

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

**Erlandas Barzda**

**INFORMACINIŲ TECHNOLOGIJŲ IŠTEKLIŲ  
DUOMENŲ BAZĖ**

Magistro darbas

**Vadovas  
doc. dr. B. Paradauskas**

**KAUNAS, 2005**

## SUMMARY

The subject of this master work is the internet information resource database. This work also handles the problems of old information systems which do not meet the new contemporary requirements. The aim is to create internet information system, based on object-oriented technologies and tailored to computer users' needs. The internet information database system helps computers administrators to get the all needed information about computers network elements and easy to register all changes into information database. Nowadays a lot of organizations use computer networks based on internet technologies. Often computer and their users are widely distributed. One factor of this information resource database is system mobility and flexibility. You can access database from any computer which has internet browser and could logon to the Internet. Of course data security is the important criterion of this information resource database. The users may access the permitted information only by the use of their own passwords. Another factor is data operative registering, "HelpDesk" table, data query from different layers and its gaining on time.

Internet technology database was analyzed and projected using UML notation. All documentation was valuable during database creation process but that will also help for future programmers update this system flexibility or change database structure using reverse engineering.

## TURINYS

IVADAS .....	7
1. IT IŠTEKLIŲ INFORMACINĖS SISTEMOS SUKŪRIMO AKTUALUMAS IR SISTEMOS UŽDAVINIAI .....	9
1.1 Kuriamos informacinės sistemos funkciniai reikalavimai .....	9
1.2 Anketinės apklausos rezultatai .....	10
1.3 CASE priemonės pasirinkimas .....	11
1.3.1 Metodologijos pasirinkimas .....	11
1.3.2 UML programinio paketo reikalavimai .....	12
1.4 Aukščiausiojo lygio kontekstinė diagrama .....	13
1.5 Veiklos panaudojimo atvejų modelis .....	14
1.6 Veiklos objektų modelis .....	15
1.7 Veiklos procesų diagramos .....	17
1.8 Kompiuterizuojamų panaudojimo atvejų modelis .....	18
1.9 Šiuo metu naudojamos Access duomenų bazės atitikimas kuriamai IT informacinei sistemai .....	20
1.10 Atviro kodo duomenų bazės .....	21
1.11 Duomenų bazės valdymo sistemų komponentai .....	22
1.12 Rinkoje siūlomų duomenų bazių valdymo sistemų lyginamoji charakteristika .....	23
1.13 Programinės bazės pasirinkimas .....	24
1.14 Informacijos sistemai keliami reikalavimai .....	24
2 PALIKUONINIŲ DB PAKARTOTINĖS INŽINERIJOS PROCESAS .....	26
2.1 Pakartotinės inžinerijos taikymas palikuoninės DB diagramai suformuoti .....	32
2.2 Palikuoninės Access DB duomenų migracija į MySQL DB .....	33
3. INTERNETINĖS DB PALAIKYMŲ PROGRAMINIO ĮRANKIO PROJEKTAVIMAS .....	35
3.1 Projektavimo proceso parinkimas .....	35
3.2 Išteklių valdymo sistemos funkciniai reikalavimai .....	36
3.3 Vartotojų sąsajos modelis .....	43
3.4 Projektuojamos duomenų bazės lentelių struktūros parinkimas .....	44
3.5 Panaudojimo atvejų realizacijos .....	46
3.6 Programinių klasių aprašymai .....	56
3.7 Duomenų lentelių aprašymas su duomenų valdymo programa „PHPmyAdmin“ .....	61
3.8 Sistemos fizinis išpildymo modelis .....	71
3.9 Sistemos testavimas .....	72
3.10 Vartotojo vadovas .....	72
3.10.1 Prisijungimas prie „informacinės sistemos“ duomenų bazės .....	72
3.10.2 Naujos darbo vietos registravimas .....	74
3.10.3 Sudėtinės įrangos registravimas .....	77
3.10.4 Pirktos įrangos registravimas .....	78
3.10.5 Detalių paskirstymas .....	79
3.10.6 Ataskaitų formavimas .....	80
3.10.7 Užsakymų, gedimų registravimas .....	81
3.11 Administratoriaus vadovas .....	82
4. SUKURTO INŽINERINIO PRODUKTO ĮVERTINIMAS .....	83
5. IŠVADOS .....	84
6. NAUDOTOS LITERATŪROS SĄRAŠAS .....	86
7. NAUDOTŲ SUTRUMPINIMŲ SĄRAŠAS .....	87
1 PRIEDAS. Anketos .....	88
2 PRIEDAS. Testavimo duomenys .....	91
3 PRIEDAS. Kompaktinėje plokštelėje pateiktų programų sąrašas .....	92
4 PRIEDAS. Access to MySQL, PhpMyAdmin programų išvesties rezultatai .....	93

## Lentelių sąrašas

1.1 lentelė UML modeliavimo programos .....	11
1.2 lentelė Atviro kodo duomenų bazių palyginimas.....	22
3.1 lentelė Programinės klasės.....	55
3.2 lentelė Darbo vieta .....	55
3.3 lentelė Įrenginiai.....	56
3.4 lentelė Vartotojai.....	56
3.5 lentelė Vartotojų valdymas .....	57
3.6 lentelė Periferija .....	57
3.7 lentelė Programinė įranga .....	58
3.8 lentelė DB .....	58
3.9 lentelė TableDelete.....	59
3.10 lentelė CDbox .....	60
3.11 lentelė DVD .....	61
3.12 lentelė Darbuotojai .....	61
3.13 lentelė HDD .....	62
3.14 lentelė Kompiuteriai .....	63
3.15 lentelė LAN plokštė.....	64
3.16 lentelė Nekomercinė įranga .....	64
3.17 lentelė SIS plokštė .....	65
3.18 lentelė Vaizduoklis .....	65
3.19 lentelė Buvimo vieta.....	66
3.20 lentelė Gedimai .....	67
3.21 lentelė Detalių paskirstymas .....	67
3.22 lentelė Periferija .....	68
3.23 lentelė Naujas Įrenginys .....	69
3.23 lentelė Užsakyta programinė įranga.....	69

## Paveikslų sąrašas

1.1 pav. Kuriamos sistemos uždaviniai.....	8
1.2 pav. Aukšto lygio kontekstinė diagrama .....	12
1.3 pav. Veiklos panaudojimų atvejų diagrama .....	13
1.4 pav. Veiklos sąveikų diagrama .....	14
1.5 pav. Veiklos objektų diagramos.....	15
1.6 pav. Kompiuterinės įrangos vartotojo problemų/užsakymų registravimo diagrama.....	16
1.7 pav. Išteklių apskaitos proceso veiklos diagramos .....	17
1.8 pav. Vartotojo registravimo ir įrangos priskyrimo veiklos diagramos.....	17
1.9 pav. Įrangos vartotojo problemų/užsakymų registravimo panaudojimo atvejų diagrama. .	18
1.10 pav. Išteklių apskaitos proceso panaudojimo atvejų diagrama.....	18
1.11 pav. Vartotojo registravimo panaudojimo atvejų diagrama .....	19
1.12 pav. Duomenų bazės valdiklio komponentai.....	21
2.1 pav. Palikuoninių DB programų reorganizavimas atvirkštinės inžinerijos pagalba.....	25
2.2 pav. Pagrindinis DB atvirkštinės inžinerijos metodologijos procesas .....	26
2.3 pav. Esysių ryšių diagramos pavyzdys .....	27
2.4 pav. Blokinė diagrama, apibendrinanti metodo pagrindinius žingsnius. ....	28
2.5 pav. Reliacinės schemos pradinis RTO grafas.....	31

2.6 pav. Galutinis RTO grafas po visų sąryšių ir jų kardinalumų nustatymo.	32
2.7 pav. Palikuoninės Access duomenų bazės diagrama	33
2.8 pav. Access to MySQL programos pasirinkimo meniu	34
2.9 pav. Duomenų bazės ir programos rašymo pasirinkimo langas	34
3.1 pav. Projektavimo proceso parinkimas	35
3.2 pav. Vartotojo sąsajos modelis	43
3.2 pav. phpMyAdmin programos pradinis puslapis	44
3.3 pav. Lentelės „Kompiuteriai“ įrašų formatų nustatymo forma	44
3.4 pav. Lentelės „Kompiuteriai“ įrašų redagavimo forma	44
3.5 pav. Lentelės „Kompiuteriai“ išorinių raktų nustatymo forma	44
3.6 pav. Duomenų bazės struktūros sąryšių diagrama	45
3.7 pav. Užsakymų registravimo analizės klasių diagrama	46
3.8 pav. Užsakymų registravimo sąsajos trasų diagrama	47
3.9 pav. Užsakymų registravimo programavimo trasų diagrama	47
3.10 pav. Formų sąsajų diagrama užsakymų registravimo funkcijomis atlikti	48
3.11 pav. Detali užsakymų registravimo diagrama	48
3.12 pav. Išteklių apskaitos panaudojimo atvejų diagramos realizacija	49
3.13 pav. Išteklių apskaitos vartotojo sąsajos trasų diagrama	50
3.14 pav. Išteklių apskaitos programavimo trasų diagrama	51
3.15 pav. Formų sąsajų diagrama redagavimo funkcijomis atlikti	52
3.16 pav. Detali vartotojo įrangos redagavimo diagrama	52
3.17 pav. Vartotojų registravimo panaudojimo atvejų diagramos realizacija	
3.18 pav. Vartotojų registravimo trasų diagrama	53
3.19 pav. Vartotojų valdymo programavimo trasų diagrama	53
3.20 pav. Detali vartotojų valdymo diagrama	54
3.21 pav. Pilna „informacinės sistemos“ klasių diagrama	55
3.22 pav. Duomenų bazės itdb_s struktūra	61
3.23 pav. Duomenų bazės lentelė „CDbox“	62
3.24 pav. Duomenų bazės lentelė „DVD“	62
3.25 pav. Duomenų bazės lentelė „Darbuotojai“	63
3.26 pav. Duomenų bazės lentelė „HDD“	63
3.27 pav. Duomenų bazės lentelė „Kompiuteriai“	64
3.28 pav. Duomenų bazės lentelė „LanPlok“	65
3.29 pav. Duomenų bazės lentelė „Nekommer_prog“	66
3.30 pav. Duomenų bazės lentelė „SisPlok“	66
3.31 pav. Duomenų bazės lentelė „Vaizduok“	67
3.32 pav. Duomenų bazės lentelė „bVieta“	67
3.33 pav. Duomenų bazės lentelė „gedimai“	68
3.34 pav. Duomenų bazės lentelė „pask_det“	68
3.35 pav. Duomenų bazės lentelė „periferija“	69
3.36 pav. Duomenų bazės lentelė „uzsak_irenginys“	70
3.37 pav. Duomenų bazės lentelė „uzsak_programa“	70
3.38 pav. Fizinis išpildymo modelis	71
3.39 pav. Pradinis „Informacinės sistemos“ prisijungimo langas	72
3.40 pav. Registracija patekti į duomenų administravimo puslapį	72
3.41 pav. Pagrindinis duomenų valdymo sistemos langas	73
3.42 pav. 2 srities išvestas meniu langas	74
3.43 pav. 2 srities užsakymų registravimo langas	74
3.44 pav. 2 srities ataskaitų modelio langas	74
3.45 pav. Vartotojo kompiuterinės darbo vietos duomenys	75
3.46 pav. Duomenų redagavimo sritis	75
3.47 pav. Vartotojo naujos kompiuterinės darbo vietos įvedimo langas	75

3.48 pav. Kompiuterinės darbo vietos redagavimo langas.....	76
3.49 pav. Kompiuterinės darbo vietos šalinimo langas .....	76
3.50 pav. CD įrenginių langas .....	76
3.51 pav. Sisteminių plokščių langas.....	77
3.52 pav. HDD įrenginių langas .....	77
3.53 pav. Vaizduoklių langas .....	78
3.54 pav. Periferinių įrenginių langas .....	78
3.55 pav. Programinės įrangos langas.....	78
3.56 pav. Pirktos įrangos langas .....	78
3.57 pav. Pirktos įrangos įvedimo langas.....	79
3.58 pav. Detalių paskirstymo langas .....	79
3.59 pav. Detalių įvedimo langas .....	79
3.60 pav. Darbuoto pasirinkimas iš „menu“ lango.....	80
3.61 pav. Atsakingo darbuotojo turimos įrangos detalios ataskaitos išvedimo langas.....	80
3.62 pav. Darbuotojo paieškos langas.....	80
3.63 pav. Darbuotojo darbo vietos išvedimo langas .....	80
3.64 pav. Laisvos įrangos ataskaitos langas .....	81
3.65 pav. Vartotojų prisijungimo langas .....	81
3.66 pav. Gedimų registravimo langas.....	81

## IVADAS

Vystantis informacinių technologijų techninėms ir programinėms priemonėms vis daugiau atsiranda galimybių vartotojus sujungti į vientisą informacinį tinklą. Pastaruoju metu vis labiau plečiasi TCP/IP pagrindu realizuoti tinklai, nereikalaujantys ypatingo vartotojo sąsajos funkcionalumo bei gebantys prisitaikyti prie įvairių duomenų perdavimo grandžių.

1992 metais įkurtas ISOC Interneto komitetas standartizavimo Internetinį TCP/IP protokolą ir nuo to laiko TCP/IP protokolo pagrindu realizuoti tinklai tampa pagrindiniu komunikavimo būdu išstumdami iki šiol dar naudotas failų sistemų FTP, Telnet, Gopher sistemas. Kartu su internetinio komunikavimo priemonėmis sukuriama vis daugiau duomenų bazių valdymo programinės įrangos.

Ypač akivaizdus internetinių duomenų bazių paplitimas (informacijos paieškos sistemos, internetiniai žodynai, enciklopedijos, įmonių nutolusių kompiuterinių darbo vietų informacijos priėjimo programinės įrangos). Kuriamos žmogiškųjų išteklių nuotolinio valdymo sistemos.

Didelės įmonės, organizacijos turinčios savo funkcinių padalinių skirtinguose geografiniuose taškuose neišvengiamai susiduria su personalo, techninės įrangos administravimo keblumais. Išauga galingesnių serverio pagrindu realizuotų duomenų bazių poreikis. Iš kitos pusės kliento dalies programinė įranga turėtų būti kiek įmanoma paprastesnė ir patogesnė. Lokaliuose tinkluose tokios problemos dar sprendžiamos naudojantis suvienodinta programine įranga. Kita vertus čia lengviau ir paprasčiau atlikti kompiuterinio tinklo administravimą. Situacija pasikeičia esant nutolusiems vartotojams. Sulėtėja vartotojų aptarnavimas, sunkiau atlikti kompiuterinių įrenginių apskaitą. Šiuolaikinių internetinių technologijų panaudojimas gali būti vienas iš šios problemos sprendimo variantų.

Darbe bus nagrinėjamas firmos „Informacinės sistemos“ duomenų bazės realizuotos Access aplinkoje atitikimas naujiems pateiktiems informacinės sistemos reikalavimams ir jos pakeitimas į naują internetinę duomenų bazę.

Apibrėžiame pagrindinius šio darbo tikslus:

- Įvertinti naujai kuriamos informacinių technologinių išteklių sistemos vartotojų funkcinius reikalavimus.
- Išanalizuoti palikuoninę ACCESS duomenų bazę ir įvertinti jos informacinį atitikimą naujiems funkciniais reikalavimams.

- Išanalizuoti atviro kodo duomenų bazių rinką ir jų valdymo įrankius labiausiai atitinkančius naujus IS reikalavimus.
- Atlikti palikuoninės DB migraciją į naują IS technologinę aplinką.
- Sudaryti nutolusios DB palaikymo programinę įrangą ir įvertinti sudarytos informacinės sistemos atitikimą funkciniais reikalavimams.

Šiame darbe nagrinėsime reliacinių duomenų bazių realizuotų SQL pagrindu valdymo sistemų panaudojimą taikant internetinių sistemų privalumus.

Šio darbo praktinis tikslas – remiantis šiuolaikinėmis CASE priemonėmis suprojektuoti ir aprašyti informacinių technologijų išteklių duomenų bazę reikalingą veiklos subjektų informacinių sistemų aprašymui bei analizei.

Informacinės sistemos duomenys bus saugomi interneto tinklo serveryje. Duomenų bazės administratorius, darbuotojai ar verslo partneriai turės galimybę lengvai disponuoti jiems prieinama informacija, gauti informaciją apie įmonę, registruoti užsakymus klientams, registruoti informacinių sistemų įrenginių gedimus, perduoti informacinius pranešimus kitiems registruotiems vartotojams nepaisant jų buvimo vietos. Šios sistemos aktualumas ypač didelis firmos informacinių išteklių valdytojams, žmogiškųjų išteklių personalo vadovams, techniniams darbuotojams, turto apskaitininkams.

Informacinėje sistemoje saugomi ir registruojami duomenys apie įmonės turimą techninę įrangą: kompiuteriai, jų dalys, periferiniai įrenginiai (spausdintuvai, skeneriai, tinklo šakotuvai ir pan.), programinė įranga. Įvesta informacija apie kiekvieną iš šių elementų - iš kokių sudėtinių dalių surinktas kompiuteris, kokia periferinė įranga prie jo prijungta, kokia programinė įranga įdiegta, kas yra atsakingas už techninę ir programinę įrangą. Taip pat kaupiama informacija apie elementų tarpusavio sujungimų fizinių grandžių charakteristikas, tinklo protokolus ir pan. Duomenų bazėje taip pat saugoma visų informacinės sistemos įrengimų įsigijimo informacija. Ši sistema taip pat gali būti naudinga kompiuterine įranga prekiaujančioms firmoms, ar teikiančioms kompiuterinės įrangos priežiūros paslaugas, interneto paslaugų tiekėjams turintiems savo atstovybes skirtinguose miestuose.

Darbe pateiktoms diagramoms kurti ir analizuoti naudojame šiuolaikinę CASE priemonę „Rational Rose Enterprise“

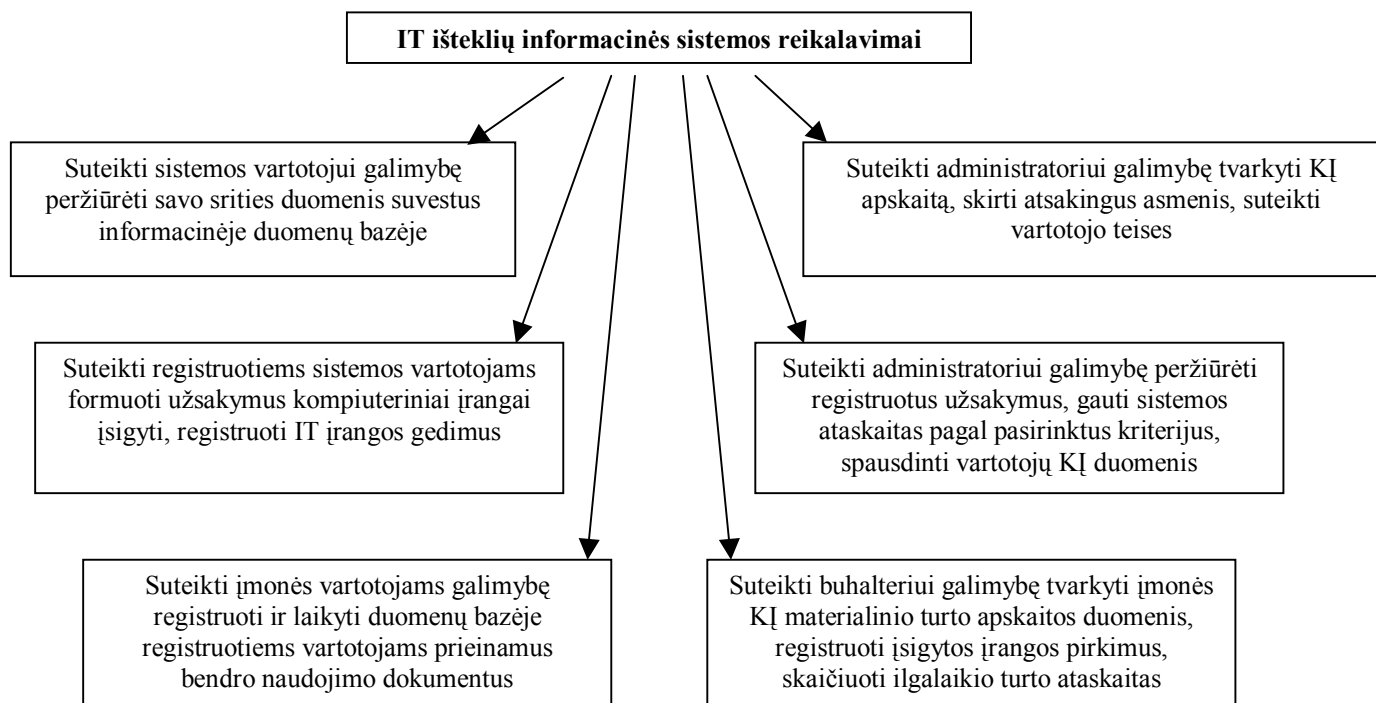


# 1. IT IŠTEKLIŲ INFORMACINĖS SISTEMOS SUKŪRIMO AKTUALUMAS IR SISTEMOS UŽDAVINIAI

Įmonėje „Informacinės sistemos“ IT išteklių apskaitai naudojama Microsoft Access duomenų bazė. Įmonės veiklos sritis- kompiuterinės įrangos priežiūra ir diegimas. tačiau augant aptarnaujamų darbo vietų skaičiui, plečiantis aptarnaujamų įmonių padaliniais, atsiranda centralizuoto administravimo būtinybė, išskyla glaudesnio bendravimo tarp KĮ sistemų administratoriaus ir kompiuterių vartotojų poreikis. Sparčiai plečiantis informacinei sistemai bei daugėjant išteklių tampa praktiškai neįmanoma nuolat ir sistemingai fiksuoti pasikeitimų, nepatogu, o kartais ir neįmanoma prieiti prie norimos informacijos. Yra pavojus prarasti duomenis arba jų dalį laikant keletą failo kopijų skirtinguose vietose ir neatnaujinant naujausių duomenų. Todėl būtų ypač patogu naudoti programinę informacinę sistemą, kuri leistų bet kuriuo metu gauti reikalingą informaciją ir ją valdyti tik autorizuotiems sistemos vartotojams. Šiai dienai yra prieinamos naujos DB, programinė bei techninė įranga leidžianti pereiti prie kokybiškai naujo IS vartotojų aptarnavimo.

Esant minėtoms aplinkybėms informacinių sistemų srityje kuriamai naujai informacinei sistemai nustatome 6 pagrindinius reikalavimus.

## 1.1 Kuriamos informacinės sistemos funkciniai reikalavimai



1.1 pav. Kuriamos sistemos reikalavimai

Apibrėžiant šiuos veiklos reikalavimus iš dalies buvo remtasi tokių įmonių kaip AB „Lietuvos Telekomas“, savo vidinėje struktūroje plačiai naudojančias IT technologijas, patirtimi. Tai

internetinio tinklo panaudojimas vartotojų kompiuterinės įrangos apskaitai vesti, vidinių vartotojų užsakymams formuoti, gedimų registravimui „HelpDesk“, bendro naudojimo dokumentų archyvui.

## 1.2 Anketinės apklausos rezultatai

Sistemos aktualumui patvirtinti buvo sudarytos anketos (Priedas Nr.1) ir naudojantis atsitiktine imtimi apklausti kelių firmų informacinių sistemų vartotojai bei jų administratoriai.

Atlikus apklausą su 20 kompiuterinės įrangos vartotojų bei 5 administratoriais išaiškėjo sekantys aspektai:

- a. Informacijos perdavimui tarp kliento ir administratoriaus 15 atvejų iš 20 naudojamas telefoninis ryšys.
- b. Visi apklausti administratoriai nesinaudoja jokia Internetu prieinama įrangos apskaitos sistema.
- c. Dauguma apklaustų vartotojų bei administratorių nėra patenkinti dokumentų pasiekiamumu.
- d. 90% vartotojų savo kompiuterinėje darbo vietoje turi internetinę prieigą.
- e. Visi administratoriai į klausimą „ar jums reikalinga informacinė sistema skirta kompiuterinės įrangos registravimui bei analizei“ atsakė „Taip“.

Pagal gautus anketinius rezultatus galima spręsti apie projektuojamos sistemos aktualumą. Subjektyviam faktoriui eliminuoti čia buvo galima panaudoti projektavimo proceso efektyvumo prognozavimo metodus (12). Tačiau dėl nedidelės apklausose pateiktų klausimų apimties subjektyvumo faktorius nebuvo įvertintas.

## 1.3 CASE priemonės pasirinkimas

### 1.3.1 Metodologijos pasirinkimas

Programinės įrangos projektavimui atlikti ir dokumentacijai sukurti naudosime UML (unifikuota modeliavimo kalba). UML tai standartinių metodų visuma skirta specifikuoti, vizualiai atvaizduoti, konstruoti bei dokumentuoti pasirinktą sistemą.

UML pasirinkimą programos modeliavimui lėmė tai, jog UML atstovauja rinkiniui, šiuo metu vienam iš geriausių, inžinerinių praktinių sprendimų modeliuoti kompleksines sistemas. UML yra labai svarbi objektiškai orientuotų programų projektavimo dalis. Tam, kad aprašyti projektuojamos sistemos dizainą, UML naudoja grafinę notaciją. UML panaudojimas padeda projektavimo grupėms bendradarbiauti, kuriant atskiras projekto dalis. UML pateikia ir sistemos dokumentaciją reikalingą ne tik kūrimo procese, bet ir atliekant sistemos priežiūrą pasibaigus visiems kūrimo etapams. Turint išsamų sistemos aprašymą ateityje lengviau atlikti sistemos atnaujinimą ar sistemos duomenų išgavimą.

Objektnių modeliavimo (OM) kalbų atsiradimas siejamas su 1970 metais, kuomet daugelis OM pradininkų eksperimentavo su skirtingais objektnio programavimo analizės ir dizaino metodais. Nuo 1989 iki 1994 OM kalbų skaičius pakilo nuo 10 iki daugiau nei 50. Dažniausiai vartotojai nelikdavo patenkinti viena ar kita modeliavimo kalba. Pradedant nuo 1990 metų atsiranda naujų metodų savyje jungiančių keleta ar keliolika jau ankščiau sukurtų OM. Taip susiformavo keleta ryškesnių metodų (6).

Pirmieji autoriai Booch, OMT, ir OOSE metodai, Grady Booch, Jim Rumbaugh, ir Ivar Jacobson buvo motyvuoti sukurti unifikuotą modeliavimo kalbą dėl 3 priežasčių:

1 šie metodai jau pakankamai išvystyti nepriklausomai vienas nuo kito, taigi buvo prasmė tęsti evoliuciją kartu negu atskirai.

2 unifikuojant semantiką ir notacijas buvo tikėtasi rinkos palankumo OO programų modeliavimo srityje.

3 buvo tikėtasi, kad metodų bendradarbiavimas pagerins visų trijų OO metodų našumą

Booch, Rumbaugh, and Jacobson pastangų dėka 1996 pasirodė UML 0,9 ir 0,91 dokumentacija.

Po 1996 tapo aišku, jog UML kūrimo kryptis pasiteisina, buvo įkurtas UML partnerių konsorciumas. Padedant ir kitoms organizacijos buvo suvienytos jėgos kuriant UML 1.0 specifikaciją.

Kiekviena UML diagrama suformuota taip, kad leistų kūrėjams ir vartotojams peržiūrėti programinę sistemą skirtinga perspektyva ir skirtingu abstrakcijos lygiu.

UML diagramos paprastai braižomos grafiniu redaktoriumi turi:

1. panaudojimo atvejų diagramas;
2. klasių diagramas;
3. bendradarbiavimo diagramas;
4. būsenų diagramas;
5. veiklos diagramas;
6. komponentų diagramas;
7. įrengimų diagramas.

### 1.3.2 UML programinio paketo reikalavimai

Kuriant sistemą yra svarbu ne tik projektavimo etapuose naudojamos abstrakcijos, bet ir vykdant sistemos priežiūros, atnaujinimo bei patobulinimo darbus pageidautina turėti tą pačią projektavimo priemonę. Taigi projektuojant sistemą racionaliau pasitelkti platesnių galimybių projektavimo paketus idant ateityje vykdant sistemos atnaujinimą ar papildymą būtų galima tikėtis daugiau naujovių iš projektavimo įrankių paketų.

Šiuolaikiniams UML programiniams modeliavimo įrankiams keliami reikalavimai:

1. UML 1.4 ar aukštesnės versijos palaikymas;
2. pagrindinių standartinių diagramų generavimas;
3. diagramų eksportas, spausdinimas;
4. kodo generavimas;
5. daugiau nei vienos programavimo kalbos palaikymas;
6. pilna dokumentacija;
7. atvirkštinės inžinerijos galimybė.

