

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Renatas Narmontas

**Struts karkaso panaudojimo tyrimas mobilaus  
ryšio infrastruktūros valdymo sistemoje**

Magistro darbas

Vadovas

doc. dr. E. Karčiauskas

2008-05-26

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Renatas Narmontas

**Struts karkaso panaudojimo tyrimas mobilaus  
ryšio infrastruktūros valdymo sistemoje**

Magistro darbas

Recenzentas

doc. S. Maciulevičius  
2008-05-26

Vadovas

doc. dr. E. Karčiauskas  
2008-05-26

Atliko

IFM-2/2 gr. stud.  
R. Narmontas  
2008-05-26

Kaunas, 2008

# TURINYS

<b>TURINYS</b> .....	<b>3</b>
<b>PAVEIKSLŲ SĄRAŠAS</b> .....	<b>6</b>
<b>LENTELIŲ SĄRAŠAS</b> .....	<b>7</b>
<b>1. ĮVADAS</b> .....	<b>8</b>
1.1. MAGISTRO BAIGIAMOJO DARBO TIKSLAI .....	9
1.2. DOKUMENTO PASKIRTIS .....	9
1.3. SANTRAUKA .....	9
1.4. MAGISTRO BAIGIAMOJO DARBO STRUKTŪRA.....	10
<b>2. ARCHITEKTŪROS MODELIS-VAIZDAS-VALDIKLIS ANALIZĖ</b> .....	<b>11</b>
2.1. APACHE STRUTS KARKASAS .....	11
2.2. MVV ARCHITEKTŪRA .....	11
2.3. PRIVALUMAI IR TRŪKUMAI .....	14
<b>3. MOBILAUS RYŠIO INFRASTRUKTŪROS VALDYMO SISTEMOS PROJEKGINĖ DALIS</b> .....	<b>16</b>
3.1. SISTEMOS TIKSLAS .....	16
3.2. EGZISTUOJANTYS SPRENDIMAI .....	17
3.3. VEIKLOS KONTEKSTAS .....	18
3.4. PANAUDOJIMO ATVEJAI.....	20
3.5. FUNKCINIAI SISTEMOS REIKALAVIMAI .....	21
3.6. NEFUNKCINIAI SISTEMOS REIKALAVIMAI .....	21
3.7. REIKALAVIMAI DUOMENIMS .....	23
3.8. PROJEKTO RIZIKOS .....	26
3.8.1. <i>Sistemos kūrimo rizikos</i> .....	26
3.8.2. <i>Rizikų valdymo planas</i> .....	26
3.9. ARCHITEKTŪROS SPECIFIKACIJA .....	27
3.9.1. <i>Sistemos statinis vaizdas</i> .....	28
3.9.2. <i>Sistemos dinaminis vaizdas</i> .....	30
3.9.3. <i>Sekų diagramos</i> .....	31
3.9.4. <i>Sistemos išdėstymo vaizdas</i> .....	33

3.9.5.	<i>Komponentų diagrama</i> .....	34
3.10.	SISTEMOS ĮGYVENDINIMAS .....	34
3.10.1.	<i>Struts karkasas</i> .....	35
3.10.2.	<i>Tiles karkasas</i> .....	35
3.10.3.	<i>Saugumas</i> .....	36
3.10.4.	<i>Internacionalizacija</i> .....	37
3.10.5.	<i>Duomenų bazės kūrimas</i> .....	37
3.10.6.	<i>Grafinė vartotojo sąsaja</i> .....	38
3.11.	SISTEMOS TESTAVIMAS.....	39
3.11.1.	<i>Testuojama programinė įranga</i> .....	39
3.11.2.	<i>Testavimo įrankiai ir aplinka</i> .....	39
3.11.3.	<i>Testų metodų aprašymas</i> .....	39
3.11.4.	<i>Testavimo rezultatai ir išvados</i> .....	40
3.12.	SISTEMOS DOKUMENTACIJA .....	41
<b>4.</b>	<b>ŽINIATINKLIŲ PROGRAMŲ KŪRIMO KARKASŲ TYRIMAS</b> .....	<b>43</b>
4.1.	ŽINIATINKLIŲ PROGRAMŲ KŪRIMO KARKASAI.....	43
4.1.1.	<i>JSF</i> .....	43
4.1.2.	<i>Spring MVC</i> .....	44
4.1.3.	<i>Stripes</i> .....	44
4.1.4.	<i>Struts</i> .....	44
4.1.5.	<i>Struts2</i> .....	44
4.1.6.	<i> Tapestry</i> .....	45
4.1.7.	<i>Wicket</i> .....	45
4.1.8.	<i>WebWork</i> .....	45
4.2.	VERTINIMO KRITERIJAI .....	46
4.2.1.	<i>Karkasų vertinimas</i> .....	46
4.2.2.	<i>Darbo pasiūlymų pasiskirstymas</i> .....	49
<b>5.</b>	<b>EKSPERIMENTINĖ DALIS</b> .....	<b>50</b>
5.1.	ĮGYVENDINTI PANAUDOJIMO ATVEJAI .....	50
5.2.	TESTUOTI PANAUDOJIMO ATVEJAI .....	50
5.3.	NAŠUMAS.....	50
5.4.	KODO METRIKOS.....	50
5.5.	PROJEKTAVIMO LAIKO ĮVERTINIMAS.....	50

<b>6. IŠVADOS.....</b>	<b>56</b>
<b>7. LITERATŪRA .....</b>	<b>57</b>
<b>8. ANALYSIS OF STRUTS FRAMEWORK IN MOBILE NETWORK INFRASTRUCTURE MANAGEMENT SYSTEM .....</b>	<b>60</b>
<b>9. TERMINŲ IR SANTRUMPŲ ŽODYNAS.....</b>	<b>61</b>
<b>PRIEDAS NR. 1 PROGRAMINĖS ĮRANGOS ĮDIEGIMO PATVIRTINIMO AKTAS.....</b>	<b>63</b>
<b>PRIEDAS NR. 2 PANAUDOJIMO ATVEJŲ SĄRAŠAS .....</b>	<b>64</b>
<b>PRIEDAS NR. 3 LICENZIJA .....</b>	<b>67</b>
LICENZIJOS TEKSTAS .....	67
LICENZIJOS REGISTRACIJOS FORMA .....	68
<b>PRIEDAS NR. 4 SISTEMOS KATALOGO STRUKTŪRA.....</b>	<b>69</b>
<b>PRIEDAS NR. 5 PRIDĖTO KOMPAKTINIO DISKO TURINYS .....</b>	<b>70</b>
<b>PRIEDAS NR. 6 KOMPAKTINIS DISKAS.....</b>	<b>71</b>

## PAVEIKSLŲ SĄRAŠAS

1 pav. Modelis-Vaizdas-Valdiklis architektūra .....	12
2 pav. Programos modelis .....	13
3 pav. Sistemos veiklos konteksto diagrama .....	18
4 pav. Panaudojimo atvejų diagrama .....	20
5 pav. Esybių – ryšių diagrama.....	23
6 pav. Duomenų bazės modelis .....	24
7 pav. Sistemos išskaidymas į paketus.....	28
8 pav. Paketo Prisijungimas klasių diagrama .....	29
9 pav. Paketo Administravimas klasių diagrama.....	30
10 pav. Veiklos diagramos šablonas .....	31
11 pav. Sekos diagrama.....	32
12 pav. Bendradarbiavimo diagrama .....	32
13 pav. Sistemos išdėstymo vaizdas .....	33
14 pav. Sistemos komponentų diagrama .....	34
15 pav. Išdėstymo momentinė ekrano kopija .....	38
16 pav. Login lango testavimas .....	41
17 pav. Darbo pasiūlymų pasiskirstymas .....	49
18 pav. Paslaugų sąrašo langas .....	51
19 pav. Naujos paslaugos pridėjimo langas .....	52
20 pav. Eilučių skaičiaus priklausomybė nuo puslapių skaičiaus (1).....	53
21 pav. Eilučių skaičiaus priklausomybė nuo puslapių skaičiaus (2).....	54
22 pav. Kodo eilučių procentinis palyginimas .....	54

## LENTELIŲ SĄRAŠAS

Lentelė Nr. 1 Magistro baigiamojo darbo struktūra.....	10
Lentelė Nr. 2 Populiariausių programų palyginimas .....	17
Lentelė Nr. 3 Veiklos įvykių sąrašas.....	19
Lentelė Nr. 4 Pagrindinių sistemos aktorių sprendžiami uždaviniai .....	20
Lentelė Nr. 5 Sistemos funkciniai reikalavimai .....	21
Lentelė Nr. 6 Sistemos nefunkciniai reikalavimai.....	21
Lentelė Nr. 7 Duomenų bazės esybės parašymai .....	24
Lentelė Nr. 8 Sistemos kokybės įvertinimas .....	25
Lentelė Nr. 9 Galimos sistemos kūrimo rizikos .....	26
Lentelė Nr. 10 Rizikų valdymo planas.....	26
Lentelė Nr. 11 Vaizdai ir modeliai.....	27
Lentelė Nr. 12 Microsoft Windows XP reikalavimai .....	33
Lentelė Nr. 13 Žiniatinklių programų kūrimo karkasų palyginimas .....	48
Lentelė Nr. 14 Kodo eilučių skaičius pirmam puslapiui.....	51
Lentelė Nr. 15 Kodo eilučių skaičius antram puslapiui .....	52
Lentelė Nr. 16 Rekomendacijos vartotojui.....	55
Lentelė Nr. 17 Sistemos katalogo struktūra .....	69

# 1. Įvadas

Žiniatinklio (angl. web) [2] programų kūrimo bumą prasidėjo XX a. dešimtmetyje. Nuo to laiko Java programavimo kalba pradėjo dominuoti tokio tipo programų kūrimo srityje. Tačiau yra ir atviro kodo projektų. Paskutiniaisiais metais atviro kodo žiniatinklio programų kūrimo Apache Struts (toliau Struts) karkasas išsikovojo savo vietą šalia kitų.

Šio magistrinio darbo tikslas yra ištirti žiniatinklio programų kūrimo Struts karkaso panaudojimo savybes mobilios ryšio infrastruktūros valdymo sistemoje (MRIVS). Visa informacija yra saugoma reliacinėje duomenų bazių valdymo sistemoje.

Daug sistemų naudoja žiniatinklio tipo programas įvairiems savo uždaviniams spręsti, tačiau mažai jų yra pakankamai lanksčių, dažniausiai jas sunku internacionalizuoti ir jų vystymo bei tobulinimo laikas yra ilgas. Šiame darbe mes remiamės tokiais savybėmis kaip: lankstumas, internacionalizavimas, saugumas, tolerancija klaidoms ir komponentais grįsta.

Lankstumą mes suprantame kaip vis naujų ir naujų savybių pridėjimą į sistemą. Programa taip pat turi būti daugiakalbė. Vienu metu skirtingi vartotojai gali matyti programą įvairiomis kalbomis. Pradiniame etape yra numatytos dvi kalbos: lietuvių ir anglų, tačiau nesunkiai galima kalbų skaičių plėsti. Tai taip pat yra ir sistemos lankstumo dalis. Sistemoje naudojami įvairūs konfidencialūs duomenys, todėl turi būti užtikrintas duomenų saugumas. Įvairūs žurnalai (angl. log) padeda administratoriams išvengti klaidų. Paskutinis reikalavimas yra reikalingas norint sistemą plėsti. Komponentai dažniausiai yra vienas nuo kito nepriklausomi, todėl turi nedidelę įtaką, jei atliekami kokie nors pakeitimai.

Tai reiškia, kad programa turi būti suskaidyta į kelis lygmenis (angl. layer) [2] ir programos tekstas neturi būti įkoduotas (angl. hard-coded) [2], nes kitu atveju programuotojų darbo kiekis augs eksponentiškai.

Viena iš svarbesnių dalių yra pavaizdavimo lygmuo. Šis komponentas pateikia vartotojui įvairius duomenis. Mes naudojame žymes (angl. tags) [2], kad atskirti šį lygmenį nuo programos kodo. Tai leidžia keisti JSP (angl. JavaServer Page) failus nekeičiant Java programos kodo. Yra daugiau darbo, kai informacija pateikiama žymių iteracijoje. Tačiau šie papildomi darbai yra minimalūs lyginant su Java kodo pridėjimu ir atnaujinimu į JSP failus.

Kita svarbi dalis yra vartotojo įvedamų duomenų patikra (angl. validation) [2]. Visi vartotojo duomenys turi būti patikrinti, kad išvengti klaidų ar sistemos nestabilumo. Šis patikrinimas yra atliekamas serverio pusėje. To priežastis yra ta, kad kliento pusėje įmanoma išjungti patikros mechanizmus, pvz. neaktyvavus java skriptų.

Pagrindinis programos tikslas ir yra laikytis šių savybių.



## 1.1. Magistro baigiamojo darbo tikslai

Pagrindiniai magistro baigiamojo darbo tikslai: ištirti žiniatinklio programų kūrimo Struts karkaso panaudojimo savybes; suprojektuoti ir įgyvendinti mobilaus ryšio infrastruktūros valdymo sistemą, užtikrinančią operatyvų resursų valdymą; išnagrinėti Struts karkaso tinkamumą projektuojamoje sistemoje bei palyginti Struts karkasą su kitais egzistuojančiais atviro kodo karkasais žiniatinklio programoms kurti.

Laukiami rezultatai:

- Ištirtos pagrindinės Struts karkaso panaudojimo savybės;
- Išnagrinėtas Struts karkaso tinkamumas projektuojamoje sistemoje;
- Sukurta programinė įranga, kuri atitinka 3-iajame skyriuje pateiktus sistemos reikalavimus;
- Išanalizuoti ir palyginti tarpusavyje įvairūs atviro kodo karkasai žiniatinklio programoms kurti.

## 1.2. Dokumento paskirtis

Šis dokumentas yra Kauno technologijos universiteto Informatikos fakulteto Programų inžinerijos katedros IFM-2/2 gr. studento Renato Narmento Informatikos inžinerijos mokslo magistro baigiamasis darbas „Struts karkaso panaudojimo tyrimas mobilaus ryšio infrastruktūros valdymo sistemoje“.

## 1.3. Santrauka

Rinkoje yra daug įvairių žiniatinklio programų kūrimo karkasų. Modelis-Vaizdas-Valdiklis (MVV) architektūra tampa „*de facto*“ standartu tokių žiniatinklio programų kūrimo[6]. Šiame darbe palyginami įvairūs žiniatinklio programų kūrimo karkasai, pabrėžiami jų privalumai ir trūkumai. Šiame darbe nagrinėjami Apache Struts karkaso panaudojimo aspektai mobilaus ryšio infrastruktūros valdymo sistemoje. Buvo sukurta lanksti, internacionalinė, saugi, klaidoms tolerantiška ir MVV architektūra paremta sistema.

## 1.4. Magistro baigiamojo darbo struktūra

Magistro baigiamojo darbo struktūra pateikta lentelėje Nr. 1.

Lentelė Nr. 1 Magistro baigiamojo darbo struktūra

Skirius	Aprašymas
<a href="#">Architektūros Modelis-Vaizdas-Valdiklis analizė</a>	Šiame skyriuje pateikiama Modelio-Vaizdo-Valdiklio (MVV) architektūra. Aprašomas MVV architektūrą naudojančio Apache Struts karkaso panaudojimas, išvardijami tokios architektūros pagrindiniai privalumai ir trūkumai.
<a href="#">Mobilaus ryšio infrastruktūros valdymo sistemos projektinė dalis</a>	Šiame skyriuje aprašoma mobilaus ryšio infrastruktūros valdymo sistema, jos sukūrimo tikslai. Nagrinėjami egzistuojantys sprendimai, analizuojamas veiklos kontekstas, aprašomi panaudojimo atvejai, funkciniai ir nefunkciniai sistemos reikalavimai, bei reikalavimai duomenims. Įvardinamos projekto rizikos, bei jų valdymo sprendimai. Pateikiama sistemos architektūros specifikacija, sekų diagramos, sistemos dinaminis bei išdėstymo vaizdai. Aprašomas sistemos testavimas.
<a href="#">Žiniatinklių programų kūrimo karkasų tyrimas</a>	Šiame skyriuje tiriami įvairūs žiniatinklio programų kūrimo karkasai, atliekamas jų vertinimas pagal įvairius iš anksto pasirinktus kriterijus.
<a href="#">Eksperimentinė dalis</a>	Šiame skyriuje aprašoma eksperimento eiga, pateikiami gauti rezultatai.
<a href="#">Išvados</a>	Šiame skyriuje pateikiamos magistro baigiamojo darbo išvados.
<a href="#">Literatūra</a>	Šiame skyriuje nurodoma naudota literatūra.
<a href="#">Terminų ir santrumpų žodynas</a>	Šiame skyriuje pateikiamas terminų ir santrumpų žodynas, sąvokų bei žodžių trumpas paaiškinimas.
<a href="#">Priedai</a>	Šiame skyriuje pateikiami įvairūs priedai.

## 2. Architektūros Modelis-Vaizdas-Valdiklis analizė

Šiame skyriuje pateikiami Modelio-Vaizdo-Valdiklio (MVV) architektūros pradmenys. Aprašomas MVV architektūrą naudojančio Apache Struts karkaso panaudojimas, tokios architektūros pagrindiniai privalumai ir trūkumai.

### 2.1. Apache Struts karkasas

Apache Struts yra atviro kodo karkasas, naudojamas žiniatinklio (angl. Java WEB) [2] programoms kurti. Jis buvo sukurtas J2EE žiniatinklio programoms kurti naudojant Modelio-Vaizdo-Valdiklio (sutr. MVV) architektūrą (angl. Model-View-Controller, sutr. MVC). Pradininkas buvo Craig'as McClanahan'as, tačiau 2000 m. gegužės mėnesį nusprendė atiduoti programos kodą atviro kodo bendruomenei. Dabar pagrindinis sponsorius yra Apache Software Foundation organizacija [25].

Dauguma svetainių (angl. website) pateikia statinius tinklalapius (angl. web pages) [34]. Žiniatinklio programos gali imti duomenis iš duomenų bazių ir pritaikius verslo logikos taisykles pateikti individualiam vartotojui pritaikytą atsakymą.

