

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

Egidijus Bartkus

**INTEGRUOTOS ĮVAIRIŲ TIEKĖJŲ  
VIRTUALAUS PREKIŲ SANDĖLIO SISTEMOS  
SUKŪRIMAS IR TYRIMAS**

Magistro darbas

Darbo vadovas  
doc. Kęstutis Motiejūnas

KAUNAS, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

Egidijus Bartkus

**INTEGRUOTOS ĮVAIRIŲ TIEKĖJŲ  
VIRTUALAUS PREKIŲ SANDĖLIO SISTEMOS  
SUKŪRIMAS IR TYRIMAS**

Magistro darbas

Recenzentas

Dr. doc. Dalius Rubliauskas

2007 05 29

Vadovas

doc. Kęstutis Motiejūnas

2007 05 29

Atliko

IFM-1/2 gr. stud.

Egidijus Bartkus

2007 05 29

KAUNAS, 2007

## TURINYS

<b>IŽANGA.....</b>	<b>8</b>
<b>1. PROJEKTAVIMO IR PROBLEMINIŲ UŽDAVINIŲ, BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ .....</b>	<b>10</b>
1.1 REIKALAVIMŲ ANALIZĖ .....	10
1.1.1 Reikalavimų užrašymas .....	10
1.1.2 Reikalavimų analizavimo metodai.....	10
1.2 DAUGIASLUOKSNĖS SISTEMOS .....	11
1.3 PATIKIMŲ SISTEMŲ PROJEKTAVIMAS .....	12
1.4 INFORMACINIŲ SISTEMŲ SAUGUMAS.....	13
1.5 OBJEKTINIS PROJEKTAVIMAS .....	14
1.6 TIRIAMOS SISTEMOS ANALIZĖ .....	14
1.6.1 Taikymo sritis ir pradiniai reikalavimai.....	14
1.6.2 Probleminiai uždaviniai .....	16
1.6.3 Probleminių uždavinių sprendimai .....	16
1.6.3.1 Duomenų integravimas iš įvairių tiekėjų .....	16
1.6.3.2 Synchronizacija tarp įmonių grupės serverių .....	18
1.6.3.3 Prekių komplektavimas-suderinamumas .....	19
1.6.3.4 Antkainių valdymas .....	20
1.6.3.5 Integracija su „Rivilė“ .....	20
1.6.4 Realizavimo technologijos pasirinkimas.....	20
1.6.5 Egzistuojančių sistemų lyginamoji analizė .....	21
1.7 PROGRAMŲ SISTEMOS KOKYBĖS VERTINIMO CHARAKTERISTIKOS .....	23
1.8 ANALIZĖS IŠVADOS .....	26
<b>2. INTEGRUOTOS VIRTUALAUS SANDĖLIO SISTEMOS REALIZACIJA .....</b>	<b>27</b>
2.1 VEIKLOS KONTEKSTAS.....	27
2.2 PANAUDOJIMO ATVEJAI .....	28
2.3 FUNKCINIAI REIKALAVIMAI .....	29
2.4 NEFUNKCINIAI REIKALAVIMAI .....	30
2.5 PROGRAMINIO SPRENDIMO ARCHITEKTŪROS SPECIFIKACIJA .....	31
2.5.1 Bendras architektūros vaizdas.....	31
2.5.1.1 Atvaizdavimo lygis (UD).....	32

2.5.1.2	Valdymo sluoksnis (BLL) .....	32
2.5.1.3	Priėjimo prie duomenų sluoksnis (DAL) .....	32
2.5.2	Modelis-Vaizdas-Valdiklis šablonas .....	33
2.5.3	Detalus architektūros vaizdas .....	34
2.5.3.1	„Admin Engine“ modulio architektūra .....	35
2.5.3.2	„XPriceList Data Manager“ modulio architektūra.....	36
2.5.3.3	„Cart Manager“ modulio architektūra .....	37
2.5.3.4	„Providers Service“ modulio architektūra.....	38
2.5.3.5	„Plug-in Handler“ modulio architektūra .....	39
2.5.3.6	„Plug-in’s“ modulių architektūra .....	40
2.5.4	Pagrindiniai duomenų vaizdai .....	41
2.5.4.1	Kompiuterinių prekių saugojimo lentelės .....	41
2.5.4.2	Užsakymų saugojimo lentelės.....	42
<b>3.</b>	<b>EKSPERIMENTINIS SISTEMOS KOKYBĖS IR NAUDINGUMO TYRIMAS.....</b>	<b>44</b>
3.1	SISTEMOS KOKYBĖS TYRIMAS .....	44
3.2	REALIZUOTOS SISTEMOS NAUDINGUMO ĮVERTINIMAS.....	48
3.3	TOLIMESNIO SISTEMOS TOBULINIMO, PLĖTOJIMO GALIMYBĖS.....	53
	<b>IŠVADOS.....</b>	<b>54</b>
	<b>LITERATŪRA .....</b>	<b>56</b>
	<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS.....</b>	<b>58</b>
	<b>PRIEDAS 1: ĮDIEGIMO DOKUMENTAS.....</b>	<b>I</b>
	<b>PRIEDAS 2: APKLAUSA.....</b>	<b>II</b>

# **The Development and Research of an Integrated Virtual Warehouse System of Various Providers**

## **Summary**

The objective of this project was to develop a computerized virtual warehouse for companies that sell computer hardware and software. During its execution, the analysis of design and technology solutions was performed. Further, five basic goals of system realization were formulated, the potential solutions for which were presented in the thesis. The architecture of the developed software is based on the principle of three layer design. During the quality analysis of the system, it was evaluated as an average system except that its maintainability rate was rather low. Its comparison with the alternative systems disclosed its superiority over them from the viewpoint of its expediency.

## PAVEIKSLIUKŲ SĄRAŠAS

<i>1 pav. Daugiasluoksnės sistemos modelis</i> .....	11
<i>2 pav. Sistemos sinchronizavimo lygiai</i> .....	18
<i>3 pav. Antkainių medis</i> .....	20
<i>4 pav. ISO 9126 modelis</i> .....	24
<i>5 pav. Sistemos veiklos kontekstas</i> .....	27
<i>6 pav. Sistemos panaudojimo atvejų diagrama</i> .....	28
<i>7 pav. Projekto valdymo sistemos išskaidymas į sluoksnius</i> .....	32
<i>8 pav. Modelis-Vaizdas-Valdiklis šablono diagrama</i> .....	33
<i>9 pav. Sistemos modulių diagrama</i> .....	35
<i>10 pav. “Admin Engine“ modulio klasių diagrama</i> .....	36
<i>11 pav. “XPriceList Data Manager“ modulio klasių diagrama</i> .....	37
<i>12 pav. “Cart Manager“ modulio klasių diagrama</i> .....	38
<i>13 pav. “Providers Service“ modulio klasių diagrama</i> .....	39
<i>14 pav. “Plug-in Handler“ modulio klasių diagrama</i> .....	40
<i>15 pav. “Plug-in’s“ modulių klasių diagrama</i> .....	40
<i>16 pav. Kompiuterinių prekių saugojimo lentelių diagrama</i> .....	42
<i>17 pav. Užsakymų saugojimo lentelių diagrama</i> .....	43
<i>18 pav. Eksperimentinio tyrimo aspektai</i> .....	44
<i>19 pav. Naudingumo svorinių įverčių palyginimas</i> .....	50
<i>20 pav. Naudingumo ir realizuotų galimybių procentinis palyginimas</i> .....	51
<i>21 pav. Realizuotų funkcijų ir svorinių naudingumo įverčių grafikas</i> .....	52

## LENTELIŲ SĄRAŠAS

<i>1 lentelė. Probleminiai sistemos realizavimo uždaviniai .....</i>	<i>16</i>
<i>2 lentelė. Pasirinktų egzistuojančių sistemų galimybių palyginimas .....</i>	<i>23</i>
<i>3 lentelė. ISO 9126 modelio kriterijai ir subkriterijai .....</i>	<i>24</i>
<i>4 lentelė. Sistemos panaudojamumo grupės.....</i>	<i>28</i>
<i>5 lentelė. Virtualaus sandėlio kokybės vertinimas pagal ISO 9216 standartą.....</i>	<i>45</i>
<i>6 lentelė. Egzistuojančių sistemų naudingumo palyginimas su realizuota sistema (xPriceList) .</i>	<i>48</i>
<i>7 lentelė. Galimybių skaičiaus ir svorinio naudingumo įverčio priklausomybė .....</i>	<i>52</i>

## IŽANGA

### **Pagrindimas**

Daugelyje besivystančių pasaulio šalių, taip pat ir Lietuvoje, šiuo metu vyksta spartus kompiuterizavimo procesas. Bene greičiausiai ir efektyviausiai kompiuterizuojami įvairūs verslo sričių proceso uždaviniai. Jų kompiuterizavimu siekiama palengvinti, pagreitinti bei kokybiškai patobulinti darbą vis greitėjančiame ir konkurencingame pasaulyje.

Greitas ir tikslus informacijos gavimas, lengvas bei centralizuotas jos valdymas, panaudojimas siekiant naudos, tai viena esminių sąlygų, kuri įgalina komercine veikla užsiimančias įmones likti konkurencingomis. Lietuvoje plačiai paplitusios ir išitvirtinusios įvairios apskaitos bei kitos panašaus pobūdžio informacinės sistemos, kurias dar galima būtų pavadinti pirmo būtinumo kompiuterizavimo uždaviniais. Pastaruoju metu pereinama ir prie antro kompiuterizavimo etapo, kai kuriamos įvairios informacinės sistemos specifiniams verslo uždaviniams spręsti, būdingiems tik tam tikrai verslo šakai ar atskirų įmonių veiklai. Svarbus antro kompiuterizavimo etapo požymis yra stiprus kompiuterinių sistemų integravimas. Jis apima ne tik naujai kuriamas tikslines verslo proceso uždavinių kompiuterizavimo sistemas, bet dažnai ir pirmo etapo metu įsigytas kompiuterines programas.

Specializuotų ir integruotų sistemų kūrimas dažnai užtrunka ilgai, o iš įmonių užsakovių pareikalauja daug piniginių lėšų. Atsižvelgiant į tai, mažoms ir vidutinėms įmonėms integruoti sprendimai yra sunkiai prieinami. Vienas iš būdų palengvinančių tokių sistemų įsigijimą yra kompanijų apsijungimas su panašaus profilio įmonėmis, pateikiant vieną bendrą užsakymą kompiuterines sistemas kuriančioms organizacijoms. Būtina sąlyga – įmonės užsakovės turi spręsti tą patį verslo proceso uždavinį. Pavyzdžiui, virtualaus prekių sandėlio sistemos kūrimas.

### **Tikslas**

Sukurti įvairių tiekėjų virtualaus prekių sandėlio sistemą bei atlikti realizuotos sistemos kokybės ir naudingumo tyrimą.

### **Uždaviniai**

Darbo uždaviniai yra tiesiogiai išvedami iš šio darbo pagrindinio tikslo:

- a) atlikti projektavimo ir probleminių uždavinių, bei technologinių sprendimų analizę;
- b) remiantis atliktos analizės išvadomis realizuoti „Integruotą įvairių tiekėjų virtualaus prekių sandėlio sistemą“;
- c) atlikti sukurtos sistemos kokybės ir naudingumo tyrimą.



## **Dokumento struktūra**

Skyrius 1: *Analitinė dalis*. Šiame skyriuje atliekama analizės ir projektavimo metodų analizė bei aptariamos pasirinktos technologinės priemonės. Išanalizuojamos kuriamos sistemos problemos ir galimi sprendimo būdai.

Skyrius 2: *Projektinė dalis*. Šis skyrius aprašo virtualaus sandėlio realizavimą analizės metu pasirinktomis priemonėmis. Taip pat trumpai apibūdina technologijas, architektūros specifiką ir glaustai pateikia sprendimo klasių diagramas bei duomenų vaizdą.

Skyrius 3: *Eksperimentinio tyrimo dalis*. Šiame skyriuje pateiktas realizuotos sistemos tyrimas, kurio tikslas yra sistemos kokybės įvertinimas ir sistemos naudingumo palyginimas.

# 1. PROJEKTAVIMO IR PROBLEMINIŲ UŽDAVINIŲ, BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ

## 1.1 Reikalavimų analizė

### 1.1.1 Reikalavimų užrašymas

Egzistuoja įvairios reikalavimų fiksavimo formos, bet labiausiai paplitę variantai yra reikalavimų fiksavimas į tam tikros formos dokumentus pvz.: *Volere* šablonas ir reikalavimų fiksavimas naudojant automatizuotas reikalavimų apdorojimo sistemas pvz.: *Oracle CASE* įrankis.

*Volere* šablonas naudojamas pradiniam sistemos kūrimo etape ir yra kaip pagrindas užregistruoti vartotojų reikalavimus. Reikalavimai skirstomi į skyrius pagal jų tipus, su detaliais aprašymais. Juose kaupiami reikalavimai, kuriuos pateikia vartotojai per interviu arba kurie nustatomi analizuojant sistemos objekto veiklą. Tai atviras šablonas, kurį galima pritaikyti konkrečiam atvejui. Skyrius, kurie netinka nagrinėjamam objektui, galima išmesti arba sukurti naują skyrių, kuris leidžia specifiškai apibūdinti dalykinės srities charakteristikas. Šablono pagrindu galima aprašyti pagrindinius funkcinis bei nefunkcinius reikalavimus, sudarančius bendrą vartotojo reikalavimų modelį [6].

*Oracle CASE* metodo pagrindu įdiegta *Oracle Designer* priemonė kartu su kitomis *Oracle* priemonėmis suteikia galimybę kompiuterizuotai atlikti daugelį IS kūrimo darbų. *Oracle* siūloma technologija yra viena iš tobuliausių, ji užtikrina integruotą sistemos projektavimą bei programinės įrangos kūrimą [6].

### 1.1.2 Reikalavimų analizavimo metodai

Plačiausiai žinomas ir naudojamas analizavimo metodas yra reikalavimų transformavimas į UML diagramų panaudojimo atvejus. Vieninga modeliavimo kalba (UML) leidžia atlikti orientuotą į objektą sistemos analizę ir projektavimą naudojantis viena nuoseklia kalba programų sistemos rezultatams specifiškai apibūdinti, vaizduoti, ir dokumentuoti [10].

Kitas analizavimo metodas yra sistemos specifikavimas formaliais metodais. Pagrindinė šio metodo problema, kad ne visus reikalavimus galima formaliai specifiškai apibūdinti. Taip pat daugelis formalų specifikavimo kalbų yra gana sudėtingos, reikalaujančios daug laiko jų suvokimui ir praktiniam pritaikymui. Toks formalizmo metodas daugiau taikytinas kritinėms sistemoms, reikalaujančioms specifikacijos teisingumo ir patikimumo įrodymo. Beje, labai dažnai formalizmu apsiribojama tik pateikiant formalias specifikacijas, nes tolesniam plėtojimui trūksta paskatinimo [7].

Trečias reikalavimų analizavimo variantas yra specializuota PĮ pvz.: *Oracle CASE*. Tokiuose sistemos viskas atliekama vienoje vietoje, taip pat, kai kuriais atvejais galima pasinaudoti automatinio generavimo įrankiais. Pagrindinė problema, kad tokios sistemos dažniausiai neapima reikiamo detalumo lygio ir daugiau pasitarnauja priimant projektinius sprendimus [8].

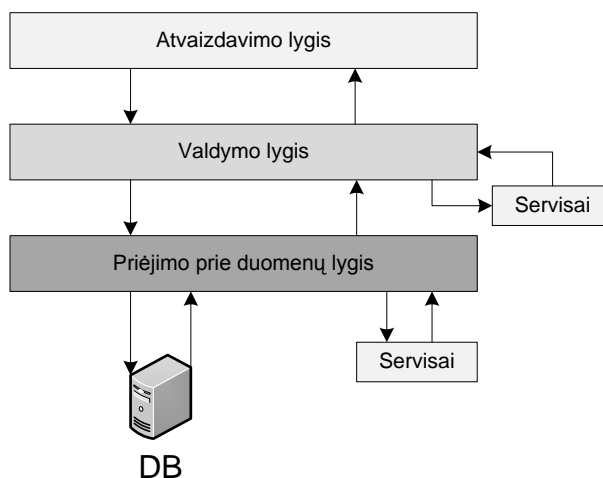
UML suteikia daugiau galimybių nei *Oracle CASE* metodas [11], kuris neužtikrina vartotojo reikalavimų pilnos specifikacijos sudarymo. Naudojantis UML kalba aprašytais dokumentais galima realizuoti pilnai veikiančią sistemą. Tuo tarpu *Oracle CASE* palaiko tik dalį objektinio programavimo principų, todėl galutinis šio metodo rezultatas dažnai dar neatspindi visų sistemos aspektų.

Naudojant UML, vartotojo funkcinius reikalavimus betarpiškai tenkinantys rezultatai – formos, ataskaitos, meniu, kurie dažniausiai išreiškiami per sąsajos elementus, nėra pilnai specifikuojami, o specifikacijos forma neadekvati vartotojo turimam jų įvaizdžiui. Būtent šiems reikalavimams specifikuoti yra paranku naudoti *Oracle CASE* [11] priemones, nes jos turi tam reikiamas priemones – modulių diagramas.

Šiame darbe reikalavimų analizė ir sistemos projektavimas bus atliekamas naudojant RUP modelį, kuris remiasi UML diagramomis.

## 1.2 Daugiasluoksnės sistemos

Vienas iš efektyviausių, kuriamos PĮ sistemos, sudėtingumo mažinimo būdų yra programinio kodo skaidymas į sluoksnius (lygius). Dažniausias naudojamas trijų sluoksnių principas, kai atskiriami vartotojo sąsajos, programos logikos ir duomenų priėmimo sluoksniai (*1 pav.*).



1 pav. Daugiasluoksnės sistemos modelis

Lygiai, kurie daugiau ar mažiau yra tik principas koku būdu organizuojamas programinis kodas į stipriai susijusias panašių užduočių zonas yra naudingi sistemos palaikomumui. Esminis faktorius yra tai, kad pakeitimų darymas viename lygyje sąlygoja minimalius pakeitimus kituose lygiuose, kas programinį kodą padaro kur kas labiau palaikomą. Lygiai taip pat suteikia tam tikrą abstrakcijos lygį nuo jį sudarančių realizacijos ypatumų. Kas įgalina pakeisti reliacinę duomenų bazę į XML be jokių pakeitimų kituose lygiuose [1].

Pagrindinė problema taikant sluoksnius, kaip ir su visom kitom strategijom, proteguojančiom ryšių tarp sistemos komponentų minimizavimą, yra papildomas nereikalingas sudėtingumas. Jūs rizikuojate perprojektuoti savo sistemą pritaikydami ją įvairiausiems pasikeitimams, kurie greičiausiai niekada neįvyks ir nėra tikrų sistemai keliamų reikalavimų dalis [1].

