, C++ ir PL/SQL.[13].

Programavimo kalbų standartai SQL-92.

### 3.1.4. MySQL

MySQL yra nemokama atviro kodo programinė įranga, kurios įkūrėjai – švedai David Axmark, Allan Larsson ir suomis Michael „Monty“ Widenius. MySQL – populiariausia pasaulyje atvirojo kodo duomenų bazių valdymo sistema. Šiuo metu MySQL priklauso Oracle kompanijai.

Naujausia MySQL versija – MySQL 5.6.5, tačiau palaikymas nėra nutrauktas ir MySQL 5.1.62, MySQL 5.5.21 versijoms.

MySQL yra reliacinė duomenų bazių valdymo sistema, palaikanti daugelį naudotojų, dirbanti struktūrinės užklausų kalbos (SQL) pagrindu. MySQL funkcionuoja Windows 2000/XP/Vista, Windows Server2003/2008, Mac OS, Linux, Unix, Solaris, AIX, HP-UX Itanium, Amiga OS, BSD platformose [6, 9].

MySQL veikia kliento–serverio principu, t.y naudodamiesi MySQL klientine programa, siunčiame SQL užklausas, pagal kurias serveris atlieka jam nurodytus darbus. MySQL ypač populiaru yra web srityje. Kuriant dinaminis puslapius ar portalus dažnai pasirenkama būtent MySQL. Šiuo metu MySQL yra bene greičiausia duombazių sistema. Ji lengvai instaliuojama įmonės serveryje ir administruojama.

MySQL turi šias programavimo priemones: C, C++, C#, Java, Perl, Python. Kiekvienai šių kalbų sukurtos specialios bibliotekos (API). Taip pat MySQL duomenų bazėms yra sukurta ODBC sąsaja MyODBC, leidžianti duomenis pasiekti bet kuria kalba, neturinčia specialios bibliotekos, tačiau palaikančia ODBC komunikavimo mechanizmą. PHP kalba jai parašytas valdymo įrankis phpMyAdmin. MySQL Connectors (ODBC, JDBC, .NET) – aplikacijų kūrimas įvairiomis programavimo kalbomis [21, 32].

MySQL siūlo keletą unikalių saugyklų variklių, kurie skirti specifiniams aplikacijų reikalavimams atitikti. Vienas iš jų – InnoDB – skirtas labai dideliems transakcijų kiekiams apdoroti. Kitas – MyISAM – saugyklos variklis skirtas palaikyti didelio apkrovimo interneto puslapius su terabaito dydžio duomenų saugyklomis. Kadangi MySQL yra atvirojo kodo, vartotojas pats gali kurti savo saugyklų variklius ir juos naudoti [6, 9, 32].

MySQL programavimo kalbų standartai SQL-92, ANSI/ISO SQL.

### **3.1.5. PostgreSQL**

PostgreSQL, iš pradžių vadinama Postgres, buvo sukurta Kalifornijos Berkeley Universitete ir pateikta kompiuterių mokslo profesoriaus Michael Stonebraker. Profesorius ir jo absolventai Postgres aktyviai plėtojo aštuonerius metus. Per tą laiką, Postgres nusistatė taisykles, procedūras, indeksų tipus ir objektines-reliacines sąvokas. Vėliau sistemos vardas buvo pakeistas į PostgreSQL.

Paskutinė išleista versija – PostgreSQL 9.1.2 [31].

PostgreSQL yra platinama pagal BSD licenciją kuri suteikia kūrėjų bei platintojų nepriklausomą sistemos panaudojimą arba modifikavimą pagal vartotojo poreikius. Ši sistema jau turi tvirtą reputaciją dėl patikimumo, duomenų vientisumo ir teisingumo.

PostgreSQL yra galinga objektinė-reliacinė duomenų bazių valdymo sistema.

PostgreSQL gali funkcionuoti Linux, Unix, AIX, BSD, HP-UX, SGI-IRIX, Mac OS X, Solaris, Windows platformose [6, 9].

PostgreSQL palaikomi programavimo įrankiai: PL/pgSQL, PL/Perl, C, C++ bei Java.

Be standartinių SQL galimybių, PostgreSQL duomenų bazių valdymo sistema palaiko[15]:

- MVCC (Multi-Version Concurrency Control). Užklausi pateikia tuometinį duomenų bazės vaizdą, leidžia padaryti savo pakeitimus, būnant nematomam kitų užklausių atžvilgiu iki tol, kol transakcija vyksta.
- išsaugotas procedūras ir triggerius įvairiomis kalbomis (įskaitant vidinę PL/pgSQL),
- didelių binarinių objektų laikymą bei daugybę kitų funkcijų,
- turi vidinį programavimo interfeisą, pritaikytą dirbti su C, C++, Java, Perl, Python, Ruby,
- yra sukurtos ODBC tvarkyklės,
- didžiąja dalimi suderinama su ANSI SQL 92/99 standartais.

Programavimo kalbų standartas PL/pgSQL.

PostgreSQL, panašiai kaip ir MySQL, veikia kliento–serverio principu, t.y. naudojantis klientine programa, siunčiamos SQL užklauso, pagal kurias serveris atlieka jam nurodytus darbus[6].

### **3.1.6. Informix**

Informix yra IBM kompanijos įsigyta ir vystoma duomenų bazių valdymo sistema. Jos funkcionalumas, našumas ir palaikymas atitinka geriausių komercinių produktų lygį.

Paskutinė išleista versija Informix Dynamic Server (IDS) 11.7. Informix licencijos tipas – nuosavybinė licencija. Informix yra reliacinė duomenų bazių valdymo sistema, galinti funkcionuoti Linux, Unix, Windows, Mac OS(10.5.2 ir vėlesnė), Solaris 64-bit, AIX, BSD platformose [6, 9].

Informix DBVS taikoma didelėse įmonėse ir organizacijose kur reikia apdoroti didžiulius duomenų kiekius.

Šioje DBVS taip lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau taip pat kaip ir SQL Server, duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.

Informix palaikomi programavimo įrankiai: Java ir COM+.

Informix SQL kalbų standartas SQL-92 ir ANSI SQL [19, 21].

### 3.1.7. DB2

IBM DB2 yra vienas iš kompanijos IBM kuriamų produktų. DB2 duomenų bazių valdymo yra laikoma viena pirmųjų DBVS naudojančių SQL programavimo kalbą. Nors dinaminiai SQL sakiniai sistemoje DB2 buvo nuo pat jos egzistavimo pradžios, standarte SQL1 šių sakinių nėra. Panašius SQL sakinius įdiegus daugelyje komercinių DBVS, sistemos DB2 dinaminiai SQL sakiniai tapo priimtu standartu. Paskutinė versija – DB2 9.7 [9, 16]. Licencijos tipas – nuosavybinė.

IBM DB2 yra objektinė-reliacinė duomenų bazių valdymo sistema. IBM DB2 funkcionuoja Windows(32bit, 64 bit), AIX, HP-UX, Linux, Linux iSeries, Linux pSeries, Linux xSeries, Linux zSeries, Solaris (x64) platformose [16, 33].

Ši programa turi grafinę vartotojo sąsają ir SQL programavimo kalbą. Darbui su DB2 yra jungiamasi per atskirą vartotoją, duomenų bazė pajungiama per Control Center. Lentelės gali būti kuriamos „Next“ principu. Norint pradėti dirbti DB2, reikia tam tikro pasirengimo bei išmanymo, tačiau ši DBVS gana nesudėtinga.

IBM DB2 palaikomi programavimo įrankiai:

- SQL, XQuery, Xpath;
- C/C++ (CLI, ODBC and embedded SQL);
- Java (JDBC and SQLJ);
- COBOL;
- PHP;
- Perl;
- Python;
- Ruby on Rails;
- .NET languages;
- OLE-DB;
- ADO.

IBM DB2 SQL kalbų standartas SQL-92 [33, 35].

### 3.1.8. Firebird

Firebird yra naudojama pagal IPL (angl. InterBase Public License) licencija, kuri yra visiškai nemokama naudojant tiek asmeniniams, tiek komerciniams tikslams. Paskutinė išleista versija Firebird 2.5.1 [21].

Firebird yra reliacinė duomenų bazių valdymo sistema. Firebird DBVS funkcionuoja Linux, Windows and Mac OS X platformose.

Firebird yra galingas tačiau nereikalaujantis didelių konfigūravimo ar administravimo darbų, duomenų bazės serveris. Firebird tinka tiek darbui vienam vartotojui su viena duomenų baze, tiek verslui. Vienas Firebird serveris gali palaikyti kelias duomenų bazes su galybe prisijungusių klientų.

Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.

Firebird turi daug prisijungimo variantų, įskaitant į C/C++ ar Delphi integruotus paketus, ODBC, JDBC, PHP, dbExpress, .NET ir per tiesiogines API užklausas [6, 9].

Firebird palaiko SQL 99 ir ANSI SQL kalbos standartą.

Naudojant Firebird'o PSQL (Procedural SQL), galima kurti galingas procedūras, kurios bus saugomos ir vykdomos serveryje. Išorinės funkcijos – tai vartotojų sukurtos funkcijos, UDF (User Defined Function) bibliotekos. Jos turėtų būti kuriamos C, C++ arba Delphi programavimo kalbomis. Sukurta biblioteka praplečia Firebird galimybes pagal konkrečią užduotį [60].

### 3.2. Duomenų gavyba

Duomenų gavybą galima apibūdinti kaip procesą, kurio metu iš duomenų bazių yra išgaunama informacija. Paruošti duomenys perduodami duomenų gavybos algoritmui, kuris pateikia rezultatą grafiniu ar kitokiu pavidalu. Duomenų kokybė yra labai svarbus aspektas duomenų gavybos procese, nes tik su kokybiškais duomenimis galima kokybiškai atlikti duomenų išgavimą. Dažnai šią sąlygą įvykdyti yra gana sunku, kadangi realūs duomenys beveik nebūna paruošti duomenų gavybai, nes jie turi būti integruojami iš skirtingų duomenų šaltinių, gali turėti klaidų arba neatitikimų [22, 39].

Duomenų gavybos analizė dažniausiai atliekama naudojant vieną arba keletą pasirinktų medodų. Duomenų išgavimo iš duomenų bazės technika ir duomenų gavybos algoritmai priklauso ne tik nuo pačių duomenų, bet ir nuo jų kilmės, užduoties struktūros, ir pan. Vienas iš didžiausių duomenų išgavimo privalumų yra tas, kad programinė įranga gali rasti tokius modelius, kurie nėra lengvai pastebimi ir akivaizdūs, tačiau labai naudingi tiriamoje srityje [4, 5, 39].

### 3.3. Duomenų gavybos algoritmai

#### 3.3.1. Neuroninių tinklų algoritmas

*Neuroninių tinklų algoritmas (angl. Neural Network)*. Šie algoritmai yra naudojami prognozavimo bei klasifikavimo uždaviniams spręsti. Tai netiesiniai modeliai savo struktūra primenantys biologinius neuroninius tinklus. Neuroniniai tinklai suteikia galimybę mokytis iš duomenų ir praplėsti ateities numatymo ribas. Algoritmas naudojamas didelio kompleksškumo duomenų analizei. Pats tinklas yra sudarytas iš kelių sluoksnių: įvesties, išvesties ir vieno ar daugiau paslėptų sluoksnių. Įvesties sluoksnyje yra elementą aprašanti informacija, išvesties sluoksnyje yra gaunamas rezultatas, o paslėptuose sluoksniuose realizuojama uždavinio logika [24, 49].

Panašiai kaip sprendimų medžio algoritme, kiekvienam įėjimo atributui paskaičiuojama tikimybė, kai žinomi visi galimi išėjimai. Sluoksniai yra susiję tik vienas su kitu, tačiau ne tarpusavyje. Tai reiškia kad įėjimo sluoksnyje mazgai nėra tarpusavyje surišti, o rišasi tik su viduriniu sluoksniu, o jei jo nėra - su išėjimo sluoksniu. Kiekvienas išėjimas yra netiesinė funkcija, susumuota iš įėjimo funkcijų [34, 49].

#### 3.3.2. Sprendimų medžių algoritmas

*Sprendimų medžių algoritmas (angl. Decision trees)*. Tai yra klasifikavimo ir regresijos algoritmas. Šį algoritmą galima įsivaizduoti kaip modelį, kuris yra panašus į medį. Išsiskojimai reiškia atsakymus į siekiamus išsiaiškinti klausimus. Diskrečių dydžių atveju, algoritmas ieško galimo ryšio tarp keleto atributų reikšmių priklausomai nuo pasitaikančių ryšių, juos skirsto į dvi šakas. Kad nustatyti, kur sprendimų medis išsiskakos, tolydžių požymių grupavimui, algoritmas naudoja tiesinę regresiją [45].

Sprendimų medžio algoritmo vizualinė išraiška susideda iš dviejų dalių. Pirmos – medžio, kuriame stebimas tiek populiacijos pasiskirstymas, tiek pasirinkto atributo konkrečios būsenos pasiskirstymas. Antros – priklausomybių įtaka. Čia stebima, kurie dydžiai labiausiai įtakoja analizuojamą atributą. Silpnesnius ryšius galima eliminuoti [59].

Pagrindiniu sprendimų medžių privalumu laikomas jų aiškumas ir suprantamumas. Šio algoritmo minusas tas, jog jam reikia pateikti pilnai užpildytus duomenis, o toks duomenų formavimas užima laiko [45].

#### 3.3.3. Bajeso algoritmas

*Bajeso algoritmas (angl. Naive Bayes)*. Tai yra klasifikavimo algoritmas, naudojamas spėjimų modeliavimui. Kadangi skaičiavimų prasme šis algoritmas yra vienas paprasčiausių – taikomas preliminariems įvertinimams, vėliau spėjimus galima tikslinti pasitelkiant kitus algoritmus [42, 48].

Taikant Naive Bayes algoritmą traktuojama, kad visi įėjimo stulpelių atributai yra nepriklausomi. Šis algoritmas taikomas tik diskretiems arba diskretizuotiems požymiams tirti.



### 3.3.4. Klasterių algoritmas

*Klasterių algoritmas (angl. Cluster).* Kad sugrupuotų duomenų rinkinius į klasterius, turinčius panašias charakteristikas, šis algoritmas naudoja iteratyvias metodikas. Grupavimai yra reikalingi duomenų peržiūrai, duomenyse esančių netikslumų identifikacijai ir prognozių kūrimui. Klasterių modeliai apibrėžia duomenų rinkinio tarpusavio ryšius, kuriuos nustatyti rankiniu būdu yra labai sudėtinga. Norint sukurti klasterių modelį, nereikia apibrėžti prognozuojamo stulpelio. Algoritmas apmoko modelį pagal duomenyse egzistuojančius ryšius ir pagal algoritmo atpažįstamus klasterius. Visų pirma algoritmas identifikuoja duomenų rinkinio tarpusavio ryšius ir sugeneruoja eilę klasterių, atsižvelgdamas į esamus ryšius [27, 42, 44]. Yra du būdai paskaičiuoti priklausomumą kuriai nors grupei. Pirmasis – tikimybės priklausymo vienai ar kitai grupei skaičiavimas (kur tikimybė didžiausia – tai grupei objektas priklausys). Kitas būdas – artimiausio kelio iki atitinkamos grupės radimas.

### 3.3.5. Asociacijų taisyklės

*Asociacijų taisyklės (angl. Association rules).* Algoritmo veikimas: peržiūrimi visi duomenų įrašai, tuo būdu atrenkami tokie, kurie turi panašių sąsajų. Įrašai grupuojami į tyrėjo apibrėžtą grupių skaičių. Kiekvienai grupei generuojama charakterizuojanti taisyklė, pagal kurią galima aptikti dėsningumus naujiems atvejams: nuspėti, kokiai kategorijai jie priklausys. Asociacijų algoritmas taikomas tik diskretiems dydžiams [26, 43].

### 3.3.6. Sekų grupavimo algoritmas

*Sekų grupavimo algoritmas (angl. Sequence Clustering).* Sekų grupavimo algoritmas yra prognozuojantis algoritmas. Šis algoritmas gali dirbti tik su vienai sričiai priklausančiais duomenimis. Algoritmas gali nuspėti, kas gali įvykti. Sekų grupavimo algoritmas panašus į grupavimo algoritmą, tačiau jeigu grupavimo algoritme užtenka surasti skirtingų požymių grupes, tuomet čia reikia išskirti panašių veiksmų grupes. Tai gali būti žiniatinklio puslapių navigacijos veiksmų seka, prekių krepšelio papildymo seka ir pan. [50].

Algoritmo veikimas paremtas tuo, kad ieškomas įvykių pasitvirtinimas. Pirmiausia yra ieškomas įvykio pasitvirtinimas įvykus vienam įvykiui, toliau ieškomas – įvykus dviems įvykiams ir t.t. Yra fiksuojamos jų įvykimo tikimybės, kurios vėliau pateikiamos rezultatuose.

### 3.3.7. Logistinės regresijos algoritmas

*Logistinės regresijos algoritmas (angl. Logistic Regression).* Neuroninių tinklų algoritmo atmaina, kai nenaudojamas vidurinis neuronų sluoksnis. Taikant šį algoritmą tiesiogiai apsprendžiama, koks ryšys yra tarp įėjimo atributų stulpelių, kad būtų galima nuspėti išėjimo atributo reikšmes. Šiam algoritmui įėjimais gali būti tiek diskretūs tiek tolydūs atributai.

Logistinės regresijos algoritmas taikomas tekstinės analizės, klasifikacijos bei įvertinimo uždaviniams išspręsti. Iš esmės logistinės regresijos algoritmo parametrai tokie patys, kaip ir neuroninio tinklo. Nėra tik viduriniojo sluoksnio neuronų skaičių apsprendžiančio parametro [47].

### 3.3.8. Tiesinės regresijos algoritmas

*Tiesinės regresijos algoritmas (angl. Linear Regression).* Sprendimų medžio algoritmo atmaina kai žinomų galimų būsenų skaičius yra mažesnis arba lygus nei numatoma prognozuojant naujų duomenų porciją. Šiuo atveju niekada negausime dviejų tiesių lygčių, tik vieną, taigi sprendimų medis niekur neišsišakos. Paprasčiausias taikymo pavyzdys: turime ledų paklausos duomenis, priklausančius nuo temperatūros. Uždavinys – nustatyti, kiek reikia užsakyti ledų, jei prognozuojama tam tikra temperatūra, kad nebūtų nei pertekliaus, nei stygiaus [46].

### 3.3.9. Laiko sekų algoritmas

*Laiko sekų algoritmas (angl. Time series).* Tai regresijos algoritmas naudojamas kuriant intelektualios duomenų gavybos modelius, siekiant nuspėti tolydžius stulpelius. Laiko sekų modelio prognozavimas pagrįstas tendencijomis, gautomis iš pradinio duomenų rinkinio, modelio kūrimo

metu. Tai gali būti produkto pardavimo tendencijos, kainų augimo prognozavimas ir panašiai. Iki šiol aptarti algoritmai prognozėms pasitarnavo, apmokytam modeliui pateikus porciją naujų duomenų. Algoritmas turi galimybę atlikti kryžminį spėjimą. Apmokius algoritmą dviem skirtingomis, tačiau susijusiomis eilutėmis, galima naudoti gautą modelį, bandant nuspėti vienos eilutės elgesį atsižvelgiant į kitos eilutės elgesį. Laiko sekų algoritmas iš esamų duomenų pritaikius autoregresijos sprendimų medžio principą, gali išvesti tendencijas apie esamų duomenų būsimas reikšmes, kas ir yra prognozė. Laiko sekų algoritmui reikalinga, kad numatomi stulpeliai būtų tolydūs. Kiekvienam modeliui galima tik viena atvejų eilutė [42].

### **3.4. Integruojami duomenų gavybos įrankiai**

#### **3.4.1. Weka**

*Weka (Waikato Environment for Knowledge Analysis)* yra populiarus nemokamas mašininio mokymo programinės įrangos paketas, sukurtas JAVA kalbos pagrindu. Weka paketas buvo sukurtas ir tobulinamas Waikato universitete Naujojoje Zelandijoje. Weka paketą sudaro algoritmų ir vizualizacijos įrankių rinkiniai, skirti duomenų analizei ir prognozavimo modeliavimui. Grafinė vartotojo sąsaja sudaro puikias galimybes tuo naudotis. Algoritmai gali būti naudojami Weka programos aplinkoje arba iškviečiami iš išorinių Java programų. Šiai sistemai sukurta nemažai priedų, kurių pagalba įdiegiami nauji ar patobulinti algoritmai. Gali būti dirbama tiek komandiniame, tiek vaizdiniame režimuose. Weka turi klasterizacijos, klasifikacijos, asociacijų bei skirstymo pagal spėjamus atributus algoritmus. Turi Experimenter vartotojo sąsają, kuri leidžia sistemingai palyginti Weka sistemos duomenų aibę per mašininis apmokymo algoritmus [52].

Palaikomos DBVS: MySQL, PostgreSQL, MS SQL Server, Oracle.

#### **3.4.2. RapidMiner**

*RapidMiner* yra nemokamas mašininio mokymo programinės įrangos paketas, skirtas duomenų išrinkimui, tekstų, prognozavimo analizėms. Jis naudojamas mokslinių tyrimų, švietimo, mokymo, prototipų, programų kūrimo ir pramoniniams taikymams. Produktas yra platinamas pagal AGPL atviro kodo licenciją ir yra prieinamas nuo 2004 m. Sukurtas JAVA programavimo kalbos pagrindu.

Grafinė sąsaja suteikia puikias galimybes išnaudoti produkto privalumus. Keletas svarbesnių paketo savybių:

- Import – platus duomenų užkrovimo pasirinkimas: failai (xml, csv, excel, Access),
- Export - platus pasirinkimas, kokiais formatais eksportuoti duomenis.
- Data transformation – duomenų apdorojimas (tipų keitimas, pridėjimas, šalinimas, filtravimas, rūšiavimas ir t.t.) [53].