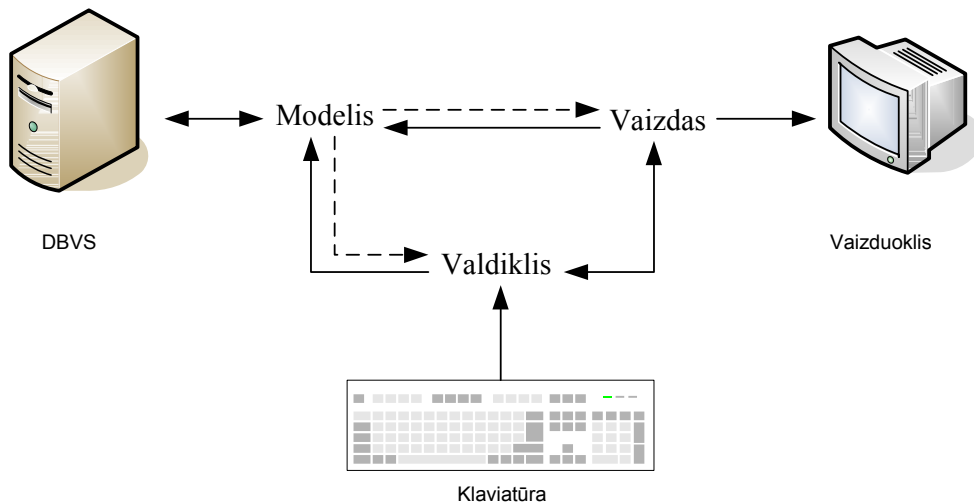
Žiniatinklio programos, paremtos JavaServer Pages (angl. sutr.. JSP) technologija, kartais sumaišo duomenų bazių, puslapio dizaino ir valdymo operacijų sekos kodus. Praktikoje tokios didelės programos tampa sunkiai prižiūrimos ir eksploatuojamos.

Vienas iš būdų kodams atskirti – naudoti Modelio-Vaizdo-Valdiklio architektūrą. Modelis reprezentuoja verslo arba duomenų bazės kodą; Vaizdas reprezentuoja puslapio dizaino kodą; Valdiklis pristato navigacinį kodą. Apache Struts karkasas yra sukurtas tam, kad padėtų programuotojams kurti žiniatinklio programas, kurios naudoja šią MVV architektūrą. Karkasas padeda atskirti pavaizdavimo lygmenį. Taip pat pasižymi tokiomis savybėmis kaip: internacionalizavimas, išsamios žymių bibliotekos (angl. tag library), išsklotinės (angl. tiles) ir formų patikra (angl validation).

### 2.2. MVV architektūra

Apache Struts karkasas paremtas MVV architektūra, kuri suskaido programą į tris skirtingas dalis: duomenų modelį, vartotojo grafinę sąsają ir valdymo logiką. Tokiu būdu kiekvienos dalies pakeitimai minimaliai įtakoja kitas dalis.

MVV architektūra sudaryta iš modelio, vaizdo ir valdiklio [4, 14, 22, 37].



1 pav. Modelis-Vaizdas-Valdiklis architektūra

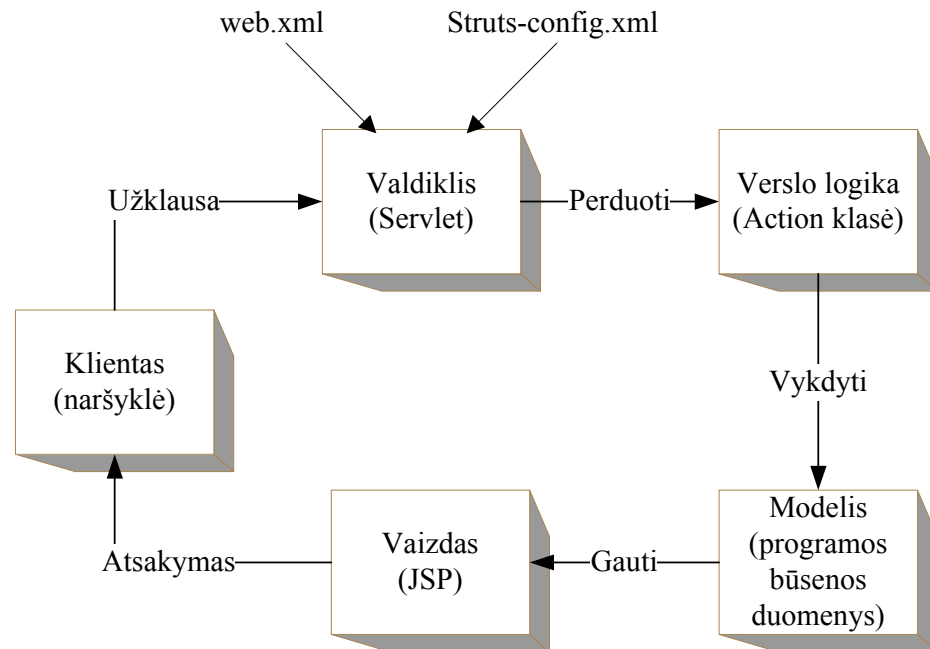
Pagrindinė Struts karkaso dalis yra MVV stiliaus valdiklis. Valdiklis užpildo tarpą tarp modelio (pvz. JDBC) ir vaizdo (pvz. JSP).

Modelis – reprezentuoja duomenų objektus. Modelis yra toks darinys, kuris naudojamas veiksmams atlikti ir gautam rezultatui pateikti vartotojui.

Vaizdas – naudojamas modelio komponento pavaizdavimui ekrane. Tai objektas, kuris rodo duomenų objekto dabartinę būseną. Vaizdo komponentas yra surišamas su JSP failais, kurie naudoja Struts žymes, HTML ir JSP. Visi JSP failai yra valdomi Tiles karkaso. Naudojant Tiles karkasą mes pernaudojame (angl. reuse) pavaizdavimo komponentus, tokius kaip paveikslus, HTML, bei kitus vizualius komponentus.

Valdiklis – apibrėžia grafinės sąsajos reakciją vartotojo veiksmų atžvilgiu. Tai objektas, kuris manipuliuoja modeliu.

Programos, kurios naudoja MVV architektūrą, modelis pavaizduojamas 2 paveiksle [13]:



2 pav. Programos modelis

Pagrindiniai tokios architektūros privalumai yra šie [22]:

- Patikimumas. Pavaizdavimo ir transakcijų sluoksniai dažniausiai yra atskirti vienas nuo kito. Tai leidžia keisti programos išvaizdą neperdarant modelio ar valdiklio programinio kodo.
- Pakartotinis panaudojimas (angl. reuse) ir pritaikymas. MVV leidžia naudoti skirtingus vaizdo tipus tam pačiam serverio pusės kodui (pvz. tiek HTTP, tiek WAP).
- Greitas tobulinimas. Trumpesnis tobulinimo laikas, nes Java programuotojai kuria valdiklį, o vaizdą paprastai kuria HTML ir JSP programuotojai.
- Priežiūra. Pavaizdavimo ir verslo logikos atskyrimas palengvina priežiūrą ir tobulinimą.

Yra keletas populiarių servlet'ų konteinerių rinkoje. Vieni servlet'ų konteineriai turi būti prijungti prie išorinių žiniatinklio serverių, o kiti turi ir nuosavus žiniatinklių serverius. Populiariausi servlet'ų konteineriai yra šie: Borland Enterprise Server, Sun One Application Server, Jetty, JRun, Apache Tomcat, IBM WebSphere, Weblogic Application Server. Šiame darbe panaudotas Apache Tomcat servlet'ų konteineris.

## 2.3. Privalumai ir trūkumai

Kaip ir bet kuri kita architektūra, programa ar sistema, taip ir MVV architektūra turi savus privalumus ir trūkumus [12, 13].

Pagrindiniai Struts MVV architektūros privalumai yra šie:

- Standartinis HTTP. Struts sukurtas naudoti standartinį HTTP request-response modelį, žinomą daugeliui žiniatinklių programuotojų.
- Standartinis veiksmų užrašymas (angl. logging) [2]. Struts gali naudoti konteinerio numatytąjį sistemos žurnalą (angl. logging system) [2] ir nereikalauja jokio papildomo paketo ir specialių žinių.
- Neutralus modelis. Struts neturi išankstinio nusistatymo kokio nors modelio atžvilgiu.
- Lengvas (angl. lightweight). Yra tik kelios pagrindinės Struts klasės, kurias programuotojai turi žinoti kaip jomis naudotis.
- Stipri programuotojų bendruomenė. Susirašinėjimas el. paštu yra aktyvus.
- Stipri gamintojų bendruomenė. Struts buvo integruotas į kai kurias kitas sistemas, pvz. IBM WebSphere.
- Gerai dokumentuota. Išėities tekstai yra taip gerai dokumentuoti, kad beveik niekada netenka nagrinėti jų pačių. Pakanka perkaityti dokumentaciją.
- Išplečiamas. Visi numatytieji parametrai gali būti keičiami. Pagrindinės klasės gali būti paveldėtos ir naudojamos vietoj jų.

Pagrindiniai MVV architektūros trūkumai yra šie:

- Derinimas (angl. debug). Nėra automatinio derinimo palaikymo galimybės. Programuotojai turi patys dirbtinai kurti sustojimo taškus (angl. breakpoints) išvedant informaciją į konteinerio log failą.
- Duomenų modelis. Nėra numatytojo duomenų modelio. Visa prieiga prie duomenų modelio yra palikta kaip užduotis programuotojui.
- Vienas ActionServlet'as. Programoje gali būti naudojamas tik vienas ActionServlet'as. Kitu atveju gali sukelti konfigūracinių failų priešpriešą.
- Specialios žinios. Reikalauja Struts komponentų sąveikos darbo žinių. Programuotojams būtina suprasti kelias specialias Struts klases ir žinoti kaip jos sąveikauja.

- Nėra oficialaus gamintojo palaikymo. Apache Software Foundation organizacija yra savanoriška. Neturi mokamo personalo, kuris pateiktų garantuotą atsakymą.
- Lėtos formalios versijos. Palyginamai lėtai išleidžiamos naujos versijos lyginant su kitais projektais. Programuotojai turi naudoti tarpines versijas (angl. nightly build) [2], kad galėtų naudoti paskutinius patobulinimus. Taip pat nėra versijų išleidimo grafikų.
- I18n apribojimai. Struts pranešimų resursai yra pakankamai geri internacionalizuoti žymes ir klaidų pranešimus tačiau nėra skirti valdyti didesnius tekstų blokus.
- JSP išimčių lokalizavimas. Dauguma sistemos lygio pranešimų, tokių kaip JSP išimtys (angl. exceptions) [2], nėra lokalizuojamos ir visada pranešimai rodomi anglų kalba.

### **3. Mobilaus ryšio infrastruktūros valdymo sistemos projektinė dalis**

Šiame skyriuje aprašoma mobilaus ryšio infrastruktūros valdymo sistema, jos sukūrimo tikslai. Nagrinėjami egzistuojantys sprendimai, analizuojamas veiklos kontekstas, aprašomi panaudojimo atvejai, funkciniai ir nefunkciniai sistemos reikalavimai, bei reikalavimai duomenims. Įvardinamos projekto rizikos, bei jų valdymo sprendimai. Pateikiama sistemos architektūros specifikacija, sekų diagramos, sistemos dinaminis bei išdėstymo vaizdai. Aprašomas sistemos testavimas.

#### **3.1. Sistemos tikslas**

Juntamas poreikis naudoti „protingas“ mobilaus ryšio infrastruktūros valdymo sistemas. Daugėjant paslaugoms nebeužtenka paprastų užrašų knygučių aprašyti kas ir kur įdiegta. Reikia tiksliai žinoti ir identifikuoti kokia įranga dalyvauja teikiant klientams vienokią ar kitokią paslaugą. Dažniausiai vienai paslaugai teikti yra naudojama vienu serverių kombinacija, kitai – kitų serverių kombinacija. Taip pat būna, kad ir kelios paslaugos naudojasi tuo pačiu serveriu. Reikia tiksliai identifikuoti kokios paslaugos nukentės sugedus vienam ar kitam serveriui.

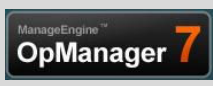



Šio magistrinio darbo tikslas yra sukurti mobilaus ryšio infrastruktūros valdymo sistemą. Visa informacija bus saugoma reliacinėje duomenų bazėje. Turi būti numatyta galimybė gauti įvairias ataskaitas, statistikas bei grafikus ar kitokiais pjūviais. Paslaugos vartotojams dažniausiai būna paprastos ir lengvai naudojamos, tačiau jų techninis įgyvendinimas dažnai būna susijęs su dviem ar daugiau serverių bei susipynęs su kitomis paslaugomis. Šiame magistriniame darbe bus numatyta galimybė identifikuoti visas paslaugas, kurių veikimui įtakos turės vieno ar kito serverio ar komponento gedimas. Taip pat šios programinės įrangos pagalba bus galima lengvai surasti vieno ar kito serverio dislokacijos vietą.



### 3.2. Egzistuojantys sprendimai

Šiuo metu pasaulinėje rinkoje yra nemažai produktų, tačiau dauguma jų yra brangūs. Žemiau pateikiame populiariausių produktų apžvalgą.

Lentelė Nr. 2 Populiariausių programų palyginimas

				
Pavadinimas	ManageEngine OpManager	Enterprise Server	Alchemy Network Inventory	Numara™ Track-It!
Pardavėjas	AdventNet Inc	Microforge.net Limited	M.I.S. Helpers	Numara™ Software, Inc
Kaina	6995\$ + priedai (nuo 400\$)	895\$	299\$ + priedai (nuo 399\$)	1200\$
Operacinė sistema	Windows	Windows	Windows	Windows

AdventNet Inc firmos siūlomas ManageEngine OpManager integruotas sprendimas tinklo, sistemų ir programų priežiūrai. Realiu laiku stebi tinklo būklę, spausdintuvų išnaudojimo resursus, surenka įvairių serverių statistiką. Turi virš 160 grafikų ir virš 40 raportų įvairiais pjūviais. Juos galima eksportuoti įvairiais formatais: HTML, PDF, CSV. Tai labai brangus ir galingas įrankis.

Microforge.net Limited firma įkurta 2000 m. Jos siūlomas produktas Enterprise Server 1.4 versijos yra plačiai naudojamas nuo mažų privačių kompanijų iki karinių organizacijų, valdžios departamentų, universitetų, mokyklų daugiau nei 35-iose skirtingose šalyse. Programinė įranga automatiškai skenuoja tinklą, renka detalią informaciją apie įdiegtą programinę įrangą, sistemos nustatymus ir aparatūrą. Saugo programinės įrangos licenzijų ir finansinę informaciją. Leidžia sukurti detalias ataskaitas. Taip pat turi iš karto paruoštų įvairių raportų šablonų.

M.I.S. Helpers kompanijos siūlomas produktas Alchemy Network Inventory skirtas lokalaus tinklo programinės ir sistemų informacijai saugoti. Turi raportų eksporto į išorines

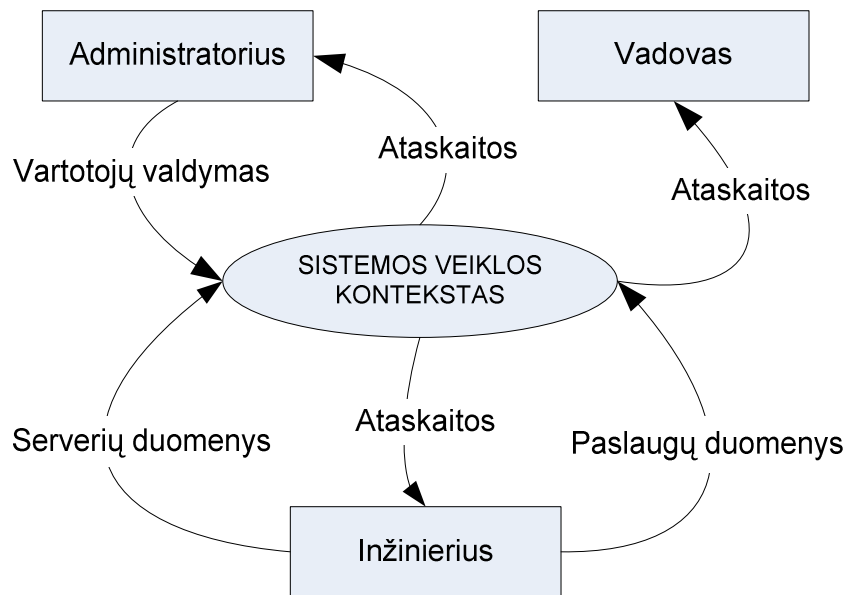
saugyklas galimybę. Automatiškai gali surinkti informaciją apie operacinės sistemos savybes, procesorius, atminties ir standžiojo disko informaciją, instaliuota programinę įrangą, spausdinimo įrenginius, operacinės sistemos aplinkos kintamuosius ir t.t. Tai puiki programinė įranga firmoms, kurioms svarbu turėti informaciją apie tinkle esančių kompiuterių įvairius parametrus nuo procesorių, atminties ir standžiojo disko informacijos iki įdiegtos programinės įrangos, spausdintuvų, tinklo plokščių ir dar daugiau.

Numara™ Software, Inc kompanijos siūlomas produktas Numara™ Track-It!® taip pat surenka informaciją apie programinę ir sisteminę įrangą. Synchronizuojama su Microsoft Active Directory. Turi žiniatinklio grafinę vartotojo sąsają. Automatiškai skenuoja tinklą įtraukdamas ne vien tik personalines darbo vietas, bet ir spausdinimo įrenginius, routerius, switchus ir t.t. Surenka sistemos nustatymo parametrus. Seka įsigytų licenzijų skaičius ir galiojimo laikus.

Nepavyko rasti programinės įrangos, kuri saugotų informaciją ne vien tik apie įvairius tinklo elementus (serverius, router'ius, switch'us) ir jų įvairius parametrus tačiau ir apie juose esančias paslaugas, kaip tos paslaugos susijusios su serveriais, nuo kiek serverių priklauso kokybiškas paslaugos pateikimas galutiniam vartotojui.

### 3.3. Veiklos kontekstas

Mobilaus ryšio infrastruktūros valdymo sistemos modelio veiklos diagrama pateikta 3 paveiksle.