### 1.3 Patikimų sistemų projektavimas

Patikimų sistemų projektavimo principai grindžiami tuo, kad neįmanoma sukurti tobulo produkto, kuris neturėtų jokių defektų, tačiau galima sukurti pakankamai patikimas sistemas [3]. Panagrinėkim parašutistus. Nei vienas jų nepirks ir nepasitikės parašutais iš įmonės sakančios, kad jos parašutai visada išsiskleis. Parašutistai žino, kad nesvarbu kiek kartų viską patikrinsi, kokių procedūrų laikysies vis tiek yra tikimybė, kad tavo pagrindinis parašutas neišsiskleis. Šiems atvejams kaip tik ir skirtas atsarginis parašutas. Tokį nelaimių vengimo būda galima vadinti pertekliškumo principu.

Kuriant PĮ irgi reikia laikytis panašių principų. Neįmanoma pagaminti visiškai patikimo komponento, tačiau kai vienas komponentas tikrina ne tik save bet ir kitų sistemos komponentų darbą galima surinkti vieną pakankamai patikimą sistemą. Pagrindiniai PĮ patikimumo didinimo principai:

- a) defektų mažinimas;
- b) sprendimo paprastumas – padeda išvengti klaidų;
- c) apsaugų pertekliškumas – leidžia numatyti, kad įvykus gedimui vienoje sistemos dalyje bus kita perteklinė sistema, kuri galės atlikti tas pačias funkcijas;
- d) klaidų aptikimas ir apdorojimas.

Šių punktų reikia laikytis ir įgyvendinant šio darbo PĮ, nes bet koks netikslus veikimas arba prastovos klientams kainuos pinigus ir prarastus klientus.

## 1.4 Informacinių sistemų saugumas

Informacinių sistemų saugumas tapo viena iš aktualiausių problemų greitai didėjant kompiuterizacijos lygiui ir kompiuterinių tinklų integracijai. Įvairiuose internetiniuose šaltiniuose ir techninėje literatūroje galima rasti begales patarimų, metodikų ir produktų, kurie nagrinėja programinės įrangos saugumo problemas ir metodus kaip jas galima spręsti.

Nepaisant specifinių kiekvienos sistemos saugumo problemų yra išskiriamos bendrinės PĮ saugumo problemos[5]. Šias problemas turi įvertinti bet kuris PĮ produktas, kuriam aktuali saugumo problema. Bendrinės PĮ saugumo problemos:

- a) buferio persipildymas;
- b) teksto formatavimo problemos;
- c) sql kodo įterpimas;
- d) komandų įterpimas;
- e) išskylančių klaidų neteisingas apdorojimas;
- f) tarp puslapiniai skriptai;
- g) tinklo informacinių srautų nesaugojimas;
- h) „stebuklingų“ url naudojimas ir paslėptos formos;
- i) neteisingas ssl protokolo naudojimas;
- j) lengvų slaptažodžių naudojimas;
- k) nesugebėjimas išsaugoti ir apsaugoti informacijos;
- l) informacijos nutekėjimas;
- m) neleistas failų priejimas;
- n) masyvų ribų tikrinimas;
- o) pasitikėjimas tinklo adresų informacija;
- p) signalų lenktyniavimo situacijos;
- q) neautentifikuotas apsikeitimas raktais;
- r) nenaudojimas kriptografiškai stiprių atsitiktinių skaičių;
- s) sudėtingas sistemos panaudojamumas.

Kai kurias iš šių problemų, padeda išspręsti valdomų programavimo aplinkų pasirinkimas pvz.: Java arba .NET. Tokiose sistemose atliekamas griežta kintamųjų tipų kontrolė, stebimi masyvų ir įvairių operacijų veiksmi, automatizuotai valdoma atmintis, taip minimizuojamos atminties nutekėjimo ir buferio perpildymo problemos [2].

Visas kitas čia paminėtas problemas būtina įvertinti sistemos projektavimo, kūrimo ir diegimo metu. Bet kuriame iš šių etapų padaryta klaida, vėliau gali kainuoti labai brangiai.

## 1.5 Objektinis projektavimas

Dauguma šiuolaikinių programinių sistemų kuriamos taikant objektinio projektavimo principus. Objektinis projektavimas - tai modelis, kuris leidžia į programos objektus žiūrėti kaip į juodas dėžes, kurios tik siunčia ir gauna pranešimus [9].

Objektinio projektavimo principai:

- a) *Inkapsuliacija* - Objektas – tai juoda dėžė, kuri tik siunčia ir gauna pranešimus. Visi objektai gali dirbti nepriklausomai ir žinoti apie kitus objektus reikia tik tiek, kad būtų galima pasinaudoti jų teikiamomis sąsajomis;
- b) *Paveldimumas* - Baziniai objektai realizuoja savo teikiamas funkcijas, kurias toliau galima lanksčiai ir lengvai išplėsti paveldimuose objektuose. Pakeitus bazinį objektą (pavyzdžiui, ištaisius klaidą), pakeitimai persiduoda ir visiems to objekto palikuonims;
- c) *Polimorfizmas* - Tai savybė, kuri apibūdina kodo formos keitimą. Vienas aprašas gali būti naudojamas su skirtingais duomenų rinkiniais.

Tinkamas objektnių principų taikymas leidžia išskaidyti sistema į atskirus mažai tarpusavyje priklausomus modulius. Kūrių kiekvienas vykdo tam tikras funkcijas. Toks sistemos dalių atskyrimas ir struktūrizavimas palengvina sistemos palaikymą ir vėlesnio tobulinimo darbus, kas tiesiogiai atsiliepia kuriamos sistemos projekto kaštams.

Taikant objektnį projektavimą, sistemos architektūros kūrimo metu, reikia nepamiršti, kad yra visas sąrašas sugalvotų ir plačiai dokumentuotų projektavimo šablonų. Kurie nagrinėja visas dažniausiai pasitaikančias sistemų kūrimo problemas ir leidžia net pačias sunkiausias sumažinti iki valdomų ir prognozuojamų sistemos komponentų. Tinkamas objektinio projektavimo šablonų pritaikymas gali sutaupyti nemažai laiko ir išspręsti problemas apie kurias iš pradžių gal net neįtariame.

## 1.6 Tiriamos sistemos analizė

### 1.6.1 Taikymo sritis ir pradiniai reikalavimai

**Taikymo sritis** - elektroninis virtualaus sandėlio sprendimas įmonėms, kurios prekiauja kompiuterine ir programine įranga.

Toliau, šiame skyriuje, pateikiami reikalavimai gauti iš užsakovo, dar prieš pradėdant reikalavimų rinkimą vadovaujantis *Volere* šablonais, t.y. pradiniai vartotojo reikalavimai-norai kuriamai sistemai.

Bendriniai reikalavimai sistemai(verslo aspektu):

- a) būti operatyvia įmonės grupės (IG) reklamos priemone;
- b) atlikti bendrą IG kainodarą;
- c) būti bendro IG marketingo dalis.

Šio projekto užsakovas tai ne viena konkreti įmonė, o įmonių grupė susidedanti iš keleto narių. Jų tikslas yra vykdyti bendrą kainų politiką ir rinkodarą. Po aukščiau pateiktais b ir c reikalavimais, jie supranta tokius dalykus kaip: automatinis prekių, prekių kainų, antkainių ir t.t. sinchronizavimas tarp jų naudojamų sistemų.

Užsakovo pateikti pradiniai funkciniai reikalavimai:

- 1) minimizuoti rankinį darbą skirtą interneto svetainės priežiūrai ir atnaujinimui;
- 2) sandėlio integravimas su įmonės interneto svetaine – įmonės WEB portalo sukūrimas;
- 3) duomenų siuntimas iš tiekėjų – visos IG narės perka savo prekes iš didmenininkų tokių kaip ACME, GNT ir t.t. Visi jie(didmenininkai) suteikia galimybes gauti jų teikiamų prekių sąrašus ir kiekius elektroniniu būdu per WEB servisus;
- 4) duomenų sinchronizacija tarp vienos įmonių grupės serverių – galimybė automatiškai sinchronizuoti prekių sąrašus, kainas, antkainius ir kitą informaciją tarp skirtingų įmonių serverių;
- 5) kompiuterinių prekių katalogo valdymas:
  - a. Prekių grupavimas;
  - b. Prekių komplektavimas-suderinamumas – čia suprantama kaip galimybė pačiam vartotojui susidėlioti norimą kompiuterį, kuris būtų suderinamas(veikiantis);
  - c. Lanksčiai valdomi antkainiai – visoms prekėms, prekių grupei, vienai konkrečiai prekei;
  - d. Garantinis aptarnavimas.
- 6) užsakymų valdymas:
  - a. užsakymų-pasiūlymų valdymas;
  - b. prekių krepšelis;
  - c. akcijos;
  - d. ataskaitų spausdinimas
- 7) sistemos integracija su apskaitos sistema „Rivilė“.
- 8) išsamus vartotojo vadovas.

## 1.6.2 Probleminiai uždaviniai

Iš ankstesniame skyrelyje aprašytų pradinių reikalavimų, žemiau esančioje lentelėje (*1 lentelė*) pateikti pagrindiniai sistemos realizavimo probleminiai uždaviniai. Siekiant sėkmingai išspręsti šiuos uždavinius, analizės ir projektavimo metu turės būti atsakyta į visus lentelėje pateiktus klausimus.

*1 lentelė. Probleminiai sistemos realizavimo uždaviniai*

<b>Probleminiai uždaviniai</b>	<b>Pagrindiniai klausimai</b>
<i>Duomenų integravimas iš įvairių tiekėjų</i>	<ul style="list-style-type: none"> <li>• Ką daryti jei iš skirtingų tiekėjų pateikiami skirtingi duomenų rinkiniai;</li> <li>• Ką daryti su tokiom pačiom prekėm tik iš skirtingų tiekėjų, kaip jas pateikti vartotojui.</li> </ul>
<i>Sinchronizacija tarp įmonių grupės serverių</i>	<ul style="list-style-type: none"> <li>• Kurio serverio duomenys svarbesni;</li> <li>• Ką daryti jei informacija apie tą pačią prekę skiriasi;</li> <li>• Kada prekės pašalinamos iš sistemos;</li> </ul>
<i>Prekių komplektavimas-suderinamumas</i>	<ul style="list-style-type: none"> <li>• Kokiu būdu bus nustatoma ar prekės suderinamos, kas tokia informacija suves, jos suvedimo būdas.</li> </ul>
<i>Antkainių valdymas</i>	<ul style="list-style-type: none"> <li>• Kokie tikslūs antkainių valdymo algoritmai;</li> </ul>
<i>Integracija su „Rivilė“</i>	<ul style="list-style-type: none"> <li>• Kokie protokolai/duomenų formatai;</li> </ul>

## 1.6.3 Probleminių uždavinių sprendimai

Šiame skyrelyje pateikiami 1.6.2 skyrelyje iškeltų probleminių sistemos realizavimo uždavinių sprendimų aprašymai.

### 1.6.3.1 Duomenų integravimas iš įvairių tiekėjų

Detaliau paanalizavus iš įvairių tiekėjų gaunamą informaciją ir pabandžius ją integruoti į vieną sistemą, iškeltos tokios pagrindinės problemos:

1. skirtingi tiekėjų naudojami kodai, jų unikalumo užtikrinimas;
2. skirtingų tiekėjų skirtingai nurodomi tie patys kokios nors prekės gamintojai pvz.: „intel co.“ ir „intel“;
3. to paties gamintojo, ta pati prekė, tačiau tiekama kelių tiekėjų, kartais net su skirtingomis kainomis ir kitais parametrais;
4. tiekėjų nebeteikiamų prekių pašalinimas iš sistemos.

Kiekvienos iš šių problemų galimas sprendimas:

1. Sistemoje naudoti papildomą sudėtinį prekės identifikavimo lauką. Jis susideda iš dviejų dalių [*tiekėjoPrefiksas + tiekėjoNaudojamasKodas*].



*tiekėjoPrefiksas* – nustatomas sistemos kūrėju, realizuojant automatinį duomenų importavimą tam tikram tiekėjui, šitie priefiksai niekada nesikeičia. Pvz.: acme, gnt.

*tiekėjoNaudojamasKodas* – kodas gaunamas iš tiekėjo WEB serviso pateikiamų duomenų.

Naudojant toki sudėtinį kodą, užtikrinamas kiekvienos prekės unikalumas. Sekantį kartą importuojant duomenis galima be klaidų atnaujinti jau egzistuojančių prekių informaciją arba įvesti informaciją apie naujas prekes.

2. Siekiant išspręsti problemą su skirtingai užrašomais prekės gamintojo pavadinimais, tam kad vėliau sistemos klientas galėtų be vargo atsifiltruoti duomenis, galima sukurti gamintojų filtrų sistemą.

Gamintojų filtrų sistema susidėtų iš dviejų esminių punktų:

- a) iš tiekėjų ateinantys pavadinimai administratoriaus yra priskiriami tam tikram vienam gamintojo filtrui. Pvz.: jei vienas gamintojas naudoja „intel co.“, kitas gamintojas „intel“, tai administratorius sukuria filtrą „Intel“ ir jam priskiria kiekvieno konkretaus tiekėjo naudojamas reikšmes. Tokiu būdu prekės su „intel co.“ ir „intel“ gamintojais, atliekant paiešką pagal gamintojo filtrą „Intel“, bus rodomos vartotojui;
  - b) siekiant sumažinti administratoriui tenkanti darbą, realizuojamas dalinis gamintojų filtrų sistemos automatizavimas. Jo esmė: priskirtų grupėms konkretaus tiekėjo naudojamų aprašymų analizavimas ir automatinis naujai gautų prekių priskyrimas gamintojų grupėms. Jei koks nors kodas dar nėra priskirtas jokiame filtrui, tokia prekė priskiriama grupei „Nežinomas“. Prekės priskirtos grupei „Nežinomas“ neturėtų būti rodomos galutiniam sistemos klientui.
3. Labai dažnai visiškai ta pati prekė yra tiekama kelių tiekėjų už skirtingą kainą. Tokių situacijų pašalinimui galima sukurti prekių apskaitos kodo posistemę. Jos esmė būtų tokia:
    - a) Prekės gali būti grupuojamos pagal tam tikrą kodą. Jas grupuoja administratorius;
    - b) Klientui visos vieno apskaitos kodo prekės rodomos kaip viena prekė;
    - c) Klientui rodoma pigiausia kaina iš visos prekių grupės;

d) Klientui rodomas bendras prekių kiekis esantis grupėje.

Ši posistemė veikia gretimai su paprastomis prekėmis, kurios nėra sugrupuotos pagal apskaitos kodą.

4. Didžioji dalis tiekėjų nebeteikiamų prekių tiesiog neįtraukia į atnaujinamų duomenų sąrašus, todėl susiduriama su sandėlyje nebesamų prekių pašalinimo problema.

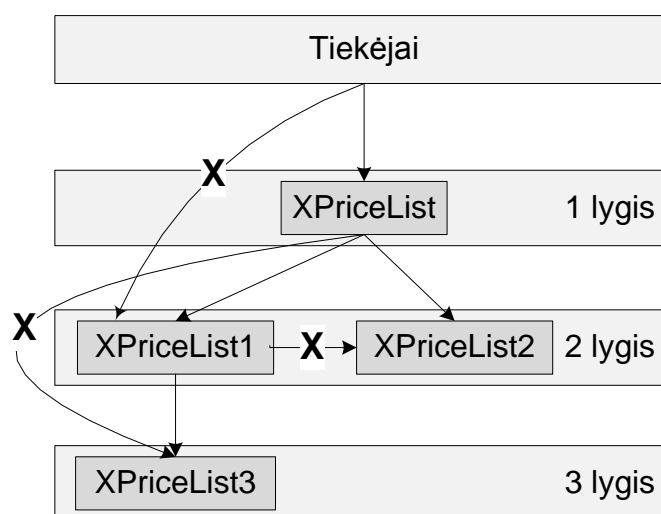
Norint ją išspręsti, kiekvienai prekei galima numatyti du papildomus parametrus: *prekės lygis* – kuriame lygyje prekė buvo sukurta ir *prekės atnaujinimo data* – kada paskutini kartą atnaujinta prekės informacija.

Pagal prekės lygį nustatoma, kurias prekes reikia automatiškai šalinti. Šalinamos tik iš aukštesnių lygių atėjusios prekės. T.y. prekės, kurios buvo įvestos konkrečios sistemos administratoriaus, niekada automatiškai nepašalinamos.

Prekės atnaujinimo data automatiškai pakeičiama kiekvieną kartą importuojant duomenis iš tiekėjų. Atlikus importavimo procedūrą iššaukiamas automatinis nereikalingų prekių pašalinimo procesas. Jis pašalina visas prekes kurios buvo atnaujintos seniau nei tam tikras, per parametrus nustatytas, laiko intervalas ir yra iš aukštesnio lygio.

### 1.6.3.2 Sinchronizacija tarp įmonių grupės serverių

Pagrindinė problema sinchronizuojant duomenis tos pačios sistemos viduje, tai nustatymas kurie duomenys yra viršesni. Šios problemos sprendimui sistemas galima suskirstyti į lygius (2 pav.).



2 pav. Sistemos sinchronizavimo lygiai

Suskirsčius sistemas į lygius ir priėmus, kad kiekviena žemesnio lygio sistema gali sinchronizuotis tik su viena, vienu lygiu aukštesnio lygio, sistema, išsprendžiamos visos duomenų viršumo problemos.

Duomenys atėję iš aukštesnio lygio visada skaitomi naujesniais ir atnaujinama sistemoje esančius duomenis. Senų duomenų trynimas atliekamas tokiu pačių principu kaip ir atliekant importavimą iš tiekėjų WEB servisų t.y. jei tam tikra prekė atėjusi iš aukštesnio lygmens nebuvo atnaujinama tam tikrą laiko intervalą ji yra pašalinama.

### 1.6.3.3 Prekių komplektavimas-suderinamumas

Suderinamo kompiuterio susirinkimo galimybės realizavimui siūloma tokia posistemė, kuri analizuotų prie kiekvienos prekės pateikiamus suderinamumo parametrus. Kompiuterį surinkinėti bus galima tik iš detalių, kurios turi nurodytus suderinamumo parametrus.

Patys suderinamumo parametrai, tai tekstinė eilutė. Susidedanti iš raktinių žodžių su išanksto numatytais funkciniais simboliais, vienas nuo kitų atskiriamų kabliataškiais.

Kompiuterių sisteminės plokštės turės atskirą požymį nurodomą ne per parametrų eilutę. Sistemines plokštės skirsis nuo kitų komponentų tuo, kad į kompiuterių jų įtraukti bus galima tik vieną ir prie jų pateiktas suderinamumo parametras bus imamas už bazę nustatant kokios kitos detalės gali būti įtrauktos į komplektą.

Žemiau pateikiamas galimas pavyzdys:

Sisteminė plokštė – „1ppAMD;2x1ppDDR400;4xPCIE;2PCI;“

Šitame kode „pp“ ir „x“ yra funkciniai simboliai. Funkciniai simboliai visada eina prieš raktinį žodį.

„[skaičius]pp“ – reiškia, kad tam tikras skaičius komponentų yra privalomas.

„[skaičius]x“ – nurodo, kad tam tikras komponentų skaičius yra maksimalus ir daugiau netelpa.

Pagal pateiktą suderinamumo kodą matome, kad šiam kompiuteriui būtinai turime priskirti vieną prekę su kodu AMD ir bent vieną prekę su kodu DDR400, bet ne daugiau kaip dvi tokias prekes(DDR400). Taip pat galime priskirti maksimaliai keturias prekes su kodu PCIE ir maksimaliai dvi prekes su kodu PCI.

