

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
PROGRAMŲ INŽINERIJOS KATEDRA**

**DOMAS GREIČIUS**

**RENGINIŲ ORGANIZAVIMO SISTEMOS  
KONCEPTUALINIO MODELIO TRANSFORMACIJŲ TYRIMAS**

**MAGISTRO DARBAS**

**KAUNAS, 2009**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
PROGRAMŲ INŽINERIJOS KATEDRA**

**RENGINIŲ ORGANIZAVIMO SISTEMOS  
KONCEPTUALINIO MODELIO TRANSFORMACIJŲ TYRIMAS**

MAGISTRO DARBAS

Autorius: .....  
Domas Greičius, IFM-3/2 gr. stud.  
2009 gegužės d.

Vadovas: .....  
prof. Eduardas Bareiša  
2009 gegužės d.

Recenzentas: .....  
doc. Rita Butkienė  
2009 gegužės d.

KAUNAS, 2009

# TURINYS

<b>TURINYS</b> .....	<b>3</b>
<b>IVADAS</b> .....	<b>7</b>
<b>1 ANALIZĖS DALIS</b> .....	<b>8</b>
1.1 KONCEPTUALAUS MODELIAVIMO APŽVALGA.....	8
1.1.1 Pateikimo modelis.....	9
1.2 MODELIU TRANSFORMACIJOS.....	10
1.3 PROJEKTO TIKSLAS.....	13
1.4 RENGINIŲ ORGANIZAVIMO PROBLEMOS ANALIZĖ.....	13
1.4.1 Kūrimo problemos.....	13
1.4.2 Sistemos kontekstas.....	22
1.4.3 Vartotojo charakteristikos.....	23
1.4.4 Vartotojo problemos.....	24
1.4.5 Vartotojo tikslai.....	24
1.4.6 Egzistuojantys sprendimai.....	25
<b>2 PROJEKGINĖ DALIS</b> .....	<b>29</b>
2.1 REIKALAVIMŲ SPECIFIKAVIMAS.....	29
2.1.1 Įvadas.....	29
2.1.2 Sistemos apribojimai.....	30
2.1.3 Funkciniai reikalavimai.....	31
2.1.4 Nefunkciniai reikalavimai.....	39
2.2 SISTEMOS PROJEKTAVIMAS.....	44
2.2.1 Sistemos objektinis modelis.....	44
2.2.2 Sistemos dinaminis modelis.....	48
2.2.3 Išdėstymo vaizdas.....	54
<b>3 EKSPERIMENTINĖ DALIS</b> .....	<b>55</b>
3.1 SISTEMOS PROBLEMINĖ SRITIS.....	55
3.1.1 Pradinis konceptualinis modelis.....	56
3.2 KONCEPTUALINIO MODELIO TRANSFORMACIJA.....	58
3.2.1 Aprašymas.....	60
3.3 TRANSFORMACIJOS ITAKA SISTEMOS ARCHITEKTŪRAI.....	62
3.4 RENGINIO DALYVIŲ PARINKIMO TYRIMAS.....	62
3.4.1 Ajax užklausų skaičiaus mažinimas.....	62
3.4.2 MySQL užklausų optimizavimas, indeksų dėliojimas.....	63
3.4.3 Javascript naudojimas.....	67
3.4.4 Kodo eilučių pokytis.....	69
<b>IŠVADOS</b> .....	<b>70</b>
<b>LITERATŪRA</b> .....	<b>71</b>
<b>TERMINŲ IR SANTRAUPŲ ŽODYNAS</b> .....	<b>73</b>
<b>PRIEDAI</b> .....	<b>75</b>

# PAVEIKSLŲ SĄRAŠAS

PAV. 1 SISTEMOS KONTEKSTAS.....	23
PAV. 2 DUOMENŲ BAZĖS SCHEMA.....	35
PAV. 3 KLASIŲ IŠSKAIDYMAS Į PAKETUS .....	44
PAV. 4 PAGRINDINIO VALDIKLIO KLASIŲ PAKETAS.....	45
PAV. 5 KALBŲ VALDIKLIO KLASIŲ PAKETAS.....	46
PAV. 6 RENGINIŲ VALDIKLIO KLASIŲ PAKETAS.....	47
PAV. 7 PANAUDOJIMO ATVEJŲ DIAGRAMA.....	48
PAV. 8 SISTEMOS AKTORIAI.....	48
PAV. 9 SUKURTI KONTAKTĄ VEIKLOS DIAGRAMA.....	49
PAV. 10 TVARKYTI KONTAKTUS VEIKLOS DIAGRAMA.....	50
PAV. 11 IEŠKOTI VIEŠŲ RENGINIŲ VEIKLOS DIAGRAMA.....	51
PAV. 12 SUKURTI NAUJĄ RENGINĮ VEIKLOS DIAGRAMA.....	52
PAV. 13 TVARKYTI RENGINIUS VEIKLOS DIAGRAMA.....	53
PAV. 14 SISTEMOS IŠDĖSTYMO VAIZDAS.....	54
PAV. 15 PRADINIO KONCEPTUALINIO MODELIO DALYVIŲ VALDYMO LANGO PATEIKIMO OBJEKTAS.....	56
PAV. 16 PRADINIO KONCEPTUALINIO MODELIO VEIKLOS DIAGRAMA.....	57
PAV. 17 PASKIRTIES KONCEPTUALINIO MODELIO VEIKLOS DIAGRAMA.....	59
PAV. 18 PASKIRTIES KONCEPTUALINIO MODELIO DALYVIŲ VALDYMO LANGO PATEIKIMO OBJEKTAS.....	60
PAV. 19 AJAX UŽKLAUSŲ SKAIČIUS, KVIEČIANT (ŠALINANT) DALYVIUS.....	63
PAV. 20 KONTAKTŲ IŠRINKIMO UŽKLAUSOS NAUDOJANT MYISAM IR INNODB VYKDYMO LAIKAI.....	64
PAV. 21 DALYVIŲ VALDYMO LANGO, PAKVIESTŪJŲ SĄRAŠO DOM ELEMENTAI.....	68
PAV. 22 DALYVIŲ ŽYMĖJIMO GREIČIO Palyginimas pradiniam ir galutiniam VARIANTE.....	68
PAV. 23 PRADINIS RENGINIO DALYVIŲ VALDYMO LANGAS.....	75
PAV. 24 PAKEISTAS RENGINIO DALYVIŲ LANGAS.....	75