3 pav. Sistemos veiklos konteksto diagrama

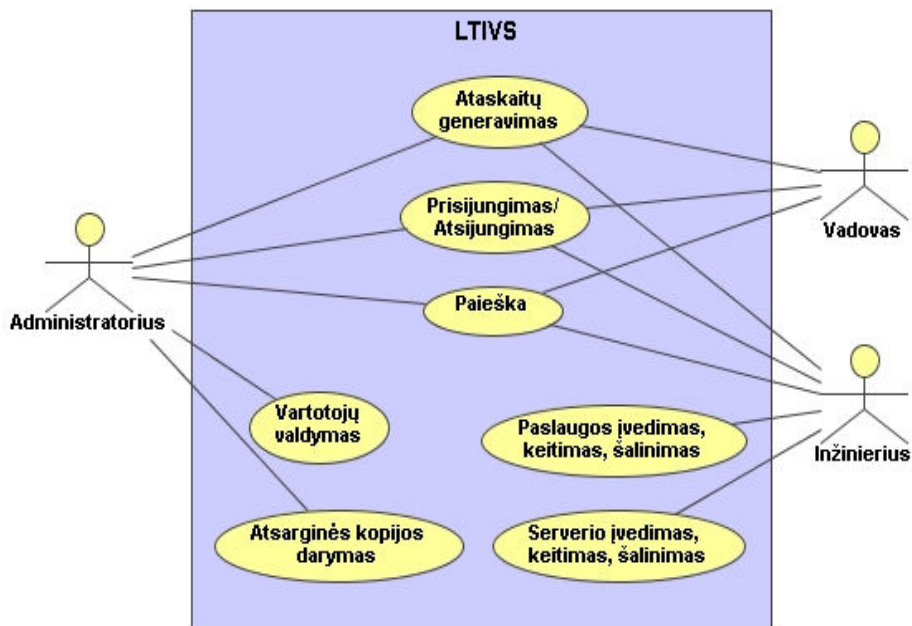
Sistemos veiklos padalinimas pateiktas lentelėje Nr. 3.

Lentelė Nr. 3 Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Inžinierius suveda atsinaujinusius duomenis apie serverį (įvedimas, modifikavimas, trynimas)	Serverių duomenys (in)
2	Inžinierius suveda atsinaujinusius duomenis apie paslaugą (įvedimas, modifikavimas, trynimas)	Paslaugų duomenys (in)
3	Administratorius suveda atsinaujinusius duomenis apie vartotojus (įvedimas, modifikavimas, trynimas)	Vartotojų valdymas (in)
4	Vadovas, administratorius ar inžinierius gauna duomenis apie serverius ar paslaugas	Ataskaitos (out)

### 3.4. Panaudojimo atvejai

Mobilaus ryšio infrastruktūros valdymo sistemos panaudojimo atvejų schema pavaizduota 4 paveiksle. Pilnas panaudos atvejų sąrašas ir jų apibūdinimas pateiktas darbo priede Nr. 2.



4 pav. Panaudojimo atvejų diagrama

Pagrindiniai sistemos aktoriai ir jų sprendžiami uždaviniai pateikiami lentelėje Nr. 4.

Lentelė Nr. 4 Pagrindinių sistemos aktorių sprendžiami uždaviniai

Vartotojas	Sprendžiami uždaviniai
Administratorius	Sistemos vartotojų valdymas ir sistemos duomenų atsarginės kopijos darymas.
Vadovas	Informacijos paieška ir ataskaitų generavimas
Inžinierius	Paslaugų ir serverių informacijos valdymas, paieška ir ataskaitų generavimas

### 3.5. Funkciniai sistemos reikalavimai

Aprašant sistemos darbą buvo įvardinti funkciniai sistemos reikalavimai. Jie pateikiami lentelėje Nr. 5.

Lentelė Nr. 5 Sistemos funkciniai reikalavimai

<b>Eil. Nr.</b>	<b>Funkcinis reikalavimas</b>	<b>Panaudojimo atvejo numeris</b>
1	Vartotojo prisijungimas prie sistemos	1
2	Ataskaitos generavimas	2
3	Paslaugos arba serverio fizinės vietos paieška	3
4	Naujų vartotojų sukūrimas sistemoje, esančių vartotojų parametrų keitimas, nebenaudojamų vartotojų trynimas iš sistemos	4
5	Duomenų apie paslaugas ir serverius atsarginės kopijos darymas	5
6	Duomenų apie paslaugas arba serverius įvedimas	6, 7
7	Duomenų apie paslaugas arba serverius keitimas	6, 7
8	Duomenų apie paslaugas arba serverius ištrynimas	6, 7
9	Rodoma prisijungusio vartotojo informacija	1 – 7
10	Keičiamos arba trinamos informacijos patvirtinimas	6, 7

### 3.6. Nefunkciniai sistemos reikalavimai

Sukurtos sistemos nefunkciniai sistemos reikalavimai pateikiami lentelėje Nr. 6.

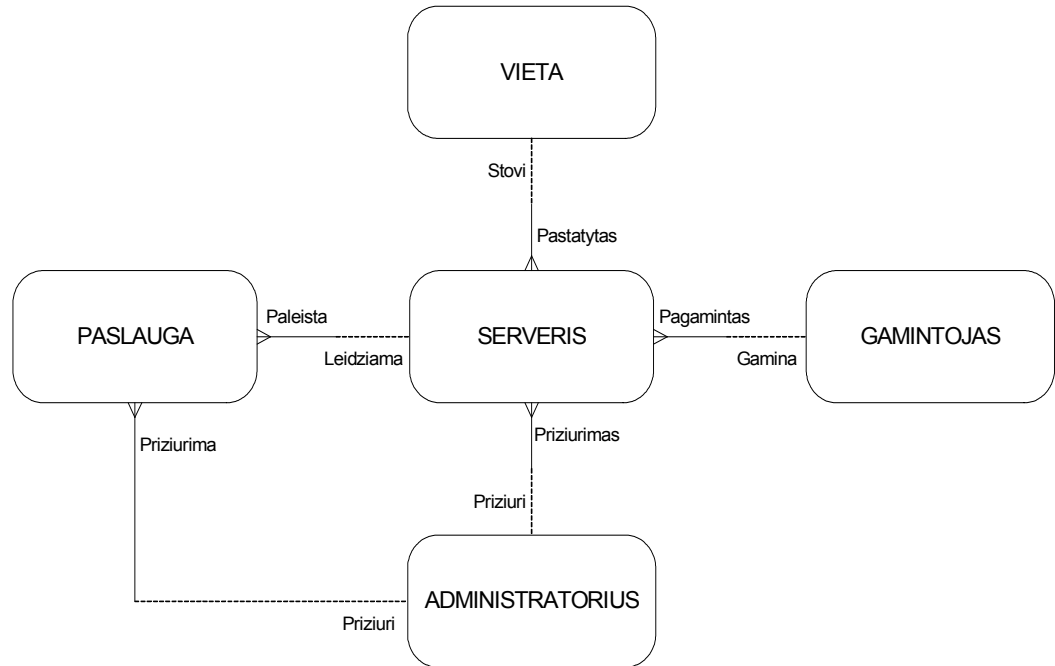
Lentelė Nr. 6 Sistemos nefunkciniai reikalavimai

<b>Eil. Nr.</b>	<b>Nefunkcinis reikalavimas</b>	<b>Panaudojimo atvejo numeris</b>
1	Neįkyri sąsaja	1, 2, 4 – 7

<b>Eil. Nr.</b>	<b>Nefunkcinis reikalavimas</b>	<b>Panaudojimo atvejo numeris</b>
2	Sąsaja anglų ir lietuvių kalba	1 – 7
3	Sistema turi būti nesudėtinga naudotis sistemos vartotojams	1 – 7
4	Informacijos paieškos laikas	3
5	Sistemos veikimo nepertraukiamumas	5
6	Sistemos veiksnumas	1 – 7
7	Sistemos daugiakalbiškumas	1 – 7
8	Sistemos konfidencialumas	1
9	Sistemos duomenų vientisumas	2 – 7
10	Priimtini komponentai	1 – 7
11	Politinis korektiškumas	1 – 7
12	Įstatymu laikymasis	1 – 7

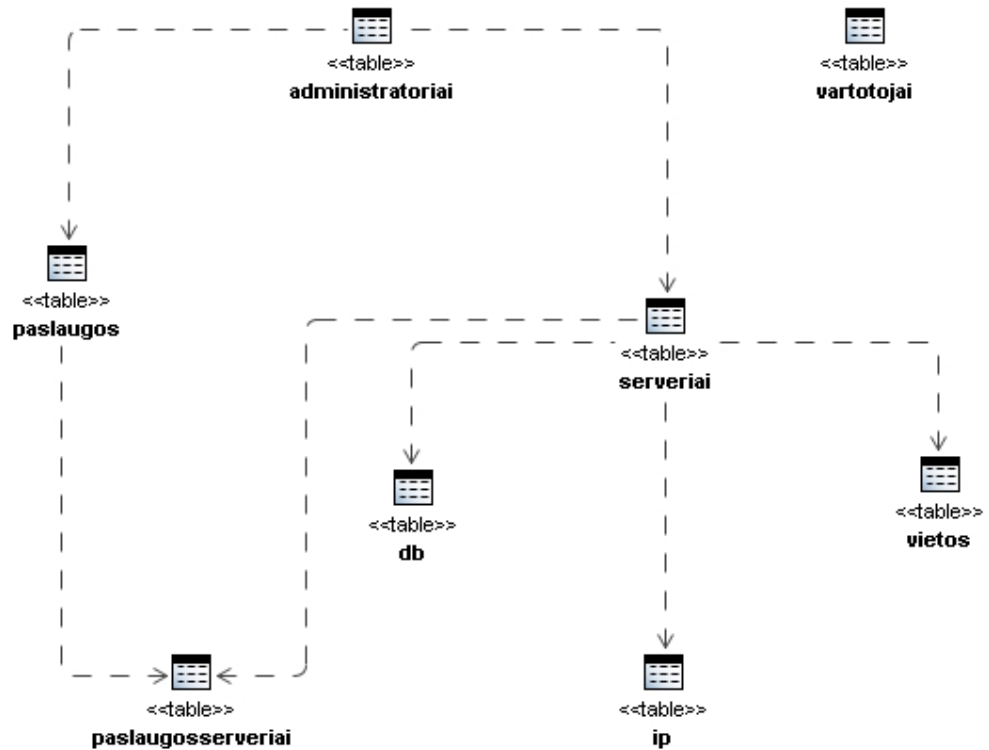
### 3.7. Reikalavimai duomenims

5 paveiksle pavaizduota supaprastinta sistemos esybių–ryšių diagrama.



5 pav. Esybių – ryšių diagrama

6 paveiksle pateikiamas duomenų bazės modelis.



6 pav. Duomenų bazės modelis

Lentelėje Nr. 7 pateikiamas duomenų bazės esybių aprašymai.

Lentelė Nr. 7 Duomenų bazės esybės parašymai

Esybė	Aprašymas
Serveris	Saugoma informacija apie serverius
Paslauga	Saugoma informacija apie klientams teikiamas paslaugas
Administratorius	Saugoma informacija apie paslaugas ir serverius prižiūrinčius administratorius
Gamintojas	Saugoma informacija apie serverių gamintojus
Vieta	Saugoma informacija apie serverio pastatymo vietas
IP	Saugoma informacija apie serverių IP adresus
DB	Saugoma informacija apie serveriuose įdiegtas duomenų bases



Esybė	Aprašymas
Vartotojai	Saugoma informacija apie sistemos vartotojus
PaslaugosServeriai	Saugoma informacija apie paslaugų ryšius su serveriais

Sukurtos sistemos kokybė aprašyta lentelėje Nr. 8.

Lentelė Nr. 8 Sistemos kokybės įvertinimas

Eil. Nr.	Kriterijus	Žymėjimas	Svoris (S)	Įvertinimas (I)
1	Korektiškumas	A	10	
2	Patikimumas	B	9	
3	Efektyvumas	C	4	
4	Integralumas	D	7	
5	Panaudojamumas	E	4	
6	Palaikomumas	F	5	
7	Testuojamumas	G	7	
8	Lankstumas	H	3	
9	Pernešamumas	I	1	
10	Pakartotinis panaudojamumas	J	3	
11	Sąsajos galimybė	K	4	
			$\Sigma = 57$	

Sistemos įvertinimą ( $Z_x$ ) apskaičiuosime pagal formulę:

$$Z_x = \frac{S_A I_A + S_B I_B + S_C I_C + S_D I_D + S_E I_E + S_F I_F + S_G I_G + S_H I_H + S_I I_I + S_J I_J + S_K I_K}{57}$$

Čia X nurodo kokio tipo vartotojas buvo apklaustas (Administratorius, vadovas ar inžinierius). Jei apklaustų vartotojų bus daugiau nei 1, tai galutinį sistemos įvertinimą (Z) apskaičiuosime pagal formulę:

$$Z = \frac{\sum_{i=1}^n Z_x}{n}$$

### 3.8. Projekto rizikos

Vykdam projektą yra numatytas rizikos valdymas, kurio metu stengiamasi sumažinti rizikos atsiradimo tikimybę, bei minimizuoti jų neigiamą poveikį projektui.

#### 3.8.1. Sistemos kūrimo rizikos

Lentelėje Nr. 9 pateikiame galimas sistemos kūrimo metu iškiliančias rizikas.

Lentelė Nr. 9 Galimos sistemos kūrimo rizikos

Eil. Nr.	Faktorius	Tikimybė	Nuostoliai/Rezultatai
1	Vadovavimo praktikos stoka	Didelė	Vėluojantis projektas
2	Netikslus kainų įvertinimas	Vidutinė	Pasikeitę projekto kaštai
3	Pasikeitę užsakovo reikalavimai	Vidutinė	Vėluojantis projektas
4	Perdėtai suspaustas sistemos kūrimo grafikas	Maža	Vėluojantis projektas
5	Žemas produktyvumas	Vidutinė	Vėluojantis projektas
6	Stichinės nelaimės	Labai maža	Vėluojantis projektas
7	Infliacija	Maža	Pasikeitę projekto kaštai

#### 3.8.2. Rizikų valdymo planas

Lentelėje Nr. 10 pateikiame rizikų valdymo planą.

Lentelė Nr. 10 Rizikų valdymo planas

Eil. Nr.	Faktorius	Sprendimas
1	Vadovavimo praktikos stoka	Išklaustyti vadovavimo kursus; Pasamdyti patarėjus.
2	Netikslus kainų įvertinimas	Numatyti rezervą neplanuotoms išlaidoms

<b>Eil. Nr.</b>	<b>Faktorius</b>	<b>Sprendimas</b>
3	Pasikeitę užsakovo reikalavimai	Informuoti užsakovą apie pailgėjusį galutinio sistemos pristatymo terminą. Perskaičiuoti išlaidas, susijusias su projekto kūrimu.
4	Perdėtai suspaustas sistemos kūrimo grafikas	Papildomai pasamdyti specialistų sistemos kūrimo paspartinimui.
5	Žemas produktyvumas	Pasamdyti aukštesnės kvalifikacijos specialistų sistemos kūrimo paspartinimui
6	Stichinės nelaimės	Apdrausti projektą draudimo bendrovėje nuo stichinių nelaimių
7	Infliacija	Numatyti rezervą neplanuotoms išlaidoms

### 3.9. Architektūros specifikacija

Architektūra pateikiama laikantis UML nustatytų standartų bei atsižvelgiant į RUP rekomendacijas. Naudojami šie pagrindiniai sistemos architektūrą aprašantys vaizdai ir modeliai, kuriais remiantis jie buvo projektuojami.

Lentelė Nr. 11 Vaizdai ir modeliai

<b>Vaizdai</b>	<b>Modeliai</b>
Panaudojimo atvejų vaizdas	PA diagrama (angl. Use case). Žr. 3.4 skyrių.
Sistemos statinis ir dinaminis vaizdas	Klasių diagrama (angl. Class) Veiklos diagrama (angl. Activity) Sekų diagrama (angl. Sequence) Bendradarbiavimo diagrama (angl. Collaboration) Būsenų diagrama (angl. State)
Išdėstymo vaizdas	Komponentų diagrama (angl. Component)

Vaizdai	Modeliai
	Išdėstymo diagrama (angl. Deployment)
Duomenų vaizdas	Duomenų bazės modelis (angl. Database)

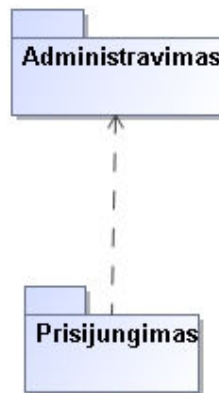
Šio projekto tikslas – lokalaus tinklo infrastruktūros valdymo sistemos sukūrimas. Projektas yra tiriamojo pobūdžio, todėl projekto pabaigoje turės būti pateiktos išvados dėl problemos sprendimo koncepcijos tinkamumo ir, jeigu to prireiks, alternatyvių sprendimų pasiūlymai.

Svarbiausi užsakovo suformuoti išankstiniai apribojimai sprendimui:

- Turi būti naudojama kliento – serverio architektūra
- Serverio dalis turi veikti Windows XP platformoje
- Kliento funkcijos turi būti pasiekiamos naudojantis naršyklėmis
- Duomenys turi būti saugomi reliacinėje duomenų bazėje
- Realizacijos modelis

### 3.9.1. Sistemos statinis vaizdas

Programinė įranga skaidoma į tokius paketus: vartotojo sąsajos ir sistemos.



