Kauno Technologijos Universitetas
INFORMATIKOS FAKULTETAS
Informacijos sistemų katedra

Ilgalaikių kontraktų vykdymo finansinės kontrolės IS

Magistro tezės

Magistrantas : Darius Martynaitis
Vadovas : doc.dr.Bronius Paradauskas
Recenzentas: doc.dr.Vytautas Pilkauskas

2005

1. Įvadas....3
2. Kontraktų sudarymo uždavinys....4
2.1 Dalykinės srities apibrėžimas....4
2.2 Uždavinio sąlygos....6
2.2.1 Uždavinio sąlygų keliami reikalavimai....6
2.2.2 Prekių identifikavimo problema....9
2.3 Egzistuojančios programinės įrangos problemai spręsti analizė....11
2.4 Uždavinio sprendimo kriterijai....12
2.5 Egzistuojantys sistemų modeliai ir sistemos tinkančios daliniam uždavinio sprendimui....14
2.5.1 Tipinis kontraktų modelis prekybai....14
2.5.2 Verslo valdymo sistemos....16
2.5.3 Egzistuojančių modelių ir sistemų tinkančių daliniam uždavinio sprendimui lyginamoji analizė....16
2.6 Sistemos projektavimo metodo ir priemonių parinkimas....18
2.7 Organizacijos veiklos analizė....20
2.7.1 Veiklos sąveikų modelis....20
2.7.2 Veiklos tikslų modelis....21
2.7.3 Veiklos panaudojimo atvejų modelis....22
2.7.4 Veiklos objektų modelis....23
2.7.5 Kontrakto sudarymo proceso diagrama....24
2.8 Analizės išvados....25
3. Informacijos sistemos projektavimas....26
3.1 Modifikuotas kontraktų modelis pagal uždavinio sąlygą....26
3.2 Modifikuoto kontaktų duomenų modelio praktinio taikymo kriterijai....28
3.3 Reikalavimų modelis....29
3.3.1 Vartotojų panaudojimo atvejų modelis....29
3.3.2 Specifikacijos panaudojimo atvejams....31
3.3.3 Programinių klasių diagrama....34
3.3.4 Vartotojo sąsajos modelis....35
3.4 Sistemos modelis....36
3.4.1 Sekų diagramos....36
3.4.2 Komponentų diagrama....38
3.4.3 Duomenų bazės modelis....38
3.4.4 Vartotojo sąsajos ir programinių klasių sąryšio diagrama....41
3.4.5 Sistemos proceso diagrama....43
3.4.6 Realizacijos modelis (Paskirstymo diagrama)....44
3.5 IS diegimo planas……..…….44
4. Sistemos programinės įrangos testavimas ir vartotojo vadovas……..46
4.1 Sukurtos sistemos vertinimas....46
4.2 Sistemos ateities tobulinimo perspektyvos ir alternatyvos....47
4.3 Testavimo modelis....48
4.4 Trumpas vartotojo vadovas....52
5. Darbo išvados....57
6. Trumpinių žodynas....58
7. Literatūros sąrašas....59
8. Santrauka anglų kalba....60
9. Priedai....61

1. Įvadas

Naujos, bei naujai atsirandančios technologijos leidžia daugumai sričių automatizuotis – sutaupoma laiko, finansinių resursų, o taipogi naujos sistemos garantuoja vis mažesnę klaidos tikimybę. Kuo mažiau įmonėje automatizuoti procesai, tuo didesnė tikimybė atsirasti klaidai, darbo našumas automatiškai tampa mažesnis ir galiausiai žmogus tampa nepajėgus susitvarkyti su dideliais informacijos kiekiais, tenka samdyti papildomų darbuotojų – padidėja organizacijos išlaidos, o klaidos tikimybė išlieka ta pati. Atsiradus klaidai, organizacija gali turėti nenuspėjamų padarinių, kurie gali pakenkti finansams arba pakenkti kitoms organizacijoms. Klaidas ištaisyti gali būti neįmanoma, nors jeigu ir įmanoma, tai vis vien būtų padaryta žala.

Šiame darbe automatizuosime dalį operacijų organizacijai, kuri atlieka tarpininko vaidmenį prekybos srityje. Prekės yra perkamos iš tiekėjų ir perparduodamos klientams. Su klientais ir pirkėjais yra pasirašomos sutartys, tačiau sutarčių nepakanka. Su tiekėjais yra pasirašomi kontraktai, kuriuose būna nurodytas prekių tipas,kiekis ir laikotarpis per kurį turi būti pristatytos prekės. Klientų yra keli, su kiekvienu klientu galima turėti daugiau negu vieną kontraktą. Prekių kiekiai būna dideli, prekės pristatomos dalimis už jas atsiskaitoma taipogi dalimis. Kuriamos IS pagrindiniai tikslai būtų: registruoti kontraktų duomenis ir gauti informaciją apie kiekvieno kontrakto būseną – įgyvendintas ir neįgyvendintas sąlygas.Visa tai registruoti ir stebėti žmogui, neturint specializuotos programinės įrangos per daug rizikinga, kaip minėta dėl didelės klaidos tikimybės. Kadangi organizacijos veikla yra prekyba, todėl iškyla dar viena problema – prekių identifikavimas – kiekviena prekė turi standartizuotą savo kodą, kuris nusako gamintoją, prekės rūšį ir t.t. Tačiau būna tokių atvejų ,kad prekių kurių prigimtis ir tipas yra visiškai skirtingi kodai sutampa. Sutikrinti kodus ,kai prekių yra daug, žmogus taip pat yra bejėgiškas

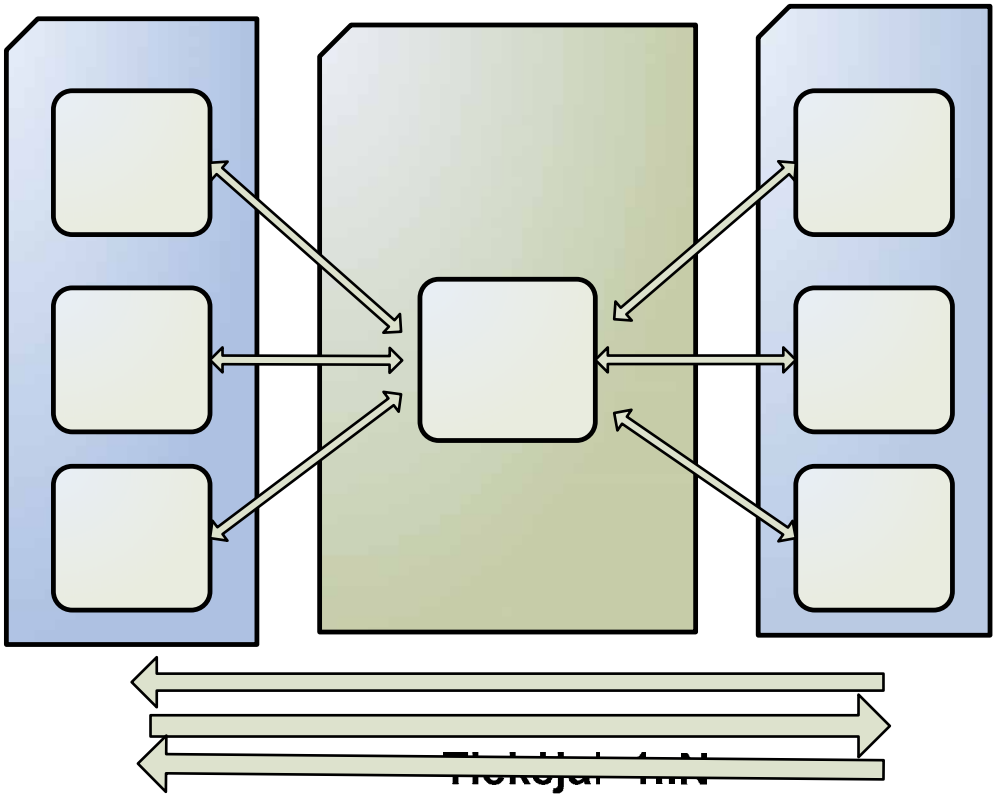
Šio darbo tikslas buvo – atlikti organizacijos automatizuojamų procesų analizę, suprojektuoti ir sukurti sistemą, naudojant optimalias priemones.

Antrame skyriuje yra pateikiamas detalus problemos parašas, parenkamas problemos sprendimo kelias, projektavimo metodų parinkimas, trečiame skyriuje - sistemos projektavimas, ketvirtame – sukurtos sistemos testavimas ir pateikiamas trumpas sistemos vartojimo vadovas. Penktame skyriuje pateikiamos išvados

2. Kontraktų finansinės kontrolės uždavinys

2.1 Dalykinės srities apibrėžimas

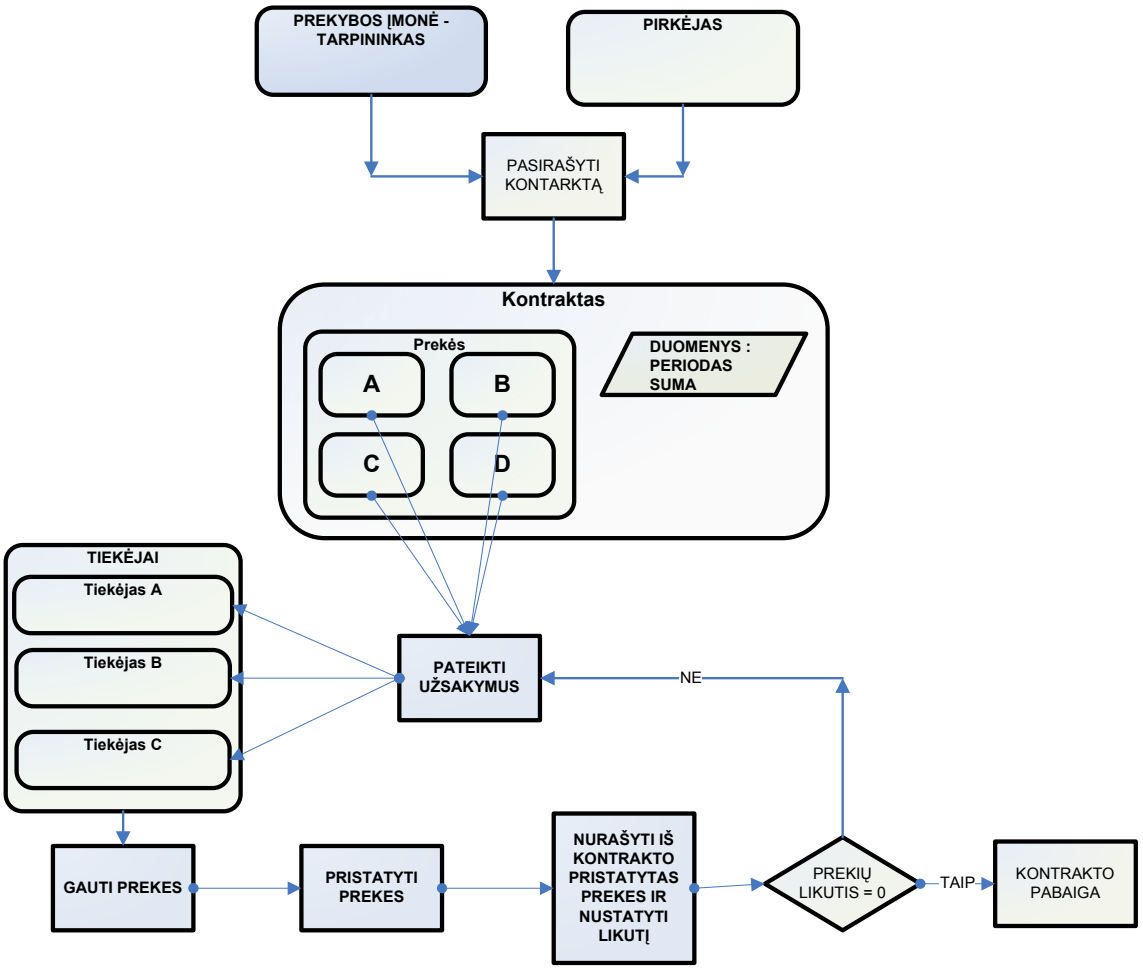
Kiekviena organizacija stengiasi automatizuoti savo veiklą. Automatizavimo aktualumas buvo aptartas įvade (2 psl). Iškyla kita problema – ką pirmiausia reikia automatizuoti, kokia turi būti automatizuojamų procesų hierarchija. Panagrinėsime organizacija,kuri atlieka tarpininko vaidmenį prekybos srityje. Egzistuoja trys objektų tipai : tiekėjai, pirkėjai ir pati organizacija. Tarp šių objektų egzistuoja prekių, pinigų ir užsakymų srautai. Šių objektų ryšį galime atvaizduoti žemiau pateikta schema. (1 pav)



1 pav Duomenų srautai nagrinėjamoje organizacijoje

Kaip matome užsakymų, prekių ir pinigų srautas kerta nagrinėjamos organizacijos sritį. Tai reiškia, kad šie srautai turi būti sekami ir žinomos jų reikšmės bei kiekiai. Nagrinėjamos organizacijos sritį pasiekęs kiekvienas srautas yra apdorojamas : dalis pinigų gautų iš pirkėjų nusėda prekybos įmonėje – tarpininke, taip pat čia yra apdorojami užsakymai ir paskirstomi tiekėjams pagal jų tipą. Vėliau yra surenkamos prekės ir

pristatamos klientams. Tiekėjų ir klientų skaičius, kaip matome iš pateiktos schemos (1 pav) yra (1..N), šiuo atveju (šioje schemoje) N nėra griežtai apibrėžtas, taigi klientų ir pirkėjų skaičius nebūtinai turi būti vienodas. Visa ši vykdoma komercinė veikla yra dokumentuojama – pasirašomi kontraktai dėl prekių tiekimo, kainų ir laikotarpio per kurį turi būti įgyvendintos visos kontarkto sąlygos. Analizuojamoje organizacijoje, taip pat kaip ir sako temos pavadinimas „Ilgalaikių kontraktų ...“, vyrauja dideli kontraktai : pasirašyto kontrakto sąlygos yra tenkinamos atskirais etapais, suteikiant kai kurioms sąlygoms prioritetus. Atsiskaitymas, šiuo atveju būtų už prekes, taipogi vyksta etapais, kai pristatomos tam tikros prekės. Pasirašius kelis kontraktus su skirtingais klientais, prekės gali būti perkamos iš vieno ar kelių tiekėjų, tai priklauso nuo prekių pobūdžio. Kontrakto gyvavimo ciklą analizuosime atsižvelgdami į schemą(2 pav)



2 pav Kontrakto gyvavimo ciklo schema

Iš schemos matyti : kontraktą pasirašo prekybos įmonė-tarpininkas ir klientas. Kontraktą sudaro duomenys : periodas ir suma. Periodas – laikotarpis skirtas įgyvendint kontakto sąlygoms. Suma – bendra kontrakto suma už prekes, kurių kaina yra pardavimo kaina pirkėjui. Kontraktą sudaro, šiuo atveju, keturios prekių grupės A,B,C,D su tam tikais egzemplioriais ir jų kiekiais. Šioms prekėms pagaminti yra pateikiamas užsakymas tiekėjams, pagal jų gaminamų prekių rūšį. Pagaminamos prekės yra pristatomos pirkėjui ir sužymimos kontrakto sąlygoje, kaip jau pristatytos. Jeigu visos prekės yra jau sužymėtos ir kontrakto data yra toleruojama pagal kontrakto sąlygas reiškia kontrakto sąlygos išpildytos. Jeigu prekės nėra visos sužymėtos,tada jos yra užsakomos ir vėl pristatomos pirkėjui. Prekės sužymimos ,kaip jau pristatytos tik tokiu atveju, jeigu už jas apmoka pirkėjas, arba bent atsiunčia pasirašytą sąskaitą.

2.3 Uždavinio sąlygos

2.3.1 Uždavinio sąlygų keliami reikalavimai

Visos prekės, kurios bus įtraukiamos į kontraktą bus pasirenkamos iš katalogo. Prekių katalogo struktūrą atvaizduoja schema (3 pav.)

Pagal organizacijos aprašą išskirsime duomenų aibes. Prekių katalogo struktūrą aprašysime aibe $P=\{P_1,P_2,P_3,P_4,P_5,P_6\}$, kur

$\mathbf{P_1=<A_1,...,A_N>}$, prekių kodų aibė ,kurios elementų skaičius yra N – įrašų kiekis kataloge ,

$\mathbf{P_2=<B_1,\ldots, B_N>}$, prekių pavadinimų aibė, kiekviena prekė(produktas) turi savo pavadinimą kuris nėra unikali

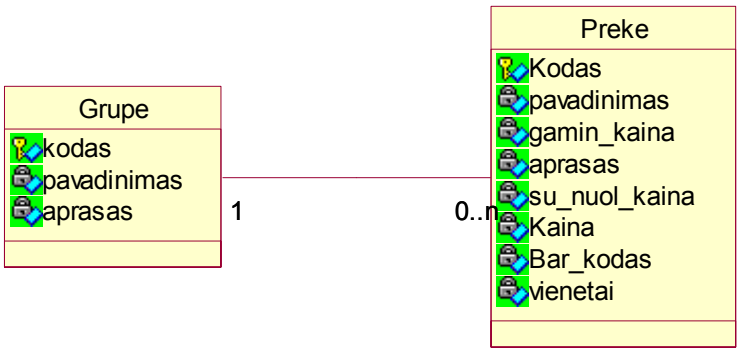
$\mathbf{P_3=<C_1,\ldots,C_N>}$, gamintojo prekių kainą aibė,.

$\mathbf{P_4=<D_1,\ldots,D_N>}$, prekių aprašų aibė.

$\mathbf{P_5=<E_1,\ldots,E_N>}$, pačios organizacijos nustatytų kainų aibė.

$\mathbf{P_6=<F_1,\ldots,F_N>}$, prekių kainų su nuolaida aibė.

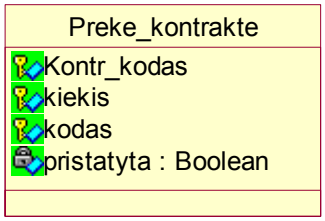
$\mathbf{P_6<F_i>=P_5<E_i>-k}$, teisinga lygybė, jeigu i-ajam $\mathbf{P_5}$ poaibio elementui, šiuo atveju būtų tam tikros prekės kaina, suteiksime **k** dydžio nuolaidą



3 pav. Prekių katalogo duomenų modelio schema

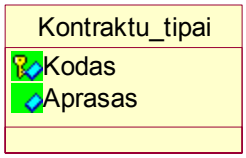
Analogiškai galima aprašyti ir esybę **PRODUCT GROUP,** R={R_1,R_2,R_3}, kur R_1 prekių grupių kodai, R_2 - pavadinimai (durys, langai, rėmai), R_3 – aprašai.

Kontrakte egzistuojančios prekės,kurios jau yra pristatytos turi būti atžymėtos, todėl esybėje, kurioje bus saugojamos prekės, kurios įtrauktos kontrakte, turi būti įvestas naujas atributas šiuo atveju būtų loginio tipo laukas **<<Pristatyta>>** (4pav)



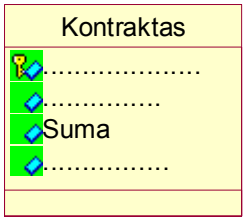
4Pav. Prekių įtrauktų į kontraktą esybė

Egzistuoja keletas kontraktų tipų. Kiekvienas tipas – susitarimas tarp organizacijos ir pirkėjo. Tarkim susitariama taip, kad ateityje kontrakto išpildymo data gali būti nukelta dėl tam tikrų sąlygų arba vienos prekės pakeistos kitomis. Šiuo atveju aktualu yra įvesti standartizuotus tarp dviejų šalių kontraktų tipus, tarkime įvedus kontraktą su tam tikru tipu IS neleis redaguoti prekių sąrašo. Aktuali esybė **<<AGREEMENT TYPE>>** (5pav)



5 Pav. Kontraktų tipų esybė

Kiekvienam kontraktui yra būtina žinoti jo bendrą sumą, kad ateityje galima būtų matyti iškilusias finansines problemas. Todėl kontrakto esybėje įvedamas atributas **<<SUMA>>** (6pav)



6Pav. Kontrakto esybė

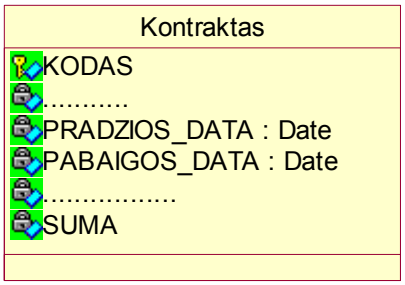
Tarkime kontraktų esybei pakanka aprašyti aibės K={K_1,K_2, R},kur

K_1=<A_1,...,A_N> , kontraktų kodų aibė ,kurios elementų skaičius yra N – įrašų kiekis kataloge **.** Aibės elementai yra unikalūs**.**

K_2=<B_1,…, B_N>, kontraktų sumų aibė, kurios elementų skaičius yra N. Aibės elementai nėra unikalūs (atributas <<Suma>>)

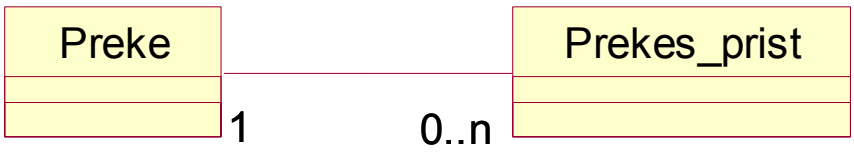
R – apsibrėžime prekių esančių kontrakte aibę, tarkime **P** – prekių katalogo aibė, tada $R \equiv P$. **R** aibės elementas.P_5 prekių kainų aibė P_5 =**<E_1,…,E_M >, M** – prekių kiekis kataloge. Tarkime turime atsitiktinį kontraktą K_i, $K_i \in K$,i={1..N}, tuomet $B_i = \sum_{i=1}^{M} Ei$, $B_i \in K_2$

Norint vesti kontraktų finansinę kontrolę, aktualu identifikuoti kontraktus ,taipogi žinoti galiojimo pradžios ir pabaigos datas. Todėl kontrakto esybėje įvedami nauji atributai <<**KODAS**> - unikalus atributas, **<<PRADŽIOS_DATA>>, <<PABAIGOS_DATA>>** - datos tipo atributai (7.Pav)



7 Pav. Kontrakto esybė

Kadangi įvade buvo minėta,kad atsikaitymas už prekes, taipogi tiekimas vyks dalimis todėl aktuali esybė, kurioje bus saugoma prekių partijos kiekiai, pristatymo datos bei sumos.