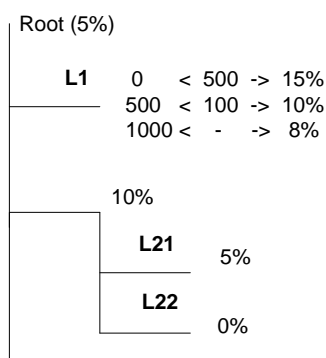
Aprašinėjant paprastus komponentus t.y. ne sistemines plokštes, funkciniai simboliai nenaudojami, tačiau vienas komponentas gali turėti kelis raktinius žodžius pvz.: „PCIE;VGA10;“.

### 1.6.3.4 Antkainių valdymas

Išanalizavus užsakovo pateiktas galimas antkainių variacijas, panašių sistemų teikiamas galimybes ir numatomus vėlesnius poreikius, siūloma realizuoti antkainių posistemį laikantis tokių taisyklių:

- kiekviena nauja prekė pažymima kaip turinti automatinį antkainį;
- kiekviena prekė gali turėti neautomatinį, ranka nurodomą, konkretų antkainį;
- kiekvienai prekių medžio(katalogo) šakai galima priskirti tam tikrą antkainį;
- priskiriami antkainiai gali būti priklausomi nuo konkrečios prekės savikainos;
- automatinis antkainis gaunamas susumuojant visų šakų kurioms ta prekė priklauso antkainius.

Žemiau esančiame paveiksliuke (3 pav.) pavaizduotas galimas antkainių medis. Jį nagrinėjant pagal anksčiau pateiktas taisykles matome, kad prekė priklausanti šakai L22 turės 15% antkainį. Prekė priklausanti šakai L21 - 20%. O prekės priklausančios šakai L1 turės kintantį antkainį ir jis bus 20%, 15% arba 12% priklausomai nuo tos prekės savikainos.



3 pav. Antkainių medis

### 1.6.3.5 Integracija su „Rivilė“

Susisiekus su apskaitos sistemos „Rivilė“ kūrėjais, buvo išsiaiškinta, kad ji palaiko WEB servisus per kuriuos galima perduoti duomenis. WEB servisui perduodami duomenys yra XML formato. Peržvelgus pateiktą XML failo struktūrą jokių keblų vietų nerasta. Automatinės ir pastovios sistemos sinchronizacijai su „Rivilė“ palaikymui, geriausias variantas būtų sukurti Windowsinį servisą.

### 1.6.4 Realizavimo technologijos pasirinkimas

Ne taip senai atsirado tokios sąvokos kaip kliento-serverio architektūra, ir tokios priemonės kaip Java, ActiveX, XML, ASP.NET. Šios technologijos sukūrė naujas verslo erdves, o tiksliau

panaikino tam tikras ribas. Dėl pakankamai naujų galimybių ir kokybinio darbo pasikeitimo, internetinės realizacijos šiuo metu turi vieną didžiausių svarbų PĮ technologijų pasaulyje.

Tai technologijos, sukuriančios darbo vietas, kurios leidžia dirbti „kartu, bet atskirai“ [12].

Apžvelgę sistemai keliamus reikalavimus galime iš anksto numatyti, kad jos kūrimui reikės technologijų kurios leistų realizuoti WEB portalą, naudotis WEB servais ir paleisti nuolat veikiančią „Servisą“.

Šiuo metu pasaulyje naudojamos trys pagrindinės WEB platformos: *PHP*, *.NET* ir *Java*. Yra ir kitų variantų, bet jie kur kas mažiau išplėtoti ir dažnai susiję su papildomomis kūrimo ar galimybių trūkumo problemomis, todėl jų plačiau nenagrinėjant buvo atmesti.

*PHP* tai greitai išmokstama ir pakankamai galinga programavimo kalba WEB puslapių kūrimui. Tačiau, dėl nepakankamai išvystytų IDE, su ja sunkiau kurti ir naudoti WEB servais. Taip pat *PHP* praktiškai neturi galimybės veikti kaip „Servisas“ ar jo atitinkmuo.

*.NET* ir *Java* tai dvi, šiuo metu, aktyviai konkuruojančios technologijos. Jos abi remiasi ta pačia „valdomo kodo“ koncepcija. Ši koncepcija išsprendžia nemažai problemų susijusių su atminties valdymu, masyvų ribų valdymu ir t.t. Pagal savo teikiamas galimybes abi šios platformos yra faktiškai identiškos. Realus ir pagrindinis skirtumas yra tai, kad *Java* gali veikti beveik visose egzistuojančiose operacinėse sistemose, kai tuo tarpu *.NET* realiai yra pririšta prie „*Windows OS*“.

Galiausiai sistemos realizavimui buvo nuspręsta pasirinkti *.NET* platformą. Apsisprendimą nulėmė kelios priežastys. Pirmiausia tai, kad sistemai pateiktuose reikalavimuose nurodyta naudoti „*Microsoft SQL Server*“ DBVS, o *.NET* platformą ir jei skirti IDE turi geresnę integraciją su šia duomenų bazių sistema. Antra kūrėjai turi didesnę patirtį dirbant su *.NET* platforma nei su *Java*.

### 1.6.5 Egzistuojančių sistemų lyginamoji analizė

Atliekant egzistuojančių sistemų analizę, nepavyko gauti detalių, visų sistemas sudarančių dalių, aprašymų. Dėl šios priežasties, žemiau pateiktoje lentelėje (2 lentelė) lyginami tik viešai prieinami WEB portalų moduliai.

Moduliai buvo lyginami pagal tokias galimybes:

- a) *Prekių medis* – ar yra pateikiamas kompiuterinių prekių medis;
- b) *Prekių krepšelis* – ar yra galimybė susidėti norimas prekes į virtualų krepšelį ir vėliau jas užsisakyti arba atsispausdinti;

- c) *Galimybė susikomplektuoti suderinamą kompiuterį* – ar galima, iš atskyrų dalių, susikomplektuoti kompiuterį, kurio sudedamosios dalys būtų suderinamos;
- d) *Paieška* – elementari paieška pagal įvestą žodį;
- e) *Detali paieška* – ar yra galimybė ieškoti pagal įvairius parametrus pvz.: gamintojas, kaina, kiekis ir t.t.;
- f) *Kainų pokytis* – ar koku nors būdu pažymimos prekės, kurių kaina neseniai pasikeitė;
- g) *Kainų kitimo grafikas* – ar galima pažiūrėti kaip keitėsi prekės kaina praeityje;
- h) *Naujos prekės* – ar kaip nors išskiriamos naujos prekės;
- i) *Akcijinės prekės* – ar sistema turi galimybę išskirti akcijines prekes su mažesne kaina;
- j) *Kiekis sandėlyje* – ar rodoma kokios šiuo metu turimos prekių atsargos;
- k) *Prekės aprašymas* – ar pateikiamas detalus prekės aprašymas;
- l) *Gamintojo aprašymas* – ar pateikiamas prekės gamintojo aprašymas;
- m) *Prekių sąrašo spausdinimas* – ar yra galimybė, pagal tam tikrus parametrus, išsifiltruotą sąrašą atsispausdinti;
- n) *Prekių sąrašo eksportavimas* – ar yra galimybė išsifiltruotą arba visą sąrašą gauti kitu formatu pvz.: excelis, wordas ir pan.
- o) *Prekių sąrašo puslapiavimas* – ar vartotojui suteikiama galimybė pasirinkti po kiek prekių, vienu metu, rodoma ekrane;
- p) *Užsakymų registravimas* – ar yra galimybė į prekių krepšelį susidėtas prekes iškart užsisakyti;
- q) *Registruoti klientai* – ar sistema palaiko nuolatinių klientų registravimą;
- r) *Prekių palyginimas* – ar įmanoma, pasirinktas dvi ar daugiau, prekes palyginti tarpusavyje.

2 lentelė. Pasirinktų egzistuojančių sistemų galimybių palyginimas

<b>WEB portalas</b>	Fortakas	BMS	Sonex	Optimalūs Kompiuteriai	Fis Kompiuteriai	Spektras	Lemita	Vebnetas	SisNeta
<b>Galimybės</b>									
Prekių medis	✓	✓	✓	✓	✓	✓	✓	✓	✓
Prekių krepšelis	✓	✓	-	-	✓	✓	-	-	✓
Galimybė susikomplektuoti suderinamą kompiuterį	-	-	-	-	✓	-	-	-	-
Paieška	✓	✓	✓	✓	✓	✓	✓	✓	✓
Detali paieška	-	✓	✓	-	-	-	-	-	✓
Kainų pokytis	-	✓	-	-	-	-	-	-	-
Kainų kitimo grafikas	-	-	-	-	-	-	-	-	-
Naujos prekė	-	✓	-	-	✓	-	-	-	-
Akcijinės prekės	-	✓	-	-	✓	-	-	-	-
Kiekis sandėlyje	✓	✓	-	-	✓	✓	✓	-	✓
Prekės aprašymas	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gamintojo aprašymas	-	✓	-	-	-	-	-	-	-
Prekių sąrašo spausdinimas	-	-	✓	-	-	-	✓	-	-
Prekių sąrašo eksportavimas	-	-	✓	-	-	✓	-	-	-
Prekių sąrašo puslapiavimas	-	-	-	-	-	-	-	-	-
Užsakymų registravimas	-	✓	-	-	✓	✓	-	-	-
Registruoti klientai	-	-	-	-	✓	✓	-	-	-
Prekių palyginimas	-	✓	-	-	-	-	-	-	-

Apžvelgus lyginamas sistemas išsiskiria du lyderiai „BMS“ ir „Fis Kompiuteriai“, nedaug nuo jų atsilieka ir „Spektras“. Kuriamos programos tikslas yra suteikti kuo daugiau galimybių vartotojui ir konkuruoti su visomis rinkoje esančiomis alternatyvomis, todėl ji turėtų realizuoti bent jau tiek pat galimybių, kiek dvi pirmaujančios sistemos. Kitos sistemos realizuoja gerokai mažiau galimybių, tačiau pažvelgę į palyginimo lentelę pastebime, kad jų realizuojamos galimybės yra beveik tokios pačios pvz.: prekių medis, paieška, kiekis sandėlyje, prekės aprašymas. Galime daryti išvadą, kad tai yra svarbiausios tokioms sistemos galimybės ir į jas reiktų atkreipti didesnę dėmesį projektavimo metu.

### 1.7 Programų sistemos kokybės vertinimo charakteristikos

Sukurtos programinės įrangos kokybės vertinimui naudosim ISO 9126 standartą [4]. Jis pateikia kokybės modelį, kuris yra pritaikomas bet kokiai programinei įrangai. Žemiau

esančiame paveikslėlyje (4 pav.) pateikti šeši pagrindiniai šį modelį sudarantys produkto kokybės kriterijai.



4 pav. ISO 9126 modelis

Pagrindinius kriterijus sudarantys subkriterijai ir jų aprašymai pateikiami žemiau esančioje lentelėje (3 lentelė).

3 lentelė. ISO 9126 modelio kriterijai ir subkriterijai

Kriterijus	Subkriterijus	Apibrėžimas
<b>Funkcionalumas</b>	Tinkamumas	Programinės įrangos atributai nurodantys esančias ir teisingas funkcijas reikalingas tam tikrai užduočiai.
	Tikslumas	Programinės įrangos atributai nurodantys gražinamus teisingus ar sutartus rezultatus, ar efektus.
	Bendradarbiavimas	Programinės įrangos atributai nurodantys galimybes sistemai sąveikauti su specifikuotom sistemom.
	Atitikimas	Programinės įrangos atributai kurie padaro programinę įrangą atitinkančią standartus, konvencijas ar įstatymus ir panašius aprašus.
	Apsauga	Programinės įrangos atributai saugantys nuo neteisėto panaudojimo ir priėjimo, nesvarbu tyčinio ar ne, prie programos ar duomenų.
<b>Patikimumas</b>	Išbaigtumas	Programinės įrangos atributai nurodantys klaidų dažnį programinėje įrangoje
	Pakantumas klaidoms	Programinės įrangos atributai nurodantys sistemos galimybes išlaikyti tam tikrą veikimo lygį įvykus klaidoms.



	Atstatomumas	Programinės įrangos atributai nurodantys galimumą atstatyti normalų veikimo darbą ir atgaminti duomenis tiesiogiai paveiktus klaidų bei laiką ir pastangas reikalingas tai atlikti.
<b>Lengvumas naudoti</b>	Suprantamumas	Programinės įrangos atributai nurodantys vartotojui reikalingas pastangas, kad suprastų loginius principus ir jų pritaikymą.
	Išmokstamumas	Programinės įrangos atributai nurodantys vartotojui reikalingas pastangas, kad išmokti naudotis programa.
	Darbingumas	Programinės įrangos atributai nurodantys vartotojo pastangas atliekant operacijas ir operacijų kontrolę.
<b>Efektyvumas</b>	Laiko naudojimas	Programinės įrangos atributai nurodantys sistemos veikimo laikus ir atliekamų operacijų skaičius per tam tikrą laiką.
	Resursų naudojimas	Programinės įrangos atributai nurodantys sunaudojamų resursų kiekį ir kiek laiko jie yra naudojami.
<b>Palaikomumas</b>	Analizuojamumas	Programinės įrangos atributai nurodantys reikalingą pastangų kiekį siekiant nustatyti netikslumus, klaidų priežastis ar identifikuoti dalis kurias reikia modifikuoti.
	Keičiamumas	Programinės įrangos atributai nurodantys pastangų kiekį reikalingą modifikavimui, klaidų pašalinimui ar aplinkos pakeitimams.
	Stabilumas	Programinės įrangos atributai nurodantys riziką apie nenumatytus efektus atliekant modifikavimą.
	Testuojamumas	Programinės įrangos atributai nurodantys pastangas reikalingas ištestuoti programinę įrangą.
<b>Pernešamumas</b>	Pritaikomumas	Programinės įrangos atributai nurodantys galimybes pritaikyti skirtingoms aplinkoms nenaudojant jokių kitokių priemonių kaip tos kurios priklauso tai programinei įrangai.
	Instaliuojamumas	Programinės įrangos atributai nurodantys pastangas reikalingas instaliuoti programinę įrangą specifikuotoje aplinkoje.
	Prisitaikymas	Programinės įrangos atributai kurie leidžia programinei įrangai laikytis standartų ir konvencijų susijusių su portabilumu.
	Pakeičiamumas	Programinės įrangos atributai nurodantys galimybes naudoti šią programinę įrangą vietoj kitos programinės įrangos tos programinės įrangos aplinkoje.

Remiantis šiais kriterijais įvertintos sistemos kokybė turėtų būti neprastesnė kaip vidutiniška.

## 1.8 Analizės išvados

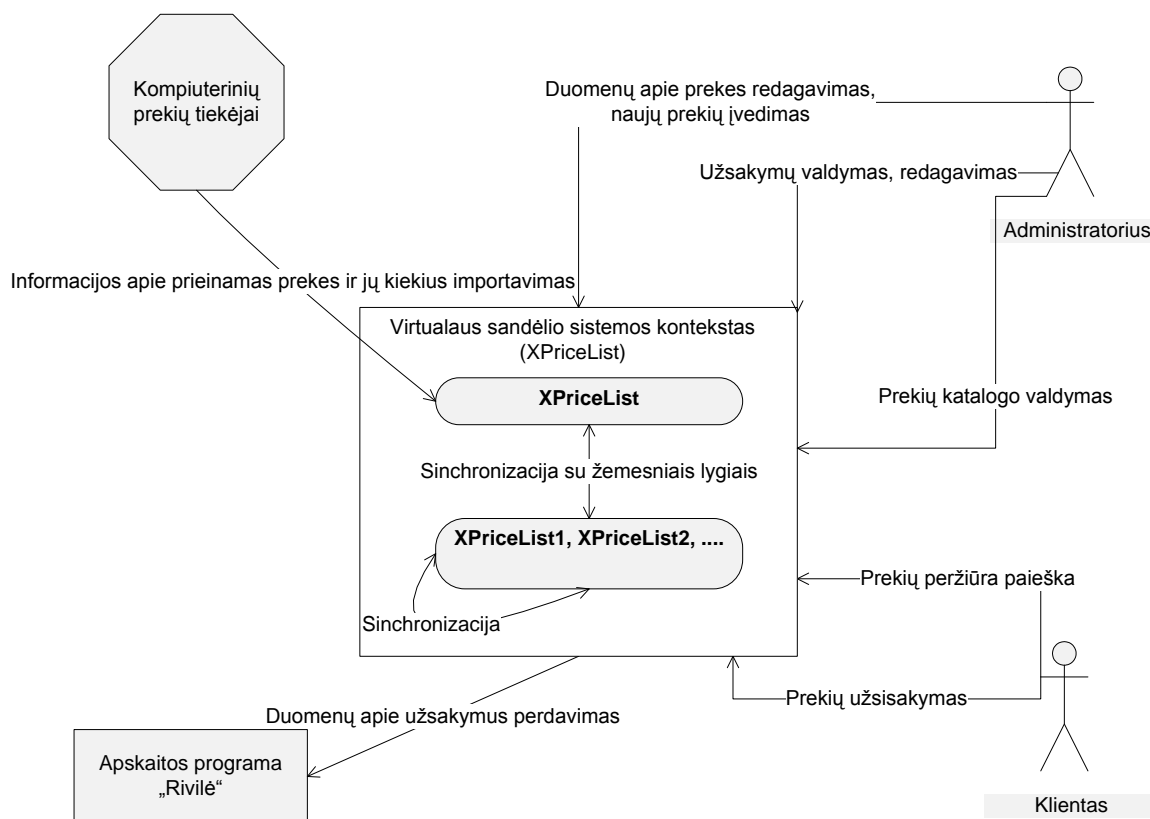
1. Apibrėžta sistemos taikymo sritis - elektroninis virtualaus sandėlio sprendimas įmonėms, kurios prekiauja kompiuterine ir programine įranga.
2. Išanalizavus vartotojo iškeltus pradinis reikalavimus sistemai, nustatyti penki probleminiai sistemos realizavimo uždaviniai: duomenų integravimas iš įvairių tiekėjų, sinchronizacija tarp įmonių grupės serverių, prekių komplektavimas-suderinamumas, antkainių valdymas, integracija su apskaitos programa „Rivilė“. Visiems jiems pateikti galimi sprendimo variantai.
3. Palyginus tris, WEB sistemų realizavimui skirtas ir plačiausiai naudojamas platformas (PHP, Java, .NET), taip pat atsižvelgus į sistemai keliamus reikalavimus ir numatomus poreikius jų realizavimui pasirinkta .NET sistemos kūrimo platforma.
4. Atlikta egzistuojančių sistemų lyginamoji analizė. Pateiktoje sistemų palyginimo lentelėje nustatytos dažniausiai realizuojamos funkcijos: prekių medis, paieška, prekių kiekis sandėlyje, prekės aprašymas.
5. Apžvelgti esminiai akcentai susiję su sistemos architektūros kūrimu, saugumo ir patikimumo užtikrinimu.
6. Apibrėžti kuriamos sistemos kokybės vertinimo reikalavimai, jie remiamasi tarptautinio ISO 9126 standarto rekomendacijomis.