# LENTELIŲ SARAŠAS

## **SANTRAUKA**

Darbo tikslas yra išnagrinėti renginių organizavimo problemas, sukurti Renginių organizavimo sistemą, bei naudojant konceptualaus modeliavimo metodikas atlikti modelių transformacijų tyrimą. Darbo metu buvo išnagrinėta eilė kitų panašių sistemų, ir kuriant šią sistemą buvo atsižvelgta į jų trūkumus bei privalumus.

Modeliai ir modeliavimo metodai yra būdas nagrinėti ir valdyti šiuolaikinių sistemų uždavinių apimtį ir sudėtingumą naudojant įvairias abstrakcijas. Pats modelis yra tam tikra abstrakcijos rūšis, įgalinanti atvaizduoti sistemą ar jos dalį abstrakčiau, išryškinant esmines savybes, praleidžiant ar paslepiant mažiau svarbias. Sistemos projektavime buvo naudotos konceptualaus modeliavimo metodikos, kurios yra sėkmingai taikomos grafiškai atvaizduoti informacines sistemas konceptualiame lygmenyje. Vėliau taikant šias metodikas buvo atliktas eksperimentinis tyrimas su viena iš sukurtos sistemos dalių.

Darbe yra panaudota 24 paveikslai, 27 lentelės, 47 literatūros šaltiniai.

## **SUMMARY**

### **Research of events organizing system's conceptual model transformation.**

The object of this work is to analyze the problems of events' organizing, and to create Events' organizing system, that would correspond clients' requirements, and expectations of users. During this work I explored a number of similar systems, and all their pros and cons has been taken into consideration while creating this system.

Models and modeling techniques is one of the ways to analyze and control the complexity and volume of problems of modern systems using various abstraction levels. Model by itself is a certain type of abstraction that helps to visualize a system or part of it more abstractly, highlighting key features, while skipping or hiding less important ones. In the development of the system, I used methods of conceptual modeling that are successfully applied in graphic development of systems at the conceptual level. Later using these methods I did an experiment with one of the parts of the system.

In this work I used 24 pictures, 27 tables, and 47 literature sources.

## **IVADAS**

Pagrindinė problema, kuriai išspręsti skirtas šis projektas, yra renginio paskelbimas svečiams, pakvietimų išsiuntinėjimas į renginyje kviečiamiems asmenims, bei svečių registravimas renginyje. Kvietimų išsiuntinėjimas turi būti supaprastintas iki minimumo. Dalyvių reakcijos užfiksavimas, neturėtų kelti jokių sunkumų kvietimą gavusiam asmeniui, net jei jo kompiuterinis raštingumas yra žemas. Kvietimą gavęs asmuo turėtų turėti galimybę pranešti organizatoriui apie dalyvavimo statusą, net nesinaudodamas kompiuteriu. Sistema turi būti patogi tiek kviečiamam asmeniui (kuris yra perspektyvus sistemos naudotojas kaip organizatorius), tiek ir renginių organizatoriui. Tai turėtų pasireikšti patogia ir aiškia vartotojo sąsaja, neįkyriu dizainu, neperkrautu informacijos kiekiu.

Sistemos privalumai pašalina administravimo sunkumus, kurie yra susiję su dalyvių registravimu ir tvarkymu. Sistema padeda automatiškai surinkti, rikiuoti ir tvarkyti visą renginio informaciją internete.

**Tyrimo objektas** – atlikti tyrimą su konceptualinio modelio transformacijomis

**Tyrimo tikslas** – atlikti renginio dalyvių valdymo dalies konceptualaus modelio transformaciją, bei padaryti pakeitimų palyginimą su ankstesne sistemos versija.

Darbas yra suskirstytas į keturias pagrindines dalis:

**Analitinė dalis.** Aptariamas bendras projekto tikslas, atliekama panašių sistemų analizė, apžvelgiami sistemos kūrimo įrankiai, bei konceptualaus modeliavimo bei modelių transformavimo koncepcijos.

**Projektinė dalis.** Pateikiama magistrantūros studijų metu sukurtos renginių organizavimo sistemos projektinė dokumentacija.

**Eksperimentinė dalis.** Nustatoma probleminė sistemos dalies vieta, bei jai sukuriamas pradinis konceptualinis modelis, ir atliekama šio modelio transformacija į paskirties modelį, aptariama kaip buvo atliekamas sistemos funkcionalumo perdarymas, siekiant išpildyti atsiradusius naujus reikalavimus, bei atliekamo funkcionalumo optimizavimas.

**Išvados.** Šioje dalyje pateikiu apibendrintą informaciją apie sukurtos sistemos funkcionalumą ir išskirtinumą, modelių ir jų transformacijų svarbą sistemos modeliavime, bei kokie tolimesni darbai galėtų būti atlikti.