8. Pav Pristatytu prekiu esybė

Iš 8.Pav matome kad esybe <<Prekes_prist>> turi kardinalumą (1;0..n) su prekių esančių kontrakte esybė. Šioje esybėje yra išskaidyta prekė kontrakte pagal kiekio atributą, kad galima būtų matyti kada ir kokie prekių kiekiai pristatyti.

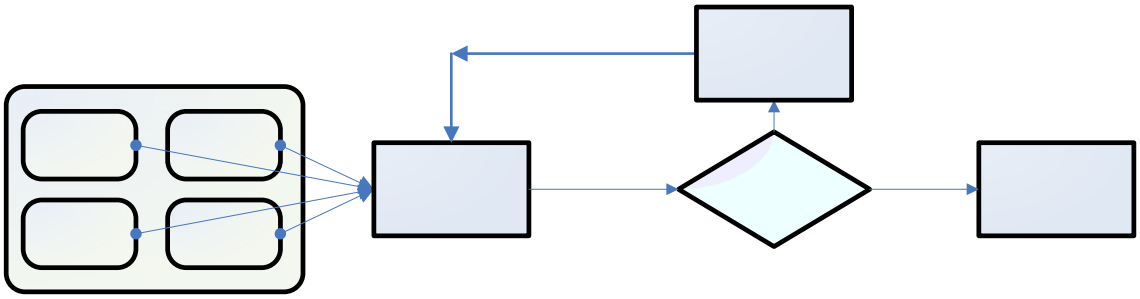
Tarkime prekės kontrakte kiekį aprašome aibe **A ,** o prekės partijų kiekį aibe **B.** Jeigu tenkinama sąlyga $B \in A$, tuomet $A=\sum_{i=1}^{m} Bi$, kaip matome i={1..m}, šiuo atveju **m** konkrečios prekės partijų kiekis.

2.3.2 Prekių identifikavimo problema

Kadangi ,kaip buvo minėta skirelyje 2.2.1 turėsime ir prekių katalogą - iškyla dar viena problema – prekių identifikavimo problema.

E.prekyboje prekėms žymėti yra įvestas standartinis kodas vadinamas BAR kodu. BAR kodas nusako prekės tipą, kilmės šalį ir gamintoją. Visuomet yra įvertinama tikimybė gauti prekes su vienodais BAR kodais. Vienos rūšies konkreti prekė neturi unikalaus BAR kodo pasaulyje. Tarkime prekybos įmonė importuoja prekes iš skirtingų

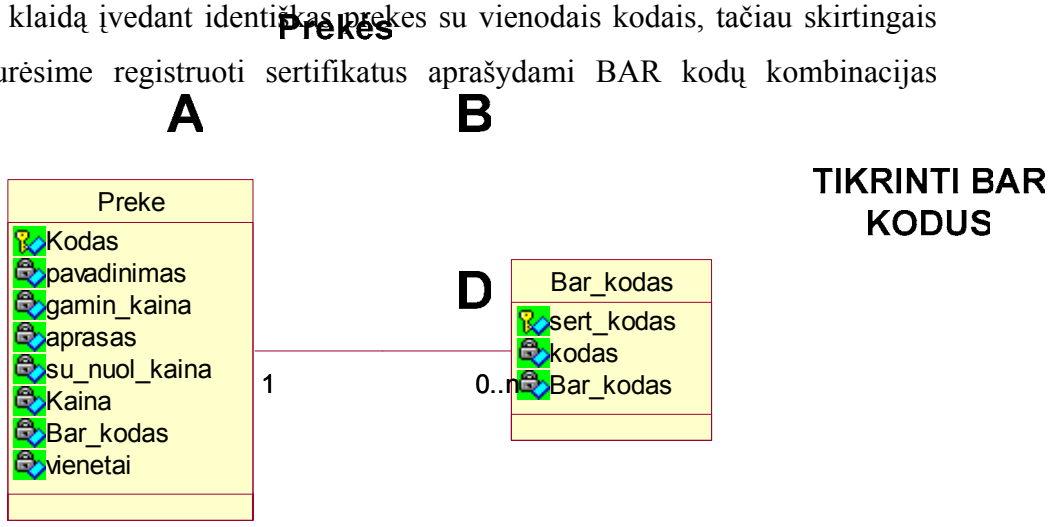
tiekėjų. Įvežamos kelios visiškai skirtingo pobūdžio prekės. Šios prekės gali turėti vienodus BAR kodus. Norint realizuoti šias prekes reikia modifikuoti BAR kodą pagal tam tikras taisykles. Realizuojant prekes per tarpininką, tarpininkui sąskaitoje-faktūroje pateikiamas BAR kodas. Jei jis sutampa su turimų prekių BAR kodais, tarpininkas tuomet pats sprendžia šią problemą. Problemos grafinė analizė pateikta (9 .Pav)



9.Pav Prekių identifikavimo ciklas

Galimas ir kitas problemos variantas. Tarkime identiškas prekes su vienoda gamybos įranga, gamina skirtingi gamintojai esantys skirtingose valstybėse. Šiuo atveju turėsime identiškas prekes su skirtingais BAR kodais.

Jeigu prekės yra gaminamos su vienoda įranga ir yra gamintojo skaitomos identiškai vienodos, tuomet jos turi turėti atitikties sertifikatą. Tokiu atveju, kad sistema netraktuotų kaip klaidą įvedant identiškas prekes su vienodais kodais, tačiau skirtingais BAR kodais, turėsime registruoti sertifikatus aprašydami BAR kodų kombinacijas (10.Pav).



10.Pav sertifiaktų esybės

Kaip jau minėta įvade organizacija atlieka tarpininko vaidmenį prekybos srityje, todėl yra įvertinama tikimybė gauti prekes su vienodais BAR kodais. Šios problemos sprendimo būdas turi būti automatizuotas.

2.3 Egzistuojančios programinės įrangos problemai spręsti analizė

Lyginamajai analizei pasirinksime keletą egzistuojančių produktų, skirtų kontraktų bei duomenų srautų valdymui. Pasirinksime tokias sistemas, kurios tenkina arba dalinai tenkina (2.2.1) pastraipoje keliamus reikalavimus. Įvesime keletą kriterijų pagal kuriuos bus atliekama lyginamoji analizė

- Prekių katalogas – galimybė gauti informaciją apie prekes
- BAR kodo kontrolė – apdoroti BAR kodus kaip minėta 2.2.2 sk
- Sutarčių su klientais / pirkėjais įvedimas – saugoti informaciją apie sudarytas sutartis
- Kontraktų finansinė kontrolė – kontroliuoti (stebėti) kiekvieno kontrakto prekių ir pinigų srautus.

Lyginimui pasirinksime Hansa Financials (Hansa World Enteprice), Pragma , Centas ir Saikas. Visi iš išvardintų paketų atlieka kai kurias išvardintas funkcijas. Detalesnę informaciją pateiksime lentelėje (11 pav.)

	Prekių katalogas	BAR kodo kontrolė	Sutartys	Kontraktų finansinė kontrolė
Hansa Financials	+	+	+	+
Pragma	+	+	-	-
Centas	+	-	-	-
Saikas	+	+	+	-

11 Pav. Programinės įrangos palyginimas

Iš lentelės matome, kad visus keliamus reikalavimus atitinka tik Hansa Financials IS. Rinktis šią sistemą yra gana problematiška – jos kaina yra didelė, programa reikalauja priežiūros, kontraktų finansinės kontrolės modelis yra tiesiogiai susietas su buhalterija, ko

nereikalauja uždavinio sąlygos. Programinį paketą sudaro virš 40 modulių. Ši programinė įranga yra anglų kalba. Plačiau – (http://www.hansaworld.com)

„Pragma‘‘ pasirinkome, kaip populiariausią buhalterinės apskaitos sistemą, „Centas‘‘ – pati pigiausia Lietuvoje parduodama buhalterinės apskaitos sistemą. Sistema „Saikas‘‘ turi galimybę saugoti informaciją apie pasirašytas sutartis su pirkėjais ir tiekėjais, tačiau kontraktų finansinės kontrolės ši sistema neturi.

2.5 Uždavinio sprendimo kriterijai

Šiame skyriuje išskirsime kriterijus, kuriuos turi tenkinti kontraktų finansinės kontrolės sistema. Norėdami lengviau suformuluoti vertinimo kriterijus, apžvelkime sistemos taikymo srities ypatybes.

Viena iš pagrindinių sąlygų būtų – informacijos pateikimas apie kontraktus reliame laike. Egzistuoja daug kontraktų, kiekvienas kontraktas turi savo sąlygas – prekės,kurios turi būti pristaytos per tam tikrą laiką. Esant reikalui sistema turi greitai suteikti duomenis apie kontraktus arba kontrakta, tarkime kokios prekės jau yra pristatytos , kokios prekės turi būti pristaytos, už kokią sumą pinigų dar teks išvežti prekių. Taipogi atžymint jau pristatytas prekes pastebėti ar teisingai už jas buvo atiskaityta. Kaip minėta (**1. Įvadas),** vienas žmogus nepajėgus greitai susidoroti su tokiu duomenų kiekių, o su šia sistema dirbti pakanka vieno žmogaus. Žinoma, visada galima nusamdyti daugiau darbo jėgos, tačiau techninio sprendimo ekonominis naudingumas bus didesnis.

Remdamiesi šiomis prielaidomis, paminėsime šiuos mus tenkinančios sistemos kriterijus:

1) Sistema turi būti *automatizuota* ir *autonomiška* – atlikti kontraktų apdorojimą reliama laike, duoti pastabas apie artėjantį kontrakto pasibaigimo laiką . Visa tai turi įvykti su kuo mažesne žmogaus intervencija.
2) Galimybė *apdoroti* IS esančią informaciją
3) Sistema turi būti *ekonomiškai naudinga* – ji darbą privalo atlikti efektyviau nei vienas ar keli darbuotojai, o jos eksploatacija ilgalaikiu požiūriu turi kainuoti mažiau.

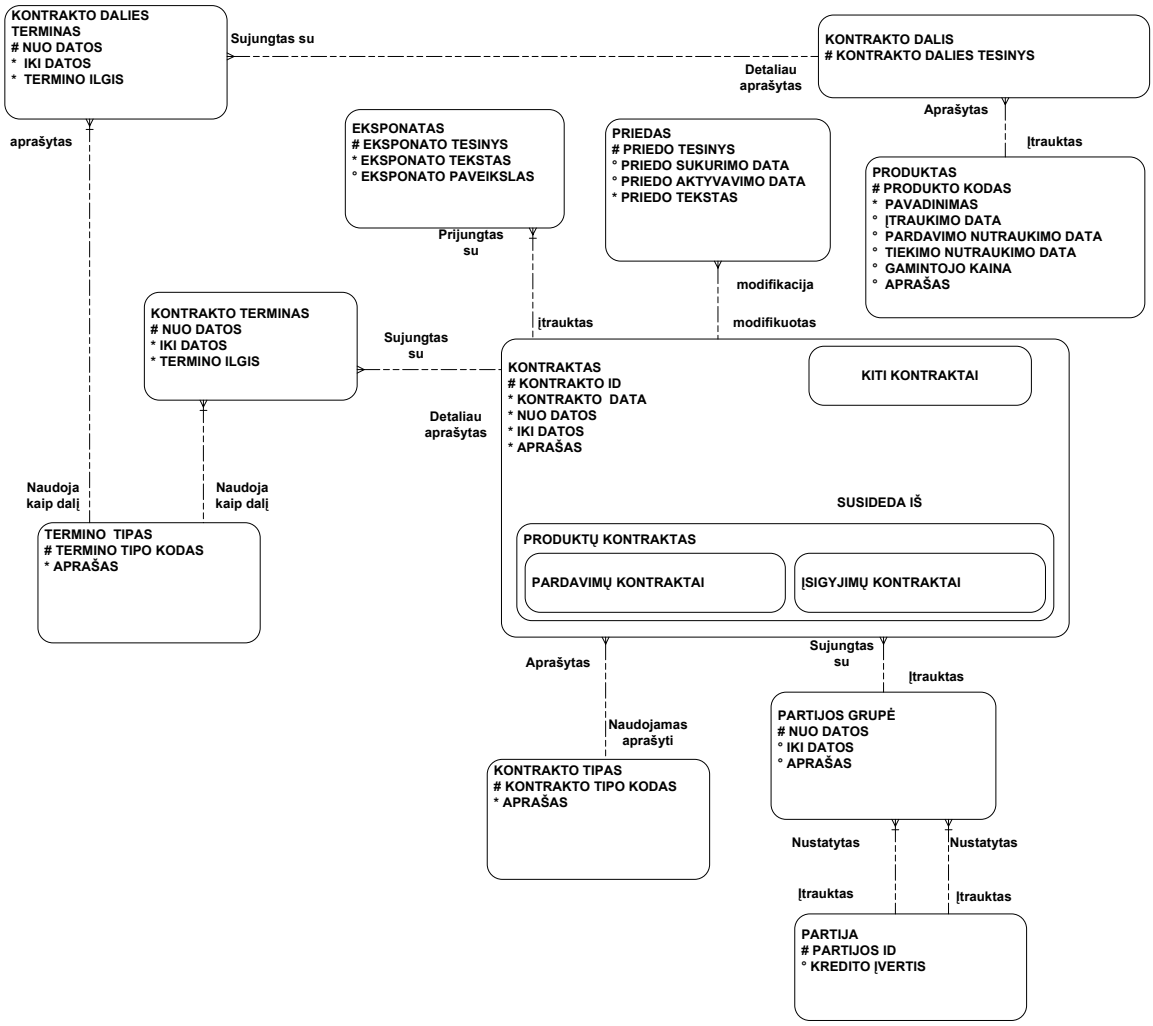
4) Sistema turi būti *lanksti* – iškilus naujiems funkciniams reikalavimams, sistema turi būti patobulinta reikalaujant kuo mažesnių resursų. Taipogi esant reikalui sujungta su kita IS
5) Sistema turi būti *saugi* – duomenys turi būti apsaugoti nuo nutekėjimo pašaliniams asmenims.

Patenkinus šiuos kriterijus laikysime, kad sistema atlieka savo darbą pagal iškeltas sąlygas nepriekaištingai.

2.5 Egzistuojantys sistemų modeliai ir sistemos tinkančios daliniam uždavinio sprendimui

2.5.1 Tipinis kontraktų modelis prekybai

Organizacijos, sudarinėjančios kontraktus , duomenų modelį galime konkretizuoti žemiau pateikta schema (12 pav):



12 pav. Kontraktų sudarymo duomenų modelis

Aprašysime mūsų nagrinėjamam atvejui aktualias esybes :

PRODUKTAS – aprašomas produktas, šiuo atveju būtų prekės. Kiekviena prekė turi savo unikalų kodą , pavadinimą ir aprašą

TERMINO TIPAS – kontrakto laikotarpio tipas, kurį kiekviena organizacija nusistato individualiai pagal savo veiklos tipą ir susitarimą su klientais dėl kelių kontraktų tipo. Taip pat ši esybė saugo aprašą apie kiekvieną tipą.

KONTRAKTO TIPAS – kontrakto tipas, kaip ir anksčiau aprašytame TERM TYPE, galioja tos pačios taisyklės. Kiekviena organizacija individualiai susikuria savo kontraktų tipus ir taisykles galiojančias kiekvienam tipui. Gali būti aibė ir tokių organizacijų kurie naudoja standartizuotus kontraktų tipus.

KONTRAKTAS – aprašomi kontraktai- pirkimas, pardavimas, kontraktų ID, kiekvieno kontrakto aprašas. Per esybę **KONTRAKTO DALIS** jungiasi su produktų **PRODUKTAI** esybę

PRODUKTAI - aprašomi produktai, turintys unikalų savo kodą ,pavadinimą,aprašą , įtraukimo ir išėmimo iš produkcijos datas, bei gamintojo nustatytas kainas.

Šis universalus duomenų modelis yra siūlomas kaip pagrindas projektuojant duomenų bazę programinei įrangai skirtai administruoti sutartis ir kontraktus. Modelis gali būti modifikuojamas prijungiant arba eliminuojant esybes bei atributus. Šis modelis yra siūlomas **Len Silverston** išleistoje knygoje **The Data Model Resource Book: A Library of Logical Data and Data Warehouse Designs**

2.5.2 Verslo valdymo sistemos

Verslo valdymo sistemos (VVS) – Šių sistemų tikslas yra padėti mažoms,vidutinio dydžio ir didelėms organizacijoms automatizuoti procesus, priimti pelningesnius sprendimus bei spartinti augimą Populiariausios pasaulyje sistemos būtų : Microsoft buisness solutions, Scala, Axapta, Microsoft Navision. Microsoft buisness solutions sistema apjungta iš sistemų Microsoft Navission ir Axapta. Toliau

paanalizuosime ir aprašysime populiariausią sistemą Microsoft buisness solutions. Sistema aprėpia šiuos mums aktualius pagal uždavinio sąlygas, modulius

- Verslo portalas – klientams prieigos suteikimas prie tam tikrų duomenų bei taikomųjų programų
- Ryšių su klientais valdymas (CRM) – klientų grupių valdymas, marketingo kampanijų kūrimas, pardavimų organizavimas.
- El.Komercija – Klientų arba tiekėjų sistemos prijungimas prie savosios
- Projektų valdymas – sutarčių sudarymo organizavimas
- Tiekimo grandinės valdymas – Sandėlių organizavimas, užsakymų valdymas, poreikių su klientais planavimas.

Kiekvienas modulis gali būti koreguojamas pagal organizacijos keliamus funkcinius reikalavimus. Taipogi mūsų nagrinėjamu atveju galime koreguoti modulius **Projektų valdymas** ir **Tiekimo grandinės valdymas** pagal savus funkcinius reikalavimus aptartus 2.2.1 sk.

2.5.3 Egzistuojančių modelių ir sistemų tinkančių daliniam uždavinio sprendimui lyginamoji analizė

Aptarėme du metodus, kuriais remdamiesi galime projektuoti ir įgyvendinti IS. Šioje pastraipoje šiuos metodus palyginsime ir parinksime optimalų variantą pagal uždavinio keliamus reikalavimus. Išskirsime keletą lyginimo kriterijų, pagal kuriuos turi būti suprojektuota sistema ir koks turi būti gautas rezultatas :

1) Lankstumas – kuriant sistemą taip pat sukurtai sistemai, iškilus arba pasikeitus funkciniams reikalavimams pasikeitimai turi būti įgyvendinti sunaudojant kuo mažiau resursų
2) Vartotojo sąsajos paprastumas – sistema turi būti kuo aiškesnė iš vartotojo pusės. Sistemą turi būti adaptuojama dideliam vartotojų spektrui

3) Kaina – sistema turi būti sukurta panaudojant optimalias projektavimo ir realizavimo priemones, atsižvelgiant į kainos ir galimybių santykį
4) Saugumas – sistemos duomenys turi būti nepasiekiami nepageidautinų asmenų
5) Adaptyvumas – sistema turi būti prijungiama prie kitų sistemų. Įgyvendinimas turi būti atliktas reikalaujant kuo mažesnių resursų.

Išskirsime lyginamųjų subjektų pavadinimus:

- Sistemos realizavimas panaudojant tipinį kontraktų modelį prekybai, pagal Len Silverston, kaip DB pagrindą
- Sistemos realizavimas jau egzistuojančiose VVS sistemose, atlikus individualius pakeitimus

Palyginimus matome (13 pav)

	Realizavimas su tipiniu kontraktų modeliu	**Realizavimas su VVS**
Lankstumas	●	●
Vartotojo sąsaja	●	○
Kaina	●	○
Saugumas	●	●
Adaptyvumas	●	●*

● - yra, ○ – nėra

* - gamintojo numatytas.