1.1 lentelė

UML modeliavimo programos

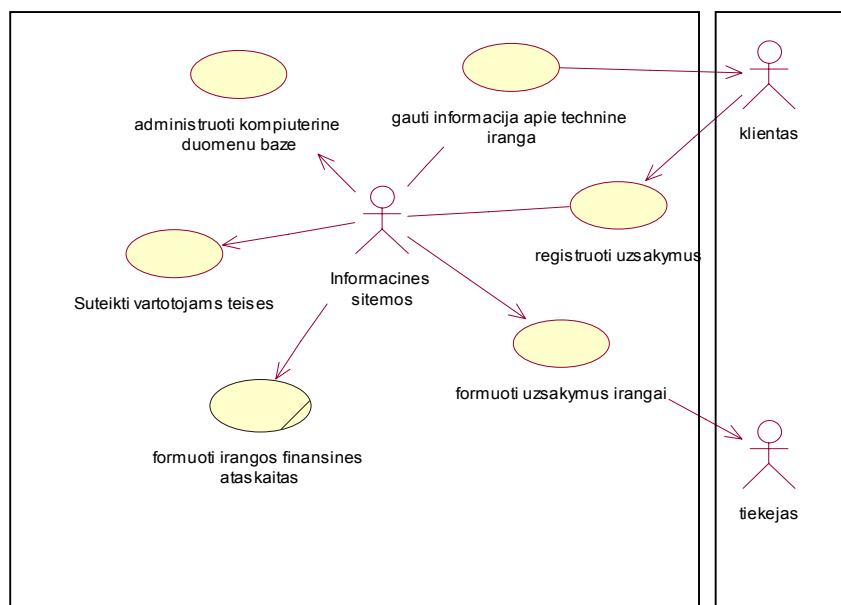
UML modeliavimo programa	Gamintojas	Internetinis adresas	UML diagramų palaikymas						Platforma	Programavimo kalba	Kodo generavimas	Atvirkštinė inžinerija	Paketo kaina
			Use Case	Klasės diagramos	Sekų diagramos	Būsenų diagramos	Veiklos diagramos	Diagramų savybės					
Rational Rose Enterprise	IBM	<a href="http://www.rational.com/rose">www.rational.com/rose</a>	Y	Y	Y	Y	Y	Y	Win	C++, ADA, VB, JAVA, Corba	Y	Y	2442\$
Enterprise Architect Professional	Sparx systems	<a href="http://www.sparxsystems.com">www.sparxsystems.com</a>	Y	Y	Y	Y	Y	Y	Win	C++, Delfi, VB, JAVA	Y	Y	149\$
MagicDraw UML Professional	MagicDraw	<a href="http://www.magicdraw.com">www.magicdraw.com</a>	Y	Y	Y	Y	Y	Y	Java VM	Java, C++, IDL	Y	Y	899\$
UML Studio	PragSoft	<a href="http://www.pragsoft.com">www.pragsoft.com</a>	Y	Y	Y	Y	Y	Y	Win	Java, C++, IDL	Y	Y	500\$

Nors Rational Rose paketas yra ženkliai brangesnis už kitų firmų UML objektinio modeliavimo įrankius, tačiau RR turi platų spektrą sudėtinių programų.

RR suite produktai: Requisite Pro, Soda, Test manager. Šie išplėstiniai RR produktai savo funkcionalumu yra turtingesni negu tarkim bet kurios versijos Enterprise Architect (4) paketo funkcijos. Tai sudėtiniai įrankiai projekto gyvavimo ciklo palaikymui, testavimui, atvirkštinės inžinerijos panaudojimui bei kodo programinio kodo generavimui. Taigi žvelgiant į sistemos funkcionalumo tobulinimą ateityje patogiausia būtų naudoti IBM firmos UML įrankį Rational Rose Enterprise.

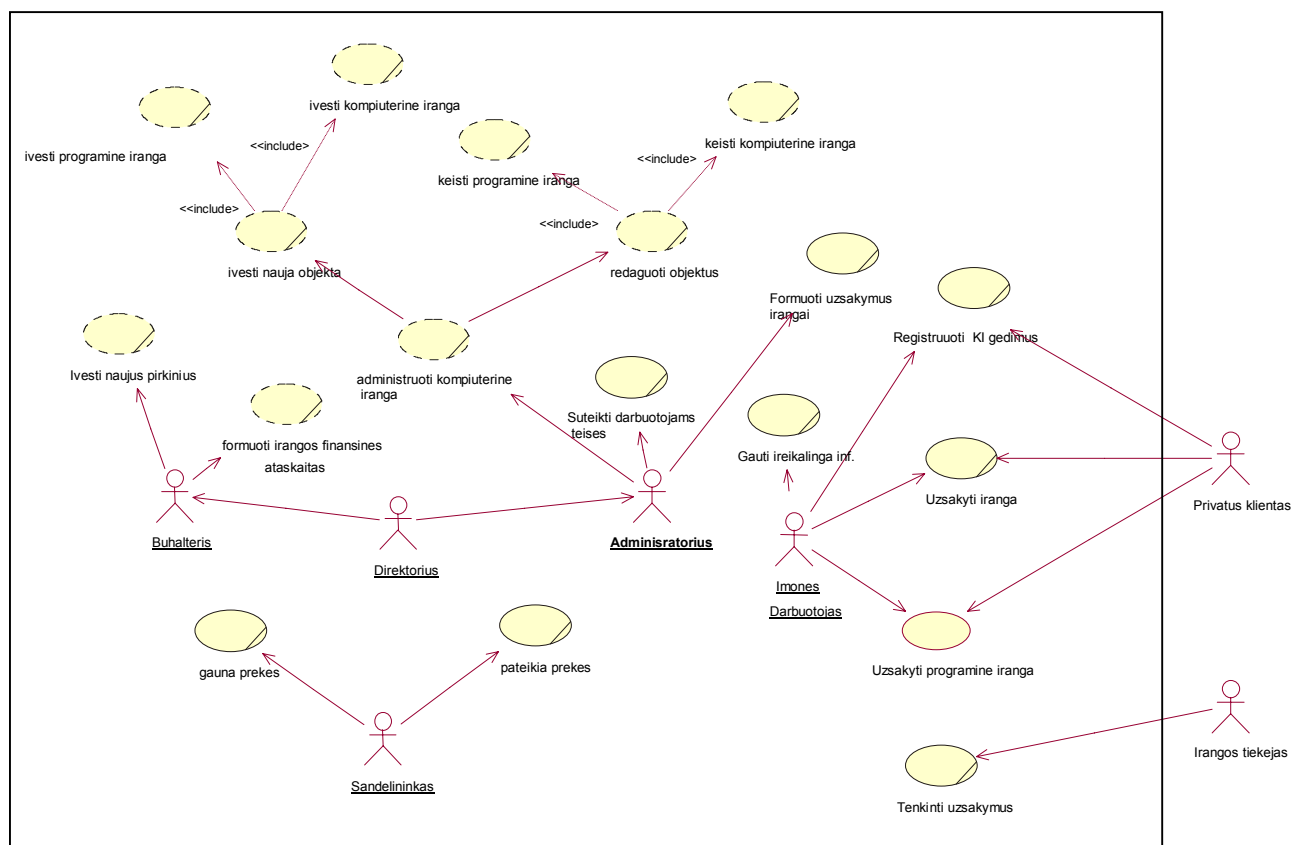
#### 1.4 Aukščiausio lygio kontekstinė diagrama

Šia diagrama pavaizduosime firmos „Informacinės sistemos“ veiklą bei ryšius su išoriniu pasauliu.



1.2 pav. Aukščiausio lygio kontekstinė diagrama

## 1.5 Veiklos panaudojimo atvejų modelis



1.3 pav. Veiklos panaudojimų atvejų diagrama

Šioje diagramoje atvaizduojame firmos „Informacinės sistemos“ vidinius vykdomus veiklos procesus. Detalizuotoje organizacijos struktūros diagramoje atsispindi vidinių sistemos dalyvių atliekami darbai.

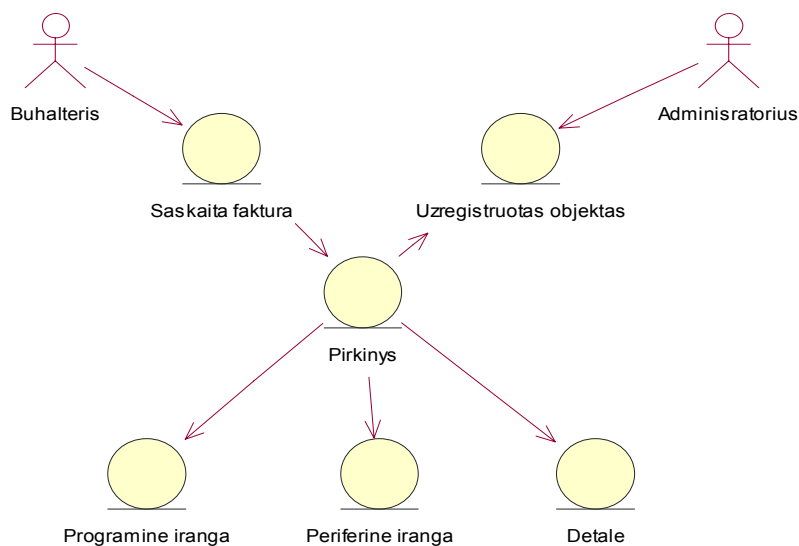
### Pagrindiniai aktoriai:

- **Administratorius**
  1. Įveda naujus elementus, registruoja pasikeitimus IS duomenų bazėje.
  2. Administruoja kompiuterinį tinklą.
  3. Administruoja vartotojų teises.
- **Klientas** – bet kuris aptarnaujamos įmonės darbuotojas registruoja techninės ir programinės įrangos gedimus, užsako papildomą įrangą, detales.
- **Būhalteris** – įmonės „Informacinės sistemos“ darbuotojas, registruoja naujus pirkimus, formotis IT turto likutinės vertės ataskaitas.

- Darbuotojas – įmonės „Informacinės sistemos“ KĮ vartotojas registruoja techninės ir programinės įrangos gedimus, užsako papildomą įrangą, detales.
- Tiekėjas – tenkina įmonės „Informacinės sistemos“ užsakymus.
- Sandėlininkas – gauna ir pateikia prekes pagal užsakymą.

## 1.6 Veiklos objektų modelis

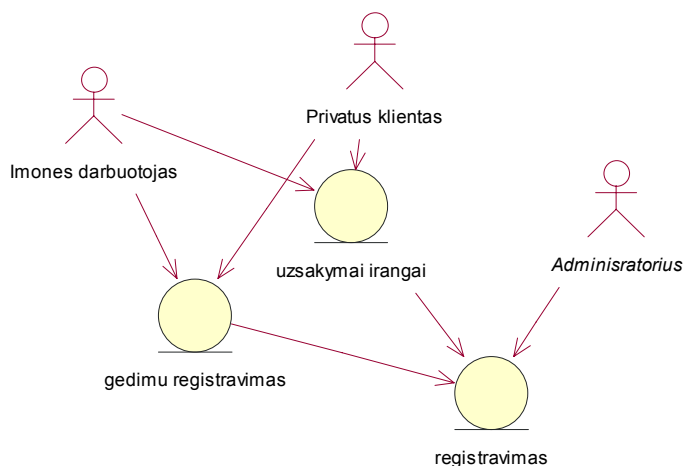
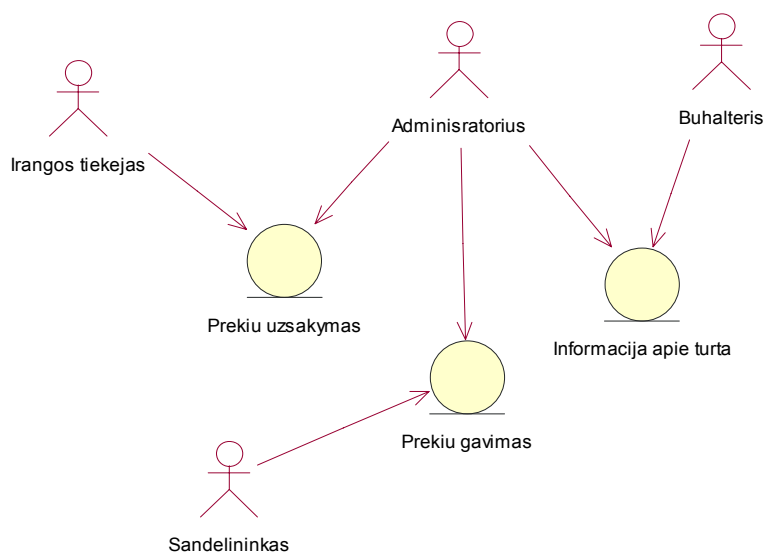
**Veiklos sąveikų diagrama:**



**1.4 pav. Veiklos sąveikų diagrama**

Veiklos sąveikų diagrama parodo kaip yra tarpusavyje susiję veiklos objektai ir aktoriai. Tvarkant informacinės sistemos išteklius dalyvauja du pagrindiniai aktoriai – administratorius ir buhalteris. Buhalteris išrašo gauna pirktų išteklių (detalių, programinės įrangos, periferinių įrenginių) sąskaitas faktūras ir fiksuoja juose esančią informaciją (pirkinio kainą, tiekėją, sąskaitos numerį, pirkimo datą). Administratorius turėdamas šią informaciją registruoja informaciją apie išteklius – kompiuterių, periferinės įrangos, programinės įrangos ryšius, vietą, kompiuterių sudedamąsias dalis.

## Veiklos objektų diagramos:



1.5 pav. Veiklos objektų diagramos

Pagrindiniai veiklos objektai (Pav. Nr.1.4), tvarkant informacinės sistemos išteklius yra įrangos/detalių tiekėjas, kuris gavęs iš sistemos administratoriaus nurodymą užsako prekes, Sandėlininkas, iš kurio administratorius gautas prekes pasiima. Administratorius užfiksuoja visą informaciją apie turimus išteklius ir suteikia reikalingus duomenis Buhalteriui.

Access duomenų bazėje (toliau palikuoninė DB) yra realizuota tik administratoriaus KĮ apskaitos, ir buhalterio duomenų suvedimo funkcijos. (1.3 pav.) Dokumentacijos apie palikuoninės DB projektavimą bei kūrimą nėra išlikusios, tai pagrindines IS funkcijas išgauname remiantis DB formų įrašų bei programinės įrangos vartotojų pagalba. CASE priemone naujai aprašome išteklių apskaitos procesą.

Pagrindinės palikuoninės DB kompiuterizuoti išteklių apskaitos funkcijos:



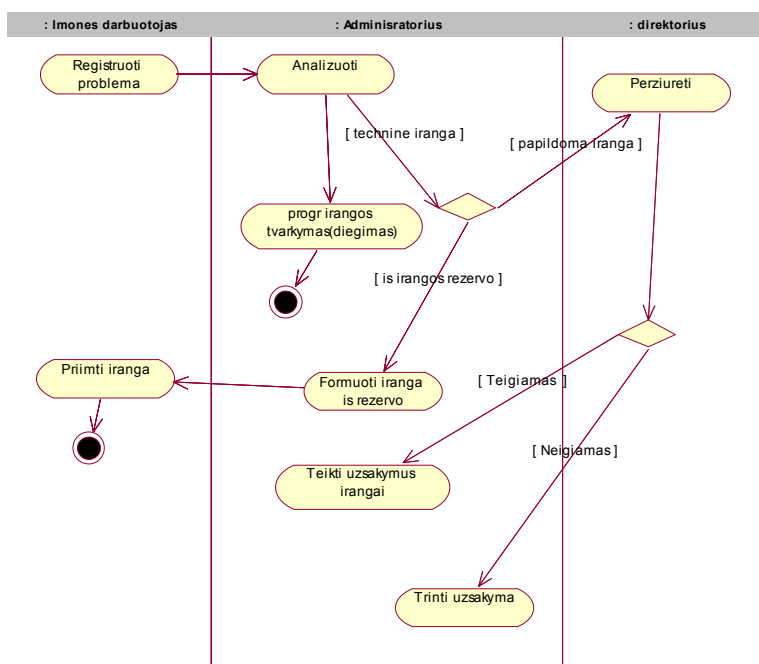
- Peržiūrėti duomenis – gali atlikti bet kuris prie sistemos prieinantis vartotojas
- Spausdinti ataskaitas – gali atlikti bet kuris prie sistemos prieinantis vartotojas
- Registruoti pirkimą – gali atlikti bet kuris prie sistemos prieinantis vartotojas
- Registruoti, redaguoti naują objektą, – gali atlikti bet kuris prie sistemos prieinantis vartotojas

Apima:

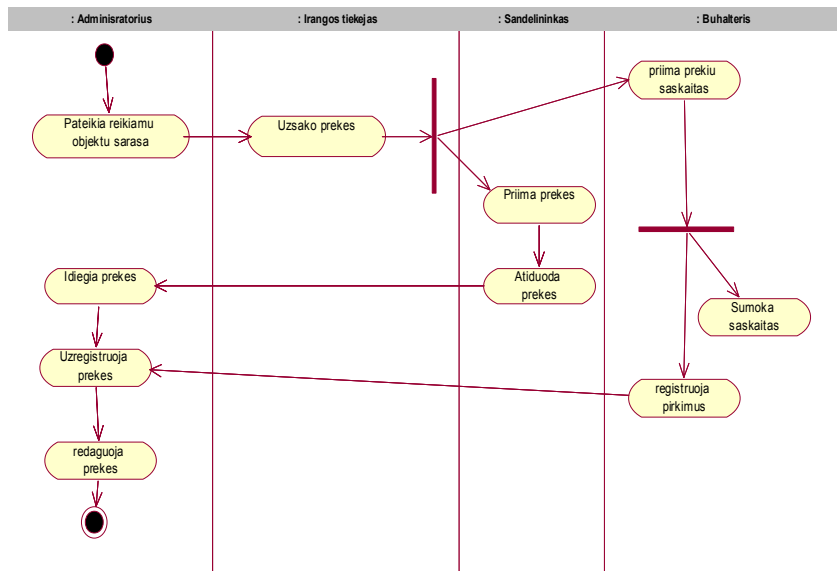
- Registruoti(redaguoti) įrangos konfigūraciją;
- Registruoti(redaguoti) buvimo vietą;
- Registruoti(redaguoti) programinę įrangą;
- Registruoti(redaguoti) atsakingą vartotoją.

## 1.7 Veiklos procesų diagramos

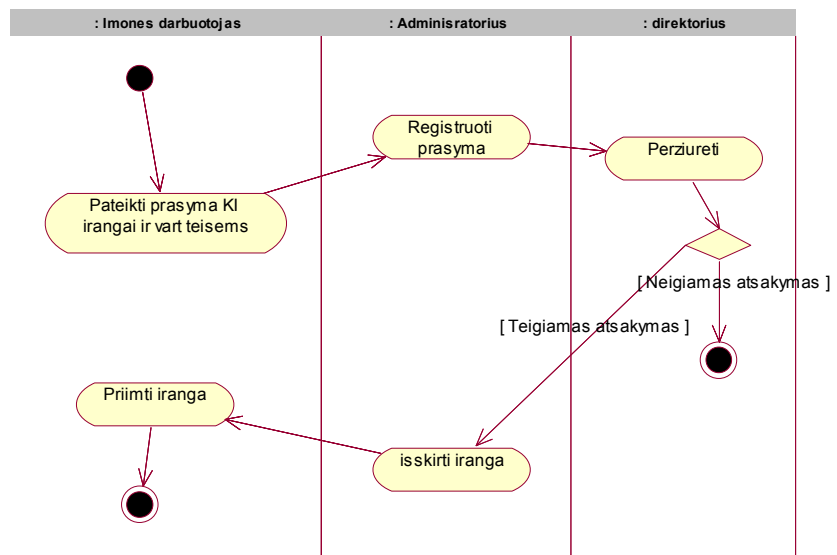
Veiklos panaudojimo atvejų modelyje veiklos diagramą naudosime veiklos procesų modeliavimui.



1.6 pav. Kompiuterinės įrangos vartotojo problemų/užsakymų registravimo veiklos diagrama.



1.7 pav. Išteklių apskaitos proceso veiklos diagramos



1.8 pav. Vartotojo registravimo ir įrangos priskyrimo veiklos diagramos

## 1.8 Kompiuterizuojamų panaudojimo atvejų modelis

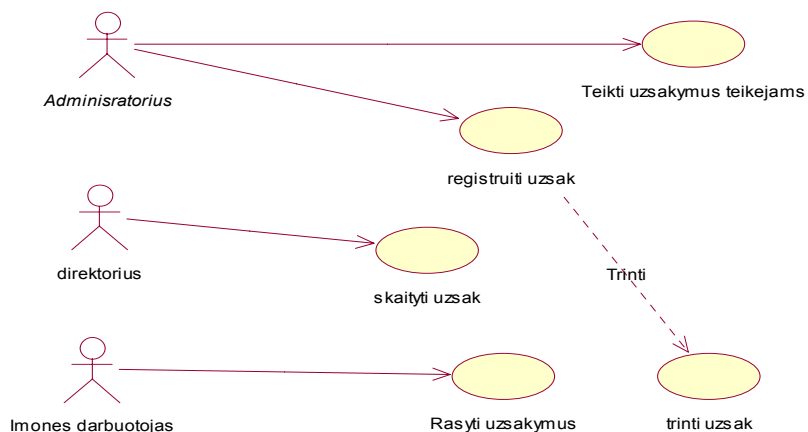
Išskirsime tuos veiklos panaudojimo atvejų modelius, kuriuos numatome kompiuterizuoti.

Kompiuterizuosime 3 veiklos panaudojimo atvejus:

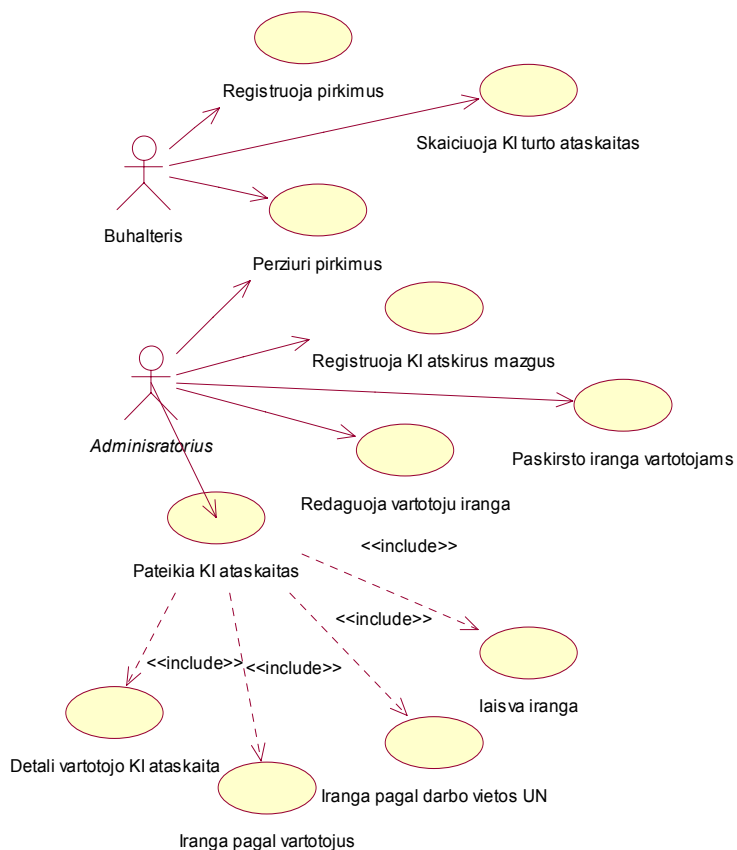
1. Įrangos vartotojo problemų/užsakymų registravimo
2. Išteklių apskaitos proceso (realizuotas palikuoninėje DB)

### 3 Vartotojų registravimo proceso

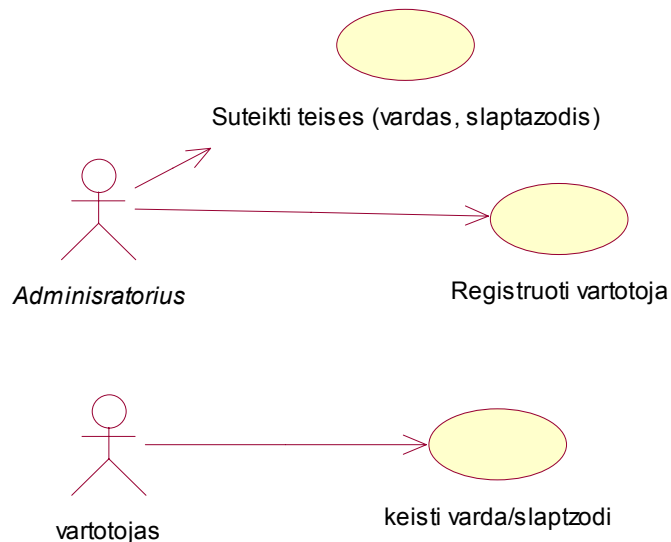
Sudarome aukšto lygio panaudojimo diagramas kiekvienam panaudojimo atvejui.



**1.9 pav. Įrangos vartotojo problemų/užsakymų registravimo panaudojimo atvejų diagrama.**



**1.10 pav. Išteklių apskaitos proceso panaudojimo atvejų diagrama**



**1.11 pav. Vartotojo registravimo panaudojimo atvejų diagrama**

## 1.9 Šiuo metu naudojamos Access duomenų bazės atitikimas kuriamai IT informacinei sistemai

Panagrinėsime pagrindinius MS Access DB privalumus ir trūkumus bei palyginsime DB atitikimą naujai projektuojamai IS.

### **Access DB privalumai:**

1. Patogi grafinė DB kūrimo bei redagavimo sąsaja.
2. Diegiama DB ir veikianti viename kompiuteryje. DB failas laisvai gali būti kopijuojamas į kitus kompiuterius, taip pat juose sėkmingai aktyvuojamas.
3. Išplėtos ir grafine sąsaja praturtintos ataskaitų formavimo galimybės.

### **Access DB trūkumai:**

1. Vienu metu su ta pačia Access DB negali dirbti keli vartotojai.
2. Nepakankama DB apsauga. Informacija gali būti pavokta nukopijuojant DB rinkmeną, ar pakeista koreguojant duomenis.
3. Access DB neturi serverio pagrindu sukurtos architektūros. Tai kliento pusės programinė įranga, norint dirbti lokaliame ar Interneto tinkle kiekvienas vartotojas turi turėti įdiegtą Access DB ir ja naudotis.

### **Pagrindiniai reikalavimai duomenų bazei:**

Remiantis sudarytais kompiuterizuojamais atvejais matome, jog tokiai informacinei sistemai suprojektuoti tinkamiausia būtų naudoti kliento –serverio technologiją bei internetinį tinklą.

Pagrindinė tam priežastys:

1. DB saugoma Interneto serveryje;
2. nesudėtinga klientinė dalis;

Išnagrinėjus palikuoninės DB tolimesnio panaudojimo galimybes iškyla dvi alternatyvos:

1. Naudoti tą pačią Access DB ją papildomai praplėtus. Prieiga per Internetą realizuojama naudojant ASP programavimo technologijas.
2. Pasirinkti naują nebrangią duomenų bazę ir į ją perkelti palikuoninės DB struktūrą ir duomenis.

Kadangi augant duomenų bazių reikalavimams ir plečiantis IS funkcionalumui apžvelgsime rinkoje platinamų DB charakteristikas.

Paieškos kriterijai:

1. Atviro kodo duomenų bazė, turinti komercinį palaikymą;
2. DB turinti pakankamai plačias programinių įrankių galimybes;
3. Palaikanti duomenų atstatymo procesus;
4. Sistema neturi priklausyti nuo vienos operacinės sistemos ar konfigūracijos.
5. Užtikrinti darbą su WEB tarnybine stotimi.

Prieš pasirinkdami DBVS apžvelgsime atviro kodo duomenų bazių aktualumą bei jų savybes.

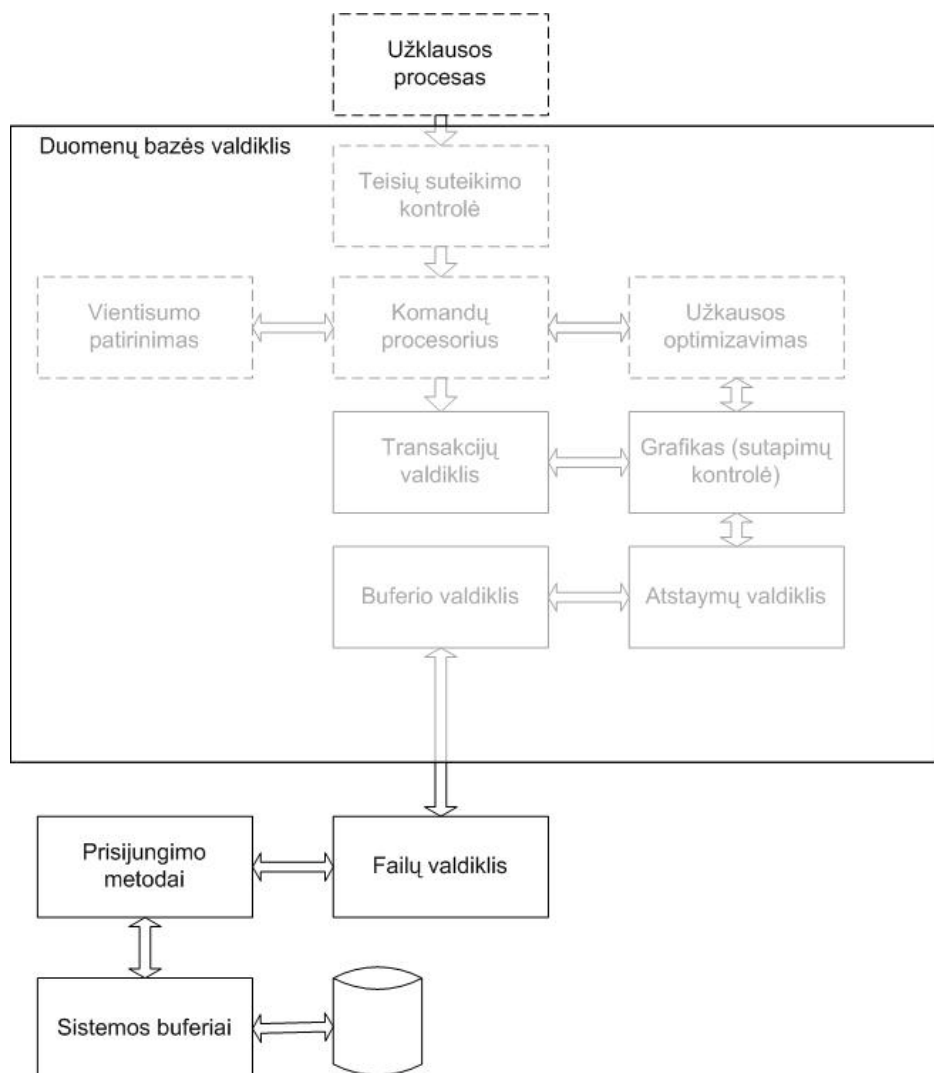
## 1.10 Atviro kodo duomenų bazės

Duomenų bazės jau egzistuoja daugiau kaip 30 metų. Įvairios komercinės duomenų bazės puikiai realizavo daugelį duomenų aprašymo ir saugojimo teorijų. Šios sistemos pateikia vartotojams modernią neatsiejamą valdymo sąsają, operacijų valdymą, palaikymą, duomenų kopijavimą, duomenų schemų atstatymą. Tačiau komercinių duomenų bazių įsigijimo kainos svyruoja nuo tūkstančio iki keliolikos tūkstančių litų. Mažoms ir vidutinėms firmoms ši kaina yra gana aukšta.