# 1 ANALIZĖS DALIS

## 1.1 Konceptualaus modeliavimo apžvalga

[OSCA] Objektiškai orientuoto konceptualaus modeliavimo fazės metu, aprašomi visi objektai bei komponentai, atspindintys nagrinėjamą sritį, išvengiant bet kokių aptarimų apie įgyvendinimą. Sistemos konceptualus modelis privalo nusakyti visas jos savybes (statines ir dinamines) ir privalo neturėti jokių įgyvendinimo detalių, išskyrus tas kurios jau yra numatytos reikalavimuose. Sistema tai skirtingų dalių rinkinys, kurios sudaro unikalią visumą skirtą tam tikram tikslui. Šios sąveikos egzistavimas tarp sistemos komponentų yra labai svarbus faktas, ir tokia sąveika sąlygoja sistemos atsiradimą, kuri, kaip visuma, turi funkcionalumą, kuris yra juntamas aplinkos.

Programinės įrangos inžinerijos disciplina tradiciškai aprašo sistemas trimis aspektais: duomenų, valdymo ir proceso. Sutelkiant dėmesį į specifikavimo metodus, galima apibrėžti tokius metodų tipus:

- Funkcinės specifikacijos metodas, aprašantis sąveikas, kurios yra svarbios sistemos veikimo srityje.
- Elgsenos specifikavimo metodai, kurie atspindi funkcijų vykdymo eiliškumą
- Ryšių specifikavimo metodai, naudingi, aprašant kaip sistema bendrauja su išoriniai objektais, arba kaip sistemos komponentai bendrauja vienas su kitu tam kad galėtų vykdyti savo funkcijas.
- Dekompozicijos specifikavimo metodai, leidžia aprašyti sistemos komponentus, aiškiai apibrėžiant jų reikšmę aplinkoje.

Vienas iš pagrindinių privalumų, kurie pateisina objektiškai orientuoto modeliavimo paradigmos naudojimą, yra tai kad ji aukščiau išvardintus metodus traktuoja vienodai pagal nekintamą objekto koncepciją.

Žvelgiant į sistemą kaip į bendraujančių objektų visumą yra tinkamas būdas atlikti sistemos struktūrizavimą, sudarant nagrinėjamą sistemos identifikavimą ir atitinkamą objektų vaizdavimą ir ryšius tarp objektų. Skirtingi OO metodai pateikia eilę būdų ir diagramų aprašyti keturias prieš tai apibrėžtas sritis. Tam kad suvoktume visą šita informaciją, objektiškai orientuotas metodas naudoja keturis modelius, kurie apibrėžia objektų visumą keturiais vienas kitą papildančiais požiūriais. Tikrovėje, kiekvienas modelis objektiškai orientuotame metode pateikia tam tikro konceptualios schemos aspekto vaizdą. Tikslas yra padaryti sistemos



specifikaciją, kuri būtų konceptualios schemos rezultatas, paprastesnę ir funkcionalesnę. Kiekvienas modelis pateikia eilę metodų (kartu su atitinkamomis UML diagramomis), tam kad pateiktą informaciją, kuri yra svarbi. Objektiškai orientuotas metodas ypatingai formuoja konceptualios schemos kūrimą naudojant šiuos modelius:

- Objektinis modelis, aprašo struktūrą ir statinius ryšius tarp klasių, kurios turi būti įvardintos probleminėje srityje.
- Dinaminis modelis, apibūdina galimas duotos klasės objektų veiksmų sekas ir koku atžvilgiu vyksta bendravimas tarp objektų.
- Funkcinis modelis, apima semantiką susijusią su objekto būsenų pasikeitimais, kurie atsiranda dėl sistemoje vykstančių įvykių.
- Pateikimo modelis, praplečia tradicinį konceptualinį modeliavimą įtraukiant vaizdą, atsakingą už informacijos, aprašančios vartotojo sąsajos charakteristikas bei būdus, kuriais vartotojas bendraus su sistema, užfiksavimą.

Pirmieji trys modeliai aprašo statinę ir dinaminę informaciją, kuri yra reikalinga objektiškai orientuotos konceptualios schemos kūrimui. Pateikimo modelis pateikia ketvirtą vaizdą, kuris yra reikalingas aprašyti kaip vartotojai sąveikaus su sistema. Todėl, tai kaip vartotojai bendraus su sistema ir supras pačią sistemą tampa pagrindiniu modeliavimo komponentu, visai kaip statinių ir dinaminių aspektų specifikavimas, aprašytas aukščiau.

Išsiaiškinti kaip vyks bendravimas tarp vartotojų ir sistemos, probleminės srities lygyje, taip pat yra labai svarbus. Ši informacija yra užfiksuojuama, panaudojant sąsajos šablonus. Juos nustatius konceptualaus modeliavimo srityje, šie šablonai leis apibrėžti kiekvienos sąsajos savybes, nepriklausomai nuo tam tikro įgyvendinimo, kuris bus galiausiai pasiektas, kaip galutinio programinės įrangos produkto, dalis.

### **1.1.1 Pateikimo modelis**


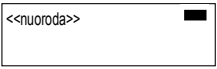

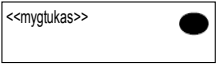
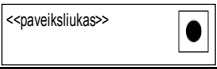


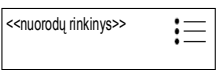
[BAUM] Kalbant apie pateikimo modelį, H. Baumeister, N. Koch ir L. Mandel savo pranešime pasiūlė pirmiausia jį apibrėžti kaip vartotojo sąsajos komponentų, tokių kaip tekstas, nuorodos, formos, mygtukai ir kt., rinkinį. Pateikimo modelis yra schemiška vartotojo sąsaja, kuri duoda suprasti šių elementų vietą, santykinį dydį, bet neparodo jų galutinės išvaizdos.