#### **3.4.3. Orange**

*Orange* yra nemokamas ir laisvai platinamas duomenų analizės paketas. Komponentai parengti C++ kalba, gali būti laisvai integruojami į kuriamus projektus. Turi septynis analizės ir prognozių modeliavimo algoritmus: sprendimų medžio, Naivo-Bayeso, grupavimo pagal dažniausius atributus, logaritminės regresijos, grupavimo pagal nurodytas taisykles, taip pat kombinuotą iš keleto SVM algoritmą. Orange įrankis turi algoritmo įvertinimo metodus, nustatyti jų panaudojimo tinkamumui. Orange sistemoje yra realizuota daug sistemos mokymo (angl. machine learning) algoritmų, taip pat papildomų įrankių duomenų įvedimui ir apdorojimui. Orange yra sudedamoji dalis iš pagrindinių Python modulių ir vykdo įvairias funkcijas. Tai apima daugelį uždavinių, tokių kaip pasirinkimų medžio atspausdinimas, parametrų aibė ir kt. Orange turi aibę grafinių objektų, kurie naudoja metodus iš pagrindinės bibliotekos ir Orange modulių. Vizualinio programavimo dėka, grafiniai objektai gali būti surinkti į aplikaciją pasinaudojus vizualinio programavimo įrankiu pavadintu Orange Canvas [54].

Orange skirta sistemos mokymui ir duomenų analizei, tinka tiek patyrusiam vartotojui, tiek pradedančiajam vartotojui, kuris nori plėtoti ir testuoti savo paties algoritmus.

Keletas svarbesnių Orange savybių:

- Duomenų įvedimas/išvedimas (Orange gali nuskaityti ir įrašyti tekstą su tarpais failus ir kt. formatus),
- „Iki“ duomenų apdorojimas (galimybė pažymėti aibę, diskretizacija),
- Prognozavimo modeliai (klasifikavimo medis, Naive Bayesian klasifikatorius, kNN (k artimiausių kaimynų), loginė regresija, taisyklėmis grįsti algoritmai (pavyzdžiui CN2)),
- Duomenų aprašymo metodai, įvairios vizualizacijos priemonės, saviorganizuojantys tinklai, hierarchinis klasterizavimas, daugiamačių skalių metodas ir kt.

Tai yra įrankis "Oracle" kūrėjams ir administratoriams ypač SQL ir PL/SQL programų kūrėjams, veiklos vadovams ir duomenų bazių administratoriams [54].

#### **3.4.4. Oracle Data Mining**

*Oracle Data Mining.* Kompanijos Oracle duomenų bazių valdymo sistemos pakete esantis komponentas Oracle Data Mining (ODM) turi nemažai įvairių duomenų gavybos algoritmų. Skirstomi į prižiūrimus (dėsningumo požymį pasirenka duomenų išgavėjas) ir neprižiūrimus apsimokymo algoritmus. Oracle Data Mining palaikomi algoritmai: Naivo-Bayeso, sprendimų medžio, regresijų, keleto algoritmų kombinacija – SVM, du klasterizacijos algoritmai, asociacijų, atributų prastinimo, anomalijų aptikimo, atributų „svorio“ radimo.

ODM (Oracle data mining) leidžia automatinę informacijos išgavimą iš duomenų bazės. Ši technologija apima paslėptų asociacijų tarp skirtingų duomenų atributų suradimą, duomenų klasifikavimą pagal pavyzdžius ir duomenų grupavimą, kad būtų galima sukurti susijusių duomenų vaizdą. Efektyvūs, su Oracle duomenų baze susieti, erdviniai duomenys gali būti materializuoti įtraukimui į duomenų gavybos paraiškas [55].

#### **3.4.5. MiningMart**

*MiningMart* yra atvirojo kodo programa sukurta duomenų bazių MySQL, PostgreSQL arba Oracle naujesnėms versijoms, duomenų išsamiai ir visapusei analizei. Tai yra ypač pažangi realaus laiko analitinių duomenų apdorojimo technologija, kuri suteikia galimybę sprendimą priimančioms asmenims tuojau pat, per kelias sekundes, tik pateikus užklausą, gauti apibendrintus reikiamus duomenis bei gali rasti užslėptas duomenų priklausomybes ir jų išsidėstymo bei pasikartojimo dėsningumus [56].

### **3.5. Biomedicinos samprata**

Kalbant apie biomediciną visų pirma reikia ją apsibrėžti – tai yra tyrimų sritis, apimanti medicinos teorinius aspektus ir biologijos mokslus, nagrinėjančius gyvąsias sistemas, įskaitant anatomiją, fiziologiją, genetiką, patologiją, mikrobiologiją. Jeigu tradicinė medicina yra susijusi su medicinos žinių praktiniu taikymu, tai biomedicina apima ribas ką medicina gali atlikti. Biomedicinos sritis atsirado dėl jos raidos ypatumų ir dėl didėjančios biomedicinos svarbos, gausybės atradimų bei biomedicininių tyrimų ir naujų technologijų [65].

Biomedicininių tyrimų tikslas – pagerinti diagnostikos ir profilaktikos procedūras, geriau suprasti ligos eigą ir patogenezę. Medicinos tobulėjimas pagrįstas tyrimais, kurie privalo remtis eksperimentais, kuriuose žmogus yra tyrimo subjektas. Biomedicininių tyrimų srityje yra dviejų rūšių tyrimai, t.y. medicininiai tyrimai, kurių esmė – pacientų diagnozė ir gydymas ir medicininiai tyrimai, kurių pagrindinis objektas – mokslinis eksperimentas, tiesiogiai nenumatantis diagnostinės ar gydomosios veiklos [64, 65].

Biomedicininiai tyrimai yra jautri sritis, nes čia, skirtingai nei įprastinėje kasdienėje klinikinėje praktikoje, susiduriama su žmogaus kūno, biologinės medžiagos, taip pat privačios sveikatos informacijos panaudojimu ne konkrečiam paciento gydymo, profilaktikos ar slaugos, bet

mokslinio tyrimo tikslais, t. y., ne konkretaus asmens sveikatos labui, o naujo mokslinio žinojimo plėtojimui [63].

### 3.6. Duomenų gavybos panaudojimo atvejai

Galima nuspėti, kad panašių sistemų, skirtų medicininių duomenų apdorojimui, yra sukurta tikrai ne viena, bet detalią informaciją apie jas rasti yra pakankamai sudėtinga, nes pačios sistemos nėra viešai publikuojamos.

V. Špečkauskienė savo daktaro disertacijoje „Informacinio klinikinių sprendimų rengimo metodo sudarymas ir tyrimas,, analizavo duomenų gavybos ir analizės metodų taikymą e. sveikatos sprendimų rengimo sistemose.

Autorės darbo tikslas buvo sudaryti ir iširti informacinį klinikinių sprendimų rengimo metodą e. sveikatos sistemai, skirtą racionaliai panaudoti platų taikomų algoritmų rinkinį evoliucionuojantiems elektroninės sveikatos įrašams bei įvairiarūšių sprendžiamų problemų analizei.

Rengiant disertaciją buvo gauti šie nauji moksliniai rezultatai:

- sukurtas klinikinių sprendimų rengimo metodas, kuriame griežtai apibrėžta veiksmų seka apjungianti daugelį duomenų gavybos ir analizės algoritmų, kurių veikimas vertinamas iteracinių procesų metu;
- metodo iteraciniuose procesuose, atliekančiuose duomenų rinkinio kokybės analizę, įdiegti darbo metu nustatyti kriterijai duomenų rinkinio pertvarkymui taip, kad jis geriausiai tiktų naudojamiems algoritmams, o algoritmai pasiektų aukščiausius rezultatus formuojant klinikinių sprendimų rengimo modelius naudojamus klinikinėje praktikoje;
- metodas skirtas klinikinių sprendimų rengimui tinka ne vienos, o įvairių su mediciniais duomenimis susijusių problemų sprendimui, atranka tinkamiausius algoritmus, įvertina duomenų kokybę ir specifiką, formuoja klinikinių sprendimų rengimo modelius sudarytus iš reikšmingiausių duomenų rinkinio atributų ir yra nepriklausomas nuo analizuojamų klinikinių duomenų [25].

M. Kučinskas savo darbe „Efektyvaus manipuliavimo duomenimis informacinėse medicinos sistemose tyrimas“ analizavo problemas, kurios neigiamai veikia medicinos personalo darbą, nagrinėjo medicinai skirtų informacijos sistemų realizacijos ypatumus, vertino duomenų pasiekiamumo svarbą, pateikė projektinį sprendimą, realizuotą su .NET technologija. Tyrė projektinėje dalyje sukurtos informacinės sistemos duomenų manipuliavimo komponentų efektyvaus naudojimo galimybes. Buvo analizuojama medicinai skirtų informacijos sistemų realizacijos ypatumai, pabrėžiama duomenų pasiekiamumo svarba, vardijamos efektyvaus manipuliavimo duomenimis problemos. Širdies centro kardiologinių tyrimų surinkimo ir analizės programų sistemai realizuoti buvo pasirinkta MS SQL Server duomenų bazių valdymo sistema.

Sukurtas produktas – kardiologinių tyrimų surinkimo bei analizės programų sistema. Tai produktas, skirtas organizacijai, kurios pagrindinė veikla – kardiologinių bei kardiochirurginių duomenų kaupimas bei pakartotinis jų panaudojimas. Šio produkto pagalba registruojami KMU Širdies centre aptarnaujamų pacientų būklės, jiems atliekamų tyrimų, operacijų, statistinei analizei naudojami CARDS duomenys, iš užregistruotų duomenų formuojamos diagnozės, ataskaitos, vykdomi pacientų perkėlimai iš vienos skyriaus kitą, išrašymai namo ar kitą gydymo įstaigą [11].

T. Vileiniškis savo magistro darbe „Intelektualios duomenų gavybos algoritmų taikymo tyrimas Microsoft SQL Server 2005 Data Mining Designer priemonėmis“ analizavo kaip galima duomenų bazėje esantį klientų sąrašą patikrinti ar nėra klaidingų bei dalinai įvestų duomenų. Kaip nuspėti galimus pardavimus ateityje, taipogi surinkti informaciją apie potencialiai „įdomius“ klientus, pritaikant intelektualios duomenų gavybos algoritmus. Tyrimo sritis – intelektualios duomenų gavybos algoritmų taikymo tyrimas Microsoft „Data Mining Designer“ priemonėmis. Darbo metu realizuoti skirtingi Microsoft intelektualios duomenų gavybos algoritmai. Tyrimui buvo naudotas Microsoft SQL Server 2005 Integration Services (SSIS) paketas, turintis visas reikiamas priemones. Duomenų

tyrimui atlikti naudojama Data Source View Designer programa, turinti keletą reikiamų priemonių [39].

Renata Norbutaitė bakalauro darbe „Gerklų ligų diagnostikos informacinė sistema“ sukūrė ir aprašė informacinę sistemą, kuri padėtų gydytojams greičiau ir patogiau registruoti gerklų ligomis sergančius pacientus bei kaupti jų tyrimų duomenis. Šios sistemos prirėikė, nes ligų istorijų rašymas atskiruose dokumentuose buvo ilgas ir painus procesas, o tyrimų failų kaupimas atskirai nuo dokumentų tapdavo komplikuoatas, kai gydytojams reikėdavo surasti tam tikro paciento vizito tyrimų failus, kurie neturi aiškios komplektavimo sistemos. Sukurta sistema padėjo pasiekti numatytus tikslus. Paspartinatas pacientų, sergančių gerklų ligomis, jų vizitų, registravimas. Tyrimų failų kaupimas tapo paprastas ir aiškus, kadangi failai sistemoje sudėti prie atitinkamo paciento vizito. Sistema realizuota dinamine interpretuojama programavimo kalba PHP bei viena iš reliacinių duomenų bazių valdymo sistemų MySQL. Ji taikoma konkreitiems atvejams, t.y. pacientų, sergančių gerklų ligomis, informacijos ir tyrimų rezultatų kaupimui. Ši sistema palengvins darbą gydytojams, kuriems bus paprasčiau priskirti visą tyrimo medžiagą konkrečiam pacientui per konkretų vizitą. Taip pat bus daug paprasčiau tyrėjams, kurie norės rasti tyrimų informaciją apie tam tikrą ligą [36].

<http://sveikas.lt/lt/sveikatos-testas/> interneto svetainėje pasirinkus nusiskundimų grupes galima atlikti sveikatos testą ir operatyviai sužinoti galimą diagnozė [37].

<http://ligosimptomai.lt> - interneto svetainė padedanti rasti galimas ligas pagal jų simptomus ir požymius. Naudotis ja paprasta – simptomų sąrašė radus tinkamą, pasirinkti, tokiu būdu simptomas patenka "Mano simptomų" sąrašą ir sistema pradeda ieškoti labiausiai tinkančių ligų [38].

Informacinė sistema „Medialog“ – sistema, kuri automatizuoja ligonio ligos istorijos įvedimą, statistikos vedimą, gydymo proceso planavimą. Tai yra galinga sistema, padedanti padidinti gydytojo darbo efektyvumą. Gydytojas gali peržiūrėti laboratorinių tyrimų rezultatus, o tai leidžia priimti sprendimus dėl tolimesnio gydymo, taip pat rezultatai gali būti peržiūrėti kitų gydytojų [40].

### **3.7. Kitų autorių atlikti DBVS palyginimai**

Slovėnų autorius Uroš Maleš 2010 m. savo darbe „Benchmarking of various relational database management systems“ atliko lyginamąją DBVS analizę.

Autoriaus darbo tikslas buvo palyginti Oracle, MS SQL, IBM DB2, MySQL ir PostgreSQL duomenų bazių valdymo sistemas ir gautais rezultatais nustatyti, kurios DBVS yra geresnės tenkinant tam tikrus poreikius.

DBVS testavimas buvo atliekamas dešimtimis kartų, palyginimai parodė duomenų bazių valdymo sistemų našumą su numatyta konfigūracija. Autorius išsiaiškino kad užklausas greičiausiai vykdo Oracle ir PostgreSQL DBVS. Testuodamas atviro kodo duomenų bazių valdymo sistemas MySQL ir PostgreSQL, autorius padarė išvadą, kad PostgreSQL veikia patikimiau negu MySQL DBVS.

Stabilumui užtikrinti ir gebėjimui atlikti sudėtingas užklausas autorius rekomenduoja naudoti PostgreSQL duomenų bazių valdymo sistemą. Autorius siūlo prieš pradėdant naudoti vieną ar kitą DBVS, pirmiausia atlikti testavimą ir pasirinkti tinkamiausią darbui duomenų bazių valdymo sistemą [41].

Vytautas Jankevičius magistro darbe „Atvirojo kodo duomenų bazių serverių analizė aptarnavimo paslaugų teikimo valdymo sistemos pagrindu“ siekė ištirti populiariausius atvirojo kodo duomenų bazės serverius ir nustatyti, kuris padės aptarnavimo paslaugų teikimo valdymo sistemai dirbti greičiau. Šiuo darbu apžvelgiamos panašios sistemos, jų funkcijos, pasirinkti įrankiai, po to jie lyginami. Ypatingas dėmesys atkreipiamas į duomenų bazių serverius, kadangi tokiose sistemose reikia greito duomenų pateikimo. Autoriaus straipsnyje nagrinėjami trys atvirojo kodo duomenų bazių serveriai: MySQL, PostgreSQL ir Firebird. Šios sistemos buvo testuojamos pagal tam tikrus kriterijus. Atkreiptas dėmesys į architektūrą, valdymą ir transakcijas.

Atlikus analizę autorius mano, kad pagrindiniai sistemos vertinimo kriterijai yra vartotojo sąsaja, vartotojo patogumas bei sistemos greitis. Jis teigia, kad visi trys aptarti duomenų bazių serveriai yra labai panašūs skiriasi tik kelios funkcijos arba jos vadinamos skirtingais vardais, nors atlieka tuos pačius veiksmus. MySQL ir PostgreSQL duomenų bazių serveriai turi spartinančios atminties galimybes, Firebird tokios galimybės neturi. Vertinant duomenų bazių serverius pagal paieškos įvykdymo kriterijų lėčiausias iš nagrinėjamų duomenų bazių serverių yra PostgreSQL. Jei pagrindinis kriterijus paieška pagal visą lauką - greičiausias MySQL, jei kriterijus paieška pagal kontekstą – greičiausias Firebird [8].

Andrius Laukaitis magistro darbe „XML dokumentų palaikymas laisvo kodo duomenų bazių valdymo sistemose“ tyrė ir lygino duomenų bazių valdymo sistemas: Microsoft SQL, Oracle, IBM DB2, ar yra galima duomenų gavyba per XML dokumentus, kuriose platformose DBVS veikia, kokias funkcijas palaiko. Analizės metu nustatyta, kad laisvai platinamose duomenų bazių valdymo sistemose, tokiose kaip MySQL, PostgreSQL nėra šios galimybės. Tyrimo metu patikrinta kaip greitai šios duomenų bazių valdymo sistemos sąveikauja su savo gamintojų sukurtais sujungimo ryšiais [51].

### 3.8. Analitinės dalies išvados

Analizuojant literatūrą buvo rasti kai kurie DBVS tyrimai ir palyginimai, tačiau dauguma jų apsiriboja tik keletu DBVS ir yra bendro pobūdžio. Kai kuriuose DBVS tyrimuose analizė atlikta kiek platesnė, tačiau tyrimai buvo atlikti jau gerokai anksčiau. Šiuo metu naujausios DBVS versijos kai kuriais aspektais yra pasikeitusios ir tobulesnės, todėl atliktos ankstesnės analizės jau šiek tiek pasenusios [8, 9, 60].

Renkantis DBVS biomedicininį duomenų saugojimui ir apdorojimui svarbi duomenų tipų įvairovė ir galimybė naudotis specializuotais duomenų gavybos įrankiais [39, 30]. Nagrinėjant atliktus DBVS tyrimus buvo pasigesta išsamesnių tyrimų apie programavimo kalbų, duomenų tipų ir SQL standartų palaikymą. Taip pat mažai kalbama apie DBVS naudojamus duomenų gavybos įrankius.

### 3.9. Sistemos projektavimas

#### 3.9.1. Projekto kūrimo pagrindas

Sistemos projektavimo pagrindiniai aspektai: pateikti išsamų kuriamos sistemos vaizdą pateikti schemas, kaip funkcionuos kuriama sistema, schematiškai aprašyti duomenų gavybos algoritmą, kuris analizuos įvestus paciento duomenis ir atrinks galimų ligų sąrašą. Ligos paieškai aprašyti, simptomams ir tyrimams charakterizuoti pateikta ontologija.

#### 3.9.2. Projekto dalyviai

##### Vartotojai

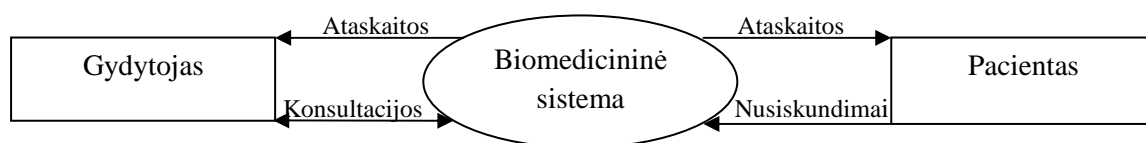
Sistemos vartotojas – gydytojas, turintis pakankamai gebėjimų išmokti dirbti su sistema.

3.1 lentelė. Vartotojas

Vartotojo kategorija	Gydytojas
Vartotojo sprendžiami uždaviniai:	Sistemos įrašų, duomenų pildymas, redagavimas, šalinimas, peržiūra. Peržiūrimos ir atspausdinamos ataskaitos. Duomenų bazės kopijų darymas.
Patirtis dalykinėje srityje:	Įprastas darbuotojas
Patirtis informacinėse technologijose:	Naudojimosi IT pagrindai
Papildomos vartotojo charakteristikos:	Turi mokėti paleisti programą
Apsimokymo poreikis:	Reikalingas apmokymas dirbti DBVS
Amžiaus grupė:	20-65

### 3.9.3. Funkciniai modelio reikalavimai

#### Veiklos kontekstas

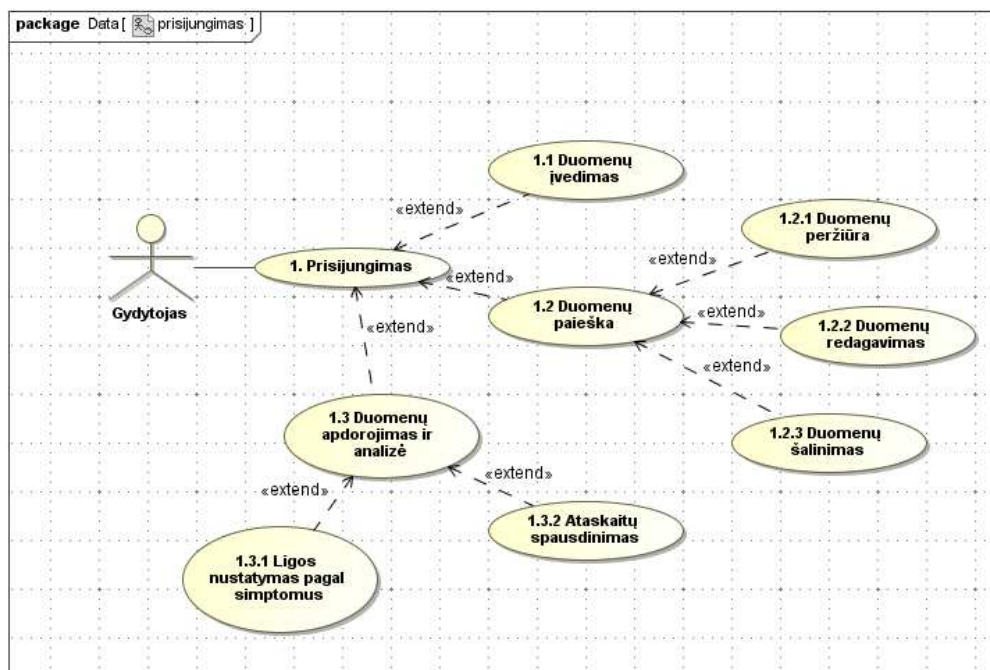


3.1 pav. Aukščiausio lygio duomenų srautų diagrama

### 3.9.4. Produkto veiklos sfera

#### Sistemos ribos

Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama:



3.2 pav. Panaudojimo atvejų diagrama

#### Panaudojimo atvejų sąrašas

3.2 lentelė. Panaudojimo atvejo „Prisijungimas“ aprašymas

Pavadinimas	Prisijungimas	Numeris	1
Aprašymas	Prisijungti prie sistemos		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi žinoti prisijungimo duomenis		
Vykdymo sąlyga	Sistema patvirtina prisijungimą		
Po sąlyga	Gydytojas gali dirbti su sistema		

3.3 lentelė. Panaudojimo atvejo „Duomenų įvedimas“ aprašymas

Pavadinimas	Duomenų įvedimas	Numeris	1.1
Aprašymas	Gydytojas suveda naujus duomenis apie pacientą		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos		
Vykdymo sąlyga	Gydytojas, prisijungęs prie sistemos įveda paciento vardą ir pavardę, paciento adresą, telefoną ir asmens kodą. Įveda nusiskundimus dėl sveikatos, tyrimo rezultatus.		
Po sąlyga	Duomenys išsaugomi sistemoje.		