13 pav. Metodų palyginimas

Iš lentelės matyti kad realizavimas su VVS neturi tik vartotojo sąsajos ir kainos teigiamo įvertinimo kriterijaus. VVS daugumos vartotojų sąsajos būna sudėtingos , reikalaujančios papildomų mokymų jas adaptuotis paprastiems vartotojams. VVS kainos būna labai didelės, net patys gamintojai nesiūlo jų įsigyti mažoms įmonėms. Tarkime Microsoft savo produktą **Business Solutions** siūlo įmonėms Lietuvoje kurių metinė apyvarta 2,5-20 milijonų litų

Atlikę lyginamąją analizę priemonių tinkamų uždavinio sprendimui, sistemą projektuosime ir įgyvendinsime naudodami kaip DB pagrindą tipinį kontraktų modelį prekybai.

Toliau apatarsime projektavimo eigą bei metodus, modifikuosime kontraktų modelį pagal savo uždavinio keliamus reikalavimus.

2.6 Sistemos projektavimo metodo ir priemonių parinkimas

Sistema gali būti projektuojama dviem metodais : struktūriniu ir objektiniu (naudojant UML). Šiuo metu labiau paplitęs yra objektinis projektavimo metodas, nes jis turi vieningą terminologiją ir leidžia atvaizduoti kuriamą sistemą grafiškai UML diagramomis. Naudojant UML sistema gali būti specifikuota (UML turi standartus), vizualizuota ir dokumentuota. UML privalumus pateiksime žemiau :

- Supaprastėja komunikacija, visi kalba ta pačia kalba ⇨ iššvaistoma mažiau laiko
- Reikalavimai lengviau apibrėžiami ir dokumentuojami ⇨ mažiau "pamirštų" vietų
- Vartotojai įtraukiami į programos kūrimą nuo pat pradžių ⇨ mažiau perdarymų pabaigoje
- Priemonė išsaugoti sukauptas žinias firmoje, net jei žmonės ją palieka
- Sutaupo laiko susipažįstant su jau sukurtomis sistemomis

Pagal uždavinio sąlygą projektavimui labiausiai tinka naudoti RUP (Rational Unified Process). RUP – tai programų kūrimo procesas, kuris :

- Susijęs su UML
- Komercinis
- Palaikomas įrankių
- Palengvinantis darbą komandoje
- Konfiguruojamas

- Apimantis geriausias programų kūrimo praktikas
- Objektiškai orientuotas

RUP apima šias praktikas:

- Kurti programinę įrangą iteratyvią
- Valdyti reikalavimus
- Naudoti komponentines (component-based) architektūras
- Programinę įrangą modeliuoti vizualiai (naudojant diagramas)
- Nuolat tikrinti programinės įrangos kokybę
- Kontroliuoti programinės įrangos pasikeitimus

Išanalizavę objektinį projektavimo metodą bei RUP galima teigti ,kad šie metodai yra tinkami pagal mūsų kuriamos programinės įrangos pobūdį bei tipą.

Projektuoti sistemas naudonat UML yra siūloma aibė priemonių. Vienos populiariausių yra Rational Rose ir Magic Draw. (14 lent) matome šių priemonių palyginimus

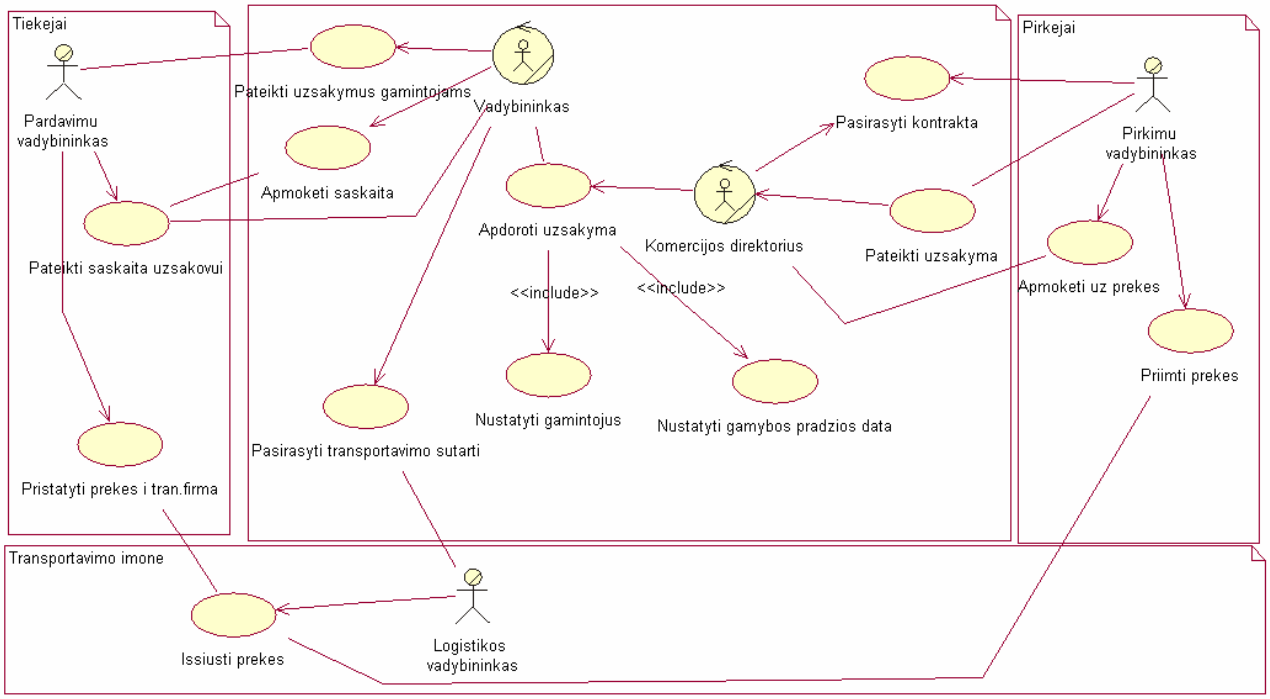
Pasirinkimo kriterijai	**Rational Rose**	**MagicDraw**
Pilnas UML(1.3 versijos) palaikymas	+	+
Diagramų suderinamumo kontrolė	+	+
Modelio navigavimas	+	+
Diagramų pasirinkimo sąrašai	+	+
Diagramų spausdinimas	+	+
Diagramų eksportavimas	+	+
Diagramos kopijavimas į laikinąją atmintį	+	-
Kodo generavimas, atvirkštinė inžinerija	+	+
Elementų pavadinimų kartojimasis diagramose	-	+
Failo suspaudimo galimybė	+	+
Diagramų išsaugojimas grafinių bylų formatu	+	+
Galimybė pasirinkti paketo grafinę vartotojo sąsają	-	+

14 Lentelė. Rational Rose ir MagicDraw paketų palyginimas

Iš lentelės (13 pav) matome, kad priemonės skirtos projektavimui naudojant UML yra daugmaž lygiavertės, tačiau yra kitas aspektas : Magic Draw yra sukurtas naudojant JAVA platforma, dėl to reikalauja didelių techninių resursų. Turint nors ir pakankamų parametrų techninę įrangą,kurios reikalauja Magic Draw, vistiek Magic Draw atliks savo darbą ilgiau negu Rational Rose. Atisžvelgdami į šį aspektą rinksimės Rational firmos produktą Rational Rose.

2.7 Organizacijos veiklos analizė

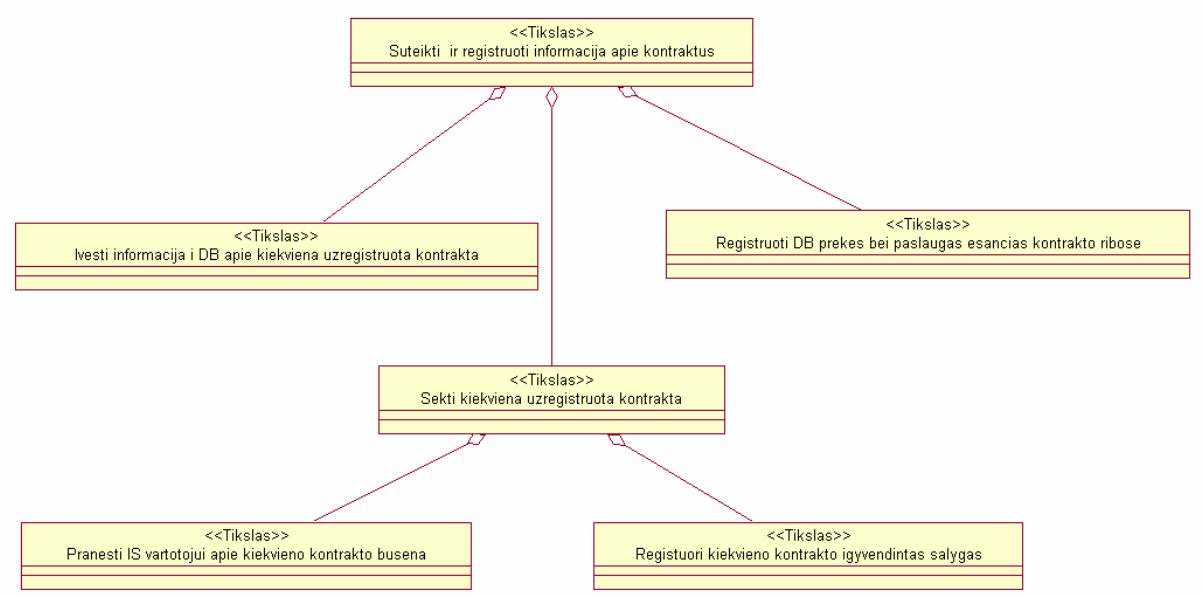
2.7.1 Veiklos sąveikų modelis



15.Pav Veiklos sąveikų modelis

Veiklos sąveikų modelis (15.Pav)atvaizduoja organizacijos ryšius su išorinių organizacijų aktoriais ir panaudojimo atvejais. Šis modelis yra naudingas sistemos projektuotojui detaliau suprasti organizacijos vidų ir išorę, tačiau šis modelis neaprašo projektuojamos sistemos funkcijų.

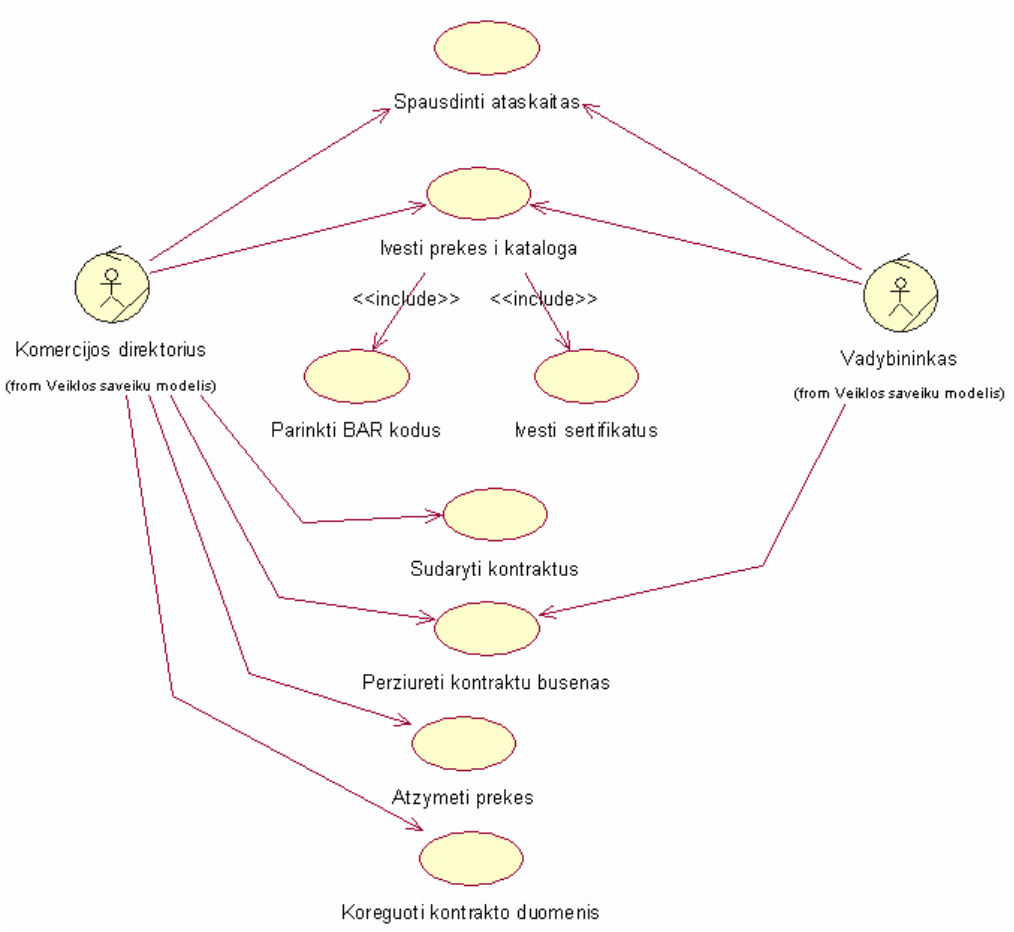
2.7.2 Veiklos tikslų modelis



(16 Pav.) Tikslų modelis

Tikslų modelis (16.Pav) atvaizduoja funkcinius sistemos reikalavimus. Kiekvienas tikslas turi potikslius. Sistema laikysime realizuota tik tokiu atveju, jei visi tikslai su savo potiksliais bus įgyvendinti. Sistemos programuotojas turėdamas tikslų modelį gali įsivaizduoti kokio tipo sistema bus kuriama ir kokias funkcijas atliks, kuo labiau detalizuotas modelis – savaime aišku tuo lengviau suprast sistemos keliamus funkcinius reikalavimus.

2.7.3 Veiklos panaudojimo atvejų modelis

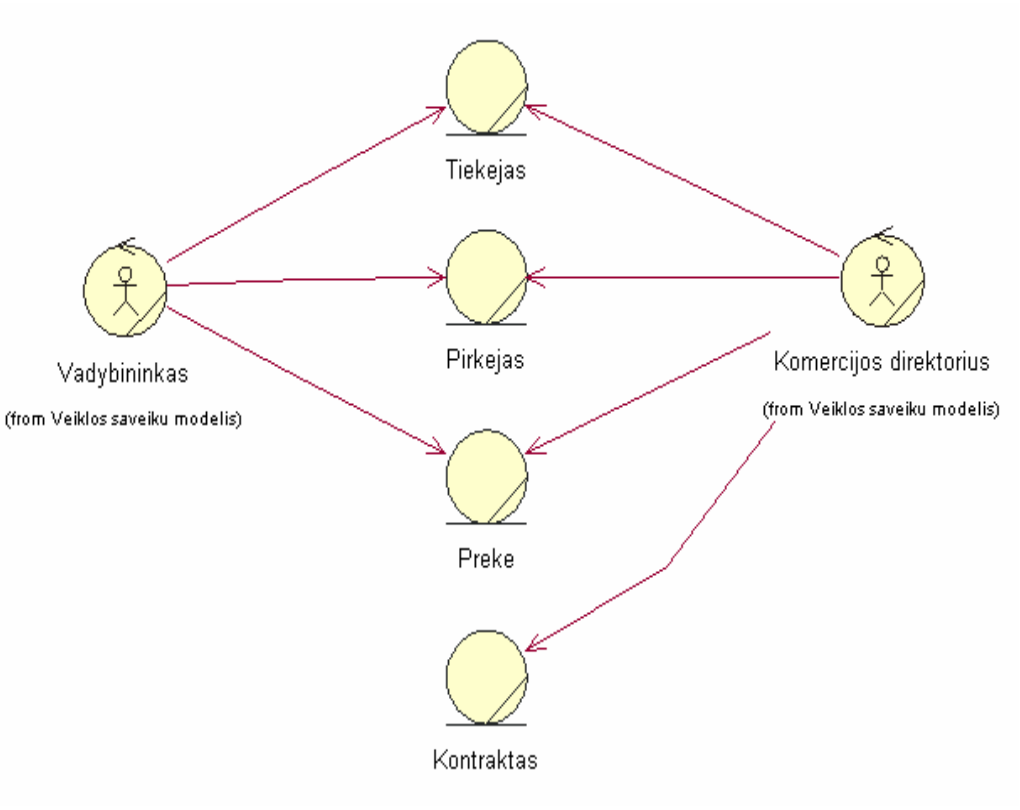


17.Pav Veiklos panaudojimo atvejų modelis

Veiklos panaudojimo atvejų modelis (17.Pav) atvaizduoja Koki veiklą,kuri turi būti automatizuota, organizacijoje atlieka skirtingi būsimos sistemos vartotojai.

Šis modelis yra aktualus realizuotojui identifikuojant vartotojų tipus, taip pat suteikia galimybę žinoti į kokias veiklas turės teisę vartotojų grupės, kokias veiklas galės atlikti skirtingi vartotojai vienodomis teisėmis. Vėliau projektuojant sistemą šio modelio pagrindu yra kuriamas detalesnis vartotojų panaudojimo atvejų modelis 3.3.1sk

2.7.4 Veiklos objektų modelis



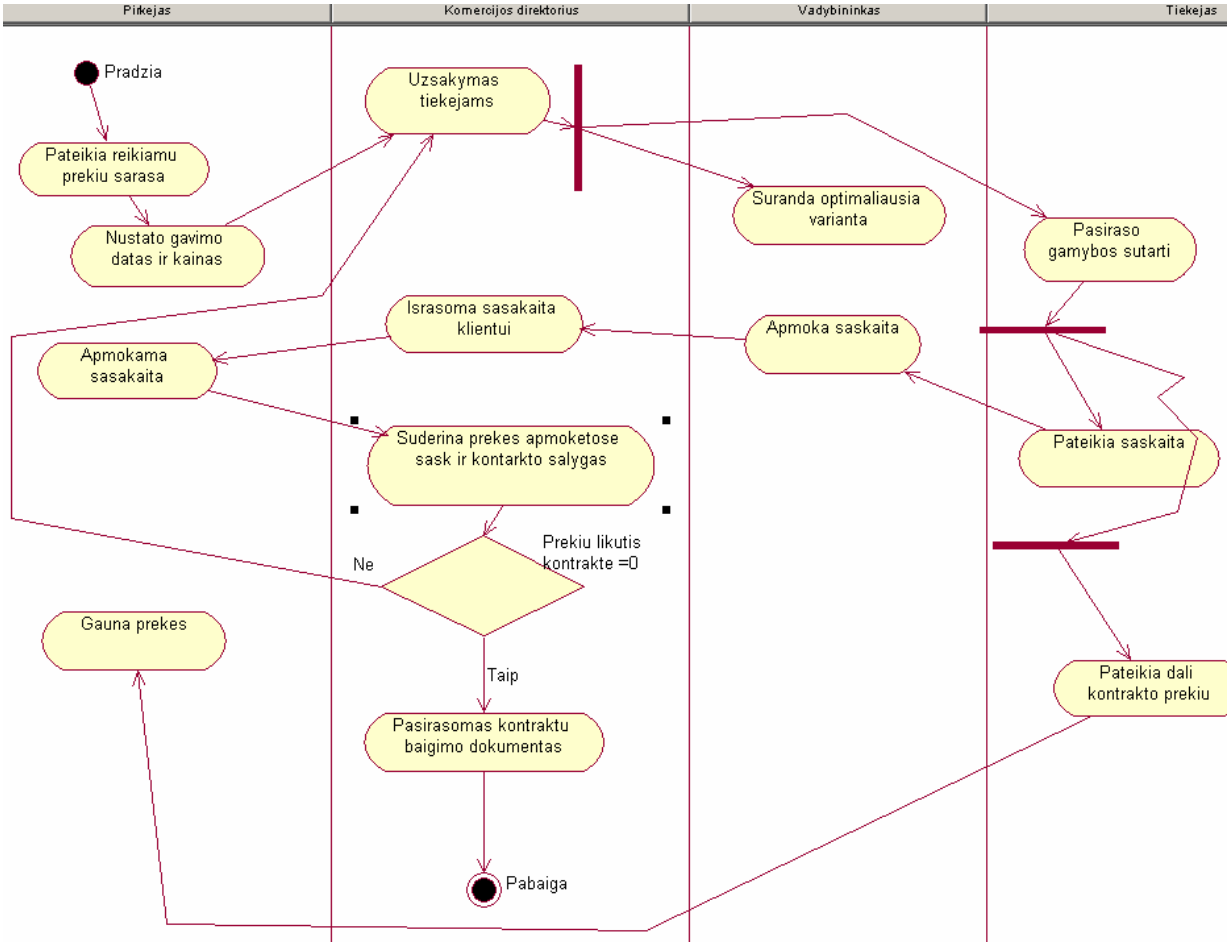
18.Pav Veiklos objektų modelis

Veiklos objektų modelis (18.Pav) atvaizduoja su kokiais verslo objektais susiduria būsimo sistemos vartotojai.Šis modelis aktualus projektuojant programines klases bei duomenų bazės modelį – kiekvienas objektas gali būti aprašytas esybe arba programine

2.7.5 Kontrakto sudarymo proceso diagrama

Procesų diagramos atvaizduoja organizacijos procesus, kurie turi būti automatizuoti. Kiekvienas procesas turi savo pradžią ir pabaigą, sąlygos operatorius ir veiklumus **(activity).** Procesų diagrama taipogi atvaizduoja kokią proceso dalį atlieka tam tikras aktorius. Galime teikti kad procesų diagrama yra detalizacija panaudojimo atvejams (**use case**) iš panaudojimų atvejų diagramos