Pateikimo modelis susideda iš statinio pateikimo modelio ir dinaminio pateikimo modelio. Statinis pateikimo modelis yra pateikiamas, naudojant UML kompozicines diagramas, kurios paaiškina kaip vartotojo sąsaja yra sudaryta iš vartotojo sąsajos komponentų. UML būsenų

diagramos, siekiant paaiškinti šių elementų veikimą, yra naudojamos dinamiame pateikimo modelyje.

Statiniame pateikimo modelyje, vartotojo sąsajos objektas gali būti paprastas vartotojo sąsajos elementas pvz., tekstas ar mygtukas, arba vartotojo sąsajos objektų rinkinys. Tokio rinkinio ypatingas tipas yra pateikimo objektas, kuris priklauso nuo navigacinio objekto būsenos. Vartotojo sąsajos objektai gali turėti savo būsenas, kaip pvz., mygtukas kurio būsena gali būti paspaustas arba nepaspaustas, ir šitas būsenas savo veiksmis keičia vartotojas.

Žemiau esančioje lentelėje Lentelė Nr. 1, pateikiu dažniausiai naudojamus vartotojo sąsajos objektus bei jų semantiką:

Grafinis vaizdas	Paiškinimas
	Pateikimo objektas yra vartotojo sąsajos objektas, kuris priklauso nuo navigacinio mazgo būsenos, ir kurį sudaro kiti vartotojo sąsajos objektai tame tarpe ir kiti pateikimo objektai.
	Nuoroda yra sritis, kurią galima paspausti ir kuri yra navigacijos pradinis taškas.
	Tekstas yra simbolių, pateiktų kartu su formatavimo informacija, seka.
	Mygtukas yra sritis, kurią galima paspausti ir kuri yra susieta su tam tikru veiksmu, aprašyti dinamiame pateikimo modelyje.
	Paveiksliukas yra daugialypės terpės objektas.
	Forma yra skirta vartotoju įvesti informaciją pvz., įvesti vardą, arba paieškos žodį
	Vartotojo sąsajos objektų rinkinys yra pateikiamas kaip patogus būdas, pakartotinai pateikti vartotojo sąsajos objektus.
	Nuorodų rinkinys yra pateikiamas kaip patogus būdas, pakartotinai pateikti nuorodų rinkinį.

Lentelė Nr. 1 Vartotojo sąsajos objektai ir jų semantika

## 1.2 Modelių transformacijos

Modeliai ir modeliavimo metodai yra būdas nagrinėti ir valdyti šiuolaikinių sistemų uždavinių apimtį ir sudėtingumą naudojant įvairias abstrakcijas. Pats modelis yra tam tikra abstrakcijos rūšis, įgalinanti atvaizduoti sistemą ar jos dalį abstrakčiau, išryškinant esmines savybes, praleidžiant ar paslepianč mažiau svarbias.

[STUI] Literatūroje galima aptikti keletą modelių transformacijų apibrėžčių. Viena jų skelbia, kad modelio transformavimas yra „automatinis paskirties modelio sukūrimas iš pirminio modelio pagal transformavimo apibrėžtį“. Transformavimo apibrėžtis yra transformavimo taisyklių aibė, nustatanti kaip paskirties modelio aprašas gaunamas iš pirminio modelio aprašo. Transformavimo taisyklė yra aprašas, nurodantis kaip viena ar kelios pirminės kalbos konstrukcijos gali būti transformuojamos į vieną ar kelias paskirties kalbos konstrukcijas. Dar viena apibrėžtis teigia, kad modelių transformacija – tai „vienos modelių aibės atvaizdavimas (mapping) į kitą arba save pačią, kur projektuotojas nustato atitikimus tarp elementų pradiniam ir paskirties modeliuose“. Ši apibrėžtis neakcentuoja automatinio modelių kūrimo, taigi numato ir rankinį projektavimą, kuris yra paplitęs praktikoje, ypač kai nagrinėjami aukšto lygmens modeliai.

Norint transformuoti modelius, pirmiausia juos reikia išreikšti kuria nors modeliavimo kalba (pvz., struktūriniais modeliams galima naudoti UML). Pagal T. Menzo (Mens) modelių transformacijas galima skirstyti į *heterogeninę* (lot. exogenous) ir *homogeninę* (lot. endogenous) transformacijas. Heterogeninė (egzogeninė) transformacija yra tokia transformacija, kai pirminis ir paskirties modeliai yra išreikšti skirtingomis kalbomis, pvz., kodo generavimas, apgražos inžinerija, migravimas. Kadangi tiek pirminiai, tiek paskirties modeliai gali būti keli, modelių transformacijose gali būti panaudotos kelios skirtingos kalbos. Homogeninė (endogeninė) transformacija yra transformacija tarp modelių, užrašytų ta pačia kalba, pvz., optimizavimas, faktorizavimas, prastinimas.

[STUI] Heterogeninei transformacijai apibūdinti yra naudojamas transliacijos terminas, homogeninei transformacijai apibūdinti – perfrazavimo terminas. T. Menzo dar skiria ir horizontaliąsias ir vertikaliąsias transformacijas. Horizontalioji transformacija yra tokia, kai pirminis ir paskirties modeliai yra tame pačiame abstrakcijos lygmenyje (pvz. sudalinimas). Ir priešingai – vertikalojoje transformacijoje (pvz., detalizavimas) dalyvaujantys modeliai yra atvaizduojami skirtinguose abstrakcijos lygmenyse.