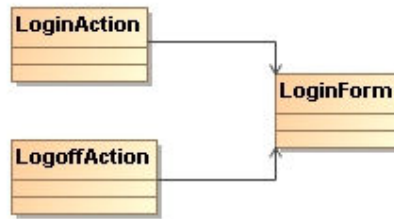
7 pav. Sistemos išskaidymas į paketus

Pateikiamas trumpas sistemos paketų aprašymas:

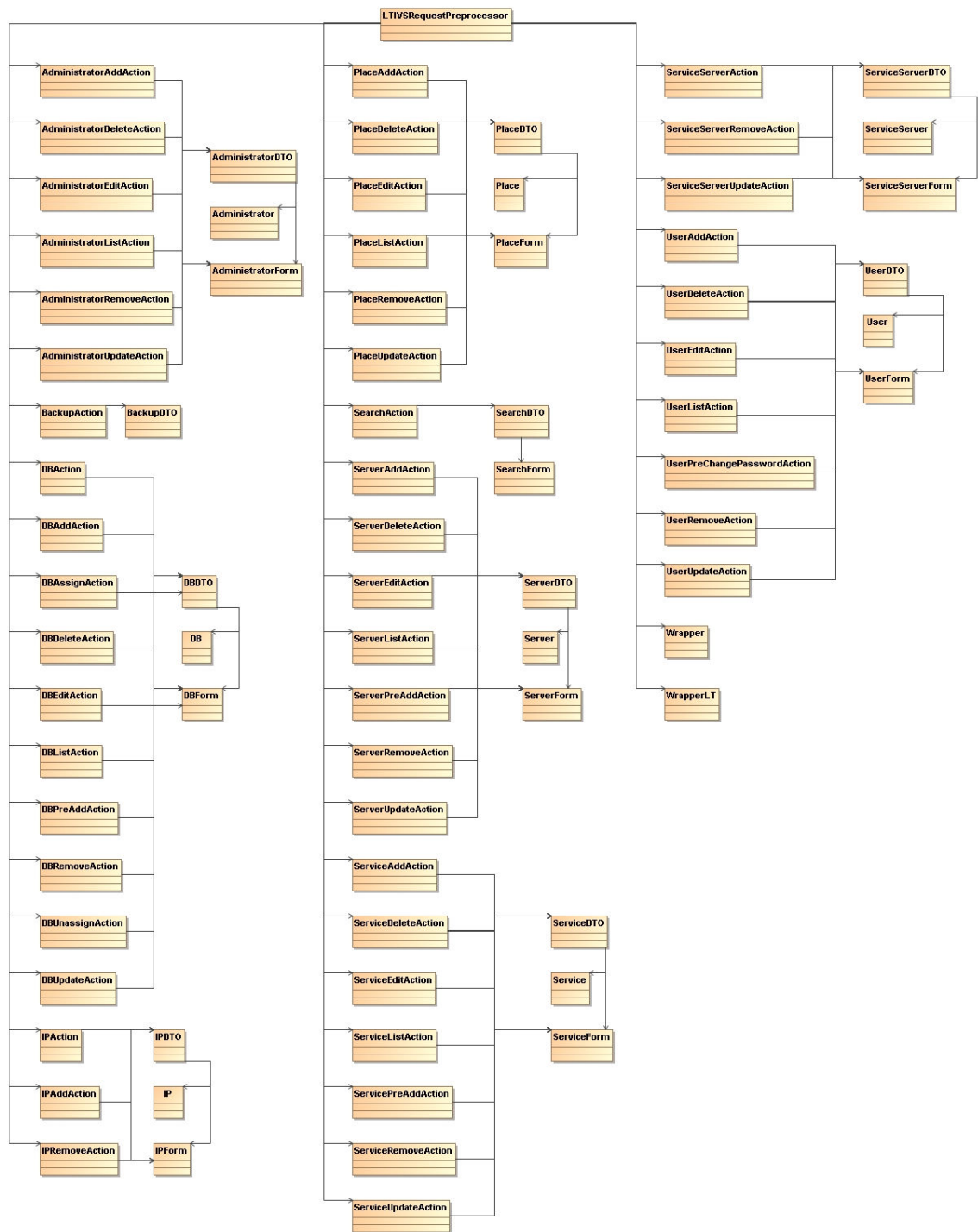
- Administravimas – sistemos dalis, kurios pagalba yra administruojama visa reikalinga informaciją (gamintojai, serveriai, paslaugos ir t.t.). Šis paketas naudoja paketus ataskaitos bei prisijungimas.

- Prisijungimas – sistemos dalis, kurios pagalba yra atliekamas prisijungimas (prisijungimo metu nustatoma vartotojo rolė bei atitinkamai aktyvuojamas sistemos kontekstas).

Kiekvienam paketui yra sudaryta klasių diagrama. Paketų klasių diagramos pateikiamos 8 ir 9 paveiksluose.



8 pav. Paketo Prisijungimas klasių diagrama

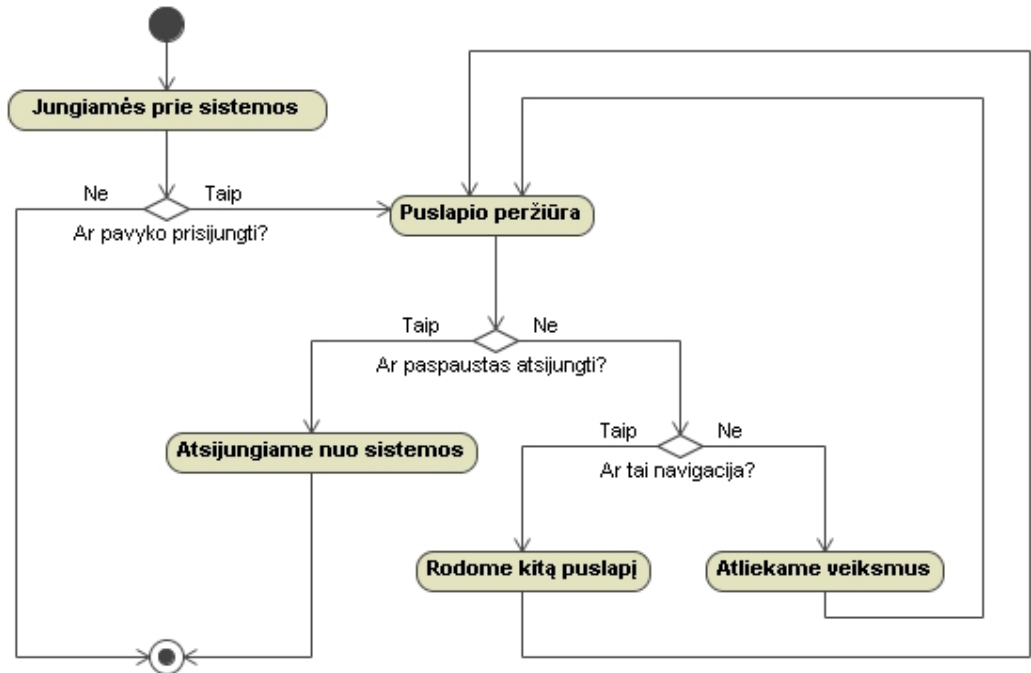


9 pav. Paketo Administravimas klasių diagrama

### 3.9.2. Sistemos dinaminis vaizdas

Mobilaus ryšio infrastruktūros valdymo sistemos sąsaja pasiekama žiniatinklio pagalba. Visi veiksmai turi identišką darbo eigos struktūrą: prisijungus prie sistemos pateikiamas pradinis vartotojo puslapio turinys iš kurio, navigacijos pagalba, galima pereiti

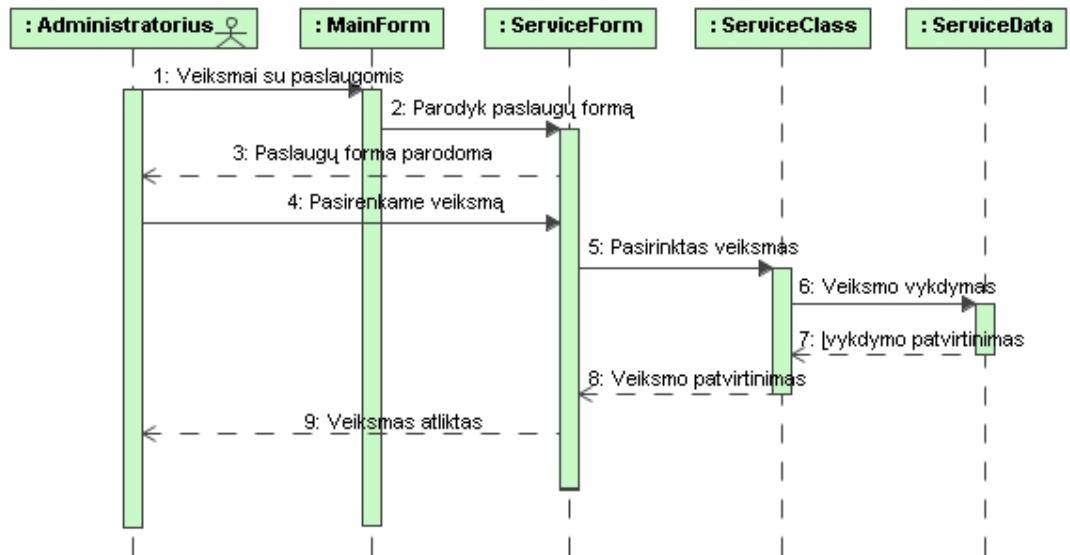
prie kitų puslapių arba atlikti tam tikrus veiksmus su puslapyje esančiais objektais (juos sukurti, modifikuoti ar ištrinti). 10 paveiksle pateikiamas veiklos diagramos šablonas.



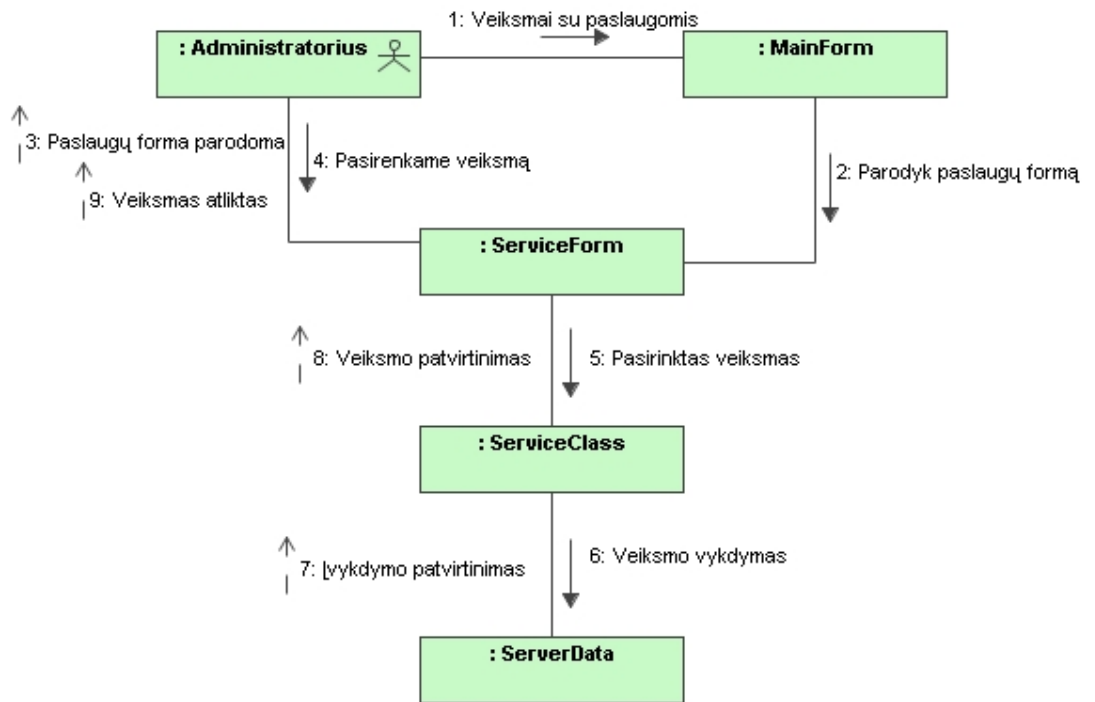
10 pav. Veiklos diagramos šablonas

### 3.9.3. Sekų diagramos

Žemiau taip pat pateikiami sekos ir bendradarbiavimo diagramų šablonai. Šablonuose panaudotas abstraktus žodis "veiksmas", kuris gali būti vienas iš veiksmų: įvedimas, modifikavimas ar trynimas. Pasirenkant veiksmą yra tuo pačiu perduodami ir reikalingi operacijai duomenys (nauji duomenys įvedimo veiksmo atveju, keičiami duomenys modifikavimo veiksmo atveju ar nurodome ką trinsime trynimo veiksmo atveju).



11 pav. Sekos diagrama



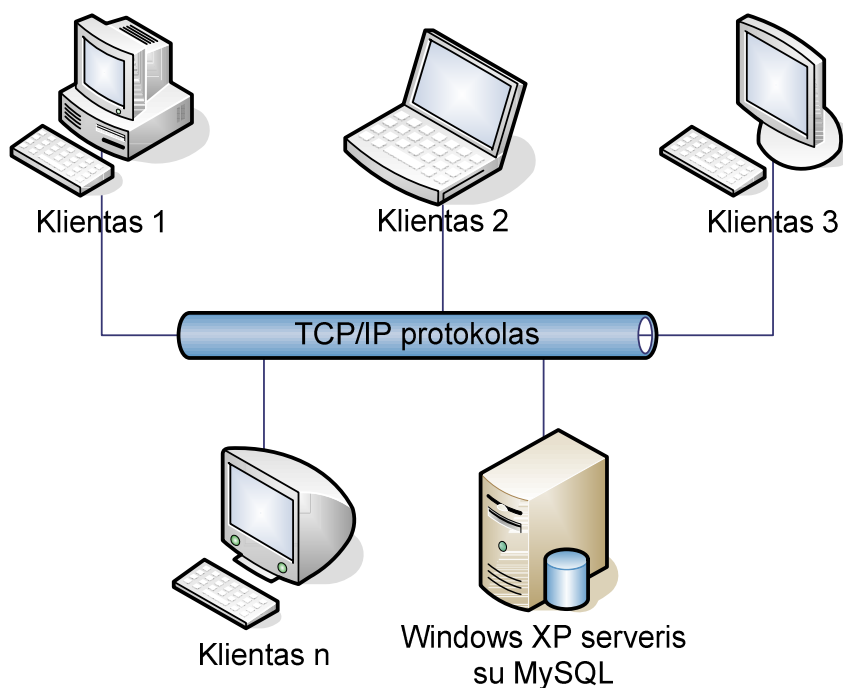
12 pav. Bendradarbiavimo diagrama

Būsėnų diagrama nėra pateikiama, nes žiniatinklio sąsaja pagrįstos sistemos retai aprašomos šiomis diagramomis.



### 3.9.4. Sistemos išdėstymo vaizdas

13 paveiksle pavaizduotas sistemos išdėstymas. Duomenų bazių valdymo sistema gali veikti ir kliento kompiuteryje.



13 pav. Sistemos išdėstymo vaizdas

Minimalūs reikalavimai techninei įrangai tiesiogiai priklauso nuo naudojamos operacinės sistemos Microsoft Windows versijos bei jai keliamų minimalių reikalavimų. Minimalūs ir rekomenduojami reikalavimai techninei įrangai naudojant Microsoft Windows XP yra pateikiami lentelėje Nr. 12 [20]:

Lentelė Nr. 12 Microsoft Windows XP reikalavimai

Microsoft Windows XP	Minimalūs	Rekomenduojami
Kompiuteris ir procesorius	233 MHz	1 GHz
Atmintis	128 Mb	1 Gb
Kietasis diskas	1,5 Gb	40 Gb
Lankstus įrenginys	CD-ROM arba DVD-ROM	DVD-RW
Monitorius	VGA	SVGA

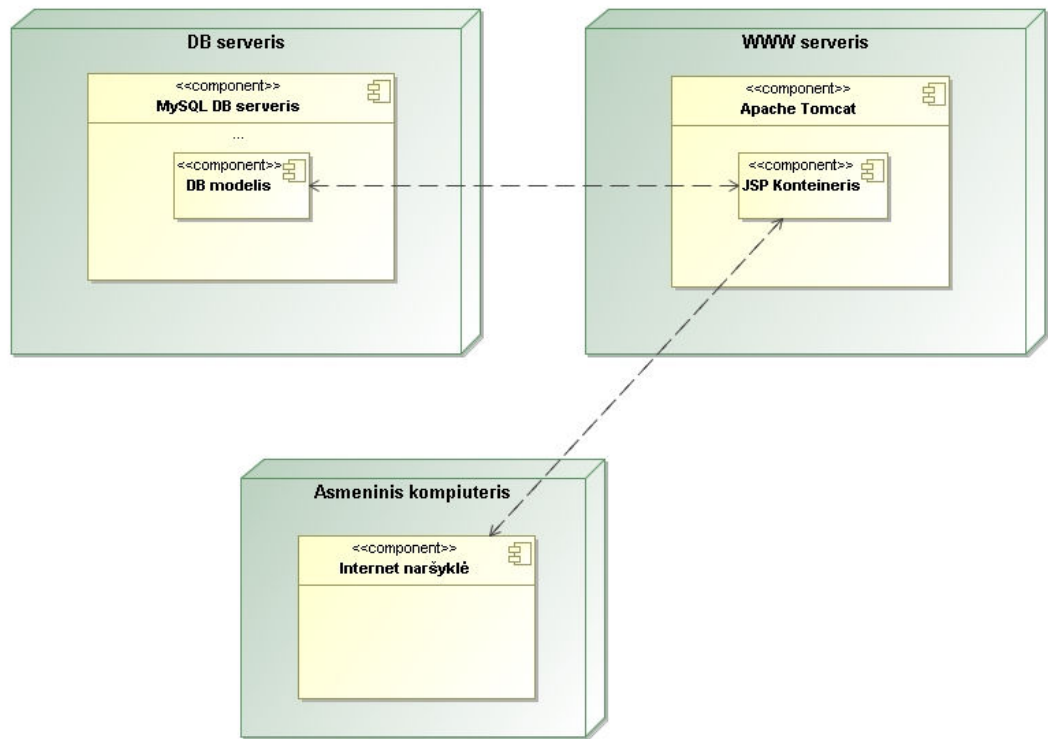
Serveryje turi būti įdiegta tokia programinė įranga:

- MySQL 5.0

- Apache Tomcat 6.0
- Struts 1.2
- Java 1.5.0

### 3.9.5. Komponentų diagrama

Sistemos komponentų diagrama pateikiama 14 paveiksle.



14 pav. Sistemos komponentų diagrama

### 3.10. Sistemos įgyvendinimas

Sistemos įgyvendinimui buvo naudojama Windows XP SP2 [20] operacinė sistema. Sistemos kodas buvo rašomas Eclipse 3.3.0 versija naudojant Java EE 5-ą versiją [1, 10, 16] ir transliuojamas naudojant Apache Ant [3]. Buvo naudojamas Apache Tomcat 6.0 [32] žiniatinklių serveris. Duomenų bazių valdymo sistema – MySQL 5.0 [21].

Reikia paminėti, kad aukščiau išvardinta programinė įranga yra kokybiška ir nemokama[33].