## 2. INTEGRUOTOS VIRTUALAUS SANDĖLIO SISTEMOS REALIZACIJA

Šiame skyriuje nagrinėjamos techninės projekto detalės. Konkretizuojami priimti sprendimai, pateikiamos architektūros, modulių ir duomenų vaizdų diagramos.

Pirmuose skyriuose trumpai aptariami priimti techniniai sprendimai. Vėliau apžvelgiamas bendras sistemos architektūros vaizdas, tuo pačiu pateikiamos esminės sistemos architektūrą paaškinančios schemas. Galiausiai pateikiami pagrindiniai sistemos duomenų vaizdai.

### 2.1 Veiklos kontekstas



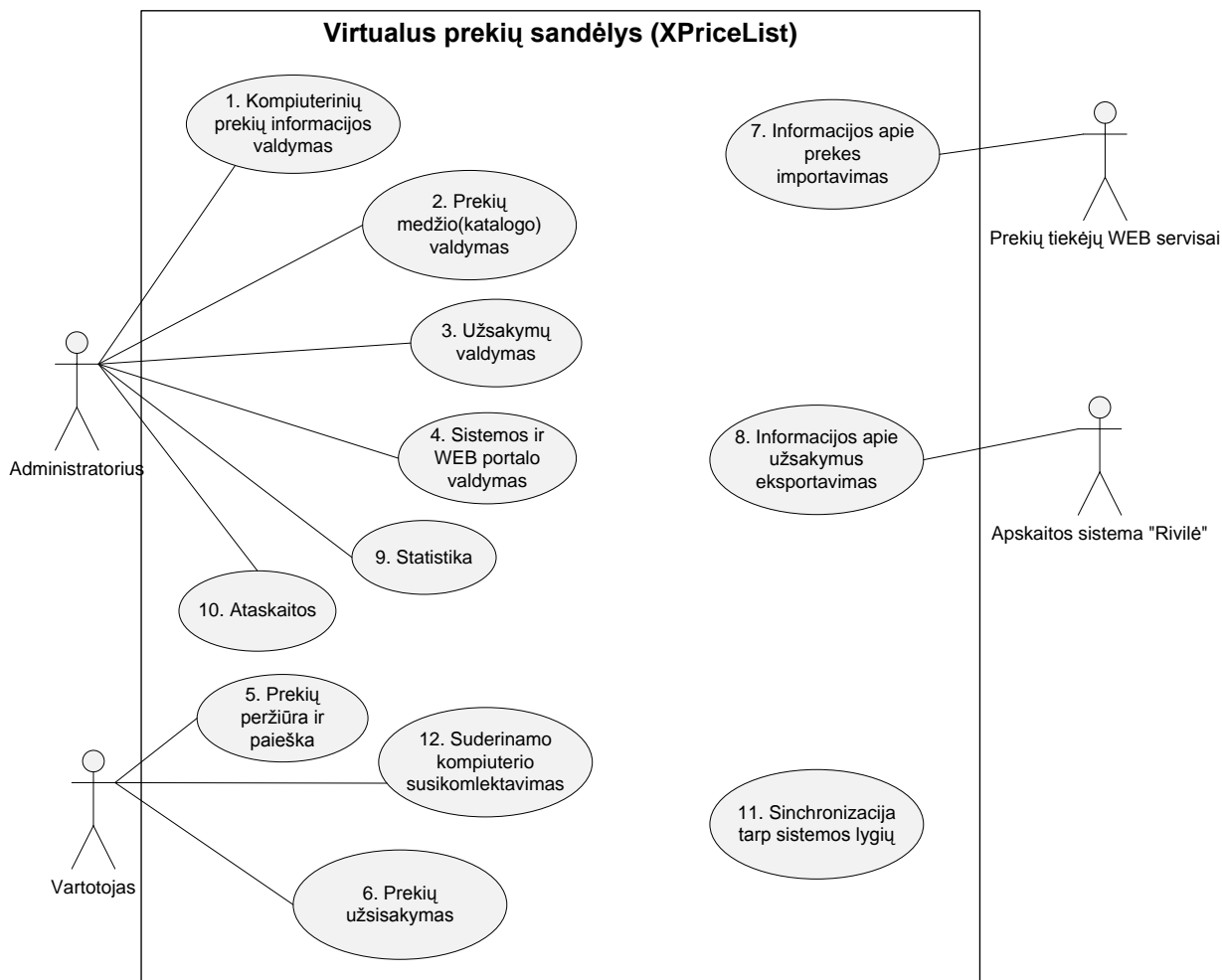
5 pav. Sistemos veiklos kontekstas

Aukščiau esančiame paveiksliuke (5 pav.), pateikiamas sistemos veiklos kontekstas. Čia matome sistemos veikimo metu vykstančius duomenų mainus ir pagrindinius jos vartotojus.

Vidurinėje paveiksliuko dalyje, sistema yra išskaidyta į kelias dalis (XPriceList, XPriceList1...). Tuo siekiama parodyti, kad sistema dažniausiai neegzistuos kaip atskiras vienetas, bet bus tokių pačių sistemų grupės dalis, kurios tarpusavyje reguliariai keičiasi duomenimis.

## 2.2 Panaudojimo atvejai

Nagrinėjant sistemos veiklos kontekstą, bei surinktus sistemos reikalavimus, buvo sudaryta žemiau pateikta sistemos panaudojimo atvejų diagrama (6 pav.). Šiame darbe pateikiami tik aukščiausio lygio panaudos atvejai. Kiekvieno iš jų detalesnis nagrinėjimas nepateikiamas.



6 pav. Sistemos panaudojimo atvejų diagrama

Panagrinėję diagramą, lengvai išskiriame penkias pagrindines sistemos panaudojamumo grupes arba modulius (4 lentelė).

4 lentelė. Sistemos panaudojamumo grupės

Grupė	Sudarantys panaudos atvejai
<i>Sistemos administravimas</i>	1. Kompiuterinių prekių informacijos valdymas; 2. Prekių medžio(katalogo) valdymas; 3. Užsakymų valdymas; 4. Sistemos ir WEB portalo valdymas; 9. Statistika; 10. Ataskaitos.
<i>Sistemos teikiamų paslaugų naudojimas</i>	5. Prekių peržiūra ir paieška; 6. Prekių užsisakymas; 12. Suderinamo kompiuterio susikomplektavimas;
<i>Prekių informacijos importavimas</i>	7. Informacijos apie prekes importavimas.

<i>Užsakymų informacijos eksportavimas</i>	8. Informacijos apie prekes eksportavimas.
<i>Sinchronizacija</i>	11. Sinchronizacija tarp sistemos lygių.

### 2.3 Funkciniai reikalavimai

Sistemos reikalavimai buvo renkami naudojantis *Volere* šablonu. Žemiau pateikiami pagrindinių surinktų funkcinių reikalavimų pavadinimai:

1. Klientui, prekės turi būti rodomos medžio forma;
2. Klientas turi galėti susidėti prekes į krepšelį;
3. Klientas turi galėti valdyti krepšelio turinį;
4. Klientas turi galėti užsisakyti į krepšelį sudėtas prekes;
5. Administratorius turi galėti valdyti sistemai pateiktus užsakymus;
6. Klientas turi galėti pats susikomplektuoti suderinamą kompiuterį iš atskirų dalių;
7. Kliento susikomplektuotas kompiuteris turi būti traktuojamas kaip atskiras prekės vienetas;
8. Administratorius turi galėti valdyti iš kokių prekių ir kaip gali būti komplektuojamas suderinamas kompiuteris;
9. Klientas turi galėti atlikti detalią paiešką tarp parduodamų prekių;
10. Klientas ir administratorius turi turėti galimybę matyti kainų pokyčius ir jų grafiką;
11. Klientas turi matyti kurios prekės yra naujos arba akcijinės;
12. Administratorius turi galėti valdyti, kurios prekės yra naujos arba akcijinės;
13. Klientas turi galėti matyti, kiek tam tikros prekės vienetų šiuo metu yra sandėlyje;
14. Klientas turi galėti matyti detalų bet kurios prekės aprašymą;
15. Klientas turi galėti matyti bet kurios prekės gamintojo aprašymą;
16. Klientas turi galėti atsispausdinti išsifiltruotą prekių sąrašą;
17. Klientas turi turėti galimybę išsieksporuoti norimą priekių sąrašą į „Excel“ dokumentą;
18. Sistema turi leisti registruoti klientus;
19. Sistema turi leisti registruotiems klientams prisijungti per WEB puslapio dali;
20. Sistema turi atsiminti įvairią registruotų klientų informaciją;
21. Sistema turi galėti suteikti nuolaidą tam tikriems registruotiems klientams;
22. Sistemos valdymas turi būti leidžiamas tik prisijungus prie specialaus puslapio;
23. Sistemoje turi būti galimybė valdyti administratorių teises;
24. Sistema turi importuoti duomenis iš įvairių tiekėjų;
25. Sistema turi apjungti visus importuotus prekių duomenis į vientisą sistemą;
26. Sistema turi rodyti pigiausią prekę iš visų galimų tokių pačių prekių;

27. Sistema turi leisti nustatyti standartinį antkainį visoms prekėms;
28. Sistema turi leisti nustatyti tam tikrą antkainį vienai konkrečiai prekių šakai;
29. Sistema turi leisti nustatyti prekių antkainių režius priklausomai nuo kainos dydžio;
30. Sistema turi leisti sumuoti antkainius priklausomai nuo prekės padėties prekių medyje;
31. Sistema turi leisti suskirstyti prekes į grupes;
32. Sistema turi leisti sudėlioti prekių grupes į katalogą(medį);
33. Sistema neturi rodyti klientui prekių nepriskirtų tam tikrai grupei;
34. Sistema turi leisti įvesti prekes ir administratoriui, t.y. ne tik importuoti iš tiekėjų;
35. Turi būti rankinio prekių importavimo galimybė iš tam tikro formato failo;
36. Sistema turi galėti sinchronizuoti informaciją su žemesniais savo lygiais;
37. Sistema turi atskirti informaciją iš aukštesnių lygių nuo žemesniuose lygiuose įvestos;
38. Sistema turi galėti leisti nurodyti prekės atvežimo datą jei prekės šiuo metu nėra sandėlyje;
39. Sistema turi leisti administratoriui valdyti užsakymų būsenas t.y. gautas, vykdomas ir pan;
40. Sistema turi leisti siųsti automatinius pranešimus klientui ir administratoriui apie užsakymo sukūrimą arba jo būsenos pasikeitimus;
41. Sistema turi leisti administratoriui valdyti WEB puslapio dalį t.y. sukurti naujus puslapius, keisti informaciją ir pan.;
42. Sistema turi galėti suformuoti įvairias ataskaitas apie šiuo metu turimas kompiuterines prekes;
43. Sistema turi galėti pateikti įvairias ataskaitas apie klientų pateiktus užsakymus;
44. Sistema turi leisti reguliariais laiko intervalais eksportuoti duomenis apie užsakymus į apskaitos sistemą „Rivilė“.

## 2.4 Nefunkciniai reikalavimai

### a) Reikalavimai sistemos išvaizdai

45. Lengvai skaitoma ir suprantama sąsaja;
46. Neįkyri sąsaja. Sistema neturi po kelis kartus klausti to paties dalyko;
47. Sistema turi teikti kontekstinę pagalbą.

### b) Reikalavimai panaudojamumui

48. Sistema turi būti panaudojama be išankstinio apmokymo (80% sėkmingas panaudojimas pirmu bandymu);
49. Sistemoje turi būti naudojama nacionalinė(lietuvių) kalba.

c) *Reikalavimai vykdymo charakteristikoms*

50. Sistema turi aptarnauti iki 10 administratorių vienu metu;

51. Sistema turi aptarnauti iki 50 klientų vienu metu.

d) *Reikalavimai veikimo sąlygoms*

52. Sistema turi veikti visose populiariose interneto naršyklėse.

e) *Reikalavimai sistemos priežiūrai*

53. Sistemos generuojamų dokumentų pritaikymas pasikeitusiems reikalavimams per vieną mėnesį.

f) *Reikalavimai saugumui*

54. Sistema turi garantuoti saugų vartotojo prisijungimą;

55. Kiekvienas prisijungęs vartotojas mato tik savo duomenis.

g) *Kultūriniai-politiniai reikalavimai*

56. Sistema neturi įžeisti jokių religinių ar etninių grupių.

h) *Teisiniai reikalavimai*

57. Sistema turi veikti pagal Lietuvos respublikoje galiojančius įstatymus.

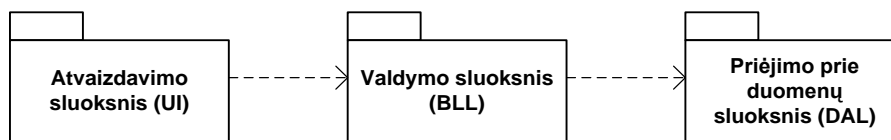
## 2.5 Programinio sprendimo architektūros specifikacija

### 2.5.1 Bendras architektūros vaizdas

Internetinių sistemų projektavimui yra naudojamas trijų sluoksnių architektūrinis modelis [13]. Šiai sistemai taip pat panaudotas šis modelis.

Daugiasluoksnių projektavimo principas yra tas, kad sistema sudalinama į posistemas pagal tam tikrus loginius sluoksnius. Programa, kuri vadovaujasi tokia architektūra turi puslapius, kurie atsako tik už vaizdinį pateikimą. Šie puslapiai, savo ruožtu, kreipiasi į valdymo (angl. *Business Logic Layer – BLL*) sluoksnį, kuris atsako už programos veiksmų logiką. Pastarasis kreipiasi į priėjimo prie duomenų sluoksnį (angl. *Data Access Layer – DAL*) tam, kad gautų reikiamus duomenis, ir grąžina visą informaciją atgal į atvaizdavimo sluoksnį (angl. *User Interface - UI*).

Principinis sistemos sudalinimas į sluoksnius pateiktas toliau esančiame paveiksliuke (7 pav.).



7 pav. Projekto valdymo sistemos išskaidymas į sluoksnius

### 2.5.1.1 Atvaizdavimo lygis (UI)

**Apibrėžimas.** Realizuota sistema yra sukurta naudojantis internetine sąsaja, todėl vartotojo sąsaja tai internetinių puslapių rinkinys. Šie puslapiai pateikia vartotojui sistemoje esančią informaciją bei sistemos teikiamas funkcijas. *Atvaizdavimo sluoksnis* yra atsakingas už bendrą atvaizdavimo struktūrą. Jis formuoja ir pateikia visą sistemos informaciją vartotojui.

**Atsakomybės.** *Atvaizdavimo sluoksnis* (UI) atsakingas už puslapių generavimą ir dinamiškai generuojamą turinį. Tai yra jis, bendradarbiaudamas su kitais lygiais, gauna tam tikrą informaciją arba duomenis ir pagal suprogramuotą struktūrą sugeneruoja informacijos pateikimo formą.

Kita šio lygio funkcija yra apdoroti gaunamą iš vartotojo informaciją (rasti įvestus duomenis ir t.t.) ir perduoti ją *Valdymo sluoksniui* (BLL).

### 2.5.1.2 Valdymo sluoksnis (BLL)

**Apibrėžimas.** Visos funkcijos, kurios turi būti realizuotos programoje, turi tam tikrus loginius scenarijus ir tam tikrus apribojimus arba vykdymo sąlygas. Visą šį veiklos priklausomumą galima vadinti veiklos taisyklėmis. *Valdymo sluoksnis* kaip tik ir yra atsakingas už visą veiklos logiką realizuotą programoje. Šis sluoksnis yra valdantis sluoksnis.

**Atsakomybės.** Čia yra aprašoma didžioji dalis programos veiklos logikos. Veiklos logika apima:

- a) visus reikalingus skaičiavimus;
- b) darbo srautų valdymą (taip pat sesijos duomenų stebėjimą ir valdymą);
- c) visų *Atvaizdavimo sluoksnio* duomenų užklausų apdorojimą.

### 2.5.1.3 Priėjimo prie duomenų sluoksnis (DAL)

**Apibrėžimas.** Priėjimo prie duomenų modulio paskirtis yra aprūpinti *Valdymo sluoksnį* reikalingais duomenimis bei rašyti iš jo gaunamus duomenis į duomenų bazę. Kitaip tariant, šis sluoksnis yra tarpininkas tarp fizinės duomenų bazės ir programos, o konkrečiu atveju *Valdymo sluoksnio*.



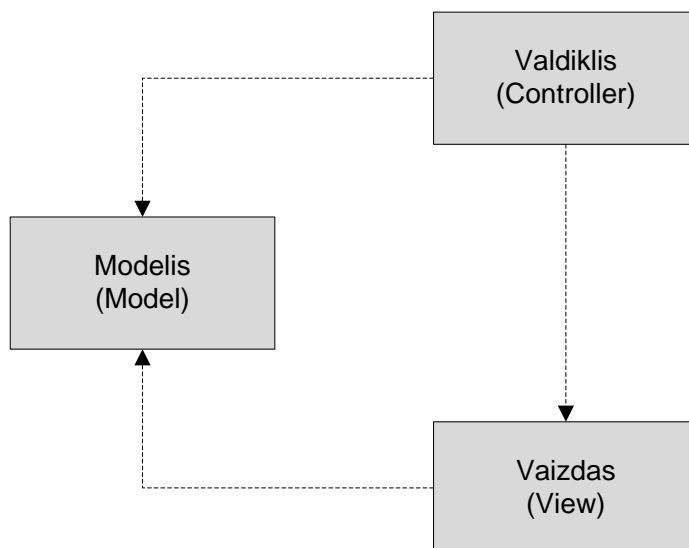
**Atsakomybės.** Šis sluoksnis atsako už ryšį su duomenų baze. Užtikrina ryšio palaikymą, nutraukimą ir duomenų gavimą iš duomenų bazės.

Šiame sluoksnyje yra tik minimalūs patikrinimai ar veiklos logikos elementai, kadangi visą tai turi užtikrinti *Valdymo sluoksnis*. Čia išlieka tik tokie veiklos logikos elementai, kaip:

- a) neegzistuojančio objekto patikrinimas;
- b) išorinio rakto ryšio apribojimai.

### 2.5.2 Modelis-Vaizdas-Valdiklis šablonas

Beveik visi atvaizdavimo modulio puslapiai realizuoti naudojantis *Modelis-Vaizdas-Valdiklis* (angl. *Model-View-Controller* arba MVC) programavimo šablonu (8 pav.).



8 pav. Modelis-Vaizdas-Valdiklis šablono diagrama

Šis šablonas naudojamas sistemose su sudėtingom vartotojo sąsajom, kurios gali naudoti ir atvaizduoti tuos pačiu duomenų masyvus skirtingais būdais ir skirtingais vaizdais. Taip pat šis šablonas tinkamas griežtam atvaizdavimo sluoksnio atskyrimui nuo valdymo sluoksnio [14].

*Modelis(Model)* – valdo sistemos duomenų elgesį. Atsako į užklausas apie savo būseną (dažniausiai iš *Vaizdo* klasės) ir atsako į instrukcijas pakeisti dabartinę būseną (dažniausiai iš *Valdiklio* klasės)

*Vaizdas(View)* – valdo ir atvaizduoja informaciją.

*Valdiklis(Controller)* – interpretuoja vartotojo atliekamas pelės ir klaviatūros įvestis (gaudo įvykius), informuoja *Modelio* ir *Vaizdo* klases apie pasikeitimus kai to reikia [15].