Pateiksime pagrindinio proceso : kontrakto sudarymo, diagramą (19 pav.):



19 Pav. Kontrakto sudarymo diagrama

2.8 Analizės išvados

- Atlikta egzistuojančios programinės įrangos problemai spręsti analizė. Atlikus šią analizę nuspręsta kurti savo programinę įrangą - nerasta programinės įrangos kurios funkcijos patenkintų visus uždavinio keliamus reikalavimus

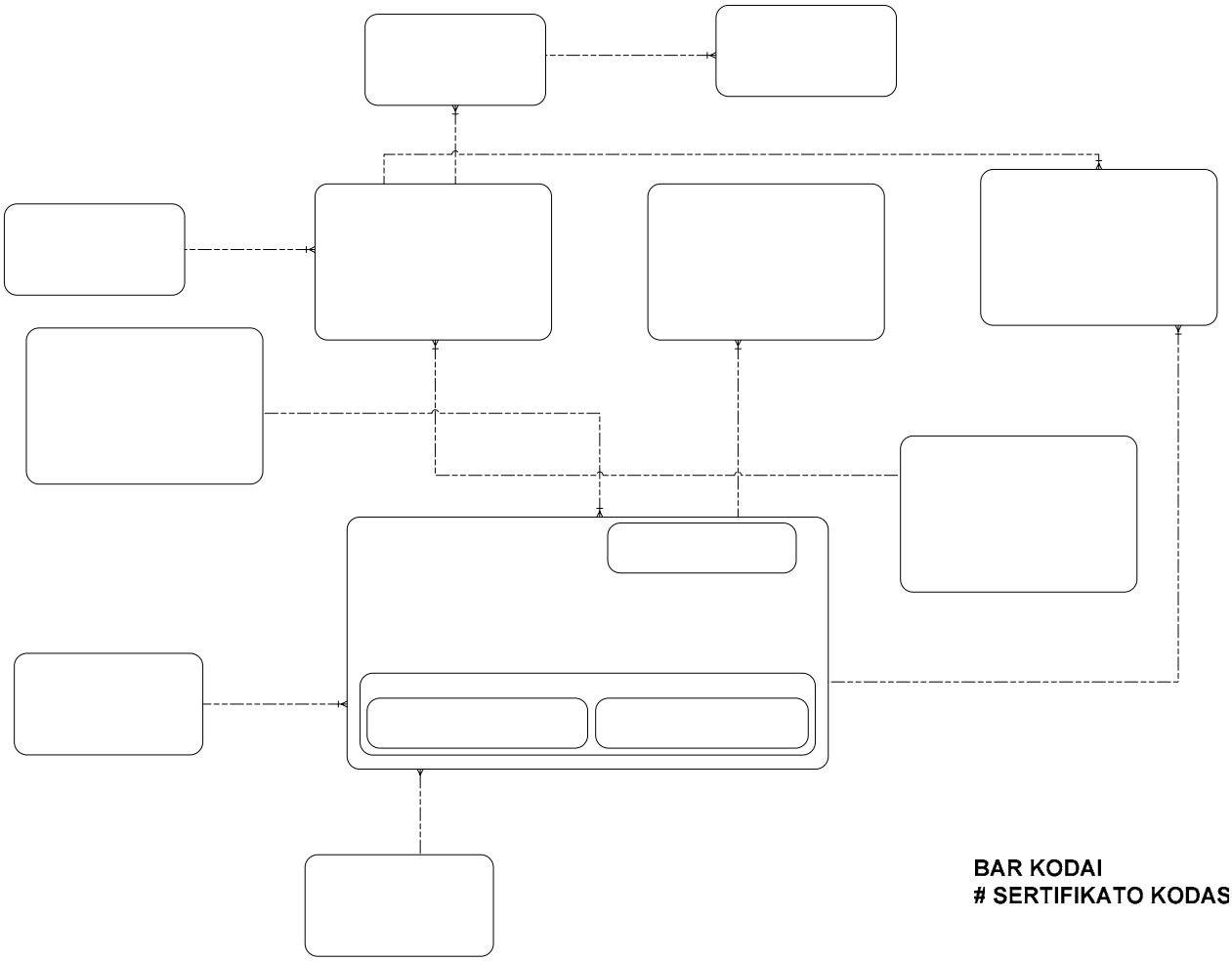
- Atlikta egzistuojančių priemonių bei modelių tinkamų daliniam uždavinio sprendimui analizė – Verslo valdymo sistemų (VVS) ir tipinio kontraktų duomenų modelio pagal Len Silverston. Buvo palygintos galimybės : modifikuoti populiarios VVS Microsoft Buisness Solutions CRM modulį arba naudoti kaip DB schemą (struktūrą) tipinį kontraktų modelį. Atsižvelgiant į VVS dideles kainas ir sudėtingumą nuspręsta naudoti tipinį kontraktų modelį ir kurti savo programinį kodą

- Atlikta projektavimo metodų bei priemonių analizė. Buvo palygintos projektavimo metodų galimybės – struktūrinio ir objektinio. Pasirinktas buvo objektinis projektavimo metodas naudojant UML ir RUP metodiką. Pasirinkus metodą buvo palygintos galimybės projektavimo priemonių – Rational Rose ir Magic Draw. Pasirinkta naudoti Rational Rose.

3. Informacijos sistemos (IS) projektavimas

3.1 Modifikuotas kontarktų modelis pagal uždavinio sąlygą

Mūsų nagrinėjamu atveju dalis esybių yra nenaudojama tipiniame kontraktų modelyje pagal **Len Silverston**, todėl modifikuojame šį duomenų modelį pagal savo keliamus reikalavimus. Kadangi kaip buvo minėta įvade,kad atsiskaitymai ir prekių pristatymai vyks dalimis , tai būtina įsivesti ir savo esybes, kuriose būtų saugojama informacija apie išvežtas prekes ir gautus pinigus už jas, esybės,kurios bus nenaudojamos, bus eliminuotos. Šis modifikuotas modelis (20 Pav.) bus naudojamas, kaip DB bazės schema (struktūra) kuriamoje IS



20 pav. Modifikuotas kontraktų sudarymo duomenų modelis

Šiame modifikuotame modelyje (14 pav.) įvedėme naują esybes **PRODUKTAS KONTRAKTE, PRODUKTŲ GRUPĖ,TIEKĖJAS,PIRKĖJAS,BAR KODAI** ir **PRISTATYTOS PREKĖS** ir naujus atributus esybėse **KONTRAKTAS** ir **PRODUKTAS :**

- **ESYBĖS**

PRODUKTAS KONTRAKTE – šioje esybėje bus saugoma informacija apie tas prekes, kurios jau yra įtrauktos į kontraktą.

PRODUKTŲ GRUPĖ – kadangi kuriama IS privalės turėti ir prekių katalogą, buvo įvesta ši esybė kuri saugos informaciją apie prekių grupes. Kiekviena grupė turės savo unikalų kodą, aprašą bei pavadinimą.

TIEKĖJAS – esybė aprašanti duomenis apie tiekėjus, pavadinimas, unikalus įmonės kodas, adresas ir t.t

PIRKĖJAS - esybė aprašanti duomenis apie pirkėjus,analogiška esybei TIEKĖJAS

BAR KODAI – esybė, kurioje bus saugomi identiškų prekių BAR kodai žr.sk. 2.2.2

PRISTATYTOS PREKĖS – kadangi prekių pristatymai vyks dalimis šioje esybėje atžymėsime prekių partijas,kurios bus pristatytos klientui

- **ATRIBUTAI**

PRODUKTAS.KAINA – nurodoma prekės kaina, kuri yra taikoma klientui

PRODUKTAS.KAINA SU NUOLAIDA – nurodoma prekės kaina su nuolaida,kuri yra taikoma klientui

PRODUKTAS KONTRAKTE.PRISTATYTA – atributas nurodantis, ar konkreti prekė jau pristatyta klientui. Šį atributą ,kiekviena organizacija, kuri prisitaikys šį modifikuotą modelį savo komercinei veiklai gali naudoti individualiai pagal savo nustatytas taisykles

PRODUKTAS KONTRAKTE.KIEKIS - atributas nusakantis prekių kiekį kontrakte

KONTRAKTAS.SUMA – tai bendra suma už prekes, kurios yra įtrauktos į kontraktą. Tarkime bazinę prekių katalogo struktūrą,prekių kurios yra kontrakte, aprašysime aibe P={P_1,P_2,P_3}

P_1=<C_1,...,C_N> , prekių kodų aibė ,kurios elementų skaičius yra N – prekių kiekis sudarytame kontrakte

P_2=<K_1,…, K_N>, kainų aibė, kiekviena prekė(produktas) turi savo kainą kuri nėra unikali

P_3=<S_1,…,S_N >, prekių kiekių aibė

Tarkime atributo KONTRAKTAS.SUMA reikšmę apibrėšime kintamuoju ***sum*** . Šiuo atveju pagal aibės **P** aprašą būtų teisinga lygybė

$$sum = \sum_{i=1}^{N} Ki * Si$$

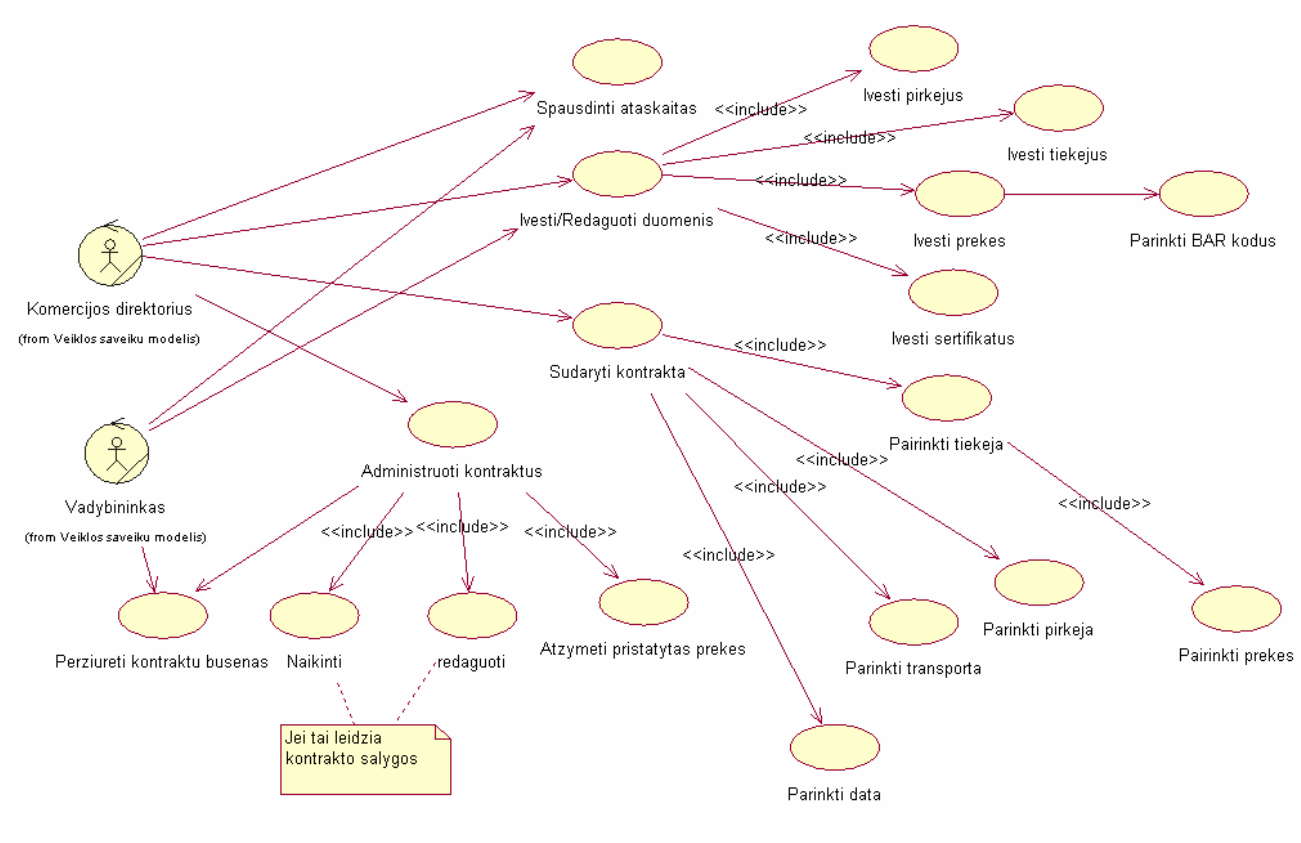
Tarkime jeigu tipinį kontraktų modelį aprašysime neapibrėžtų elementų aibe A, modifikuotą kontraktų modelį neapibrėžtų kintamųjų aibe B, tai galėsime teigti, kad aibė B yra aibės A poaibis. $B \subset A$.

3.2 Modifikuoto kontaktų duomenų modelio praktinio taikymo kriterijai

Modifikuotą kontraktų duomenų modelį galima pritaikyti organizacijose, kurios veikla yra identiška veiklai,kuri aprašyta 2.1. pastraipoje, kuriant bei projektuojant IS. Praktinio taikymo kriterijus yra tai, kad šį duomenų modelį reikia transformuoti į RDB (Reliacinė duomenų bazė) arba į ODB (Objektinė duomenų bazė). Atlikus transformaciją gausime pagrindinę duomenų bazę skirtą pagrindiniams duomenų srautams saugoti kuriamoje IS. Transformacijos atliekamos naudojant tam skirtą programinę įrangą, kuri transformuoja ER duomenų modelį į tam tikras RDB arba ODB. Gautos DB pagrindu yra kuriama programinė įranga, kuri yra unikali pagal kiekvienos organizacijos poreikius – turi savo unikalius algoritmus ir gautos DB (2pav žr.Priedai) panaudojimo metodus.

3.3 Reikalavimų modelis

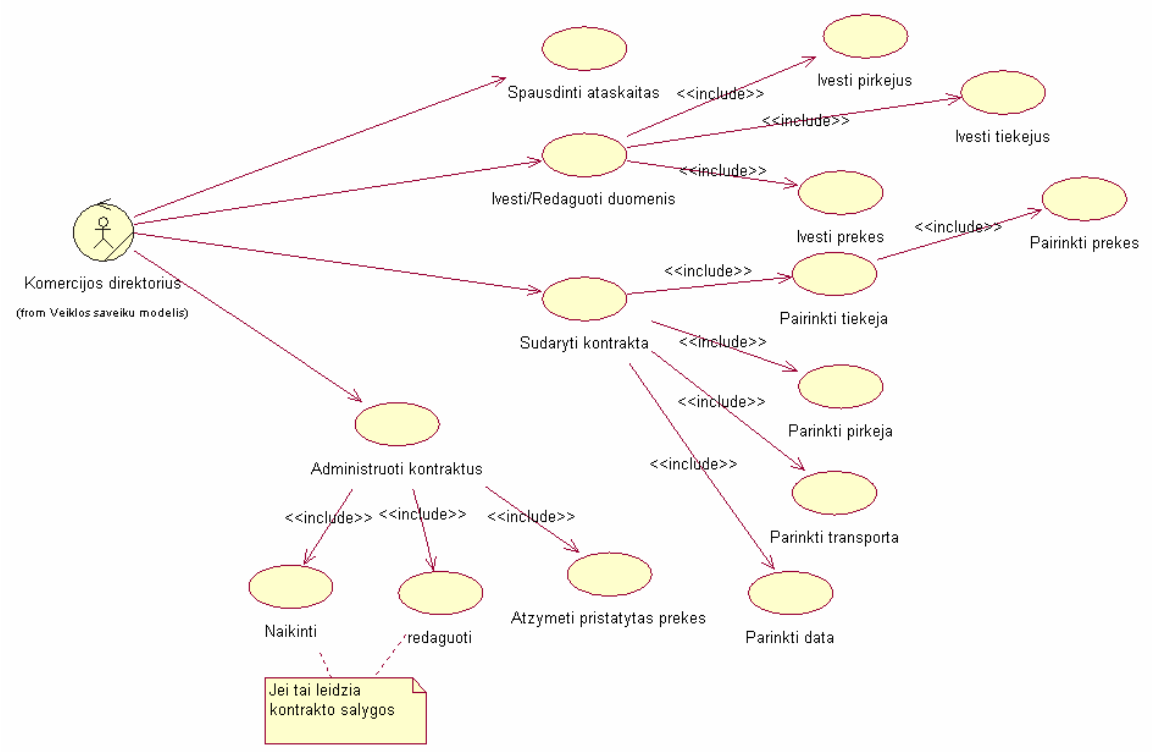
3.3.1 Vartotojų panaudojimo atvejų modelis



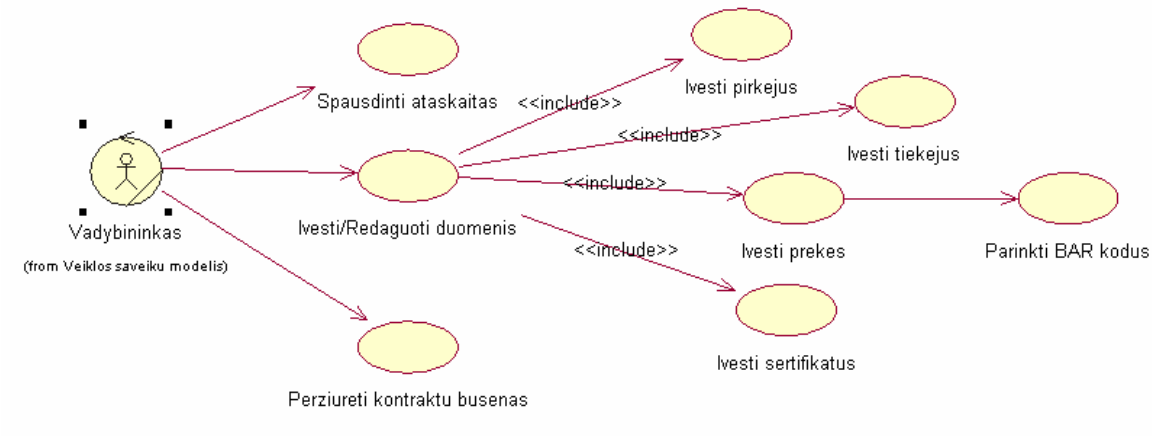
21 Pav vartotojų panaudojimo atvejų modelis (UML diagrama)

Vartotojų panaudojimo atvejų modelis (21.Pav) atvaizduoja operacijas, kurias atliks sistemos vartotojai. Galime ir kitaip suformuluoti modelio aprašą : modelis atvaizduoja sistemą vartotojų atžvilgiu. Kiekvienas atvejis (use case) gali būti detalizuotas iki reikiamo lygio. Kuo gilesnė detalizacija – tuo lengviau suprasti kiekvieno vartotojo vaidmenį kuriamoje sistemoje.

Šis modelis gali būti skaidomas į atskirus modelius, kurių skaičius būtų lygus vartotojų skaičiui bendrame modelyje. Šiuo atveju gautume du modelius – Komercijos direktoriaus panaudojimo atvejų modelį ir Vadybininko panaudojimo atvejų modelį (22,23.Pav).