Kaip alternatyva komercinėms duomenų bazėms yra atviro kodo duomenų bazės (AKDB). Atviro kodo programų idėja yra vartotojui laisvai prieinamas programos šaltinis ir kodas. AKDB programų licenzijose paprastai yra teigiama, jog tai yra nemokamos platinimui programos nekomerciniams tikslams arba jos skirtos platinimui su kitomis programomis pasinaudojant ta pačia licenzija. Praktikoje plačiai paplitę ir atviro kodo programų komercinės licencijos. Tokių sistemų kaina yra kur kas mažesnė negu komercinių DB. Nebrangios AKDB įmonėms gali būti pakankamai patrauklios, jeigu tik jos sugeba užtikrinti reikiamą funkcionalumą ir yra konkurencingos komercinėms DB.

## 1.11 Duomenų bazės valdymo sistemų komponentai

Duomenų bazių valdymo sistemų (DBVS) architektūra yra pakankamai įvairi, priklauso nuo sukurtos programos poreikių. Taigi sunku aprašyti struktūrą charakteringą visoms DBVS. Tačiau dauguma DBVS dirba tais pačiais principais ir turi panašius komponentus. Duomenų bazės valdiklis susideda (1.12 pav.) iš dalies DBVS, kuri priima užklausas ir perduoda jas failų valdikliui atsakingam iš duomenų paėmimą iš fizinės duomenų saugyklos, paprastai kieto disko. Proceso metu duomenys patalpinami pagrindinėje atmintyje. Buferio valdiklis konvertuoja duomenis iš pagrindinės atminties į papildomąją.



1.12 pav. Duomenų bazės valdiklio komponentai

## 1.12 Rinkoje siūlomų duomenų bazių valdymo sistemų lyginamoji charakteristika

### 1.2 lentelė

Atviro kodo duomenų bazių palyginimas

DB pavadinimas	Internetinis puslapis	Platforma	Programavimo interfeisas	XML	Duomenų atstatymas	Duomenų kopijavimas	DB stabilumas	Duomenų apsauga	Trigerių palaikymas	Alternatyviųjų lentelių tipų palaikymas	Licenzija, mokeskis	Palaikymo mokeskis	SQL standarto palaikymas
PostgreSQL	<a href="http://www.postgresql.org">http://www.postgresql.org</a>	Linux, Solaris, HP-UX, AIX, IRIX, FreeBSD, OpenBSD, NetBSD, MacOS, SCO OpenServer, SCO Unixware, BeOS, BSDi, Compaq Tru64, QNX, Windows	ODBC, JDBC, C/C++, Embedded SQL (in C), Tcl/Tk, Perl, Python, PHP	-	+	-	aukštas	vidutinė	+	spec. lentelių tipai būdingi tik PostgreSQL	BSD (\$0)	atliekamas komercinis aptarnavimas	pliniai palaiko SQL-92 standartą
SAP DB	<a href="http://www.sapdb.org">http://www.sapdb.org</a>	Linux, Solaris, HP-UX, Compaq Tru64, AIX, Windows	ODBC, JDBC, C/C++, Precompiler (Embedded SQL), Perl, Python, PHP	+	+	-	vidutinis	vidutinė	+		GPL/LGPL (\$0, būtinas aptarnavimo kontraktas)	atliekamas komercinis aptarnavimas	palaiko didžią dalį SQL-92 standartą
My SQL	<a href="http://www.mysql.com">http://www.mysql.com</a>	xLinux, Solaris, HP-UX, MacOS, AIX, SCO, IRIX, FreeBSD, NetBSD, OpenBSD, Windows, BSDi, DEC, OS/2, Compaq Tru64	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	+	+	+	labai aukštas	aukšta	-	ISAM, BerkeleyDB, InnoDB, HEAP, Merge, Gemmini	GPL/ komercinė, \$0 (vidiniam naudojimui, darbai sąsajoje su Web serveriu)	atliekamas komercinis aptarnavimas	palaiko didžią dalį SQL-92 standartą
FireBird	<a href="http://firebird.sourceforge.net/">http://firebird.sourceforge.net/</a>	Linux, Solaris, FreeBSD, HP-UX, MacOS, Windows	ODBC, JDBC, C/C++, PHP	-	-	-	vidutinis	vidutinė	+		Borland public	atliekamas komercinis aptarnavimas	palaiko didžią dalį SQL-92 standartą
Interbase	<a href="http://www.borland.com/interbase">http://www.borland.com/interbase</a>	Windows, Linux, Solaris, Java™		+							Interbase 7.1 serveris - vartotojas (200\$) Interbase neribojamas vartotojų skaičius (4000\$)	atliekamas komercinis aptarnavimas	palaiko didžią dalį SQL-92 standartą

## 1.13 Programinės bazės pasirinkimas

Išnagrinėjus keletą atviro kodo DB charakteristikų, parinkome viena iš labiausiai tinkančių šiam atvejui duomenų bazių valdymo sistemų - MySQL. Nors MySQL neturi grafinės sąsajos funkcionalumo lyginant su Access, kiti privalumai yra ženklūs:

- Integruota su WEB serveriu naudojant programines kalbas;
- Reikalavimas vartotojo autentiškumo;
- Galima įdiegti su daugiau nei 20 operacinių sistemų;
- Didelės apimties DB;
- MySQL GPL licenzija;

Pasirenkame PHP programavimo kalbą. Pagrindiniai PHP kalbos parinkimo kriterijai:

- suteikia galimybes sukurti informacinės sistemos išteklių valdymo schemą valdomą per Internetą;
- pilnai suderinama su MySQL;
- pakankama įdiegti PHP WEB serveryje, ir iš bet kurios darbo vietos programa bus prieinama per interneto naršyklę neįdiegiant jokios papildomos programinės įrangos. (Tai reikėtų daryti pavyzdžiui kuriant sistemą su Delphi, Visual Basic ar kitomis kalbomis).

## 1.14 Informacijos sistemai keliami reikalavimai

### 1. Serverio dalies reikalavimai

#### **WWW serverio platforma RetHat Linux**

Įdiegta:

PHP 4 programa;

DBVS – MySQL 4 serveris.

### 2. Kliento dalies reikalavimai

**WWW naršyklė** – kurios pagalba būtų pasiekama Internetinė svetainė, skirta informacinės sistemos išteklių valdymui.

Kuriama sistema turi užtikrinti firmos duomenų privatumą, todėl yra būtina vartotojų autorizacija, kurią atlieka prisijungimo sistema, reikalaujanti prisijungimo vardo ir slaptažodžio.

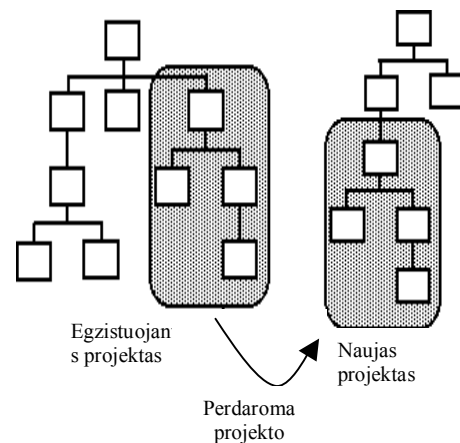
### 3. Vartotojo sąsajai keliami reikalavimai



- Naujo LAN mazgo įvedimas.
- Sąsaja pateikiama WWW naršykle, informacija suvedama formų pavidalu.
- Mazgų aprašymai.
- Duomenų išvedimas (ataskaitos).
- Mazgų buhalterinių lentelių skaičiavimas.
- Mazgų paieška pagal užduotą kriterijų.
- Pasirinktų duomenų spausdinimas.
- Vartotojų teisių administravimas.
- Vartotojų užklausų/ problemų registravimas.

## 2 PALIKUONINIŲ DB PAKARTOTINĖS INŽINERIJOS PROCESAS

Ankščiau programų atvirkštinė inžinerija nebuvo tokia aktuali, kokia yra dabar. Šiandien tapo aišku, jog sena DB architektūra sunkiai įgyvendina naujus reikalavimus dizaino, veikimo funkcionalumo atnaujinimui. Palikuoninių DB programų kodai tapo sunkiai suprantami, jau vien dėl to, jog dalis jų parašyti senesnėmis, šiuo metu retai naudojamomis programavimo kalbomis ir dėl specialistų trūkumo šių programų palaikymas (atnaujinimas) yra gana brangus, dažniausiai neduodantis laukiamo rezultato. Iškyla būtinybė iš esmės pakeisti jau egzistuojančias DB programas naujomis. Tam, kad neprarasti duomenų ir jų struktūrų reikalinga kažkoku būdu išgauti formalias specifikacijas iš palikuoninės programos kodo. Tokiu atveju taikoma atvirkštinės inžinerijos metodologija.

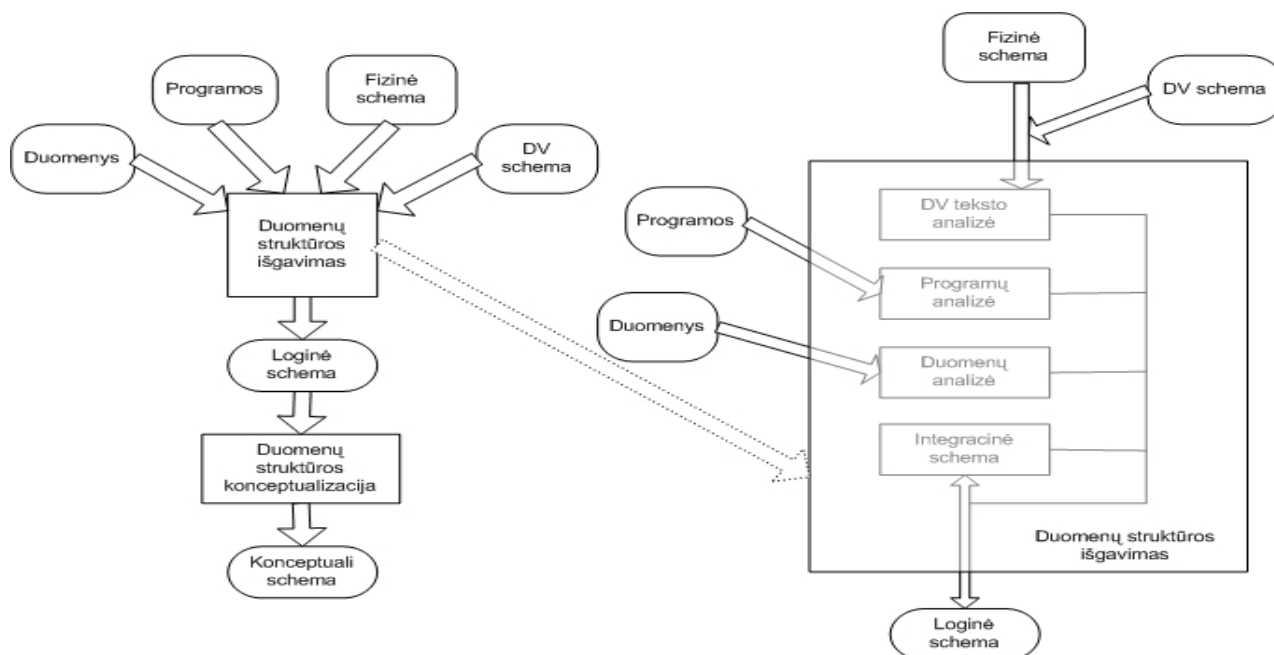


**2.1 pav. Palikuoninių DB programų reorganizavimas atvirkštinės inžinerijos pagalba.**

Organizacijos renkasi sistemos reorganizaciją, kas reiškia jų egzistuojančių informacijos sistemų atnaujinimą, situacijose, kuriose tai kainuoja pigiau nei sistemos pakeitimas. Bendru atveju, atvirkštinė inžinerija (AI) gali būti apibūdinta kaip procesas atskleidžiantis kaip sistema dirba. Vadinasi būtina nustatyti ir suprasti visus esamos sistemos komponentus bei ryšius tarp jų. AI padeda sutvarkyti ar perdaryti sistemos komponentus, kad padidinti jų funkcionalumą ir našumą. AI proceso išėiga taip pat gali būti pakartotinai panaudota kaip verslo architektūros komponentų šaltinis. Vienas tokių svarbių informacijos sistemos komponentų yra jos duomenų bazė. Todėl, duomenų bazės atvirkštinė inžinerija yra būtina, kad semantiškai praturtinti ir dokumentuoti duomenų bazę bei išvengti didelio duomenų kiekio, saugomo esamoje liktinėje duomenų bazėje, praradimų, jei esamos duomenų bazės savininkai nori taikyti reorganizaciją, eksploatuoti ir sutvarkyti duomenų bazės dizainą.

Nagrinėjamos DB atvirkštinės inžinerijos metodologija (1) yra padalyta į du pagrindinius procesus.

1. duomenų struktūros išgavimas
2. duomenų struktūros konceptualizacija.



## 2.2 pav. Pagrindinis DB atvirkštinės inžinerijos metodologijos procesas

Pradinė DBAI metodologijos fazė susideda iš pilnos DV schemos išgavimo įtraukiant visas žinomas ir numanomas duomenų struktūras bei ryšius. Kai kurios DV, pagrinde DBVS pateikia aprašymus ir pilnas duomenų schemas. Tačiau didesnės problemos atsiranda nagrinėjant standartinius failus kuriems kaip taisyklė nėra jokių egzistuojančios programos struktūros aprašymų. Kiekvienos programos dalies aprašymų formuluočių analizė pateikia dalinę programos failų ir įrašų struktūrą. Duomenų įrašų ir laukų aprašymai pateikia svarbią, tačiau dalinę informaciją.

Jeigu duomenų konstrukcija ar ryšiai nėra aiškiai ir tiksliai apibrėžti DB schemoje, daugumoje atvejų jie yra išversti į procedūrinius programų kodus ir pasiskirstę per visas programas. Schemos išgavimas tokiu atveju nėra tikslus ir šiai kompleksinei problemai spręsti iki šiol nėra sukurta jokių deterministinių metodų. Kruopšti informacinių šaltinių analizė (procedūrų turiniai, dokumentacija, duomenų turiniai ir taip toliau) pastiprina šių specifikacijų akivaizdumą.

DV teksto analizė: analizuojant duomenų struktūros aprašymus išgauti detalią duomenų konstrukciją ir ryšius, kas leistų nustatyti pradinę loginę schemą.

Programų analizė: analizuojant procedūrinius kodus nustatyti numanomas duomenų struktūras.

Duomenų analizė: analizuoti duomenis, kad surasti duomenų pavyzdžius parodančius duomenų struktūras bei savybes. Taip pat patikrinti hipotezes.

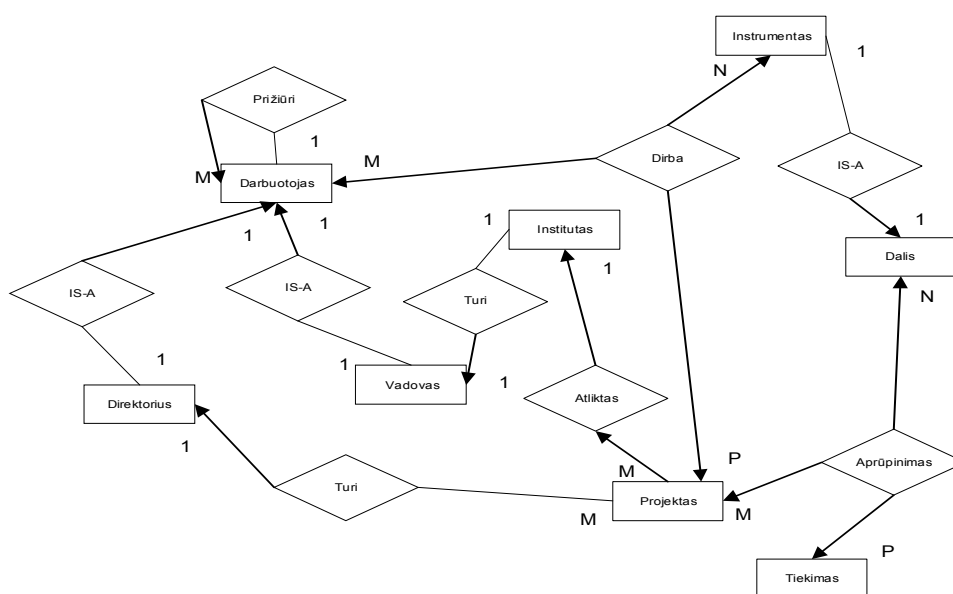
Schemų integracija: sujungti schemas nustatytas ankstesniuose etapuose.

Galutinis šios fazės produktas yra išbaigta loginė schema, kuri savyje turi tiek tikslas, tiek numanomas struktūras bei ryšius.

Apžvelgę programų atvirkštinę inžineriją, norint ateityje išvengti probleminių situacijų sprendimų svarbu užtikrinti sekančius etapus:

1. Projektuojant DBVS svarbus kriterijus programinės įrangos palaikymo tęstinumas;
2. Programavimo dokumentacijos sukūrimas ir tobulinimas;
3. Šiuolaikinių programinių paketų naudojimas.

Kalbant apie DBAI tradiciškai, duomenų analizės fazės rezultatas yra esybių-ryšių (ER) (11) diagrama (2.3 pav.), kuri yra transformuojama į ryšius reliaciniame modelyje.

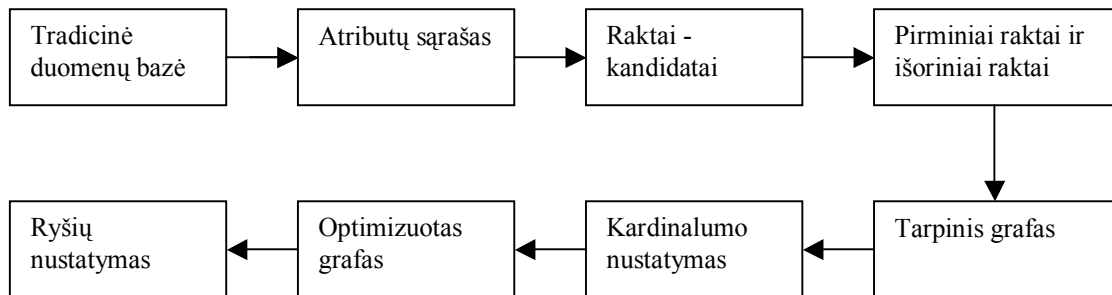


2.3 pav. Esių ryšių diagramos pavyzdys

Kaip nustato tradicinis reliacinis modelis, esybės ir keli sąryšiai sukuria ryšius. Šie sąryšiai apima ryšius su atributais, „daug su daug“ ( $M:M$ ) ir  $n$ -arius ( $n \geq 3$ ) sąryšius (tokie kurie jungia tris ar daugiau esybių). Atitinkamas ryšys turi kiekvienos iš dalyvaujančių esybių pirminį raktą ir papildomus atributus, jei tokių yra. Du likę sąryšių tipai, „vienas su vienu“ (1:1) ir „daug su vienu“ ( $M:1$ ) be atributų, iššaukia pridėjimą atributo(u), kuris suformuoja vieno ryšio su kitu pirminį raktą. Vėliau atributas(ai) formuoja išorinį raktą ir yra traktuojamas kaip kiti atributai originaliai esantys ryšyje, į kurį yra įtraukiami. Tačiau, toks duomenų bazės modelio supratimas paprastai prarandamas, kai netenkama projektuotojų. Tokiu atveju, tampa labai sunku (beveik neįmanoma) prižiūrėti ir derinti esamos liktinės duomenų bazės schemas, taigi atvirkštinė inžinerija yra būtina. Atvirkštinės inžinerijos procesas išgauna konceptualią schemą iš paprastos schemas. Tikslas yra išgauti ir sužinoti kaip galima

daugiau būtinos informacijos apie konceptualų modelį, kas leistų atlikti atvirkštinę inžineriją liktinei duomenų bazei.

Vienas iš reinžinerijos būdų (11) apima detalų esamos liktinės duomenų bazės tyrinėjimą, kad nustatyti pirminius ir išorinius raktus visuose duotuose ryšiuose. Baziniai žingsniai apibendrinti blokinėje diagramoje (2.4 pav.)



**2.4 pav. Blokinė diagrama, apibendrinanti metodo pagrindinius žingsnius.**

Išgavimo proceso metu, pirminės duomenų bazės charakteristikos yra tiriamos siekiant nustatyti ir patikrinti kiekvieno ryšio galimą raktą(us), kad nustatyti jo pirminį raktą. Vėlesnis sprendimas priklauso nuo domenų ir atributo(u) verčių galimame rakte palyginimo su kitais atributais visuose esamuose ryšiuose, kad nustatyti, kuris duoto ryšio galimas raktas turi atitinkamą išorinį raktą(us) kituose ryšiuose. Visą šitai daroma todėl, kad mes nagrinėjame homonimus ir sinonimus. Pirmasis pasitaiko, kai atributai tokiu pačiu vardu turi skirtingas reikšmes. Antrasis yra galimas, kai atributai su ta pačia prasme turi skirtingus vardus. Kitais žodžiais, nėra būtina turėti atributus, kurie turi vienodus vardus, lygiareikšmius semantiškai, ir sistema turi atsižvelgti į šiuos du vardų konfliktų tipus.

Informacija apie galimus ir išorinius raktus yra būtina ir pakankama, kad nubraižyti grafą atitinkantį ER diagramą. Jis rodo visus elementarius ir binarinius sąryšius, kurie egzistuoja tarp ryšių reliacinėje schemoje. Jo viršūnės yra ryšiai, o briaunų, jungiančių dvi viršūnes, skaičius priklauso nuo atveju, kai vieno ryšio pirminis raktas yra kaip išorinis raktas kitame ryšyje, skaičiaus, viena briauna vienam atvejui. Po to nustatomi kiekvienos nuorodos, pavaizduotos minėtame grafe, minimalūs ir maksimalūs kardinalumai. Tačiau, kai kurias nuorodas reikia pašalinti iš grafo, kadangi jos vaizduoja papildomą informaciją apie išorinius raktus. Vėlesnės informacijos šaltinis yra išoriniai raktai susiejantys simetrines ir tranzityvines nuorodas. Dėl šios priežasties, tai yra esminis žingsnis, kad optimizuoti grafą nustatant ir šalinant visą papildomą informaciją susietą su išoriniais raktais.

Paskutinis proceso žingsnis apima (kai įmanoma) žymų pradinio conceptualaus projekto pagerinimą, pasirenkant sąryšius su atributais,  $M:M$  ir  $n$ -arius sąryšius, ir nustatant *is-a* nuorodas. Kad nustatyti įrašus, kurie bus įtraukti į Raktus-Kandidatus, reikia išanalizuoti duoto ryšio visas galimas atributų kombinacijas Pavyzdžiui, kad nustatyti galimus raktus bet kokio ryšio, sakykime  $R$ , reikia atsižvelgti į visas galimas ryšio  $R$  atributų kombinacijas. Tačiau, siekiant patobulinti procesą yra pritaikoma keleta euristicų. Visos turimos omenyje euristicos remiasi apibrėžimu, kad galimas raktas yra minimali atributų aibė, kuri unikalčiai identifikuoja kiekvieną įrašą duotame ryšyje. Tuo remiantis, mes pradėdame tikrindami pavienius atributus, du atributus kartu ir t.t. Kita euristika išvedama iš apibrėžimo sakančio, pašalinam iš nagrinėjimo kiekvieną poaibio kombinaciją, kuriai jau yra nustatytas galimas raktas.

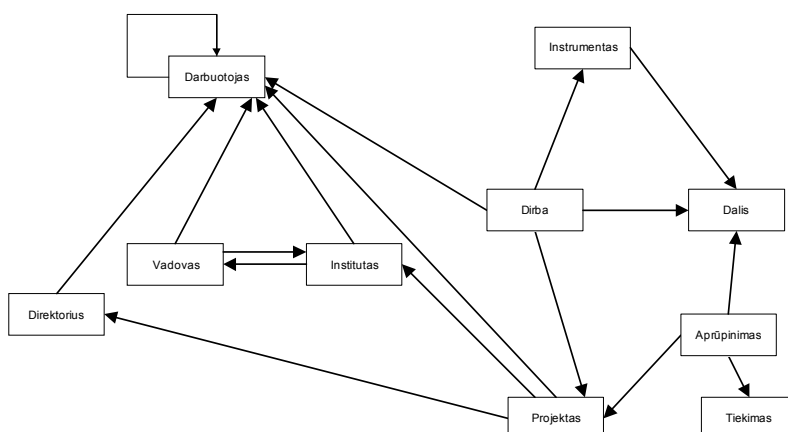
Tiksčiau kalbant, kad išgauti ryšio  $R$  galimus raktus, nagrinėjame ryšio  $R$  atributų aibės visų galimų poaibių aibę, žymimą  $P(R)$ . Visų galimų poaibių aibės elementai yra aibės ir mes domimės visais visų galimų poaibių aibės elementais, išskyrus tuščią aibę. Visų galimų poaibių aibė veda prie tinklinės struktūros, kur viršūnės yra visų galimų poaibių aibės elementai ir briaunos jungia dvi viršūnes remiantis vieno visų galimų poaibių aibės elemento įtraukimu į kitą.

Naudojamas Kandidatų –raktų nustatymo algoritmas nustato visus įvedamo ryšio  $R$  galimus raktus. Pagrindinė idėja yra patikrinti ar keli atributai turi unikalias vertes kiekviename ryšio  $R$  įrašė. Atributų aibė  $s$  yra pripažįstama kaip galimas raktas ryšio  $R$  jei ir tik jei aibė  $s$  yra minimali ir kiekvienas įrašas ryšys  $R$  turi unikalias vertes atributų kombinacijoms aibėje  $s$ , kuri negali būti tuščia.

Minimalus ir maksimalus algoritmo apskaičiuotų reikšmių skaičius yra  $\sum_{i=1}^n c_i$  ir  $\sum_{i=1}^n 2^{c_i}$ , atitinkamai, kur  $c_i$  yra ryšio  $R_i$  laipsnis (atributų skaičius) ir  $n$  yra ryšių skaičius duotoje reliacinėje schemeje. Apatinė riba yra pasiekama tuo atveju kai kiekvienas vienetinis atributas yra ryšio galimas raktas. Viršutinė riba yra pasiekama, kai kiekvienas ryšys turi vieną galimą raktą, kuris yra visų jo atributų kombinacija.

Po to kai visi galimi raktai yra nustatyti, sekantis žingsnis yra ištirti jų būvimą išoriniais raktais kituose ryšiuose. Kad šitai pasiekti, realizuojamas išorinių raktų nustatymo algoritmą, kuris randa kiekvienam ryšiui, kurio galimi raktai buvo rasti kandidatų raktų- nustatymo algoritme, atstovaujantį išorinį raktą(us) bet kuriuose duotose ryšiuose. Šis žingsnis yra būtinas, kad apsispręsti dėl nuorodų tarp duotų ryšių, įskaitant paveldėjimo, agregavimo, apibendrinimo nuorodas.

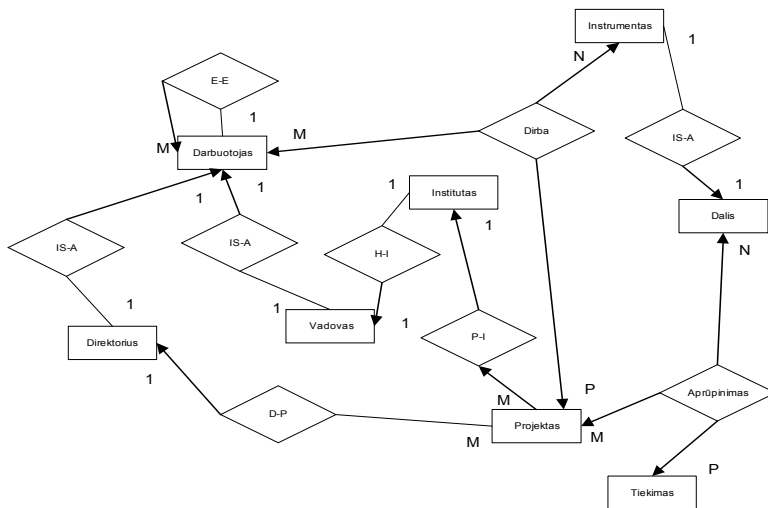
## Sąryšinis tarpinis orientuotas grafas



2.5 pav. Reliacinės schemos pradinis RTO grafas.

Turint išorinius raktus Informacijos yra pakankama, kad vykdyti atvirkštinės inžinerijos procesą. Naudojant šią informaciją, gali būti išgautos visos galimos nuorodos tarp ryšių, pavaizduotų duotoje reliacinėje schemoje. Šios nuorodos formuoja pagrindą likusiems žingsniams, kurie veda prie reikalingo IER modelio. Viršūnės RTO grafe yra ryšiai ir dvi viršūnės sujungtos nuoroda, parodyti, kad išorinis raktas ryšyje, kuris atitinka pirmą viršūnę vaizduoja pirminį raktą ryšio, kuris atitinka antrą viršūnę. Viršūnės ir nuorodos yra vaizduojamos mažais stačiakampiais ir orientuotomis rodyklėmis, atitinkamai.