3.4 lentelė. Panaudojimo atvejo „Duomenų paieška“ aprašymas

Pavadinimas	Duomenų paieška	Numeris	1.2
Aprašymas	Gydytojas pagal ligos simptomus atrenka duomenis		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atlieka duomenų paiešką sistemoje		
Po sąlyga	Sistema pateikia paieškos duomenis pagal nurodytus kriterijus		

3.5 lentelė. Panaudojimo atvejo „Duomenų peržiūra“ aprašymas

Pavadinimas	Duomenų peržiūra	Numeris	1.2.1
Aprašymas	Gydytojas pagal pasirinktus kriterijus atlieka surastų duomenų peržiūrą		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atlieka duomenų paiešką sistemoje		
Po sąlyga	Sistema pateikia paieškos duomenis pagal nurodytus kriterijus		

3.6 lentelė. Panaudojimo atvejo „Duomenų redagavimas“ aprašymas

Pavadinimas	Duomenų redagavimas	Numeris	1.2.2
Aprašymas	Gydytojas pagal pasirinktus kriterijus atlieka surastų duomenų peržiūrą, duomenis redaguoja		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atlieka duomenų paiešką sistemoje		
Po sąlyga	Sistema pateikia paieškos duomenis pagal nurodytus kriterijus		

3.7 lentelė. Panaudojimo atvejo „Duomenų šalinimas“ aprašymas

Pavadinimas	Duomenų šalinimas	Numeris	1.2.3
Aprašymas	Gydytojas pagal pasirinktus kriterijus atlieka surastų duomenų peržiūrą, pasenusius duomenis pašalina		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atlieka duomenų paiešką sistemoje		
Po sąlyga	Sistema pateikia paieškos duomenis pagal nurodytus kriterijus		

3.8 lentelė. Panaudojimo atvejo „Duomenų apdorojimas ir analizė“ aprašymas

Pavadinimas	Duomenų apdorojimas ir analizė	Numeris	1.3
Aprašymas	Gydytojas pagal ligos simptomus apdoroja duomenis		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje.		
Vykdymo sąlyga	Gydytojas atlieka duomenų paiešką sistemoje		
Po sąlyga	Sistema pateikia paieškos duomenis pagal nurodytus kriterijus		

3.9 lentelė. Panaudojimo atvejo „Ligos nustatymas pagal simptomus“ aprašymas

Pavadinimas	Ligos nustatymas pagal simptomus	Numeris	1.3.1
Aprašymas	Gydytojas nustato ligą pagal simptomus		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir paciento duomenys ir nusiskundimai apie sveikatą turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atspausdina ataskaitą apie paciento ligą.		
Po sąlyga	Ataskaita atspausdinta ir įsepta į paciento kortelę.		

3.10 lentelė. Panaudojimo atvejo „Ataskaitų spausdinimas“ aprašymas

Pavadinimas	Ataskaitų spausdinimas	Numeris	1.3.2
Aprašymas	Gydytojas atspausdina ataskaitas		
Aktorius	Gydytojas		
Prieš sąlyga	Gydytojas turi būti prisijungęs prie sistemos ir duomenys turi būti suvesti sistemoje		
Vykdymo sąlyga	Gydytojas atspausdina ataskaitą apie paciento ligą.		
Po sąlyga	Ataskaita atspausdinta ir įsepta į paciento kortelę.		



### 3.9.5. Funkciniai reikalavimai

3.11 lentelė. Funkcinis reikalavimas „Duomenų įvedimas“

Reikalavimas #:	1			
Aprašymas:	Įveda naujus duomenis			
Pagrindimas:	Duomenų apie pacientą kaupimas			
Šaltinis:	Gydytojas			
Tikimo kriterijus:	Užpildyti duomenų laukelius			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	1	Konfliktai:	Nėra	

3.12 lentelė. Funkcinis reikalavimas „Duomenų redagavimas“

Reikalavimas #:	2			
Aprašymas:	Redaguoja ir papildo duomenis apie pacientą			
Pagrindimas:	Suvesti papildomi duomenys			
Šaltinis:	Gydytojas			
Tikimo kriterijus:	Reikia papildyti paciento duomenis			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	1	Konfliktai:	Nėra	

3.13 lentelė. Funkcinis reikalavimas „Duomenų šalinimas“

Reikalavimas #:	3			
Aprašymas:	Pašalina duomenis			
Pagrindimas:	Pašalinami pasenę duomenys			
Šaltinis:	Gydytojas			
Tikimo kriterijus:	Reikia pašalinti pasenusius ir nebereikalingus duomenis			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	1	Konfliktai:	Nėra	

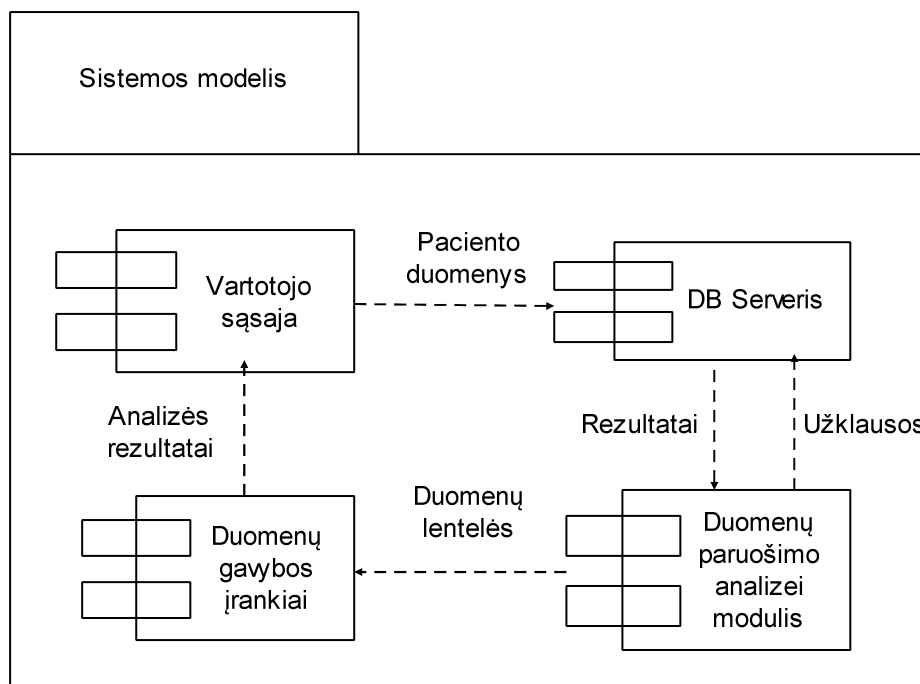
3.14 lentelė. Funkcinis reikalavimas „Duomenų atrinkimas“

Reikalavimas #:	4			
Aprašymas:	Atrenka duomenis			
Pagrindimas:	Pagal paciento nusiskundimus sistema atrenka ligą			
Šaltinis:	Gydytojas			
Tikimo kriterijus:	Reikia atrinkti ligą pagal tinkančius simptomus			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	1	Konfliktai:	Nėra	

3.15 lentelė. Funkcinis reikalavimas „Ataskaitų spausdinimas“

Reikalavimas #:	5			
Aprašymas:	Atspausdina ataskaitas			
Pagrindimas:	Paciento ligos istorijos saugojimas			
Šaltinis:	Gydytojas			
Tikimo kriterijus:	Atspausdinta ataskaita			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	1	Konfliktai:	Nėra	

### 3.9.6. Sistemos komponentai



3.3 pav. Apibendrinta sistemos modelio komponentų diagrama

Sistemą sudaro tokie komponentai:

- vartotojo sąsaja – duomenų įvedimui. Vartotojo sąsaja gali būti realizuojama įvairiomis priemonėmis: programavimo kalba Java, C++, per interneto naršyklę su PHP, ASP ir pan. arba standartinė DBVS.
- DB serveris – kompiuteris, kuriame saugoma duomenų bazė, programinė įranga.
- duomenų paruošimo analizei modulis – papildomos lentelės, kurios sudaromos duomenų bazėje SQL kalba formuojant atitinkamas užklauskas ir pateikiamos DG įrankiui. Duomenų gavybos įrankiui nėra reikalinga pateikti visus esamus saugykloje duomenis todėl tuo atveju yra formuojamos užklauskos ir jų pagalba parengiamos papildomos lentelės, kurios pateikiamos tolesnei DG analizei.
- duomenų gavybos įrankis skirtas apdoroti suformuotas ir pateiktas papildomas lenteles ir pateikia duomenų klasterinės analizės rezultatus.

### 3.9.7. Duomenų bazės schema ir lentelių specifikacijos

Duomenų bazės projektavimui buvo pasirinktas sąryšių realizavimo modelis. Toks modelis yra vienas iš labiausiai naudojamų, kadangi lentelės tarpusavyje gali būti susietos naudojant unikalius laukus. Tai yra labai svarbus aspektas kadangi elementai tarpusavyje yra labai glaudžiai susiję.

Projektuojant duomenų bazę bus naudojami tik sisteminiai duomenų tipai, t. y. bus apsiribota tik duomenų bazių valdymo sistemos siūlomais standartiniais duomenų tipais. Duomenų bazės modelio schemoje vaizduojamos visos projektuojamos duomenų bazės lentelės su visais egzistuojančiais atributais bei ryšiais.



3.16 lentelė. Lentelės „Pacientas“ atributų aprašymas

<b>Pacientas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
paciento_id	Integer	Unikalus kodas identifikuojantis pacientą sistemoje
vardas	VARCHAR (15)	Paciento vardas
pavarde	VARCHAR (15)	Paciento pavardė
lytis	VARCHAR (10)	Paciento lytis
gimimo_data	DATE	Paciento gimimo data
adresas	VARCHAR (15)	Paciento gyvenamoji vieta
miestas	VARCHAR (15)	Informacija apie paciento gyvenamąją vietovę.

3.17 lentelė. Lentelės „Gydytojas“ atributų aprašymas

<b>Gydytojas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
gydytojo_id	VARCHAR (8)	Unikalus kodas identifikuojantis gydytoją sistemoje
vardas	VARCHAR (15)	Gydytojo vardas
pavarde	VARCHAR (15)	Gydytojo pavardė
kvalifikacija	VARCHAR (15)	Gydytojo kvalifikacija

3.18 lentelė. Lentelės „Nusiskundimas“ atributų aprašymas

<b>Nusiskundimas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
paciento_id	INTEGER	Unikalus kodas identifikuojantis pacientą sistemoje
pavadinimas	VARCHAR (30)	Nusiskundimo pavadinimas
aprasymas	CLOB	Nusiskundimo aprašymas
simpt_nusisk_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis Simptomo_Nusiskundimo tipą
kuno_dalis_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis kūno dalį

3.19 lentelė. Lentelės „Simpt\_Nusisk\_tipas“ atributų aprašymas

<b>Simpt_Nusisk_tipas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
simpt_nusisk_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis Simptomo_Nusiskundimo tipą sistemoje
pavadinimas	VARCHAR (30)	Simptomo/Nusiskundimo tipo pavadinimas
aprasymas	CLOB	Simptomo/Nusiskundimo tipo aprašymas

3.20 lentelė. Lentelės „Sim\_lig“ atributų aprašymas

<b>Sim_lig</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
Sim_lig_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis ligos simptomą sistemoje
pavadinimas	VARCHAR (30)	Ligos simptomo pavadinimas
aprasymas	CLOB	Ligos simptomo aprašymas
simpt_nusisk_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis Simptomo_Nusiskundimo tipą sistemoje
ligos_id	VARCHAR (8)	Unikalus kodas sistemoje identifikuojantis ligą sistemoje
kuno_dalis_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis kūno dalį sistemoje

3.21 lentelė. Lentelės „Veik\_lig“ atributų aprašymas

<b>Veik_lig</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
veik_lig_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis ligos veiksnį sistemoje
pavadinimas	VARCHAR (30)	Ligos veiksnio pavadinimas
aprasymas	CLOB	Ligos veiksnio aprašymas
veiksnys_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis veiksnį sistemoje
ligos_id	VARCHAR (8)	Unikalus kodas sistemoje identifikuojantis ligą sistemoje

3.22 lentelė. Lentelės „Veik\_pac“ atributų aprašymas

<b>Veik_pac</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
pavadinimas	VARCHAR (30)	Ligos veiksnio pavadinimas
aprasymas	CLOB	Ligos veiksnio aprašymas
paciento_id	INTEGER	Unikalus kodas identifikuojantis pacientą sistemoje
veiksnys_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis veiksnį sistemoje

3.23 lentelė. Lentelės „Veiksny“ atributų aprašymas

<b>Veiksny</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
veiksny_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis veiksnį sistemoje
pavadinimas	VARCHAR (30)	Ligos veiksnio pavadinimas
aprasymas	CLOB	Ligos veiksnio aprašymas

3.24 lentelė. Lentelės „Tyr\_lig“ atributų aprašymas

<b>Tyr_lig</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
tyr_lig_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis ligos tyrimą sistemoje
pavadinimas	VARCHAR (30)	Ligos veiksnio pavadinimas
aprasymas	CLOB	Ligos veiksnio aprašymas
ligos_id	VARCHAR (8)	Unikalus kodas sistemoje identifikuojantis ligą sistemoje
tyrimo_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimą sistemoje
organo_id	VARCHAR (6)	Unikalus kodas identifikuojantis organą sistemoje

3.25 lentelė. Lentelės „Kūno\_dalis“ atributų aprašymas

<b>Kūno_dalis</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
kūno_dalis_id	VARCHAR (6)	Unikalus kodas sistemoje identifikuojantis kūno dalį sistemoje
pavadinimas	VARCHAR (30)	Kūno dalies pavadinimas
aprasymas	CLOB	Kūno dalies aprašymas

3.26 lentelė. Lentelės „Organizmo\_sistema“ atributų aprašymas

<b>Organizmo_sistema</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
organizmo_id	VARCHAR (6)	Unikalus kodas identifikuojantis organizmo sistemą sistemoje
pavadinimas	VARCHAR (30)	Organizmo sistemos pavadinimas
aprasymas	CLOB	Organizmo sistemos aprašymas

3.27 lentelė. Lentelės „Organas“ atributų aprašymas

<b>Organas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
organo_id	VARCHAR (6)	Unikalus kodas identifikuojantis organą sistemoje
pavadinimas	VARCHAR (30)	Organo pavadinimas
aprasymas	CLOB	Organo aprašymas
organizmo_id	VARCHAR (6)	Unikalus kodas identifikuojantis organizmo sistemą sistemoje

3.28 lentelė. Lentelės „Liga“ atributų aprašymas

<b>Liga</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
ligos_id	VARCHAR (8)	Unikalus kodas identifikuojantis ligą sistemoje
pavadinimas	VARCHAR (30)	Ligos pavadinimas
aprasymas	CLOB	Ligos aprašymas

3.29 lentelė. Lentelės „Tyrimo\_tipas“ atributų aprašymas

<b>Tyrimo_tipas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
tyrimo_tipas_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimo tipą sistemoje
pavadinimas	VARCHAR (30)	Tyrimo tipo pavadinimas
aprasymas	CLOB	Tyrimo aprašymas

3.30 lentelė. Lentelės „Tyrimas“ atributų aprašymas

<b>Tyrimas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
tyrimo_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimą sistemoje
pavadinimas	VARCHAR (30)	Tyrimo pavadinimas
tyrimo_tipas_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimo tipą sistemoje

3.31 lentelė. Lentelės „Tyrimo\_duomenys“ atributų aprašymas

<b>Tyrimo_duomenys</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
tyrimo_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimą sistemoje
data/laikas	DATE	Data/laikas kada turi būti atliekamas tyrimas
nuo	VARCHAR (6)	Tyrimo duomenų reikšmė (intervalas)
iki	VARCHAR (6)	Tyrimo duomenų reikšmė (intervalas)
reiksme	VARCHAR (6)	Tyrimo duomenų reikšmė (skaičius)
matavimo_vienetai	VARCHAR (8)	Matavimo vienetai

3.32 lentelė. Lentelės „Atliktas\_tyrimas“ atributų aprašymas

<b>Atliktas_tyrimas</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
a_tyrimo_id	VARCHAR (6)	Unikalus kodas identifikuojantis pacientui atliktą tyrimą sistemoje
pavadinimas	VARCHAR (30)	Atlikto tyrimo pavadinimas
aprasymas	CLOB	Atlikto tyrimo aprašymas
tyrimo_tipas_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimo tipą sistemoje
paciento_id	INTEGER	Unikalus kodas identifikuojantis pacientą sistemoje

3.33 lentelė. Lentelės „Atlikto\_tyrimo\_duomenys“ atributų aprašymas

<b>Atlikto_tyrimo_duomenys</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
tyrimo_id	VARCHAR (6)	Unikalus kodas identifikuojantis tyrimą sistemoje
data/laikas	DATE	Data/laikas kada buvo atliktas tyrimas
nuo	VARCHAR (6)	Tyrimo duomenų reikšmė (intervalas)
iki	VARCHAR (6)	Tyrimo duomenų reikšmė (intervalas)
reiksme	VARCHAR (6)	Tyrimo duomenų reikšmė (skaičius)
matavimo_vienetai	VARCHAR (8)	Matavimo vienetai
paciento_id	INTEGER	Unikalus kodas identifikuojantis pacientą sistemoje

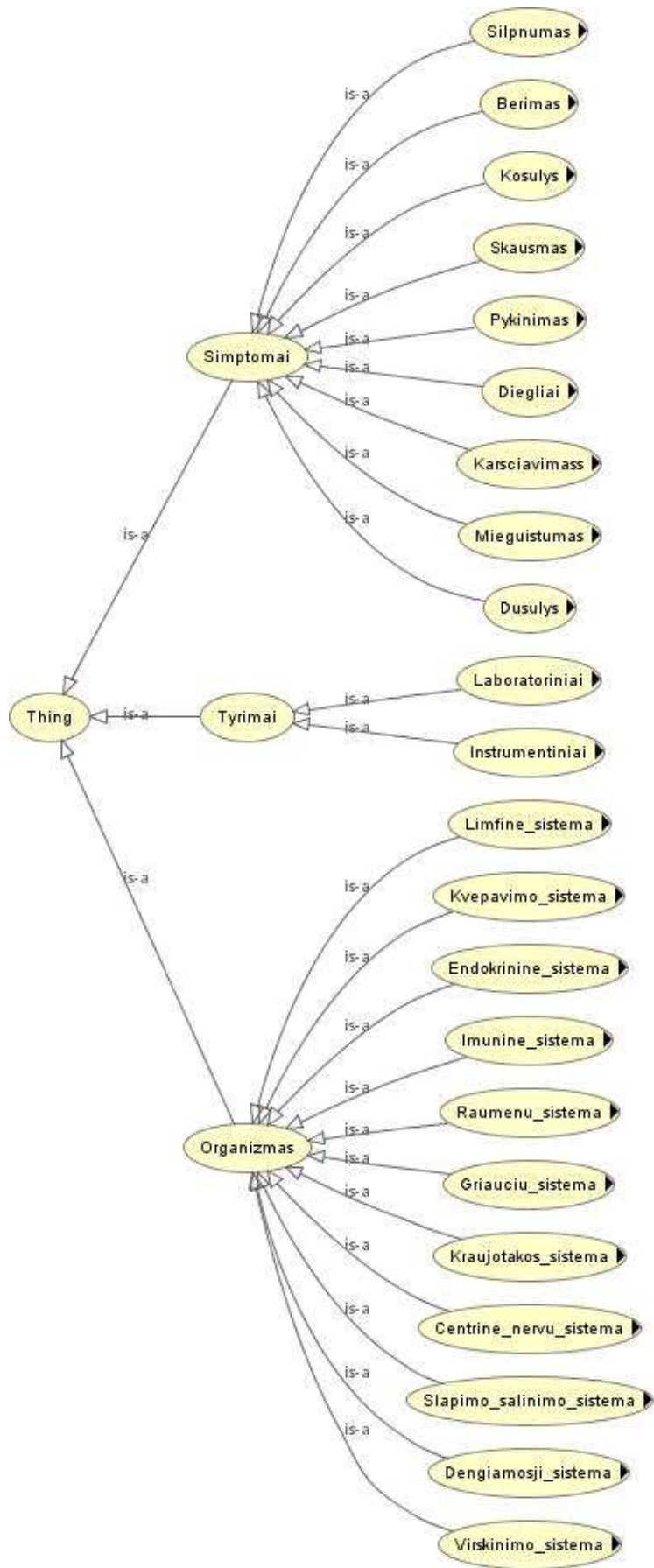
3.34 lentelė. Lentelės „Diagnozė“ atributų aprašymas