### 3.10.1. Struts karkasas

Pagrindinis valdiklio komponentas yra servlet'as (ActionServlet klasė), kuris atsakingas už užklausų apdorojimą naudojant specialų XML failą Struts-config.xml. Tai programos logikos kontrolės pagrindas. Šį failą lengva suprasti, net jeigu programa yra didelė. Ten saugomas universalusis išteklių identifikatorius (angl. URI) [2] ir nurodoma ActionServlet'ui kaip jisai turi elgtis.

Vaizdo komponente Struts naudoja JSP aplinką, kuri leidžia naudoti įvairią kombinaciją iš statinių HTML ir dinaminio turinio, kuris keičiamas priklausomai nuo specialių veiksmo žymių. Struts naudoja įvairias žymes: Bean, HTML, Logic, Nested, Template ir Tiles.

Viena pagrindinių Struts naudojimo savybių yra ta, kad modelio komponentas neturi jokio vaizdo komponento kodo. Taip pat ir JSP failai, kurie neturi jokio nukreipimo (angl. forward) logikos kodo. Visa elgsena yra valdoma Struts-config.xml failo pagalba.

Žemiau pateikiame ištrauką iš Struts-config.xml failo:

```
...
<action path="/login" type="org.renatasnarmontas.Itivs.LoginAction"
    name="loginForm" scope="request" input="page.login">
    <forward name="success" path="page.welcome" />
    <forward name="error" path="page.login" />
</action>
...
```

Šis veiksmas yra atliekamas, kai URL baigiasi „/login“. Užklausa ateina iš Tiles karkaso puslapio vardu „page.login“. Priklausomai nuo to, kokia reikšmė yra gražinama, yra parodomas arba „page.welcome“ arba „page.login“ Tiles karkaso puslapis.

### 3.10.2. Tiles karkasas

Tiles karkasas yra integruotas į Struts karkasą, tačiau nėra aktyvuojamas pagal numatytąsias reikšmes (angl. default). Šis karkasas yra naudingas, nes leidžia kurti pernaudojamus pavaizdavimo komponentus, pvz. išdėstymą, HTML, paveikslukus ar kitus vizualesius komponentus.

Pagrindinis MRIVS išdėstymas yra padalintas į keturis regionus: antraštė (angl. header), meniu, turinys (angl. body, content) ir puslapinė poraštė (angl. footer) [34]. Tai leidžia lengvai pavaizduoti informaciją kiekviename iš šių regionų, pvz. prisijungusio vartotojo vardą antraštėje. Visa pavaizdavimo struktūra saugoma XML faile tiles-def.xml.

Žemiau pateikiame ištrauką iš failo tiles-def.xml:

```
<tiles-definitions>
    <definition name="base.definition" path="/pages/siteLayout.jsp">
        <put name="header" value="/pages/header.jsp" />
        <put name="navigation" value="/navigation.do" />
        <put name="footer" value="/pages/footer.jsp" />
    </definition>

    <!-- Tiles Definition of Login page-->
    <definition name="page.login" extends="base.definition">
        <put name="title" value="Login page" />
        <put name="body" value="/pages/login.jsp" />
    </definition>

    ...
</tiles-definitions>
```

### 3.10.3. Saugumas

Svarbi sistemos dalis yra saugumas. Sistemoje naudojami konfidencialūs duomenys todėl kiekvienam vartotojui suteikiama jo prieigos rolė. Šiuo metu naudojama sistemoje tik dviejų tipų rolės: administratoriai ir paprasti vartotojai. Esant reikalui naują rolę įdėti nesunku. Sistema buvo nuo pat pradžių projektuojama taip, kad ją būtų lengva plėsti.

Kiekvienas vartotojas, prisijungdamas prie MRIVS, gauna savo rolę. Kai tik vartotojas užsiregistruoja sistemoje, Session objekte mes saugome jo prisijungimo vardą iki tol, kol vartotojas atsijungia arba neatlieka jokių veiksmų per iš anksto numatytą laiką.

Mes naudojame žymę, priklausomai nuo to reikia vartotojui rodyti informacijos dalį ar nereikia, pvz.:

```
<%
String role = (String) session.getAttribute("ROLE");
if ("administrator".equals(role))
{
%>
<br />
<b><i>
<bean:message key="navigation.page.adminmenu" />
</i></b>
<br />
<% } %>
```

Duotame pavyzdyje parodome papildoma administratorių meniu, jei vartotojo rolė yra „administrator“.

### 3.10.4. Internacionalizacija

Kiekviena naršyklė užklausoje siunčia informacija apie šalį ir kalbą, pvz. lt\_LT, lv\_LV, ru\_RU, en\_EN ir t.t. Resursų faile yra nurodomos poros: raktas-reikšmė. Kiekviena reikšmė bus parodyta ta kalba, kokia yra nurodyta naršyklės siunčiamoje užklausoje.

Projektuojamoje sistemoje anglų kalba yra numatytoji. Jei užklausoje yra nurodyta tokia šalis ir kalba, kurios resursų failo dar nėra sukurta, tuomet visi pranešimai yra parodomi anglų kalba.

Žemiau pateikiame ištrauką iš numatytojo failo MessageResources.properties:

```
# -- Login --
login.title=LANIMP - Please login
login.message=Please login
login.username=Username:
login.password=Password:
login.button.signon=Log In
```

Ta pati ištrauka iš failo MessageResources\_lt.properties:

```
# -- Login --
login.title=LTIVS – Prašau prisijungti
login.message=Prašau prisijungti
login.username=Vartotojas:
login.password=Slaptažodis:
login.button.signon=Įėjimas
```

### 3.10.5. Duomenų bazės kūrimas

Duomenų bazė buvo sukurta MySQL 5.0 duomenų bazių valdymo sistemoje. Žemiau pateikiame ištrauką iš duomenų bazės kūrimo skripto:

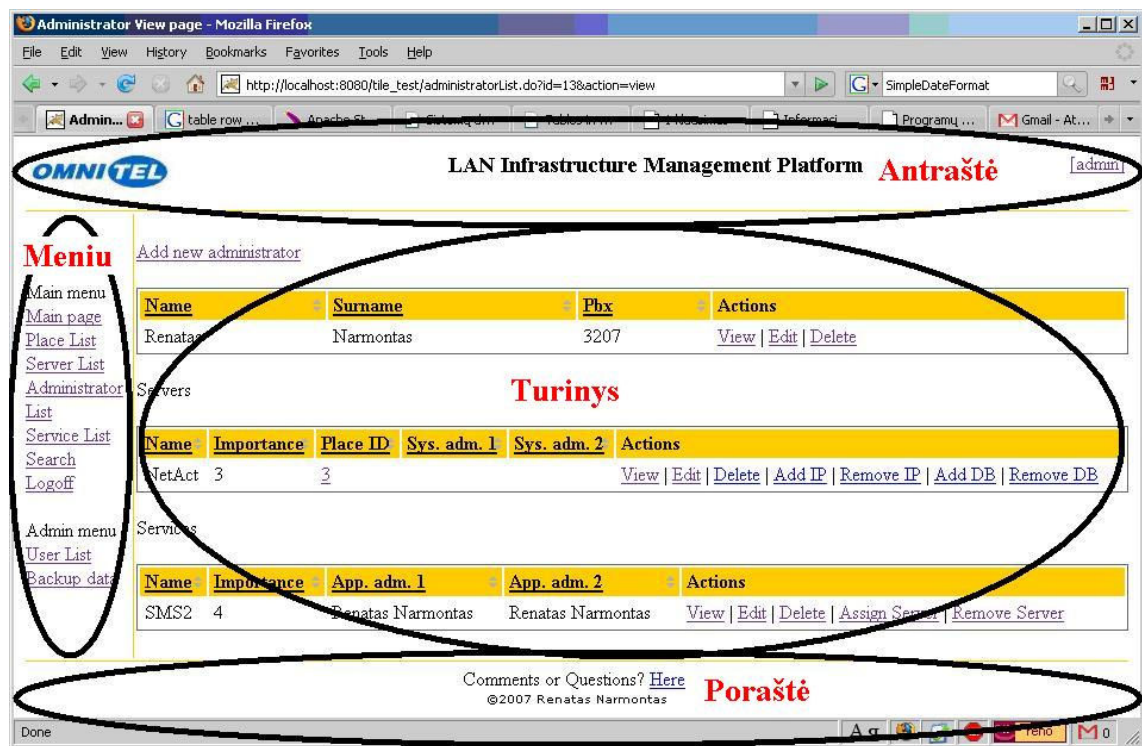
```
DROP DATABASE IF EXISTS Itivs; CREATE DATABASE Itivs;
...
CREATE TABLE Itivs.db
(db_id int(10) unsigned NOT NULL auto_increment PRIMARY KEY,
serveriai_id int(10) unsigned NULL,
pavadinimas varchar(64) NULL,
aprasymas varchar(1000) NULL,
svarba smallint(1) NULL,
tipas varchar(20) NULL,
versija varchar(64) NULL,
pataisymai varchar(1000) NULL,
garantija varchar(1000) NULL,
pirmas_dba_admin int(10) unsigned NULL,
antras_dba_admin int(10) unsigned NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
...
```

Šis kodas sukuria per naują duomenų bazę, ištrindamas seną ir sukurdamas naują. Aukščiau pateiktame pavyzdyje parodyta kaip sukurti ltivs.db lentelę su visais jos laukais, naudojant SQL [2] kalbą. Pirminis raktas yra db\_id, kuris naudojamas vienareikšmiškai identifikuoti kiekvienam lentelės įrašui.

### 3.10.6. Grafinė vartotojo sąsaja

Grafinė vartotojo sąsaja buvo kuriama naudojant JSP technologiją, kuri leidžia dinamiškai generuoti HTML failus. Šiai sąsajai unifikuoti buvo naudojamas Tiles karkasas. Jisai suteikia galimybę pernaudoti įvairius vizualius komponentus.

Naudojant Tiles karkasą buvo sukurti pernaudojami pavaizdavimo komponentai. 15 paveiksle matome puslapio išdėstymo momentinę ekrano kopiją (angl. screenshot) [2]



15 pav. Išdėstymo momentinė ekrano kopija

Kaip parodyta 15-ame paveiksle, išdėstymas susideda iš keturių regionų: antraštės, meniu, turinio ir puslapiinės poraštės. Čia yra panaudojamas komponentų pernaudojamumas. Mes visiems puslapiams galime naudoti tą pačią antraštę, meniu ir puslapiinę poraštę. Taip pat galima parametrų pagalba perduoti informaciją į šiuos regionus pvz.: prisijungusio vartotojo vardas (šiuo atveju admin).

### **3.11. Sistemos testavimas**

Sistemos testavimo tikslas buvo išnagrinėti testavimo galimybes bei metodus. Testavimo metu buvo naudojami vienetų, integraciniai, priėmimo ir aukšto lygmens testavimo metodai. Testavimo metu buvo patikrinti įvairūs elementai: grafinė vartotojo sąsaja, sistemos duomenys.

#### **3.11.1. Testuojama programinė įranga**

Testuojamą programinę įrangą galima suskaidyti į tokius paketus: prisijungimas, administravimas ir ataskaitos.

Pasitelkus prisijungimo paketą galima prisijungti prie sistemos. Juo yra nustatoma, ar vartotojas yra registruotas sistemoje, ir yra nustatomas registruoto vartotojo vaidmuo (pagal kurį yra aktyvuojama sistemos darbo aplinka).

Naudojant administravimo paketą yra atliekamas paslaugų, serverių duomenų, administratorių, sistemos vartotojų ir kitų duomenų administravimas.

Naudojant ataskaitų paketą yra paimami duomenys iš duomenų bazių valdymo sistemos.

#### **3.11.2. Testavimo įrankiai ir aplinka**

Testavimui buvo naudojama Microsoft Windows XP operacinė sistema su įdiegta tokia programine įranga:

- MySQL – duomenų bazių valdymo sistema;
- Microsoft Office – ataskaitų rašymui [20];
- Internet Explorer – tinklo naršyklė [20];
- Apache Tomcat – JSP konteineris;
- Java JDK – Java programavimo rinkinys [7];
- Eclipse – programavimo ir testavimo aplinka;
- Parasoft Jtest – testavimo aplinka.

#### **3.11.3. Testų metodų aprašymas**

Kuriama sistema buvo testuojama vienetų lygiu. Vienetu yra laikoma mažiausia testuojama programos dalis. Buvo naudotas baltos dėžės metodas.

Taip pat buvo testuojama integracijos lygiu. Šio testo metu buvo analizuojama kaip sąveikauja sistemos tarpusavyje. Tai leido įvertinti ir užtikrinti nepriekaištingą sistemos

veikimą. Integracijos lygio testavimo metu buvo testuojama vartotojo sąsaja. Buvo naudotas tiek juodos dėžės, tiek baltos dėžės metodas.

Priėmimo testavimo metu buvo tikrinama, ar sistema atitinka reikalavimų specifikaciją. Sistemos funkcijos buvo lyginamos su reikalavimų specifikacijoje aprašytais funkcijomis ir buvo tikrinta, ar sukurta programinė įranga pateikia laukiamus rezultatus. Buvo naudotas juodos dėžės metodas.

Aukšto lygio testavimas buvo atliekamas įdiegus programinę įrangą užsakovo įmonėje. Jo metu buvo testuojamas:

- Saugumas;
- Našumas;
- Atstatymas.

Saugumo testų metu buvo tikrinta, kad neautorizuoti vartotojai negalėtų pasinaudoti sistema. Prieiga suteikiama tik autorizuotiems vartotojams.

Našumo testų metu buvo tikrintas programinės įrangos darbas su dideliais duomenų kiekiais.

Atstatymo testų metu buvo tikrinta, ar nėra sugadinami duomenys programinės įrangos lūžimo metu.

### 3.11.4. Testavimo rezultatai ir išvados

Buvo sukurti 23 automatiniai testavimo atvejai. Testų vykdymo rezultatus matome toliau pateiktame programos išvedinyje (angl. output) [2].

```
...../pages/siteLayout.jsp
/pages/siteLayout.jsp
/pages/siteLayout.jsp
.....
Time: 2.188

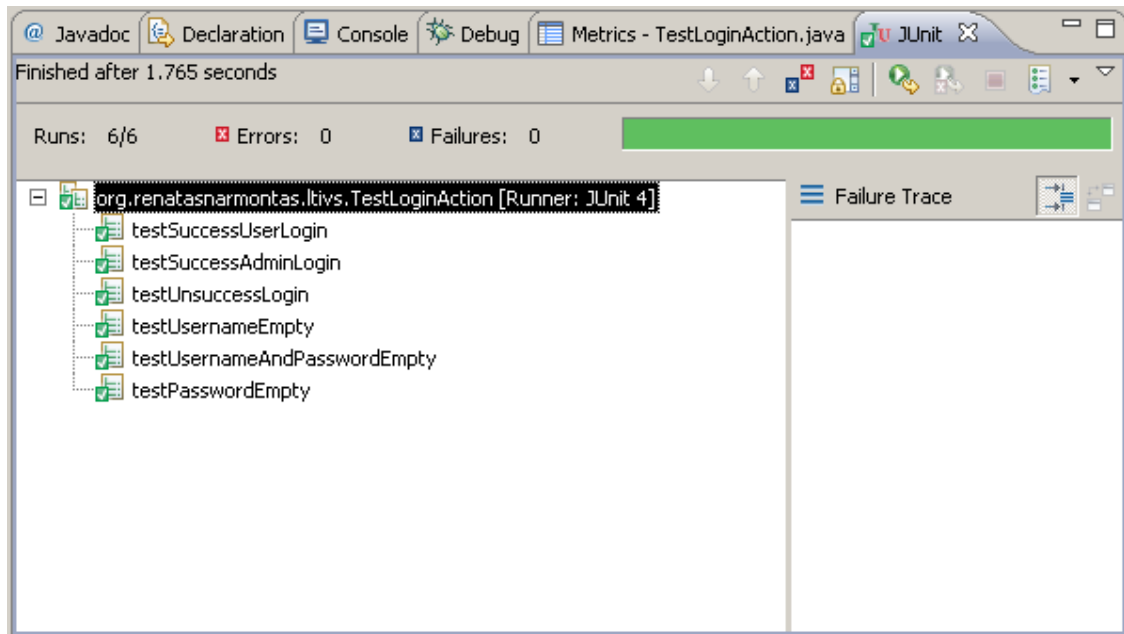
OK (23 tests)
```

Visi testai buvo įvykdyti sėkmingai. Kiekvienas taškas išvedinyje reiškia, kad testas davė tokį rezultatą, kokio ir buvo tikėtasi. Gavus rezultatą ne tokį, kokio buvo laukiama, būtume pamatę raidę „F“ (angliško žodžio Fail pirma raidė).

Integraciniai testai buvo testuojami pasinaudojant JUnit įskiepiu (angl plugin) Eclipse programai.



Vieno iš integracinių testų vykdymo rezultatus matome 16 paveiksle.



16 pav. Login lango testavimas

Visų testų rezultatai buvo teigiami. Todėl galime teigti, kad testuotos klasės ir SQL funkcijos veikia pagal planuotus funkcinis reikalavimus.

### 3.12. Sistemos dokumentacija

Sukurta sistema turi du skirtingus vartotojų lygius: paprastus vartotojus ir administratorius.

Paprasti vartotojai gali atlikti tokius veiksmus:

- Vietos įdėjimą, modifikavimą, ištrynimą;
- Serverių įdėjimą, modifikavimą, ištrynimą, IP adreso priskyrimą, IP adreso ištrynimą, duomenų bazės priskyrimą, duomenų bazės ištrynimą, informacijos apie vietą parodymą;
- Administratorių įdėjimą, modifikavimą, ištrynimą, jų prižiūrimų paslaugų duomenų bazių ir serverių nustatymą bei parodymą;
- Paslaugų įdėjimą, modifikavimą, ištrynimą, paslaugos priskyrimą serveriui, paslaugos ištrynimą iš serverio, pilnos informacijos apie paslaugą parodymą;
- Įvairios informacijos paiešką;
- Slaptažodžio keitimą.

Tik administratoriai gali sukurti sistemos vartotojus, juos modifikuoti ar užblokuoti/atblokuoti bei ištrinti iš sistemos.

Kiekvienam vartotojui kalba (anglų arba lietuvių) parenkama automatiškai pagal numatytuosius Windows interneto naršyklės nustatymus. Lengva pridėti naują kalbą.