Taip pat yra būdas kardinalumų RTO grafe nustatymui. Bendru atveju nuorodos RTO grafe (2.5 pav.) pagal savo kardinalumą yra klasifikuojamos į dvi grupes, 1:1 ir  $M:1$ . Šis klasifikavimas remiasi RTO grafo apibrėžimu, kuriame nuoroda yra nukreipta iš  $R_2$  į  $R_1$ , kad atspindėti ryšio  $R_1$  pirminio rakto buvimą kaip išorinio rakto ryšyje  $R_2$ . Pagal apibrėžimą, pirminis raktas privalo turėti unikalią vertę kiekvienam įrašui ryšyje  $R_1$ , o įrašų aibė iš ryšio  $R_2$  gali turėti keletą reikšmių atitinkamam išoriniam raktui. Remiantis šituo, sąryšiai tarp  $R_1$  ir  $R_2$  tampa žymėjimu iš  $R_2$  į  $R_1$ , po nepaisymo ryšyje  $R_2$  įrašų, kurių vertė lygi išoriniam raktui, kuris atitinka ryšio  $R_1$  pirminį raktą. Kitais žodžiais, yra neįmanoma turėti du įrašus iš  $R_1$  susieto su tuo pačiu įrašu ryšyje  $R_2$  turint omenyje tą patį išorinį raktą. Nuorodos ar ryšio kardinalumas yra 1:1 jei ir tik jei daugiausia vienas įrašas iš  $R_2$  saugo ryšio  $R_1$  pirminio rakto vertę. Kitaip, kardinalumas yra klasifikuojamas kaip  $M:1$  (tuo atveju yra galima turėti daugiau kaip vieną įrašą iš  $R_2$ , susietą su duotu įrašu iš  $R_1$ ).



**2.6 pav. Galutinis RTO grafas po visų sąryšių ir jų kardinalumų nustatymo.**

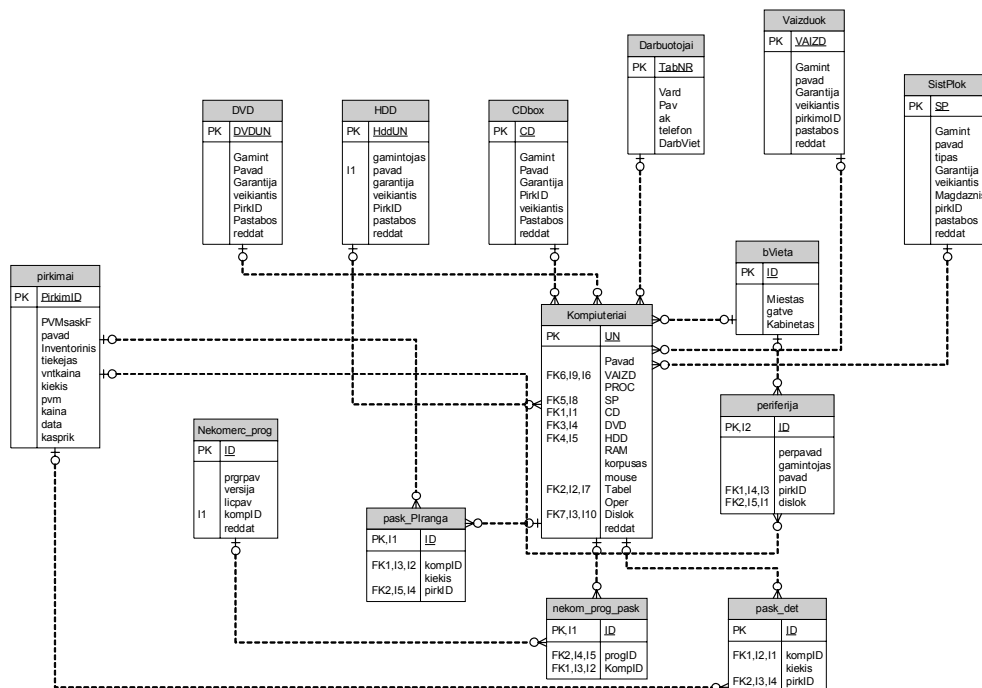
Pavaizduota schema (2.6 pav.) yra galutinis RTO grafas, po to kai visi sąryšiai yra identifikuoti. Likę maži stačiakampiai yra tik ryšiai, atitinkantys esybes.

Matoma, jog ER struktūros modelio išgavimas nėra taip lengvai realizuojamas, ypač, kai nėra žinoma nieko apie buvusios sistemos duomenų struktūrą ir pačius duomenis. Atvirkštinės inžinerijos taikymo aktualumas išauga didesnėse sistemos, kai naujos sistemos kūrimo kaštai pakankamai dideli.

## 2.1 Pakartotinės inžinerijos taikymas palikuoninės DB diagramai suformuoti

Atvirkštinę inžineriją atliksime pasinaudodami Microsoft Visio programinio paketo pagalba. Visio programos meniu lange pasirenkame: „Database- Reverse Engineering” nurodome Access DB tvarkyklę. Toliau nurodome pasirinktų grafinių objektų tipus: lentelės, pirminiai raktai, išoriniai raktai, indeksai. Galutiniame variante programa suformuoja palikuoninės DB diagramą (2.2 pav.)

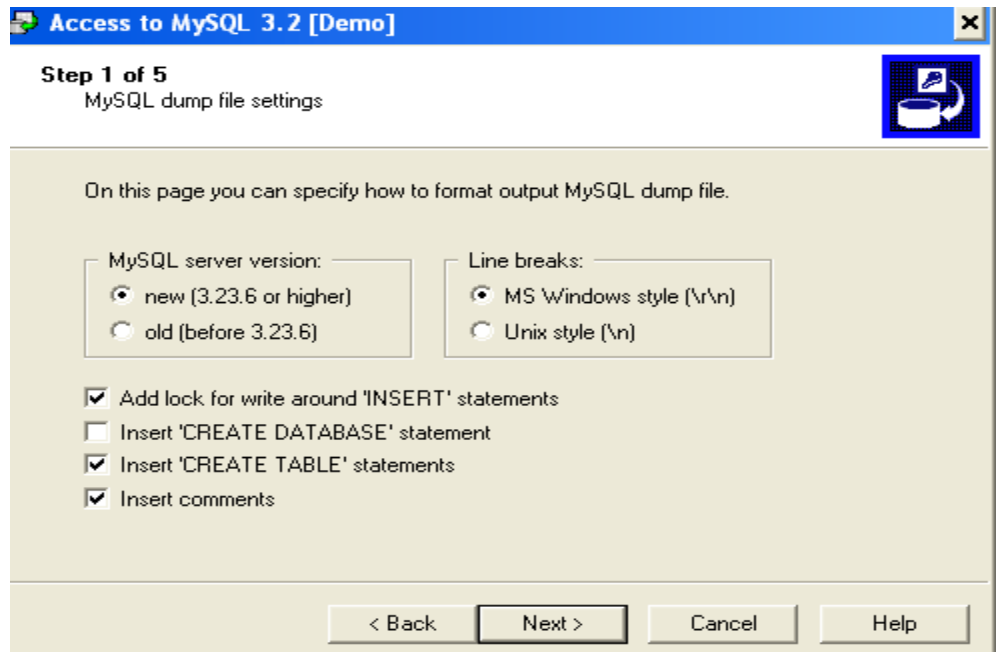




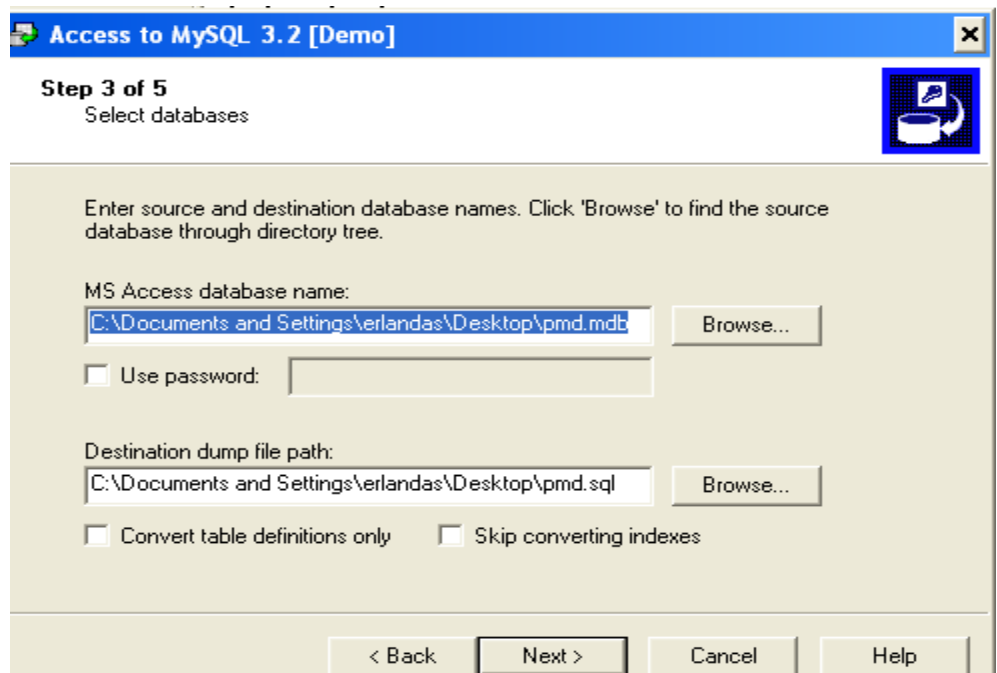
2.2 pav. Palikuoninės Access duomenų bazės diagrama

## 2.2 Palikuoninės Access DB duomenų migracija į MySQL DB

Pakartotinės inžinerijos pagalba išgavę Access DB struktūrą jos nekrengusime, kadangi ji iš dalies atitinka 2 skyriuje apibrėžtus projektuojamos naujos IS išteklių apskaitos reikalavimus. Tam, kad nereiktų naujai formuoti lentelės SQL sakiniiais ir ranka perrašinti lentelių įrašus (palikuoninės DB duomenis) duomenų perkėlimui iš vienos duomenų bazės į kitą bus reikalinga programinė įranga sugebanti atpažinti Access lentelių įrašus, jų atributus, žodinius raktus. Rinkoje yra sukurta platus spektras programų atliekančių duomenų transformacijas iš vienos duomenų bazės valdymo sistemos į kitą. Šiuo atveju panaudosime „Access to MySQL“ programinį įrankį. Žemiau (pav. Nr.20,21) pateikiame programos ekranų pavyzdžius. Ši programa nors ir nesudaro galimybės nustatyti lentelių išorinių raktų, tačiau tokiu atveju remiantis Access DB diagrama (19 pav.) turime pilną informatyvumą apie lentelių elementų semantinius ryšius ir duomenų struktūras. Čia nėra sudėtinga šiuos ryšius nustatyti ir MySQL lentelių programoje phpMyAdmin rankiniu būdu. Priede Nr.7.4 pateikiamas Access transformacijos į MySQL išeities kodas.



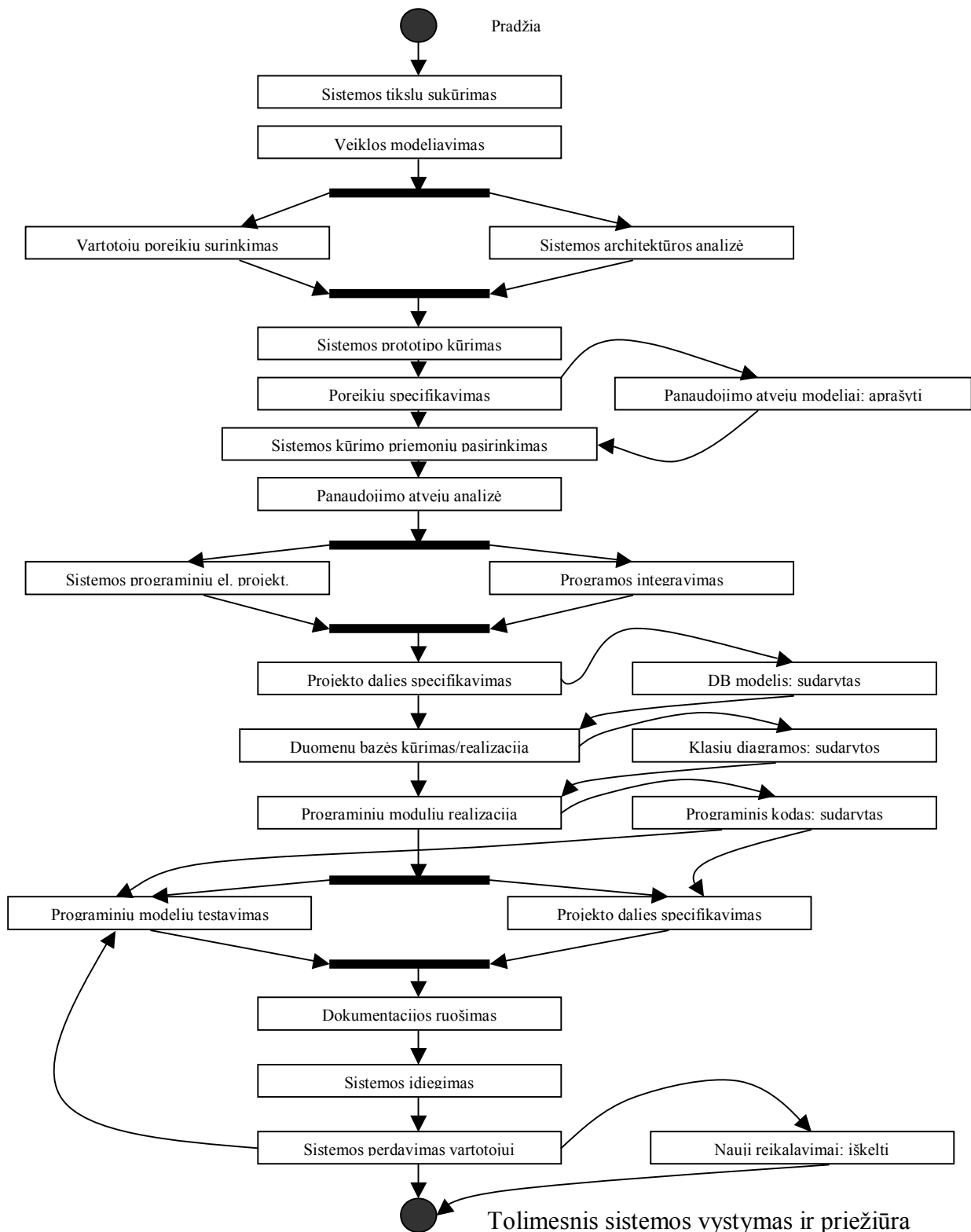
### 2.3 pav. Access to MySQL programos pasirinkimo meniu



### 2.4 pav. Duomenų bazės ir programos rašymo pasirinkimo langas

### 3. INTERNETINĖS DB PALAIKYMŲ PROGRAMINIO ĮRANKIO PROJEKTAVIMAS

#### 3.1 Projektavimo proceso parinkimas



3.1 pav. Projektavimo proceso pasirinkimas

Kiekvieno etapo pabaigoje grįžtama į prieš tai buvusį etapą ir tokiu būdu didinamas sistemos projektavimo efektyvumas ( dėl algoritmo sudėtingumo tai nėra pažymėta). Tai iteracinis metodas, kurį ir pasirenkame sistemos projektavimui.

### 3.2 Išteklių valdymo sistemos funkciniai reikalavimai

Sudarome specifikacijas visiems analizės dalyje sudarytiems kompiuterizuojamiems panaudojimo atvejams.

#### 1. Prisijungimas prie sistemos

**Sistemos dalyvis:** Administratorius, buhalteris

**Sąlyga:** Dalyvis turi turėti vartotojo vardą ir slaptažodį

**Įvykiai:**

<i>Sistemos dalyvio veiksmai</i>	<i>Sistemos reakcija</i>
<i>1. Dalyvis jungiasi prie išteklių valdymo sistemos (<a href="http://itdb.serveriai.lt">http://itdb.serveriai.lt</a>)</i>	<i>1 Sistema pateikia pagrindinį puslapi</i>
<i>2. Sistemos dalyvis pasirenka meniu „Administratoriui“</i>	<i>2. Sistema pateikia prisijungimo langą</i>
<i>3. Sistemos dalyvis įveda slaptažodį ir vardą.</i>	<i>3 Sistema tikrina dalyvio įvestus duomenis. 3.1 Sistema atidaro pagrindinį išteklių sistemos valdymo langą.</i>
<i>4. Sistemos dalyvis naudoja išteklių valdymo sistema</i>	
<i>5. Nesėkmės atvejai</i>	
<i>5.1 Sistemos dalyvis įveda slaptažodį ir vardą.</i>	<i>5.1 Blogas vartotojo vardas (sistema gražina pradinį prisijungimo langą (1))</i>
<i>5.2 Sistemos dalyvis įveda slaptažodį ir vardą.</i>	<i>5.2 Blogas vartotojo slaptažodis (sistema gražina pradinį prisijungimo langą (1))</i>
<i>6. Būtinios sąlygos</i>	<i>6. Dalyvis turi teisingai įvesti vardą ir slaptažodį.</i>

2. Pirktoji įrangos registravimas sistemoje, redagavimas, trynimas.

**Sistemos dalyvis:** Administratorius, buhalteris

**Sąlyga:** Dalyvis yra prisijungęs prie sistemos

**Įvykiai:**


<i>Sistemos dalyvio veiksmai</i>	<i>Sistemos reakcija</i>
<i>1. Dalyvis įvykių juostoje pasirenka punktą „Pirkimai“</i>	<i>1. Sistema atidaro „Pirktoji įrangos peržiūra“ langą</i>
<i>2. Dalyvis „Pirktoji įrangos peržiūra“ lange pasirenka punktą „Įvesti duomenis“</i>	<i>2. Sistema atidaro „Naujo pirkimo registravimas“ langą</i>
<i>3. Įrašęs duomenis dalyvis spaudžia mygtuką „Įvesti“</i>	<i>3. Sistema registruoja languose įvestus duomenis į duomenų bazę</i>
<i>4. Dalyvis „Pirktoji įrangos peržiūra“ lange pasirenka punktą „Redaguoti duomenis“</i>	<i>4. Sistema atidaro „Pirkimo redagavimo“ langą</i>
<i>5. Pakeitęs duomenis dalyvis spaudžia mygtuką „Registruoti pakeitimus“</i>	<i>5. Sistema registruoja languose įvestus duomenis į duomenų bazę</i>
<i>6. Dalyvis „Pirktoji įrangos peržiūra“ lange pasirenka punktą „Trinti duomenis“</i>	<i>6. Sistema atidaro „Pirkimo trynimo langą“ langą</i>
<i>7. Dalyvis nurodo trinamos įrašo ID ir spusteli mygtuką Trinti“</i>	<i>7. Sistema ištrina įrašą iš duomenų bazės</i>
<i>8. Sistemos dalyviui pavyksta atlikti 1-7 punkte nurodytus veiksmus</i>	
<i>9. Nesėkmės atvejai</i>	
	<i>9.1 Nepavyksta įvesti duomenis ( Pirkimo ID yra pasikartojantis arba &lt;&gt;skaičius) Sistema grąžina į „Pirktoji įrangos peržiūra“ langą“</i>
	<i>9.2 Nepavyksta ištrinti įrašą (neteisingai nurodomas įrašo ID) Sistema grąžina į „Pirktoji įrangos peržiūra“ langą“</i>
<i>5. Būtinios sąlygos</i>	<i>Dalyvis turi teisingai įvesti naujai formuojamo ar trinamo įrašo ID numerį.</i>

3. Konkrečios darbo vietos konfigūracija, atskirų KĮ mazgų peržiūra, naujų mazgų įvedimas ir registravimas ir trynimas iš DB.

**Sistemos dalyvis:** Administratorius, buhalteris

**Sąlyga:** Dalyvis yra prisijungęs prie sistemos

**Įvykiai:**




<p>1. Dalyvis įvykių juostoje pasirenka punktą „Peržiūra“</p>	<p>1. Sistema atidaro pagalbinį meniu langą „Pagrindiniai kompiuterinės darbo vietos mazgai“</p>
<p>2. Dalyvis meniu lange „Pagrindiniai kompiuterinės darbo vietos mazgai“ pasirenka „Darbo vietos konfigūracija“ punktą</p>	<p>2. Sistema atidaro langą „Vartotojo darbo vietos kompiuterinė įranga“</p>
<p>3. Dalyvis lange „Pagrindiniai kompiuterinės darbo vietos mazgai“ pasirenka „Vartotojo darbo vietos kompiuterinė įranga“ punktą „Formuoti naują darbo vietą“, (spusteli ant piktogramos )</p>	<p>3. Sistema atidaro langą „Kompiuterinės darbo vietos įrangos parinkimas“</p>
<p>4. Dalyvis „Kompiuterinės darbo vietos įrangos parinkimas“ lange parenka iš esančio meniu sąrašo esančius kompiuterinės įrangos mazgus, bei užpildo duomenų langus reikalingais duomenimis ir paspaudžia įvedimo jungtuką.</p>	<p>4. Sistema suformuoja naują vartotojo darbo vietą ir užregistruoja duomenų bazėje.</p>
<p>5. Dalyvis lange „Pagrindiniai kompiuterinės darbo vietos mazgai“ laukelyje  šalia redagavimo piktogramos  įveda DV ID numerį ir spusteli piktogramą „Redaguoti darbo vietos duomenis“</p>	<p>5. Sistema atidaro langą „Vartotojo darbo vietos kompiuterinė įranga“</p>
<p>6. Dalyvis „Vartotojo DV kompiuterinės įrangos“ lange atlieka duomenų redagavimo veiksmus ir spusteli įvedimo jungtuką</p>	<p>6. Sistema išsaugo pakeistus darbo vietos duomenis duomenų bazėje.</p>

7. Sistemos dalyviui pavyksta atlikti 1-6 punkte nurodytus veiksmus	
8. Nesėkmės atvejai	
	8.1 Nepavyksta įvesti duomenis (Kompiuterio unikalus numeris yra pasikartojantis arba <>skaičius) Sistema formuoja klaidos pranešimą grąžina į „Kompiuterinės darbo vietos parinkimo langą“
	8.2 Nepavyksta ištrinti norimą įrašą (neteisingai nurodomas įrašo ID) Sistema grąžina į „Pirktos įrangos peržiūros langą“
9. Būtinios sąlygos	Dalyvis turi teisingai įvesti naujai formuojamo įrašo ID numerį.

4. Vartotojų registravimas į duomenų bazę, šalinimas iš duomenų bazės teisių suteikimas  
**Sistemos dalyvis:** Administratorius

**Sąlyga:** Dalyvis yra prisijungęs prie sistemos

**Įvykiai:**

1. Dalyvis įvykių juostoje pasirenka punktą „Kl naudotojai“	1. Sistema pagrindiniame lange išveda lentelę „Kl naudotojai“
2. Dalyvis meniu lange „Kl naudotojai“ pasirenka „Įvesti“ piktogramą 	2. Sistema atidaro langą „Darbuotojo duomenų įvedimas“
3. Dalyvis lange „Darbuotojo duomenų įvedimas“ pasirinktuose teksto languose surašo darbuotojo duomenis, priskiria jam unikalų numerį ID išspusteli įvedimo mygtuką.	3. Sistema registruoja duomenis duomenų bazėje.
4. Dalyvis meniu lange „Kl naudotojai“ pasirenka „Trinti“ piktogramą 	4. Sistema atidaro langą „Pasalinimas iš duomenų bazės“
5. Dalyvis lange „Kl naudotojai“ laukelyje  alia redagavimo piktogramos įveda DV ID numerį ir spusteli piktogramą „Redaguoti“	5. Sistema atidaro langą „Kl naudotojai redagavimo langą“

<i>darbo vietos duomenis“</i>	
6. Dalyvis „Kl naudotojų redagavimo lange“ lange atlieka duomenų redagavimo veiksmus ir spusteli įvedimo jungtuką.	6. Sistema išsaugo pakeistus darbo vietos duomenis duomenų bazėje.
7. Sistemos dalyviui pavyksta atlikti 1-6 punkte nurodytus veiksmus	
8. Nesėkmės atvejai	
	8.1 Nepavyksta įvesti duomenis (Kl naudotojo UN pasikartojantis arba <>skaičius) Sistema formuoja klaidos pranešimą gražina į „Kl naudotojų langą“
	8.2 Nepavyksta ištrinti norimą įrašą (neteisingai nurodomas įrašo ID) Sistema gražina į „Kl naudotojų langą“
9. Būtinios sąlygos	Dalyvis turi teisingai įvesti naujai formuojamo įrašo ID numerį.

5. Ataskaitų formavimas iš duomenų bazėje esamų įrašų.

**Sistemos dalyvis:** Administratorius, buhalteris, registruotas vartotojas

**Sąlyga:** Dalyvis yra prisijungęs prie sistemos

1. Dalyvis įvykių juostoje pasirenka punktą „Ataskaitos“	1. Sistema pagalbiniam lange išveda lentelę „Ataskaitos“
2. Dalyvis meniu lange „Ataskaitos“ pasirenka „Pateikti KI pagal vartotojus“	2. Sistema atidaro langą „Darbuotojo turima kompiuterinė įranga“
3. Dalyvis lange „Darbuotojo turima kompiuterinė įranga“ iš meniu slenkančios juostos parenka reikiamą vartotoją ir spusteli mygtuką „Ieškoti Kl pagal darbuotojo vardą“	3. Sistema išveda užklauso duomenis. Sistema pateikia galimybę suformuoti ataskaitą atspausdinti spausdintuve.