<b>Diagnozė</b>		
<b>Atributas</b>	<b>Tipas</b>	<b>Aprašymas</b>
gydytojo_id	VARCHAR (8)	Unikalus kodas identifikuojantis gydytoją sistemoje
paciento_id	INTEGER	Unikalus kodas identifikuojantis pacientą sistemoje
data	DATE	Ligos diagnozės nustatymo data
ligos_id	VARCHAR (8)	Unikalus kodas identifikuojantis ligą sistemoje
tikimybe	VARCHAR (15)	Ligos buvimo/nebuvimo tikimybė

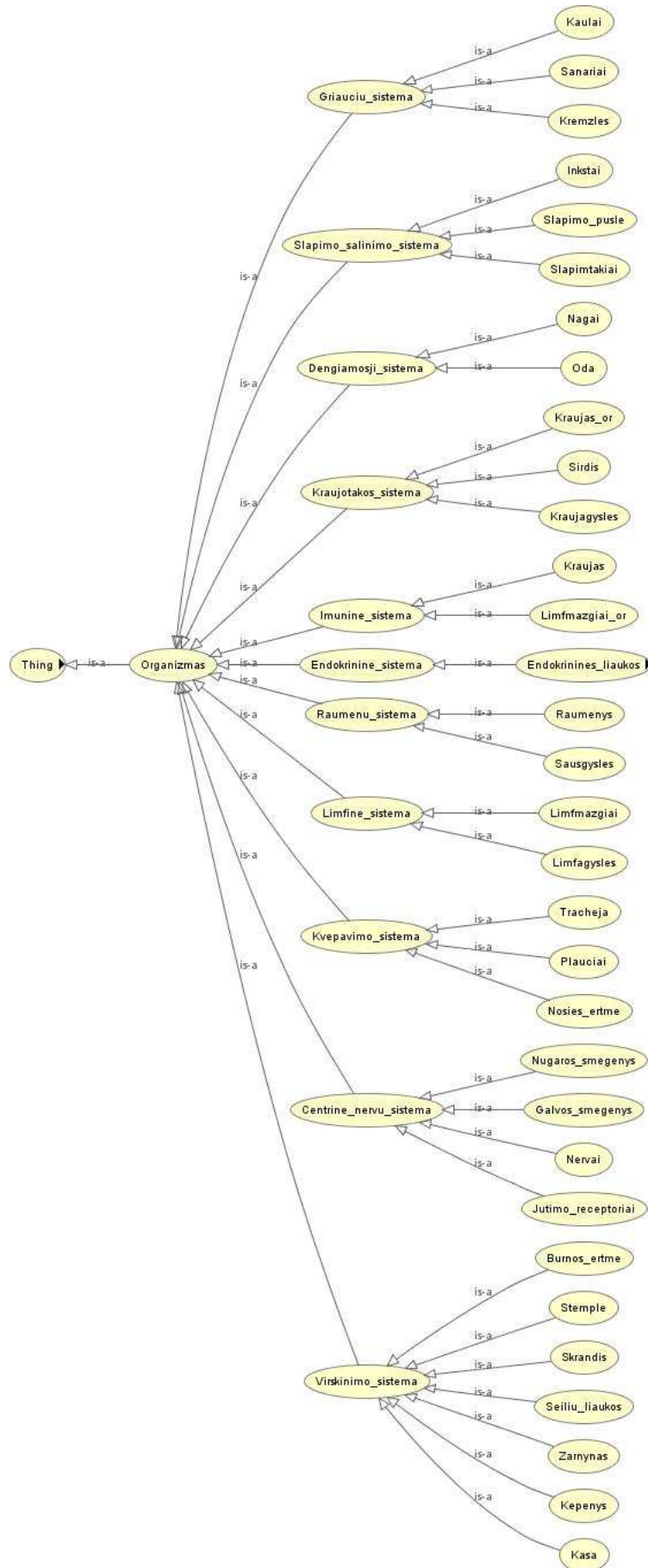
### 3.9.8. Ontologija

Kad būtų lengviau suprasti, ligos paieškai aprašyti, simptomams ir tyrimams charakterizuoti pateikta ontologija. Ontologiją sudaro trys stambios grupės: organizmas, simptomai ir tyrimai, tai yra esminiai dalykai diagnozuojant ligą. Organizmas, simptomai ir tyrimai yra suklasifikuoti ir sugrupuoti.

Ontologijoje įvardintos stambesnės ligų grupės, ligai būdingi nusiskundimai, simptomai. Įvertinus visus šiuos dalykus sudaryta ontologija, kuri paaiškina, kaip turint nusiskundimus ir tyrimo rezultatus būtų ieškoma liga. Pradžioje renkamasi stambesnes grupes ir leidžiamasi simptomų ir tyrimų atšakomis, kol jau galima pasirinkti konkrečius duomenis. Ontologija buvo suprojektuota naudojant *Protege* programinę įrangą.



3.5 pav. Bendras ontologijos vaizdas[38]

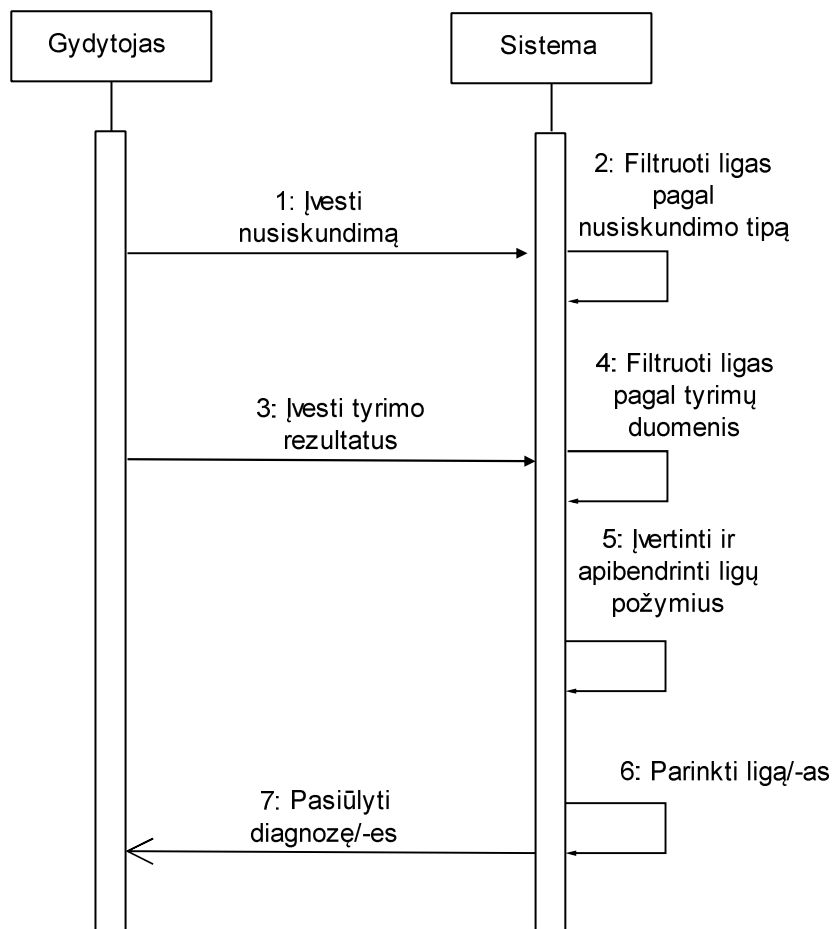


3.6 pav. Organizmo ontologijos schema[38]



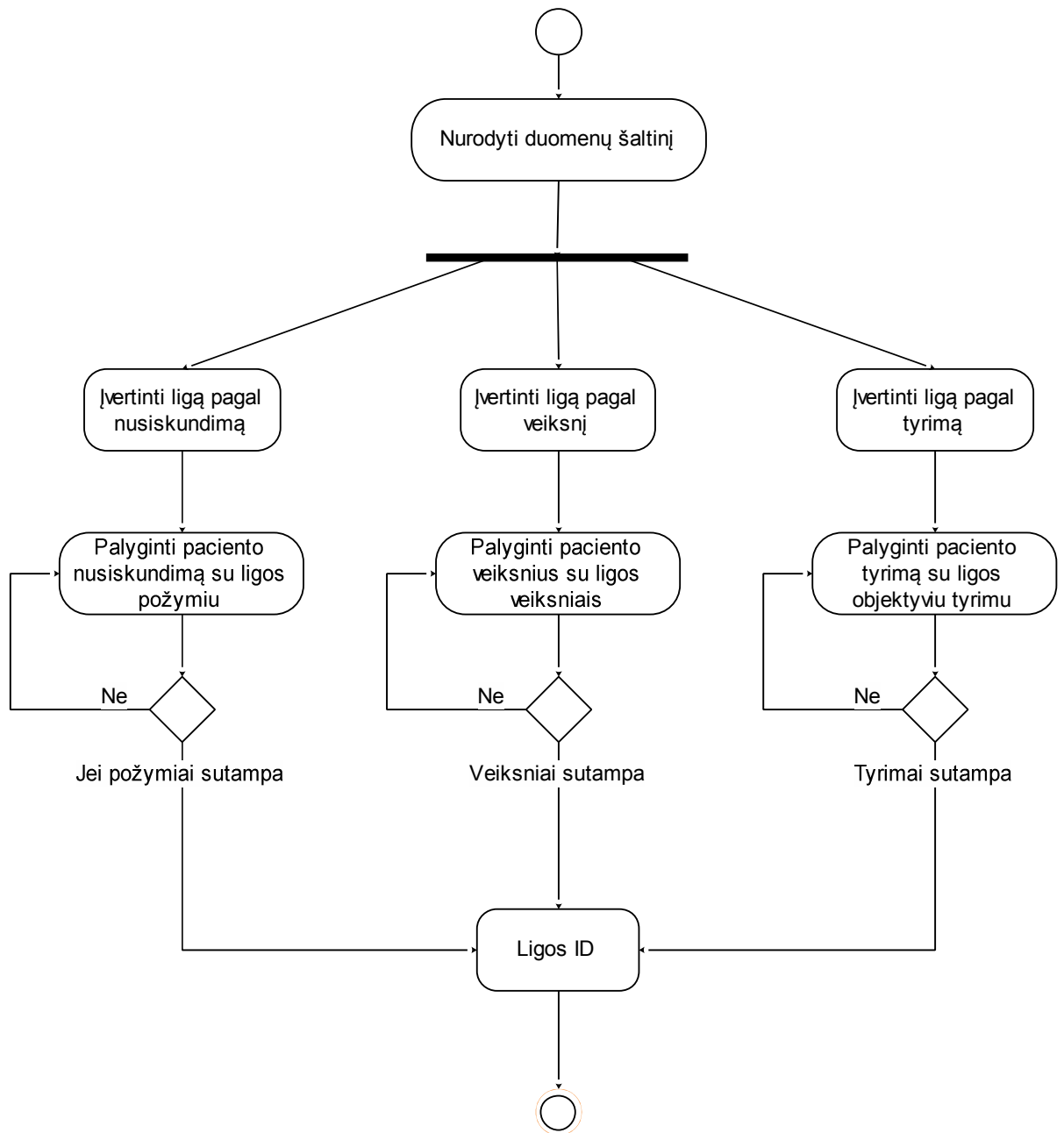
### 3.9.9. Veiklos procesų modelis

Veiklos procesų modelio schemoje 3.7 paveiksle pavaizduota kokie procesai vyksta tarp gydytojo ir sistemos. Pagal nusiskundimus ir tyrimo rezultatus sistema atrenka požymius būdingus konkrečiai ligai. Konkretaus paciento nusiskundimai/simptomai ir tyrimų duomenys lyginami su būdingais atrinktų ligų požymiais. Skaičiuojamas atitikmens įvertis. Pagal suskaičiuotus įverčius, sistema pasiūlo galima/-as ligą/-as ir diagnozę/-es. Tam iliustruoti, pateikiamos UML veiklos diagramos.



3.7 pav. Veiklos procesų modelio schema

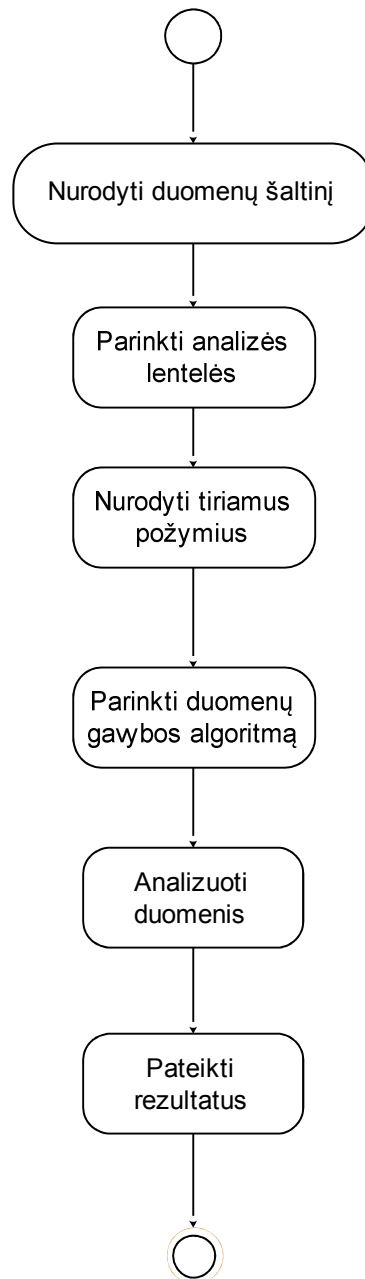
3.8 paveiksle atvaizduota veiksmų sekos schema, charakterizuojanti ligos radimo algoritmą. Iš duomenų šaltinio atrenkami reikiami duomenys. Kadangi ligos diagnozavimo procesas priklauso nuo daugelio dalykų, tuo eiliškumu ir atvaizduota schema. Pirmiausia ligos simptomai įvertinami pagal paciento nusiskundimus – pastarieji lyginami su objektyviais ligos simptomais ir požymiais, jeigu požymiai sutampa, tuomet gaunamas ligos ID, jeigu nesutampa, tuomet toliau ieškoma atitikimų. Analogiškas veiksmas atliekamas su veiksniais – paciento veiksniai yra lyginami su veiksniais būdingais tam tikrai ligai. Veiksniai gali būti tokie kaip lytis, mityba, žalingi įpročiai ir pan. Paskutiniu etapu yra palyginami tyrimai, t.y. paciento turimi atlikti tyrimai su ligos objektyviais tyrimais, reikšmėmis, tyrimų normatyvais. Palyginimo būdu gaunamas ligos ID pagal veiksnius, nusiskundimus ir tyrimus.



3.8 pav. Veiksmų sekos schema, charakterizuojanti ligos radimo procesą

### 3.9.10. Duomenų analizės/apdorojimo schema

Žemiau pateikta veiksmų sekos schema (žr. 3.9 pav.), charakterizuojanti tolesnius duomenų analizės veiksmus. Duomenų bazėje suformavus užklausą sukuriama papildoma lentelė, kurioje atrenkami duomenys tolesnei analizei. Įvykdomas duomenų gavybos įrankio prisijungimas prie duomenų bazės, kurioje yra sukaupti reikalingi duomenys. Nurodomas duomenų šaltinis. Tuomet parenkamos reikiamos lentelės ir importuojamos/perduodamos duomenų gavybos įrankiui. Nurodomi tiriami požymiai, pasirenkamas naudoti analizei duomenų gavybos algoritmas. Duomenų gavybos įrankiui apdorojus ir pateikus rezultatus vykdoma rezultatų analizė.



3.9 pav. Veiksmų sekos schema, charakterizuojanti duomenų analizės veikimą

### 3.9.11. Nefunkciniai reikalavimai

#### Reikalavimai sistemos išvaizdai

3.35 lentelė. Nefunkcinis reikalavimas „Vartotojo sąsaja“

<b>Reikalavimas #:</b>	<b>1</b>			
Aprašymas:	Lengvai suprantama vartotojo sąsaja			
Pagrindimas:	Suprantamas naudojimas, lengvai įskaitoma.			
Šaltinis:	Užsakovas			
Tikimo kriterijus:	Vartotojo sąsajoje bus aiškūs programos komponentai ir galimi veiksmai su jais			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:		2
Priklausomybės:	Nėra	Konfliktai:	Nėra	

### **Reikalavimai panaudojamumui**

3.36 lentelė. Nefunkcinis reikalavimas „Sistemos įsisavinimas“

<b>Reikalavimas #:</b>	<b>1</b>		
Aprašymas:	Reikalingas vartotojo apmokymas		
Pagrindimas:	Kad išmokti naudotis sistema reikia praeiti mokymo kursą		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Reikalingi apmokymai darbui su sistema		
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	3
Priklausomybės:	Nėra	Konfliktai:	Nėra

3.37 lentelė. Nefunkcinis reikalavimas „Lengvai prieinama“

<b>Reikalavimas #:</b>	<b>2</b>		
Aprašymas:	Sistema lengvai prieinama		
Pagrindimas:	Kadangi programą reikės greitai įjungti ir dirbti su ja, ji turėtų būti lengvai prieinama		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Jungiantis prie programos reikės įvesti vartotojo vardą ir slaptažodį		
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	2
Priklausomybės:	Nėra	Konfliktai:	Nėra

### **Reikalavimai vykdymo charakteristikoms**

3.38 lentelė. Nefunkcinis reikalavimas „Duomenų įvedimas“

<b>Reikalavimas #:</b>	<b>3</b>		
Aprašymas:	Sistema lengvai prieinama		
Pagrindimas:	Kadangi programą reikės greitai įjungti ir dirbti su ja, ji turėtų būti lengvai prieinama		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Jungiantis prie programos reikės įvesti vartotojo vardą ir slaptažodį		
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	2
Priklausomybės:	Nėra	Konfliktai:	Nėra

### **Reikalavimai saugumui**

Programa naudoja daug informacijos apie vartotoją, todėl sistemos saugumas turi būti užtikrintas.

3.39 lentelė. Reikalavimas saugumui „Duomenų apsauga nuo neteisėtos veiklos“

<b>Reikalavimas #:</b>	<b>1</b>		
Aprašymas:	Duomenų apsauga nuo neteisėtos veiklos		
Pagrindimas:	Pacientų duomenys yra konfidencialūs, todėl turi būti apsaugotas priėjimas sistemos funkcijų, kuriomis atitinkami vartotojai neturi teisės naudotis. Duomenys turi būti apsaugoti nuo praradimo, daromos duomenų bazės kopijos.		
Šaltinis:	Užsakovas		
Tikimo kriterijus:	Sistema turi turėti apsaugą nuo neleistino priėjimo bei duomenų praradimo.		
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1
Priklausomybės:	Nėra	Konfliktai:	Nėra

### **Egzistuojantys sprendimai**

Panašaus pobūdžio sistemų kurias galima būtų galima naudoti, nerasta, kadangi tai specifinė sistema ir atlieka specifines užduotis, todėl spręsti apie kokybę ir vertinti analogiškas sistemas yra pakankamai sudėtinga.

### ***Reikalavimai esamų duomenų perkėlimui***

Keliant į sistemą jokio duomenų transformavimo nereiks. Popierinė informacija apie pacientus bus perkelta į elektroninę formą.

### ***Vartotojo dokumentacija ir apmokymas***

Sistema neturi būti sudėtinga ir reikalauti ilgų gydytojų mokymų, kad galima būtų ja naudotis. Sistema bus įdiegta, vartotojui reikės tik tinkamai susipažinti kaip dirbti su sistema.

### ***Perspektyviniai reikalavimai***

Sistemai vystyti ypatingų papildomų priemonių nereikia, kadangi viskas remiasi jau veikiančia sistema. Tiesiog pridedamos naujos funkcijos į jau naudojamą duomenų bazę.

## IV. MODELIO TYRIMO REZULTATŲ APRAŠYMAS

### 4.1. DBVS charakteristikų palyginimas

Tyrimo iširtos aukščiau darbe aprašytos DBVS: Microsoft SQL Server, Oracle, MySQL, PostgreSQL, DB2, Firebird ir Informix. Duomenys tyrimui naudojami pagal paskutines išleistas DBVS versijas, įvertinant jų gamintojų tinklalapiuose pateiktus duomenis ir oficialiai deklaruojamus dokumentus.

Nagrinėjant konkretų DBVS panaudojimo atvejį ne visos charakteristikos vienodai svarbios ir aktualios. Biomedicininį duomenų saugojimui ir apdorojimui yra svarbi duomenų tipų įvairovė ir galimybė naudotis specializuotais duomenų gavybos įrankiais. Biomedicininiai duomenys pasižymi kompleksišku [36, 37], simbolių tipų įvairove, aktualus ne tik duomenų saugojimas, bet ir DBVS naudojami apdorojimo įrankiai, kurie gali būti integruoti (vidiniai) arba pritaikomi (išoriniai) [29, 30, 9].