Pilna sistemos dokumentacija pateikiama kompaktiniame diske.

## 4. Žiniatinklių programų kūrimo karkasų tyrimas

Šiame skyriuje tiriami įvairūs žiniatinklių programų kūrimo karkasai, atliekamas jų vertinimas pagal iš anksto pasirinktus kriterijus.

Beveik visi modernūs žiniatinklių programų kūrimo karkasai naudoja MVV architektūra. Verslo logikos ir pavaizdavimo sluoksniai yra nepriklausomi vienas nuo kito. Valdiklis koordinuoja vartotojo užklaudas ir serverio pusėje atliekamus veiksmus. Tai jau tapo „*de facto*“ žiniatinklių programavime. [18]

Palyginsime Struts karkasą su kitais atviro kodo žiniatinklių kūrimui naudojamais karkasais. Palyginsime šiuos karkasus:

- JSF [17];
- Spring MVC [28];
- Stripes [29];
- Struts [25];
- Struts 2 [25];
- Tapestry [30];
- WebWork [35];
- Wicket [36].

### 4.1. Žiniatinklių programų kūrimo karkasai

Žemiau pateikiame populiariausių žiniatinklių programų kūrimo karkasų pagrindinius privalumus ir trūkumus.

#### 4.1.1. JSF

JSF karkaso pagrindiniai privalumai yra šie:

- Standartinis J2EE
- Greitas ir lengvas programų kūrimas
- Įvairus navigacijos karkasas

JSF karkaso pagrindiniai trūkumai yra šie:

- Nesusidoroja su REST ir saugumu
- Nėra vieno gamintojo.

### 4.1.2. Spring MVC

Spring MVC karkaso pagrindiniai privalumai yra šie:

- Lengva integracija su pavaizdavimo karkasais: JSP/JSTL, Tiles, Velocity, FreeMarker, Excel, XSL, PDF.
- Lengva testuoti

Spring MVC karkaso pagrindiniai trūkumai yra šie:

- Sudėtinga konfigūruoti, daug XML failų.
- Daug JSP kodo rašymo.
- Per daug lankstus. Nėra pagrindinio valdiklio.

### 4.1.3. Stripes

Stripes karkaso pagrindiniai privalumai yra šie:

- Lengvas konfigūracijos valdymas.
- Gerai dokumentuotas (lengva mokytis)
- Entuziastinga bendruomenė

Stripes karkaso pagrindiniai trūkumai yra šie:

- Maža bendruomenė
- Nepakankamai aktyviai vystomas projektas lyginant su kitais
- Įkoduoti (angl. hard-coded) [2] URL'ai į ActionBeans.

### 4.1.4. Struts

Struts karkaso pagrindiniai privalumai yra šie:

- Daug informacijos ir pavyzdžių
- Viena iš geriausių HTML tag'ų biblioteka

Struts karkaso pagrindiniai trūkumai yra šie:

- Sudėtinga perprasti ActionForms.
- Sudėtingas testavimas. Kol kas atliekamas StrutsTestCase pagalba.
- Sklando gandai, kad projektas „miręs“.

### 4.1.5. Struts2

Struts 2 karkaso pagrindiniai privalumai yra šie:

- Paprasta architektūra, lengva vystyti.
- Tag'ų biblioteką lengva pritaikyti FreeMarker arba Velocity karkasams.

- Valdikliu arba puslapiais paremta navigacija.

Struts 2 karkaso pagrindiniai trūkumai yra šie:

- Prastai organizuota dokumentacija.
- Per didelis dėmesys naujoms savybėms
- Paieškos sistemos duoda nuorodas į Struts 1.x

#### **4.1.6. Tapestry**

Tapestry karkaso pagrindiniai privalumai yra šie:

- Vieną kartą išmokus tampa labai našus
- HTML šablonai tinka dizaineriams.
- Sveika ir stipri vartotojų bendruomenė.
- Daug naujovių tarp versijų išleidimų.

Tapestry karkaso pagrindiniai trūkumai yra šie:

- Pragmatiška dokumentacija
- Sudėtinga išmokti. Mažai pavyzdžių.
- Ilgas versijų išleidimo ciklas.

#### **4.1.7. Wicket**

Wicket karkaso pagrindiniai privalumai yra šie:

- Labiau tinka Java, o ne žiniatinklių programuotojams.
- Sąveika tarp puslapių ir vaizdų.
- Aktyvi bendruomenė, palaikymas iš kūrėjų.

Wicket karkaso pagrindiniai trūkumai yra šie:

- HTML šablonai neatskirti nuo Java kodo.
- Reikalingos objektinio programavimo žinios.
- Viskas padaryta Java kalba

#### **4.1.8. WebWork**

WebWork karkaso pagrindiniai privalumai yra šie:

- Nesudėtinga architektūra. Ją lengva vystyti.
- Tag'ų biblioteką lengva pritaikyti FreeMarker arba Velocity karkasams.
- Valdikliu arba puslapiais paremta navigacija.

WebWork karkaso pagrindiniai trūkumai yra šie:

- Maža bendruomenė.
- Prastos kokybės dokumentacija.

## 4.2. Vertinimo kriterijai

Žiniatinklių programų kūrimo karkasų vertinimui naudojome šiuos kriterijus:

- Sąrašai. Ar lengva ilgus sąrašus suskaidyti puslapiais? Ar galima juos rūšiuoti?
- Žymių galimybė (angl. bookmark). Ar vartotojas gali pažymėti puslapius ir po to paprastai prie jų sugryžti?
- Patikra (angl. validation). Ar lengva panaudoti patikrą ir ar yra palaikoma kliento pusės (JavaScript) patikra?
- Testai (angl. testability). Ar lengva testuoti valdiklius nenaudojant konteinerių?
- Integracija su Spring karkasu. Ar karkasas palaiko Spring karkaso vartojimą tarpiniame sluoksnyje (angl. middle tier)?
- Internacionalizavimas. Ar palaiko i18n? Ar lengva pakeisti pranešimus valdiklyje?
- Puslapių išklotinės. Kokius puslapių išklotinių mechanizmus palaiko karkasas? (S – SiteMesh karkasas, T – Tiles karkasas).
- Įrankiai. Ar yra įrankių (angl. IDE) [2] programų kūrimui.
- Įgūdžių nauda. Ar karkaso mokėjimas padės lengviau rasti darbą?
- Darbo pasiūla. Darbuotojų, mokančių dirbti su šiuo karkasu, poreikis skelbimuose.

### 4.2.1. Karkasų vertinimas

Rūšiuojamus ir suskaidymus puslapiais duomenų sąrašus Struts, Struts 2, Spring MVC ir WebWork karkasai vaizduoja tag'u bibliotekos pagalba. Tapestry karkasas turi Table komponentę. JSF karkasas turi dataTable komponentę be numatytosios rūšiavimo galimybės. Programuotojai turi pasirašyti rūšiavimą atskirai.

Žymių galimybė WebWork ir Struts2 karkasuose realizuota vardų adresų erdvėje (angl. namespace). Struts, ir Spring karkasuose galima pilnai kontroliuoti URL. Tapestry numatytas ribotas URL panaudojimas. Wicket karkasas leidžia priskirti URL kiekvienam puslapiui (angl. mount).

JSF karkasas turi negražius numatytuosius pranešimus, tačiau lengva juos konfigūruoti. Spring ir Struts karkasai naudoja išbaigtą sprendimą – Commons Validator.

Struts 2 naudoja OGNL galingoms išraiškoms. Tapestry taip pat turi galingą patikros mechanizmą. Stripes ir Wicket karkasuose patikra atliekama tik serverio pusėje.

Testuoti Spring ir Struts 2 karkasus lengva su mock'ais (pvz. EasyMock, jMock, Spring Mocks). Tapestry karkasas yra sunkiai testuojamas nes puslapių klasės yra abstrakčios. JSF karkaso puslapių klasės lengvai testuojamos ir yra panašios į Struts 2. Wicket karkasas testams turi galingą mechanizmą – WicketTester. Stripes karkasas gali būti testuojamas MockRoundtrip pagalba. Struts karkaso testavimui naudojamas StrutsTestCase įrankis.

Visi karkasai gali sąveikauti su Spring karkasu. Struts karkasas naudoja ContextLoaderPlugin ir Support klases. WebWork turi integruotą SpringObjectFactory. Tapestry karkase Spring lengvai integruojamas Hivemind pagalba. JSF karkase – naudojant JSF-Spring bibliotekas.

Nėra priimto standarto valdiklio pranešimų internacionalizavimui. Struts. Stripes. Spring ir JSF karkasai naudoja vieną resursų failą kiekvienai kalbai. Struts2, Tapestry ir Wicket skiria atskirus failus kiekvienam puslapiui ar veiksmui.

Puslapių išklotinėms naudojami pagrindė Tiles ir/arba SiteMesh karkasai. SiteMesh karkasas gali būti naudojamas daugumoje karkasų. SiteMesh nerekomenduojamas arba nėra gerai palaikomas tik JSF, Tapestry ir Wicket karkasuose. Tiles karkasas gali būti naudojamas Struts, Struts2, Spring ir JSF karkasuose.

Struts ir JSF karkasai turi daugiausia įrankių. Stripes ir Wicket karkasai neturi oficialių programų kūrimo įrankių. Spring karkaso įrankis atlieka tik XML patikrą, nėra grafinės vartotojo sąsajos kūrimo funkcionalumo.

Struts vis dar plačiai naudojamas. JSF karkasas vis labiau populiarėja. Tapestry karkasui būtina geresnė rinkodara. Wicket ir Stripes karkasai yra praktiškai nežinomi plačiosioms masėms.

Darbo skelbimų pasiūlos grafikas pateiktas 17 paveiksle.

Žiniatinklių programų kūrimo karkasų palyginimas pateiktas lentelėje Nr. 13.

Lentelė Nr. 13 Žiniatinklių programų kūrimo karkasų palyginimas

Karkasas	Sąrašai	Žymės	Patikra	Testai	Spring	i18n	Išklotinės	Įrankiai	Nauda	Darbo sk.
JSF	±	n.d.	√	√	√	√	T	14	√√	>500
Spring MVC	√	√	√	√	√	√	S/T	3	√	>400
Stripes	n.d.	√	±	√	n.d.	n.d.	S	–	n.d.	n.d.
Struts	√	√	√	±	√	√	S/T	10	√	>2000
Struts2	√	√	√√	√	√	√	S/T	4	√	>800
Tapestry	n.d.	±	√	±	√	√	n.d.	7	±	<50
Wicket	√	√	±	√√	n.d.	n.d.	n.d.	3	n.d.	n.d.
WebWork	√	√	√√	√	√	√	S	1	±	<30

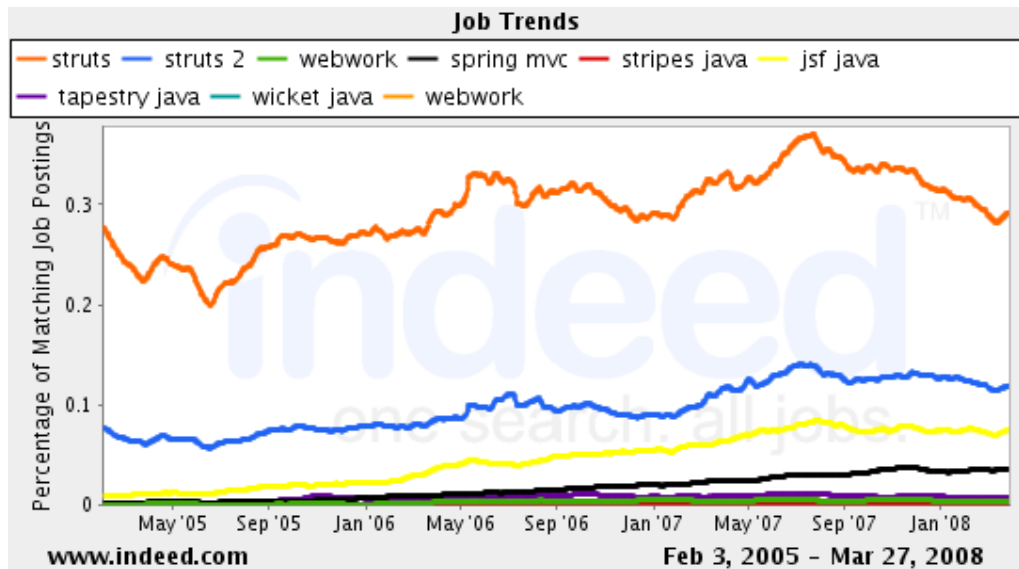
Lentelės Nr. 13 legenda:

Žymėjimas	Paaškinimas	Žymėjimas	Paaškinimas	Žymėjimas	Paaškinimas
√√	L. gerai	±	Dalinis	M/T, M arba T	S – SiteMesh; T – Tiles; S/T – abu galima
√	Gerai	–	Nėra	n.d.	Nėra duomenų/nežinoma



## 4.2.2. Darbo pasiūlymų pasiskirstymas

Darbo pasiūlymų, kurių reikalavimuose yra paminėti žiniatinklių programų kūrimo įvairūs karkasai, kitimą matome 17 paveiksle [15]. Nagrinėjami darbo pasiūlymai nuo 2005 m. vasario 3 d. iki 2008 m. kovo 27 d.



17 pav. Darbo pasiūlymų pasiskirstymas

Iš paveikslo matome, kad daugiausia ieškoma darbuotojų, kurie žino, išmano ir turi patirties naudojant Struts arba Struts 2 žiniatinklių programų kūrimo karkasus. Paskutiniu metu matoma tendencija ieškoti tų darbuotojų, kurie turi patirties kurti žiniatinklių programas naudojant JSF karkasą.

## **5. Eksperimentinė dalis**

### **5.1. Įgyvendinti panaudojimo atvejai**

Realizuotoje sistemoje buvo įgyvendinti šie panaudojimo atvejai: vietų valdymas (sukūrimas, redagavimas, ištrynimasis), serverių valdymas (sukūrimas, redagavimas, ištrynimasis), IP adresų valdymas (sukūrimas, ištrynimasis), duomenų bazių valdymas (sukūrimas, ištrynimasis), administratorių valdymas (sukūrimas, redagavimas, ištrynimasis), paslaugų valdymas (sukūrimas, redagavimas, ištrynimasis), sistemos vartotojų valdymas (sukūrimas, redagavimas, ištrynimasis), atsarginės kopijos darymas.

### **5.2. Testuoti panaudojimo atvejai**

Visi įgyvendinti panaudojimo atvejai buvo testuoti. Testams buvo naudojamos įvairios interneto naršyklės: Internet Explorer ir Mozilla Firefox. Buvo naudojami skirtingi kalbų nustatymai šioms interneto naršyklėms: lt\_LT, en\_US.

Visi panaudojimo atvejai davė tokį rezultatą, kokio ir buvo laukta.

### **5.3. Našumas**

Sistemos testavimui buvo naudojamas HP nc6400 kompiuteris su Intel Core 2 2.00 GHz procesoriumi, bei 2 GB atminties. Naudota Microsoft Windows XP Professional su įdiegtu SP2 pataisų rinkiniu.

Testams atlikti buvo naudojamas StrutsTestCase karkaso įskiepis Eclipse programinei įrangai. Visiems testams atlikti vidutiniškai prireikdavo 2-jų sekundžių.

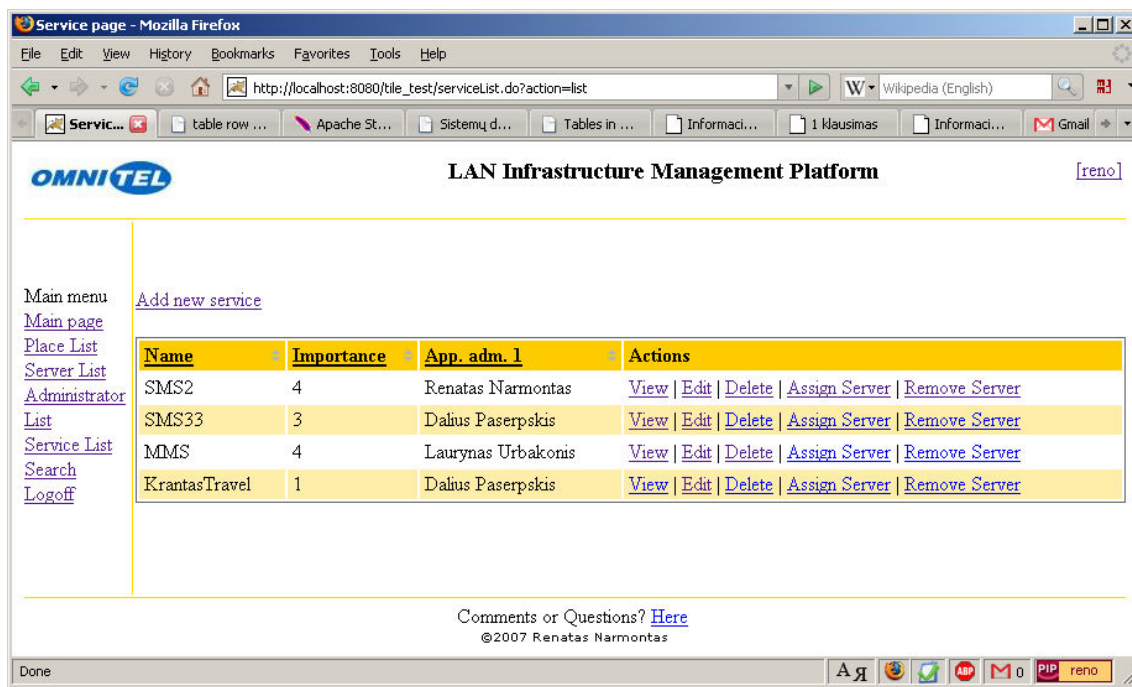
### **5.4. Kodo metrikos**

Sistemai sukurti prireikė parašyti beveik 11.000 kodo eilučių. Buvo suprojektuota daugiau kaip 100 klasių. Parašyta daugiau kaip 430 metodų ir panaudota daugiau kaip 200 klasių atributų.