## Privalumai

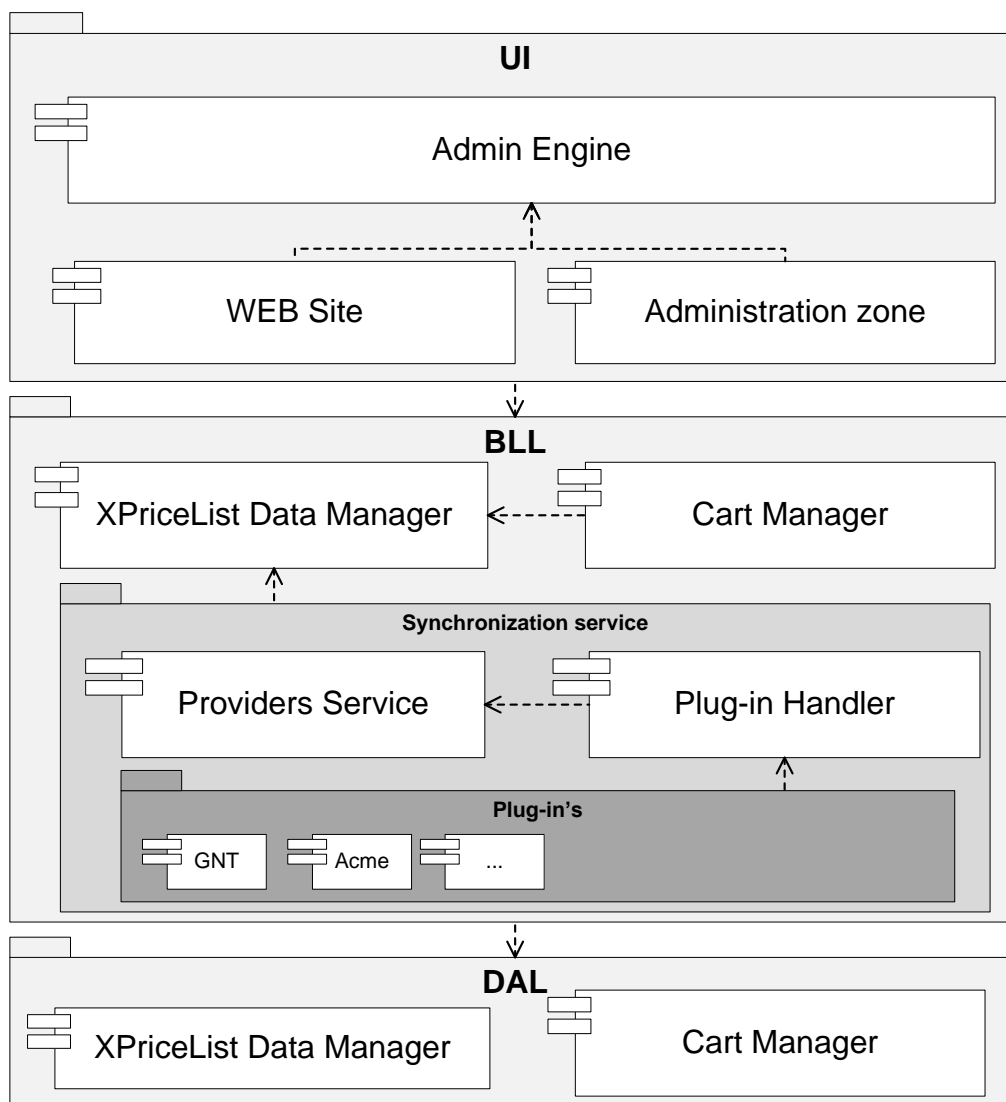
- a) *Palaiko keletą vaizdų.* Kadangi *Vaizdas* yra atskirtas nuo *Modelio* ir nėra jokios tiesioginės priklausomybės tarp šių klasių, vartotojo sąsaja gali rodyti keletą *Vaizdų* vienu metu iš tų pačių duomenų.
- b) *Prisiderinimas prie pakeitimų.* Vartotojo sąsaja yra linkusi keistis daug dažniau nei biznio logikos taisyklės. Vartotojai gali norėti kitokios spalvos, šrifto, išdėstymo ar naujų įtaisų palaikymo pvz.: mobiliųjų telefonų ar kišeninių kompiuterių. Dėl savo nepriklausomumo šis šablonas leidžia nekeičiant *Modelio* papildyti sistemą naujais *Vaizdais* [15].

## Trūkumai

- a) *Kompleksiškumas.* Šis šablonas įneša papildoma netiesiogiškumo lygį, kuris padidina bendrą sistemos sudėtingumą. Taip pat padidina įvykiais pagrįsto vartotojo sąsajos kodo kiekį, kuris yra sunkiau trasuojamas.
- b) *Dažnų atsinaujinimų kaina.* *Modelio* ir *Vaizdo* atskyrimas nereiškia kad programuotojas gali ignoruoti *Vaizdų* klases. Pavyzdžiui jei modelis kažkuriuo metu labai dažnai keičiasi, tai jis gali užversti *Vaizdą* pranešimais apie pasikeitimus. O jei *Vaizdas* yra grafiškai sudėtingas, tai jo atvaizdavimas gali užtrukti, to pasekoje *Vaizdo* rodoma būseną gali nebeatitikti tikrosios būsenos. Programuotojas turi įvertinti tokius dalykus ir galbūt grupuoti siunčiamus pranešimus *Vaizdo* klasėms [15].

### 2.5.3 Detalus architektūros vaizdas

Detalus programos architektūros vaizdas trumpai apžvelgia realizuotas klases ir jų struktūrą. Klasių aprašymai suskirstyti pagal sistemos modulius. Žemiau pateiktoje diagramoje (9 pav.) pavaizduoti visi pagrindiniai sistemos moduliai, jų grupavimas į loginius paketus ir ryšiai tarp jų.



9 pav. Sistemos modulių diagrama

### 2.5.3.1 „Admin Engine“ modulio architektūra

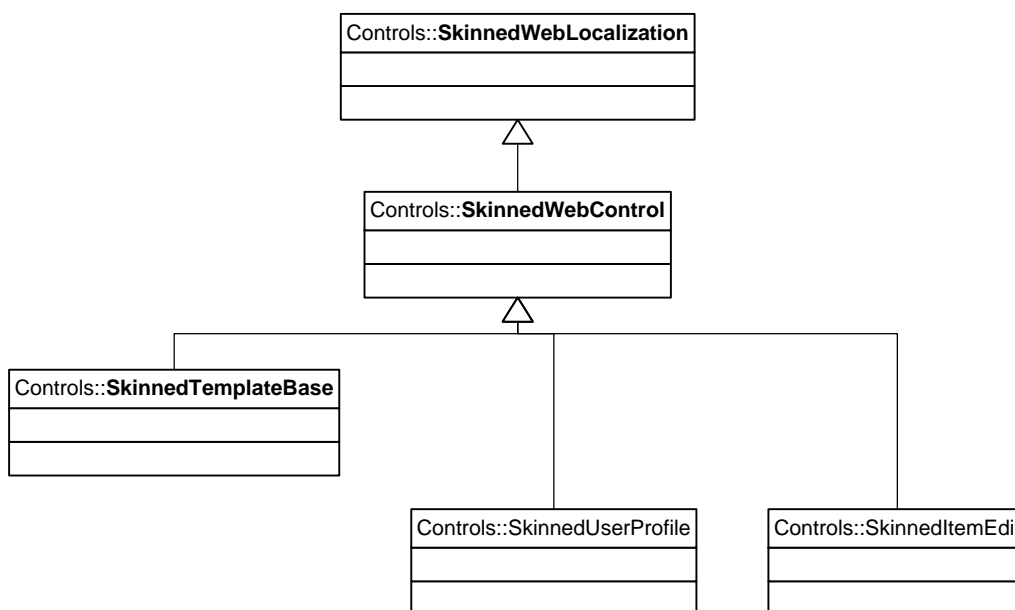
„Admin Engine“ modulis skirtas standartinių vartotojo sąsajos duomenų vaizdų kūrimo palengvinimui ir standartizavimui. Šiame modulyje valdomos sistemos palaikomos kalbos, valiutos ir panašūs parametrai.

Taip pat, šiame modulyje realizuotas specialus vartotojo sąsajos kūrimo varikliukas. Šis varikliukas leidžia automatiškai sugeneruoti vartotojo sąsaja ir jos pagalba atlikti visus standartinius duomenų apdorojimo veiksmus (įterpimą, paiešką, šalinimą, redagavimą). Tam užtenka paveldėjus specialią klasę joje nurodyti tik duomenų lentelės pavadinimą.

Žemiau pateiktos pagrindinės šio modulio klasės (10 pav.):

- SkinnedWebLocalization* – klasė skirta lokalizacijos valdymui. Pagal tai, kokia kalba pasirinkta, užkrauna atitinkamus XML kalbų failus ir realizuoja interfeisą jų panaudojimui;

- b) *SkinnedWebControl* – klasė standartizuojanti visus sistemoje naudojamus webControl'us. Realizuoja metodus ir parametrus sistemoje naudojamam atvaizdavimo varikliui;
- c) *SkinnedTemplateBase* – speciali vartotojo sąsajos generavimo klasė. Ją paveldinčioms klasėms užtenka specifiuoti naudojamos duomenų lentelės pavadinimą, o šita klasė sugeneruoja standartinę vartotojo sąsają;
- d) *SkinnedUserProfile*, *SkinnedItemEdit* – nestandartinės vartotojo sąsajos realizavimo klasės. Jos paveldi tik iš *SkinnedWebControl* klasės. Tokios klasės kuriamos tada, kai neužtenka standartinės generuojamos sąsajos teikiamų galimybių.



10 pav. „Admin Engine“ modulio klasių diagrama

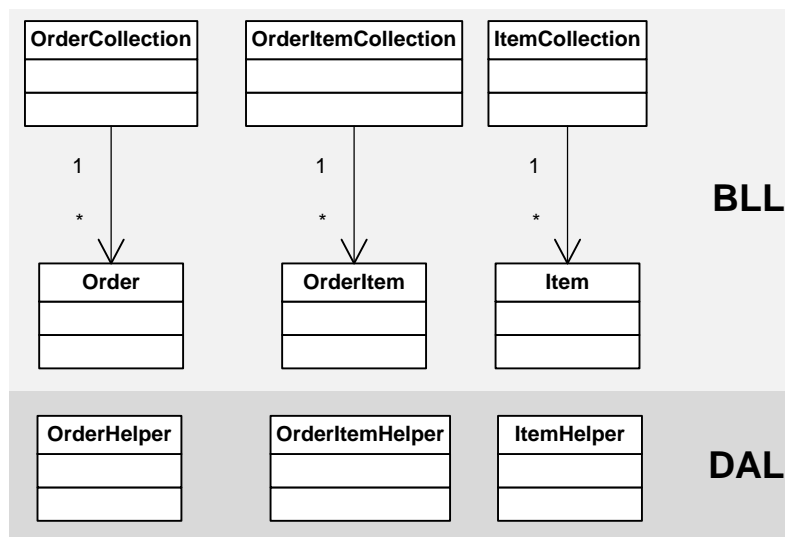
### 2.5.3.2 „XPriceList Data Manager“ modulio architektūra

„XPriceList Data Manager“ modulis skirtas sistemos bendravimui su duomenų baze. Šis modulis realizuoja du sluoksnius BLL ir DAL.

Žemiau pateiktame paveiksliuke (11 pav.) atvaizduojamos kelios pagrindinės šio modulio klasės:

- a) *Order*, *OrderItem*, *Item* – klasės realizuojančios tam tikrą sistemos objektą ir visą su juos susijusią valdymo logiką. (*Order* – užsakymas, *OrderItem* – viena užsakyta prekė, *Item* – prekė);

- b) *OrderCollection*, *OrderItemCollection*, *ItemCollection* – klasės realizuojančios atitinkamų klasių kolekcijas. Taip pat, čia realizuojami metodai veiksams su tomis kolekcijom;
- c) *OrderHelper*, *OrderItemHelper*, *ItemHelper* – klasės realizuojančios atitinkamų objektų veiksmus su duomenų bazėje esančia informacija.



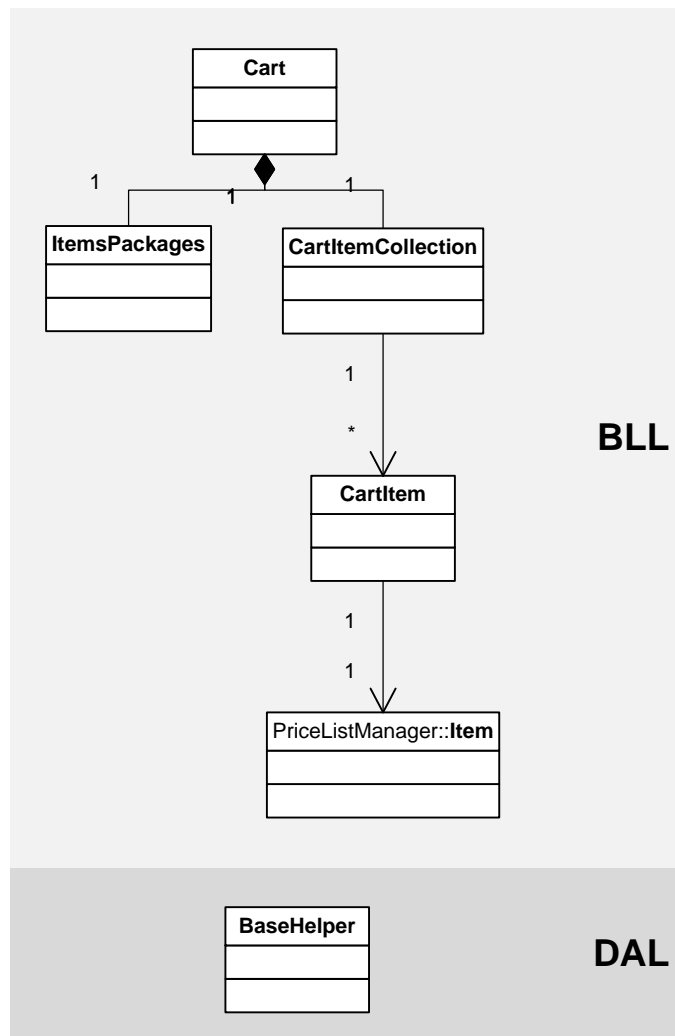
11 pav. „XPriceList Data Manager“ modulio klasių diagrama

### 2.5.3.3 „Cart Manager“ modulio architektūra

„Cart Manager“ specialus modulis skirtas prekių krepšelio realizavimui ir apdorojimui. Šis modulis apima du sluoksnius BLL ir DAL.

Žemiau pateiktame paveiksliuke (12 pav.) atvaizduojamos kelios pagrindinės šio modulio klasės:

- Cart* – pagrindinė krepšelio realizacijos klasė. Joje realizuoti įvairūs kainos, antkainių ir pan. apskaičiavimo metodai;
- ItemsPackages* – kompozicinė *Cart* klasės klasė. Šioje klasėje saugomos sukomplektuotos prekės t.y. paties vartotojo susirinkti kompiuteriai naudojantis suderinamumo varikliuku;
- CartItemCollection* – kompozicinė *Cart* klasės klasė. Šioje klasėje saugoma vartotojo pasirinktų, atskirų t.y. nesukomplektuotu prekių kolekcija;
- CartItem* – viena konkreti vartotojo pasirinkta prekė su papildoma užsakymo išsaugojimui reikalinga informacija;
- Item* – viena konkreti prekė;
- BaseHelper* – speciali klasė bendravimui su duomenų baze.



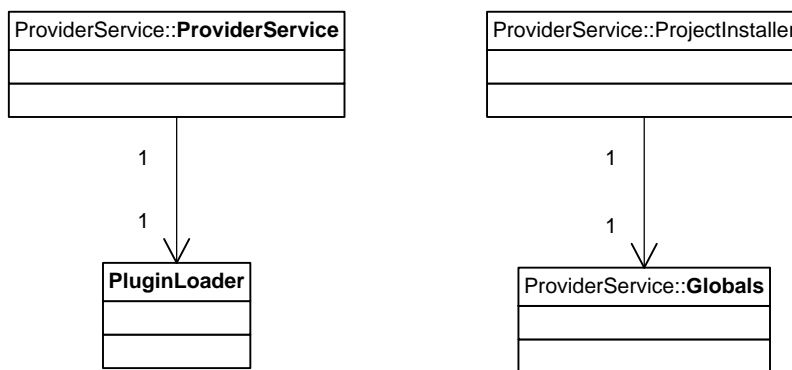
12 pav. „Cart Manager“ modulio klasių diagrama

#### 2.5.3.4 „Providers Service“ modulio architektūra

„Providers Service“ modulis realizuojantis sistemos servisą skirtą duomenų apie prekes importavimui iš tiekėjų WEB servisų. Šis servisas naudoja Plug-In modelį t.y. kiekvienam atskiram tiekėjo WEB servisui yra realizuojamas atskiras plug-in'as.

Žemiau pateiktame paveiksliuke (13 pav.) atvaizduojamos kelios pagrindinės šio modulio klasės:

- ProviderService* – pagrindinė serviso klasė, joje pradedamas ir baigiamas serviso veikimas;
- PluginLoader* – speciali klasė tiekėjų plug-in'ų užkrovimui ir paleidimui;
- ProjectInstaller* – klasė naudojama serviso instaliavimui;
- Globals* – klasė kurioje surašomi serviso naudojami duomenys pvz.: serviso pavadinimas, klaidų pranešimų talpinimo vieta ir kita.



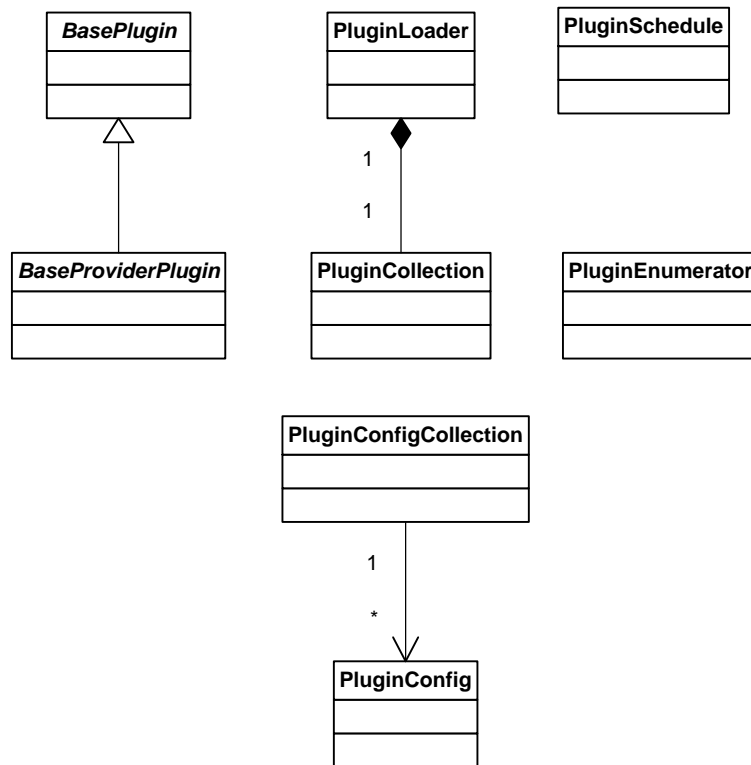
13 pav. „Providers Service“ modulio klasių diagrama

### 2.5.3.5 „Plug-in Handler“ modulio architektūra

„Plug-in Handler“ modulis skirtas tiekėjų plug-in'ų suradimui, atpažinimui ir užkrovimui. Jis, taip pat, nuskaito užkrautų plug-in'ų nustatymų failus.