4. Dalyvis meniu lange „Ataskaitos“ pasirenka „Pateikti KĮ vartotojus pagal darbo vietos ID“	4. Sistema atidaro langą „KĮ vartotojas“
5. Dalyvis lange „KĮ vartotojas“ pasirenka DV ID slenkančio meniu pagalba	5. Sistema išveda užklauso duomenis.
6. Dalyvis meniu lange „ Ataskaitos“ pasirenka „Pateikti informacija apie laisvą įrangą“	6. Sistema pagrindiniame lange išveda visų įrenginių laisvos(nepriskirtos jokiam vartotojui) įrangos sąrašus.
7. Dalyvis meniu lange „ Ataskaitos“ pasirenka „Pateikti informacija apie programinę įrangą“	7. Sistema pagrindiniame lange išveda langą „ Programinė įranga“
8. Dalyvis lange „Programinė įranga“ slenkančio meniu pagalba pasirenka reikiamos darbo vietos UN.	8. Sistema išveda konkrečioje kompiuterinėje darbo vietoje įdiegtos programinės įrangos pavadinimus.
9. Dalyvis meniu lange „ Ataskaitos“ pasirenka „Įrangos likutinės vertės skaičiavimas	9. Sistema pagrindiniame lange išveda langą „Ilgalaikio turto Likutinė vertė“ kuriame nurodoma kiekvieno įrenginio pirkimo data, pavadinimas, likutinės vertė visoms prekėms, kurių suma >P (pagal nutylėjimą 500)
10. Sistemos dalyviui pavyksta atlikti 1-9 punkte nurodytus veiksmus	
8. Nesėkmės atvejai	
	Kadangi vykdomos užklauso pagal pasirinkimą iš programos užduotų įrašų, nesėkmės atvejų dėl vartotojo kaltės neturi būti

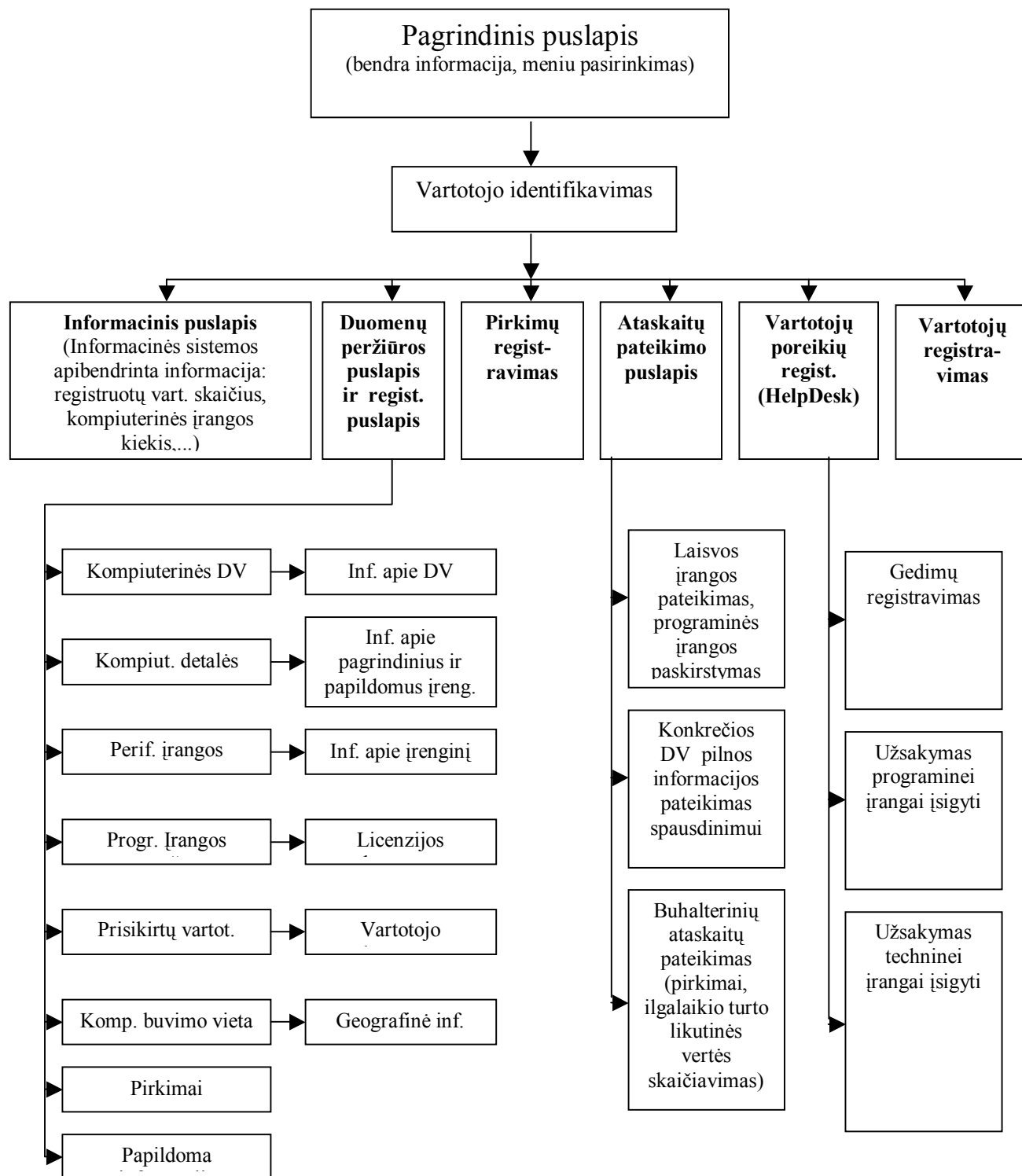
6. Užsakymų programinei, techninei įrangai, gedimų registravimas

**Sistemos dalyvis:** Registruotas vartotojas (vidinis įmonės vartotojas ar klientas)

**Sąlyga:** Dalyvis yra prisijungęs prie sistemos

<i>1. Dalyvis įvykių slenkančioje juostoje pasirenka punktą „Gedimų registravimas“</i>	<i>1. Sistema atidaro gedimų registravimo langą.</i>
<i>2. Dalyvis gedimų registravimo formoje pasirenka iš pasirinkimo meniu „darbo veitos ID“, „gedimą registravo“, „gedimo pobūdis“, taip pat apibūdina problemą.</i>	<i>2. Sistema registruoja formoje įvestus duomenis ir išsaugo duomenų bazėje. Duomenys prieinami tik sistemos administratoriui, taigi pataisyti užregistruotos problemos negalima.</i>
<i>3 Analogiškai atliekami programinės, techninės papildomos ar naujos įrangos užsakymai</i>	
<i>4. Sistemos dalyviui pavyksta atlikti 1-3 punkte nurodytus veiksmus</i>	
<i>8. Nesėkmės atvejai</i>	
	<i>Kadangi vykdomos užklauskos pagal pasirinkimą iš programos užduotų įrašų, nesėkmės atveju dėl vartotojo kaltės neturi būti</i>

### 3.3 Vartotojų sąsajos modelis



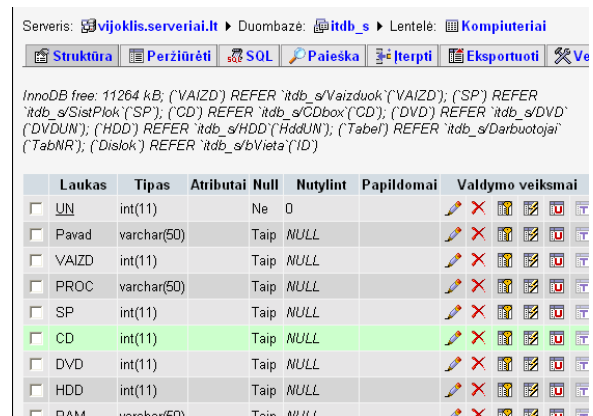
3.2 pav. Vartotojo sąsajos modelis

### 3.4 Projektuojamos duomenų bazės lentelių struktūros parinkimas

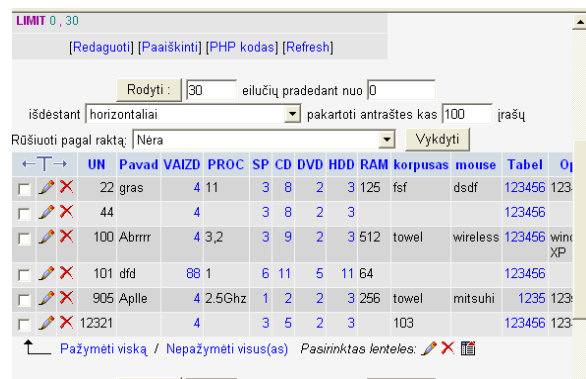
Perkėlus duomenis iš Access į MySQL duomenų bazę panaudodami „dump“ bylą (Priedas 1) PhpMyAdmin 2.6 programa galime peržiūrėti duomenų struktūras:



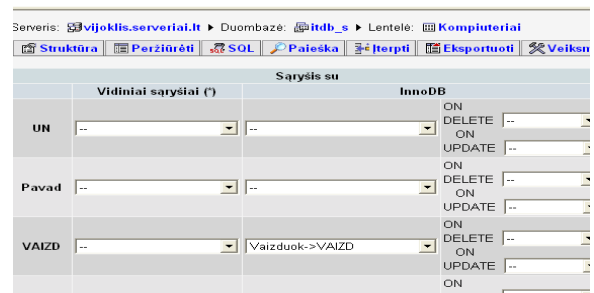
3.2 pav. phpMyAdmin programos pradinis puslapis



3.3 pav. Lentelės „Kompiuteriai“ įrašų formatų nustatymo forma

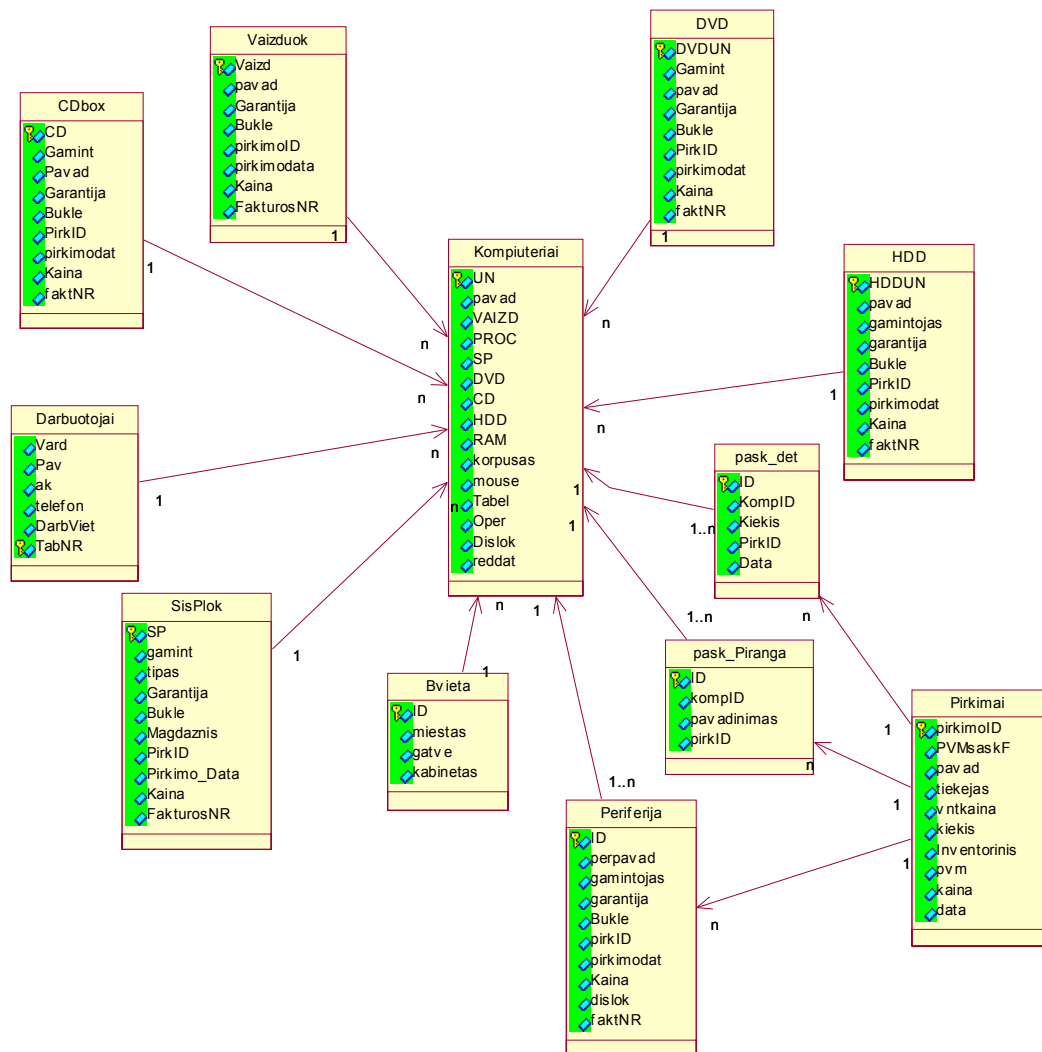


3.4 pav. Lentelės „Kompiuteriai“ įrašų redagavimo forma



3.5 pav. Lentelės „Kompiuteriai“ vidinių ir išorinių sąryšių nustatymo forma

MySQL suteikia galimybę aprašyti lentelių tarpusavio sąryšius (3.5 pav.). Tai atliekama transakcinių „InnoDB“ lentelių pagalba. PhpMyAdmin lentelių išorinių sąryšių formoje galima parinkti norimos lentelės konkretaus lauko sąryšį su kitos lentelės pasirinktu lauku. Tokiu būdu galima nustatyti lentelių išorinius raktus. Rational Rose pagalba aprašome MySQL duomenų bazėje įvestus laukus, papildome juos naujais įrašais, nustatome raktinius žodžius. Gauname išoriniais raktais tarpusavyje surištas lenteles (3.6 pav.)



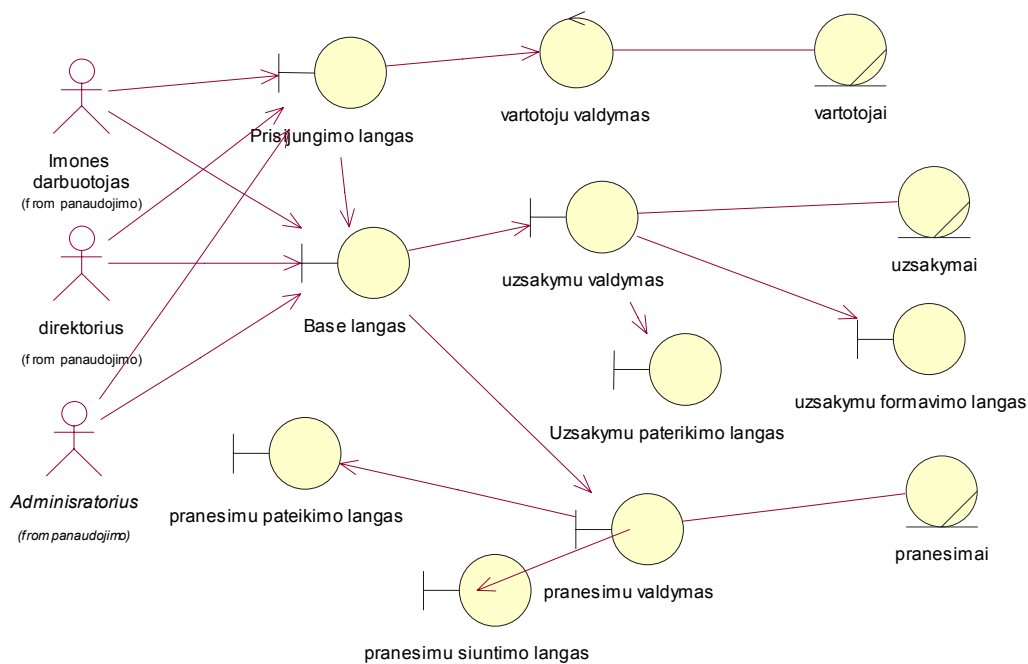
3.6 pav. Duomenų bazės struktūros sąryšių diagrama

Aprašę duomenų lenteles ir nustatę jų tarpusavio sąryšius galime pereiti prie duomenų lentelių valdymo formavimo. Tam analizės dalyje aprašėme pasirinkę PHP kalbos privalumus. Naudosime objektinį PHP programavimo metodą. Atliksime analizės dalyje panaudojimo atvejų realizacijas.

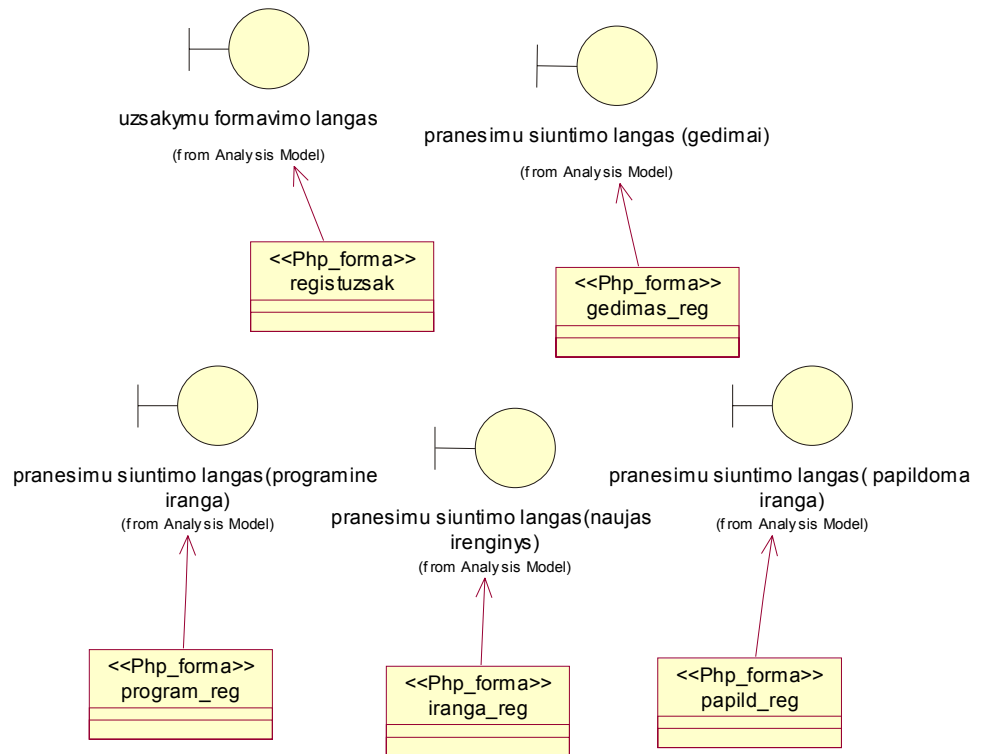
### 3.5 Panaudojimo atvejų realizacijos

Kaip ir analizės dalyje suformuotus panaudojimo atvejus taip ir jų realizacijas pavaizduosime analizės klasių diagramomis.

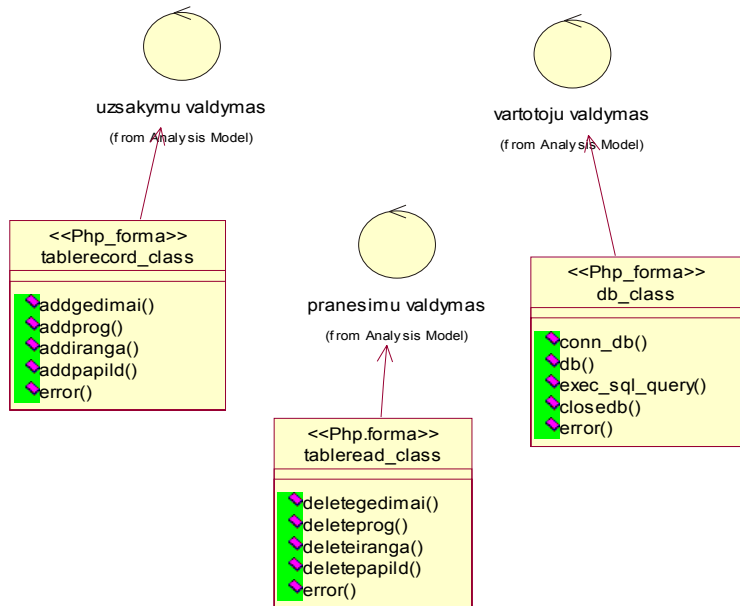
1. Užsakymų registravimo panaudojimo atvejų diagramos realizacija:



3.7 pav. Užsakymų registravimo analizės klasių diagrama.

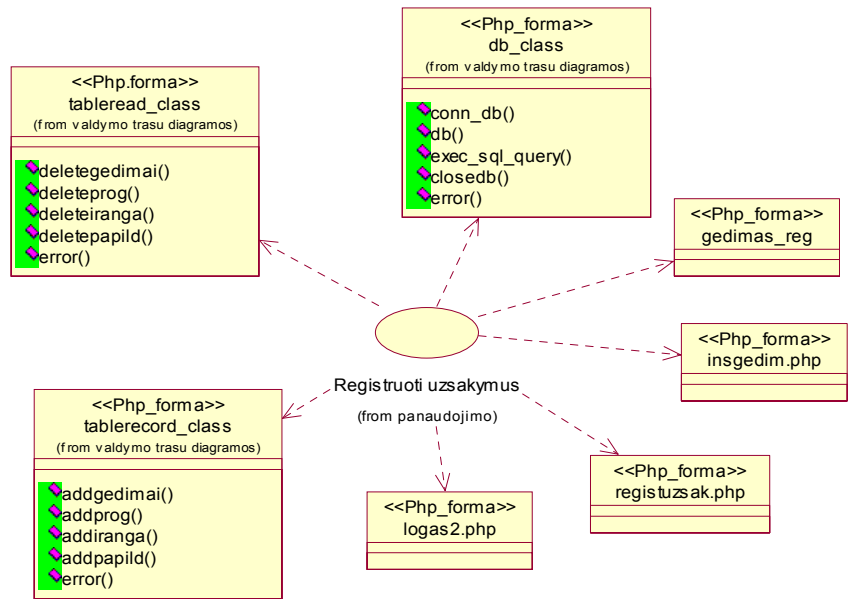


3.8 pav. Užsakymų registravimo sąsajos trasų diagrama.

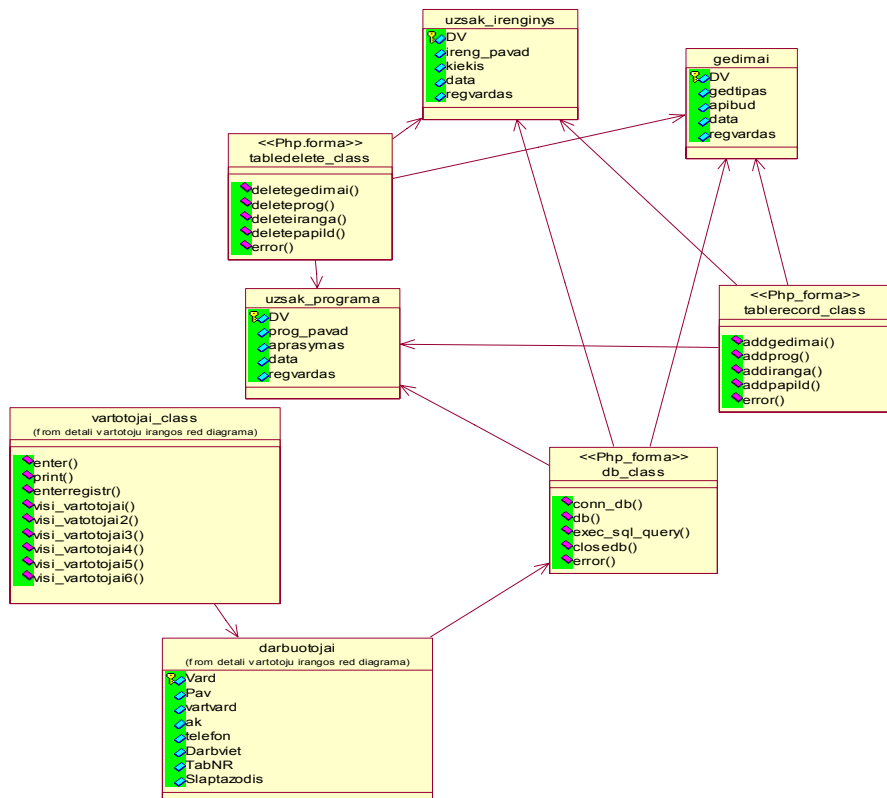


3.9 pav. Užsakymų registravimo programavimo trasų diagrama.

Grafiškai pateikiame tarpusavyje susietas klases ir php formas vartotojo įrangos redagavimo funkcijoms atlikti.



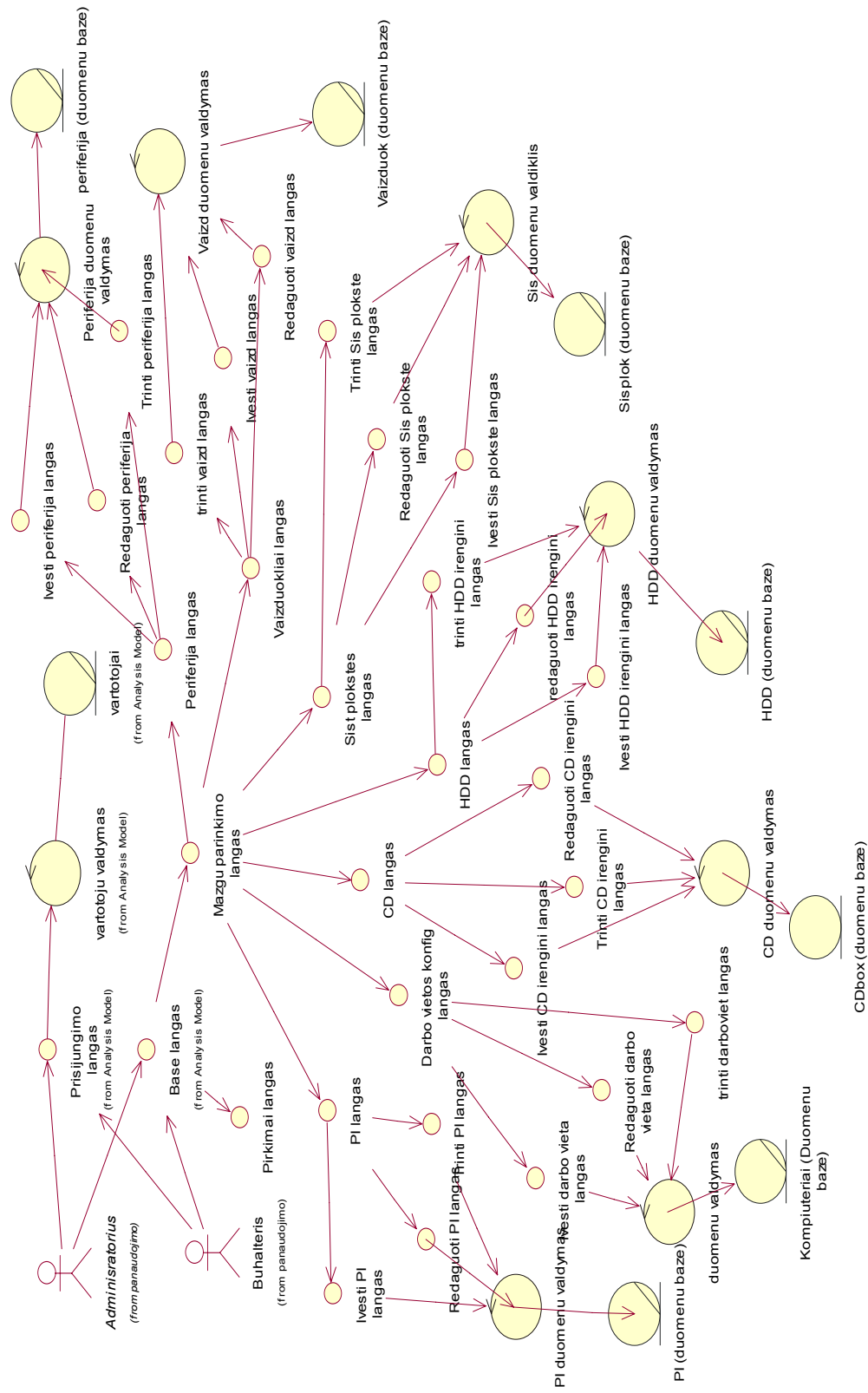
3.10 pav. Formų sąsajų diagrama užsakymų registravimo funkcijomis atlikti



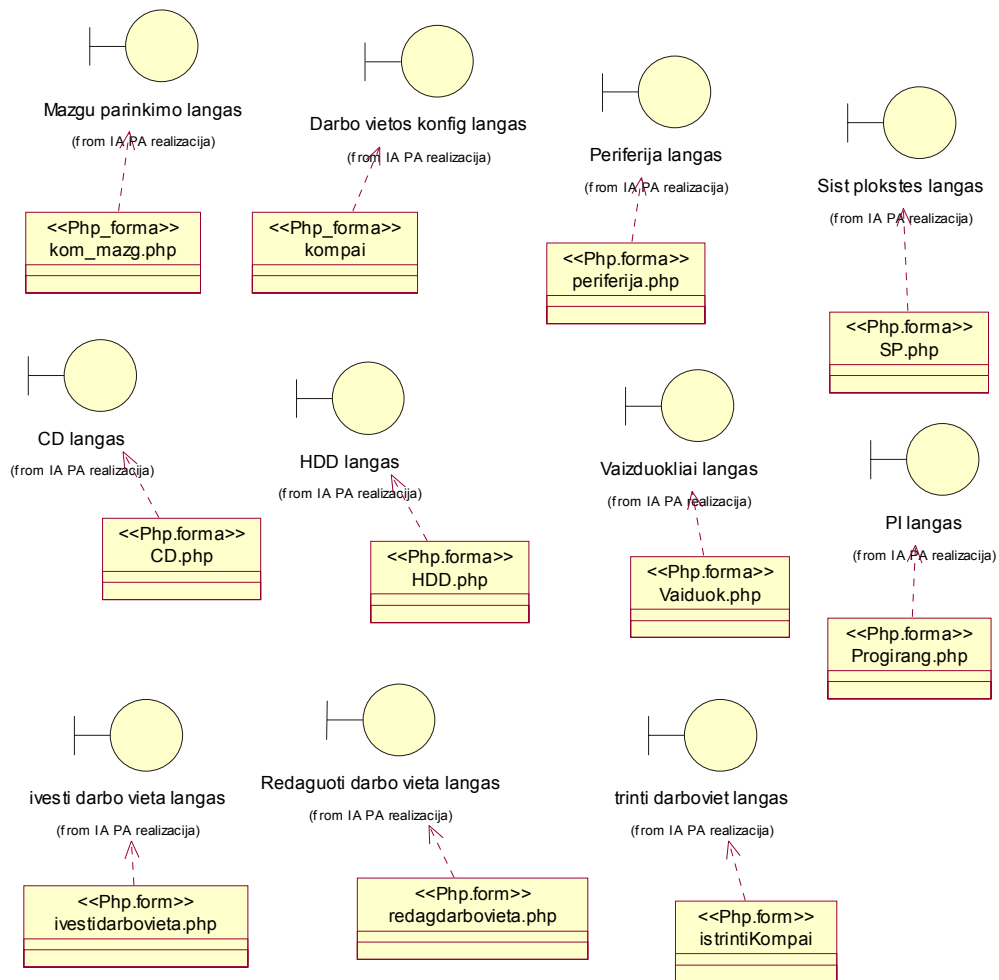
3.11 pav. Detali užsakymų registravimo diagrama



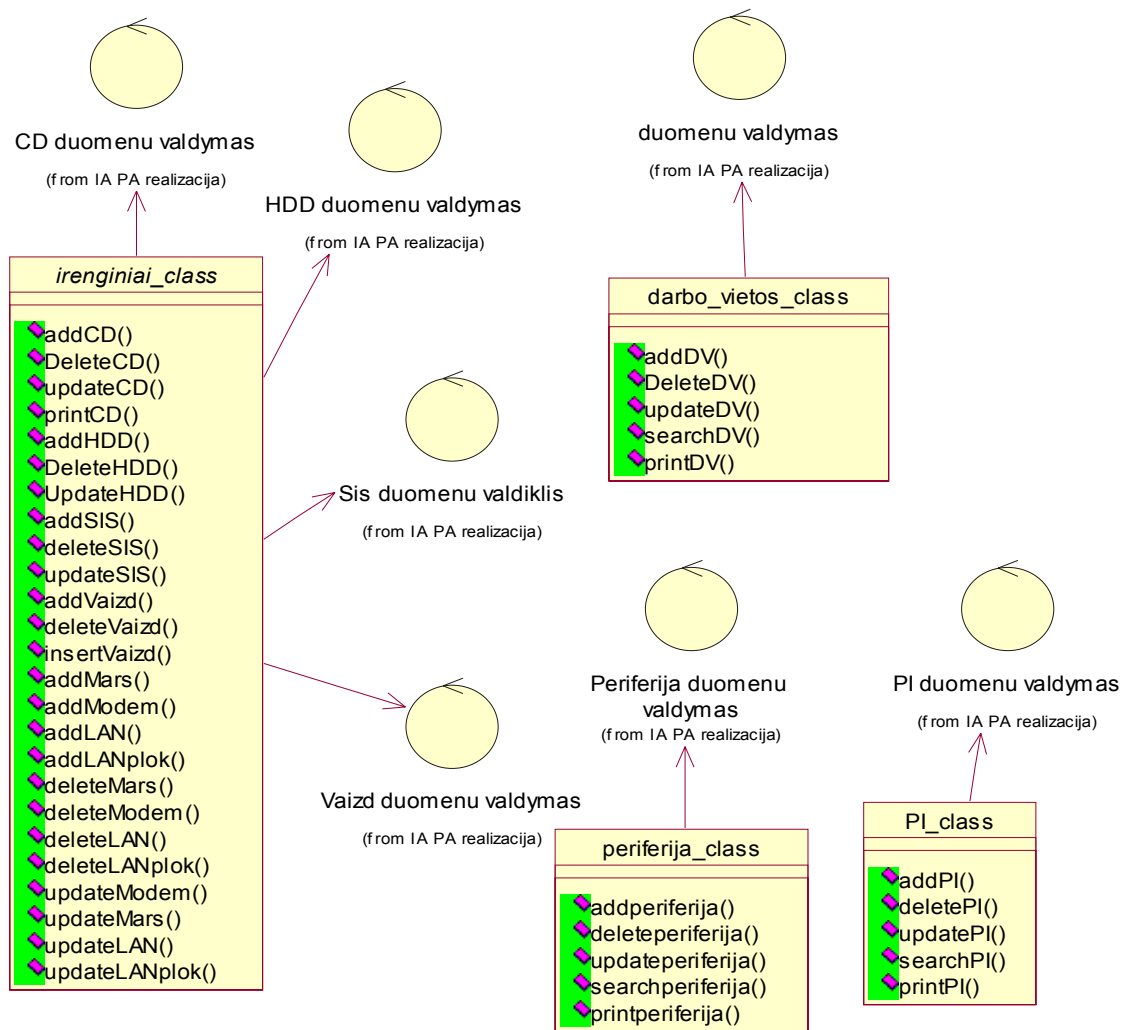
## 2. Išteklių apskaitos panaudojimo atvejų diagramos realizacija.