22 pav.Komercijos direktoriaus panaudojimo atvejų modelis



23Pav. Vadybininko panaudojimo atvejų modelis

3.3.2 Specifikacijos panaudojimo atvejams

Panaudojimo atvejis	**Spausdinti ataskaitas**
Aktorius	Visi
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu pasirenkamas norimos peržiūrėti ataskaitos punktas. 2. Paspaudžiamas pasirinkimo patvirtinimą užtikrinantis mygtukas.	2.1. Sistema išveda peržiūrai prieš spausdinimą pasirinktą suformuotą ataskaitą iš duomenų, esančių IS duomenų bazėje.
Po sąlyga	Peržiūrėtą ataskaita gali būtį atspausdinta, paspaudus spausdinimo mygtuką.
Alternatyvos (nesėkmės atvejai)	Ataskaita gali būti nepateikta, jei vartotojo nurodytam laikotarpiui nėra informacijos duomenų bazėje.
Vykdymo variantai	Ataskaitą, kurią nori peržiūrėti arba atspausdinti, vartotojas pasirenka iš IS "Ataskaitos" meniu punkto.
Veiklos taisyklės	Vartotojas turi įvesti duomenis pagal kuriuos bus filtruojami gaunami duomenys

Panaudojimo atvejis	**Įvesti / Redaguoti duomenis**
Aktorius	Visi
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu pasirenkamas norimos įvesti/koreguoti informacijos punktas. 2. Užpildoma forma 3. Spaudžiamas patvirtinimo mygtukas	3.1. Sistema tikrina ar įvesti/koreguoti duomenys yra teisingi
Po sąlyga	Nauji duomenys įrašomi į DB ir gaunamas pranešimas apie sėkmingą transakciją
Alternatyvos (nesėkmės atvejai)	1. Įvesti duomenys gali būti nekorektiški 2. Bandomas dubliuoti unikalus duomenų tipas
Vykdymo variantai	Duomenys, kuriuos norime įvesti arba redaguoti, vartotojas pasirenka iš IS "Duomenys" meniu punkto.
Veiklos taisyklės	

Panaudojimo atvejis	**Sudaryti kontraktą**
Aktorius	Komercijos direktorius
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu atidaromas nauja kontrakto kūrimo forma 2. Parenkamas laikotarpis 3. Parenkamas kontrakto ir laikotarpio tipas 4. Parenkamas klientas 5. Parenkamos prekės ir jų kiekiai 6. Galutinėje patvirtinimo formoje spaudžiamas mygtukas patvirtinti	2.1 Sistema tikrina ar korektiškai įvestos datos 5.1 Sistema tikrina ar prekių kiekis teisingai įvestas 6.1 Sistema tikrina ar teisingai užpildyta forma
Po sąlyga	Siunčiamas patvirtinimo pranešimas, sistema įrašo duomenis į DB
Alternatyvos (nesėkmės atvejai)	2. Parinktas neteisingas laikotarpis 4. Norimo kliento nėra DB 5. Norimų prekių nėra DB 6. Galutinėje patvirtinimo formoje pakeitimai atliekami nekorektiškai
Vykdymo variantai	Duomenys, kuriuos norime įvesti arba redaguoti, vartotojas pasirenka iš IS “Kontraktas->Naujas” meniu punkto.
Veiklos taisyklės	Užpildyti visi reikiami formos laukai

Panaudojimo atvejis	**Atzymėti pristatytas prekes**
Aktorius	Komercijos direktorius
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu punkto pasirenkama kontrakto administravimo forma 2. Įvedamas kontrakto kodas 3. Įvedami prekių kodai ir kiekiai 4. Išvedama galutinė patvirtinimo forma ir spaudžiamas patvirtinimo klavišas 5. Išvedama forma su prekių likučiais ir kita informacija apie konkretų kontraktą	2.1 Sistema tikrina ar su tokiu kodu kontraktas yra 3.1 Sistema tikrina ar prekių kodai ir kiekiai yra korektiški pagal kontrakte esančias prekes 4.1 Šioje formoje galime dar daryti pakeitimus 5.1 Sistema atlieka reikalingus skaičiavimus
Po sąlyga	Nauji duomenys įrašomi į DB ir gaunamas pranešimas apie sėkmingą transakciją
Alternatyvos (nesėkmės atvejai)	2. Su tokiu kodu kontrakto nėra 3. Su tokiais kodais prekių nėra. 3.1 Nekorektiški prekių kiekiai

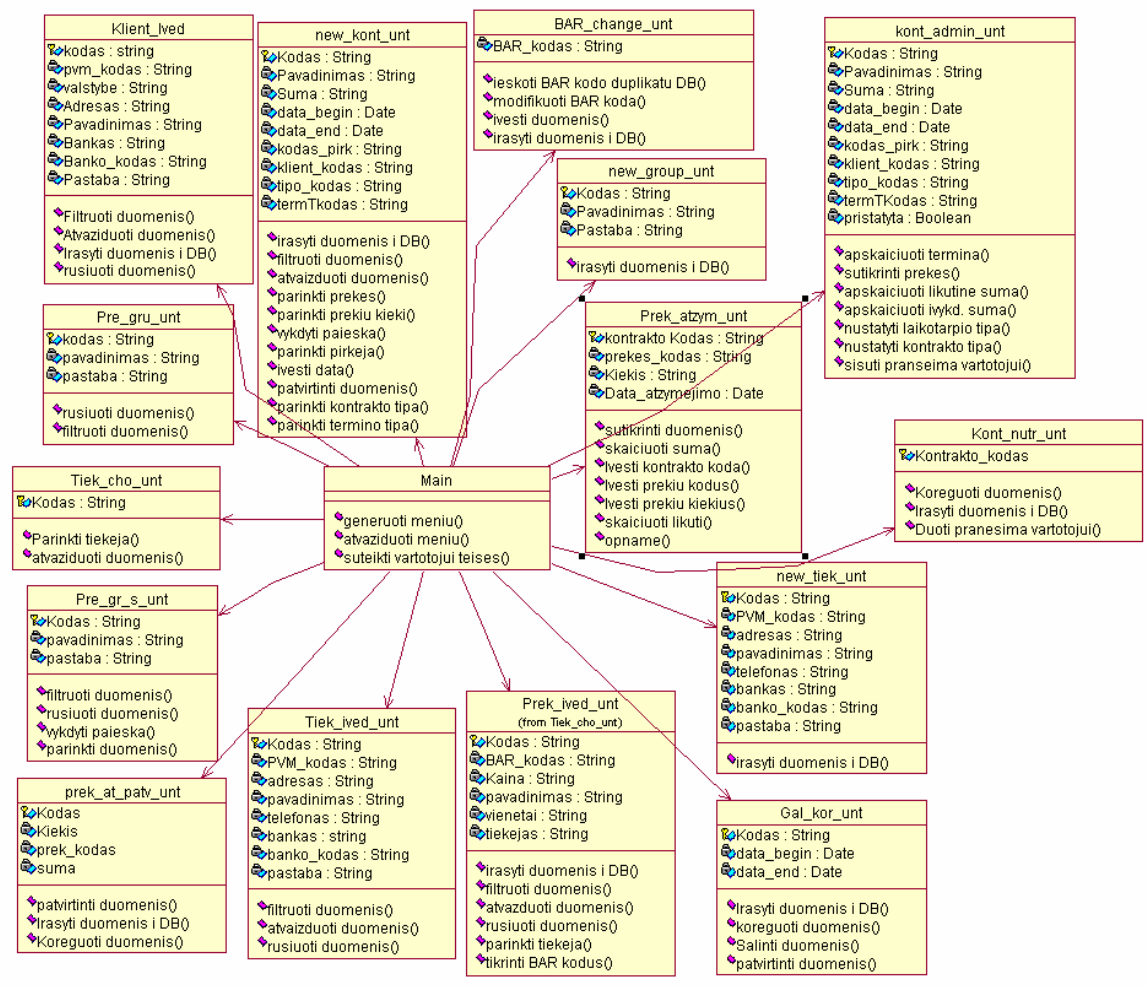
	4. Redaguoti nekorektiškai duomenys galutinėje formoje
Vykdymo variantai	Atžymėjimas prekių parenkamas iš meniu punkto administravimas->prekiu atzymejimas
Veiklos taisyklės	Pildomi visi būtini laukai, bei parenkami prekių kiekiai

Panaudojimo atvejis	**Redaguoti (kontraktą)**
Aktorius	Komercijos direktorius
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu punkto pasirenkama kontrakto redagavimo forma 2. Įvedamas kontrakto kodas 3. Įvedami/trinami prekių kodai ir kiekiai, redaguojama data 4. Išvedama galutinė patvirtinimo forma ir spaudžiamas patvirtinimo klavišas 5. Išvedama forma su prekių likučiais ir kita informacija apie konkretų kontraktą	2.1 Sistema tikrina ar su tokiu kodu kontraktas yra 2.2 Sistema tikrina ar kontrakto tipas leidžia tai daryti 3.1 Sistema tikrina ar prekių kodai ir kiekiai yra korektiški pagal kontrakte esančias prekes 4.1 Šioje formoje galime dar daryti pakeitimus 5.1 Sistema atlieka reikalingus skaičiavimus
Po sąlyga	Nauji duomenys įrašomi į DB ir gaunamas pranešimas apie sėkmingą transakciją
Alternatyvos (nesėkmės atvejai)	2. Su tokiu kodu kontrakto nėra 2.1 Kontrakto negalime koreguoti,nes tipas to neleidžia 3. Su tokiais kodais prekių nėra. 3.1 Nekorektiški prekių kiekiai 4. Redaguoti nekorektiškai duomenys galutinėje formoje
Vykdymo variantai	Redagavimas parenkamas iš meniu punkto administravimas ->kontrakto redagavimas
Veiklos taisyklės	Pildomi visi būtini laukai, bei parenkami prekių kiekiai

Panaudojimo atvejis	**Atzymėti pristatytas prekes**
Aktorius	VISI
Sistema	Kontraktų finanisinės kontrolės IS
Prieš sąlyga	Darbuotojas turi būti registruotas duomenų bazėje
Pagrindinis įvykių srautas	**Sistemos reakcija ir sprendimai**
1. Iš meniu punkto pasirenkama kontraktu būsenos forma 2. Įvedamas kontrakto kodas Kontraktą 3. Išvedami duomenys apie kontraktą	2.1 Sistema tikrina ar su tokiu kodu kontraktas yra
Po sąlyga	
Alternatyvos (nesėkmės atvejai)	2. Su tokiu kodu kontrakto nėra
Vykdymo variantai	Meniu punktas Administravimas->Kontraktu busenos
Veiklos taisyklės	Pildomi visi būtini laukai,

3.3.3 Programinių klasių diagrama

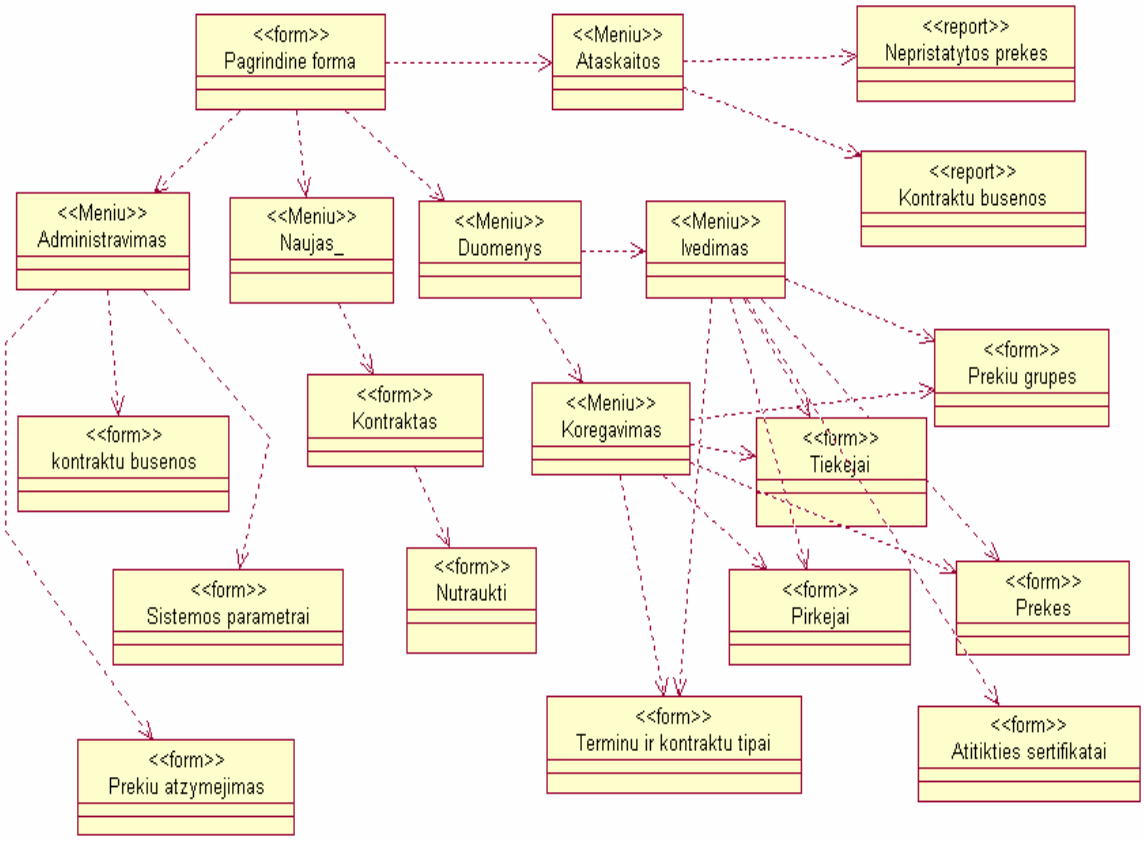
Programinių klasių diagrama nusako sąryšį tarp programinių klasių ir aprašo kiekvienos klasės metodus (operations) ir pagrindinius duomenis-kintamuosius (attributes). Programinių klasių diagrama yra naudojama kaip sistemos programinės architektūros specifikacija – tarkime projekto realizavimas yra perduodamas kitiems programuotojams, arba praėjus laikui prisireikia sistemą modifikuoti. Tokiu atveju yra matoma programinių klasių architektūra ir nustatoma kiekvienos klasės paskirtis. Mūsų projekto programinių klasių diagrama pateikta 24.Pav



24 Pav. Programinių klasių diagrama

3.3.4 Vartotojo sąsajos modelis

Vartotojo sąsaja – nefunkcinis reikalavimas. Tai įrankių valdymo paletė ,kuri atskiriems vartotojams atrodo nevienodai. Galingose sistemose yra galimybė kiekvienam vartotojui turėti savo unikalias valdymo paletes. Šiuo atveju vartotojo sąsajos modelis yra projektuojamas pagal vartotojo reikalavimus. Kuriamos sistemos vartotojo sąsajos modelis atvaizduotas (25 Pav.)

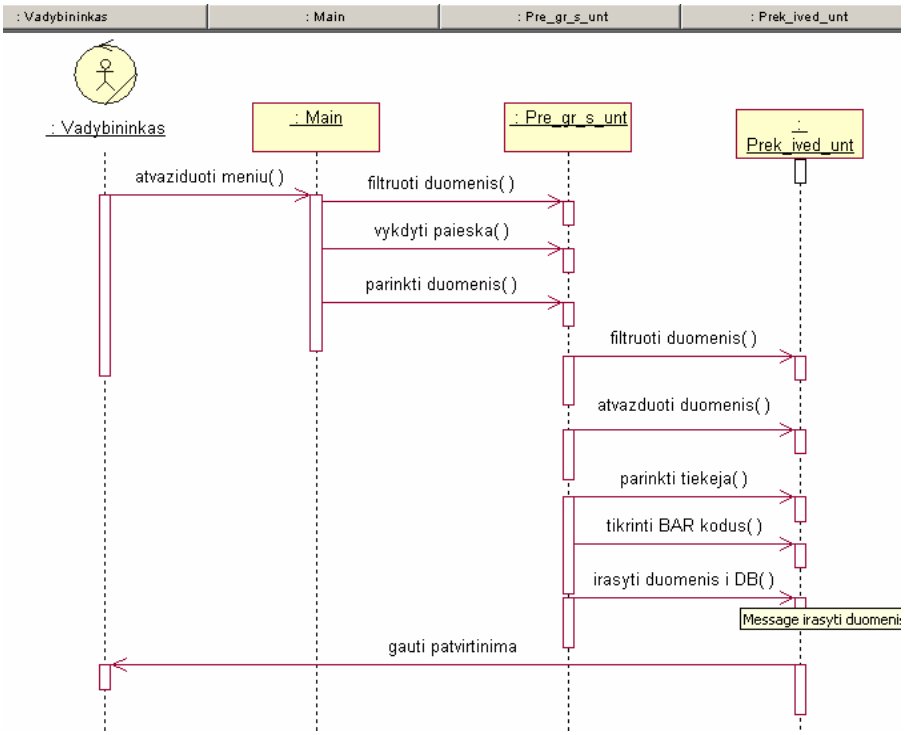


25.Pav vatotojo sąsajos modelis

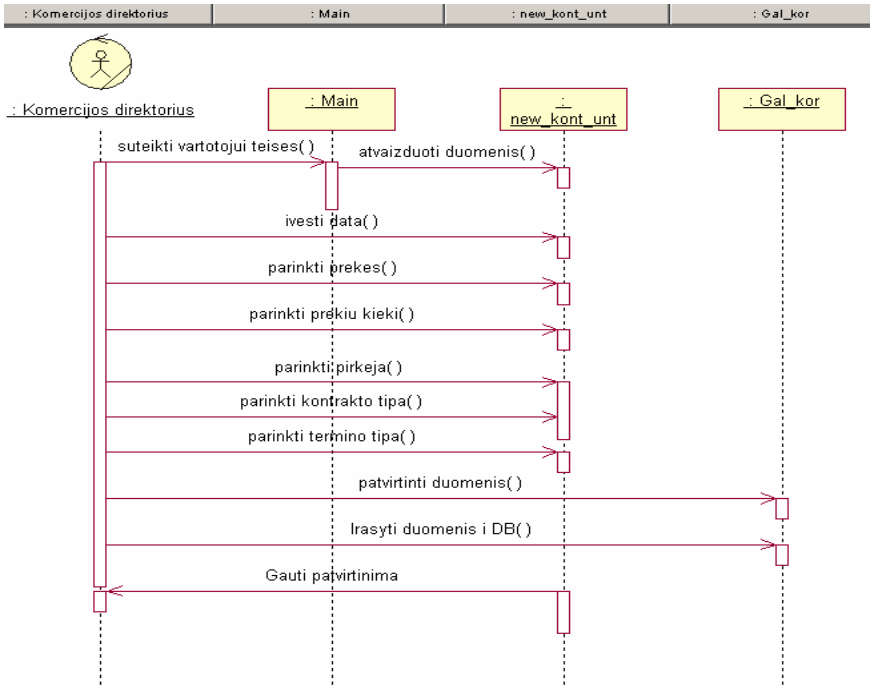
3.4 Sistemos modelis

3.4.1 Sekų diagramos

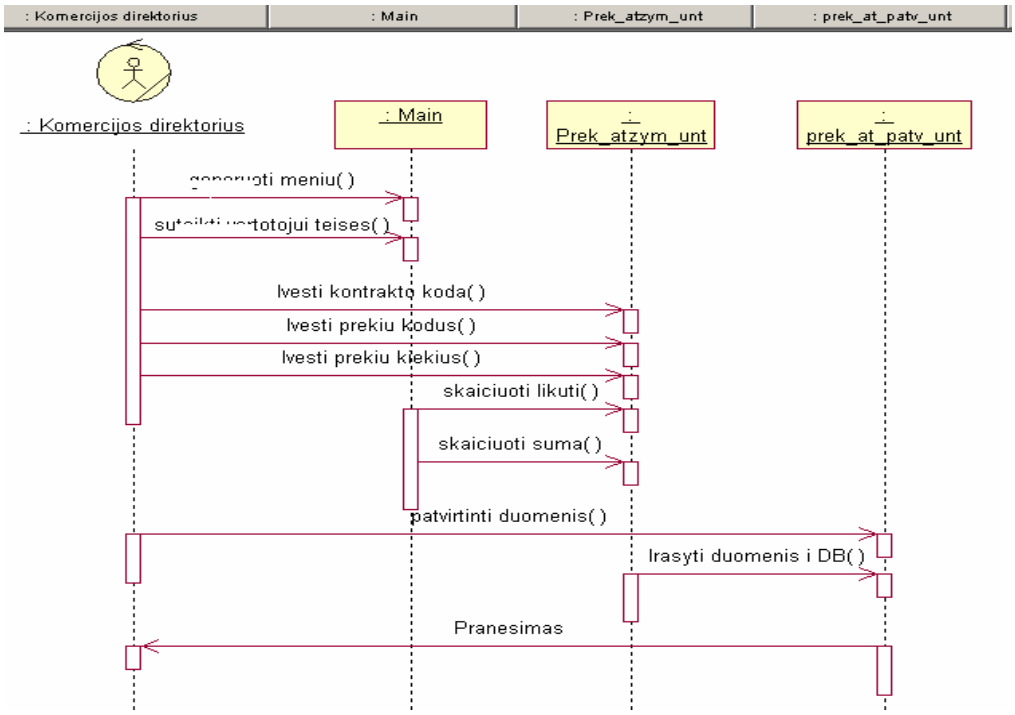
Sekų diagramos yra naudojamos specifikuoti sistemos atliekamas operacijas, tam ,kad realizuotojai ateityje programuotojas matytų kokios klasės ir kokie klasių metodai vyrauja atliekamoje operacijoje. Sekų diagramos yra kuriamos lygiagrečiai programuojant sistemą. Galime teigti, kad sekų diagramos atvaizduoja sąryšį tarp klasių metodų (operations). Pavaizduosime sekų diagramas:prekės įrašymas į katalogą, kontrakto sudarymas ir prekių atžymėjimas (26,27,28 Pav.)



26 Pav.Prekės įrašymas į katalogą



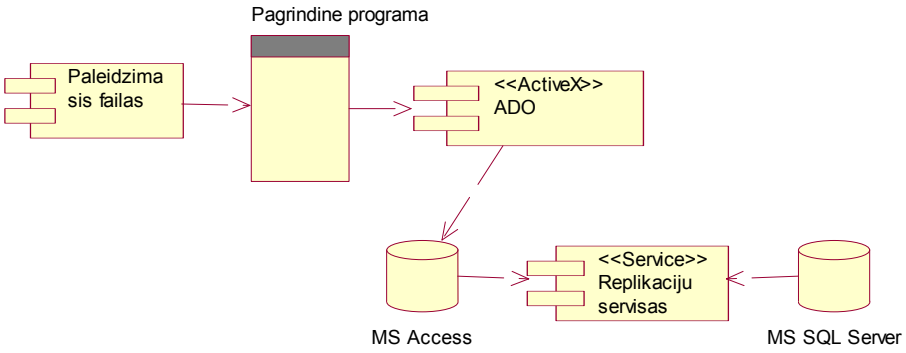
27 Pav.Naujo kontrakto sudarymas



28Pav. Prekių atžymėjimas

3.4.2 Komponentų diagrama

Komponentų diagrama atvaizduoja realizuotojui sistemą loginiame lygyje. Ši diagrama gali būti detalizuojama – kuo detalesnė, tuo aiškesnė. Komponentai tai nepriklausomos viena nuo kitos, tačiau beiškeičiančios duomenimis sistemos dalys. Komponentų diagrama pateikta 29.Pav.