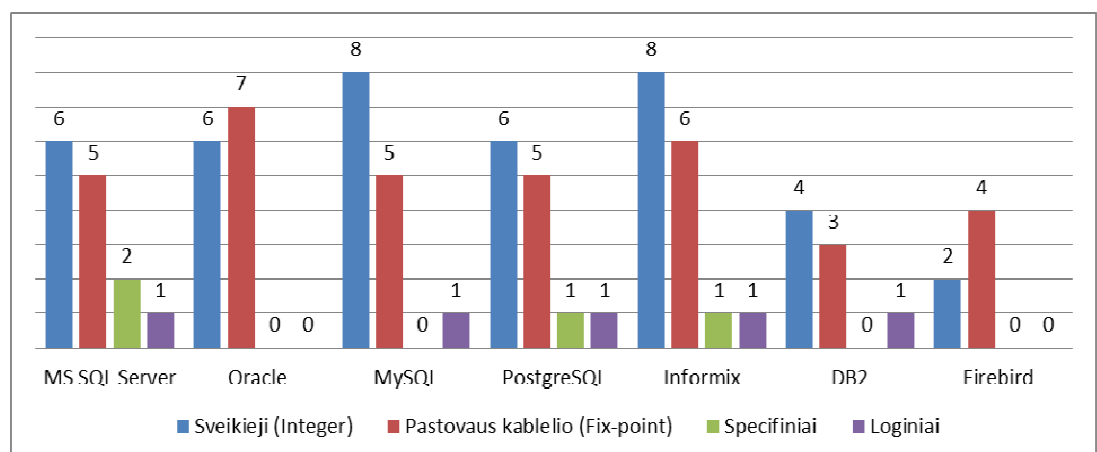
Analizuotos DBVS kokybinės ir kiekybinės charakteristikos: funkcionavimo bazė (palaikomos operacinės sistemos, SQL standartai), taikymo sritis (programavimo kalbos). Lentelėse pateikiamos nagrinėjamų DBVS palaikomos operacinės sistemos, programavimo kalbos, ir SQL standartai, o diagramose pateikiamas apibendrintas visų minėtų charakteristikų suskaičiuotas kiekis.

Analizuotos DBVS kiekybinės charakteristikos, t.y. DBVS turimi duomenų tipai ir jų įvairovė. Duomenų tipus galima būtų suskirstyti į tris pagrindines grupes: skaitmeniniai, datos, ir simboliniai. Lentelėse pateikiami nagrinėjamose DBVS esantys skaitmeniniai tipai, jų įvairovė ir suskaičiuotas kiekis, o diagramose pateikiamas apibendrintas kiekis.

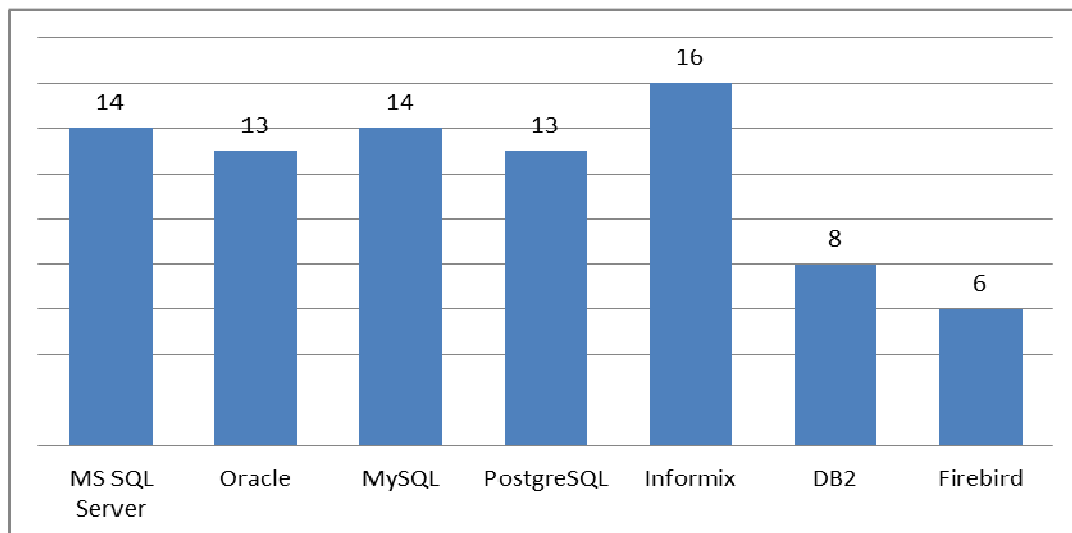
DBVS palyginimo lentelės [6, 9, 12, 14, 15, 16, 18, 19, 20, 28, 31, 33, 35, 57]:

4.1 lentelė. Skaitmeniniai (Numeric) duomenų tipai

	Sveikasis (Integer)	Pastovaus kablelio (Fix-point)	Slankaus kablelio (Floating-point)	Dvejetainis (Boolean)	Money	Loginiai (1 – True arba 0 – False)
MS SQL Server	5	2	3	1	2	1
Oracle	5	3	4	1	0	0
MySQL	7	2	3	2	0	1
PostgreSQL	5	2	2	1	1	1
Informix	7	2	4	1	1	1
DB2	4	0	2	1	0	1
Firebird	2	2	2	0	0	0



4.1 pav. Skaitmeninių duomenų tipų palyginimo diagrama

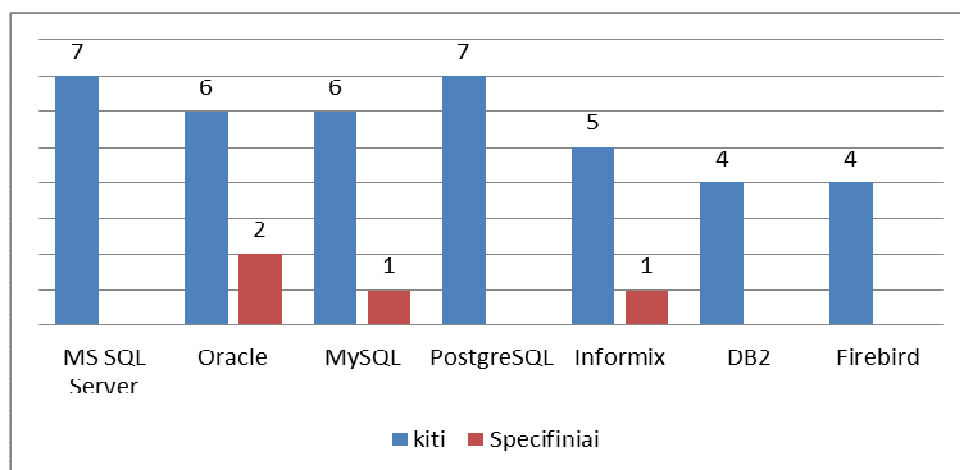


4.2 pav. Bendra skaitmeninių duomenų tipų palyginimo diagrama

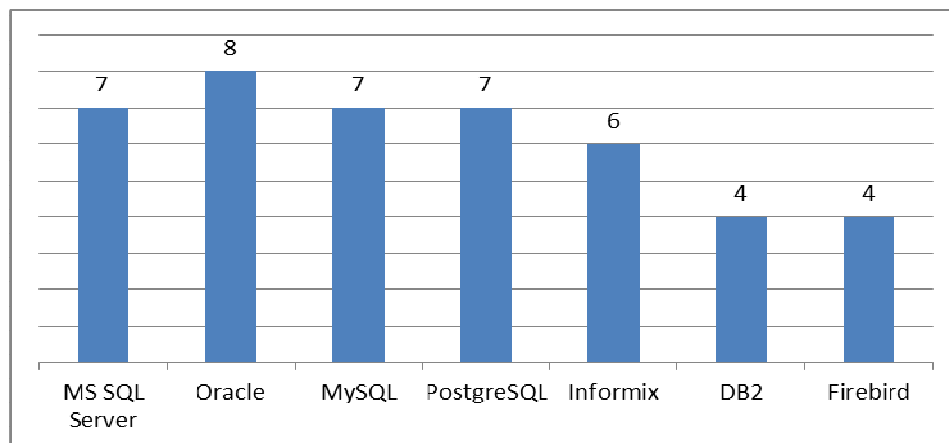
Kaip matoma iš pateiktos diagramos, skaitmeninių duomenų tipų įvairovė išsiskiria Informix, likusioji dalis DBVS turi panašų kiekį skaitmeninių tipų, o mažiausiai turi – Firebird.

4.2 lentelė. Datos duomenų tipai

DBVS	Palaikomi datos duomenų tipai	Specifiniai
MS SQL Server	7	
Oracle	6	2
MySQL	5	1
PostgreSQL	7	
Informix	4	1
DB2	4	
Firebird	4	



4.3 pav. Datos duomenų tipų palyginimo diagrama

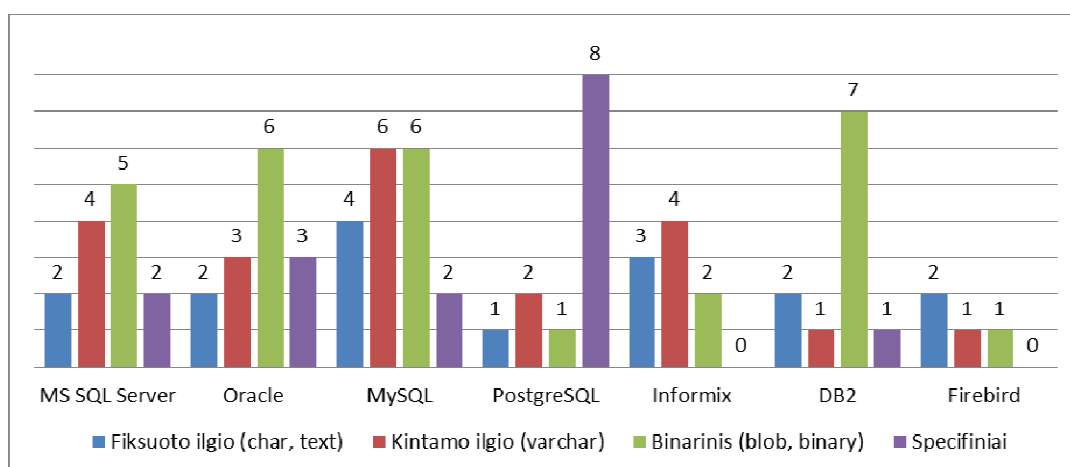


4.4 pav. Bendra datos duomenų tipų palyginimo diagrama

Diagramoje pateiktas apibendrintas datos duomenų tipų kiekis. Nagrinėjami visi egzistuojantys datos tipai ir pateiktas suskaičiuotas bendras jų kiekis. Kaip matoma diagramoje, dauguma DBVS turi panašų datos duomenų tipų kiekį ir didelių skirtumų nėra. Keliais taškais mažiau turi tik DB2 ir Firebird DBVS.

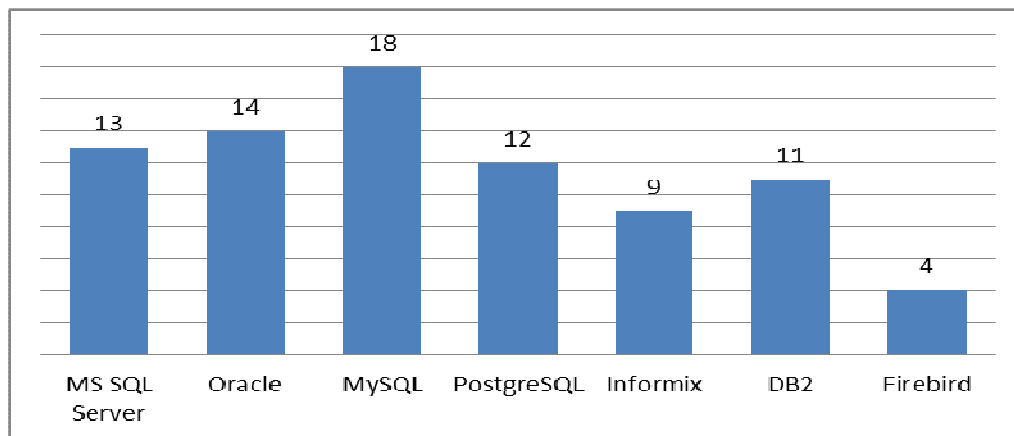
4.3 lentelė. Eilučių (String) duomenų tipai

	Fiksuoto ilgio (char, text). Simbolių pavidalas.	Kintamo ilgio (varchar). Simbolių pavidalas.	Binarinis (blob, binary).	Specifinių simbolių	Specifiniai
MS SQL Server	2	4	5	0	2 (geometriniai, geografiniai)
Oracle	2	3	6	0	2 (spatial: SDO_geometry, SDO_Raster)
MySQL	4	6	6	2 (SET ir ENUM)	
PostgreSQL	1	2	1	2 (serial)	6 (geometriniai), 4(tinklo adreso)
Informix	3	4	2		
DB2	2	1	7		(grafiniai)
Firebird	2	1	1		



4.5 pav. Eilučių (String) duomenų tipų palyginimo diagrama

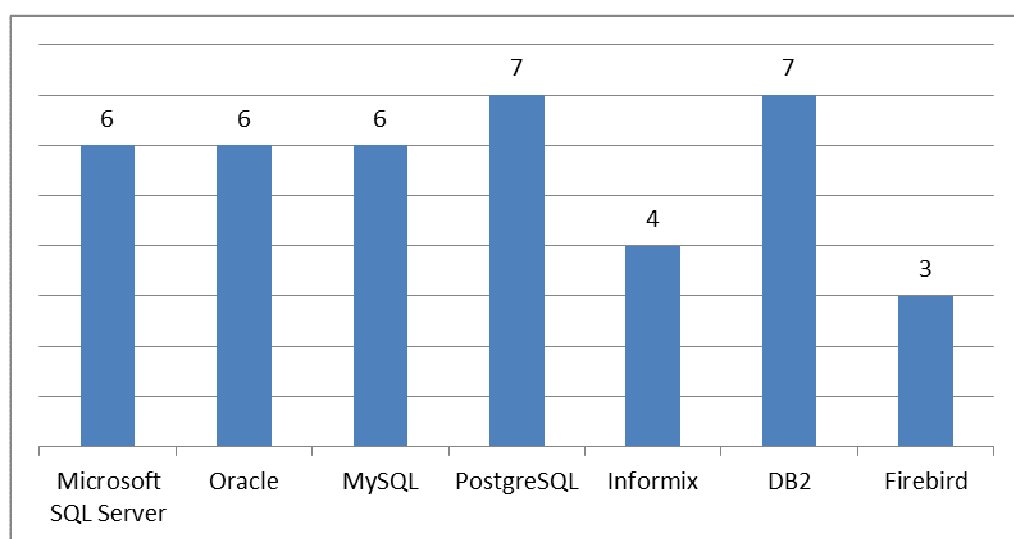




4.6 pav. Bendra eilučių (String) duomenų tipų palyginimo diagrama

4.4 lentelė. DBVS palyginimas pagal šifravimo algoritmus.

	DES	AES	3DES	RC4	FIPS	DESX	MD5	SHA1	Password	Viso
<b>Microsoft SQL Server</b>	+	128, 192, 256 bitų	+	+ 128bit	-	+	-	-	+	<b>6</b>
<b>Oracle</b>	+ 56 bitų	+ 128, 192, ir 256 bitų	+ 112 ir 168 bitų	+ 40, 56, 128, 256 bitų ilgio raktai	+	-	-	-	+	<b>6</b>
<b>MySQL</b>	+	+ 128 ir 256 bitų	+	-	-	-	+	+	+	<b>6</b>
<b>PostgreSQL</b>	+	+	+	-	-	+	+	+	+	<b>7</b>
<b>Informix</b>	56, 64 bitų	128 bitų	192 bitų	-	-	-	-	-	+	<b>4</b>
<b>DB2</b>	64 bitų	+	192 bitų	40, 64, 128 bitų	-	-	+	+	+	<b>7</b>
<b>Firebird</b>	+	+	+							<b>3</b>



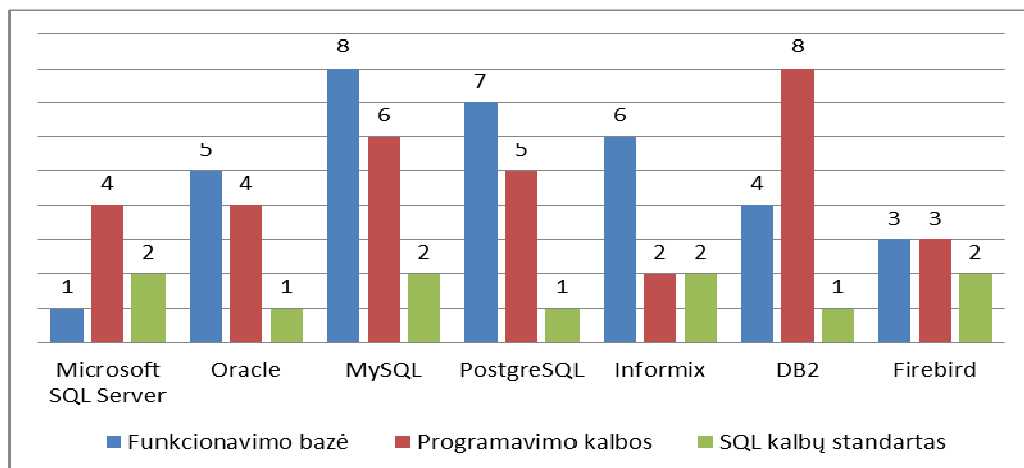
4.7 pav. Šifravimo algoritmų duomenų tipų palyginimo diagrama

Palaikomų šifravimo algoritmų skaičius panašus, mažiau turi Informix ir Firebird DBVS.

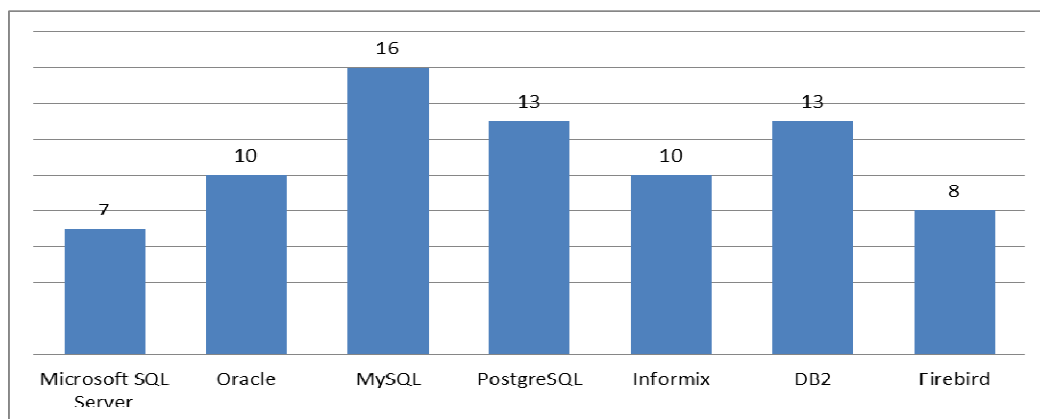
4.5 lentelė. DBVS kokybinės charakteristikos [6, 9, 12, 14, 15, 16, 18, 19, 20, 28, 31, 33, 35, 57]:

	<b>Funkcionavimo bazė</b>	<b>Programavimo kalbų palaikymas</b>	<b>SQL kalbų standartas</b>	<b>Duomenų apsauga nuo praradimo</b>	<b>Administravimas</b>
<b>MS SQL Server</b>	Windows Server 2003 Service Pack 2, Windows Server 2008, Windows Vista Windows XP Service Pack 2, Windows XP Service Pack 3	VB, C++, C#, XML	Transact-SQL(T-SQL) ir ANSI SQL	DB kopijų darymo ir atstatymo procedūros.	Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.
<b>Oracle</b>	Microsoft Windows (32-64 bit); Linux x86; Linux x86-64; Solaris (SPARC) (64-bit); AIX (PPC64); HP-UX Itanium; HP-UX PA-RISC (64-bit)	Java, C, C++ ir PL/SQL	SQL-92	DB kopijų darymo ir atstatymo procedūros.	Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.
<b>MySQL</b>	Windows 2000/XP/Vista, Windows Server 2003/2008, Mac OS, Linux, Unix, Solaris, AIX, HP-UX Itanium, Amiga OS, BSD	C, C++, C#, Java, Perl, Python	SQL-92, ANSI/ISO SQL	MySQL duomenų bazėms gali būti daromos kopijos, kurios, esant būtinybei, būtų atstatytos. InnoDB siūlo specializuotą kopijos darymą, nesustabdant vykstančių darbų su duomenimis.	Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.
<b>PostgreSQL</b>	Linux, AIX, BSD, HP-UX, SGI-IRIX, Mac OS X, Solaris, Windows	PL/pgSQL, PL/Perl, C, C++ bei Java	PL/pgSQL	SQL dump; Failų lygio kopija; Kopija realiame laike.	Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.
<b>Informix</b>	Linux, Windows, Mac OS(10.5.2 ir vėlesnė), Solaris 64-bit, AIX, BSD	Java ir COM+	SQL-92, ANSI-SQL		Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.
<b>DB2</b>	Windows(32bit, 64 bit), AIX, HP-UX, Linux, Linux iSeries, Linux pSeries, Linux xSeries, Linux zSeries, Solaris (x64)	SQL, XQuery, Xpath; C/C++ (CLI, ODBC and embedded SQL); Java (JDBC and SQLJ); COBOL; PHP; Perl; Python; Ruby on Rails; .NET	SQL-92		Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.

		languages; OLE-DB; ADO.			
<b>Firebird</b>	Linux, Windows and Mac OS X	C, C++, Delphi	SQL-99, ANSI SQL.	Duomenų kopijoms ir jų atstatymui naudojami realaus laiko dubleriai – nebūtina sustabdyti duomenų bazės serverį kopijai padaryti	Lentelių pildymą per grafinę sąsają gali atlikti ir mažiau apmokyti vartotojai, tačiau duomenų bazės projektavimą, diegimą ir priežiūrą turi vykdyti specialistai.