### **5.5. Projektavimo laiko įvertinimas**

Siekiant įvertinti kiek padeda sutaupyti laiko Apache Struts (toliau Struts) karkaso panaudojimas, reiktų sukurti kitą analogišką sistemą nenaudojant Struts karkaso. Dėl laiko trūkumo tokia pilna sistema nebuvo kuriama. Buvo paskaičiuota kiek eilučių prireikė sukurti puslapiui naudojant Struts karkasą. Taip pat buvo paskaičiuota, kiek eilučių reiktų sukurti

tokiam pačiam puslapiui nenaudojant Struts karkaso. Eksperimentui paėmėme paslaugų sąrašo langą, kurį matome 18 paveiksle.



18 pav. Paslaugų sąrašo langas

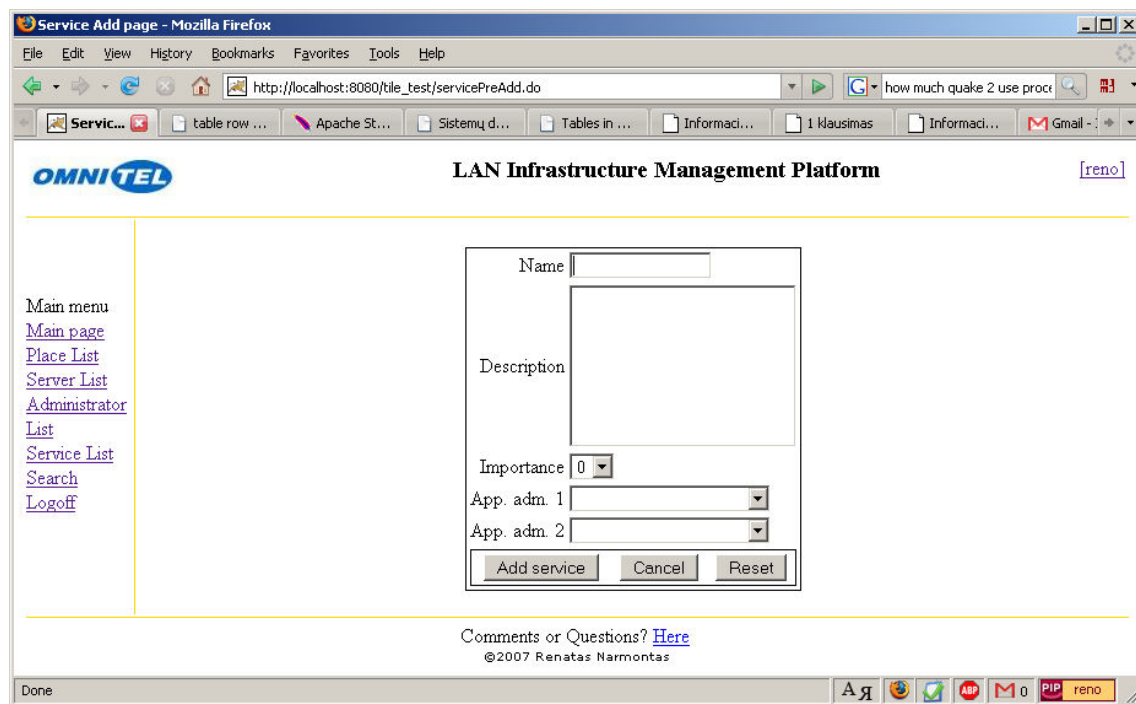
Šis langas susideda iš keturių zonų: antraštės, meniu, pagrindinio ir puslapinės poraštės. Kiekvienos dalies kodo eilučių skaičių pateikiame lentelėje Nr. 14. Taip pat naudojami du konfigūraciniai failai: struts-config.xml ir tiles-defs.xml.

Lentelė Nr. 14 Kodo eilučių skaičius pirmam puslapiui

Lango dalis (failas)	Kodo eilučių skaičius
Antraštė (header.jsp)	36
Meniu (navigation.jsp)	71
Turinio (serviceList.jsp)	43
Puslapinė paraštė (footer.jsp)	18
Struts konfigūracinis failas (struts-config.xml)	61
Tiles konfigūracinis failas (tiles-defs.xml)	19
<b>Viso:</b>	<b>248</b>

Visam puslapiui sukurti reikėjo parašyti 248-ias eilutes kodo.

Taip pat paskaičiuome kodo eilučių kiekį reikalingą 19 paveiksle esančiam langui sukurti.



19 pav. Naujos paslaugos pridėjimo langas

Šis langas taip pat susideda iš keturių zonų: antraštės, meniu, pagrindinio ir puslapiinės paraštės. Kiekvienos dalies kodo eilučių skaičių pateikiame lentelėje Nr. 15. Taip pat naudojami du konfigūraciniai failai: struts-config.xml ir tiles-defs.xml.

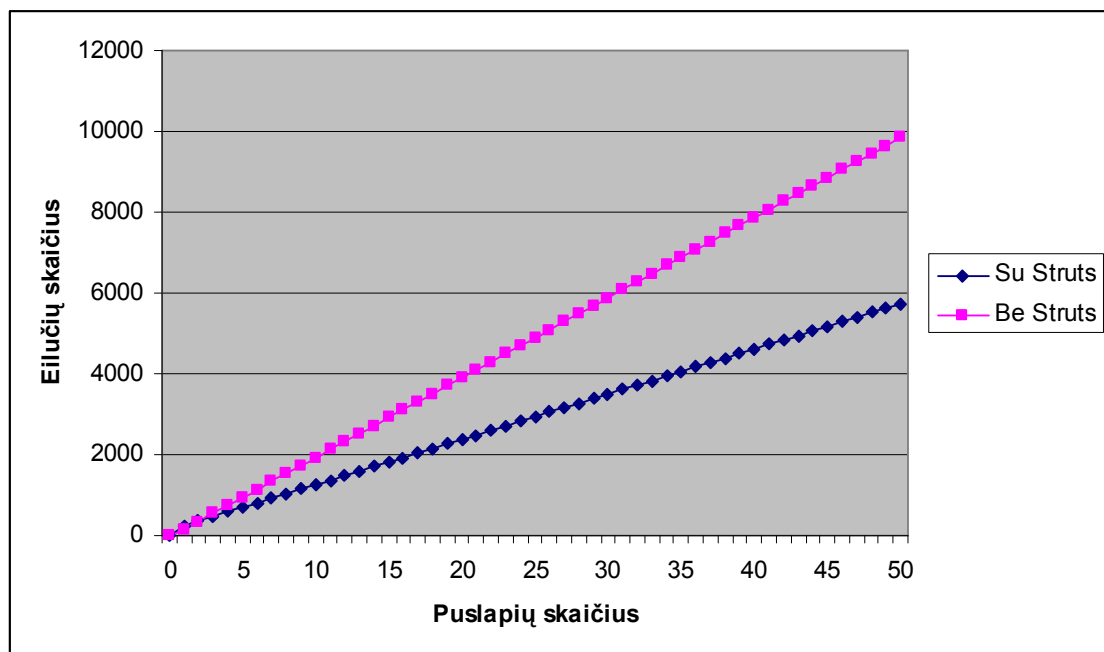
Lentelė Nr. 15 Kodo eilučių skaičius antram puslapiui

Lango dalis (failas)	Kodo eilučių skaičius
Antraštė (header.jsp)	–
Meniu (navigation.jsp)	–
Turinio (serviceAdd.jsp)	90
Puslapiinė paraštė (footer.jsp)	–
Struts konfigūracinis failas (struts-config.xml)	12
Tiles konfigūracinis failas (tiles-defs.xml)	4
<b>Viso:</b>	<b>106</b>

Kiekvienam turinio puslapiui projekte reikėdavo parašyti vidutiniškai po 95 eilutes kodo, po 12 eilučių konfigūraciniame Struts karkaso faile ir po 5 eilutes konfigūraciniame Tiles karkaso faile. Viso apie 112 eilučių.

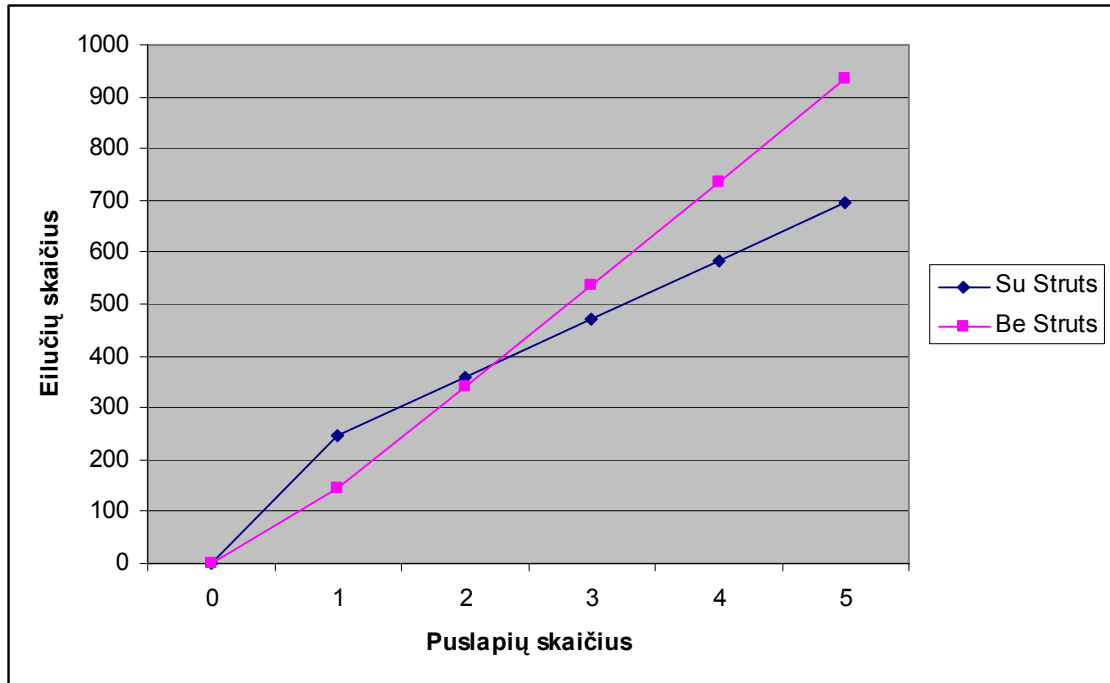
Tokie patys puslapiai buvo sukurti nenaudojant Struts karkaso. Pirmam puslapiui sukurti reikėjo parašyti 146 eilutes kodo (36, 66, 35 ir 9 eilutės atitinkamai), o antram 193 eilutes kodo (36, 66, 82 ir 9 eilutės atitinkamai). Vidutiniškai kiekvienam puslapiui būtų reikėję parašyti po 198 eilutes kodo.

20 paveiksle pateikiame kodo eilučių skaičiaus priklausomybes nuo puslapių kiekio naudojant ir nenaudojant Struts karkasą.



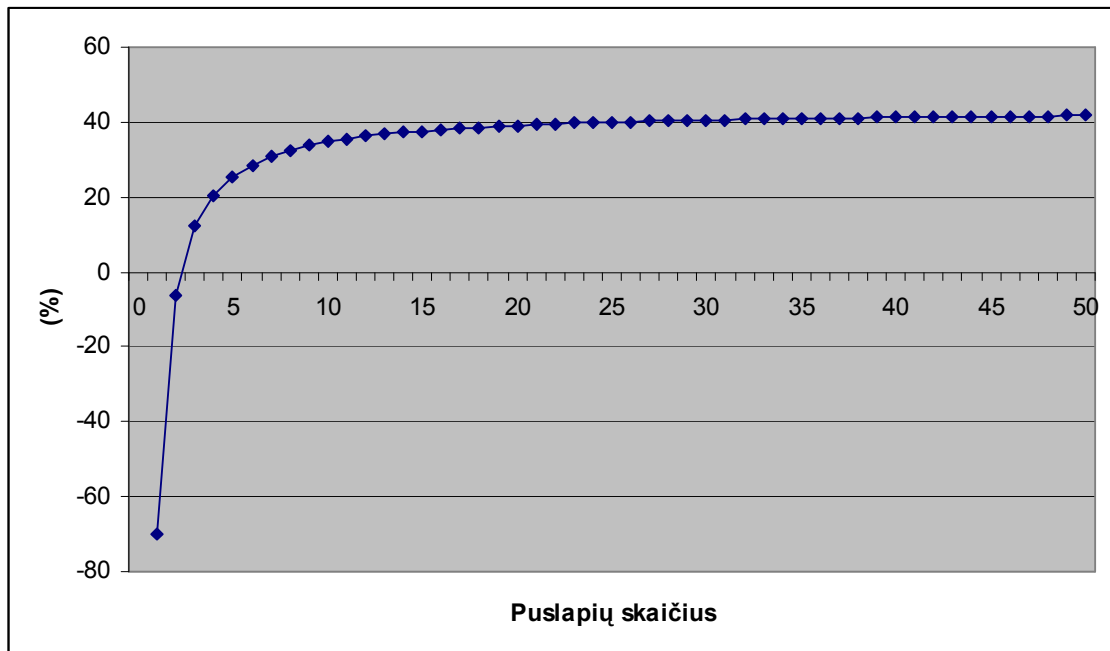
20 pav. Eilučių skaičiaus priklausomybė nuo puslapių skaičiaus (1)

To paties grafiko priartintą pirmą dalį pateikiame 21 paveiksle.



21 pav. Eilučių skaičiaus priklausomybė nuo puslapių skaičiaus (2)

Matome, kad jau nuo 3-io puslapio reikia rašyti mažiau kodo eilučių naudojant Struts karkasą. 22 paveiksle pateikiame kodo eilučių procentinį palyginimą naudojant Struts karkasą ir jo nenaudojant.



22 pav. Kodo eilučių procentinis palyginimas

Iš 22 paveikslo matome, kad vidutiniškai parašoma apie 40% kodo eilučių parašoma mažiau naudojant Struts karkasą, nei jo nenaudojant.

Struts karkaso naudojimo rekomendacijas vartotojams pateikiame lentelėje Nr. 16.

Lentelė Nr. 16 Rekomendacijos vartotojui

		Struts karkaso naudojimo patirtis		
		Didelė	Vidutinė	Maža
Puslapių skaičius (vnt.)	< 5	±	–	–
	5-15	√	√	±
	>15	√	√	√

Lentelės Nr. 16 legenda:

Žymėjimas	Paaškinimas	Žymėjimas	Paaškinimas
√	Naudoti	–	Nenaudoti
±	Reikalinga papildoma sistemos analizė		

## 6. Išvados

1. Iširtos pagrindinės žiniatinklio programų kūrimui naudojamo Apache Struts karkaso savybės. Išanalizuota Modelis-Vaizdas-Valdiklis (MVV) architektūra. Sistemos kūrimo laikas buvo sutrumpintas MVV architektūros dėka, kuri leido susikoncentruoti ties skirtingais komponentais įgyvendinimo etape.

2. Mobilaus ryšio infrastruktūros valdymo sistemos kūrime pritaikytas Apache Struts karkasas, kuris gali būti lengvai plečiamas. Panaudotas Tiles karkasas sistemos puslapių vieningai išvaizdai kurti. Kuriant produktą buvo galvojama, kad sistema bus plečiama. Struts karkasas yra tinkamas projektuojamoje sistemoje dar ir dėl to, kad yra komponentais paremta ir kad palaiko daug kalbų vienu metu skirtingiems vartotojams.

3. Sukurta ir įdiegta mobilaus ryšio infrastruktūros valdymo sistema. Darbo metu realizuotos visos planuotos sistemos funkcijos, sistema atitinka funkcinis ir nefunkcinis reikalavimus, visi panaudojimo atvejai įgyvendinti ir ištestuoti. Produktas sėkmingai įdiegtas UAB „Omnitel“ įmonėje. Su nedideliais pakeitimais sistema gali būti pritaikyta ir kitose įmonėse.

4. Atlikta žiniatinklių programų kūrimui naudojamų karkasų analizė. Lyginome įvairius karkasus tarpusavyje, išskyrėme jų privalumus ir trūkumus. Struts karkasas, nors ir senas lyginant su kitais karkasais, vis dar tinkamas tokio tipo uždaviniams spręsti.

5. Struts karkaso dėka įvairūs vizualūs komponentai buvo pakartotinai naudojami. Tai įtakojo trumpesnę programos kūrimo laiką: apie 40% mažiau kodo eilučių.



## 7. Literatūra

- [1] ARNOLD, K.; GOSLING, J.; ir HOLMSE, D. *The Java Programming Language, Fourth Edition*. Addison Wesley, 2005. 928 p. ISBN 0-321-34980-6.
- [2] Aldona [interaktyvus]. Anglų-lietuvių k. žodynas [žiūrėta 2008-05-20]. Prieiga per internetą <<http://aldona.mii.lt/pms/terminai/en-lt>>.
- [3] Ant [interaktyvus]. Apache Ant [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://ant.apache.org/>>.
- [4] CAVANESS, Ch. *Programming Jakarta Struts*. O'Reilly, 2004. 470 p. ISBN 0-596-00651-9.
- [5] CEKVENICH, V. *Struts Fast Track*. BaseBeans Engineering, San Francisko, 2002. 320 p. ISBN 0-9716619-0-1.
- [6] COUNG, N. Object-relational mapping Hibernate and Struts framework in Java. Magistro darbas. Czech Technical University in Prague, Praha, 2008.
- [7] COHEN, F. *Java testing design and automation: A guide to maximizing performance*. Prentice Hall, 2005. 433 p. ISBN 0131421891.
- [8] DIEZ, L. *Secure, scalable and component-based Webshop using Struts and Hibernate*. Magistro darbas. KTH (Kungliga Tekniska högskolan - Royal Institute of Technology), Stockholm, 2006.
- [9] DORAY, A. *Beginning Apache Struts: From Novice to Professional*. Apress/Berkeley, 2006. 508 p. ISBN 1-59059-604-8.
- [10] FORD, N. *Art of Java Web development*. Manning Publications/Greenwich, 2004. 593 p. ISBN 1-932394-06-0.
- [11] GOODWILL, J. *Mastering Jakarta Struts*. Willey Publishing Inc./Indianapolis, 2002. 255 p. ISBN 0-471-21302-0.
- [12] GOODWILL, J., ir HIGHTOWER, R. *Professional Jakarta Struts*. Wrox Press, 2004. 429 p. ISBN 0764544373.
- [13] HOLMES, J. *Struts: The Complete Reference, Second Edition*. McGraw-Hill/Osborne, 2007. 550 p. ISBN 9780072263862.
- [14] HUSTED, T., et al. *Struts in Action: Building web applications with the leading Java framework*. Manning Publications/Greenwich, 2003. 630 p. ISBN 1-930110-50-2.
- [15] Indeed [interaktyvus]. Job trends – [žiūrėta 2008-04-01]. Prieiga per internetą:

- <<http://www.indeed.com/jobtrends>>.
- [16] JAVA [interaktyvus]. JAVA [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.java.com/en/>>.
- [17] JavaServer Faces [interaktyvus]. JavaServer Faces technology [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://java.sun.com/javaee/jaserverfaces/>>.
- [18] KOFMAN, V. Modern Java Frameworks for Web Development. *Developer* [interaktyvus]. 2008 [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.developer.com/>>.
- [19] Likit [interaktyvus]. Kalbos problemas [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.likit.lt/>>.
- [20] Microsoft [interaktyvus]. Microsoft [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.microsoft.com/>>.
- [21] MySQL [interaktyvus]. MySQL developer zone [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://dev.mysql.com/>>.
- [22] ROBINSON, M.; ir FINKELSTEIN, E. *Jakarta Struts for Dummies*. Willey Publishing Inc./Indianapolis, 2004. 392 p. ISBN 0-7645-5957-5.
- [23] SiteMesh [interaktyvus]. SiteMesh overview [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.opensymphony.com/sitemesh/>>.
- [24] SOMMERVILLE, I. *Software Engineering*. Addison-Wesley, 2006. 840 p. ISBN 0321313798.
- [25] Struts [interaktyvus]. Apache Struts [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://struts.apache.org/>>.
- [26] StrutsTestCase [interaktyvus]. StrutsTestCase for JUnit [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://strutstestcase.sourceforge.net/>>.
- [27] SPIELMAN, S. *The Struts Framework: Practical Guide for Java Programmers*. Morgan Kaufmann publishers/San Francisco, 2003. 137 p. ISBN 1-55860-862-1.
- [28] Spring [interaktyvus]. Spring framework [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.springframework.org/>>.
- [29] Stripes [interaktyvus]. Stripes framework [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.stripesframework.org/>>.
- [30] Tapestry [interaktyvus]. Welcome to Tapestry [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://tapestry.apache.org/>>.
- [31] Tiles [interaktyvus]. Apache Tiles [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://tiles.apache.org/>>.