3.12 pav. Išteklių apskaitos panaudojimo atvejų diagramos realizacija.

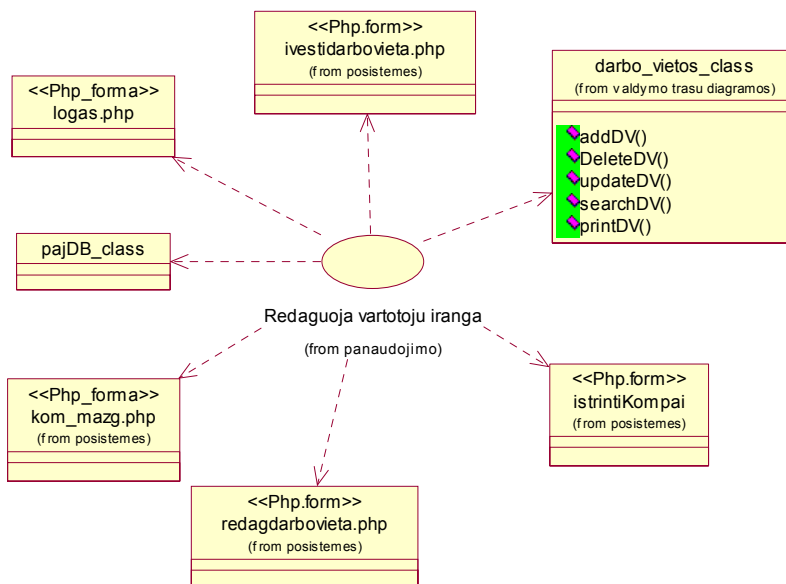


3.13 pav. Išteklių apskaitos vartotojo sąsajos trasų diagrama.

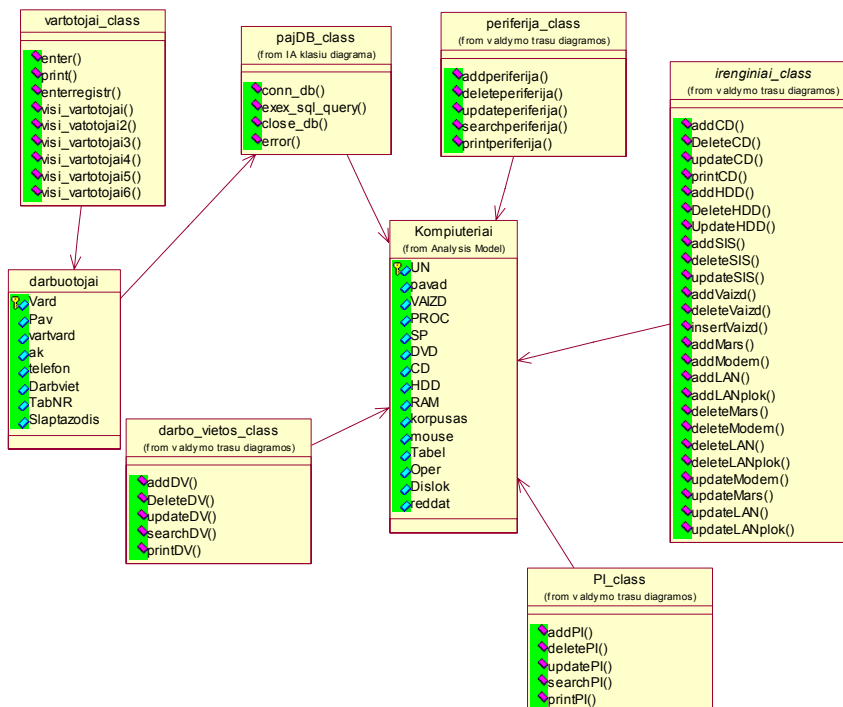


3.14 pav. Išteklų apskaitos programavimo trasų diagrama.

Grafiškai pateikiame klases ir php formas tarpusavyje susietas vartotojo įrangos redagavimo funkcijoms atlikti.



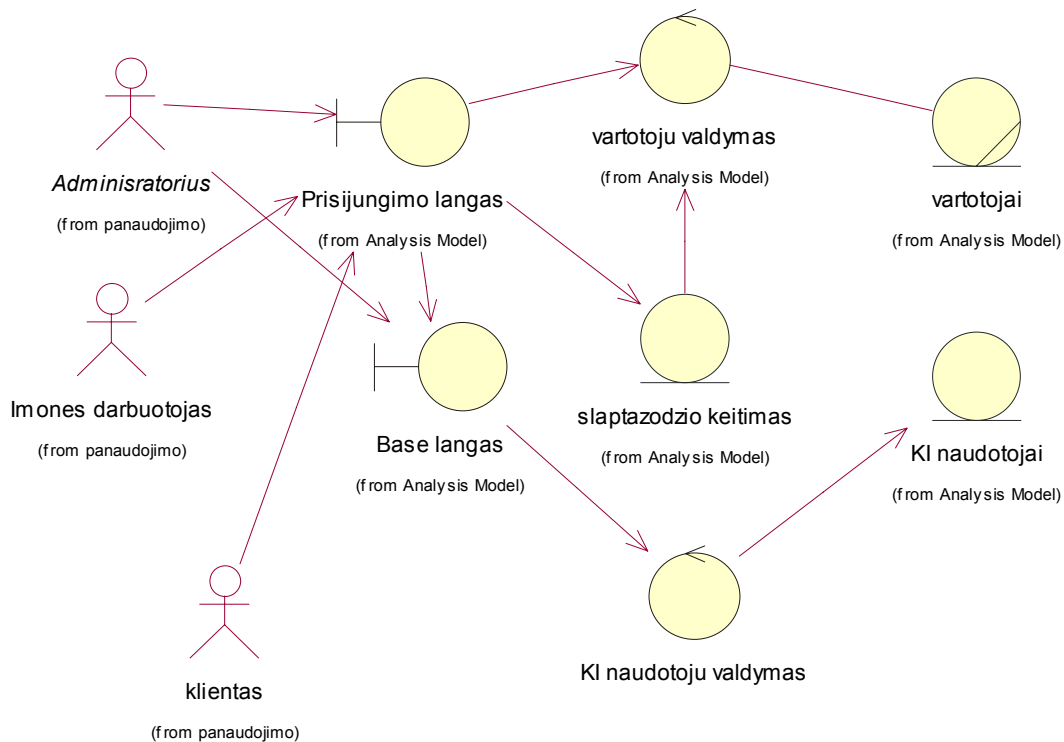
3.15 pav. Formų sąsajų diagrama redagavimo funkcijomis atlikti



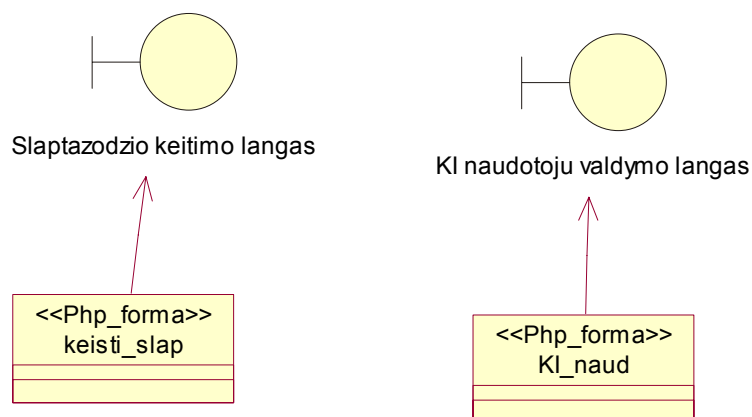
3.16 pav. Detali vartotojo įrangos redagavimo diagrama

### 3. Vartotojų registravimo proceso panaudojimo atvejų diagramos

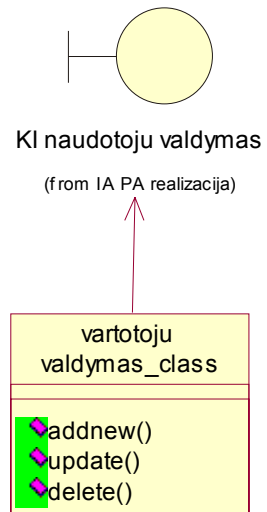
realizacija:



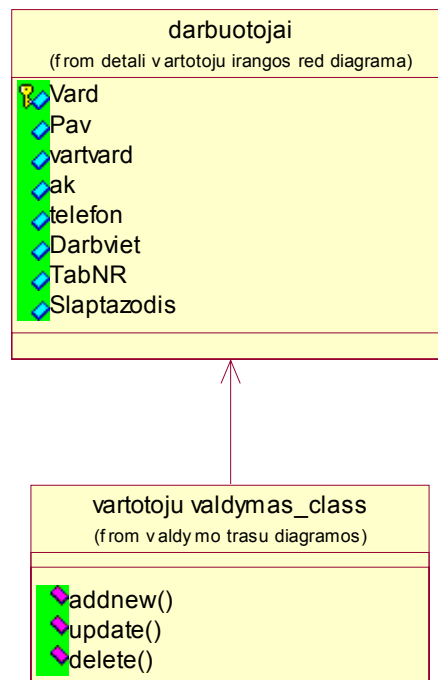
**3.17 pav. Vartotojų registravimo panaudojimo atvejų diagramos realizacija.**



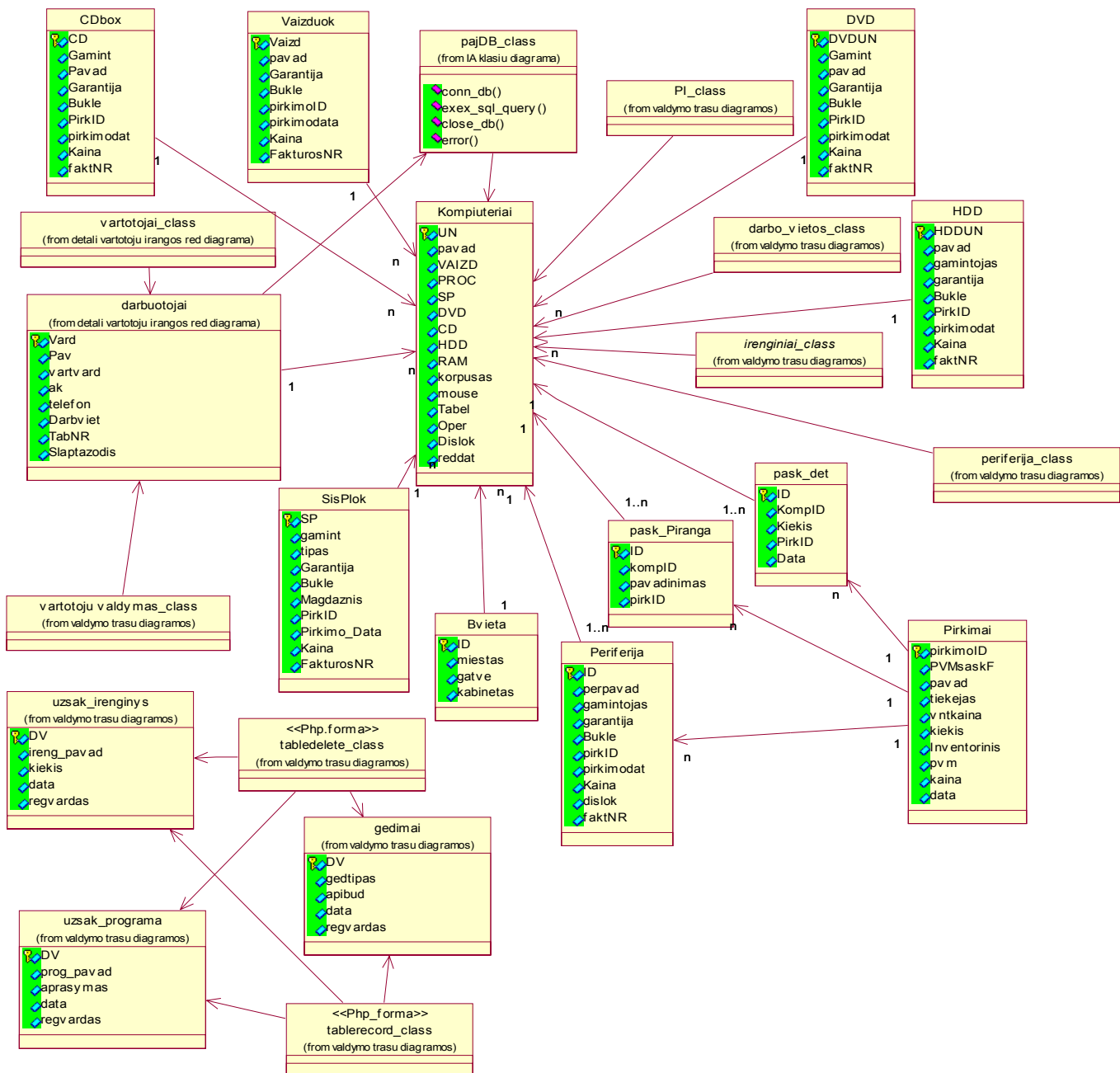
**3.18 pav. Vartotojų registravimo trasų diagrama.**



**3.19 pav. Vartotojų valdymo programavimo trasų diagrama.**



**3.20 pav. Detali vartotojų valdymo diagrama**



3.21 pav. Pilna „informacinės sistemos“ klasių diagrama

### 3.6 Programinių klasių aprašymai

PHP programavimo kalbos pagalba duomenų bazės lentelių valdymui sukūreime ir realizavome (3.1 lentelė) programines klases.

3.1 lentelė

#### Programinės klasės

Klasės pavadinimas	Panaudojimas
Darbo_vieta_class -	Darbo vietos registravimui ir redagavimui.
Irenginiai_class;	Įrenginių registravimui ir redagavimui.
Vartotojai_class;	Vartotojų identifikavimui ir išrinkimui, įrenginių išrinkimui.
Vartotojų_valdymas_class	Vartotojų registravimui ir šalinimui.
Periferija_class;	Periferinės įrangos registravimui ir redagavimui.
PI_class;	Programinės įrangos registravimui ir redagavimui.
Db_class;	Prisijungimui prie duomenų bazės.
Tabledelete_class;	Gedimų, įrangos užsakymų šalinimas.
Tablread_class.	Gedimų, įrangos užsakymų registravimas.

Žemiau aprašysime kiekvieną klasę, jos vykdomas funkcijas ir naudojamus parametrus.

- o Darbo\_vieta\_class;

3.2 lentelė

#### Darbo vieta

Funkcija	Parametrai	Aprašymas
addDV()	Visi lentelės „Kompiuteriai“ laukų pavadinimai.	Įveda naujus įrašus
DeleteDV()	ID	Trina lentelės įrašą
updateDV()	Visi lentelės „Kompiuteriai“ laukų pavadinimai	Atnaujina lentelės laukus
searchDV()	ID	Atrenka pasirinktą įrašą
printDV()	ID	Atspausdina pasirinktą įrašą



- o Irenginiai\_class;

3.3 lentelė

**Irenginiai**

Funkcijos	Parametrai	Aprašymas
addCD(); addDVD(); addHDD(); addSIS(); addVaizd(); addMars(); addModem(); addLan(); addLanplok(); addPerif();addPI()	Irenginių lentelių visi laukų pavadinimai	Iveda naujus įrašus į pasirinktas lenteles
DeleteCD(); DeleteDVD(); DeleteHDD(); DeleteSIS(); DeleteVaizd(); DeleteMars(); DeleteModem(); DeleteLan(); DeleteLanplok(); DeletePerif(); DeletePI()	Irenginių lentelių visi laukų raktinių laukų (ID) pavadinimai	Trina pasirinktų lentelės įrašus
updateCD(); updateDVD(); updateHDD(); updateSIS(); updateVaizd(); updateMars(); updateModem(); updateLan(); updateLanplok(); updatePerif(); updatePI()	Irenginių lentelių visi laukų pavadinimai	Atnaujina lentelės laukus
printCD(); printDVD(); printHDD(); printSIS(); printVaizd(); printMars(); printModem(); printLan; printLanplok; printPerif(); printPI()	Irenginių lentelių visi laukų pavadinimai.	Spausdina pasirinktas lenteles

- Vartotojai\_class;

**3.4 lentelė**

**Vartotojai**

Funkcijos	Parametrai	Aprašymas
Enterregistr()	Login, pass	Identifikuoja vartotoją pagal vartotojo prisijungimo vardą ir slaptažodį. (vartotojo prisijungimas)
Enter()	Login, pass	Identifikuoja vartotoją pagal vartotojo prisijungimo vardą ir slaptažodį (administratoriaus prisijungimas)
visi_CD(),visi_HDD(), visi_DVD(),visi_SIS() visi_Vaizd(),visi_Mars(), visi_Modem(), visi_LAN(), visi_Lanplok(),visi_perif(),visi_PI()	visi_vart	Išrenka iš įrenginių lentelių Visus įrašus

- Vartotojų\_valdymas\_class

**3.5 lentelė**

**Vartotojų valdymas**

Funkcijos	Parametrai	Aprašymas
Addnew()	Visi lentelės „Darbuotojai“ laukai	Įvesti darbuotojų duomenis
Update()	Visi lentelės „Darbuotojai“ laukai	Atnaujinti darbuotojų duomenis
Delete()	Lentelės „Darbuotojai“ ID numerio pavadinimas	Ištrinti darbuotoją

- Periferija\_class

**3.6 lentelė**

**Periferija**

Funkcijos	Parametrai	Aprašymas
addperiferija	Lentelės „Periferija“ visi laukų pavadinimai	Įveda naujus įrašus į pasirinktą lentelę
deleteperiferija	Lentelės raktinio žodžio (ID) pavadinimas	Trina pasirinktus lentelės įrašus
updateperiferija	Lentelės „Periferija“ visi laukų pavadinimai	Atnaujina lentelės laukus
Searchperiferija	Lentelės raktinio žodžio (ID) pavadinimas.	Ieško lentelės įrašų
printperiferija	periferija	Spausdina lentelę

○ PI\_class

**3.7 lentelė**

**Programinė įranga**

Funkcijos	Parametrai	Aprašymas
addPI	Lentelės „Nekomerc_Prog“ laukų pavadinimai	Įveda naujus įrašus į pasirinktą lentelę
deletePI	Lentelės „Nekomerc_Prog“ raktinio žodžio (ID) pavadinimas	Trina pasirinktus lentelės įrašus
updatePI	Lentelės „Nekomerc_Prog“ visi laukų pavadinimai	Atnaujina lentelės laukus
Searchperiferija	Lentelės raktinio žodžio (ID) pavadinimas.	Ieško lentelės įrašų
printPI	Nekomerc_Prog	Spausdina lentelę

○ Db\_class

**3.8 lentelė**

**DB**

Funkcijos	Parametrai	Aprašymas
Conn_db()	Server, user, password	Atlieka prisijunkimą prie duomenų bazės

Db()	Server, user, password, db_name	konstruktorius
Exec_sql_query()		Užklauso vykdymas
Close_db()		Atsijungti nuo duomenų bazės
Error()		Klaidos išvedimas

○ Tabledelete\_class

**3.9 lentelė**

**Tabledelete**

Funkcijos	Parametrai	Aprašymas
Deletegedimai()	Lentelės „gedimai“ ID	Pasirinkto įrašo pašalinimas
Deleteprog()	Lentelės „uzsak_programa“ ID	Pasirinkto įrašo pašalinimas
Deleteirang()	Lentelės „uzsak_irenginys“ ID	Pasirinkto įrašo pašalinimas
Deletepapild()	Lentelės „uzsak_papild“ ID	Pasirinkto įrašo pašalinimas
Error()		Klaidos išvedimas

○ Tablerecord\_class

Funkcijos	Parametrai	Aprašymas
Addgedimai()	Lentelės „gedimai“ laukų pavadinimai	Naujo įrašo įvedimas
Addprog()	Lentelės „uzsak_programa“ laukų pavadinimai	Naujo įrašo įvedimas
Addirang()	Lentelės „uzsak_irenginys“ laukų pavadinimai	Naujo įrašo įvedimas
Addpapild()	Lentelės „uzsak_papild“ laukų pavadinimai	Naujo įrašo įvedimas
Error()		Klaidos išvedimas

### 3.7 Duomenų lentelių aprašymas su duomenų valdymo programa „PHPmyAdmin“

Duomenų bazės „itdb\_s“ aprašymas

Lentelė	Valdymo veiksmai	Viso įrašų:	Tipas	Dydis	Per
<input type="checkbox"/> CDbox		11	InnoDB	32.0 KB	
<input type="checkbox"/> DVD		6	InnoDB	16.0 KB	
<input type="checkbox"/> Darbuotojai		9	InnoDB	16.0 KB	
<input type="checkbox"/> HDD		6	InnoDB	32.0 KB	
<input type="checkbox"/> Kompiuteriai		6	InnoDB	208.0 KB	
<input type="checkbox"/> LanPlok		0	InnoDB	16.0 KB	
<input type="checkbox"/> LanSak		0	InnoDB	16.0 KB	
<input type="checkbox"/> Nekomerc_prog		3	InnoDB	16.0 KB	
<input type="checkbox"/> SistPlok		5	InnoDB	16.0 KB	
<input type="checkbox"/> Vaizduok		7	InnoDB	16.0 KB	
<input type="checkbox"/> bVieta		3	InnoDB	16.0 KB	
<input type="checkbox"/> gedimai		8	InnoDB	16.0 KB	
<input type="checkbox"/> marsrut		0	InnoDB	16.0 KB	
<input type="checkbox"/> modemai		0	InnoDB	16.0 KB	
<input type="checkbox"/> nekomp_prog_pask		0	InnoDB	16.0 KB	
<input type="checkbox"/> pask_det		0	InnoDB	16.0 KB	
<input type="checkbox"/> pask_PIranga		0	InnoDB	16.0 KB	
<input type="checkbox"/> periferija		0	InnoDB	16.0 KB	
<input type="checkbox"/> pirkimai		0	InnoDB	16.0 KB	
<input type="checkbox"/> regvartot		0	InnoDB	16.0 KB	
<input type="checkbox"/> skelbimai		0	InnoDB	16.0 KB	
<input type="checkbox"/> uzsak_irenginys		0	InnoDB	16.0 KB	
<input type="checkbox"/> uzsak_programa		0	InnoDB	16.0 KB	
<input type="checkbox"/> vaizduok		0	InnoDB	16.0 KB	
<input type="checkbox"/> vartotojai		0	InnoDB	16.0 KB	

3.22 pav. Duomenų bazės „itdb\_s“ struktūra

Aprašome visas duomenų bazės „itdb\_s“ sudarančias lenteles

Laukas	Tipas	Atributai	Null	Nutylint	Papild
CD	int(5)		Ne	0	
Gamint	varchar(50)		Ne		
Pavad	varchar(50)		Ne		
Garantija	int(2)		Ne	0	
Bukle	varchar(20)		Ne		
PirkID	int(10)		Taip	0	
pirkimodat	date		Ne	0000-00-00	
Kaina	varchar(6)		Ne		
faktNR	int(10)		Ne	0	

3.23 pav. Duomenų bazės lentelė „CDbox“

## CDbox

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
CD	Unikalus CD įrenginio numeris	Sveikas skaičius, pirminis raktas
Gamint	Gamintojo pavadinimas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija, m	Sveikas skaičius <=2 skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
Pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Pirkimo kaina	<=6 simbolių
FaktNR	Pirkinio faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint	Pa
DVDUN	int(11)		Ne	0	
Gamint	varchar(50)		Taip	NULL	
Pavad	varchar(50)		Taip	NULL	
Garantija	int(2)		Taip	NULL	
Bukle	varchar(20)		Taip	NULL	
PirkID	int(10)		Taip	0	
pirkimodat	date		Taip	NULL	
Kaina	varchar(6)		Taip	NULL	
faktnr	int(10)		Ne	0	

## 3.24 pav. Duomenų bazės lentelė "DVD"

## DVD

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
DVD	Unikalus CD įrenginio numeris	Sveikas skaičius, pirminis raktas
Gamint	Gamintojo pavadinimas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija, m	Sveikas skaičius <=2 skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
Pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Pirkimo kaina	<=6 simbolių
FaktNR	Pirkinio faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint
Vard	varchar(50)		Taip	NULL
Pav	varchar(50)		Taip	NULL
vardvard	varchar(10)		Ne	
ak	int(10)		Taip	NULL
telefon	varchar(50)		Taip	NULL
DarbViet	varchar(50)		Taip	NULL
TabNR	int(11)		Ne	0
slaptazodis	varchar(15)		Ne	

3.25 pav. Duomenų bazės lentelė “Darbuotojai“

3.12 Lentelė

### Darbuotojai

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
Vard	Vartotojo vardas	<=50 simbolių
Pav	Vartotojo pavardė	<=50 simbolių
vardvard	Vartoto unikalus vardas	<=50 simbolių
ak	Asmens kodas	Sveikas skaičius <=10 skaitmenų
telefon	Telefono numeris	<=20 simbolių
DarbViet	Darbo vieta	Sveikas skaičius <=10 skaitmenų
TabNR	Tabelio numeris	Sveikas skaičius <=11 skaitmenų
slaptazodis	Pirkimo kaina	<=15 simbolių

Lentelė HDD

Laukas	Tipas	Atributai	Null	Nutylint
HddUN	int(11)		Ne	0
gamintojas	varchar(50)		Taip	NULL
pavad	varchar(100)		Ne	
garantija	int(2)		Ne	0
Bukle	varchar(20)		Ne	
PirkID	int(10)		Ne	0
pirkimodat	date		Ne	0000-00-00
Kaina	varchar(6)		Ne	
faktnr	int(10)		Ne	0

3.26 pav. Duomenų bazės lentelė “HDD“

3.13 Lentelė

### HDD

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
HddUN	HDD įrenginio uniklaus vardas	<=50 simbolių

Gamintojas	Gamintojo vardas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija,m	Sveikas skaičius <=2skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Tabelio numeris	<=6 simbolių
faktNR	Pirkimo faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint
UN	int(11)		Ne	0
Pavad	varchar(50)		Taip	NULL
VAIZD	int(11)		Taip	NULL
PROC	varchar(50)		Taip	NULL
SP	int(11)		Taip	NULL
CD	int(11)		Taip	NULL
DVD	int(11)		Taip	NULL
HDD	int(11)		Taip	NULL
RAM	varchar(50)		Taip	NULL
korpusas	varchar(50)		Taip	NULL
mouse	varchar(50)		Taip	NULL
Tabel	int(11)		Taip	NULL
Oper	varchar(11)		Taip	NULL
Dislok	int(11)		Taip	NULL
reddat	date		Ne	0000-00-00

3.27 pav. Duomenų bazės lentelė “Kompiuteriai”

3.14 Lentelė

### Kompiuteriai

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
UN	Darbo vietos uniklaus numeris	Sveikas skaičius <=11 skaitmenų
Pavad	Darbo vietos kompiuterio pavad.	<=50 simbolių
Vaizd	Pasirenkama iš Vaizd įrenginių sąrašo	Sveikas skaičius <=11 skaitmenų
PROC	Procesoriaus dažnis, Ghz	<=50 simbolių
SP	Pasirenkama iš Sisteminės plokštės įrenginių sąrašo	Sveikas skaičius <=11 skaitmenų
CD	Pasirenkama iš CD įrenginių sąrašo	Sveikas skaičius <=11 skaitmenų
DVD	Pasirenkama iš DVD įrenginių sąrašo	Sveikas skaičius <=11 skaitmenų



HDD	Pasirenkama iš HDD įrenginių sąrašo	Sveikas skaičius <=11 skaitmenų
RAM	Operatyvinės atmintis, MB	<=50 simbolių
Korpusas	Sisteminio bloko korpusas tipas	<=50 simbolių
mouse	Pelės pavadinimas (tipas)	<=50 simbolių
Tabel	Pasirenkama iš Darbuotojai lentelės sąrašo	Sveikas skaičius <=11 skaitmenų
Oper	Operacinė sistema	<=11 simbolių
Dislok	Pasirenkama iš bVieta lentelės sąrašo	Sveikas skaičius <=11 skaitmenų
reddat	Įvedimo data	data yyyy-mm-dd formatas