Modelių transformacijų svarbios charakteristikos yra tokios:

- *Automatizavimo laipsnis* (rankinis, pusiau rankinis, automatinis)
- *Sudėtingumas* (kuris gali būti įvertintas įvairiais metodais ir įrankiais)
- *Tam tikrų savybių išsaugojimas* (pvz., atliekant sudalinimą reikia išsaugoti elgseną, kai tuo tarpu struktūra gali būti pakeista).

Reikia pabrėžti, kad dauguma modelių transformacijų koncepcijų priklauso nuo sistemos požiūrių aibės, kurią gali palaikyti modeliavimo kalba. Pavyzdžiui, UML 2.0 gerai palaiko koncepcijų atskirtį ir pateikia vartotojams galimybę modeliuoti sistemas keturiais skirtingais požiūriais:

- *Statiniu struktūriniu* (klasės modeliai, kurie naudojami sistemos struktūrai aprašyti);
- *Sąveikos* (aprašo sąveiką ir komunikavimo seką tarp sąveikaujančių modelių egzempliorių);
- *Veiklos* (požiūris naudojamas, kai reikia sukurti modelius, kurie aprašo veiklos srautus sistemos viduje);
- *Būsenų* (kai reikia sukurti būsenų automatus, kurie aprašo elgseną per būsenas ir perėjimus tarp būsenų)

Aukštesnio lygmens modelių transformacijos gali būti atliekamos arba rankiniu būdu, arba automatiškai. O žemesnio lygmens modelių transformacijos visada atliekamos automatiškai.

Transformacija yra fundamentali kompiuterijos ir programų inžinerijos sąvoka, nes kiekvieną skaičiavimo operaciją galima apibrėžti kaip duomenų transformaciją. Modelių transformacija yra vieno ar daugiau pirminių modelių vertimas į paskirties modelį.

Objektinis srities sistemos modelis gali būti atvaizduojamas trijuose abstrakcijos lygmenyse:

1. *Realizavimo lygmenyje* – klasės atvaizduoja programos kodo dalis, parašytas objektine programavimo kalba;
2. *Specifikavimo lygmenyje* – klasės specifikuoja sistemos sąsajas aukštesniame abstrakcijos lygmenyje;
3. *Konceptualiajame lygmenyje* – klasės atvaizduoja abstrakčias tyrimo srities sąvokas, pvz., platformas.

### **1.3 Projekto tikslas**

Šis projektas skirtas renginio planavimo etapui supaprastinti. Veiksmai, kuriuos atlieka sistema, palengvina naudotojo darbą išsprendžiant konkrečius pirmojo etapo klausimus. Sistema yra orientuota į šias renginius organizuojančias žmonių grupes:

- Renginių organizavimo kompanijos.
- Verslo kompanijos.
- Fiziniai asmenys.
- Valstybinės institucijos.

Sistema yra aktuali visoms aukščiau išvardintoms žmonių grupėms, kuriuos organizuoja privačius arba viešus renginius. Sistemoje galima paskelbti tiek privačius, tiek viešus renginius, tačiau sistema yra labiau orientuota į asmenis, organizuojančius privačius renginius, kur yra svarbu pakviesti konkrečius asmenis į renginį.

Kuriant sistemą keliamas tikslas ją padaryti lengvai naudojamą įvairių sričių renginių organizavimui. Sistemos taikymas buvo kuriamas sprendžiant asmenines renginių organizatoriaus problemas, naudojantis jo ilgamete patirtimi organizavimo srityje.

### **1.4 Renginių organizavimo problemos analizė**

Šiame skyriuje apžvelgiamos technologijos bei sprendimai, kurie bus naudoti, kuriant renginių organizavimo sistemą. Taip pat apžvelgiami įrankiai kurių pagalba bus galima realizuoti šią sistemą.

#### **1.4.1 Kūrimo problemos**

Kuriant internetinę aplikaciją projektuotojai turi atlikti daugybę pasirinkimų [7]. Kokią sistemos architektūrą pasirinkti, kaip atlikti patį projektavimą, kokius įrankius naudojami kūrimo procese. Tai labai svarbus etapas nes nuo šio etapo atliktų sprendimų pasirinkti vienokį ar kitokį variantą didžiaja dalimi priklausys viso projekto sėkmė.

##### **1.4.1.1 Architektūra**

###### **1.4.1.1.1 MVV**

[9] MVV reiškia Modelis, Vaizdas, Valdiklis (angl. MVC – Model, View, Controller).

Tai architektūrinis šablonas, kuris šiuo metu yra be galo populiarus. Šiuo architektūriniu stiliumi yra paremti beveik visi populiariausi internetinių aplikacijų karkasai. [10]

**Modelis.** Modelis pateikia pačią internetinės sistemos informaciją ir turinį. Tai yra objektas, galintis surinkti informaciją ir turinį iš sistemos puslapių iš tam tikro šaltinio, pvz.: duomenų bazės. Modelis yra vienintelis objektas, galintis pasiekti informaciją iš šaltinių.

**Vaizdas.** Vaizdas yra mėginimas atskirti nestabiliausią sistemos dalį – vartotojo sąsają. Vaizdas yra reikalingas informacijos, kurią surinko modelis, atvaizdavimui. Vaizdas yra skirtas suformuoti išvedamiems duomenims. Visose MVV sistemose tik vaizdo failai saugo (X)HTML kodą. Jie gali būti keičiami bet kuriuo metu, visai neįtakojant sistemos funkcionalumo.