- [32] Tomcat [interaktyvus]. Apache Tomcat [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://tomcat.apache.org/>>.
- [33] PAULAVIČIŪTĖ, I. Atviro kodo produktų kokybės vertinimas ir kokybės užtikrinimo strategijos. *Informacinės technologijos 2006*: konferencijos pranešimų medžiaga. Kaunas, 2006, p. 70-74.
- [34] VLKK [interaktyvus]. VLKK konsultacijų bankas [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.vlkk.lt/konsultacijos/>>.
- [35] WebWork [interaktyvus]. WebWork [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://www.opensymphony.com/webwork/>>.
- [36] Wicket [interaktyvus]. Apache Wicket [žiūrėta 2008-05-20]. Prieiga per internetą: <<http://wicket.apache.org/>>.
- [37] Wikipedia [interaktyvus]. Model-view-controller [žiūrėta 2008-04-02]. Prieiga per internetą: <<http://en.wikipedia.org/wiki/Model-view-controller>>.

## **8. Analysis of Struts framework in mobile network infrastructure management system**

### Summary

Today there are a lot of various Web frameworks. Part of these frameworks follows the Model-View-Controller (MVC) design pattern. In this study we compared 8<sup>th</sup> most popular Web development frameworks, described the merits and demerits.

The system was developed following the MVC design pattern as well. This was accomplished by using Apache Struts, a framework designed for developing web applications. View component was developed using Apache Tiles technology. It offers an easy to use framework for page layout. We are going to work on the following properties of the system: scalable, internationalized, secure, fault tolerance and MVC component based.

The system was adopted to one of the mobile network operators.

Keywords: Apache Struts, MVC, Web frameworks.

## 9. Terminų ir santrumpų žodynas

Sutrumpinimas/Terminas	Paaškinimas
MVV	Modelis-Vaizdas-Valdiklis architektūra.
MVC (angl. Model-View-Controller)	Žr. MVV.
CSV (angl. Comma-Separated Values)	Kableliu atskirtos reikšmės.
Driver	Tvarkyklė
HTML (angl. HyperText Markup Language)	HTML kalba
GUI (angl. Graphical User Interface)	Grafinė vartotojo sąsaja.
DBVS	Duomenų bazių valdymo sistema.
KAC	Klientų aptarnavimo centras.
MRIVS	Mobilaus ryšio infrastruktūros valdymo sistema.
OMC (angl. Operation Maintenance Center)	Tinklo valdymo centras.
JDBC (angl. Java DataBase Connectivity™)	Java prisijungimo prie duomenų bazių valdymo sistemų technologija.
JSP (angl. JavaServer Pages)	Java servlet technologijos išplėtimas.
I18n (angl. internationalization)	Internacionalinis, tarptautinis.
Java	Programavimo kalba.
Java servlet'as	Žiniatinklių (žr. žiniatinklis) programų, naudojančių Java platformą, išplėtimas ir patobulinimas. Jie suteikia komponentais grįstus nuo platformos nepriklausomus metodus žiniatinklių programų kūrimui.
IT	Informacinės technologijos.
DUK (angl. FAQ)	Dažniausiai užduodami klausimai.
OS (angl. Operating System)	Operacinė sistema.

SQL (angl. Structured Query Language)	SQL kalba.
Screenshot	Momentinė ekrano kopija.
IDE (angl. Integrated Development Environment)	Programų kūrimo aplinka.
Saitas (angl. hyperlink, link)	Hipertekste (žr. hipertekstas) esanti nuoroda, turinti matomą dalį ir po ja paslėptą komponentą – nurodomo objekto adresą.
Hipertekstas (angl. hypertext)	Tekstas, kurio atskiros dalys siejamos saitais (žr. saitas).
Žiniatinklis (angl. web)	Hipertekstinės informacijos visuotinis tinklas, svarbiausias interneto komponentas.
Tinklapis (angl. web page, page)	Tai hipertekstu (žr. hipertekstas) parašytas žiniatinklio dokumentas.
Svetainė (angl. web site, site)	Tinklapių rinkinys, kuriuos sieja bendra tematika, priklausomybė vienai įstaigai ar kitokie bendri dalykai.
SSL (angl. Secure Socket Layer)	SSL protokolas.
Tiles framework	Išklotinių karkasas.
URI (angl. Uniform Resource Identifier)	Universalusis išteklių identifikatorius.
URL	Universalusis adresas.

## **Priedas Nr. 1 Programinės įrangos įdiegimo patvirtinimo aktas**

## Priedas Nr. 2 Panaudojimo atvejų sąrašas

1	PANAUDOJIMO ATVEJIS:	Prisijungimas / Atsijungimas
<b>Vartotojas/Aktorius</b>		Administratorius, Vadovas, Inžinierius
<b>Aprašas</b>		Prisijungimas prie sistemos pradėdant darbą / atsijungimas nuo jos pabaigus darbą.
<b>Prieš-sąlyga:</b>		Vartotojas nėra prisijungęs prie sistemos / Vartotojas yra prisijungęs prie sistemos
<b>Sužadinimo sąlyga:</b>		Vartotojas įvedė prisijungimo vartotojo vardą ir slaptažodį bei paspaudė Prisijungti / Atsijungti mygtuką
<b>Po-sąlyga:</b>		Vartotojas sėkmingai prisijungė / atsijungė nuo sistemos
2	PANAUDOJIMO ATVEJIS:	Ataskaitų generavimas
<b>Vartotojas/Aktorius</b>		Administratorius, Vadovas, Inžinierius
<b>Aprašas</b>		Ataskaitų generavimas įvairiais pjūviais
<b>Prieš-sąlyga:</b>		Vartotojas yra prisijungęs prie sistemos
<b>Sužadinimo sąlyga:</b>		Vartotojas užpildė ataskaitai generuoti reikalingus parametrus ir paspaudė Generuoti ataskaitą arba Generuoti raportą mygtuką
<b>Po-sąlyga:</b>		Vartotojui parodoma ataskaita arba raportas
3	PANAUDOJIMO ATVEJIS:	Paieška
<b>Vartotojas/Aktorius</b>		Administratorius, Vadovas, Inžinierius
<b>Aprašas</b>		Serverių fizinės buvimo vietos nustatymas arba paslaugai veikti reikalingų serverių nustatymas
<b>Prieš-sąlyga:</b>		Vartotojas prisijungęs prie sistemos
<b>Sužadinimo sąlyga:</b>		Vartotojas užpildė paieškai reikalingus parametrus ir paspaudė Ieškoti mygtuką



**Po-sąlyga:** Vartotojui parodoma serverio buvimo vieta arba nuo kurių serverių veikimo priklauso paslaugos veikimas

4 PANAUDOJIMO ATVEJIS: Vartotojų valdymas

**Vartotojas/Aktorius** Administratorius

**Aprašas** Naujų sistemos vartotojų sukūrimas, jų modifikavimas, ištrynimasis iš sistemos

**Prieš-sąlyga:** Vartotojas prisijungęs prie sistemos

**Sužadinimo sąlyga:** Vartotojas įvedė naujo vartotojo duomenis, juos pakeitė arba nurodė kokį vartotoją reikia ištrinti ir paspaudė atitinkamai vieną iš mygtukų: Sukurti, Modifikuoti ar Ištrinti

**Po-sąlyga:** Sistemos vartotojas buvo sukurtas, modifikuotas arba ištrintas

5 PANAUDOJIMO ATVEJIS: Atsarginės kopijos darymas

**Vartotojas/Aktorius** Administratorius

**Aprašas** Atsarginės kopijos darymas

**Prieš-sąlyga:** Vartotojas prisijungęs prie sistemos

**Sužadinimo sąlyga:** Vartotojas paspaudžia Kopija mygtuką

**Po-sąlyga:** Duomenys yra išeksportuojami iš sistemos į failą

6 PANAUDOJIMO ATVEJIS: Paslaugų įvedimas, keitimas ar šalinimas

**Vartotojas/Aktorius** Inžinierius

**Aprašas** Yra įvedami duomenys apie naują paslaugą, modifikuojami jau egzistuojančios paslaugos parametrai ar ištrinami egzistuojančios paslaugos duomenys iš sistemos

**Prieš-sąlyga:** Vartotojas prisijungęs prie sistemos

<b>Sužadavimo sąlyga:</b>	Vartotojas įveda naujos paslaugos duomenis, pakeičia egzistuojančios paslaugos parametrus ar pasirenka iš egzistuojančių paslaugų sąrašo nebereikalingą paslaugą ir paspaudžia atitinkamai mygtuką: Sukurti paslaugą, Modifikuoti paslaugą ar Ištrinti paslaugą.
<b>Po-sąlyga:</b>	Paslaugos informacija atitinkamai sukurta, pakeista ar ištrinta iš sistemos

7	PANAUDOJIMO ATVEJIS:	Serverio įvedimas, keitimas ar šalinimas
<b>Vartotojas/Aktorius</b>	Inžinierius	
<b>Aprašas</b>	Yra įvedami duomenys apie naują serverį, modifikuojami jau egzistuojančio serverio parametrai ar ištinami egzistuojančio serverio duomenys iš sistemos	
<b>Prieš-sąlyga:</b>	Vartotojas prisijungęs prie sistemos	
<b>Sužadavimo sąlyga:</b>	Vartotojas įveda naujo serverio duomenis, pakeičia egzistuojančio serverio parametrus ar pasirenka iš egzistuojančių serverių sąrašo nebereikalingą serverį ir paspaudžia atitinkamai mygtuką: Sukurti serverį, Modifikuoti serverį ar Ištrinti serverį	
<b>Po-sąlyga:</b>	Serverio informacija atitinkamai sukurta, pakeista ar ištrinta iš sistemos	

## Priedas Nr. 3 Licenzija

### Licenzijos tekstas

Mobilaus ryšio infrastruktūros valdymo sistema

Naudojimo ir platinimo licenzija

Mobilaus ryšio infrastruktūros valdymo sistema (toliau MRIVS) yra platinama kaip pabandyk-prieš-perkant. Tai reiškia:

1. Visos teisės į MRIVS išimtinai priklauso autoriui – Renatui Narmontui.
2. Kiekvienas gali naudotis šia programa nemokamai 30 (trisdešimt) dienų. Šiam 30-ies dienų periodui pasibaigus, jei Jūs norite toliau naudotis MRIVS, Jūs privalote užsiregistruoti.
3. Kartą užsiregistravus vartotojui yra suteikiama teisė naudoti MRIVS viename kompiuteryje (su vienu procesoriumi) vienu metu. Registruota MRIVS programinė įranga negali būti išnuomota ar paskolinta, tačiau gali būti perleista visam laikui trečiam asmeniui, jei priimantis asmuo sutinka su šios licenzijos sąlygomis.
4. Neregistruota MRIVS gali būti laisvai platinama su sąlyga, kad instaliacinis paketas nėra modifikuotas. Joks asmuo ar kompanija negali imti užmokesčio be raštiško autorinių teisių valdytojo sutikimo. Neregistruota MRIVS programa negali būti platinama su jokia kita programine įranga be raštiško autorinių teisių valdytojo sutikimo.
5. Kad užregistruoti MRIVS, Jūs turite užpildyti registracijos formą ir išsiųsti ją kartu su registracijos mokesčiu viename iš autorizuotų registracijos svetainių.
6. MRIVS YRA PLATINAMA „KAIP YRA“ IR BE JOKIOS RŪŠIES GARANTIJOS. JŪS PRISIIMATE VISĄ RIZIKĄ, SUSIJUSIĄ SU PROGRAMOS KOKYBE IR VYKDYMU. JEIGU PROGRAMA PASIRODYS TURINTI DEFEKTŲ, JŪS PRISIIMATE VISAS BŪTINAS TECHNINES APŽIŪROS, SUTVARKYMO AR KOREGAVIMO IŠLAIDAS.
7. JOKIU KITU ATVEJU, IŠSKYRUS GALIOJANČIAM ĮSTATYMIUI REIKALAUJANT, NEBŪS JUMS ATLYGINTA UŽ NUOSTOLIUS

(ĮTRAUKIANT BET KOKIUS BENDRUS, IŠSKIRTINIUS, ATSTITIKTINIUS AR IŠPLAUKIANČIUS IŠ PROGRAMOS NAUDOJIMO ARBA NESUGEBĖJIMO NAUDOTI PROGRAMĄ NUOSTOLIUS: DUOMENŲ PRARADIMĄ, DUOMENŲ SUGADINIMĄ, PROGRAMOS NESUDERINAMUMĄ SU KITOMIS PROGRAMOMIS AR BET KOKIUS KITUS NUOSTOLIUS, PATIRTUS JŪSŲ AR TREČIŲJŲ PUSIŲ) - NET IR TUO ATVEJU, KAI DOKUMENTAI (AUTORINĖS TEISĖS AR KITI) AR KITOS PUSĖS PRANEŠĖ APIE TOKIUS GALIMUS NUOSTOLIUS.

8. Jūs negalite kopijuoti, klonuoti, nuomuoti, paskolinti, parduoti, modifikuoti, dekompiliuoti ar kitaip atlikti atgalinę inžineriją. Bet koks neautorizuotas panaudojimas iššaukia nedelsiamą ir automatinį šios licenzijos nutraukimą ir asmuo gali būti patrauktas kriminaliniam ir/arba civiliniam baudžiamajam persekiojimui.
9. MRIVS įdiegimas ir naudojimas pažymi, kad yra patenkinamos šios licenzijos sąlygos ir su jomis yra sutinkama.
10. Jei Jūs nesutinkate su šios licenzijos sąlygomis Jūs privalote ištrinti MRIVS bylas iš kompiuterio ir liautis naudotis šiuo produktu.

Ačiū, kad naudojate MRIVS.

Renatas Narmontas

## Licenzijos registracijos forma

### MRIVS REGISTRACIJOS FORMA

Prašu atsakyti į klausimus, reikalingus registracijai ir išsiųsti į artimiausią tinklalapį.

Daugiau informacijos apie registraciją prašome žiūrėti README.TXT byloje.

Prašome užpildyti šią formą DIDŽIOSIOMIS RAIDĖMIS.

Vardas:	_____	Pavardė:	_____
Organizacija:	_____		
Adresas:	_____		
Miestas:	_____		
Šalis:	_____		
Telefonas:	_____	Faksas:	_____
El. Paštas:	_____		

## Priedas Nr. 4 Sistemos katalogo struktūra

Lentelėje Nr. 17 pateikiama sistemos katalogo struktūra ir aprašymas.

Lentelė Nr. 17 Sistemos katalogo struktūra

Katalogas	Aprašymas
./documentation	Sistemos dokumentacija
./css	Stilių aprašai
./img	Piešinių katalogas
./logs	Žurnalų katalogas
./pages	JSP failai
./WEB-INF	Struts, tiles ir web xml konfigūraciniai failai
./WEB-INF/classes	Sukompilijuotos java klasės
./WEB-INF/lib	Sistemos naudojamos bibliotekos
./WEB-INF/src	Išeities tekstai

## **Priedas Nr. 5 Pridėto kompaktinio disko turinys**

Pridėto kompaktinio disko turinys ir jo struktūra yra tokia:

**/Docs** – šis magistro baigiamasis darbas PDF formate; MRIVS vartotojo instrukcija PDF formate. Sistemos licenzija PDF formate.

**/Eclipse** – Nemokama programų kūrimo aplinka Eclipse 3.3.

**/Libraries** – visos naudotos bibliotekos.

**/struts\_1.3.8** – Struts karkasas;

**/displaytag\_1.1.1** – sąrašų/lentelių rodymo komponentė;

**/junit\_4.4** – testavimo įrankio įskiepis;

**/log4j\_1.2.15** – žurnalų vedimo komponentė;

**/mysqlconnector\_5.0.7**- MySQL DBVS JDBC tvarkyklė;

**/strutstest\_2.1.4** – testavimo įrankio įskiepis.

**/Support** – papildoma programinė įranga, naudota sistemos kūrime ir naudojime

**/MySQL\_5.0.45** – MySQL DBVS.

**/Tomcat\_6.0.14** – žiniatinklių serveris Apache Tomcat;

**/War** – pilnas sistemos paketas

## **Priedas Nr. 6 Kompaktinis diskas**