4.8 pav. Kokybinių DBVS charakteristikų palyginimo diagrama



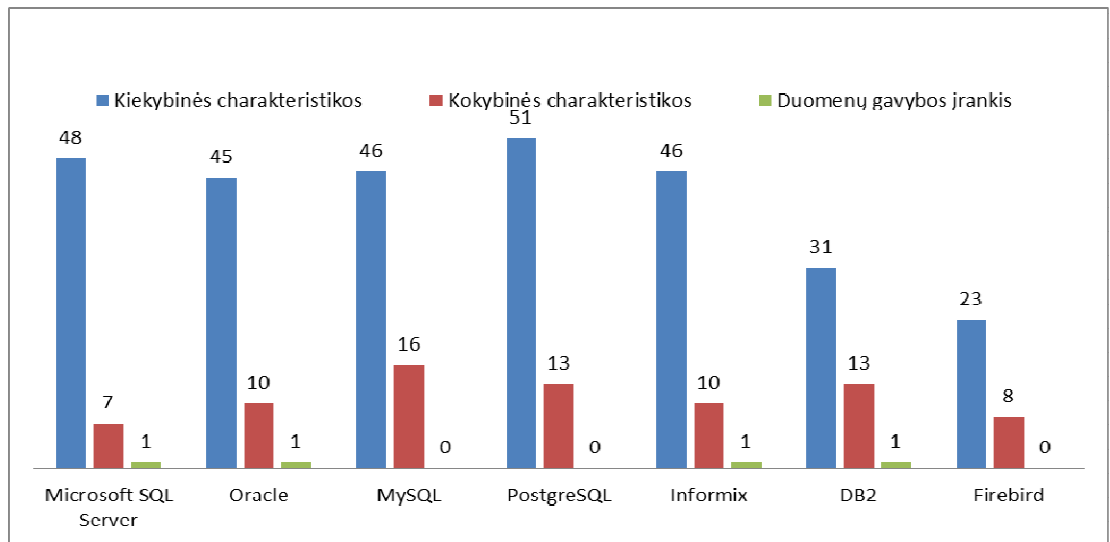
4.9 pav. Bendra kokybinių DBVS charakteristikų palyginimo diagrama

Lyginant kokybines charakteristikas, labiausiai išsiskiria MySQL DBVS, surinkusi didžiausią skaičių kriterijų, kiek mažiau taškų surinko PostgreSQL ir DB2 DBVS.

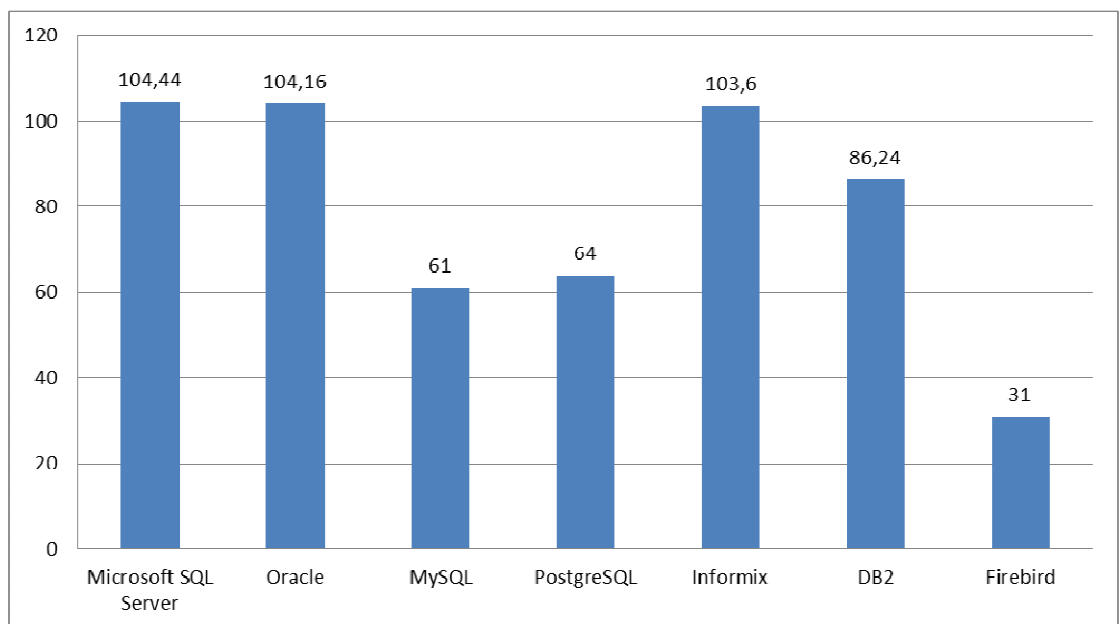
4.6 lentelėje pateikti tik integruoti DBVS DG įrankiai, t.y. siūlomi to paties gamintojo.

4.6 lentelė. DBVS duomenų gavybos įrankių palyginimas

DBVS	Duomenų gavybos įrankis
MS SQL Server	+
Oracle	+
MySQL	-
PostgreSQL	-
Informix	+
DB2	+
Firebird	-



4.10 pav. DBVS charakteristikų svorių (bendra) pasiskirstymo diagrama



4.11 pav. Visų DBVS parametru palyginimo diagrama (paskirsčius svorius)

Vertinant ir lyginant skirtingas DBVS buvo įvestas toks charakteristikų reikšmingumas: duomenų gavybai skirta 40%, o visoms kiekybinėms ir kokybinėms charakteristikoms po 30%. Taigi, surinktus kiekvienos DBVS įverčius perskaičiuota pagal jų reikšmingumą.

Kadangi DBVS duomenų gavybos įrankį turi arba ne, atitinkamai skirta 1 taškas (kas turi) ir 0 taškų (kas neturi), o kiekybinėms ir kokybinėms charakteristikoms palikta tiek kiek surinko taškų.

Visų DBVS parametru palyginimo diagramoje laikoma kad kiekvienos DBVS surinktas bendras taškų skaičius yra 100%, paskaičiuota 40% nuo turimų taškų toms DBVS, kurios turi DG įrankį ir per tą taškų skaičių dar padidinti jų turimi taškai. Tos DBVS kurios turi DG įrankį proporcingai pakyla prieš tas sistemas, kurios DG įrankio neturi.

Atlikus DBVS analizę galima teigti, kad nagrinėtos DBVS savo funkcinėmis galimybėmis yra panašios, tačiau apibendrinus visus kiekvienos DBVS parametrus, kai kurios DBVS surinko daugiau teigiamų įvertinimų (taškų) už kitas. Įvertinus rezultatus galima pastebėti, kad pirmąja šios DBVS: Oracle, MS SQL Server, Informix, DB2.

Svarbus kriterijus renkantis DBVS biomediciniui duomenų apdorojimui yra integruotų (vidinių) duomenų gavybos įrankių palaikymas. Juos turi tik MS SQL Server, Oracle, DB2, Firebird ir Informix.

Apibendrinus visus rezultatus galima teigti, kad objektinė-reliacinė DBVS Oracle yra viena iš tinkamiausių biomedicininų duomenų saugojimui ir apdorojimui kadangi turi integruotą duomenų gavybos įrankį, gali dirbti daugelyje operacinių sistemų, taip pat vienintelė iš visų nagrinėtų DBVS deklaruoja palaikanti medicininų duomenų standartą DICOM [17]. MS SQL Server gali funkcionuoti tik Windows operacinėje sistemoje, todėl ji nesirinkta.

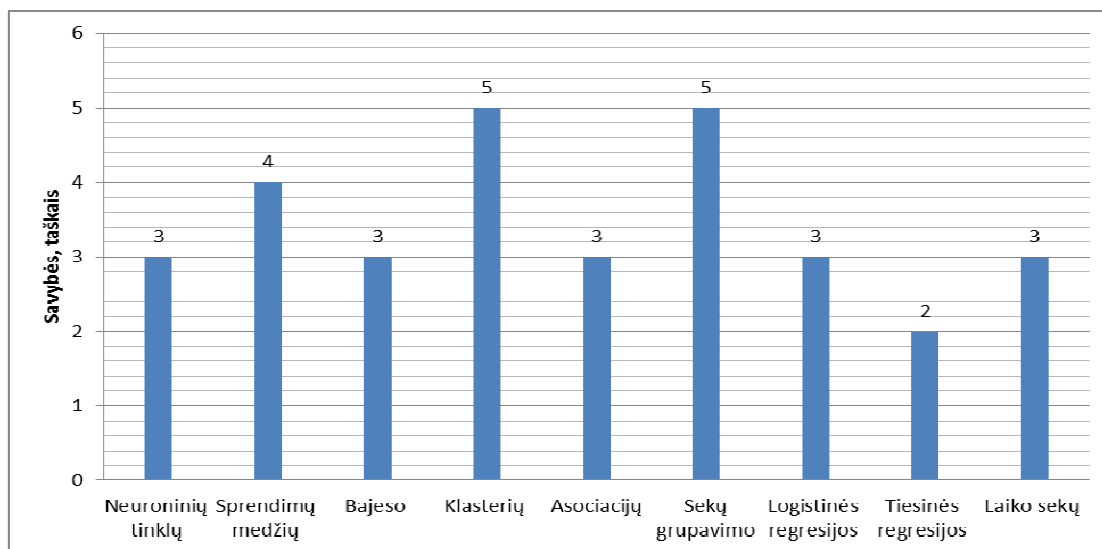
## 4.2. Duomenų gavybos algoritmų palyginimas

4.7 lentelė. Algoritmų palyginimas

Algoritmas	Klasifikacija	Segmentacija	Asociacija	Įvertinimas	Prognozė	Duomenų analizė
Neuroninių tinklų algoritmas	+	-	-	+	-	+/-
Sprendimų medžių algoritmas	+	-	+	+	-	+
Bajeso algoritmas	+	-	+/-	-	-	+
Klasterių algoritmas	+/-	+	+/-	+/-	-	+
Asociacijų taisyklės	+/-	-	+	-	-	+
Sekų grupavimo algoritmas	+/-	+	+/-	+/-	-	+
Logistinės regresijos algoritmas	+	-	-	+	-	+/-
Tiesinės regresijos algoritmas	-	-	-	+	-	+/-
Laiko sekų algoritmas	-	+	-	-	+	+

\* (+) palaikymas pritaikomas, (-) nepalaiko, (+/-) palaikymas pritaikomas dalinai.

Kaip matoma iš pateiktos algoritmų palyginimo lentelės, visi nagrinėti duomenų gavybos algoritmai turi išsamią duomenų analizės galimybę arba bent jau dalinę realizaciją.



4.12 pav. Duomenų gavybos algoritmų palyginimo diagrama

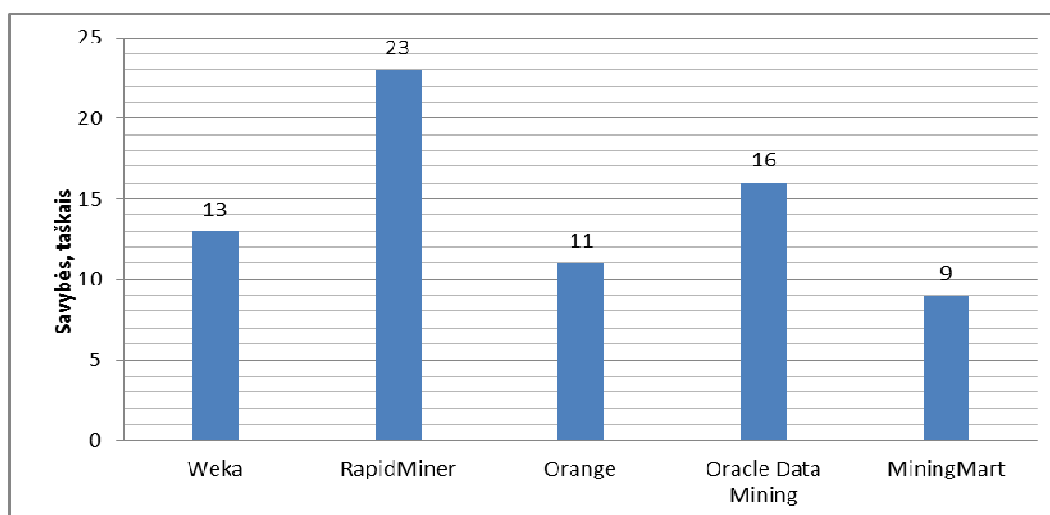
Darbe kaip tinkamiausias pasirinktas panaudoti klasterių algoritmas, kadangi jis skirsto į klasterius pagal panašius požymius. Klasterinės analizės metodu galima suskirstyti ligas į grupes pagal panašius požymius kadangi daugelio ligų požymiai gali būti labai panašūs arba net tokie pat. Sekų grupavimo algoritmas atmetas nes jis grupuoja veiksmų sekas, ieško įvykio patvirtinimo tikimybės ir tik tada skirsto į klasterius.

### 4.3. Duomenų gavybos įrankių palyginimas

4.8 lentelė. Duomenų gavybos įrankių palyginimas

	Palaikomų DBVS skaičius	Algoritmų skaičius	Java palaikymas	Grafinė vartotojo sąsaja	Duomenų eksportas	Duomenų importas
<b>Weka</b>	5	4	+	+	+	+
<b>RapidMiner</b>	10	9	+	+	+	+
<b>Orange</b>	1	7	-	+	+	+
<b>Oracle Data Mining</b>	1	11	+	+	+	+
<b>MiningMart</b>	3	5	-	+	-	-

Lentelėje pateikti ir komerciniai, ir nemokami duomenų gavybos įrankiai, skirti intelektualiai duomenų analizei atlikti.



4.13 pav. Duomenų gavybos įrankių palyginimo diagrama

Įvertinus visus kriterijus tolesniam darbui pasirinktas RapidMiner duomenų gavybos įrankis, nes jis palaiko daugiausia DBVS, turi Java palaikymą, yra lengvai suderinamas su DBVS ir turi nemažai vizualinių priemonių, išvengiama programavimo.

### 4.4. Duomenų gavybos realizacija

Pacientų duomenys yra laikomi suprojektuotoje saugykloje. Kadangi tikrus pacientų duomenis apie susirgimus ir tyrimų duomenis gauti yra sudėtinga, nes tai yra konfidenciali ir saugoma informacija, duomenys į saugyklą buvo suvesti naudojantis enciklopedijomis ir žinytais. Tyrimui atlikti į duomenų bazę buvo suvesta dešimties (10) ligų duomenys su joms būdingais simptomais, ir dešimties pacientų duomenys. Dalis ligų, suvestų į duomenų bazę, tarpusavyje savo simptomais labai panašios, kita dalis visiškai priešingos. Pacientų duomenys nėra susiję su konkrečiu asmeniu. Siekiant išgauti patikimą rezultatą pacientai buvo “susargdinti” dirbtinai, iš anksto parenkant simptomus, nusiskundimus ir tyrimus, būdingus būtent konkrečiai ligai. Tai atlikta todėl, kad tyrimas atliekamas iš informatikos inžinerijos pusės, analizei panaudojant duomenų gavybos metodus, medicininius duomenis ne medicinos specialistui būtų sunku teisingai interpretuoti.

Ruošiant duomenis klasterinės analizės tyrimui atlikti buvo suformuota užklausa, kurios rezultatas – parengta reliacinė lentelė kurią galima iš duomenų bazės eksportuoti į duomenų gavybos įrankį RapidMiner. Kadangi visa informacija, esanti duomenų bazių lentelėse, analizei nėra reikalinga

pateikti, suformuota užklausa leidžianti išfiltruoti ligas pagal paciento nusiskundimus, simptomus ir veiksnius, galinčius įtakoti ligos buvimą arba nebuvimą, taip pat pagal atliktus tyrimus, įtraukiant atlikto tyrimo reikšmę. Tuo atveju gauta lentelė *ANALIZE* kuri bus pateikta duomenų gavybos įrankiui tolesniam duomenų apdorojimui.

*ANALIZE* lentelės formavimo užklausa:

*Create table analize*

*AS select ligos\_id from veik\_lig where veiksnys\_id in (select veiksnys\_id from veik\_pac where paciento\_id = XXXXXXXXXXXX);*

*insert into analize (ligos\_id) select ligos\_id from sim\_lig where simpt\_nusisk\_id in (select simpt\_nusisk\_id from nusiskundimas where paciento\_id = XXXXXXXXXXXX);*

*insert into analize (ligos\_id) select ligos\_id from tyr\_lig where tyr\_lig\_id in (select tyrimo\_id from atlikto\_tyrimo\_duomenys where paciento\_id = XXXXXXXXXXXX)*

*AND tyrimo\_id in (select tyrimo\_duomenys.tyrimo\_id*

*FROM tyrimo\_duomenys, atlikto\_tyrimo\_duomenys, atliktas\_tyrimas*

*WHERE tyrimo\_duomenys.mataavimo\_vienetai=atlikto\_tyrimo\_duomenys."mataavimo\_vienetai"*

*AND atlikto\_tyrimo\_duomenys.reiksme*

*BETWEEN tyrimo\_duomenys.nuo AND tyrimo\_duomenys.iki);*

*select ligos\_id, count(ligos\_id) from analize group by ligos\_id;*

Duomenų bazėje suformavus užklausą galima pakankamai tiksliai suskaičiuoti ir atrinkti ligą pagal simptomus, veiksnius ir tyrimus, tačiau tikslesniam rezultatų įvertinimui bus naudojama duomenų gavyba.

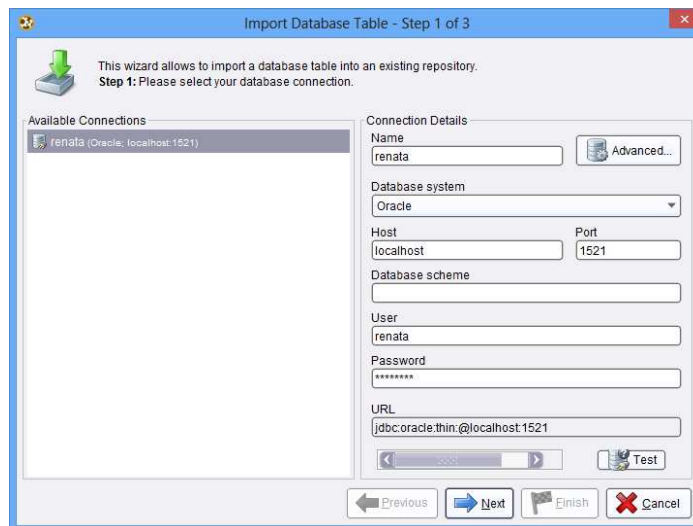
Kaip matoma žemiau pateiktame paveikslėlyje (žr. 4.14 pav), įvykdžius suformuotą užklausą duomenų bazėje, gautas rezultatas liga *E.10 (cukrinis diabetas)*.

E.05	6
E.20	1
C22.9	4
J.45	1
E.10	19
E.03.2	2
D50	2
B01	2

**4.14 pav.** Užklausa gauto rezultato Oracle DB lango fragmentas

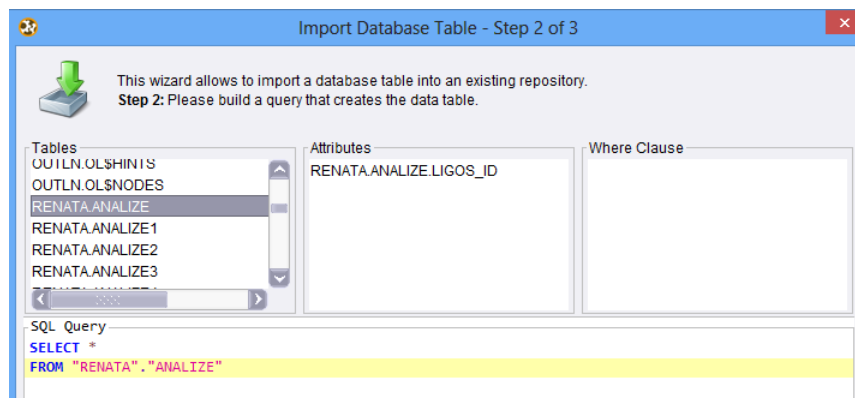
Tolesnei duomenų analizei naudojamas duomenų gavybos įrankis RapidMiner. Kad tinkamai veiktų, duomenų gavybos programinę įrangą reikėjo paruošti ir suderinti su Oracle duomenų baze kad būtų galima importuoti parengtas analizei lenteles.

Paveikslėlyje pavaizduotas (žr. 4.15 pav.) Oracle DB prisijungimo prie Rapid Miner langas. Jame nurodomi pagrindiniai duomenys reikalingi pasiimti duomenis iš duomenų bazės – duomenų bazės vardas, prisijungimo prie DB duomenys, duomenų bazės URL adresas.



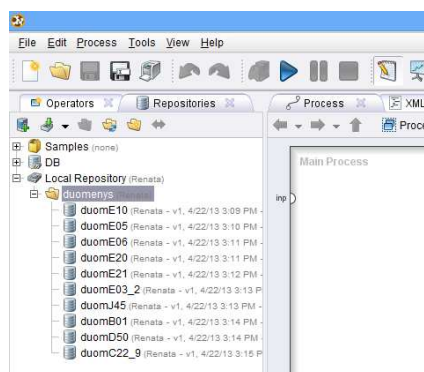
4.15 pav. Oracle DB prisijungimo prie Rapid Miner langas

Duomenų gavybos įrankio RapidMiner lango viršutinėje dalyje (žr. 4.16 pav.) matomas lentelių pasirinkimas kur žymėjimo būdu pasirenkama norima analizuoti lentelė, lango apatinėje dalyje yra SQL užklauskos formavimo langas, kur SQL užklausa galima įvykdyti lentelės pasirinkimą ir duomenys iš duomenų bazės importuojami į RapidMiner duomenų apdorojimo programą.