**Valdiklis.** Valdiklis yra sistemos smegenys. Tai yra vienintelė sistemos dalis kuri valdo kitas sistemos dalis. Kontrolerio failai vieninteliai saugo informaciją, kurios pagalba modelis surenka reikiamą informaciją ir perduoda ją vaizdui, kuris ir išveda norimą informaciją.

#### **1.4.1.1.2 SOA**

SOA(Service Oriented Architecture) – tai programinės įrangos architektūros stilius. Išvertus pažodžiui SOA reiškia į paslaugas orientuota architektūra. Šis terminas atsirado kartu su terminu WEB 2.0 apie kurį kalbėsime vėliau. Šis architektūrinis stilius leidžia išorinėms sistemoms naudotis Jūsų sistemos teikiama informacija. Tai yra jis apibrėžia paslaugos tiekėją ir paslaugos naudotoją. Paslauga tai yra tam tikra sistemos teikiama funkcija kurios veikimas nepriklauso nuo išorinės sistemos (paslaugos naudotojo) nustatymų, naudojamos programinės kalbos, ar įrangos. Duomenys tarp tiekėjo ir naudotojo yra perduodami XML formatu naudojant eilę protokolų, iš kurių populiariausi yra SOAP ir REST. [11] Taip pat jis supaprastina sistemos plečiamumą, palaikomumą, integralumą.

#### **Privalumai:**

Nepriklausoma platforma – kadangi tinklo paslaugos (Web service) gali būti teikiamos ir naudojamos įvairių serverių, klientų, su įvairiomis operacinėmis sistemomis, ši architektūra leidžia nekreipti į tai dėmesio ir teikti paslaugas įvairiems klientams neatsižvelgiant į jų techninę ar programinę įrangą.

Kodo pakartotinis panaudojimas – kadangi šis architektūrinis stilius suskaldo programą į smulkius kodo fragmentus, atliekančius tam tikrą funkciją, kurios gali būti pakartotinai panaudotos keliose internetinėse aplikacijose.

Lygiagretus kūrimas – kadangi paslaugos viena nuo kitos nepriklauso, jos gali būti kuriamos atskirai, kas sutrumpina bendra aplikacijos kūrimą.

Didesnis plečiamumas – kadangi paslaugos teikimo vieta neturi įtakos paslauga gali būti nejučiom perkelta į galingesnius serverius.

Didesnis prieinamumas – internetinė paslauga gali veikti keliuose serveriuose vienu metu, kas įgalina suteikti didesnę paslaugos pasiekiamumo tikimybę.

#### **1.4.1.1.3 REST**

Šis terminas pirmą kartą buvo paminėtas Roy Fielding daktaro laipsnio disertacijoje, kaip architektūrinis tinklu sujungtų sistemų stilius. REST reiškia Vaizduojamas Būsenų Keitimas (angl. Representational State Transfer). [12]

Pavyzdžiui yra teikiama paslauga adresu [www.serveris.lt/paslauga/5](http://www.serveris.lt/paslauga/5). Kai vartotojas savo naršyklėje įrašo šį adresą naršyklei pateikiamas rezultatas (pavyzdžiui [kainos.html](#)), dėl ko pasikeičia naršyklės būseną. Naršyklės būseną keičiasi su kiekvienu resurso atvaizdavimu iš ko ir kilo šio stiliaus pavadinimas.

REST nėra standartas, Jūs nerasite jo specifikavimo pateikiamo W3C tinklalapyje. Taip pat nei IBM ar SUN nesukurs REST kūrėjo įrankių rinkinio. Kodėl? Todėl kad tai tik architektūrinis stilius, kurį galite naudoti kurdami savo internetinę paslaugą.

**Nors REST ir nėra standartas jis naudoja standartus:**

- HTTP
- URL
- XML/HTML/GIF/JPEG/ (resurso vaizdavimui)
- text/xml, text/html, image/gif, image/jpeg, etc (MIME Tipai)
- REST charakteristikos:
  - Klientas-serveris – komunikavimo stilius pagrįstas ištrakimu, vartojantys komponentai ištraukia atvaizdą
  - Būsenų nebuvimas – kiekvienas informacijos užklauskimas iš kliento į serverį savyje turi turėti visą informaciją tam kad būtų įmanoma apdoroti užklauskimą
  - Podėlis (cache) - tam kad būtų galima pagreitinti užklauskų perdavimą, kiekvienas atsakas iš serverio turi būti pažymėtas kaip podėliuojamas (cacheable) arba nepodėliuojamas (non-cacheable).

- Universali sąsaja – visi šaltiniai yra pasiekiami naudojant daugybinę sąsają (pvz.: HTTP GET, POST, PUT, DELETE)
- Šaltinių vardai – sistemą sudaro šaltiniai kurie pavadinami naudojant URL
- Sujungti šaltinių atvaizdai – šaltinių atvaizdai yra sujungiami naudojant URL taip įgalinant klientą pereidinėti iš vienos būsenos į kitą.

## 1.4.1.2 Projektavimas

### 1.4.1.2.1 *Domain specific language*

DSL (agl. domain-specific language) – yra programavimo kalba, kuri yra sukurta ir numatyta, kad būtų naudinga konkrečiam tipo uždaviniui spręsti.

Dauguma DSL yra labiau aprašomi, nei procedūriniai. Kai kurie DSL yra paremti grafiniais elementais, vietoj tekstinių. Bet kuri DSL reikalauja konkrečios srities Transformacinio Variklio, kuris ją interpretuotų. DSL yra sukurtos konkrečiam sprendimui tam tikroje srityje, ir neketinama padaryti tai, kad ji spręstų problemas, kurios nepatenka į šią probleminę sritį. DSL metodologijos sėkmė priklausys, kaip dažnai ši realizuota struktūra gali būti keičiama, kas įtakos pakitimus ir pačioje kalboje [24]. Patirtis parodė, kad laikas DSL kalbai subręsti, gali užtrukti net kelerius metus [25].