Žemiau pateiktame paveiksliuke (14 pav.) atvaizduojamos kelios pagrindinės šio modulio klasės:

- a) *BasePlugin* – klasė realizuojanti interfeisą reikalingą užkraunamiems plug-in'ams;
- b) *BaseProviderPlugin* – klasė realizuojanti standartines funkcijas duomenų parsisiuntimui ir importavimui iš tiekėjų WEB servisų;
- c) *PluginLoader* – speciali klasė tiekėjų plug-in'ų užkrovimui ir paleidimui;
- d) *PluginCollection* – tiekėjų plug-in'ų kolekcija;
- e) *PluginSchedule* – speciali klasė, kuri pagal nustatymų failą tam tikrais laiko intervalais paleidinėja plug-in'us;
- f) *PluginEnumerator* – klasė užkraunamų plug-in'ų failų cikliškam perėjimui;
- g) *PluginConfig* – klasė užkraunanti tam tikro plug-in'o nustatymus;
- h) *PluginConfigCollection* – nustatymų klasių kolekcija.



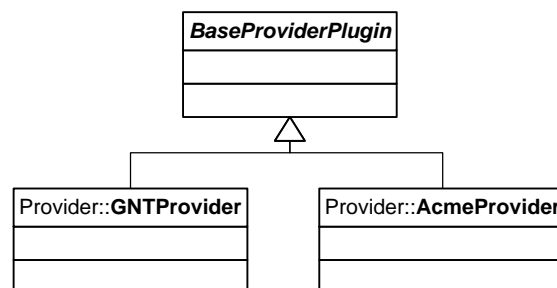
14 pav. „Plug-in Handler“ modulio klasių diagrama

### 2.5.3.6 „Plug-in‘s“ modulių architektūra

„Plug-in‘s“ moduliai realizuoja duomenų parsisiuntimą ir importavimą į sistemą priderintą prie kiekvieno konkretaus tiekėjo WEB serviso.

Žemiau pateiktame paveiksliuke (15 pav.) atvaizduojamos kelios pagrindinės šio modulio klasės:

- BaseProviderPlugin* – klasė realizuojanti standartines funkcijas duomenų parsisiuntimui ir importavimui iš tiekėjų WEB servisų;
- GNTProvider*, *AcmeProvider* – klasės adaptuotos prie konkrečių tiekėjų WEB servisų. Perdengia reikiamus standartinius *BaseProviderPlugin* klasės metodus.



15 pav. „Plug-in‘s“ modulių klasių diagrama



## 2.5.4 Pagrindiniai duomenų vaizdai

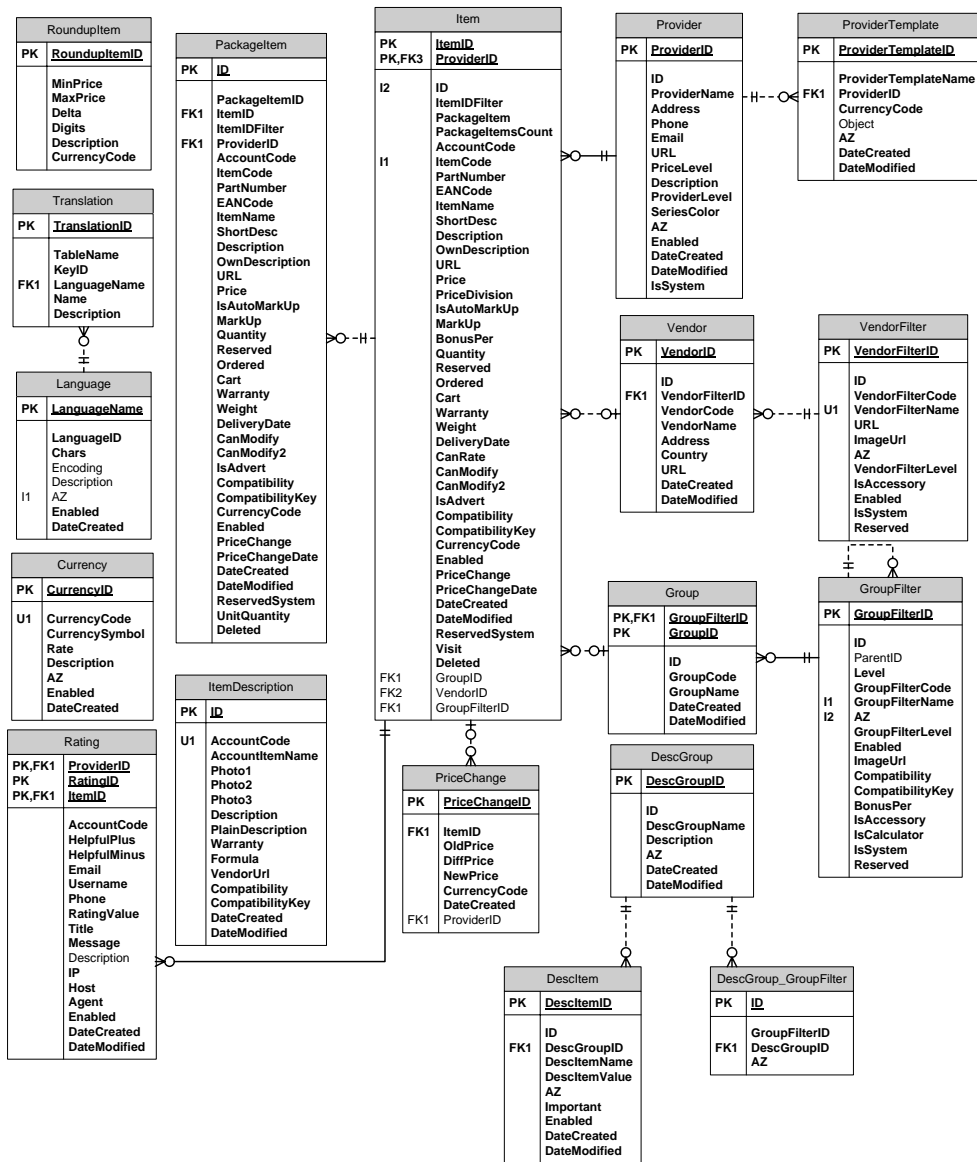
Žemiau esančiuose skyreliuose aprašytos pagrindinės duomenų bazės lentelės.

### 2.5.4.1 Kompiuterinių prekių saugojimo lentelės

Žemiau esančiame paveiksle (16 pav.) pavaizduotos lentelės saugo visą informaciją susijusia su kompiuterinėmis prekėmis ir jų atvaizdavimu klientams.

Lentelių aprašymai:

- a) *Item* – saugoma informacija apie kompiuterines prekes;
- b) *PackageItem* – saugoma informacija apie pagal apskaitos kodą sugrupuotas vienodas prekes;
- c) *Provider* – saugoma informacija apie tiekėjus;
- d) *ProviderTemplate* – serializuotame objekte saugoma informacija apie tiekėjo nustatymus;
- e) *Vendor* – saugoma informacija apie gamintojus;
- f) *VendorFilter* – saugoma informacija apie filtrus pagal kuriuos grupuojami gamintojai gauti iš tiekėjų siunčiamų duomenų;
- g) *Group* – saugoma informacija apie prekių grupes;
- h) *GroupFilter* – saugoma informacija apie filtrus pagal kuriuos grupuojamos grupės tada kai prekės rodomos klientui;
- i) *PriceChange* – saugoma informacija apie prekės kainų pasikeitimus;
- j) *Rating* – saugoma informacija apie klientų prekės įvertinimus;
- k) *ItemDescription* – saugoma informacija apie papildomus detalius prekės aprašymus;
- l) *DescGroup* – saugoma informacija apie grupių aprašymus;
- m) *DescItem* – saugoma informacija apie konkrečios grupės konkrečios prekės aprašymą;
- n) *DescGroup\_Filter* – saugoma informacija apie aprašymo grupių filtravimą;
- o) *RoundupItem* – saugoma informacija apie apvalinimo taisykles kainoms gautoms pritaikius antkainius;
- p) *Language* – saugoma informacija apie sistemoje naudojamas kalbas;
- q) *Translation* – saugoma informacija apie vertimus įvairiomis kalbomis tam tikriems lentelių laukams;
- r) *Currency* – saugoma informacija apie sistemoje naudojamas valiutas.



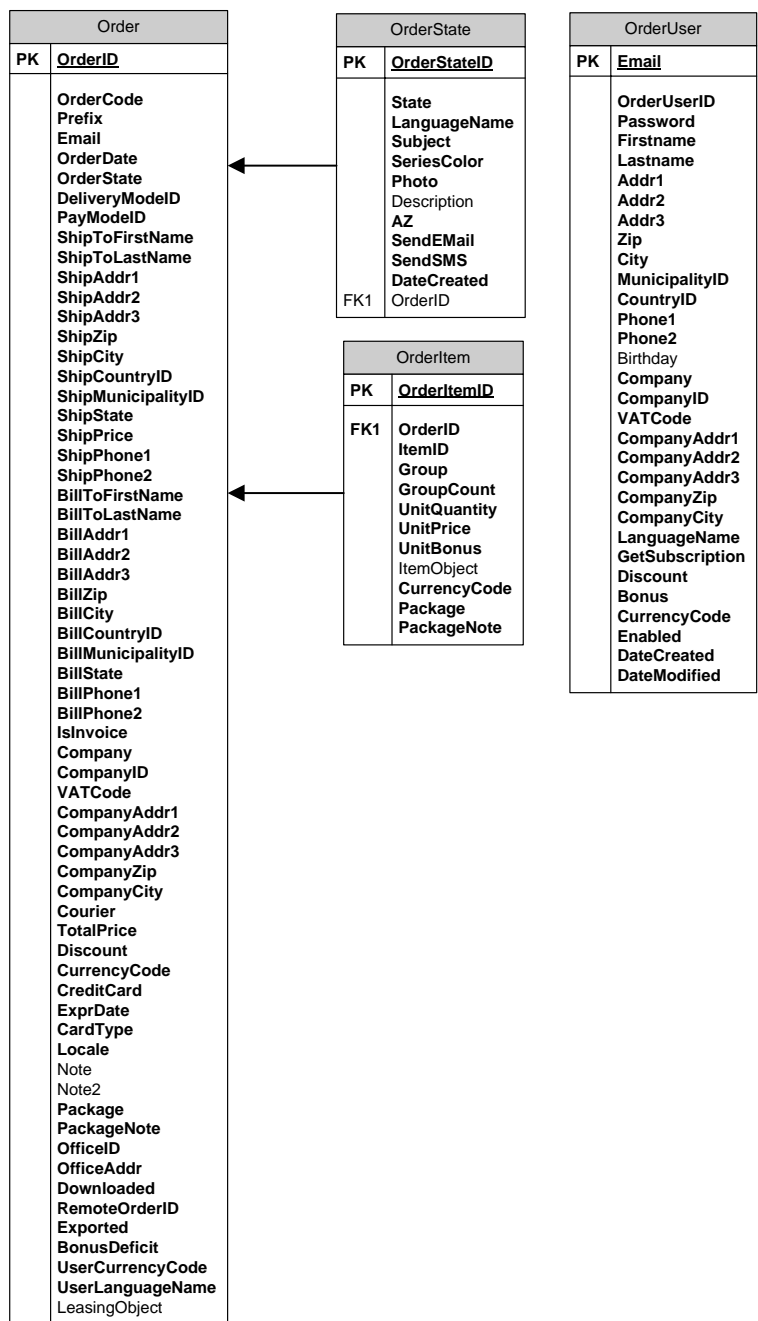
16 pav. Kompiuterinių prekių saugojimo lentelių diagrama

### 2.5.4.2 Užsakymų saugojimo lentelės

Žemiau esančiame paveiksle (17 pav.) pavaizduotos lentelės saugo visą informaciją susijusia su klientų užsakymų išsaugojimu.

Lentelių aprašymai:

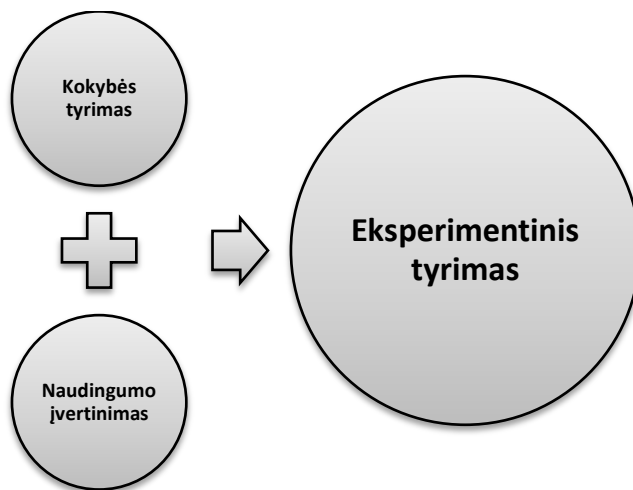
- Order* – saugoma informacija apie konkrečius užsakymus;
- OrderState* – saugoma informacija apie galimas užsakymų būsenas;
- OrderItem* – saugoma informacija apie konkrečiam užsakymui priklausančias prekes;
- OrderUser* – saugoma informacija apie registruotus vartotojus, kurie gali pateikti užsakymus.



17 pav. Užsakymų saugojimo lentelių diagrama

### 3. EKSPERIMENTINIS SISTEMOS KOKYBĖS IR NAUDINGUMO TYRIMAS

Šiame skyriuje aprašomas atliktas virtualaus prekių sandėlio sistemos eksperimentinis tyrimas ir gauti rezultatai.



18 pav. Eksperimentinio tyrimo aspektai

Kaip parodyta aukščiau pateiktame paveiksliuke (18 pav.), programų sistemos eksperimentinis tyrimas bus atliekamas dviem aspektais: pagal analitinėje dalyje pateiktus kokybės kriterijus ir atliekant realizuotos sistemos naudingumo įvertinimą.

#### 3.1 Sistemos kokybės tyrimas

Integruoto įvairių tiekėjų virtualaus sandėlio sistemos kokybės vertinimas buvo atliekamas pagal 1.7 skyriuje apibrėžtus ISO 9126 standarto kriterijus. Kokybės subkriterijų vertinimo parametrai apibrėžti remiantis sistemai iškeltais reikalavimais, o vertinimas atliekamas praėjus mėnesiui po sistemos įdiegimo pas užsakovą. Stebint vartotojų darbą su sistema, juos apklausiant ir analizuojant įvairią statistinę sistemos registruojamą informaciją.

Kiekvienas kokybės subkriterijus susideda iš vieno ar daugiau jį nusakančių parametru. Kiekvienas parametras įvertintas dešimtbalėje skalėje: 1 – kokybė labai bloga arba toks parametras išvis nerealizuotas, 10 – kokybė labai gera.

Žemiau pateiktoje lentelėje (5 lentelė) pateikiami virtualaus sandėlio sistemos kokybės vertinimo rezultatai. Kiekvienas kokybės kriterijus įvertinamas pagal visus jį sudarančius subkriterijus paskui apskaičiuojant bendrą kriterijaus įvertinimą.

5 lentelė. Virtualaus sandėlio kokybės vertinimas pagal ISO 9216 standartą

<b>Funkcionalumas</b>				
	<i>Subkriterijus</i>	<i>Subkriterijaus parametru aprašymai</i>	<i>Įvertinimas</i>	
	Tinkamumas	Sistema realizuoja visus nurodytus pradinis vartotojo reikalavimus sistemai.	10	
		Sistema realizuoja visus iš vartotojo surinktus funkcinius reikalavimus.	9	
		Sistema realizuoja visus iš vartotojo surinktus nefunkcinius reikalavimus.	10	
	Tikslumas	Sistemos atliekami nuolaidų skaičiavimai pagal leidžiamas įvesti formules yra teisingi.	9	
		Įvairūs sistemos atliekami kainų skaičiavimai yra teisingi.	9	
	Bendradarbiavimas	Sistema realizuoja duomenų importavimą iš visų prekių tiekėjų WEB servisų.	8	
		Sistema realizuoja duomenų perdavimą apskaitos sistemai „Rivilė“	10	
	Atitikimas	Sistemos formuojamos ataskaitos atitinka įstatymų numatytus reikalavimus	7	
		Sistema užtikrina įstatymiškai privalomą asmeninių duomenų saugumą/neplatinimą	9	
	Apsauga	Administraciniai sistemos vartotojai turi prisijungimo vardus ir slaptažodžius	7	
		Sistemos DB apsaugota slaptažodžiu	10	
		Visi sistemos atliekami duomenų persiuntimai per WEB servisus atliekami tik po sėkmingos vartotojo identifikacijos	10	
		Vartotojų slaptažodžiai yra koduojami	4	
	<i>Bendras</i>			8,6
	<b>Patikimumas</b>			
		<i>Subkriterijus</i>	<i>Subkriterijaus parametru aprašymai</i>	<i>Įvertinimas</i>
	Išbaigtumas	Ne daugiau kaip viena klaida per 7 dienas	5	
	Pakantumas klaidoms	Visos klaidos registruojamos ir apie jas pranešama vartotojui	7	
		Klaida kuriame nors sistemos modulyje arba vartotojo sąsajos formoje neapriboja galimybių naudotis likusia sistemos dalimi	9	
	Atstatomumas	Įvykus klaidai sistema neleidžia koreguoti duomenų. Įrašomi tik pilni duomenų paketai. Palaikomos transakcijos.	5	
<i>Bendras</i>			6,5	
<b>Lengvumas naudoti</b>				
	<i>Subkriterijus</i>	<i>Subkriterijaus parametru aprašymai</i>	<i>Įvertinimas</i>	
	Suprantamumas	Vartotojo sąsaja nenaudoja jokių neįprastų formų ar būdų. Tie patys vartotojo sąsajos valdymo principai išlaikomi visoje sistemoje.	8	
		Kur reikia pateikiami paaiškinimai pačioje	5	

		vartotojo sąsajoje.	
Išmokstamumas		Vartotojas susipažįsta ir išmoksta naudotis pagrindinėmis sistemos funkcijomis per vieną dieną (8 valandas).	7
		Sistema turi vadovą pradedančiajam.	7
		Sistema turi detalių vartotojo vadovą.	8
Darbingumas		Sistemoje naudojamas lengvai suprantamas, intuityvus ir pastovus vartotojo sąsajos grafinis dizainas.	8
		Vartotojui pateikiami pranešami apie visus jo atliekamus veiksmus. Ar jie buvo sėkmingi ar nesėkmingi ir kodėl.	8
<i>Bendras</i>			7,3
<b>Efektyvumas</b>			
	<i>Subkriterijus</i>	<i>Subkriterijaus parametru aprašymai</i>	<i>Įvertinimas</i>
Laiko naudojimas		Duomenų importavimas iš tiekėjų serverių užtrunka 30min.	7
		Duomenų sinchronizavimas tarp firmų serverių užtrunka 15min.	7
		Įprastinės vartotojo operacijos neužtrunka ilgiau kaip 5 sekundes.	8
Resursų naudojimas		Laikas kai sistema 100% apkrauna jai paskirtą WEB serverį neviršija 1 valandos per parą.	8
		Duomenų bazės dydis nedidėja eksponentiškai su kiekvienu duomenų importavimu iš tiekėjų ir sinchronizavimusi tarp skirtingų serverių.	9
<i>Bendras</i>			7,8
<b>Palaikomumas</b>			
	<i>Subkriterijus</i>	<i>Subkriterijaus parametru aprašymai</i>	<i>Įvertinimas</i>
Analizuojamumas		Prie visų klaidų pranešimų pateikiama informacija kurioje vietoje ji įvyko.	6
		Visos sistemos funkcijos yra tiksliai dokumentuotos. Viena funkcija vykdoma viename modulyje.	5
Keičiamumas		Norint atlikti pakeitimus sistemoje nereikia būti susipažinus su visa sistemos realizacija.	4
		Daugelį sistemos veikimo parametrus galima keisti per parametrus. Nereikalaujant sistemos perkompiliavimo.	8
Stabilumas		Pakeitimai viename sistemos modulyje neiššaukia pakeitimų ar klaidų kitame.	3
Testuojamumas		Automatiniai sistemos testai apima didžiąją dalį sistemos funkcijų.	2
		Sistema nereikalauja specialios aplinkos sukonfigūravimo ir paleidimo prieš pradedant jos testavimą.	4
<i>Bendras</i>			4,5