29.Pav. komponentų diagrama

3.4.3 Duomenų bazės modelis

Duomenų bazės modelis naudojamas aprašyti atributų tipus – tai aktualu realizuotojui, kuris turi žinoti kokio tipo kintamuosius naudoti manipuliuojant duomenimis su programa ir duomenų baze. Detaliai aprašysime kiekvienos esybės atributus. RDB schema pateikta – Priedai 2.Pav. DB schemą importavome iš modifikuoto kontrakto modelio, tačiau pakeitėme kai kuriuos esybių pavadinimus dėl realizacijos patogumų.

Pre_gru – informacija apie prekių grupes

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Prekių grupės kodas
Pavadinimas	Text	50	Prekių grupės pavadinimas
Pastabos	Text	50	Pastabos

Prekes – saugo informaciją apie konkrečią prekę

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Prekės kodas
Bar_kodas	Text	50	Prekės BAR kodas
Pavadinimas	Text	50	Prekės pavadinimas
Vienetai	Text	50	Prekės matavimo vienetai
Kaina	Number	4byte	Prekės kaina
Tiekejas	Text	50	Prekės tiekėjas
Kodas_grup	Text	50	Prekės kodas

Tiekejai – saugo informaciją apie tiekėjus

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Tiekėjo kodas
PVM_Kodas	Text	50	Tiekėjo PVM kodas
Adresas	Text	50	Tiekėjo adresas
Pavdinimas	Text	50	Pilnas tiekėjo pavadinimas
Telefonas	Text	50	Tiekėjo telefonas
Bankas	Text	50	Banko pavadinimas
Banko_kodas	Text	50	Tiekėjo banko sąskaitos numeris
Pastabos	Memo	256K	Pastabos

Pirkėjai – saugo informaciją apie klientus

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Pirkėjo kodas
PVM_kodas	Text	50	Pirkėjo kodas
Valstybe	Text	50	Valstybė, kurioje randasi pirkėjas
Adresas	Text	50	Pirkėjo adresas
Bankas	Text	50	Pirkėjo banko pavadinimas
Banko_kod	Text	50	Pirkėjo banko kodas
Pastaba	Memo	256K	Pastabos
Pavadinimas	Text	50	Pirkėjo pavadinimas
Telefonas	Text	50	Pirkėjo telefonas

Prekes_temp – saugo informaciją apie konkrečią prekę

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Prekės kodas
Kont_kodas	Text	50	Kontrakto kodas
Pristatyta	Yes/No	1bit	Loginio tipo laukas, nurodantis ar pristatyta prekė
Kiekis	Number	4byte	Prekių kiekis kontrakte

Termino_tipas – saugo informaciją apie visus galimus termino tipus

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Termino tipo kodas
Aprašas	Text	50	Termino aprašas

Kont_tipas – saugo informaciją apie visus galimus kontraktų tipus

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Kontrakto tipo kodas
Aprašas	Text	50	Kontrakto tipo aprašas

Prekes_prist – saugo informaciją apie pristatytas prekes

Pavadinimas	Tipas	Dydis	Reikšmė
Kont_kodas	Number	4Byte	Kontrakto kodas
Kodas	Text	50	Prekės kodas
Suma	Text	50	Pristatytos prekės už sumą
Data	Date		Prekių pristatymo data

BAR_kodai – saugo informaciją apie identiškų prekių BAR kodus

Pavadinimas	Tipas	Dydis	Reikšmė
Sert_kodas	Text	50	Sertifikato kodas
BAR_kodas	Text	50	Bar kodas

BAR_kodai_temp – tarpinė sertifikatų lentelė

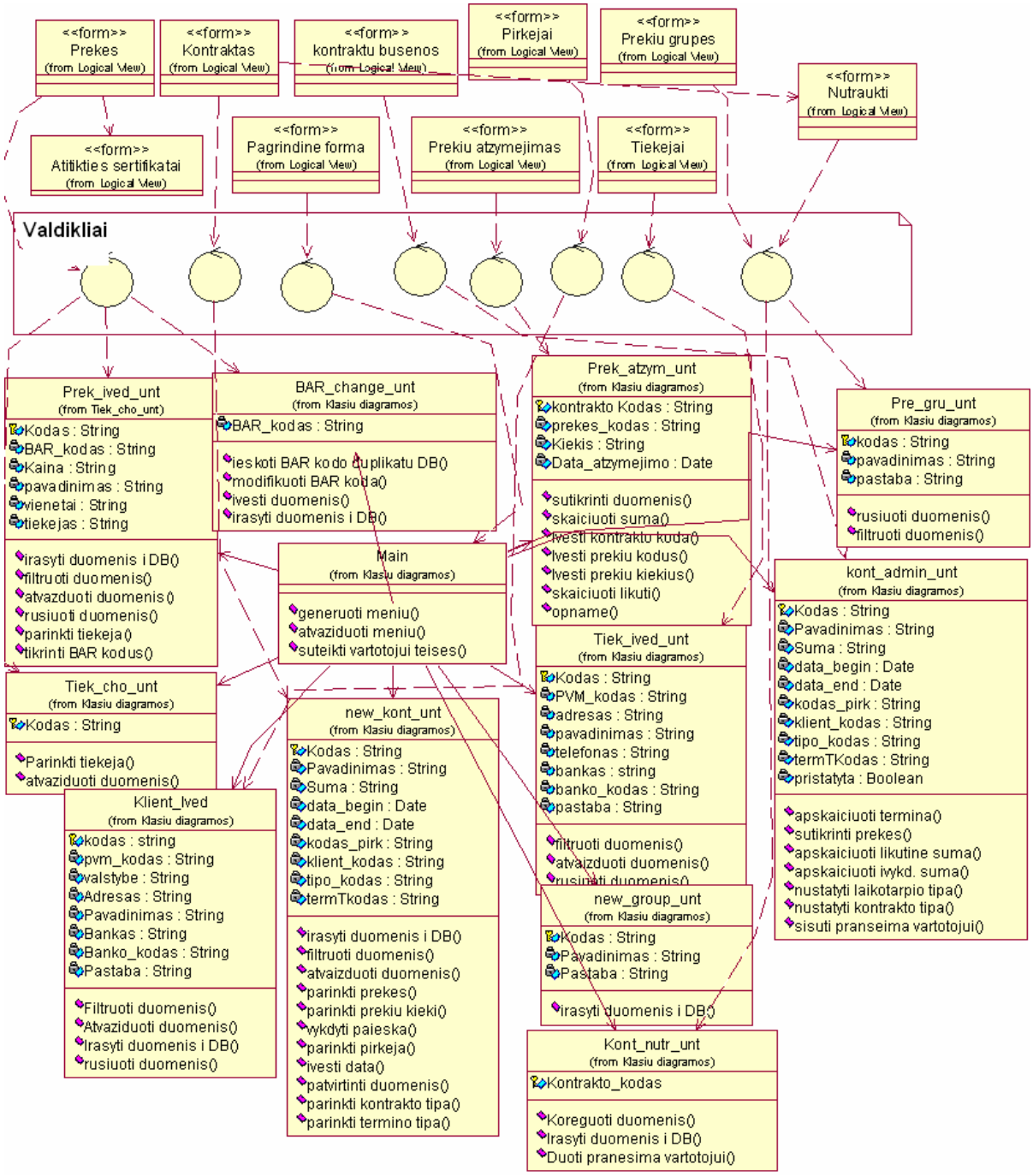
Pavadinimas	Tipas	Dydis	Reikšmė
Sert_kodas	Text	50	Sertifikato kodas
Kodas	Text	50	Prekės kodas

Kontraktas – saugo informaciją apie kontrakto duomenis

Pavadinimas	Tipas	Dydis	Reikšmė
Kodas	Text	50	Kontrakto kodas
Pavadinimas	Text	50	Kontrakto pavadinimas
Suma	Text	50	Pilna kontrakto suma
Data_begin	Date		Kontrakto pradžios data
Data_end	Date		Kontrakto pabaigos data
Data	Date		Pasirašymo data

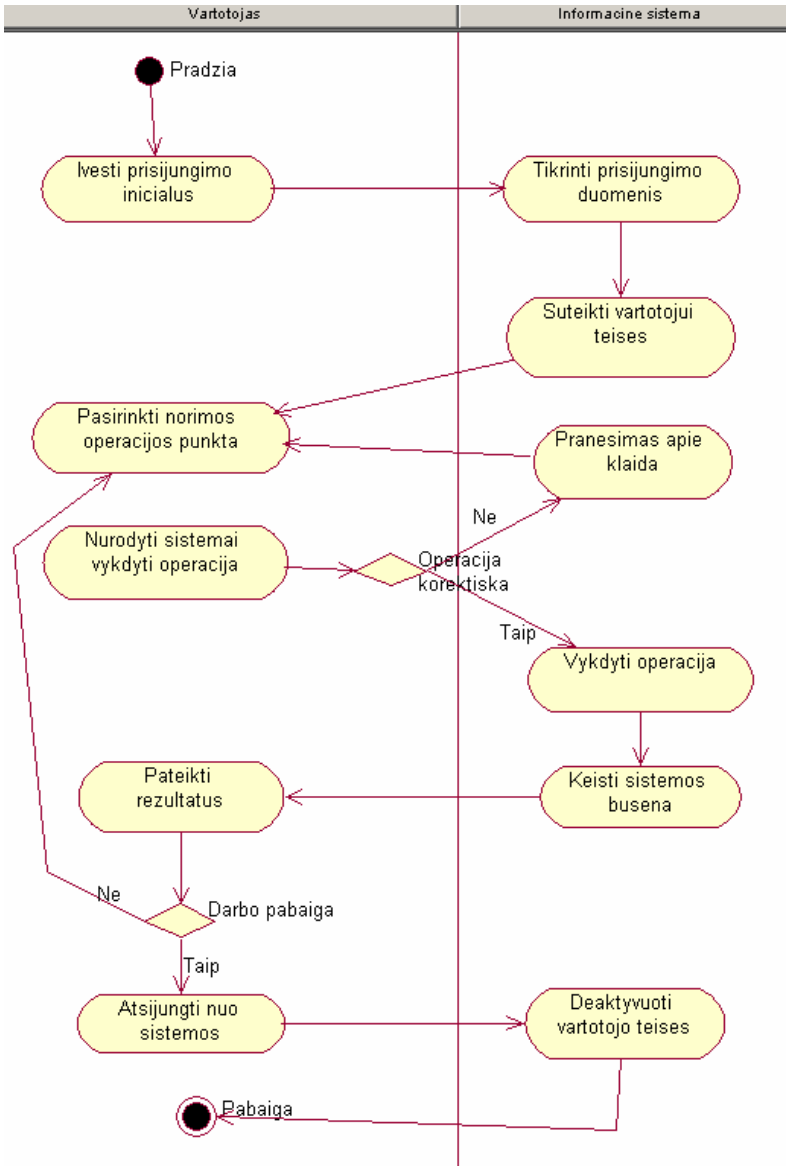
3.4.4 Vartotojo sąsajos ir programinių klasių sąryšio diagrama

Kaip jau buvo minėta anksčiau vartotojo sąsajos modelis yra kuriamas atsižvelgiant į vartotojo nefunkcinius reikalavimus – tai kiekvieno vartotojo individualumas. Sistemos realizuotojui yra aktualu žinoti vartotojo sąsajos ryšį su programinėmis klasėmis, kad ateityje modifikuojant programą, nekiltų kėblumu. Aktualu žinoti, kokia forma ar kitas vartotojo sąsajos komponentas turi poveikį konkrečiai klasei ar kitam programiniam komponentui. Vartotojo sąsajos ir programinių klasių sąryšio diagramą pateiksime 30. Pav



30 Pav Vartotojo sąsajos ir programinių klasių sąryšio diagrama

3.4.5 Sistemos proceso diagrama

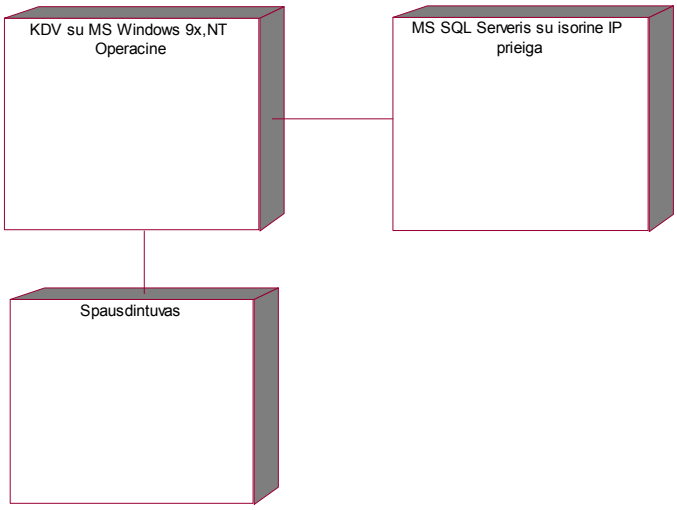


31.Pav Sistemos proceso diagrama

Sistemos proceso diagrama (31.Pav) yra naudojama atvaizduoti sistemos operacijų eigą. Sistemos realizuotojas naudoja šią diagramą ,kaip navigacinį operacijų planą realizuojamoje sistemoje.

3.4.6 Realizacijos modelis (Paskirstymo diagrama)

Paskirstymo diagrama (32.Pav) atvaizduoja sistemos komponentus fiziniame lygyje – naudojamą aparatūrą. Tai aktualu įdiegiant sistemą – žinoma kokios aparatūros reikia.



32.Pav Paskirstymo diagrama

3.5 IS diegimo planas

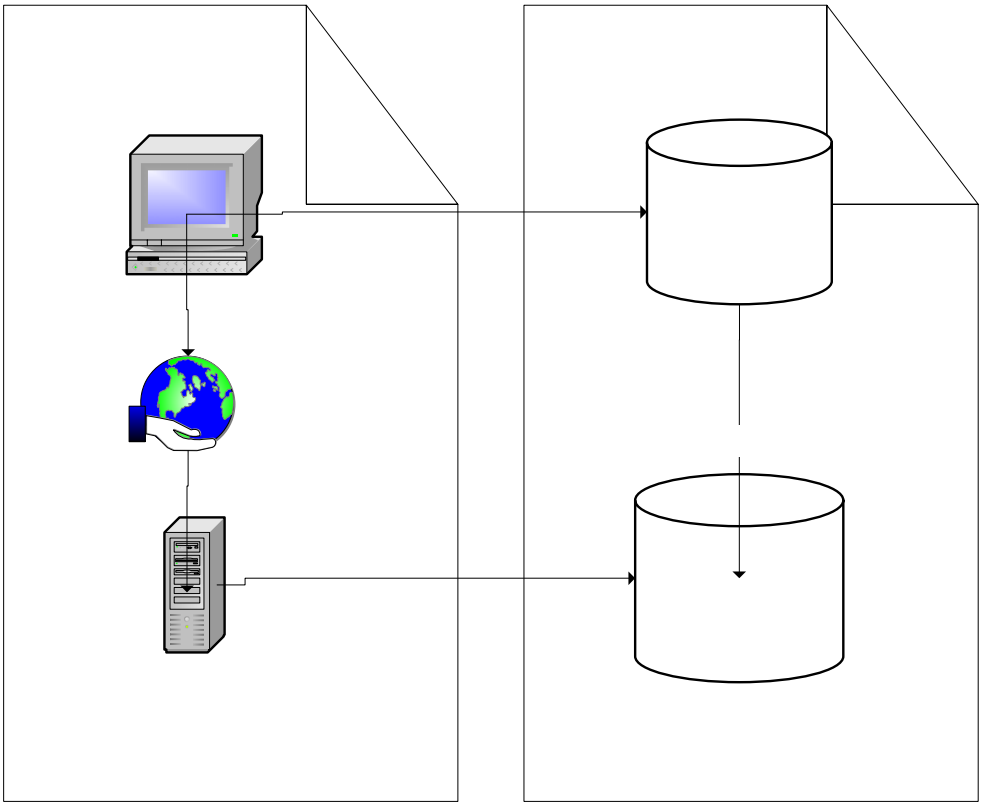
Šiame skyriuje aptarsime kokias priemones naudojome sistemos realizacijai. Kaip buvo minėta anksčiau,kad naudosime kaip DB modifikuotą kontraktų modelį, todėl transformuot duomenų modelį į RDB. Šiai opercaijai atlikti naudojome Microsoft Visio 2003 paketą.

DB programinę įrangą naudojome : vartotojo KDV – Microsoft Access, serveryje – MS SQL Server. Sistemoje numatyti du duomenų identiški duomenų šaltiniai, jeigu įvykus klaidai vienas būtų prarastas. Taipogi serverio duomenų šaltinis gali būti naudojamas esant reikalui įdiegus programinę įrangą kitame kompiuteryje – galima bus naudotis duomenimis neturint savo kompiuteryje KDV DB duomenų.

Realizuoti vartotojo sąsają, bei realizuoti programinę dalį naudojome Borland Delphi 7.0 programinį paketą ir Borland ESB Dates 1.1 bibliotekų paketą.

Šiuos įrankius pasirinkome todėl, kad nėra reikalaujama, kad sistema būtų realizuota per WEB sąsają. Borland Delphi 7.0 paketas turi visas galimybes jungtis prie RDB šaltinių, prie XML šaltinių, taipogi naudoti Microsoft OLAP funkcijas. Todėl iškilus arba pasikeitus ateityje funkciniams reikalavimams keisti sistemos architektūrai neturėtų iškilti problemų su realizavimo įrankių nepajėgumu

Grafiškai atvaizduotą sistemos diegimo planas (33 .Pav)



33 Pav. Sistemos įdiegimo planas

4. Sistemos programinės įrangos testavimas ir vartotojo vadovas

4.1 Sukurtos sistemos vertinimas

Projektuojant ir realizuojant kontraktų finansinės kontrolės IS buvo atsižvelgiama į 2.4 skyriuje išvardintus kriterijus. Šių kriterijų keliamų reikalavimų įgyvendinimas įtakojo esminius architektūrinius sprendimus sistemos projektavimo metu bei realizacijos ypatybes.

Sistema turi būti *automatizuota* ir *autonomiška* – šiam kriterijui įgyvendinti, sukurta atskira klasė, naudojanti ankščiau minėtą Borland ESB Dates 1.1 nemokamą procedūrų rinkinį skirtą atlikti operacijas su laiku ir datomis (date and time routines). Sistema tikrina kiek laiko liko iki kiekvieno kontrakto pabaigos. Laiko tarpui tarp pradžios ir pabaigos sumažėjus iki tam tikros nustatytos reikšmės – sistema pati duoda pranešimus ir nurodo kokios kontrakto sąlygos dar neįgyvendintos

Galimybė *apdoroti* IS esančią informaciją – kaip jau minėta 3.4.2 skyriuje sistema turės du duomenų šaltinius. Vienas iš jų yra vartotojo darbo vietoje, o kitas serveryje. Abu šaltiniai turi identišką informaciją – replikuoja vienas kitą. Vartotojas įdiegęs sistemą kitoje darbo vietoje gali prisijungti prie serverio ir gauti reikalingus duomenis.

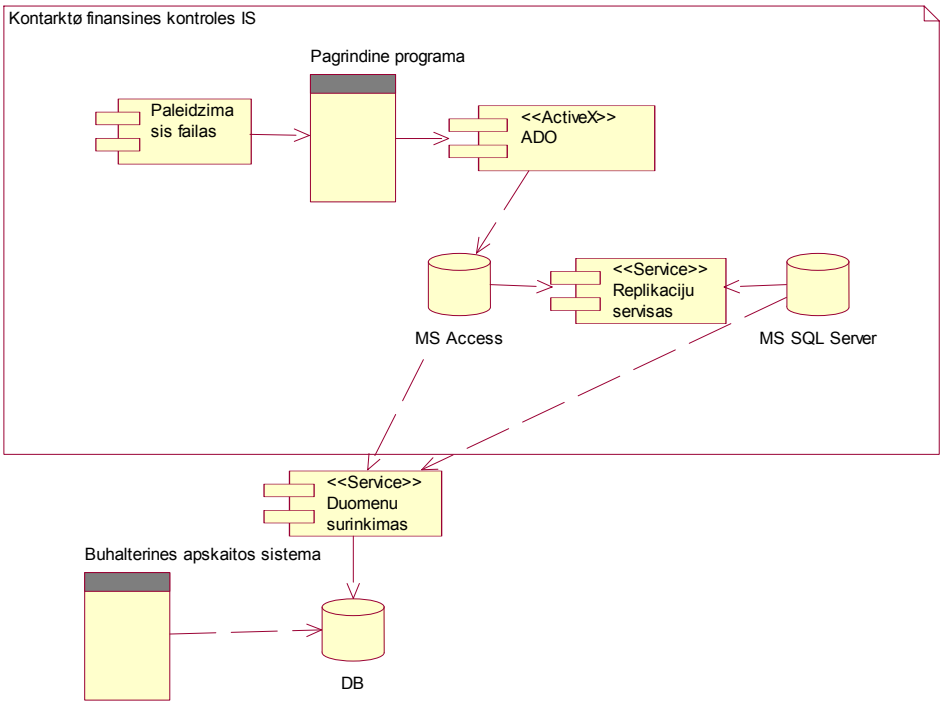
Sistema turi būti *ekonomiškai naudinga* – šis aspektas įgyvendintas dalinai. Tarkime MS SQL Server ir Borland Delphi 7.0 yra brangūs įrankiai, MS Access – įeina į paketo MS Office sudėtį. Borland ESB Dates 1.1 – nemokama. Tačiau visa tai yra pigiau negus įsigyti VVS (Verslo valdymo sistemą). Sumažinti sistemos kainą galime vietoj MS SQL Server pasirenkant Borland Interbase 6.5 duomenų bazių serverį, kuris yra komplektuojamas su Borland Delphi 7.0 paketu, tačiau tokios duomenų apsaugos kaip Microsoft SQL Server neturi

Sistema turi būti *lanksti* – sukurta sistema yra projektuota naudojat RUP metodą. Kiekviena operacijų grupė saugoma atskiruose junginiuose (unit). DB modelis yra tiesiogiai transformuotas iš tipinio kontraktų duomenų modelio – numatyta galimybė sujungti sistemą su kitomis sistemomis pvz: buhalterinės apskaitos sistemomis

Sistema turi būti *saugi* – sistemoje yra naudojamas išorinis duomenų šaltinis, todėl įmanoma įsilaužimo tikimybė. Prie serverio duomenų šaltinio yra nustatytas vartotojų kiekis, kad išvengti informacijos nutekėjimo. Serverio duomenų apsaugai naudojamos naujausios Microsoft SQL Server ir Microsoft Windows 2000 Server siūlomos apsaugos priemonės : **Kerberos** kriptografijos algoritmas duomenų kodavimui.