4.16 pav. Duomenų importavimo iš Oracle DB į Rapid Miner lango fragmentas

Visi duomenys buvo sėkmingai importuoti (žr. 4.17 pav.).



4.17 pav. Importuotų duomenų iš Oracle DB į Rapid Miner lango fragmentas

## 4.5. Klasterinė duomenų analizė

### 4.5.1. Expectation Maximization metodas

Klasterinėje analizėje požymių vektoriai yra apjungiami į vieną klasterį, jeigu jie yra panašūs arba jeigu jie yra pakankamai arti vienas kito, t.y. kai atstumas tarp jų yra nedidelis. Klasterizavimo rezultatai priklauso nuo pasirinkto algoritmo specifikos ir klasterizavimo kriterijaus. Algoritmas visų



pirma identifikuoja duomenų rinkinių tarpusavio ryšius ir sugeneruoja eilę klasterių, atsižvelgdamas į šiuos ryšius [44].

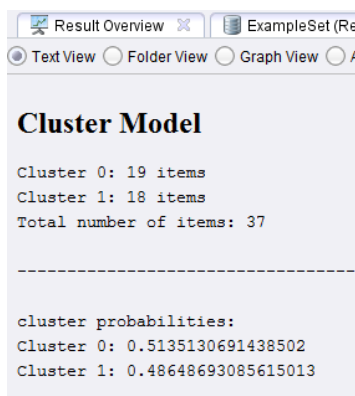
Galimybių maksimizavimo (*angl. Expectation Maximization*) atveju algoritmas naudoja tikimybinį metodą, kad suskaičiuotų tikimybę, ar duomenų taškas egzistuoja klasteryje [44].

Vizualiam algoritmo duomenų grupavimui pasirinkta išbarstytų taškų erdvė, kuri atvaizduoja visus duomenų rinkinio atvejus, kurių kiekvienas yra diagramos taškas. Klasteriai sugrupuoja diagramos taškus ir atvaizduoja algoritmo atpažįstamus ryšius. Nustatęs klasterius, algoritmas apskaičiuoja taškų grupių efektyvumą ir bando iš naujo apibrėžti grupes, kad sukurtų geresnius klasterius. Proceso metu algoritmas vykdomas tol, kol nebegali pagerinti rezultatų. Nustačius klasterius yra pakankamai sudėtinga juos interpretuoti. Todėl įvykdžius algoritmą gaunamas grafinis vaizdas-interpretacija [42].

Klasterinės analizės metodu galima suskirstyti ligas į grupes pagal panašius požymius, turint omenyje tai, kad daugelio ligų požymiai gali būti labai panašūs ir persidengti arba net būti tokie pat.

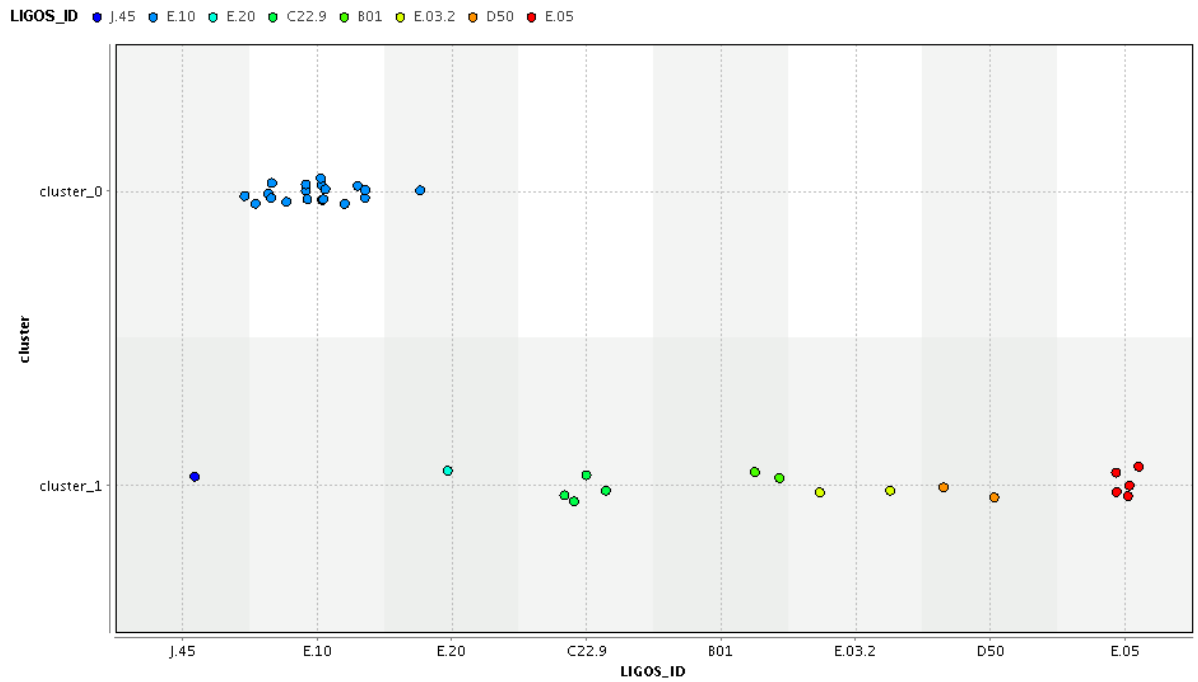
Eksperimento metu praktiškai įvykdžius algoritmą, žemiau pateiktame paveikslėlyje (žr. 4.18 pav.) pateikti gauti *Expectation Maximization* klasterizavimo operatoriaus rezultatai: modelis – kiek elementų, kiek ligų simptomų ir požymių yra priskiriama klasteriui, ir kaip duomenys suskirstyti į klasterius. Klasterių skaičius paliktas pagal nutylėjimą – 2.

Matoma kad klasterio tikimybė nuliniame klasteryje yra 0,51, t.y. didesnė už pirmojo klasterio.



**4.18 pav.** Klasterizavimo operatoriaus *Expectation Maximization* rezultatų lango fragmentas

Žemiau pateiktame paveiksle (žr. 4.19 pav.) pavaizduota kaip ligos požymiai pasiskirsto į klasterius. Ligos pateikiamos pagal spalvas. Matoma kad didžiausias sutelktų taškų skaičius yra nuliniame klasteryje, tai reiškia kad didžiausia yra *E10* ligos tikimybė, daugiausia požymių ir panašumų yra surinkta tame klasteryje. Kadangi šiuo atveju liga iš anksto yra žinoma, o klasterizavimo metodu tikrinamas jos požymių išsidėliojimas klasteriuose, galima manyti jog klasterine analize galima pakankamai tiksliai gauti ligos diagnozę.



4.19 pav. Klasterizavimo operatoriaus Expectation Maximization suskirstytų į klasterius rezultatų lango fragmentas

Pakeitus klasterių skaičių į 8, t.y. kiekvienai ligai išskiriant po klasterį, klasterių operatoriaus pateiktuose rezultatuose galima pamatyti grupavimo rezultatus, atskirus klasterius su skirtingais skaičiais. Daugiausia elementų – 19 sukaupta pirmame klasteryje, kituose klasteriuose sukaupta gerokai mažiau elementų.

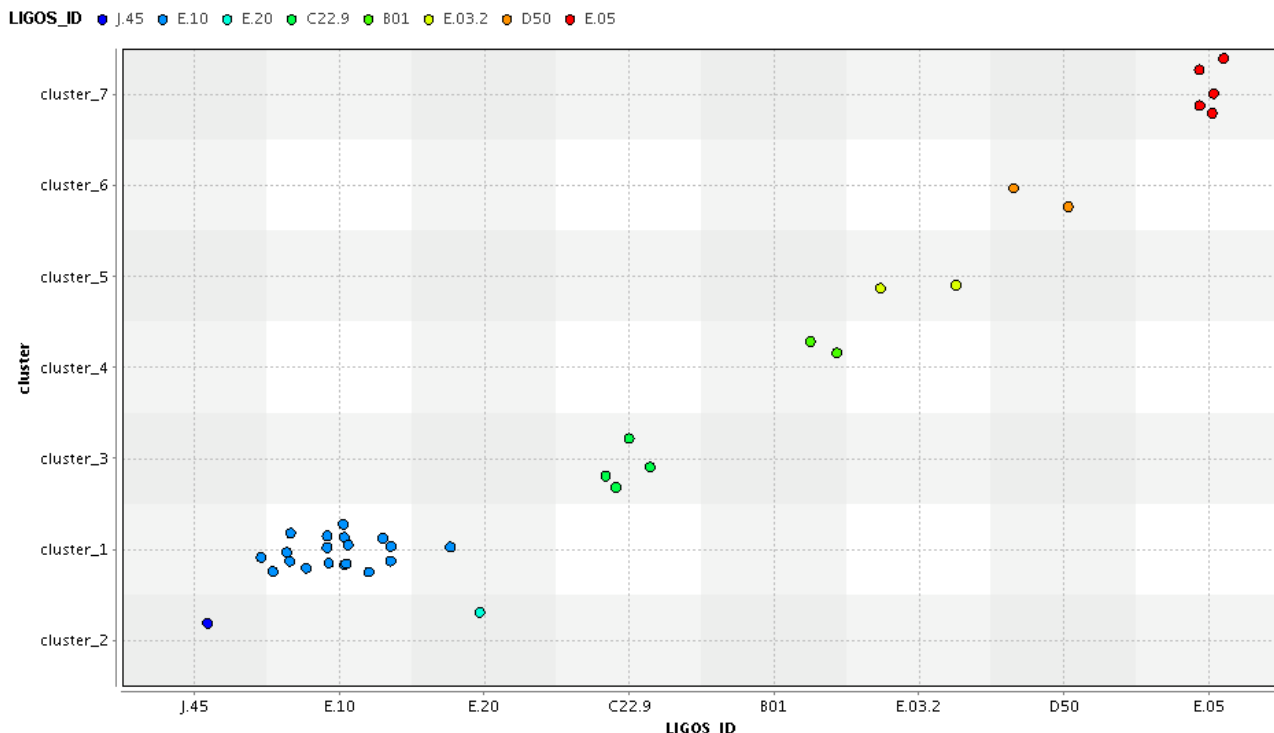
```

Cluster Model

Cluster 0: 0 items
Cluster 1: 19 items
Cluster 2: 2 items
Cluster 3: 4 items
Cluster 4: 2 items
Cluster 5: 2 items
Cluster 6: 2 items
Cluster 7: 6 items
Total number of items: 37

```

4.20 pav. Klasterizavimo operatoriaus Expectation Maximization su 8 klasteriais rezultatų lango fragmentas



**4.21 pav.** Klasterizavimo operatoriaus Expectation Maximization suskirstytų į 8 klasterius rezultatų lango fragmentas

Grafinis atvaizdavimas parodo kad pirmame klasteryje ties *E10* liga yra sutelktas didžiausias skaičius taškų-požymių, kituose klasteriuose taškų matoma gerokai mažiau. Didžiausia yra *E10* ligos tikimybė.

#### 4.5.2. *K-means* metodas

*K*-reikšmių (angl. *K-means*) atveju algoritmas naudoja atstumo rodiklį, kad priskirtų duomenų tašką artimiausiam klasteriui.

Klasterizavimo algoritmo veikimo principas: [44]

1. Objektai skirstomi į *k* pradinių klasterių.
2. Apskaičiuojami klasterių centrai.
3. Apskaičiuojamas kiekvieno objekto atstumas iki klasterių centrų.
4. Skaičiuojama kvadratinė paklaida.
5. 2-5 etapai kartojami tol, kol gauname mažiausią kvadratinę paklaidą, kuri yra mažesnė už pasirinktą reikšmę, arba objektai nepersiskirsto po kitus klasterius.

*K-means* metodo savybės:

1. Metodas yra tinkamas, kai duomenų rinkinys yra didelis. Kitaip negu hierarchiniame klasterizavime, iteracijų skaičius gali būti mažesnis negu objektų skaičius.
2. *K-means* algoritmas naudoja paprastus, pvz. Euklido atstumų matavimus, ir tai negali būti panaudota daugeliui duomenų tipų, pvz., nominalių duomenų kaip spalvos [42, 44].

*K-Means* pagrindinis parametras yra klasterių skaičius. Algoritmas prasideda atsitiktinai pasirenkant klasterių centrus. Tada kiekvienam centrui yra priskiriami jam artimiausi taškai ir perskaičiuojami nauji centrai. Sekančiai algoritmas kartojamas iteraciškai tol, kol klasterių centrai stabilizuojasi arba kol būna pasiekiamas maksimalus iteracijų skaičius [44, 58].

Žemiau pateiktame paveikslėlyje (žr. 4.22 pav.) pateikti *K-means* klasterizavimo metodo operatoriaus rezultatai kiek elementų priskiriama klasteriui ir kaip duomenys suskirstyti į klasterius. Klasterių skaičius pagal nutylėjimą paliktas 2. Matoma kad didžiausias elementų skaičius sukauptas pirmame klasteryje.

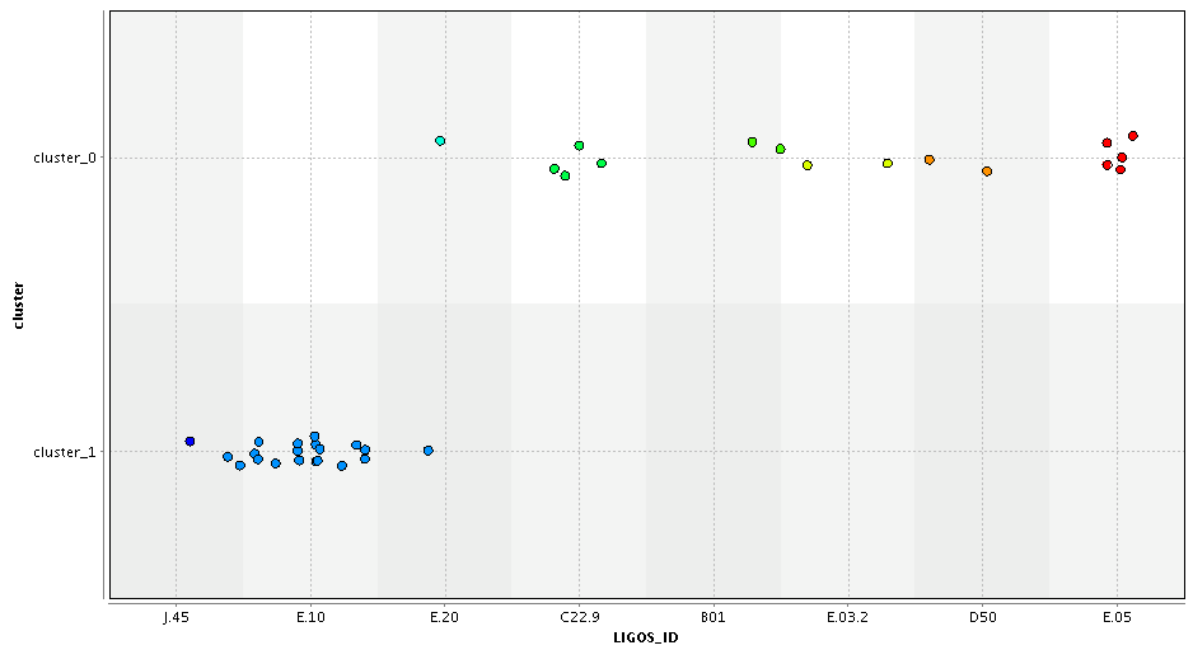
## Cluster Model

```
Cluster 0: 17 items  
Cluster 1: 20 items  
Total number of items: 37
```

4.22 pav. Klasterizavimo operatoriaus K-means rezultatų lango fragmentas

4.23 paveiksle matomas ligų požymių pasiskirstymas klasteriuose. Didžiausias susitelkimas stebimas trečiame klasteryje ties E10 (cukrinis diabetas) liga. Taikant grupavimo algoritmą ligų analizei galima pamatyti ligų požymių pasiskirstymą klasteriuose.

LIGOS\_ID ● J.45 ● E.10 ● E.20 ● C22.9 ● B01 ● E.03.2 ● D50 ● E.05



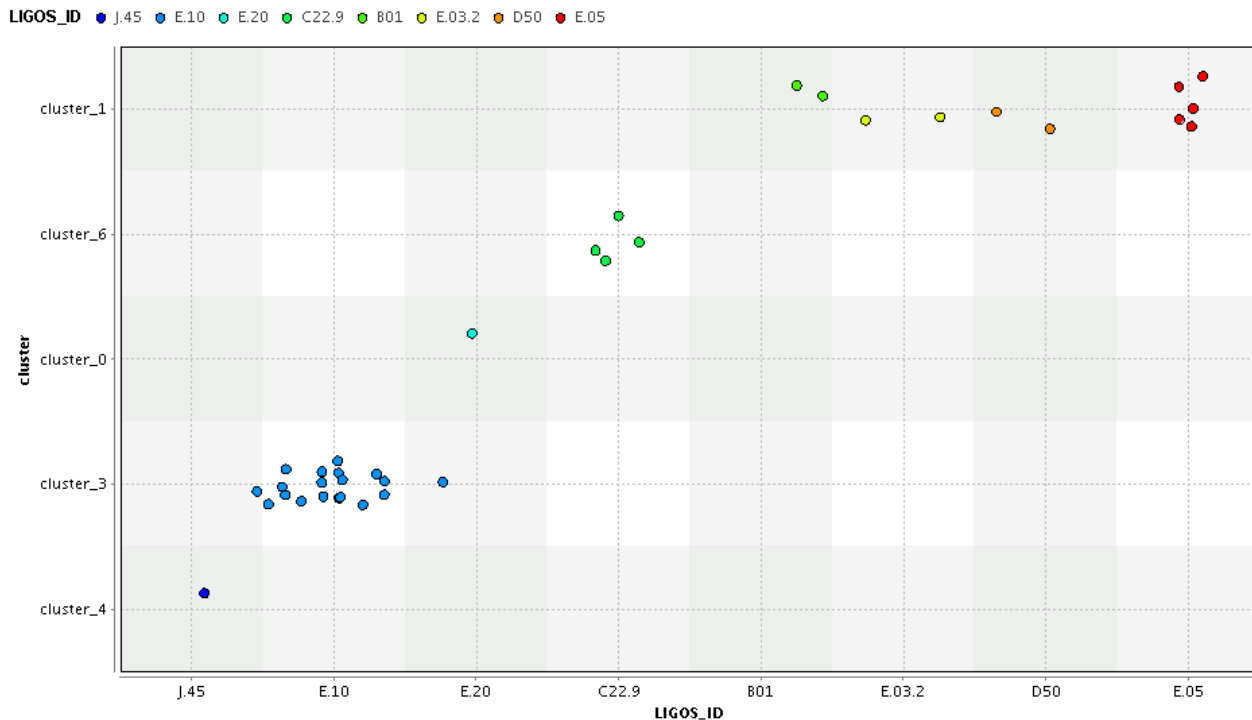
4.23 pav. Klasterizavimo operatoriaus K-means suskirstytų į klasterius rezultatų lango fragmentas

4.24 paveiksle matoma kad padidinus klasterių skaičių į 8, klasterių operatoriaus rezultatuose matoma kad daugiausia elementų – 19 sukaupta trečiame klasteryje, 12 elementų sukaupta trečiame klasteryje, kituose klasteriuose gerokai mažiau elementų, antrame, penktame ir septintame klasteryje pateikti nuliai, tai reiškia kad į šiuos klasterius nepateko jokie ligų požymiai.

## Cluster Model

```
Cluster 0: 1 items  
Cluster 1: 12 items  
Cluster 2: 0 items  
Cluster 3: 19 items  
Cluster 4: 1 items  
Cluster 5: 0 items  
Cluster 6: 4 items  
Cluster 7: 0 items  
Total number of items: 37
```

4.24 pav. Klasterizavimo operatoriaus K-means suskirstytų į 8 klasterius rezultatų lango fragmentas



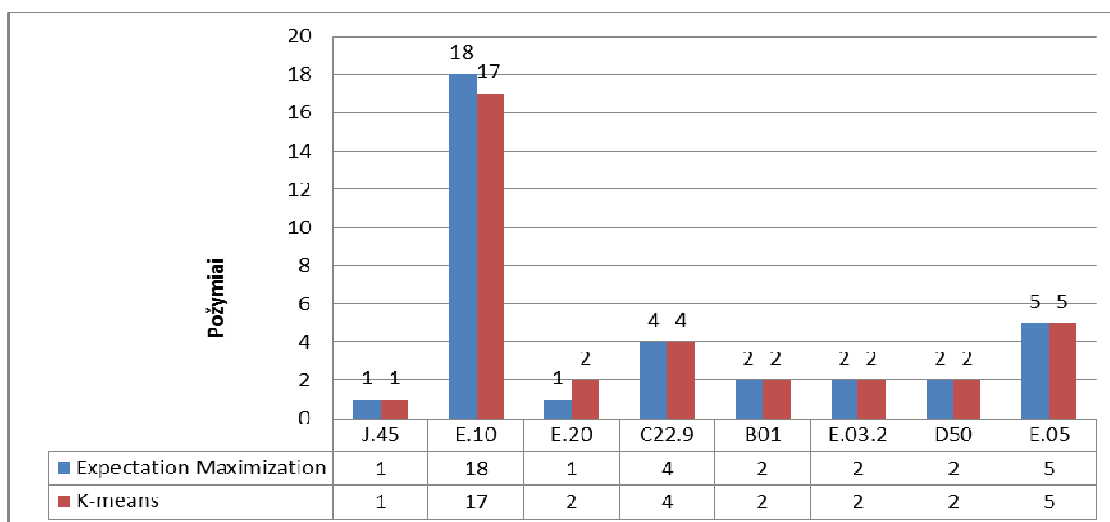
4.25 pav. Klasterizavimo operatoriaus K-means suskirstytų į 8 klasterius rezultatų lango fragmentas

Grafike 4.25 pav. atvaizduojami tik tie klasteriai kuriuose yra sukaupti ligų požymiai. Matoma kad trečiame klasteryje sukauptas didžiausias skaičius E10 ligos požymių. Klasterių operatorius parodė kad pirmame klasteryje yra kiek mažiau požymių negu trečiajame klasteryje, tačiau grafike matoma jog pirmajame klasteryje yra skirtingų ligų požymiai. Kaip jau minėta aukščiau, atliekant analizę liga iš anksto yra žinoma, o klasterizavimo metodu tikrinamas jos panašumų-simptomų-požymių išsidėliojimas klasteriuose, todėl galima manyti jog klasterine analize galima pakankamai tiksliai gauti ligos diagnozę.