<b>Pernešamumas</b>			
	<i>Subkriterijus</i>	<i>Subkriterijaus parametrų aprašymai</i>	<i>Įvertinimas</i>
	Pritaikomumas	Sistema gali dirbti Windows (98, 2000, XP, 2003, Vista) operacinėse sistemose.	9
		Sistema gali būti naudojama Linux/Unix tipo operacinėse sistemose.	1
	Instaliuojamumas	Lengvas sistemos WEB dalies diegimas	5
		Lengvas sistemos servisų diegimas	9
		Lengvas sistemos parametrų konfigūravimas	7
	Prisitaikymas	Naudojami standartiniai protokolai duomenų mainams sistemos viduje ir su tiekėjų WEB servisais.	10
	Pakeičiamumas	Sistema turi didžiąją dalį funkcinių galimybių pasitaikančių alternatyviuose sprendimuose todėl gali būti naudojama vietoje jų.	9
<i>Bendras</i>			7,1

Aukščiau pateiktoje lentelėje (5 lentelė) kokybės vertinimas susideda iš 6 pagrindinių kriterijų ir 21 subkriterijaus. Kiekvienas subkriterijus nusakomas vieno ar daugiau parametro. Vieno konkretaus subkriterijaus reikšmė apskaičiuojama pagal žemiau pateiktą (1) formulę:

$$SBK = \frac{\sum pr}{pk} \quad (1)$$

kur  $SBK$  – subkriterijus,

$pr$  – parametro reikšmė,

$pk$  – parametrų kiekis.

Kiekvienas atskiras kokybės kriterijaus įvertinimas randamas pasinaudojant 2-a formule:

$$K = \frac{\sum sk}{skk} \quad (2)$$

kur  $K$  – kriterijus,

$sk$  – subkriterijus,

$skk$  – subkriterijų kiekis.

Bendras sistemos kokybės įvertinimas gaunamas pasinaudojant 3-čia formule:

$$BI = \frac{\sum K}{kk} \quad (3)$$

kur  $BI$  – bendras įvertis,

$K$  – kriterijus,

*kk* – kriterijų kiekis.

Atlikus visus skaičiavimus gautas bendras sistemos kokybės įvertis lygus 6,9. Gautas rezultatas rodo, kad sistema yra vidutiniškos kokybės.

Pažvelgus į atskirus kiekvieno kokybės kriterijaus rezultatus pastebime, kad blogiausiai įvertintas sistemos palaikomumo parametras. Pagrindinės problema yra ta, kad beveik nėra automatinė sistemos modulių testų. O kadangi parametro „Pakeitimai viename sistemos modulyje neiššaukia pakeitimų ar klaidų kitame“ įvertinimas irgi nėra geras, tai bet kokie pakeitimai sistemoje, padaryti mažiau su ja susipažinusių žmonių, gali iššaukti kortų kaladės griūties efektą. Kita vertus funkcionalumo parametras, kuris nusako sistemos atitikimą vartotojų lūkesčiams ir pateiktiems parametrams, buvo įvertintas net 8,6 balo.

### 3.2 Realizuotos sistemos naudingumo įvertinimas

Šiame skyrelyje atliekamas realizuotos programų sistemos (pavadintos *xPriceList*) klientinės WEB dalies teikiamo naudingumo įvertinimas. Vertinimas buvo atliktas lyginant realizuotą sistemą su 1.6.5 skyrelyje analizuotomis egzistuojančiomis sistemomis.

Programos naudingumo įvertis remiasi programos galimybių įvertinimu. Kiekvienos programos realizuotos galimybės įtaka naudingumui buvo skaičiuojama remiantis svertiniu naudingumo įverčiu, kuris buvo gautas atlikus sistemos vartotojų apklausą (PRIEDAS 2: Apklausa).

Žemiau pateikiama lentelė (6 lentelė), kurioje palyginamas egzistuojančių sistemų naudingumas su realizuotos sistemos (*xPriceList*) naudingumu.

6 lentelė. Egzistuojančių sistemų naudingumo palyginimas su realizuota sistema (*xPriceList*)

<i>Galimybės</i>	Fortakas	BMS	Sonex	Optimalūs Kompiuteriai	Fis Kompiuteriai	Spektras	Lemita	Vebsites	SisNeta	<i>xPriceList</i>	Naudingumo įverčiai
Prekių medis											10
Prekių krepšelis	✓	✓	–	–	✓	✓	–	–	✓	✓	8
Galimybė sukomplektuoti suderinamą kompiuterį	–	–	–	–	✓	–	–	–	–	✓	7
Paieška	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10
Detali paieška	–	✓	✓	–	–	–	–	–	✓	–	6
Kainų pokytis	–	✓	–	–	–	–	–	–	–	✓	7



Kainų kitimo grafikas	–	–	–	–	–	–	–	–	–	✓	<b>0</b>
Naujos prekė	–	✓	–	–	✓	–	–	–	–	✓	<b>7</b>
Akcijinės prekės	–	✓	–	–	✓	–	–	–	–	✓	<b>8</b>
Kiekis sandėlyje	✓	✓	–	–	✓	✓	✓	–	✓	✓	<b>9</b>
Prekės aprašymas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	<b>10</b>
Gamintojo aprašymas	–	✓	–	–	–	–	–	–	–	✓	<b>5</b>
Prekių sąrašo spausdinimas	–	–	✓	–	–	–	✓	–	–	✓	<b>8</b>
Prekių sąrašo eksportavimas	–	–	✓	–	–	✓	–	–	–	–	<b>1</b>
Prekių sąrašo puslapiavimas	–	–	–	–	–	–	–	–	–	✓	<b>0</b>
Užsakymų registravimas	–	✓	–	–	✓	✓	–	–	–	✓	<b>7</b>
Registruoti klientai	–	–	–	–	✓	✓	–	–	–	✓	<b>6</b>
Prekių palyginimas	–	✓	–	–	–	–	–	–	–	–	<b>9</b>
<b>Apskaičiuoti duomenys</b>											
<i>Galimybių skaičius</i>	5	12	6	3	10	8	5	3	6	16	<b>18</b>
<i>Svorinis galimybių įvertis</i>	47	96	45	30	82	61	47	30	53	102	<b>118</b>
<i>Naudingumas (%)</i>	39	81	38	25	69	51	39	25	44	86	<b>100</b>
<i>Realizuotos galimybės (%)</i>	27	66	33	16	55	44	27	16	33	88	–

Žemiau pateikiamos, lentelės (6 lentelė) apatinės dalies („Apskaičiuoti duomenys“), skaičiavimo formulės.

#### Skaičiavimo formulės:

1. *Galimybių skaičius* – tai realizuotų galimybių skaičius. Ties kiekviena galimybę prie atitinkamo WEB portalo rašoma „–“ nerealizuota, arba „✓“ realizuota. „✓“ suma ir sudarys galimybių skaičių.
2. *Svorinis galimybių įvertis* – prie kiekvienos galimybės yra nurodytas naudingumo įvertis nustatytas iš vartotojų apklausos.

$$SGI = \sum (w \cdot c),$$

kur  $SGI$  – svorinis galimybių įvertis,

$w$  – naudingumo įvertis,

$c$  – galimybės realizavimo reikšmė („–“ arba „✓“).

3. *Naudingumas* – naudingumas parodo procentinį realizuotų galimybių naudingumą lyginant su maksimaliai įvertintu naudingumu.

$$N = \frac{SGI}{PN} \cdot 100\%,$$

kur  $N$  – naudingumas,

$SGI$  – svorinis galimybių įvertis,

$$PN = \sum w - \text{maksimalus svorinis galimybių įvertis} = 118.$$

4. *Realizuotos galimybės* – parodo procentinį realizuotų galimybių skaičių lyginant su maksimaliu galimybių skaičiumi.

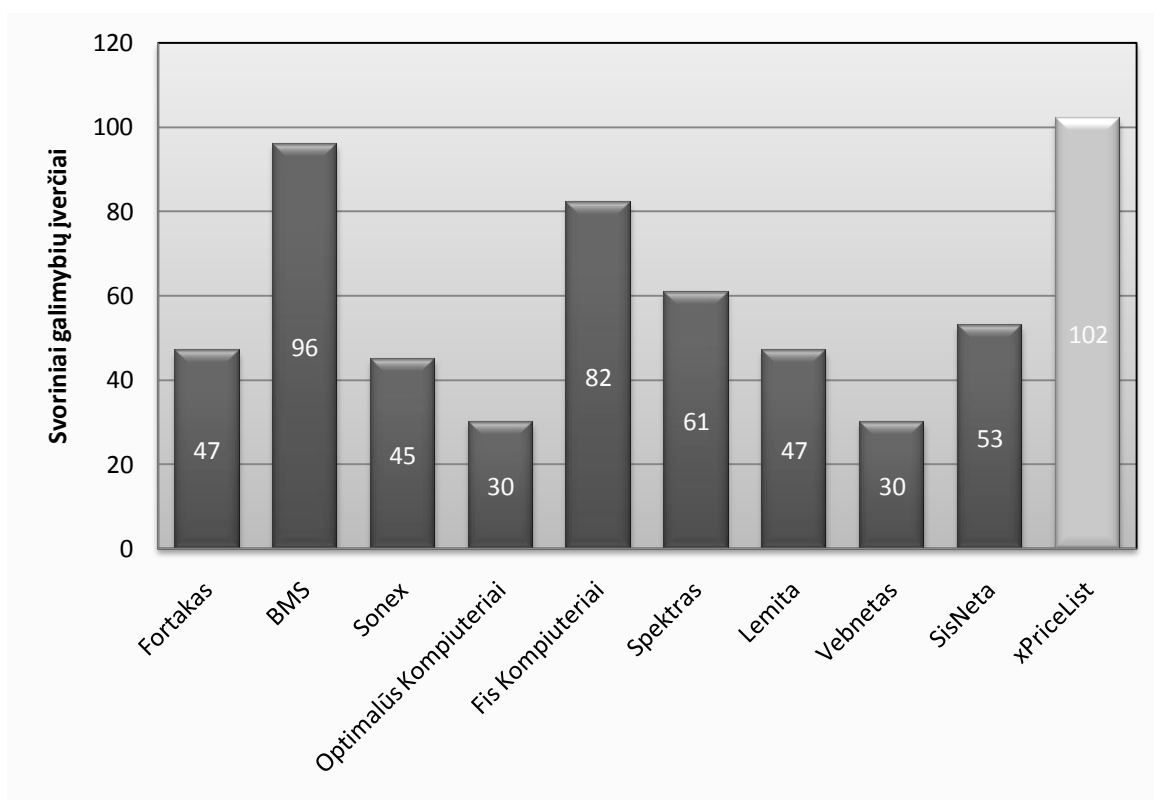
$$RG = \frac{GS}{VG} \cdot 100\%,$$

kur  $RG$  – realizuotos galimybės,

$GS$  – galimybių skaičius,

$$VG = \sum g - \text{viena galimybė, išviso galimybių yra 18.}$$

Žemiau pateiktame grafike (19 pav.), parodomi visų analizuojamų WEB portalų surinkti svoriniai galimybių įverčiai.

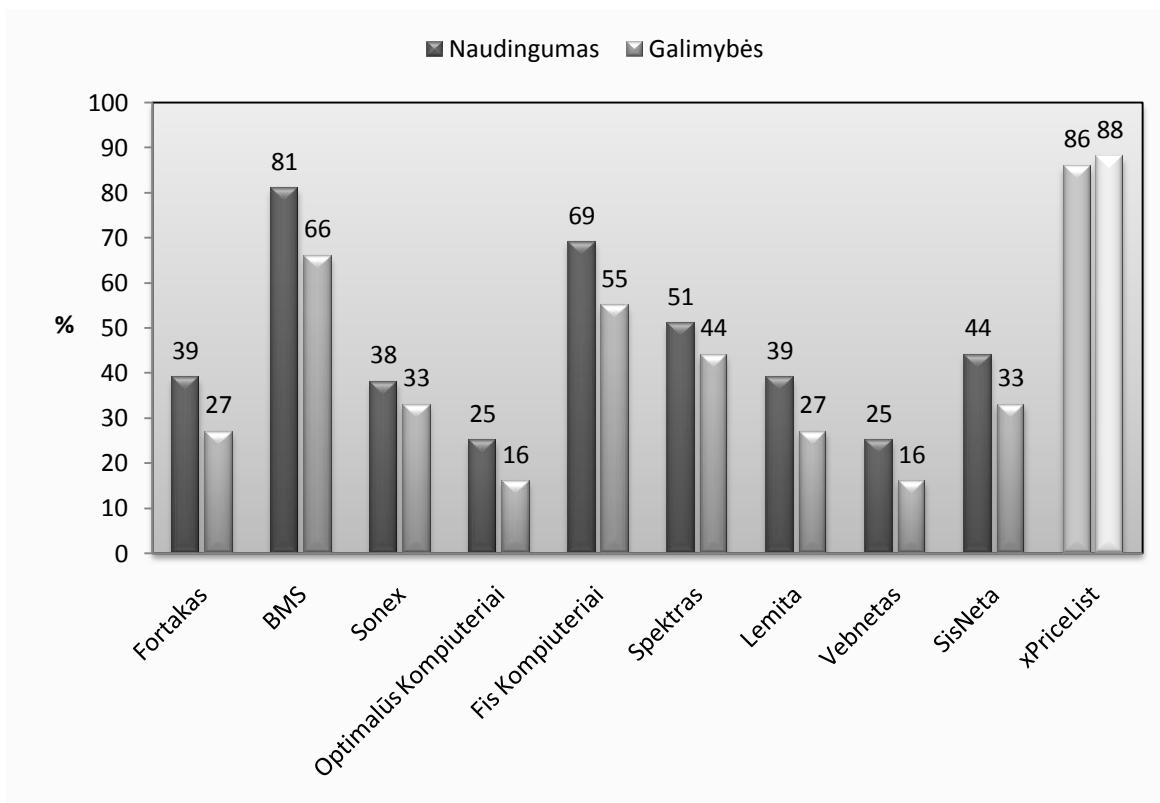


19 pav. Naudingumo svorinių įverčių palyginimas

Pateiktame grafike (19 pav.) matome, kad mūsų realizuota sistema turi didžiausią įvertį (102 taškai). Panašų įvertį turi tik dvi sistemos: *BMS* (96) ir *Fis kompiuteriai* (82). Visos kitos nagrinėtos sistemos turi gerokai mažesnius įvertinimus. Tačiau šis grafikas visiškai

neįvertina realizuotų galimybių skaičiaus ir nepalygina svorinių įverčių taškų su maksimaliai galimomis reikšmėmis.

Žemiau pateikiamas grafikas (20 pav.) kuriame atvaizduojamas palyginimas tarp procentinio realizuoto sistemų naudingumo ir procentinio realizuotų galimybių skaičiaus.



20 pav. Naudingumo ir realizuotų galimybių procentinis palyginimas

Pateiktame grafike matome, kad iš visų nagrinėtų sistemų, mūsų realizuota sistema suteikia didžiausią naudingumo procentą (86%) ir taip pat realizuoja daugiausiai nagrinėtų galimybių (86%). Artimiausios konkurentės, kaip ir pagal anksčiau pateiktą grafiką, išlieka *BMS* ir *Fis Kompiuteriai*.

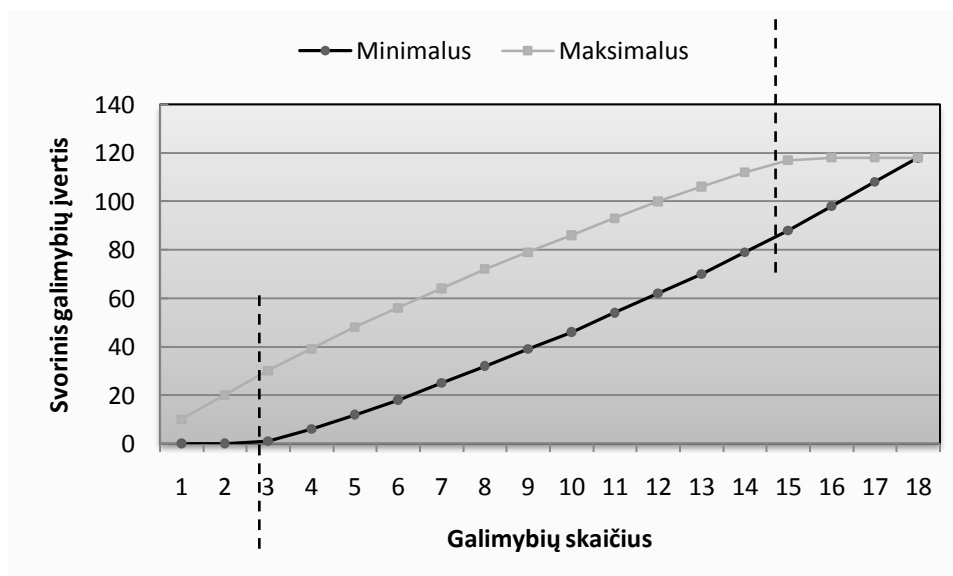
Reikia pastebėti, kad *BMS* sistema suteikia beveik vienodą naudingumo procentą 81% vietoj 86%, realizuodama beveik trečdaliu mažiau galimybių 66% vietoj 88%. Taip pat pastebime, kad vienintelės mūsų sistemos realizuotų galimybių skaičiaus procentas viršija sistemos teikiamą naudingumo procentą. Kad būtų aiškiau kodėl taip yra sudarysime realizuotų galimybių ir jų teikiamų svorinių naudingumo įverčių priklausomybių grafiką.