[24] Srities ekspertai dažnai galvoja apie aukšto lygio abstrakcijas, kol kuria sprendimų procedūras. Kaip rezultatas į tai, kuo žemesnio lygio programavimo kalba, tuo platesnė semantinė spraga, ir daug sunkesnės vartotojų užduotys. Jeigu turėtume kalbą lygiai tokio pačio lygio, kaip vartotojas mąsto, užduotis sukurti programą būtų daug lengvesnė. DSL iš tikrųjų tai ir leidžia padaryti.

Labai gerai visiems žinomas DSL pavyzdys yra HTML, kuris išgarsėjo, kaip žymų kalba hipertekstui žiniatinklyje.

#### **Privalumai:**

- DSL leidžia sprendimams būti išreikštiems kalbos savitumu tokia pačia lygyje kaip ir problema.
- Pats save dokumentuojantis kodas.
- DSL pagerina kokybę, produktyvumą, patikimumą, palaikomumą ir pakartotinį panaudojamumą
- DSL leidžia valiavimą konkrečios srities lygyje.



### **Trūkumai:**

- Didelė sukūrimo, realizavimo ir palaikymo kaina.
- Surasti, nustatyti ir plėtoti tinkamą sritį.
- Sunkumas suderinant skirtumus tarp konkrečios srities kalbos ir bendro naudojimo programavimo kalbos.
- Galimas procesoriaus efektyvumo sumažėjimas, lyginant su „rankomis“ suprogramuotos programinės įrangos.
- Sunkus derinimas (angl. „debug“).

#### **1.4.1.2.2 UML**

UML (**Unified Modeling Language**, *Vieninga modeliavimo kalba*) – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus. Tai yra grafinio modeliavimo kalba, kuri naudoja vieningą terminologiją. UML tai yra būdas vizualizuoti, specifikuoti ir dokumentuoti vystomus projektus.

#### **UML technologijos privalumai**

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

UML yra įrankis, skirtas specifikuoti programinės įrangos sistemas. Standartizuoti diagramų tipai padeda apibūdinti ir vizualiai nustatyti sistemos projektavimą ir struktūrą. Naudojant UML yra įmanoma sumodeliuoti bet kokio tipo taikomąją programą, kuri yra konkreti ir nepriklauso nuo jokios platformos. Nors UML yra orientuota į objektiškai orientuotą programavimą, bet su ja galima tai pat lengvai sumodeliuoti ir procedūrinėms kalboms, tokioms kaip C, Visual Basic ar Fortan.

UML kaip įrankio naudojimas apibrėžiant sistemos struktūrą yra rezultatyvus būdas tvarkyti dideles ir sudėtingas sistemas. Turint aiškiai matomą struktūrą yra daug lengviau įtraukti naujus žmones į jau egzistuojantį projektą.

UML [13] šablonų bibliotekų sudarymas supaprastina pakartotinių modelių ir kodo panaudojimą. UML šablonai yra grupės tarpusavyje bendradarbiaujančių objektų/klasijų, kurios gali būti paveldimos oš bendro naudojimo modeliavimo scenarijų. Kai šablonai yra kuriami naujam projektui, turimi šablonų modeliai gali būti pakartotinai panaudoti ir pakeisti projektuotojo, kad atitiktų naujo projekto poreikius.

UML diagramos yra klasifikuojamos į tris tipus [14]:

- Elgsenos diagramos. Tai toks diagramų tipas kuris vaizduoja sistemos elgsenos bruožus arba veiklos procesus. Jas sudaro būsenų ir veiklos diagramos.
- Sąveikos diagramos. Tai yra elgsenos diagramų potipis, kuris pabrėžia objektų sąveiką. Jas sudaro sekų ir bendradarbiavimo diagramos.
- Struktūrinės diagramos. Tai tokios diagramos, kurios vaizduoja reikalavimų elementus, kurie nėra priklausomi nuo laiko. Tai yra komponentų ir išdėstymo diagramos.

Žemiau yra pateiktas pagrindinių naudojamų UML diagramų sąrašas anglų kalba, kurios yra suskirstytos į tam tikrus diagramų tipus [15]:

- Use Case Diagram – panaudos atvejų diagrama.
- Class Diagram – klasių diagrama.

#### **Elgsenos diagramos**

- Statechart Diagram – būsenų diagrama.
- Activity Diagram – veiklos diagrama.

#### **Sąveikos diagramos**

- Sequence Diagram – sekų diagrama.
- Collaboration Diagram – bendradarbiavimo diagrama.

#### **Struktūrinės diagramos**

- Component Diagram – komponentų diagrama.
- Deployment Diagram – išdėstymo diagrama.

#### **Praktinės projektavimo problemos**

Reikia pripažinti, kad praktikoje taikomi projektavimo metodai dažnai neduoda naudos, kurią akcentuoja teorija. Šių pastebėjimų dėka galima teigti, kad ir pats projektavimas nėra stipriai akcentuojama veikla:

- programinę įrangą kuriančiose kompanijose dažnai nėra išskirta sistemų architekto specializacija;
- daug programinę įrangą kuriančių įmonių nenaudoja programinės įrangos projektavimui skirtos UML modeliavimo kalbos ir jai specializuotų įrankių;
- programinės įrangos architektūros dokumentacija dažnai nėra ruošiamą.