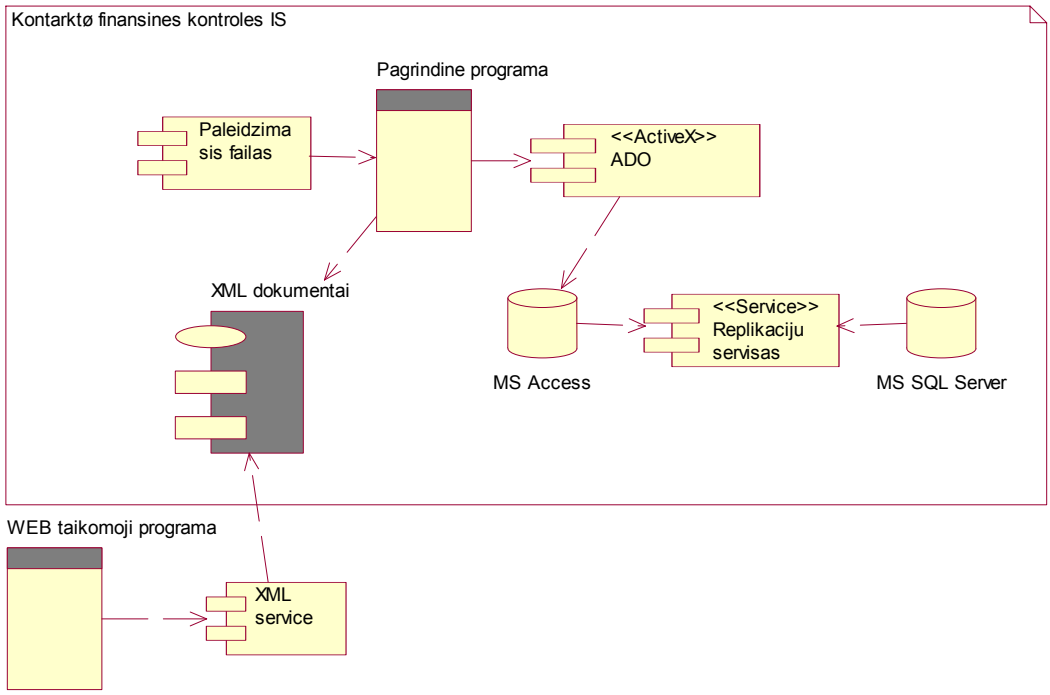
4.2 Sistemos ateities tobulinimo perspektyvos ir alternatyvos

Kaip jau minėta įvade sistema bus naudojama organizacijoje, kurios veikla yra prekyba. Prekyba arba kita komercine veikla užsiimančios organizacijos veda buhalterinę apskaitą. Kadangi mūsų sukurtai sistemai RDB schema yra importuota iš modifikuoto kontraktų modelio, tai ateityje palengvina sujungimą su buhalterinės apskaitos sistemomis, kurios gali būti paruoštos sujungimui su šiuo modeliu. Sujungimą su buhalterinės apskaitos sistemomis atvaizduosime schema 34.Pav



34.Pav ateities tobulinimo perspektyvos

Programoje yra numatoma galimybė įgyvendinti vartotojo sąsają per WEB naršyklę. Borland Delphi neturi galimybės kurti taikomųjų programų kurios veikia per WEB naršyklę, tačiau turi galimybę koduoti ir iškoduoti duomenis XML pavidalu. Tokiu atveju mums truktų tik taikomosios programos veikiančios per WEB naršyklę kuri atvaizduotų vartotojui duomenis iš XML dokumento (35.Pav).

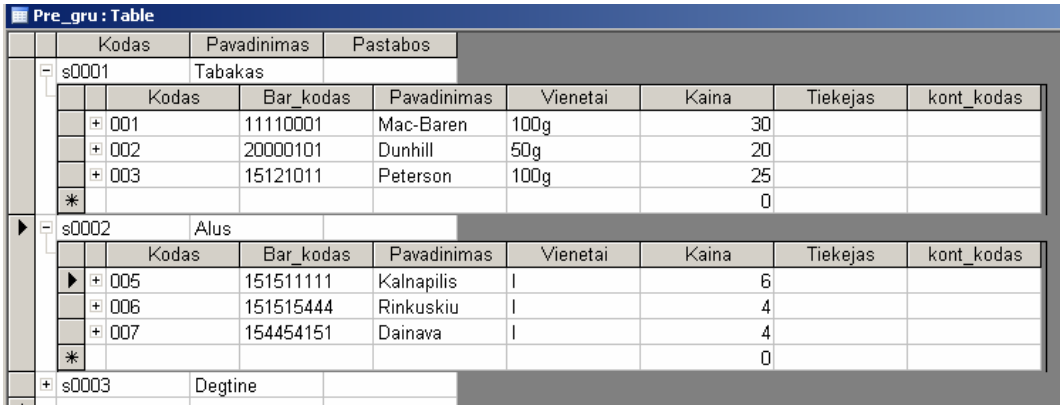


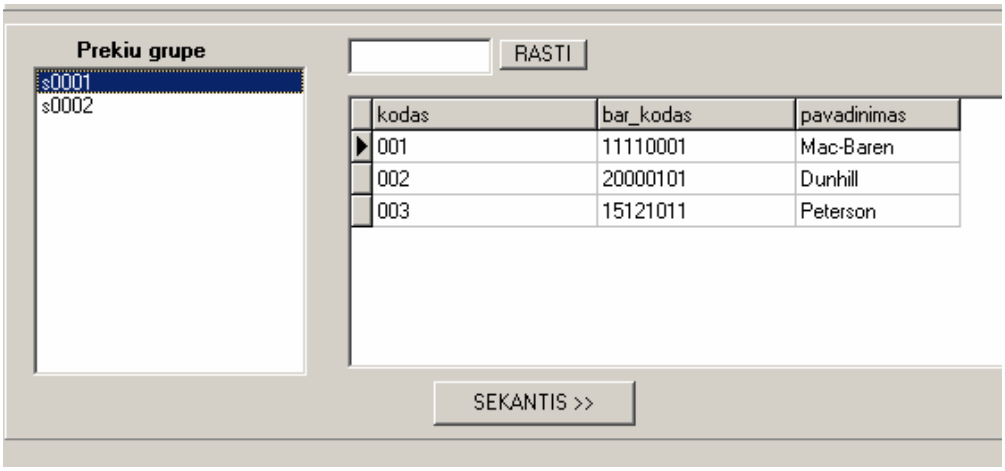
35. Pav ateities tobulinimo alternatyvos

4.3 Testavimo modelis

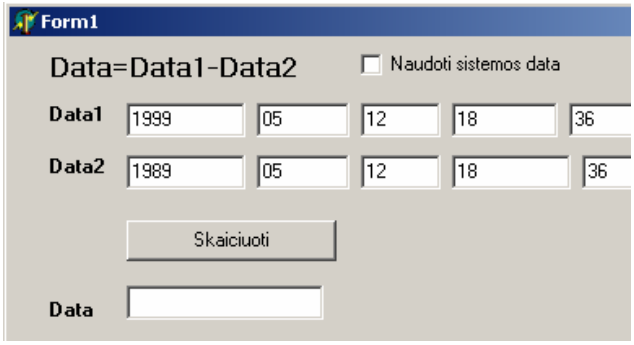
Testavimo metu buvo tikrinama, kaip funkcionuoja sukurta sistema, ar ji teisingai atlieka reikalavimuose specifikuotas funkcijas, kurios turi patenkinti vartotojo poreikius

- **Ar teisingai vykdomas prisijungimas prie sistemos ?**
  Tikrinama ar vartotojams suteikiamos tam tikros teisės
- **Ar teisingai nuskaitomi iš duomenų bazės ir išvedami duomenys?**
  Tikrinama ar duomenys iš DB atvaizduoti programoje,atitinka duomenis esančius DB. Tam yra naudojamos DBVS sukurtos duomenų peržiūros priemonės

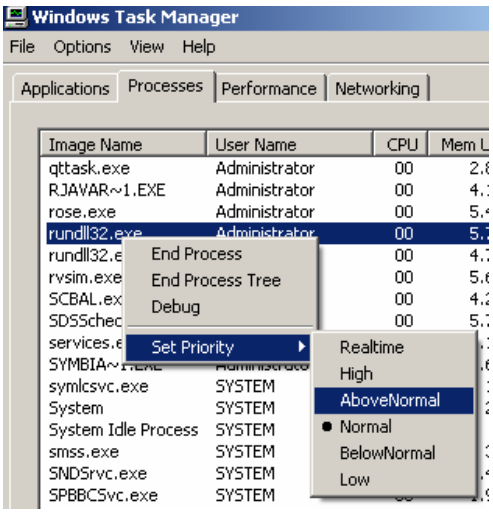




- **Ar teisingus rezultatus pateikia funkcijos naudojančios ESB Dates biblioteką ?**
  Šiuo atveju buvo sudarytas programa,kuri testuoja ESB Dates biblioteką.

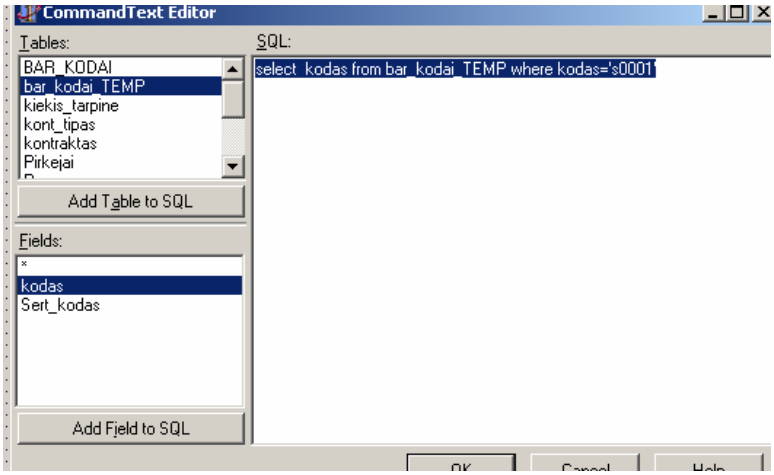


- **Ar teisingai įrašomi duomenis į DB esant sistemos perkrovimams (paleidžiamas procesas su above normal proioritetų OS procesų eilėje) ?**

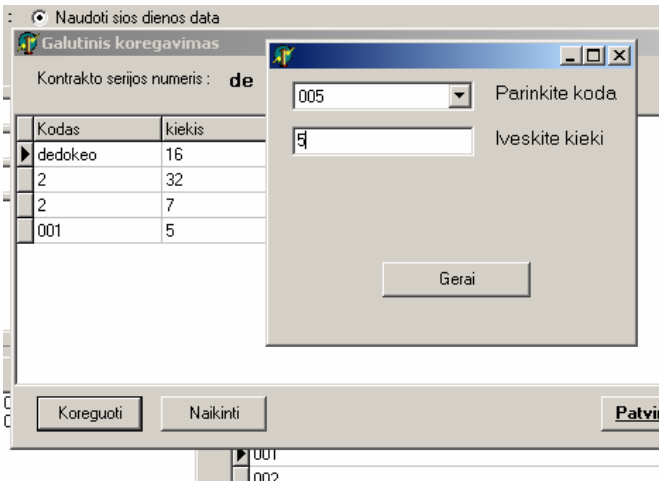


- **Ar teisingai vyksta duomenų filtracija?**

Duomenims filtruoti naudojame reliacinio skaičiavimo kalbą SQL, Borland Delphi paketas turi galimybę patikrinti jo korektiškumą



- **Ar leidžiama koreguoti duomenis ?**

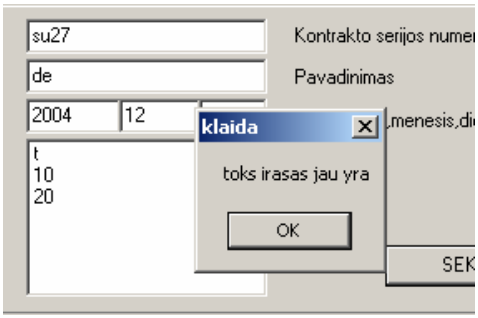


Šiuo atveju pakoregavus duomenis žiūrima, ar įvyko pakeitimų DB, filtruojant DB duomenis naudojant DBVS priemones

- **Kaip vykdoma loginė duomenų kontrolė - bandoma duplikuoti DB raktinį atributą, ESB Dates bibliotekai pateikti nekorektiškus parametrus?**

DB raktinio lauko apsauga ir kitų duomenų loginė kontrolė yra realizuota programiniame kode.

```
adotable2.next;
end; //while
adotable2.first;
adotable2.active:=false;
if x=true then application.messagebox('toks irasas jau yra','klaida
else
begin
if (edit4.text='') or (edit5.text='') or (edit6.text='') or (edit7.
then application.messagebox('ne visi laukai uzpildyti','klaida',0)
```



Sistemos testavimas vyko su personaliniu kompiuteriu, kurio techniniai parametrai pateikiami lentelėje 2.

Lentelė Nr.2. Sistemos testavimo metu naudota techninė įranga.

Procesorius	AMD Athlon Thunderbird 750 MHz
Operatyvioji atmintis	256 MB PC133 SDRAM
Kietasis diskas	Fujitsu MPF3400AT 5400 rpm ATA-100
Operacinė sistema	Windows 2000 Professional, SP-2
MDAC	1.1

Duomenų kontrolinis pavyzdys pateiktas lenetelėje (9. Priedai, 62pslp.)

4.4 Trumpas vartotojo vadovas

Sistemos techninius reikalavimus pateiksime žemiau lentelėje

Pagrindinė programa:

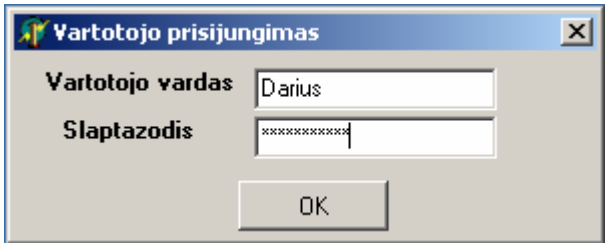
Procesorius	Intel, AMD x86 architektūra ne mažiau 700MHz
Operatyvioji atmintis	64MB RAM
Kietasis diskas	10MB laisvos vietos diske
Operacinė sistema	MS Windows : 9x,NT 4.0,2000,XP
Monitorius	16bit spalvotas min 800x600 rezoliucija

Šie reikalavimai yra identiški (Procesorius, Operatyvioji atmintis ir OS)Borland Delphi 7.0 paketui, kuriuo realizavome sistemą, todėl laikysime šiuos parametrus minimaliais suskurtai taikomajai programai.

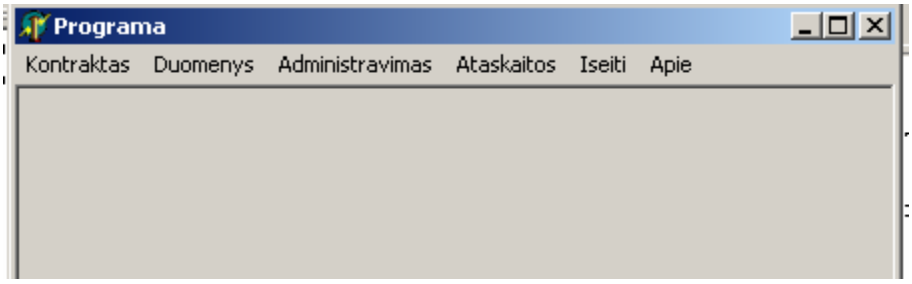
MS SQL Server replikacinė DB:

Procesorius	Intel, AMD x86 architektūra ne mažiau 1GHz
Operatyvioji atmintis	512MB RAM
Kietasis diskas	1GB laisvos vietos diske
Operacinė sistema	MS Windows 2000 Server
Monitorius	16bit spalvotas min 800x600 rezoliucija

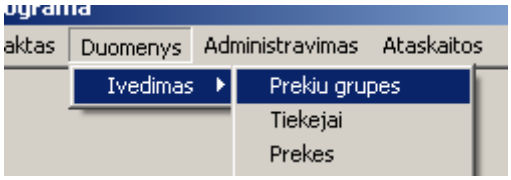
Programinė įranga įdiegiama personalinio kompiuterio diske C: [master volume] sukūrus katalogą **c:\soft.** Paleidžiamasis failas **programa.exe.** Duomenų bazės failas db1.mdb. Paleidus failą **programa.exe** atsidaro vartotojo prisijungimo langas



Įvedus prisijungimo duomenis atsidaro pagrindinis programos langas

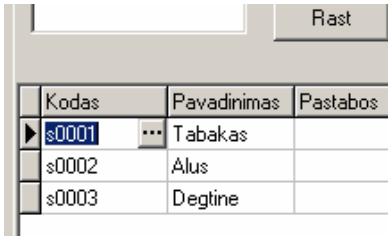


Norint užregistruoti naujas prekes reikia rinktis meniu punktą **duomenys>ivedimas>prekes**