Grafiko sudarymui naudojame žemiau pateiktą lentelę (7 lentelė). Šioje lentelėje atvaizduoti du duomenų masyvai: vienas jų nurodo mažiausius įmanomus svorinius galimybių naudingumo įverčius kitas didžiausius įmanomus svorinius galimybių naudingumo įverčius priklausomai nuo galimybių kiekio.

7 lentelė. Galimybių skaičiaus ir svorinio naudingumo įverčio priklausomybė

Galimybių skaičius	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Maž. įmanomas svorinis įvertis	0	0	1	6	12	18	25	32	39	46	54	62	70	79	88	98	108	118
Didž. įmanomas svorinis įvertis	10	20	30	39	48	56	64	72	79	86	93	100	106	112	117	118	118	118

Pagal lentelės (7 lentelė) duomenis nubraižome žemiau pateiktą realizuotų funkcijų ir svorinių naudingumo įverčių grafiką (21 pav.).



21 pav. Realizuotų funkcijų ir svorinių naudingumo įverčių grafikas

Žiūrėdami į 21 paveiksliuke pateikto grafiko maksimalių įverčių kreivę matome, kad nuo tam tikro taško papildomai realizuotos galimybės beveik nepadidina bendro sistemos naudingumo. Būtent toks, kai kurių galimybių, mažas įtakos turėjimas bendram sistemos naudingumui, paaiškina kodėl BMS sistema realizuodama beveik trečdaliu mažiau galimybių pasiekia beveik tokį patį naudingumą. BMS sistema realizuoja tik pačias naudingiausias galimybes, kai tuo tarpu mūsų realizuota *xPriceList* sistema realizuoja ir ne tokias naudingas savybes. Tokių, mažai naudos nešančių, galimybių realizavimas nėra teigiamas dalykas, nes bet koks papildomas sistemos funkcionalumas įneša papildomą sudėtingumą ir klaidų tikimybę, o jei funkcija nenaudinga, tai papildomas sudėtingumas neatneša jokios finansinės naudos. Tolesniuose sistemos tobulinimo etapuose reikėtų anksčiau ir tiksliau įvertinti projektuojamų sistemos galimybių naudingumą.

### 3.3 Tolimesnio sistemos tobulinimo, plėtojimo galimybės

Stebint vartotojų darbą su sistema, taip pat iš jų pačių atsiliepimų buvo išskirtos tokias prioritetinės tobulinimo galimybės:

*Prekių suderinamumo įvedimo vartotojo sąsajos tobulinimas.* Daugeliui vartotojų, dabartinė tekstinių kodų principu paremta suderinamumo valdymo vartotojo sąsaja, pasirodė per daug paini ir sunkiai įsisavinama. Patobulinimo tikslas būtų sukurti grafinę informacijos suvedimo posistemę. Tokiu būdu vartotojams nereikėtų įsisavinti įvairių suderinamumui naudojamų kodų. Taip pat, reiktų atlikti analizę ar neįmanoma realizuoti dalinę suderinamumo informacijos suvedimo automatizaciją pagal tam tikrus raktinius prekių aprašymų žodžius.

*Pasirinktų prekių palyginimas.* Artėjant šio projekto realizavimo pabaigos terminui užsakovai pateikė dar vieną patildomą reikalavimą sistemai: „Pasirinktų prekių savybių palyginimas tarpusavyje“. Tačiau tai nebuvo įtraukta į pradinę sutartį ir naujų galimybių realizavimas būtų nukėlęs sistemos pridavimą, todėl buvo sutarta tai atidėti tolimesniems patobulinimo etapams.

*Programos offline versija.* Įdiegus sistemą ir kompanijų vadybininkams pradėjus naudotis sistema paaiškėjo, kad vertėtų sukurti specializuotą windowsinę sistemos sąsają (atskirą programą), kuri leistų vadybininkams sparčiau dirbti su sistema ir nebūtų priklausoma nuo internetinio ryšio.

*Servisų monitoringas.* Sukurtoje virtualaus sandėlio sistemoje daugelis veiksmų atliekami automatizuotai ir reguliariais laiko intervalais, tokie kaip duomenų importavimas iš tiekėjų WEB servisų, duomenų sinchronizacija tarp vienos kompanijos serverių ir pan. Tačiau kartais įvyksta nenumatyti sutrikimai, kurie kartais netgi nėra tikros sistemos klaidos bet gali sąlygoti neteisingą sistemos veikimą arba savalaikį duomenų neatnaujinimą. Tokių problemų, kuo greitesniam, aptikimui ir neutralizavimui buvo iškelta idėja sukurti automatizuotą procesų monitoringo įrankį. Jame būtų fiksuojami visi sistemos atliekami veiksmai, taip pat ir klaidos.

*Perėjimas prie .NET 3.0 ir architektūros peržiūrėjimas.* Šio projekto realizavimo metu Microsoft kompanija išleido naujesnę .NET platformos versiją. Vienas iš pagrindinių patobulinimų naujoje versijoje yra greitesnis DataSet'ų (duomenų rinkinių) veikimas. Kadangi mūsų sistemoje DataSet'ai naudojami labai plačiai, tai leistų pagreitinti sistemos veikimą. Be kita ko perėjus ant naujesnės platformos versijos atsiveria didesnės saugumo ir vartotojo sąsajos pagerinimo galimybės. Pereinant prie naujos versijos taip pat reiktų didelį dėmesį skirti klasių architektūros refaktoringui siekiu išspręsti 3.2 skyrelyje nustatytas sistemos palaikomumo problemas.

## IŠVADOS

1. Atlikus reikalavimų surinkimui naudojamo Volere šablono ir Oracle CASE įrankio palyginimą, pasirinktas Volere šablonas, kuris suteikia galimybę pasiekti didesnę reikalavimų specifikavimo detalumą. Išnagrinėti pagrindiniai daugiasluoksnių, patikimų ir saugių sistemų projektavimo aspektai.
2. Išanalizavus vartotojo iškeltus pradinius reikalavimus sistemai, nustatyti penkti probleminiai sistemos realizavimo uždaviniai: duomenų integravimas iš įvairių tiekėjų, sinchronizacija tarp įmonių grupės serverių, prekių komplektavimas-suderinamumas, antkainių valdymas, integracija su apskaitos programa „Rivilė“. Visiems jiems pateikti galimi sprendimo variantai.
3. Palyginus tris plačiausiai naudojamą platformas, skirtas WEB sistemų realizavimui (PHP, Java, .NET), taip pat atsižvelgus į sistemai keliamus vartotojo reikalavimus ir numatomus poreikius jų realizavimui, pasirinkta .NET sistemos kūrimo platforma.
4. Remiantis suformuluotais projektiniais, techniniais bei probleminiais uždavinių sprendimais, realizuota integruota įvairių tiekėjų virtualaus prekių sandėlio sistema. Jos architektūra suskirstyta į tris pagrindinius duomenų atvaizdavimo, verslo logikos valdymo ir priėjimo prie duomenų lygius. Atvaizdavimo moduliai sukurti pasinaudojus Modelis-Vaizdas-Valdiklis projektavimo šablonu.
5. Remiantis ISO 9126 standarto rekomenduojamais kriterijais buvo atliktas sistemos kokybės vertinimo tyrimas. Jo metu gautas bendras įvertinimo balas lygus 6,96. Rezultatas rodo, kad sistema yra vidutiniškos kokybės. Išnagrinėjus atskirus vertinimo kriterijus pastebėta, kad prasčiausiai įvertintas sistemos palaikomumas. Atsižvelgiant į tai, tobulinant sistemą siūloma peržiūrėti sistemos klasių architektūrą. Funkcionalumo parametras buvo įvertintas 8,6 balo, kas rodo, kad sistemos atitikimas vartotojų lūkesčiams bei pateiktiems parametrams yra geras.
6. Atlikus sistemos užsakovų apklausos metu gautų duomenų analizę, gauti panašių sistemų teikiamų galimybių svoriniai įverčiai. Remiantis šiais įverčiais buvo atliktas sukurtos virtualaus sandėlio sistemos naudingumo tyrimas. Sudarius naudingumo ir realizuotų galimybių procentinio palyginimo su kitomis sistemomis grafiką, nustatyta, kad sukurta virtualaus sandėlio sistema suteikia daugiausia naudingumo ir realizuoja didžiausią galimybių skaičių. Taip pat pastebėta, jog tik sukurtos sistemos suteikiamų galimybių procentas viršija jos teikiamo naudingumo procentą. Norint išsiaiškinti šio skirtumo priežastis buvo sudarytas realizuotų funkcijų ir svorinių naudingumo įverčių grafikas, kuris

parodė, kad kai kurios galimybės turi tik nedidelę įtaką bendram sistemos naudingumui. Dėl pastarosios priežasties sistemoje nėra tiesioginės priklausomybės tarp sistemos teikiamo naudingumo ir jos realizuojamų funkcijų skaičiaus.

7. Virtualaus sandėlio sistema sėkmingai atiduota užsakovų atstovams UAB „Terra IT“. Šio dokumento prieduose pateikiamas „Programų sistemos perdavimo ir aprobavimo aktas“. Veikiančias programos versijas galima pasižiūrėti internetiniais adresais [www.atomik.lt](http://www.atomik.lt), [www.mikrotestas.lt](http://www.mikrotestas.lt), [www.kilobaitas.lt](http://www.kilobaitas.lt), [www.ralinga.lt](http://www.ralinga.lt) ir kt.

## LITERATŪRA

- [1] Hayden D. *Web Applications: N-Tier vs. N-Layer - Benefits and Trade-Offs*. 2005 m. [žiūrėta 2007-04-30]  
Prieiga per internetą: <<http://davidhayden.com/blog/dave/archive/2005/07/22/2401.aspx>>.
- [2] Freeman A., Jones. A. *Programming .NET Security*. O'Reilly. 2003.
- [3] Gerard J., Rajeev J. *Reliable Software Systems Design: Defect Prevention, Detection, and Containment*. Jet Propulsion Laboratory, California Institute of Technology. [žiūrėta 2007-05-01]  
Prieiga per internetą: <<http://vstte.ethz.ch/Files/holzmann-joshi.pdf>>.
- [4] *ISO 9126: The Standard of Reference*. [žiūrėta 2007 04 30]  
Prieiga per internetą: <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>
- [5] Howard M. *The 19 Deadly Sins of Software Security*. Michael Howard's Web Log. [žiūrėta 2007-04-20]  
Prieiga per internetą:  
<[http://blogs.msdn.com/michael\\_howard/archive/2005/07/11/437875.aspx](http://blogs.msdn.com/michael_howard/archive/2005/07/11/437875.aspx)>.
- [6] Butleris R.. *Reikalavimo specifikuojimo oracle case terpėje plėtra*. [žiūrėta 2007-03-30]  
Prieiga per internetą: <<http://www.leidykla.vu.lt/inetleid/inf-mok/19/str6.html>>.
- [7] Woodcock J., Davies J. *Using Z Specification, Refinement, and Proof*. University of Oxford. 1995.
- [8] Oracle Designer/2000. *Product Overview for Windows 95/NT*. Oracle Corporation. 1995.
- [9] Montlick T. *What is Object-Oriented Software? Software Design Consultants, LLC*. 1999 m. [žiūrėta 2007-05-05.]  
Prieiga per internetą: <<http://www.softwaredesign.com/objects.html>>.
- [10] Booch G., Rumbaugh J., Jacobson I. *The Unified Modeling Language User Guide*. Addison-Wesley. 1998.
- [11] Butkienė R. *Informacijos sistemų projektavimas Oracle Designer/2000 priemonėmis*. Kaunas, „Technologija“. 1998.



[12] Benett G. *Working Together, Apart : Intranet Journal*. [žiūrėta: 2007 04 11]

Prieiga per internetą: <<http://intranetjournal.com/features/idm0398-pm1.shtml>>.

[13] Joshi B. *N-Tier Applications and .NET : DotNetBips*. [žiūrėta: 2007 04 16]

Prieiga per internetą: <<http://www.dotnetbips.com/articles/displayarticle.aspx?id=213>>.

[14] *Design and Implementation Guidelines for Web Clients*. Microsoft Corporation, 2003.

[15] *Model-View-Controller*. Microsoft Corporation 2007 m. [žiūrėta 2007-05-18]

Prieiga per internetą: <<http://msdn2.microsoft.com/en-us/library/ms978748.aspx>>.

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

### Santrumpos

PĮ – programinė įranga.

RUP – *Rational Unified Process* – unifikuotas projektavimo procesas

UML – *Unified Modeling Language* – unifikuota projektavimo kalba

BLL – *Business Logic Layer* – veiklos logikos projektavimo sluoksnis

DB – duomenų bazė.

PĮ – programinė įranga.

PL – *Presentation Layer* – atvaizdavimo sluoksnis

UI – *User Interface* – vartotojo sąsaja

DAL – *Data Access Layer* – priėjimo prie duomenų projektavimo sluoksnis

ĮG – įmonių grupė

DBVS – duomenų bazių valdymo sistema

MVC – *Model View Controler* – modelis vaizdas valdiklis

ISO – *International Organization for Standardization* – tarptautinė standartizavimo organizacija

### Terminai

Refaktoringas – klasių architektūros reorganizavimas, patobulinimas

## PRIEDAS 1: ĮDIEGIMO DOKUMENTAS

UAB „Terra IT“

Įmonės kodas: 30043862  
PVM kodas: 100001154519  
El. paštas: info@terraIT.net  
Tel.: +370 37 300832, +370 687 25439  
Faks.: +370 37 300832  
Adresas: Studentų g. 65, LT-51369 Kaunas, Lietuva

### Programų sistemos PERDAVIMO IR APROBAVIMO AKTAS

200 6 m. Gruodžio 15 d.

Programų sistemos pavadinimas Įvairių tiekėjų virtualus prekių sandėlys (XPriceList).

Kūrinio tipas Elektroninė kompiuterinių prekių parduotuvė.

Programų sistemos sukūrimo data 200 6 m. Lapkričio 15 d.

Kūrinio įteikimo UŽSAKOVUI data 200 6 m. Gruodžio 01 d.

Užsakovo arba trečiojo asmens Kūrinio aprobavimo rezultatas:

Programinės įrangos sistema sėkmingai sukurta ir įdiegta.

Kūrinio aprobavimo data 200 6 m. Gruodžio 10 d.

Kūrinio originalo saugotojas - UAB TerraIT

AUTORIUS

UŽSAKOVAS

Egidijus Bartkus

(vardas, pavardė)

(parašas)

Egidijus Grigas

(vardas, pavardė)

(parašas)

## PRIEDAS 2: APKLAUSA

### Apklaustos forma

Prašome įvertinti žemiau pateiktas virtualaus sandėlio klientinės dalies galimybes pagal jų svarbumą.

Vertinimą prašome pateikti dešimtbalėje sistemoje, kur:

0 – mažiausiai svarbi galimybė;

10 – labai svarbi galimybė.

1. *Prekių medis* – prekių atvaizdavimas medžio forma  
[vertinimas \_\_\_\_]
2. *Prekių krepšelis* – vartotojo galimybė susidėti norimas prekes į virtualų krepšelį ir vėliau jas užsisakyti arba bent atsispausdinti  
[vertinimas \_\_\_\_]
3. *Galimybė sukomplektuoti suderinamą kompiuterį* – vartotojo galimybė iš atskyrų dalių susikomplektuoti kompiuterį, kurio sudedamosios dalys būtų suderinamos  
[vertinimas \_\_\_\_]
4. *Paieška* – elementari paieška pagal įvestą žodį  
[vertinimas \_\_\_\_]
5. *Detali paieška* – galimybė ieškoti pagal įvairius parametrus pvz.: gamintojas, kaina, kiekis ir t.t.  
[vertinimas \_\_\_\_]
6. *Kainų pokytis* – galimybė koku nors būdu pažymėti prekes kurių kaina neseniai pasikeitė  
[vertinimas \_\_\_\_]
7. *Kainų kitimo grafikas* – vartotojo galimybė matyti kaip keitėsi prekės kaina praeityje  
[vertinimas \_\_\_\_]
8. *Naujos prekės* – galimybė kaip nors išskirti naujas prekes  
[vertinimas \_\_\_\_]
9. *Akcijinės prekės* – galimybė išskirti akcijines prekes su mažesne kaina  
[vertinimas \_\_\_\_]
10. *Kiekis sandėlyje* – galimybė rodyti kiek šiuo metu turima konkrečios prekės vienetų  
[vertinimas \_\_\_\_]
11. *Prekės aprašymas* – galimybė pateikti detalų prekės aprašymą  
[vertinimas \_\_\_\_]

12. *Gamintojo aprašymas* – galimybė pateikti prekės gamintojo aprašymą  
Įvertinimas \_\_\_\_
13. *Prekių sąrašo spausdinimas* – galimybė pagal tam tikrus parametrus išsifiltruotą sąrašą atsispausdinti  
Įvertinimas \_\_\_\_
14. *Prekių sąrašo eksportavimas* – galimybė išsifiltruotą arba visą sąrašą gauti kitu formatu pvz.: excelis, wordas ir pan.  
Įvertinimas \_\_\_\_
15. *Prekių sąrašo puslapiavimas* – galimybė vartotojui pasirinkti po kiek prekių vienu metu rodoma ekrane  
Įvertinimas \_\_\_\_
16. *Užsakymų registravimas* – galimybė į prekių krepšelį susidėtas prekes iškart užsisakyti  
Įvertinimas \_\_\_\_
17. *Registruoti klientai* – galimybė turėti nuolatinius registruotus klientus  
Įvertinimas \_\_\_\_
18. *Prekių palyginimas* – galimybė palyginti tarpusavyje dvi ar daugiau prekes  
Įvertinimas \_\_\_\_

### **Apklauso rezultatai**

Apklausa buvo atliekama praėjus vienam mėnesiui po sistemos įdiegimo pas užsakovą. Apklausti šeši žmonės. Apklauso rezultatų suvestinė pateikiama toliau esančioje lentelėje.

<b>Galimybės</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>Vid.</b>
Prekių medis	10	10	9	10	9	10	<b>10</b>
Prekių krepšelis	8	10	7	8	8	9	<b>8</b>
Galimybė sukomplektuoti suderinamą kompiuterį	9	8	6	6	7	8	<b>7</b>
Paieška	10	10	9	10	10	10	<b>10</b>
Detali paieška	5	8	3	7	7	8	<b>6</b>
Kainų pokytis	6	7	6	7	8	9	<b>7</b>
Kainų kitimo grafikas	0	2	0	1	0	0	<b>0</b>
Naujos prekė	5	9	5	4	10	8	<b>7</b>
Akcijinės prekės	8	6	9	7	8	8	<b>8</b>
Kiekis sandėlyje	10	10	9	8	10	9	<b>9</b>
Prekės aprašymas	10	10	10	10	10	10	<b>10</b>
Gamintojo aprašymas	8	3	5	5	4	4	<b>5</b>
Prekių sąrašo spausdinimas	7	8	8	6	9	10	<b>8</b>
Prekių sąrašo eksportavimas	2	3	0	0	1	1	<b>1</b>
Prekių sąrašo puslapiavimas	0	0	0	0	0	1	<b>0</b>
Užsakymų registravimas	8	8	7	7	8	6	<b>7</b>
Registruoti klientai	6	8	5	8	7	5	<b>6</b>
Prekių palyginimas	9	10	8	10	10	10	<b>10</b>