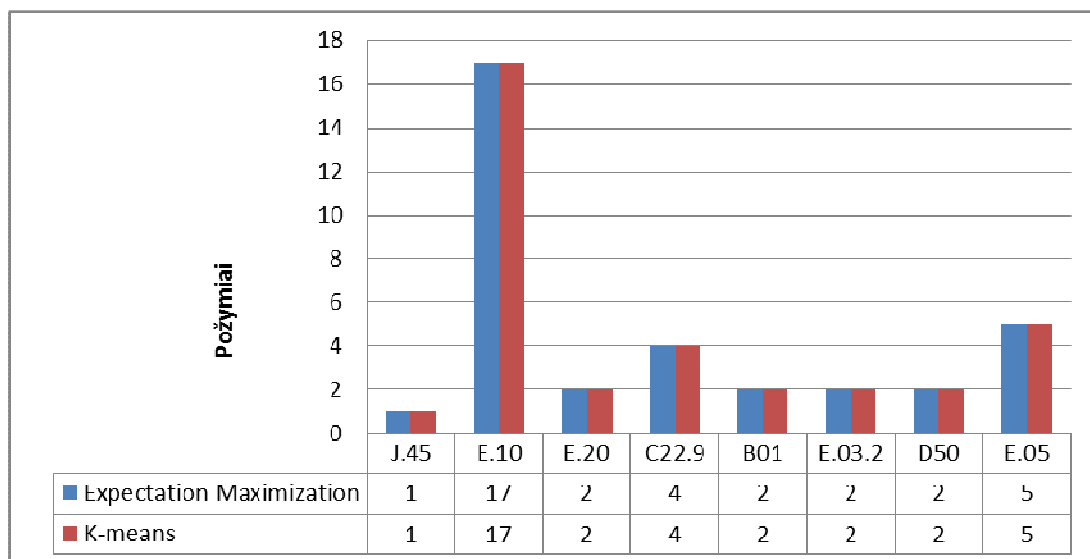
Visa tai apibendrinus, galima teigti jog gydytojas galėtų priimti ligos diagnozavimo procesą teigiamai įtakančius sprendimus.

Aukščiau aprašyti klasterizacijos metodai buvo išbandyti su visomis ligomis ir pacientais, esančiais duomenų bazėje, tačiau dėl rezultato panašumo ir nedidelio reikšmingumo diagramos darbe nebuvo vaizduojamos.

#### 4.6. Analizės rezultatų palyginimas



4.26 pav. Klasterinės analizės Expectation Maximization ir K-means metodų (su 2 klasteriais) rezultatų palyginimo diagrama



4.27 pav. Klasterinės analizės *Expectation Maximization* ir *K-means* metodų (su 8 klasteriais) rezultatų palyginimo diagrama

Palyginus gautus abiejų klasterizavimo metodų *Expectation Maximization* ir *K-means* rezultatus (žr. 4.26, 4.27 pav.) didelio skirtumo nestebima. Analizuota klasterių skaičių paliekant pagal nutylėjimą, t.y. 2 ir padidinus klasterių skaičių iki 8 pagal esamas ligas, kiekvienai ligai išskiriant po klasterį. *Expectation Maximization* duomenų gavybos metodas dviejų klasterių atveju pažymėjo vienu požymiu daugiau, tačiau tai esmės nekeičia nes abu klasterizavimo metodai tiksliai parodė tos pačios *E10* ligos tikimybę.

## IŠVADOS

- Atlikus duomenų bazių valdymo sistemų analizę galima teigti, kad visi analizuoti duomenų bazių serveriai savo funkcinėmis galimybėmis yra labai panašūs. Tyrimui atlikti buvo pasirinkta Oracle DBVS kaip viena iš tinkamiausių biomedicininų duomenų saugojimui ir apdorojimui, nes turi integruotą duomenų gavybos įrankį, gali dirbti daugelyje operacinių sistemų, taip pat vienintelė iš visų nagrinėtų DBVS deklaruoja palaikanti medicininių duomenų standartą DICOM.
- Atlikus duomenų gavybos algoritmų analizę pastebėta kad visi nagrinėti duomenų gavybos algoritmai turi išsamią duomenų analizės galimybę arba bent jau dalinę jų realizaciją. Tyrimui pasirinktas ir sėkmingai pritaikytas klasterių algoritmas, *Expectation Maximization* ir *K-means* metodai. Klasterių algoritmas pasirinktas nes skirsto į klasterius pagal panašius požymius. Klasterinės analizės metodu galima suskirstyti ligas į grupes pagal panašius požymius kadangi daugelio ligų požymiai gali būti labai panašūs arba net tokie pat.
- Atlikta komercinių ir nemokamų duomenų gavybos įrankių, skirtų intelektualiai duomenų analizei atlikti, analizė. Įvertinus visus kriterijus duomenų analizei pasirinktas RapidMiner duomenų gavybos įrankis, nes jis palaiko daugiausia DBVS, turi Java palaikymą, visus pagrindinius klasterizavimo metodus, yra lengvai suderinamas su DBVS ir turi nemažai vizualinių priemonių, išvengiama programavimo ir yra patogus dirbti.
- Šiame magistro darbe buvo sėkmingai praktiškai panaudoti du duomenų gavybos algoritmai – *Expectation Maximization* ir *K-means* metodai ligos diagnostavimo proceso analizei. Abu pasirinkti ir praktiškai pritaikyti klasterizavimo metodai pasitvirtino – jie tinka biomedicininų duomenų analizei ir pakankamai efektyviai prognozuoja ligą. Šiame darbe iširta klasterizavimo metodika efektyvi ir tinka analizuoti biomedicininiams duomenims.
- Sistema atitinka biomedicininų duomenų analizės sistemos modelio kūrimui iškeltus reikalavimus ir įrodė pasirinktų priemonių bei pasirinkto metodo efektyvumą. Visa tai apibendrinus, galima manyti jog gydytojas galėtų priimti ligos diagnostavimo procesą teigiamai įtakojančius sprendimus.

**Praktinis naujumas:** sukurto biomedicininų duomenų analizės sistemos modelio pagrindu galima kurti intelektualias biomedicininis sprendimų priėmimo sistemas gydytojams ir kitiems medicinos darbuotojams kaip pagalbinę priemonę ligų diagnostikai.

**Mokslinis naujumas:** duomenų bazių, duomenų gavybos metodų ir įrankių naudojimas biomedicinoje ligų nustatymui, analizuojant paciento duomenis. Panašių, konkrečių modelių mokslinėje literatūroje nepavyko aptikti.

## LITERATŪRA

1. Duomenų gavyba. [žiūrėta 2011 lapkričio 14 d.]. Prieiga internete: <http://beta.mokslai.lt/referatai/referatas/duomenu-gavyba-puslapis1.html>
2. Thearling. K. Information about data mining and analytic technologies [žiūrėta 2011 lapkričio 24 d.]. Prieiga internete: <http://www.thearling.com/>
3. Sakalauskas L. Duomenų gavyba. [žiūrėta 2011 lapkričio 14 d.]. Prieiga internete: [old.ututi.lt/content/get\\_content/6800](http://old.ututi.lt/content/get_content/6800)
4. What Is Data Mining? [žiūrėta 2011 lapkričio 16 d.]. Prieiga internete: <http://infolab.stanford.edu/~ullman/mining/overview.pdf>
5. Data Mining: What is Data Mining? [žiūrėta 2012 lapkričio 15 d.]. Prieiga internete: <http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm>
6. Paliulis E., Baronienė R. Duomenų bazių valdymo sistemų analizė // Jaunųjų mokslininkų darbai. ISSN 1648-8776. 2010, Nr. 3(28), p. 94-99.
7. Preidys S. Mokymosi stilių personalizavimas nuotolinėse studijose. [žiūrėta 2011 lapkričio 26 d.]. Prieiga internete: [http://www.esec.vu.lt/lt/images/stories/astos/Preidys\\_Sakalauskas.pdf](http://www.esec.vu.lt/lt/images/stories/astos/Preidys_Sakalauskas.pdf)
8. Jankevičius V. Atvirojo kodo duomenų bazių serverių analizė aptarnavimo paslaugų teikimo valdymo sistemos pagrindu. [žiūrėta 2011 lapkričio 24 d.]. Prieiga internete: [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D\\_20060526\\_113232-16907/DS.005.0.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D_20060526_113232-16907/DS.005.0.01.ETD)
9. Baronienė R. Paliulis E. Duomenų bazių valdymo sistemų tinkamumo biomedicininėms sistemoms įvertinimas // Jaunųjų mokslininkų darbai. ISSN 1648-8776. 2012, Nr. 5(38), p. 91-95.
10. Dieninis Ž. Biomedicininės informacinės sistemos realizacija internete. [žiūrėta 2012 sausio 14 d.]. Prieiga internete: [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D\\_20040922\\_110909-16825/DS.005.0.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D_20040922_110909-16825/DS.005.0.01.ETD)
11. Kučinskas M. Efektyvaus manipuliavimo duomenimis informacinėse medicinos sistemose tyrimas. Prieiga internete: [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D\\_20060525\\_222751-53672/DS.005.0.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D_20060525_222751-53672/DS.005.0.01.ETD)
12. Performance Comparison: Data Access Techniques. [žiūrėta 2011 gruodžio 12 d.]. Prieiga internete: <http://msdn.microsoft.com/en-us/library/ms978388>
13. Oracle Database Software Downloads. [žiūrėta 2011 lapkričio 22 d.]. Prieiga internete: <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>
14. SQL SERVER 2012. [žiūrėta 2011 lapkričio 28 d.]. Prieiga internete: <http://www.microsoft.com/sql/dataservices/faq.mspix>
15. PostgreSQL 8.3.17 Documentation. [žiūrėta 2011 lapkričio 28 d.]. Prieiga internete: <http://www.postgresql.org/docs/8.3/static/index.html>
16. DB2 Express. [žiūrėta 2012 sausio 4 d.]. Prieiga internete: [http://www-05.ibm.com/lt/db2/express\\_edition.html](http://www-05.ibm.com/lt/db2/express_edition.html)
17. Oracle Data Types. [žiūrėta 2012 sausio 14 d.]. Prieiga internete: [http://docs.oracle.com/cd/B28359\\_01/server.111/b28318/datatype.htm](http://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm)
18. Chapter 10. Data Types. [žiūrėta 2012 sausio 14 d.]. Prieiga internete: <http://dev.mysql.com/doc/refman/5.0/en/data-types.html>



19. Date processing in Informix Dynamic Server. [žiūrėta 2012 sausio 16 d.]. Prieiga internete: <http://www.ibm.com/developerworks/data/library/techarticle/dm-0510roy/>
20. Datatypes. [žiūrėta 2011 sausio 17 d.]. Prieiga internete: [http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/sql\\_elements001.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements001.htm)
21. UAB „MitSoft“. Duomenų bazių valdymo sistemų bei kitų kūrimo priemonių alternatyvų ir tendencijų švietimo ir mokslo srityse studija. [žiūrėta 2012 sausio 03 d.]. Prieiga per internetą: [http://www.ipc.lt/21z/duomenys/db/integralas/files/DBVS\\_studija\\_CD\\_81228%20final.doc](http://www.ipc.lt/21z/duomenys/db/integralas/files/DBVS_studija_CD_81228%20final.doc)
22. There's gold in those databases. [žiūrėta 2012 sausio 10 d.]. Prieiga internete: [http://www.statoo.com/en/publications/GoldInDB\\_SCWsep04Enbis.pdf](http://www.statoo.com/en/publications/GoldInDB_SCWsep04Enbis.pdf)
23. Insightful Miner 3 User's Guide. [žiūrėta 2011 sausio 12 d.]. Prieiga internete: <http://www.solutionmetrics.com.au/support/iminer30/uguide.pdf>
24. Mamčenko J. Duomenų gavybos technologijų taikymas išskirstytų serverių darbui gerinti. Daktaro disertacija. VGTU, 2008.
25. V. Špečkauskienė „Informacinio klinikinių sprendimų rengimo metodo sudarymas ir tyrimas,„ Daktaro disertacija. „Technologija“, Kaunas, 2011.
26. Klikūnaitė S. Duomenų gavybos metodai ryšių su verslo klientais sistemose. [žiūrėta 2012 sausio 18 d.]. Prieiga internete: [http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2009~D\\_20101125\\_190748-35252/DS.005.1.01.ETD](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2009~D_20101125_190748-35252/DS.005.1.01.ETD)
27. Efficient Retrieval of Text for Biomedical Domain using Data Mining Algorithm. [žiūrėta 2012 sausio 18 d.]. Prieiga internete: <http://thesai.org/Downloads/Volume2No4/Paper%2012-Efficient%20Retrieval%20of%20Text%20for%20Biomedical%20Domain%20using%20Data%20Mining%20Algorithm.pdf>
28. Compare SQL Server 2008 R2, Oracle 11G R2, PostgreSQL/PostGIS 1.5 Spatial Features. [žiūrėta 2012-09-15]. Prieiga per internetą: [http://www.bostongis.com/PrinterFriendly.aspx?content\\_name=sqlserver2008r2\\_oracle11gr2\\_postgis15\\_compare](http://www.bostongis.com/PrinterFriendly.aspx?content_name=sqlserver2008r2_oracle11gr2_postgis15_compare)
29. How Can Data Mining Help Bio-Data Analysis? [žiūrėta 2013 sausio 18 d.]. Prieiga internete: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.4890&rep=rep1&type=pdf>
30. Biomedical Databases and Data Mining. [žiūrėta 2013 sausio 18 d.]. Prieiga internete: [http://www.worldscibooks.com/etextbook/7772/7772\\_chap01.pdf](http://www.worldscibooks.com/etextbook/7772/7772_chap01.pdf)
31. PostgreSQL 9.1.2 Documentation. [žiūrėta 2012 sausio 6 d.]. Prieiga internete: <http://www.postgresql.org/docs/9.1/interactive/index.html>
32. MySQL Enterprise Edition. [žiūrėta 2012 sausio 6 d.]. Prieiga internete: <http://www.mysql.com/products/enterprise/>
33. DB2 database product documentation. [žiūrėta 2012 sausio 6 d.]. Prieiga internete: <https://www-304.ibm.com/support/docview.wss?uid=swg27009474>
34. Choosing an Encryption Algorithm. [žiūrėta 2012 sausio 25 d.]. Prieiga internete: <http://technet.microsoft.com/en-us/library/ms345262.aspx>
35. Getting Started with DB2 Express-C. [žiūrėta 2012 sausio 28 d.]. Prieiga internete: [http://public.dhe.ibm.com/software/dw/db2/express-/wiki/Getting\\_Started\\_with\\_DB2\\_Express\\_v9.7\\_p4.pdf](http://public.dhe.ibm.com/software/dw/db2/express-/wiki/Getting_Started_with_DB2_Express_v9.7_p4.pdf)
36. Renata Norbutaitė „Gerklų ligų diagnostikos informacinė sistema“. [žiūrėta 2012 gegužės 28 d.]. Prieiga internete: <http://www.renata.pusinskas.com/downloads/Bachelor%20thesis.pdf>
37. Sveikatos testas. [žiūrėta 2012 gegužės 29 d.]. Prieiga internete: <http://sveikas.lt/lt/sveikatos-testas/>

38. Ligos simptomai. [žiūrėta 2012 birželio 4 d.]. Prieiga internete: <http://ligosimptomai.lt/>
39. Tadas Vileiniškis Intelektualios duomenų gavybos algoritmų taikymo tyrimas Microsoft SQL Server 2005 Data Mining Designer priemonėmis. [žiūrėta 2012 birželio 4 d.]. Prieiga internete: [http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D\\_20090304\\_093218-90666/DS.005.0.01.ETD](http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D_20090304_093218-90666/DS.005.0.01.ETD)
40. Medical Information System MEDIALOG. [žiūrėta 2012 birželio 12 d.]. Prieiga internete: [http://www.armadaitgroup.com/en/products\\_and\\_services/software/medialog/](http://www.armadaitgroup.com/en/products_and_services/software/medialog/)
41. Maleš U. Benchmarking of various relational database management systems. 2010 [žiūrėta 2012 rugsėjo 9 d.]. Prieiga internete: <http://eprints.fri.uni-lj.si/1137/>
42. Data mining algorithms (analysis services - data mining). [žiūrėta 2012 gruodžio 8 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms175595.aspx>
43. Microsoft Association Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174916.aspx>
44. Microsoft Clustering Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174879.aspx>
45. Microsoft Decision Trees Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms175312.aspx>
46. Microsoft Linear Regression Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174824.aspx>
47. Microsoft Logistic Regression Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174828.aspx>
48. Microsoft Naive Bayes Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174806.aspx>
49. Microsoft Neural Network Algorithm. [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms174941.aspx>
50. Microsoft Sequence Clustering Algorithm [žiūrėta 2012 rugsėjo 9 d.]. prieiga internete: <http://msdn.microsoft.com/en-us/library/ms175462.aspx>
51. Andrius Laukaitis „XML dokumentu palaikymas laisvo kodo duomenų bazių valdymo sistemoje“. [žiūrėta 2012 rugsėjo 29 d.]. prieiga internete: [http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D\\_20090304\\_093912-24960/DS.005.0.01.ETD](http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2009~D_20090304_093912-24960/DS.005.0.01.ETD)
52. Weka 3: Data Mining Software in Java. [žiūrėta 2012 rugsėjo 30 d.]. prieiga internete: <http://www.cs.waikato.ac.nz/ml/weka/index.html>
53. Rapid Miner. [žiūrėta 2012 rugsėjo 30 d.]. prieiga internete: <http://rapid-i.com/>
54. Orange. [žiūrėta 2012 rugsėjo 30 d.]. prieiga internete: <http://orange.biolab.si/>
55. Oracle Data Mining. [žiūrėta 2012 rugsėjo 30 d.]. prieiga internete: <http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/index.html>
56. MiningMart. [žiūrėta 2012 rugsėjo 30 d.]. prieiga internete: <http://mmart.cs.uni-dortmund.de/>
57. Comparison of Database Management Systems. [žiūrėta 2012-09-25]. Prieiga per internetą: [http://www.virtualmv.com/wiki/index.php?title=Comparison\\_of\\_Database\\_Management\\_Systems](http://www.virtualmv.com/wiki/index.php?title=Comparison_of_Database_Management_Systems).
58. Bernatavičienė J. Vizualios žinių gavybos metodologija ir jos tyrimas. Daktaro disertacija. Technologijos mokslai, informatikos inžinerija (07T).2008. [žiūrėta 2012 gruodžio 15 d.]. Prieiga internete: [http://www.mii.lt/files/mii\\_dis\\_08\\_bernatavicienne.pdf](http://www.mii.lt/files/mii_dis_08_bernatavicienne.pdf)

59. Jiawei Han, Micheline Kamber, Jian Pei. Data Mining Concepts and Techniques. [žiūrėta 2012 gruodžio 15 d.]. Prieiga internete:

<http://www.informaticacalabria.altervista.org/Data%20Mining%20Concepts%20and%20Techniques%20%5BHan%20et%20al%5D%20Elsevier%20MK%203rd%20Ed%202011.pdf>

60. Popular Database Management Systems Overview. [žiūrėta 2013 gegužės 8 d.]. Prieiga internete: <http://dbconvert.com/overview.php>

61. A Comparative Study on the Performance of the Top DBMS Systems. [žiūrėta 2013 gegužės 8 d.]. Prieiga internete: <http://www.lacsc.org/papers/Paper21.pdf>

62. New DB-Engines Ranking shows the popularity of database management systems. [žiūrėta 2013 gegužės 8 d.]. Prieiga internete: [http://db-engines.com/en/blog\\_post/1](http://db-engines.com/en/blog_post/1)

63. Čekanauskaitė A., Pacientų, dalyvaujančių placebo kontroliuojamuose klinikiniuose vaistinio preparato tyrimuose Lietuvoje, informuotumas apie klinikinius tyrimus. Vilnius, 2013, 7 p. [žiūrėta 2013 gegužės 15 d.]. Prieiga internete: [http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2013~D\\_20130114\\_081944-94338/DS.005.0.01.ETD](http://vddb.laba.lt/fedora/get/LT-eLABa-0001:E.02~2013~D_20130114_081944-94338/DS.005.0.01.ETD)

64. Helsinkio deklaracija. [žiūrėta 2013 gegužės 15 d.]. Prieiga internete: [http://ismuni.lt/media/dynamic/files/84/helsinkio\\_deklaracija.pdf](http://ismuni.lt/media/dynamic/files/84/helsinkio_deklaracija.pdf)

65. What is biomedicine? [žiūrėta 2013 gegužės 15 d.]. Prieiga internete: <http://www.wisageek.com/what-is-biomedicine.htm>