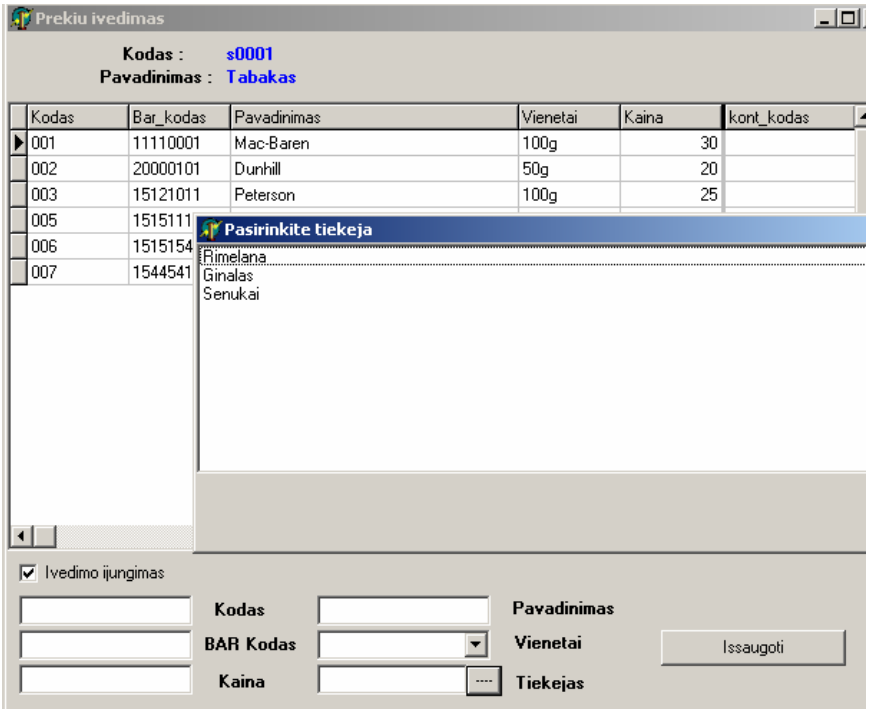


Tuomet atsidaro prekių ivedimo forma

Pasirenkama prekių grupė nuspaudus prie prekių grupės kodo **...** klavišą

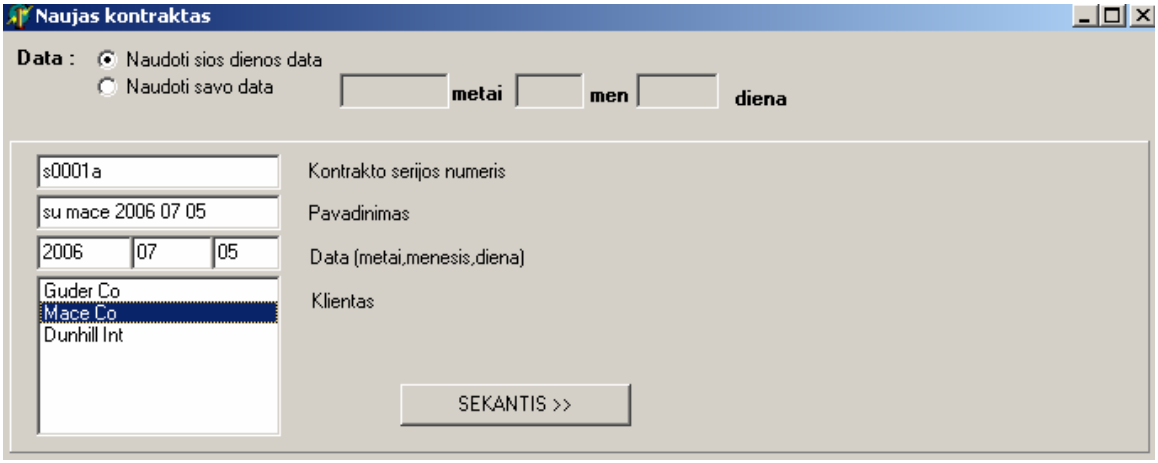


Tuomet atsidaro prekių įvedimo langas

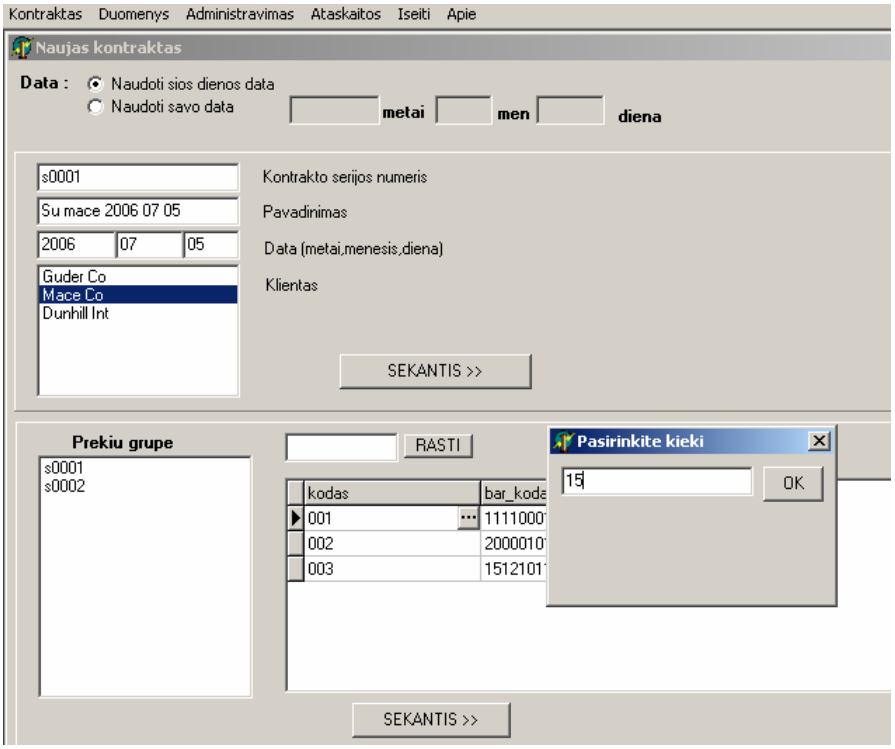


Prekės įvedamos užpildant visus reikiamus laukus

Norint užregistruoti naują kontraktą renkamės meniu punktą **kontraktas>naujas**



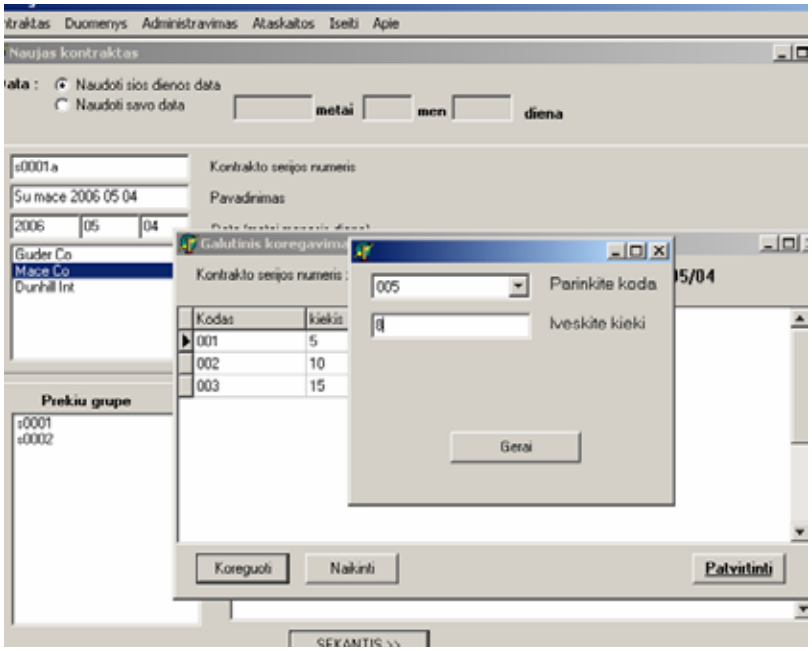
Įvedus reikiamus laukus spaudžiamas klavišas **SEKANTIS >>**

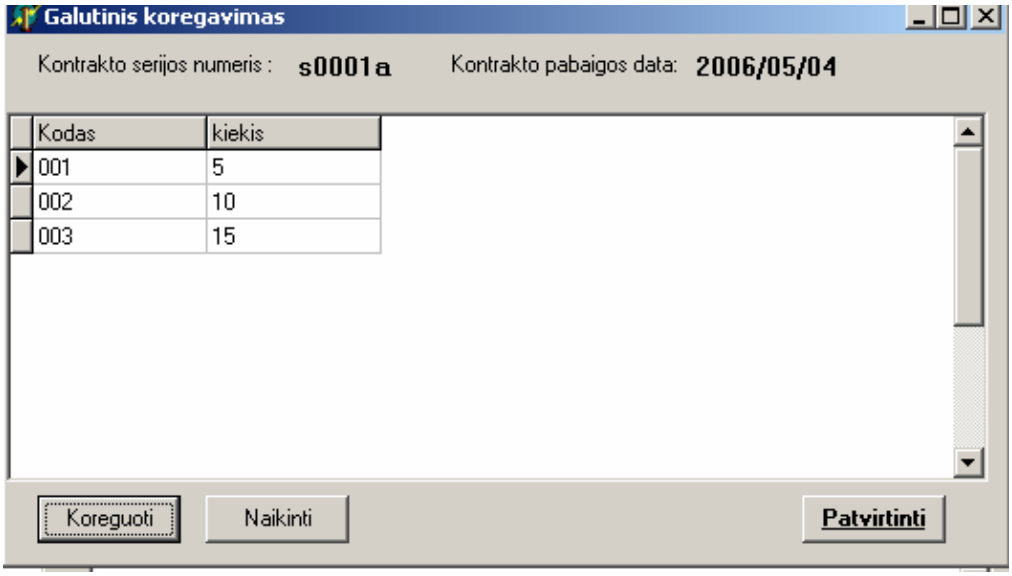


36.Pav Kiekio parinkimas

Tuomet forma prasiplečia iki prekių įtraukimo į kontraktą formos. Prekė pasirenkama prie kodo nuspaudus ..., tuomet atsidaro langas kuriame reikės nurodyti prekių kiekį kontrakte. 36.Pav

Parinkus prekių kiekius toliau spaudžiamas klavišas **SEKANTIS >>,** atsidaro galutinė patvirtinimo forma, kurioje dar galima redaguoti kontrakte esančias prekes





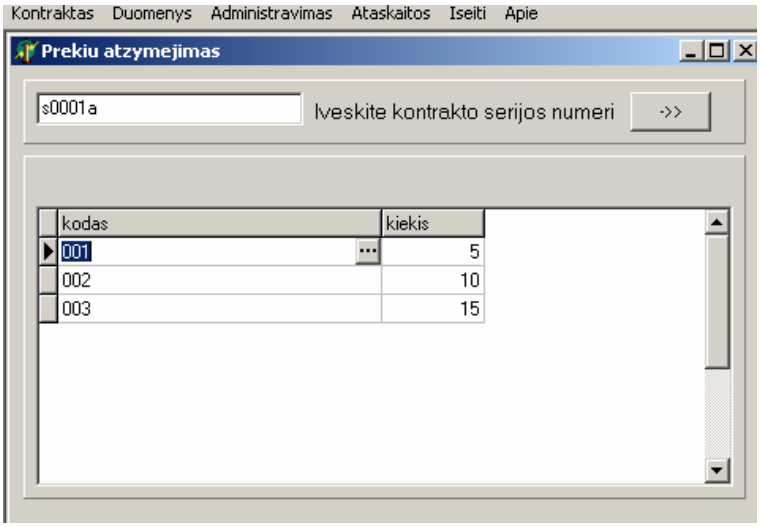
X Pav. Galutinė patvirtinimo forma

Nuspaudus **PATVIRTINTI** patvirtinamas kontraktas su pateiktu prekių sąrašu

Nuspaudus **Naikinti** trinama pažymėta prekė

Nuspaudus **<<Atgal** grįžtama atgal į prekių parinkimo formą.

Norint atžymėti prekes – **Administravimas>>Prekiu atžymėjimas**



Tuomet spaudžiamas klavišas **...** ir įvedamas prekių kiekis. Paskui atsidaro galutinė patvirtinimo forma, kurioje kaip ir kontrakto sudarymo operacijoje 36.Pav galime dar koreguoti duomenis arba galutinai užsaugoti pakeitimus

5. Darbo išvados

- Atlikta egzistuojančios programinės įrangos problemai spręsti analizė. Atlikus šią analizę nuspręsta kurti savo programinę įrangą - nerasta programinės įrangos kurios funkcijos patenkintų visus uždavinio keliamus reikalavimus
- Atlikta egzistuojančių priemonių bei modelių tinkamų daliniam uždavinio sprendimui analizė – Verslo valdymo sistemų (VVS) ir tipinio kontraktų duomenų modelio pagal Len Silverston. Buvo palygintos galimybės : modifikuoti populiarios VVS Microsoft Buisness Solutions CRM modulį arba naudoti kaip DB schemą (struktūrą) tipinį kontraktų modelį. Atsižvelgiant į VVS dideles kainas ir sudėtingumą nuspręsta naudoti tipinį kontraktų modelį ir kurti savo programinį kodą.
- Tipinis kontraktų modelis (pagal Len Silverston) buvo modifikuotas pagal kontraktų finansinės kontrolės uždavinio reikalavimus ir naudotas kaip DB pagrindas sukurtai programinei įrangai.
- Modifikuotas tipinis kontraktų modelis turi :
  - Prekių katalogą
  - Sertifikatų saugyklą
  - Galimybę saugoti fiksuotus duomenis - pristatytų prekių kiekį,datą ir pajamas už jas
- Programinė įranga realizuota naudojant šias priemones – Vartotojo sąsaja, programinis kodas – Borland Delphi 7.0, statinė DB – Microsoft Access, replikacinė duomenų bazė esanti serveryje – Microsoft SQL Server
- IS programinės įrangos testavimas parodė
  - Programinis kodas yra teisingas ir išbandytas su realiais duomenimis.
  - Programinė įranga atlieka visus jai iškeltus funkcinius reikalavimus
- Numatytos sistemos ateities tobulinimo alternatyvos ir perspektyvos
  - Alternatyvos – sukurti taikomąją programą per Web sąsają, kuri iš turimos IS duomenis iškoduotų gautus XML pavidalu
  - Perspektyvos – Sujungimas su buhalterinės apskaitos sistema.Kadangi sistema naudoja modifikuotą tipinį kontraktų modelį , todėl ateityje palengvina sujungimą su buhalterinės apskaitos,kurios naudoja tipinius duomenų modelius

6. Trumpinių žodynas

VVS – verslo valdymo sistema

DB – duomenų bazė

RDB – reliacinė duomenų bazė

ODB – objektinė duomenų bazė

SQL – duomenų bazių užklausų kūrimo kalba (structured query language)

XML – (eXtensible Markup Language) yra visuotinai priimtas ir standartizuotas, paprastas ir lankstus tekstinis formatas, kuris leidžia apibūdinti, apdoroti ir keistis struktūrizuotais duomenimis tarp daugybės skirtingų programų,

UML –unefied modeling language

OS – operacinė sistema

7. Literatūros sąrašas

[1] Rational Unified Process. Rational Software Corporation, 2000. – http://www.rational.com/RUP

[2] The Data Model Resource Book: A Library of Logical Data and Data Warehouse Designs – Len Silverston, 1996

[3] M. Snoeck, G.Dedene. Existence Dependency, the key to semantic integrity between structural and behavioral aspects of object types, *IEEE Transactions on Software Engineering*, Vol. 24 No 4, April 1998

[4] www.omg.org – UML resource page

[5] http://www.objectmentor.com/resources/articles/RUPvsXP.pdf - RUP procesas

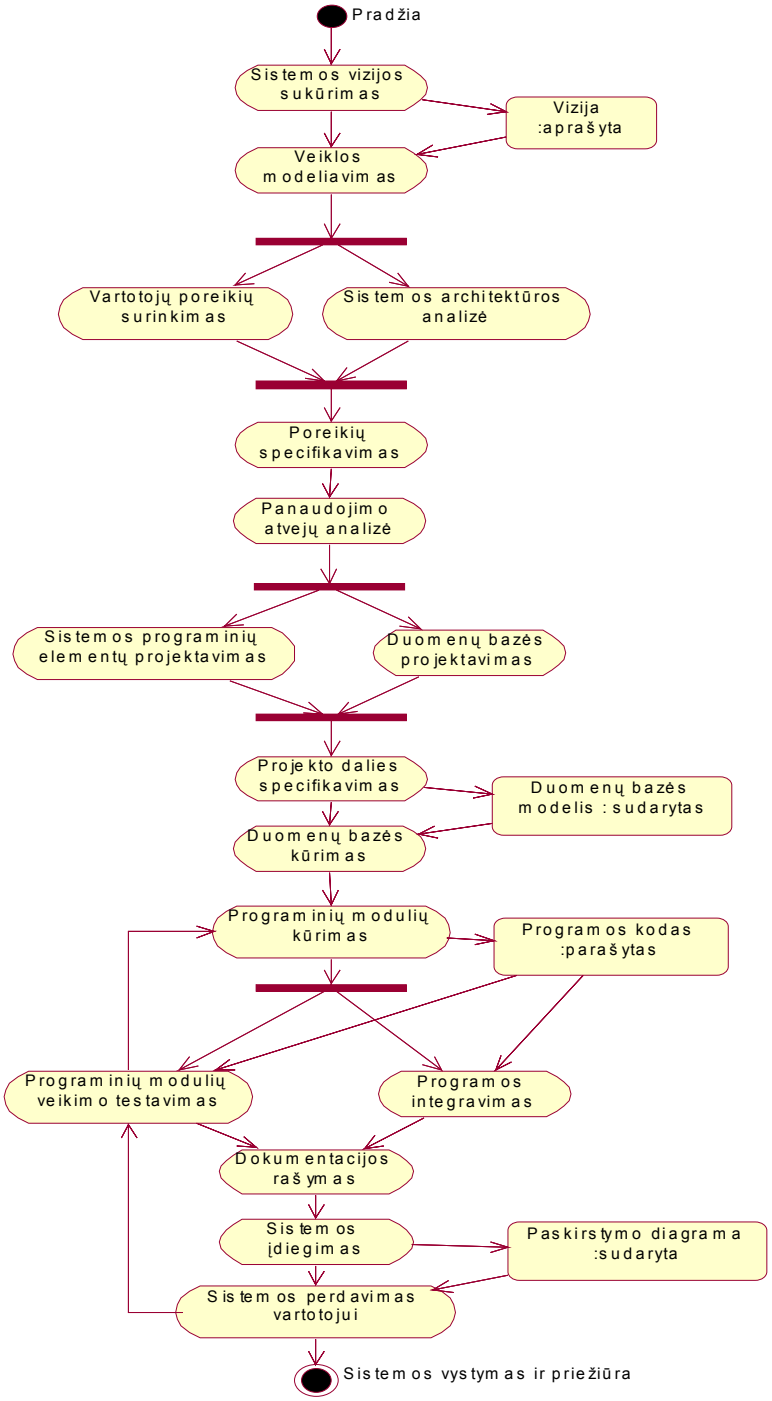
8. Santrauka anglų kalba

Long term contract financial control information system

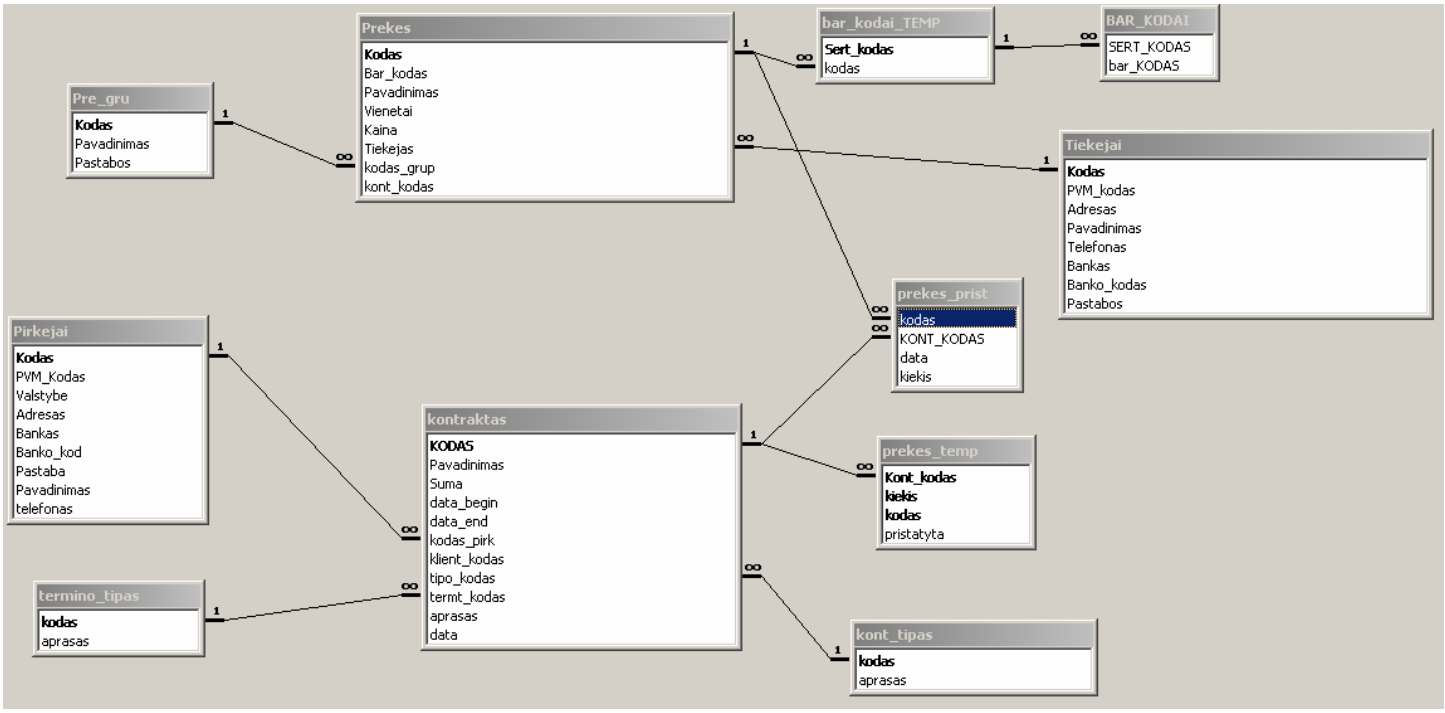
The main goal of these thesis is : to carry out all organization process that must be computerized. There are some processes which man cannot carry out withuot special software. The main critical proces is financial control of contracts. Contract begins with signing between our handing organization and customers. Each contract consists of terms and good‘s that must be delivered at time. The name of thesis „long term...
„ says that contract conditions are exercised in parts. Goods are been delivered in parts and payment is been getting in parts too. So there are many contracts, and the man is unable to control all its himself. So our software must do financial contract control : to mark parts of good of each contract, allow to see every condition of contracts, and allow to solve the problem of goods identification (BAR codes), because our organization is an agent in trading range.

We made our software, theretofore did analysis of excisting software, software that can be partially useful for us and can be upgraded using our functional requirements. We used to modificate typical agreement data model by Len Silverston and use it to transform to DB schema (realationships). Our software (projecting and coding) is made using these tools : Rational Rose, MS Visio, Borland Delphi, MS Access and MS SQL Server

9. Priedai



1.Pav RUP atvaizduotas UML diagrama



2.Pav RDB transformuota iš modifikuoto kontraktų modelio

Testavimo duomenų kontrolinis pavyzdys

Pagrindiniai kontrakto duomenys

Kontrakto serijos numeris	“s0001a”
Pavadinimas	“su mace 2006 07 05”
Nuo datos	Naudoti sistemos datą
Iki datos	“2006/07/05”
Klientas	“Mace Co”

Prekės kontrakte

Kodas	Kiekis
001	5
002	10
003	15