Laukas	Tipas	Atributai	Null	Nutylint
PlokstesUN	int(5)		Ne	0
Gamint	varchar(25)		Ne	
Pavad	varchar(25)		Ne	
Garant	int(2)		Ne	0
Bukle	varchar(25)		Ne	
PirkID	int(10)		Ne	0
PirkimoDat	date		Ne	0000-00-00
Kaina	float		Ne	0
FaktNR	int(10)		Ne	0

3.28 pav. Duomenų bazės lentelė “LanPlok“

3.15 lentelė

### LAN Plokštė

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
PlokstesUN	LAN Plokštės uniklaus vardas	Sveikas skaičius <=5 skaitmenų
gamint	Gamintojo vardas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija, m	Sveikas skaičius <=2 skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
PirkimoDat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Įrenginio kaina	Slenkančio kablelio skaičius
faktNR	Pirkimo faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint P
<u>ID</u>	int(11)		Ne	0
prgrpav	varchar(50)		Taip	NULL
versija	varchar(50)		Taip	NULL
licpav	varchar(50)		Taip	NULL
reddat	date		Taip	NULL

3.29 pav. Duomenų bazės lentelė “Nekomer\_prog“

3.16 lentelė

### Nekomercinė įranga

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
ID	Programinės įrangos ID	Sveikas skaičius <=11 skaitmenų
prgpav	Programos pavadinimas	<=50 simbolių
versija	Programos versija	<=50 simbolių
licpav	Licenzijos pavadinimas	<=50 simbolių
reddat	Įvedimo data	data yyyy-mm-dd formatas

Laukas	Tipas	Atributai	Null	Nutylint F
<u>SP</u>	int(11)		Ne	0
Gamint	varchar(50)		Taip	NULL
pavad	varchar(50)		Ne	
tipas	varchar(50)		Taip	NULL
Garantija	int(2)		Ne	0
Bukle	varchar(20)		Ne	
Magdaznis	varchar(50)		Taip	NULL
pirkID	int(10)		Ne	0
Pirkimo_data	date		Ne	0000-00-00
Kaina	float		Ne	0
FakturosNR	int(10)		Ne	0

3.30 pav. Duomenų bazės lentelė “SisPlok“

3.17 lentelė

### SIS plokštė

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
SP	Sisteminės plokštės unikalūs vardas	Sveikas skaičius <=11 skaitmenų
Gamint	Gamintojo vardas	<=50 simbolių
pavad	Įrenginio pavadinimas	<=50 simbolių
tipas	Sisteminės plokštės tipas	<=50 simbolių
Garantija	Garantija,m	Sveikas skaičius <=2 skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
Magdaznis	Magistralės dažnis, Mhz	<=50 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų

pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Plokštės kaina	Slenkančio kabelio skaičius
FakturosNR	Pirkimo faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint	P.
VAIZD	int(11)		Ne	0	
Gamint	varchar(50)		Taip	NULL	
pavad	varchar(50)		Ne		
Garantija	int(2)		Ne	0	
Bukle	varchar(20)		Ne		
pirkimoID	int(10)		Ne	0	
pirkimodata	date		Ne	0000-00-00	
Kaina	float		Ne	0	
FakturosNR	int(10)		Ne	0	

3.31 pav. Duomenų bazės lentelė “Vaizduok“

3.18 lentelė

### Vaizduoklis

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
VAIZD	Vaizduoklio uniklaus vardas	Sveikas skaičius <=11 skaitmenų
Gamint	Gamintojo vardas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija,m	Sveikas skaičius <=2skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkimoID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Įrenginio kaina	Slenkančio kabelio skaičius
fakturosNR	Pirkimo faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint	P.
ID	int(11)		Ne	0	
Miestas	varchar(50)		Taip	NULL	
gatve	varchar(50)		Taip	NULL	
Kabinetas	varchar(50)		Taip	NULL	

3.32 pav. Duomenų bazės lentelė “bVieta“

3.19 lentelė

### Buvimo vieta

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
-------------------	-----------	---------------------

ID	Eilės numeris	Sveikas skaičius <=11 skaitmenų
Miestas	Miesto pavadinimas	<=50 simbolių
gatve	Gatvės pavadinimas	<=50 simbolių
Kabinetas	Kabineto numeris	<=50 simbolių

Laukas	Tipas	Atributai	Null	Nutylint
DV	varchar(4)		Taip	NULL
gedtipas	varchar(50)		Taip	NULL
apibud	text		Taip	NULL
data	date		Ne	0000-00-00
regvard	varchar(50)		Taip	NULL

3.33 pav. Duomenų bazės lentelė “gedimai”

3.20 lentelė

### Gedimai

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
DV	Darbo vietos ID Pasirenkama iš darbo vietos sąrašo	<=4 simbolių
gedtipas	Gedimo tipas Pasirenkama iš meniu sąrašo	<=50 simbolių
apibud	Gedimo aprašymas	Teksto formatas iki 65535 simbolių
Data	Įvedimo data	data yyyy-mm-dd formatas

Laukas	Tipas	Atributai	Null	Nutylint
ID	int(11)		Ne	
kompID	int(11)		Ne	0
kiekis	varchar(50)		Taip	NULL
pirkID	int(11)		Ne	0
Data	date		Ne	0000-00-00

3.34 pav. Duomenų bazės lentelė “pask\_det”

3.21 lentelė

### Detalių paskirstymas

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
-------------------	-----------	---------------------

ID	Paskirstymo ID	Sveikas skaičius <=11 skaitmenų
kompID	Kompiuterio ID Pasirenkama iš meniu sąrašo	Sveikas skaičius <=11 skaitmenų
kiekis	Gedimo aprašymas	<=50 simbolių
Data	Įvedimo data	data yyyy-mm-dd formatas
pirkID	Pirkimo ID Pasirenkama iš meniu sąrašo	Sveikas skaičius <=11 skaitmenų
data	Įvedimo data	data yyyy-mm-dd formatas

Laukas	Tipas	Atributai	Null	Nutylint
ID	int(11)		Ne	
perpavad	varchar(50)		Ne	
gamintojas	varchar(50)		Taip	NULL
pavad	varchar(50)		Ne	
garantija	int(2)		Ne	0
Bukle	varchar(20)		Ne	
pirkID	int(10)		Ne	0
pirkimodat	date		Ne	0000-00-00
Kaina	float		Ne	0
faktnr	int(10)		Ne	0

3.35 pav. Duomenų bazės lentelė “periferija“

3.22 lentelė

### Periferija

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
ID	Periferinio įrenginio uniklaus vardas	Sveikas skaičius <=2skaitmenų
perpavad	Periferinio įrenginio tipas	<=50 simbolių
gamintojas	Gamintojo vardas	<=50 simbolių
Pavad	Įrenginio pavadinimas	<=50 simbolių
Garantija	Garantija,m	Sveikas skaičius <=2skaitmenų
Bukle	Įrenginio statusas	<=20 simbolių
PirkID	Pirkimo kodas	Sveikas skaičius <=10 skaitmenų
pirkimodat	Pirkimo data	data yyyy-mm-dd formatas
Kaina	Įrenginio kaina	Slenkančio kablelio skaičius
fakturosNR	Pirkimo faktūros numeris	Sveikas skaičius <=10 skaitmenų

Laukas	Tipas	Atributai	Null	Nutylint	Papildomai
NR	int(11)		Ne		auto_increment
DV	int(11)		Ne	0	
ireng_pavad	varchar(25)		Ne		
apibud	varchar(255)		Ne	0	
data	date		Ne	0000-00-00	
regvardas	varchar(25)		Ne		

3.36 pav. Duomenų bazės lentelė “ uzsak\_irenginys“

3.23 lentelė

### Naujas įrenginys

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
NR	Eilės numeracija	Automatiškai didėjantis
DV	Darbo vietos ID Pasirenkama iš darbo vietos sąrašo	Sveikas skaičius <=11 skaitmenų
Ireng_pavad	Įrenginio pavadinimas	<=25 simbolių
kiekis	Gedimo aprašymas	Sveikas skaičius <=5 skaitmenų
Data	Įvedimo data	data yyyy-mm-dd formatas
regvardas	Vartotojo unikalus vardas	<=25 simbolių

Laukas	Tipas	Atributai	Null	Nutylint	Papildomai
NR	int(11)		Ne		auto_increment
DV	int(11)		Ne	0	
progr_pavad	varchar(25)		Ne		
aprasymas	text		Ne		
data	date		Ne	0000-00-00	
regvardas	varchar(25)		Ne		

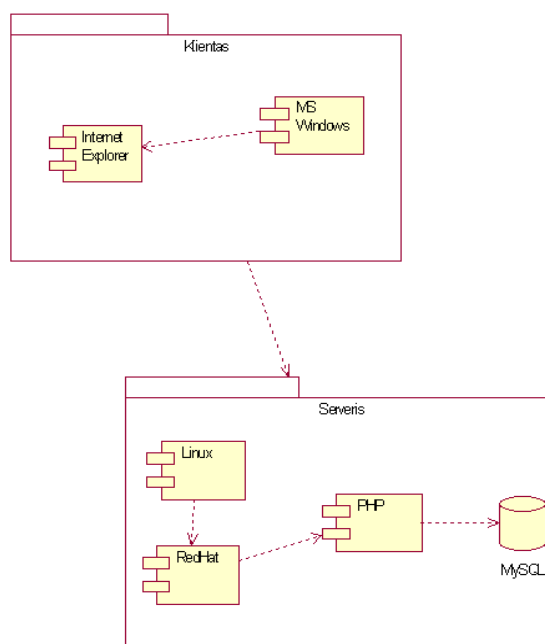
3.37 pav. Duomenų bazės lentelė “ uzsak\_programa“

3.24 lentelė

### Užsakyta programinė įranga

Lauko pavadinimas	Aprašymas	Formatas, ribojimai
NR	Eilės numeracija	Automatiškai didėjantis
DV	Darbo vietos ID Pasirenkama iš darbo vietos sąrašo	Sveikas skaičius <=11 skaitmenų
progr_pavad	Programos pavadinimas	<=25 simbolių
aprašymas	Programos aprašymas	Teksto formatas iki 65535 simbolių
Data	Įvedimo data	data yyyy-mm-dd formatas
regvardas	Vartotojo unikalus vardas	<=25 simbolių

### 3.8 Sistemos fizinis išpildymo modelis



**3.38 pav. Fizinis išpildymo modelis**

Informacinė sistema realizuota kliento-serverio technologija. Serverio operacinei sistemai panaudota Linux. Naudojamas Apache interneto serveris ir dinaminių puslapių generavimo programavimo kalba PHP. Duomenys saugomi MySQL duomenų bazėje. Prie jos jungiamasi naudojant standartines PHP funkcijas. Kliento dalis susideda iš Windows operacinės sistemos ir naršyklės Internet Explorer, kurios pagalba galima dirbti su šia sistema. Kuriant šią sistemą buvo naudotasi šiomis programomis:

1. CoffeeCup Direct FTP 5.2;
2. Microsoft FrontPage 2000;
3. TopStyle Pro 3.10;
4. PHPmyAdmin 2.6.0;
5. Rational Rose Enterprise;
6. Microsoft Visio Professional 2002

### 3.9 Sistemos testavimas

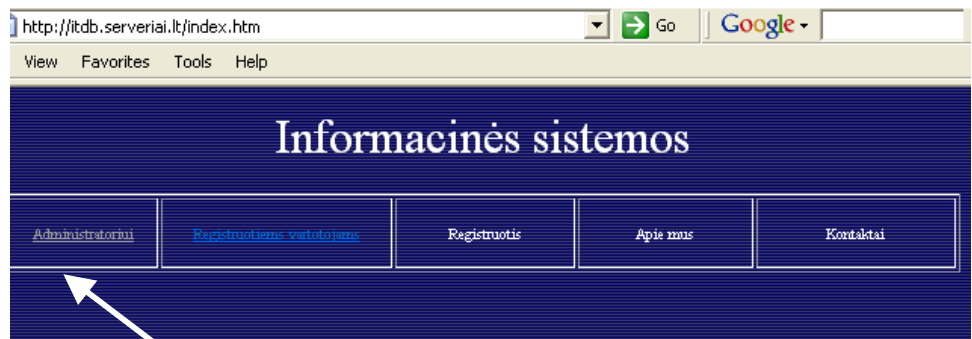
Sistemos programavimas ir testavimas atliktas to paties žmogaus

Naudojamas statinis ir dinaminis testavimas:

1. Programinės įrangos kodo peržiūra. Programuotojas peržiūri savo kodą ir tokiu būdu taiso klaidas.
2. Programinės įrangos testavimas. Testuojama su iš anksto paruoštais duomenimis, pateikiant juos sukompiliuotai programai.

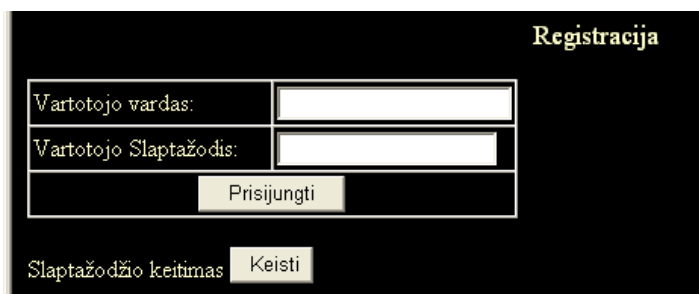
### 3.10 Vartotojo vadovas

#### 3.10.1 Prisijungimas prie „informacinės sistemos“ duomenų bazės



#### 3.39 Pradinis „Informacinės sistemos“ prisijungimo langas

Paspaudus „administratoriui“ atsidaro prisijungimo langas

A screenshot of a login form titled 'Registracija' in the top right corner. The form has a black background with white text and input fields. It contains two input fields: 'Vartotojo vardas:' and 'Vartotojo Slaptažodis:'. Below these fields is a 'Prisijungti' button. At the bottom left, there is a link 'Slaptažodžio keitimas' and a 'Keisti' button.

#### 3.40. Registracija patekti į duomenų administravimo puslapį.



Įvedus registruotą prisijungimo vardą ir slaptažodį sistema atidaro pagrindinį valdymo langą.



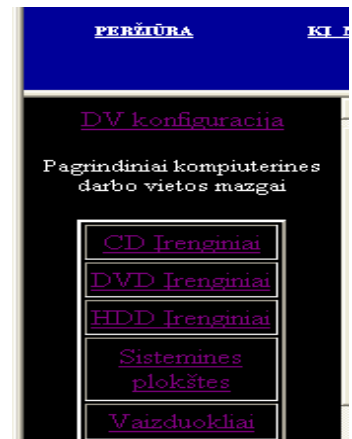
3.41pav. Pagrindinis duomenų valdymo sistemos langas.

Yra 3 valdymo sistemos sritys:

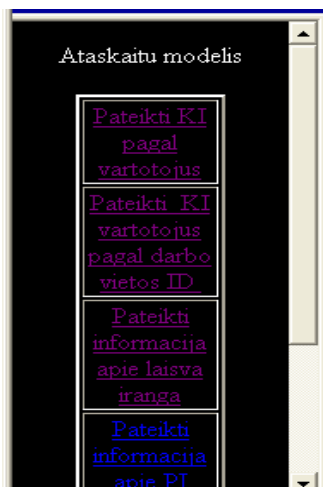
#### 1. Informacinės sistemos meniu.

Šioje srityje iš meniu galimi pasirinkimai

- a) **Peržiūra**- 2 srityje pateikia įrenginių ir darbo vietos pasirinkimo meniu langą. (3.41 pav.)
- b) **KI naudotojai** – 3 srityje pateikia įvestų į sistemą darbuotojų įvestus duomenis.
- c) **Pirkimai** – 3 srityje pateikia pirktos įrangos registravimo duomenis.
- d) **Detalių paskirstymas**- 3 srityje pateikia įrangos ir detalių paskirstymo langą.
- e) **Ataskaitos**- 2 srityje atidaro ataskaitų modelio langą



3.42 pav. 2 srities išvestas meniu langas.



3.43 pav. 2 srities užsakymų registravimo langas

3.44 pav. 2 srities ataskaitų modelio langas.

f) **Užsakymai**- 3 srityje atidaro užsakymų registravimo langą (51 pav.).

g) **Atsijungti** –atlieka vartotojo išregistravimo iš sistemos funkciją ir pasiūlo uždaryti Internet Explorer programą.

**2. Informacinės sistemos pagalbinis meniu.**

**3. Sistemos duomenų išvedimo langas.**

### 3.10.2 Naujos darbo vietos registravimas

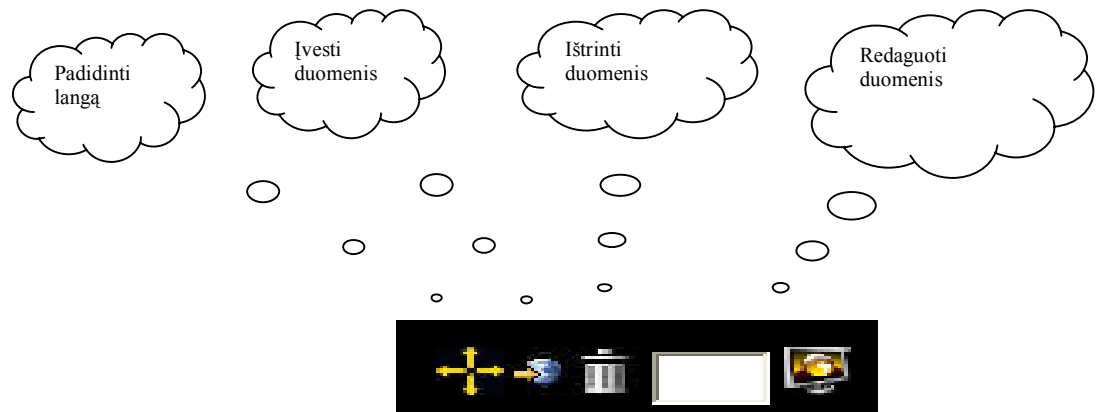
Įrenginių ir darbo vietos pasirinkimo lange parenkame „DV konfigūracija“ meniu punktą. Sistema išveda kompiuterinės darbo vietos lentelę.

Duomenų redagavimo sritis

UN	Pav	Vaizd	Proc	Sist_plok	CD	DVD	HDD	RAM	Korp	Pele	Vart_tab	O_sist	B_viet
22	gras	4	11	3	8	2	3	125	fsf	dsdf	123456	123456	103
44		4		3	8	2	3				123456		103
100	Abrrm	4	3,2	3	9	2	3	512	towel	wireless	123456	windows XP	103
101	dfd	88	1	6	11	5	11	64			123456		103
905	Apile	4	2.5Ghz	1	2	2	3	256	towel	mitsuhi	1235	1235	103
12321		4		3	5	2	3		103		123456	123456	103

3.45 pav. Vartotojo kompiuterinės darbo vietos duomenys

Lentelių laukų aprašymai randasi 3.7 skyriuje.



3.46 pav. Duomenų redagavimo sritis

Pasirenkame duomenų įvedimo langą.

Universalus numeris:

Pavadinimas:

Procesorius:

RAM atmintine:

Korpusas:

Pele:

Operacine sistema:

Ivedimo data:

Ivesti foto:  Browse...

Sistemine plokste:

Vaizduoklis:

CD irenginys:

DVD irenginys:

HDD irenginys:

Atsakingas darbuotojas:

Irenginio buvimo vieta:

3.47 pav. Vartotojo naujos kompiuterinės darbo vietos įvedimo langas

Darbo vietos registravimo lange parenkame registravimo duomenis. Dalį duomenų (standartiniai įrenginiai) parenkame iš sąrašo. Sąrašas formuojamas iš SP, Vaizd, CD, DVD, HDD ir kitų lentelių esančių informacinėje duomenų bazėje. Taigi norint užregistruoti naują darbo vietą pirmiausiai reikia įtraukti informaciją apie standartinius įrenginius. Įvedimo datą formuoja sistema automatiškai. Yra galimybė įvesti grafinį vaizdą.

Darbo vietos redagavimas

Universalus numeris: 22

Pavadinimas: gras

Procesorius: 11

RAM atmintine: 125

Korpusas: fsf

Pele: dsdf

Operacine sistema: 123456

Ivedimo data: 2004.11.09

Sistemine plokste ----- 3—PA7V600 KT600

Vaizduoklis ----- 3—Flatron L1510P 15"

CD irenginys ----- 8—freeCD—111

DVD irenginys ----- 2—DVD-/RW 4X

HDD irenginys ----- 3—80GB 7200rpm Baracuda

Atsakingas darbuotojas ----- 123456—kikas—kikaitis

Irenginio buvimo vieta ----- 103—Moletai—Jaunimo 55a

Išsaugoti pakeitimus

3.48 pav. Kompiuterinės darbo vietos redagavimo langas

Kompiuterinės DV redagavimas atliekamas keičiant vieną ar kitą laukelio įrašą ir spaudžiant „Išsaugoti pakeitimus“

Kompiuterines darbo vietos šalinimas is duomenu bazes

universalus numeris

Istrinti

3.49 pav. Kompiuterinės darbo vietos šalinimo langas

Įrašų trynimas atliekamas parenkant įrenginio unikalią numerį ir spaudžiant „Ištrinti“

### 3.10.3 Sudėtinės įrangos registravimas

DV konfigūracija

Pagrindiniai kompiuterinės darbo vietos mazgai

CD įrenginiai

DVD įrenginiai

HDD įrenginiai

Sisteminės plokštės

Vazduokliai

Periferija

Programinė įranga

Modemai

CD įrenginiai Viso: 11 įrašai.

CD_UN	Gamintojas	Pavadinimas	Garantija	Busena_Komp_ID	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_nr
1	Lite On	CD-ROM-52	2	Panaudotas	123456	2001-02-01	120	112121
2	Motorola	Wind	3	Panaudotas	12345	2001-05-02	120	120451
3	tec	stay	2	Laisvas	123244	2004-02-01	125	12333
4	loas	gogo	3	2	123212	2001-02-04	125	1313213
5	factory	step by step	1	Panaudotas	14111	2001-02-04	100	436454
6	teac	topps	2	Panaudotas	123213	2001-02-01	120	123213
7			0	Laisvas	0	0000-00-00	0	0
8	afa	freeCD	1	111	123145	2001-01-20	125	120011
9	Seagate	star2002	1	100	121245	2001-04-02	150	121244
10	sharp	ggas	1	Laisvas	112345	2004-05-01	100	737877
11			0	101	0	0000-00-00	0	0

3.50 pav. CD įrenginių langas

DV konfigūracija

Pagrindiniai kompiuterinės darbo vietos mazgai

CD įrenginiai

DVD įrenginiai

HDD įrenginiai

Sisteminės plokštės

Vazduokliai

Sisteminės plokštės Yra 5 įrašai.

SP_UN	Gamintojas	Pavadinimas	Tipas	Garantija	Busena_Komp_ID	Magistralės_daznis	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_NR
1	Intel	*MILLERSBURG*	Pentium 4 (5475)	3	Laisvas	333	126111	2002-01-01	570	123456
2	Intel	*MILLERSBURG*	Pentium 4 (5475)	2	Laisvas	100	126144	2002-03-04	860	123456
3	ASUS	PA71600 KT600	Sodet A	3	Laisvas	333	126145	2003-02-01	460	123456
5				0	Laisvas		0	0000-00-00	0	0
6	alpha	goodtime	shni	2	Laisvas	333	123456	2001-02-03	500.25	147741

3.51 pav. Sisteminių plokščių langas

konfigūracija

Pagrindiniai kompiuterinės darbo vietos mazgai

CD įrenginiai

DVD įrenginiai

HDD įrenginiai

Sisteminės plokštės

Vazduokliai

Periferija

Programinė įranga

HDD įrenginiai Yra 6 įrašai.

HDD_UN	Gamintojas	Pavadinimas	Garantija	Busena_Komp_ID	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_NR
1	SAMSUNG	40,0Gb 7200rpm 2Mb cache	3	Laisvas	125101	2002-04-05	420	124111
2	MAXTOR	80Gb 7200 rpm 3MB cache	1	Laisvas	124444	2002-05-04	455	123456
3	SEAGATE	80GB 7200rpm Barauda	1	Laisvas	145555	2001-04-05	300	125687
4			0	Laisvas	0	0000-00-00	0	0
5			0	Laisvas	0	0000-00-00	0	0
11			0	Laisvas	0	0000-00-00	0	0

3.52 pav. HDD įrenginių langas

[DV konfiguracija](#)  
 Pagrindiniai kompiuterinės darbo vietos mazgai

[CD įrenginiai](#)  
[DVD įrenginiai](#)  
[HDD įrenginiai](#)  
[Sistemines plokštes](#)  
[Vaizduokliai](#)  
[Periferija](#)

Vaizduokliai Yra 7 įrašai.								
Vaizd_UN	Gamintojas	Pavadinimas	Garantija	Busena_Komp_ID	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_NR
0			0	Laisvas	0	0000-00-00	0	0
1	LG	LCD L1710S 17"	3	Laisvas	146566	2002-10-10	750	146566
2	SAMSUNG	SM 172V LCD	1	Laisvas	127122	2001-02-04	400	789444
3	LG	Flatron L1910P 19"	1	Laisvas	127447	2004-01-02	480	111222
4	LG	Flatron x26	2	Laisvas	127111	2002-05-14	540	123222
11			0	Laisvas	0	0000-00-00	0	0
89	as	dasda	1	Laisvas	111	2004-01-01	120	213213

3.53 pav. Vaizduoklių langas

[DV konfiguracija](#)  
 Pagrindiniai kompiuterinės darbo vietos mazgai

[CD įrenginiai](#)  
[DVD įrenginiai](#)  
[HDD įrenginiai](#)  
[Sistemines plokštes](#)  
[Vaizduokliai](#)  
[Periferija](#)  
[Programine įranga](#)  
[Mūsų programos](#)

Periferija Yra 7 įrašai.									
PER_UN	PER pavadinimas	Gamintojas	Pavadinimas	Garantija	Busena_Komp_ID	Pirkimo_ID	Pirkimo_data	Kaina	buvimo vieta
1	Printeriškamėnis	HP	LaserJet 1100	1	Laisvas	0	2002-01-04	1000	124222
2	Printeriškamėnis	HP	psd 1210 all in one	2	Laisvas	0	2001-02-05	400	123454
3	skeneris	Mikrotek	Skeneris ScanMaker4800i USB Plus 1200x2400dpi, m	1	Laisvas	0	2002-10-12	1500	456546
4	skeneris	Visioneer	Skeneris 4400 42bit USB 600x1200dpi (televizoriu)	1	Laisvas	0	2001-02-04	1400	122221
10				0	Laisvas	0	0000-00-00	0	0
20				0	Laisvas	0	0000-00-00	0	0
22				0	Laisvas	0	0000-00-00	0	0

3.54 pav. Periferinių įrenginių langas

[DV konfiguracija](#)  
 Pagrindiniai kompiuterinės darbo vietos mazgai

[CD įrenginiai](#)  
[DVD įrenginiai](#)  
[HDD įrenginiai](#)  
[Sistemines plokštes](#)  
[Vaizduokliai](#)  
[Periferija](#)  
[Programine įranga](#)  
[Mūsų programos](#)

Programine įranga Yra 3 įrašai.				
UN	Programine įranga	Versija	Licenzija	Ivedimo_data
0	PHP	4	GPL	
3	MySQL	4	GPL	2004-02-29
22	Webrest	1.1	GPL	2001-02-01

3.55 pav. Programinės įrangos langas

Įrenginių registravimo, redagavimo ir trynimo operacijos atliekamos analogiškai kompiuterinės darbo vietos duomenų valdymo operacijoms.

### 3.10.4 Pirktos įrangos registravimas

[+ -](#)

Pirktos įrangos registravimas Yra 5 įrašai.									
Pirkimo_ID	PVM_sask	Pavadinimas	Tiekejas	Vnt_kaina	Kiekis	Inventorinis	Pvm	Visa_kaina_su_pvm	Pirkimo_data
3		sdisd		45	42	0	0	26	2002-01-01
4	12121321354A	diskas	aurimas	12	1	0	0.18	1.18	2004-02-03
5	123456	hard disk drive 40Gb	Baltica	0	12	0	0.18	0	2001-02-03
111				12	22	0	0.18	264	2004-10-24
121	123321	sound cards	Mikrovisata	12	25	123212	0.18	300	2004-10-24

3.56 pav. Pirktos įrangos langas

Pirktos įrangos registravimo forma

Pirkimo ID:

PVM saskaita:

Pavadinimas:

Tiekejas:

Vieneto kaina:

Kiekis:

Inventoris:

PVM:

Data:

3.57 pav. Pirktos įrangos įvedimo langas

### 3.10.5 Detalių paskirstymas

+ - [ ]

**Įrangos ir detalių paskirstymas** Viso: 10 įrašai

Eiles numeris	Darbo vietos ID	Kiekis	Pirkimo ID	Priskyrimo data
1	905	1	5	0000-00-00
2	905		5	0000-00-00
3	905	2	5	2004-09-15
4	905	1	5	2004-09-15
5	905	1	5	2004-09-15
6	905	1	5	2004-09-15
7	905	4	5	2004-09-15
8	12321	1	121	2004-11-06
9	905	10	5	2004-09-23
10	100	5	5	2004-11-07

3.58 pav. Detalių paskirstymo langas

Detalių paskyrimas

Eiles numeris:

Kiekis:

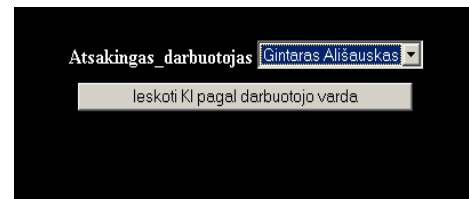
Priskyrimo data:

Darbo vietos ID ..... 12321—103 ▾

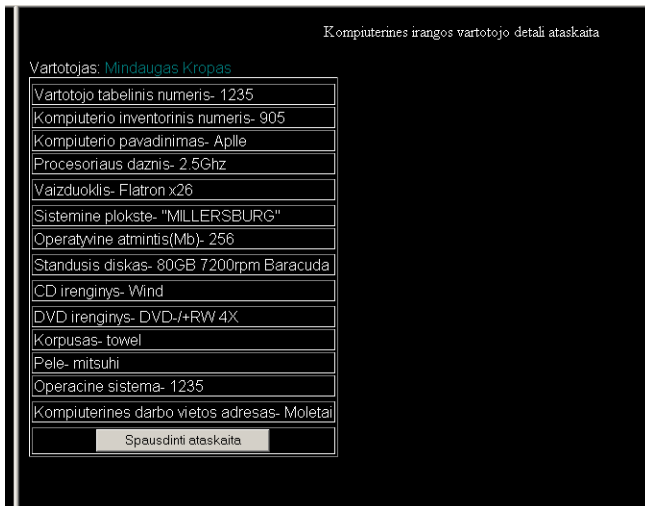
Pirkimo ID ..... 121—sound cards ▾

3.59 pav. Detalių įvedimo langas

### 3.10.6 Ataskaitų formavimas



3.60 pav. Darbuotojo pasirinkimas iš „menu“ lango.



3.61 pav. Atsakingo darbuotojo turimos įrangos detalios ataskaitos išvedimo langas.

Išvestą detalių darbuotojo turimos įrangos sąrašą galima atspausdinti spustelint „spausdinti ataskaitas“ mygtuką.



3.62 pav. Darbuotojo paieškos langas



3.63 pav. Darbuotojo darbo vietos išvedimo langas



Ataskaitu modelis

[Pateiktį KI pagal vartotojus](#)  
[Pateiktį KI vartotojus pagal darbo vietos ID](#)  
[Pateiktį informaciją apie laisvą įrangą](#)  
[Pateiktį informaciją apie PI](#)  
[Įrangos likutines vertes skaičiavimas](#)

CD įrenginiai Viso: 3 įrašai.									
CD	UN	Gamintojas	Pavadinimas	Garantija	Busena	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_nr
3		cec	stay	2	Laisvas	123244	2004-02-01	125	12333
7				0	Laisvas	0	0000-00-00		0
10		sharp	ggas	1	Laisvas	112345	2004-05-01	100	787877

DVD įrenginiai Yra 6 įrašai.									
DVD	UN	Gamintojas	Pavadinimas	Garantija	Busena	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_NR
1		PLEXTOR	DVD RW 8x4x12 IDE	2	Laisvas	124141	2002-05-02	140	145455
2		TEAC	DVD-r 4X	1	Laisvas	145147	2004-01-01	100	110204
3				0	Laisvas	0	0000-00-00		0
4		man	DVD	2	Laisvas	123456	2002-01-01	350	456789
5				0	Laisvas	0	0000-00-00		0
6		NEC	DVD r ND-1100A 4X DVD	1	Laisvas	145144	2001-02-04	210	124564

HDD įrenginiai Yra 6 įrašai.									
HDD	UN	Gamintojas	Pavadinimas	Garantija	Busena	Pirkimo_ID	Pirkimo_data	Kaina	Faktūros_NR
1		SAMSUNG	40,0Gb 7200rpm 2Mb cache	3	Laisvas	125101	2002-04-05	420	124111
2		MAXTOR	80Gb 7200 rpm 8MB cache	1	Laisvas	124444	2002-05-04	465	123456
3		SEAGATE	80GB 7200rpm Barrauda	1	Laisvas	145555	2001-04-05	300	125887
4				0	Laisvas	0	0000-00-00		0
5				0	Laisvas	0	0000-00-00		0

3.64 pav. Laisvos įrangos ataskaitos langas

### 3.10.7 Užsakymų, gedimų registravimas



KI vartotojas prisijungęs prie sistemos iš meniu lango gali pasirinkti registravimo pobūdį:

1. Gedimo registravimas;
2. Įrenginio užsakymas;
3. Programinės įrangos užsakymas.

3.65 pav. Vartotojų prisijungimo langas

Gedimų registravimo forma

DARBO VIETOS ID

GEDIMO POBŪDIS

Smulkiau Aprašykite

REGISTRAVO VARTOTOJAS

ĮVEDIMO DATA:

3.66 pav. Gedimų registravimo langas

### 3.11 Administratoriaus vadovas

Sistemos įdiegimas (serverio dalis)

1. MySQL serveryje sukuriama nauja duomenų bazė: itdb\_s
2. Iš CD diske esančio failo itdb\_s.sql duomenis perkeliame į MySQL serverį. Tam panaudojame PhpMyAdmin programinį įrankį. Dalyje „Vykdyti SQL sakinius duomenų bazėje“ nurodome -itdb\_s.sql. Tokiu būdu sukuriamos reikiamos duomenų bazės lentelės.
3. Į serverį perkeliame visus instaliacinius failus esančius CD disko kataloge „Programa“.

#### 4. SUKURTO INŽINERINIO PRODUKTO ĮVERTINIMAS

Sukurtas informacinis produktas buvo pateiktas tiems patiems vartotojams ir administratoriams, kurie pateikė anketinius duomenis IS kūrimo pradžioje (Priedas Nr.1) Vartotojams buvo suteikta IS internetinė prieiga: <http://itdb.serveriai.lt>

Beveik visi vartotojai pripažino sistemos naudingumą bei privalumą prieš lokalaus naudojimo duomenų bazes. Buvo paminėti sistemos trūkumai bei galimi papildymai. Atsižvelgiant į vartotojų pageidavimus ateityje numatoma:

1. patobulinti lentelių valdymą;
2. įvesti įrangos tiekėjams galimybę pateikti tinkamiausią konkrečiam vartotojui IT įrangos atnaujinimą, papildymą.
3. pateikti įrangos likutines vertes kiekvienam įrenginiui.

## 5. IŠVADOS

Šiame darbe nagrinėjome informacinės sistemos panaudojimą skirtą geografiškai paskirstytų klientų kreipimuisi registravimui, jų technologinių įrenginių duomenų aprašymui bei redagavimui. Informacinės sistemos analizės bei kūrimo etapuose buvo bandoma kiek įmanoma labiau priartėti prie išskeltų tikslų realizavimo. Apibendrinsime minėtus etapus:

1. Naujai kuriamos IS poreikių specifikuojimas. Šiems vartotojų poreikiams nustatyti buvo pasitelkta esamos IS analizė, bei IT vartotojų anketavimas. Vartotojų bei kompiuterinės įrangos administratorių apklausa parodė, jog naujai kuriama sistema yra aktuali vartotojams ir IT įrangos administratoriams.
2. Esamos Access pagrindu sukurtos informacinės sistemos tolimesnio vystymo perspektyvos. Pagal užduotus IS tikslus Access duomenų bazės tolimesnis panaudojimas pasirodė neperspektyvus.
3. Naujos duomenų bazės paieška. Apsiribojome atviro kodo duomenų bazėmis. Pagal IS sistemai keliamus reikalavimus MySQL labiausiai atitiko pasirinkimo kriterijus. Nemažą įtaką pasirinkimui turėjo ir programuotojo patirtis (1metai) dirbant su MySQL duomenų baze.
4. Palikuoninės duomenų bazės (Access) perkėlimas į naują (MySQL) duomenų bazę. Čia buvo būtina išsaugoti esamos informacinės sistemos duomenis ir juos perkelti į MySQL. Tam pasitelkėme atvirkštinę inžineriją. Panaudojome programinį įrankį- Visio Professional. Išgavome grafinį duomenų bazės vaizdą. Patogumo dėlei nusprendėme esamos duomenų bazės elementų- lentelių ir jų atributų nekoreguoti, bet visą struktūrą perkelti į MySQL aplinką. Tam panaudojome kitą programinį įrankį- Access to MySQL. Sistemos projekto metu kai kurios lentelės (jau MySQL) buvo papildytos naujais laukais, bei pakeisti laukų formatai.
5. Programinio įrankio kūrimo etapas. Čia svarbiausias buvo PHP kalbos objektinio programavimo principų panaudojimas duomenų bazės valdymui bei internetinių formų kūrimui. Objektinis programavimas leido efektyviau aprašyti programavimo procesą bei lengviau ateityje papildyti naujais objektais.

Sukurta informacinių technologijų išteklių duomenų bazė suteikia galimybę aprašyti įmonės informacinių technologijų įrenginius. Sistema atlieka įrenginių pirkimo registravimą, įrenginių priskyrimo atsakingiems vartotojams, duomenų atnaujinimo, papildymo išsaugojimo centralizuotoje duomenų bazėje operacijas. Taip pat sistema leidžia kompiuterinės įrangos vartotojams registruoti

užsakymus, įrangos gedimus, informacinius paklausimus. Registruoti vartotojai iš bet kurios darbo vietos turinčios ryšį su Internetu gali prisijungti prie sistemos ir gauti bei įvesti reikiamą informaciją. Taip pat registruoti pranešimą sistemos administratoriui. Tokiu būdu palengvinamas darbas kompiuterinės įrangos administratoriams, vartotojams. Efektyvesnis ir greitesnis tampa kompiuterinės įrangos vartotojų aptarnavimas.

## 6. NAUDOTOS LITERATŪROS SARAŠAS

1. CB Connolly T. and Begg C., Database system Addison Wesley, Third edition, 2002.
2. Paradauskas B., Nemuraitė L. Duomenų bazės ir semantiniai modeliai, Kaunas, 2002.
3. MySQL DBVS internetinis puslapis.  
[žiūrėta 2003 11 15] Prieiga per internetą: <<http://www.mySQL.com>>.
4. Enterprise Architect programinio paketo internetinis puslapis.  
[žiūrėta 2003 12 10] Prieiga per internetą: <<http://www.sparxsystems.com/>>.
5. UML specifikacija, [žiūrėta 2003 12 16] Prieiga per internetą:  
<<http://www.omg.org/technology/documents/formal/uml.htm>>.
6. UML istorija [žiūrėta 2003 10 27]<<http://cgi.omg.org/news/pr97/umlprimer.html>>.
7. Jeremy Allen. PHP 4 vadovas,- Kaunas: „Smaltijos“ leidykla, 2003.
8. Ian Gifillan MySQL 4 vadovas,- Kaunas: „Smaltijos“ leidykla, 2003.
9. Gudas S. Veiklos analizė ir informacinių poreikių specifikuojimas,- Kaunas „Morkūnas ir KO“, 2002.
10. Rational Rose programinio paketo internetinis puslapis [žiūrėta 2004 07 12] Prieiga per internetą: <http://www-306.ibm.com/software/rational/>.
11. Reda Alhajj. Extracting the extended entity-relationship model from a legacy relational database 2002. gruodis [žiūrėta 2004 06 30] Prieiga per internetą: <http://www.sciencedirect.com/>.
12. T.Magyla, V.Bagdonas. Methodology of operation efficiency prognosis // ISSN 1392-124X, *Information Technology and Control*, No 2 (23), 2002, Kaunas, Technologija.

## 7. NAUDOTŲ SUTRUMPINIMŲ SĄRAŠAS

- **CASE** – (Computer Aided Software Engineering) kompiuterinės programinės priemonės, skirtos projektavimui palengvinti.
- **IS** – informacinė sistema.
- **TCP/IP** (Transmission Control Protocol, IP- Internet Protocol) informacijos siuntimo protokolas.
- **ITDB** – informacinių technologijų duomenų bazė.
- **FTP** (File Transmission Protocol) failų siuntimo protokolas.
- **DB** – Duomenų Bazė.
- **KĮ** –kompiuterinė įranga.
- **DBVS** – Duomenų Bazių Valdymo Sistema.
- **DBAI** –Duomenų Bazių Atvirkštinė inžinerija.
- **HTML** – (Hyper Text Markup Language) Hipertekstinė kalba.
- **SQL** – (Structured Query Language), struktūrinė užklausų kalba.
- **PHP** – (Hypertext Preprocessor) Serverio programavimo kalba.
- **WWW** – (World Wide Web) pasaulinis kompiuterinis tinklas.
- **UML** – (Unified Modeling Language) modeliavimo kalba, naudojama objektiškai orientuotame projektavime.
- **ER** – esybių ryšių diagrama.
- **RR** – Rational Rose programinė įranga.
- **OM** – objektinis modeliavimas.

## 1 PRIEDAS. Anketos

### Anketa Nr.1 (kompiuterinės įrangos vartotojams turintiems tinklo administratorius)

1. Ar dažnai turite problemų su kompiuterine įranga?
  - b) kartą į savaitę;
  - c) kartą į mėnesį;
  - d) kartą į 3 mėnesius.
  
2. Koku būdu perduodate informaciją administratoriui apie kompiuterinės technikos problemas?
  - a) elektroniniu paštu;
  - b) telefonu;
  - c) tiesiogiai;
  - d) informacine sistema.
  
3. Kiek registruojamų kompiuterinės įrangos gedimų yra susijusių su Interneto sutrikimais. (neveikia Internetas? (proc.))
  - a) mažiau nei 25%;
  - b) nuo 25 iki 50%;
  - c) daugiau nei 50%;
  - d) Internetiniu tinklu nesinaudoju.
  
4. Ar žinote už kokią techninę ir programinę įrangą jūs esate atsakingas?
  - a) taip;
  - b) taip, bet nesu visiškai tikras;
  - c) ne.
  
5. Kaip dažnai prašote administratoriaus ar tiesioginio vadovo atnaujinti ar papildyti techninę bei programinę įrangą?
  - a) kartą į mėnesį;
  - b) kartą į 3 mėnesius;
  - c) kartą į metus.
  
6. Kiek darbo laiko (proc.) dirbate su kompiuteriu ?
  - a) iki 10% darbo laiko;
  - b) nuo 10 iki 50% darbo laiko;



- c) daugiau nei 50% darbo laiko.
7. Ar manote kad pagelbėtų efektyvesniam darbui jūsų kompiuteriui prieinama informacinė sistema skirta jūsų užsakymų/kreipimūsi susijusių su kompiuterine įranga registravimui?
- a) taip;  
b) ne;  
c) nežinau.
8. Ar jūsų kompiuterinėje darbo vietoje yra įdiegtas Internetas?
- a) taip;  
b) ne.
9. Ar esate patenkinti savo įmonės bendro naudojimo dokumentų pasiekiamumu?
- a) taip;  
b) iš dalies;  
c) ne.
10. Kiek per pastaruosius tris mėnesius kreipėtės į administratorių dėl kompiuterinės įrangos problemų?
- a) nesikreipiau;  
b) vieną;  
c) du;  
d) tris;  
e) daugiau nei tris.

## Anketa Nr. 2 (kompiuterinės įrangos administratoriams)

1. Ar naudojātės duomenų baze įrangos informacijai saugoti?
- e) Taip;  
f) Ne;  
g) Paprastos laikmenos (Word, Excel, )
2. Ar esate patenkinti savo įmonės bendro naudojimo dokumentų pasiekiamumu?
- a) taip;  
b) iš dalies;  
c) ne.
3. Kaip gaunate informaciją iš vartotojų apie kompiuterinės įrangos problemas?
- a) elektroniniu paštu;

- b) telefonu;
- c) tiesiogiai;
- d) informacinė sistema.

4. Ar jums reikalinga informacinė sistema skirta kompiuterinės įrangos registravimui bei analizei?

- a) taip;
- b) ne;
- c) nežinau.

### Anketa Nr. 3 (Sukurto programinio produkto įvertinimo anketa)

1. Kas jums patiko informacinėje sistemoje?

- a)
- b)
- c)

2. Kas Jums nepatiko dirbant su IS?

- a)
- b)
- c)
- d)

3. kaip jūs manote ar realiai tokia IS naudinga IT vartotojams ir administratoriams

- a) taip
- b) galbūt
- c) ne

4. Kokie yra IS trūkumai?

- a)
- b)
- c)

5. Kuo reikėtų papildyti pateiktą IS

- a)
- b)
- c)
- d)

## 2 PRIEDAS. Testavimo duomenys

Testavimai atliekami sistemoje registruotam vartotojui „jonjon“

3.26 lentelė

Testavimo duomenys

stuojamasis modulis	Duomenys	rezultatas
prisijungimas (login.php;login2.php)	jonjon, vejas	sėkmingas prisijungimas
prisijungimas (login.php;login2.php)	123456,123456	prisijungimas nepavyko
dv registravimas (investidarbovieta.php)	unikalus numeris-1, pavadinimas-delphi, procesorius-1ghz, ram- 512, korpusas- tower, pelė – standart, operacinė sistema- windows XP,(įrenginiai parenkami pagal meniu sąrašą)	parinkti duomenys užregistruoti
dv trynimas (istrintikompai.php)	Unikalus numeris- 22	įrašas pašalintas
pirkimo registravimas (pirkimai.php)	pirkimoid-1, pvm sąskaita- 1234566, pavadinimas- sprinteris, tiekėjas- balticum, vieneto kaina- 450, kiekis-1, inventorinis- 777788,	parinkti duomenys užregistruoti
detalių paskirstymas (paskirstymas.php)	DV, pirkimo id- pasirenkame iš meniu sąrašo, eilės numeris- 1, kiekis- 20	parinkti duomenys įvesti
gedimų registravimas( gedimas_reg.php)	DV, gedimų pobūdžio, vartotojo pasirinkimas pagal menu sąrašą, problemos apibūdinimas – „baigėsi spausdintuvo kartridžas“	parinkti duomenys užregistruoti

### **3 PRIEDAS. Kompaktinėje plokštelėje pateiktų programų sąrašas**

- a. ITDB valdymo programos kodas („php“ rinkmenos kataloge „Programinis kodas“)
- b. Projektavimo su Rational Rose paketu rinkmena (analizė ir projektavimas kataloge „ROSE“)
- c. ITDB lentelių aprašymo kodas (SQL kodas kataloge „ITDB“)

## 4 PRIEDAS. Access to MySQL, PhpMyAdmin programų išvesties rezultatai

Pilnas „Access to MySQL“ programos išvesties kodas yra pateiktas kompaktinėje plokštelėje.

```
# File name: C:\Documents and Settings\erlandas\Desktop\pmd.sql
# Creation date: 09/25/2004
# Created by Access to MySQL 3.2 [Demo]
# -----
# More conversion tools at http://www.convert-in.com

#
# Table structure for table 'CDbox'
#

DROP TABLE IF EXISTS `CDbox`;
CREATE TABLE `CDbox` (
  `CD` INT NOT NULL DEFAULT 0,
  `Gamint` VARCHAR(50),
  `Pavad` VARCHAR(50),
  `Pirkdat` CHAR(19),
  `Patabos` VARCHAR(200),
  PRIMARY KEY `PrimaryKey` (`CD`)
);

#
# Dumping data for table 'CDbox'
#

LOCK TABLES `CDbox` WRITE;
INSERT INTO `CDbox` VALUES(1, 'asdra', '', '2001-01-02', 'bkjvlkjhvkjhgkgh');
INSERT INTO `CDbox` VALUES(2, 'adfasf', '', '2002-01-01', '');
INSERT INTO `CDbox` VALUES(3, 'dghdghd', '', '2002-01-01', '');
INSERT INTO `CDbox` VALUES(4, 'debilas', '', '2003-02-02', 'gerasd daiktas');
INSERT INTO `CDbox` VALUES(5, 'debilas', '', '2002-02-02', 'bloga');
UNLOCK TABLES;
ALTER TABLE `CDbox` CHANGE `Pirkdat` `Pirkdat` DATE;

# Table structure for table 'Kompiuteriai'
#

DROP TABLE IF EXISTS `Kompiuteriai`;
CREATE TABLE `Kompiuteriai` (
  `UN` INT NOT NULL DEFAULT 0,
  `Pavad` VARCHAR(50),
  `VAIZD` INT DEFAULT 0,
  `PROC` VARCHAR(50),
  `SP` INT DEFAULT 0,
  `CD` INT DEFAULT 0,
  `DVD` INT DEFAULT 0,
  `HDD` INT DEFAULT 0,
```

```

`RAM` VARCHAR(50),
`Tabel` INT DEFAULT 0,
`Oper` INT DEFAULT 0,
`Dislok` INT DEFAULT 0,
INDEX `CDboxKompiuteriai` (`CD`),
INDEX `DarbuotojaiKompiuteriai` (`Tabel`),
INDEX `Dislok` (`Dislok`),
INDEX `DVDKompiuteriai` (`DVD`),
INDEX `HDDKompiuteriai` (`HDD`),
UNIQUE `KompID` (`UN`),
INDEX `Mon` (`VAIZD`),
INDEX `Periferija` (`Tabel`),
PRIMARY KEY `PrimaryKey` (`UN`),
INDEX `SistPIKompiuteriai` (`SP`),
INDEX `VaizduokKompiuteriai` (`VAIZD`),
INDEX `VietaKompiuteriai` (`Dislok`)
);

#
# Dumping data for table 'Kompiuteriai'
#

LOCK TABLES `Kompiuteriai` WRITE;
INSERT INTO `Kompiuteriai` VALUES(1001, 'Sonex', 2, '', NULL, 1, NULL, 25, '', 1234, NULL, NULL);
INSERT INTO `Kompiuteriai` VALUES(1002, 'Vetra II', NULL, '', NULL, 2, NULL, NULL, '', 1234, NULL, NULL);
INSERT INTO `Kompiuteriai` VALUES(1003, 'Audra', NULL, '', NULL, 3, NULL, NULL, '', 1235, NULL, NULL);
INSERT INTO `Kompiuteriai` VALUES(1004, 'Kompas', NULL, '', NULL, 2, NULL, NULL, '', 1236, NULL, NULL);
UNLOCK TABLES;

```

## „PhpMyAdmin“ programos rezultatai

Pilnas „PhpMyAdmin“ programos išvesties kodas yra pateiktas kompaktinėje plokštelėje.

```

-- phpMyAdmin SQL Dump
-- version 2.6.0-pl2
-- http://www.phpmyadmin.net
--
-- Darbinė stotis: localhost
-- Atlikimo laikas: 2004 m. Lapkričio 07 d. 22:45
-- Serverio versija: 4.0.21
-- PHP versija: 4.3.9
--
-- Duombazė: `itdb_s`
--
-----

--
-- Sukurta duomenø struktūra lentelei `CDbox`
--

CREATE TABLE `CDbox` (
  `CD` int(5) NOT NULL default '0',
  `Gamint` varchar(50) NOT NULL default '',
  `Pavad` varchar(50) NOT NULL default '',

```

```

`Garantija` int(2) NOT NULL default '0',
`Bukle` varchar(20) NOT NULL default "",
`PirkID` int(10) default '0',
`pirkimodat` date NOT NULL default '0000-00-00',
`Kaina` varchar(6) NOT NULL default "",
`faktNR` int(10) NOT NULL default '0',
PRIMARY KEY (`CD`),
KEY `Bukle` (`Bukle`)
) TYPE=InnoDB;

--
-- Sukurta duomenø kopija lentelei `CDbox`
--

INSERT INTO `CDbox` VALUES (1, 'Lite On', 'CD-ROM x52', 2, 'Panaudotas', 123456, '2001-02-01', '120', 112121);
INSERT INTO `CDbox` VALUES (2, 'Motorola', 'Wind', 3, 'Panaudotas', 12345, '2001-05-02', '120', 120451);
INSERT INTO `CDbox` VALUES (3, 'coc', 'stay', 2, 'Laisvas', 123244, '2004-02-01', '125', 12333);
INSERT INTO `CDbox` VALUES (4, 'loas', 'gogo', 3, '2', 123212, '2001-02-04', '125', 1313213);
INSERT INTO `CDbox` VALUES (5, 'factory', 'step by step', 1, 'Panaudotas', 14111, '2001-02-04', '100', 436454);
INSERT INTO `CDbox` VALUES (6, 'teac', 'loppls', 2, 'Panaudotas', 123213, '2001-02-01', '120', 123213);
INSERT INTO `CDbox` VALUES (7, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0);
INSERT INTO `CDbox` VALUES (8, 'afa', 'freeCD', 1, '111', 123145, '2001-01-20', '125', 120011);
INSERT INTO `CDbox` VALUES (9, 'Seagate', 'star2002', 1, '100', 121245, '2001-04-02', '150', 121244);
INSERT INTO `CDbox` VALUES (10, 'sharp', 'ggaas', 1, 'Laisvas', 112345, '2004-05-01', '100', 787877);
INSERT INTO `CDbox` VALUES (11, "", "", 0, '101', 0, '0000-00-00', "", 0);

-----

--
-- Sukurta duomenø struktûra lentelei `DVD`
--

CREATE TABLE `DVD` (
`DVDUN` int(11) NOT NULL default '0',
`Gamint` varchar(50) default NULL,
`Pavad` varchar(50) default NULL,
`Garantija` int(2) default NULL,
`Bukle` varchar(20) default NULL,
`PirkID` int(10) default '0',
`pirkimodat` date default NULL,
`Kaina` varchar(6) default NULL,
`faktnr` int(10) NOT NULL default '0',
`foto` varchar(25) NOT NULL default "",
PRIMARY KEY (`DVDUN`)
) TYPE=InnoDB;

--
-- Sukurta duomenø kopija lentelei `DVD`
--

INSERT INTO `DVD` VALUES (1, 'PLEXTOR', 'DVD+/-RW 8x4x12 IDE', 2, 'Laisvas', 124141, '2002-05-02', '140', 145455, "");
INSERT INTO `DVD` VALUES (2, 'TEAC', 'DVD-/+RW 4X', 1, 'Laisvas', 145147, '2004-01-01', '100', 110204, "");
INSERT INTO `DVD` VALUES (3, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0, "");
INSERT INTO `DVD` VALUES (4, 'man', 'DVD', 2, 'Laisvas', 123456, '2002-01-01', '350', 456789, "");
INSERT INTO `DVD` VALUES (5, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0, "");
INSERT INTO `DVD` VALUES (6, 'NEC', 'DVD+R/RW ND-1100A 4XDVD', 1, 'Laisvas', 145144, '2001-02-04', '210', 124564, 'profile.jpg');

-----

--
-- Sukurta duomenø struktûra lentelei `Darbuotojai`

```

```

--

CREATE TABLE `Darbuotojai` (
  `Vard` varchar(50) default NULL,
  `Pav` varchar(50) default NULL,
  `vartvard` varchar(10) NOT NULL default "",
  `ak` int(10) default NULL,
  `telefon` varchar(50) default NULL,
  `DarbViet` varchar(50) default NULL,
  `TabNR` int(11) NOT NULL default '0',
  `slaptazodis` varchar(15) NOT NULL default "",
  PRIMARY KEY (`TabNR`)
) TYPE=InnoDB;

--
-- Sukurta duomenø kopija lentelei `Darbuotojai`
--

INSERT INTO `Darbuotojai` VALUES ('222', "", "", 0, "", "", 0, "");
INSERT INTO `Darbuotojai` VALUES ('jonas', 'jonaitis', 'jonjon', 12231111, '121313', 'Utena', 1232, 'vejas');
INSERT INTO `Darbuotojai` VALUES ('Gintaras', 'Aliðauskas', 'gintali', 2147483647, '389 62245', 'Utena', 1234, 'uosis');
INSERT INTO `Darbuotojai` VALUES ('Mindaugas', 'Kropas', "", 2147483647, '381 54789', 'Anyksčiai', 1235, "");
INSERT INTO `Darbuotojai` VALUES ('Darius', 'Gutautas', 'dargut', 2147483647, '383 54788', 'Moletai', 1236, 'berzas');
INSERT INTO `Darbuotojai` VALUES ('Marius', 'Linus', "", 2147483647, 'Telefonas', 'Zarasai', 1237, "");
INSERT INTO `Darbuotojai` VALUES ('Tadas', 'Rimkus', "", 374121412, '868511111', 'Kaunas', 1474, "");
INSERT INTO `Darbuotojai` VALUES ('Tomas', 'Kutas', 'tomkut', 65654, '4545454', 'Utena', 7897, 'azuolas');
INSERT INTO `Darbuotojai` VALUES ('kikas', 'kikaitis', 'kikkik', 4545, '64646', 'utena', 123456, 'drebulė');

-----

--
-- Sukurta duomenø struktūra lentelei `HDD`
--

CREATE TABLE `HDD` (
  `HddUN` int(11) NOT NULL default '0',
  `gamintojas` varchar(50) default NULL,
  `pavad` varchar(100) NOT NULL default "",
  `garantija` int(2) NOT NULL default '0',
  `Bukle` varchar(20) NOT NULL default "",
  `PirkID` int(10) NOT NULL default '0',
  `pirkimodat` date NOT NULL default '0000-00-00',
  `Kaina` varchar(6) NOT NULL default "",
  `faktnr` int(10) NOT NULL default '0',
  PRIMARY KEY (`HddUN`),
  KEY `pavad` (`pavad`)
) TYPE=InnoDB;

--
-- Sukurta duomenø kopija lentelei `HDD`
--

INSERT INTO `HDD` VALUES (1, 'SAMSUNG', '40,0Gb 7200rpm 2Mb cache', 3, 'Laisvas', 125101, '2002-04-05', '420', 124111);
INSERT INTO `HDD` VALUES (2, 'MAXTOR', '80Gb 7200 rpm 8MB cache', 1, 'Laisvas', 124444, '2002-05-04', '455', 123456);
INSERT INTO `HDD` VALUES (3, 'SEAGATE', '80GB 7200rpm Baracuda', 1, 'Laisvas', 145555, '2001-04-05', '300', 125887);
INSERT INTO `HDD` VALUES (4, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0);
INSERT INTO `HDD` VALUES (5, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0);
INSERT INTO `HDD` VALUES (11, "", "", 0, 'Laisvas', 0, '0000-00-00', "", 0);

-----

```