Viena neefektyvaus projektavimo priežasčių – per daug detaliam modeliuojami dažnai kintantys realizacijos lygio aspektai, tuo tarpu aukštesnio lygio sistemos architektūros projektavimas vykdomas nepakankamai. Todėl dažnai projektavimo modeliai vertinami pagal detalumo lygį, tačiau nepastebimi prasti projektavimo sprendimai, kuriuose nesilaikoma projektavimo principų. Taip pat nėra išlaikomi ryšiai tarp reikalavimų analizės, aukšto lygio architektūrinių sprendimų bei detalių realizacijos sprendimų modelių – neišskiriami ir nesusiejami skirtingi modeliavimo abstrakcijos lygiai. Daugelyje įmonių nėra naudojama UML modeliavimo kalba, o projektavimo sprendimai aprašomi tekstu arba paprastomis diagramomis, nenaudojant sutartinės notacijos. Tai sukelia komunikacijos problemų bei apriboja projektavimo sprendimų dokumentavimo galimybes.

#### **1.4.1.2.3 Kūrimo Įrankiai**

Programavimo darbai atliekami naudojantis CakePHP darbinio pagrindu. (<http://www.cakephp.org/>)

CakePHP veikia PHP programavimo kalbos pagrindu. PHP yra atviro kodo kalba, kuri yra nemokama ir populiariausia programavimo kalba internetinių sistemų kūrimui.

Grynos PHP kalbos naudojimas sukuria keletą sunkumų didelių internetinių sistemų kūrėjams, kadangi nėra apibrėžto ir struktūrizuoto php kalbos pagrindo. Pavojus slypi ne PHP programavimo kalboje, bet jos panaudojime, rašant didelės apimties internetines sistemas: PHP kalbos paprastumas rašomą sistemos kodą padaro neapibrėžtą, nestruktūrizuotą ir nesutvarkytą. Parašytos sistemos kodas būna prastai organizuotas ir nesutvarkytas, o tai labai apsunkina tolimesnį sistemos plėtojimą.

Šioms problemoms spręsti yra sugalvota daugybė kuriamų sistemų darbinių pagrindų (angl. frameworks), kurie padeda struktūrizuoti visą PHP kalbos kodą.

CakePHP siūlo organizuotą ir struktūrizuotą PHP kalbos ir duomenų bazės sąveikavimo šabloną, dar vadinamą – Modelis Vaizdas Valdiklis (angl. Model View Controller). Šie trys žodžiai ir nusako PHP kalba parašytos sistemos kodo struktūrą:

- Modelis: Modelis pateikia pačią internetinės sistemos informaciją ir turinį. Tai yra objektas, galintis surinkti informaciją ir turinį iš sistemos puslapių iš tam tikro šaltinio, pvz: duomenų bazės. Modelis yra vienintelis objektas, galintis pasiekti informaciją iš šaltinių.
- Vaizdas: Vaizdas yra mėginimas atskirti nestabiliausią sistemos dalį – vartotojo sąsają. Vaizdas yra reikalingas informacijos, kurią surinko modelis, atvaizdavimui. Vaizdas yra skirtas suformuoti išvedamiems duomenims. Visose MVK sistemose tik vaizdo failai saugo (X)HTML kodą. Jie gali būti keičiami bet kuriuo metu, visai neįtakojant sistemos funkcionalumo.
- Valdiklis: Valdiklis yra sistemos smegenys. Tai yra vienintelė sistemos dalis kuri valdo kitas sistemos dalis. Kontrolerio failai vieninteliai saugo informaciją, kurios pagalba modelis surenka reikiamą informaciją ir perduoda ją vaizdui, kuris ir išveda norimą informaciją.

Javascript darbinis pagrindas „jQuery“ (<http://jquery.com/>). Tai JavaScript biblioteka sutrumpinanti kodo rašymą, o atliekanti daugiau. Naudojantis šia biblioteka nesunku įgyvendinti ajax užklausas, valdyti HTML kodą, bei pritaikyti efektus.

### **Magic draw**

Kuriant programinę įrangą, labai svarbu suprantamai ir išsamiai apibrėžti vartotojų poreikius, sukurti efektyvų programinės įrangos modelį, parengti dokumentaciją. Naudojantis "MagicDraw UML" galima lengvai atlikti šiuos uždavinius ir sutaupyti laiko bei lėšų kituose programinės įrangos kūrimo etapuose.

MagicDraw [16] UML atitinka naujausius Java bei UML technologijų standartus, turi vieną iš patikimiausių išeities kodų inžinerijos mechanizmų Java, C#, C++ ir CORBA IDL programavimo kalboms bei gali vykdyti šių kalbų kodo atvirkštinę inžineriją, duomenų bazių schemų atvirkštinę inžineriją, kodo bei duomenų bazių schemų generavimą. MagicDraw UML panaudoja "roundtrip" technologiją, leidžiančią keisti tiek OO modelį, tiek programos kodą bet koku eiliškumu, juos nuolat sinchronizuojant. MagicDraw UML yra vienas iš nedaugelio rinkoje esančių paketų, leidžiančių greičiau nubraižyti UML diagramas bei turintis UML diagramų semantinio teisingumo tikrinimo ir modelio validavimo mechanizmą. MagicDraw UML Teamwork Server programinė įranga turi komandinio darbo galimybę, leisdama daugeliui programinės įrangos kūrimo inžinierių dirbti su tuo pačiu OO modeliu vienu metu.

UML redaktorius - patogi vartotojo sąsaja sudaro sąlygas greitai ir paprastai užrašyti idėjas UML diagramomis. Pilnai palaikomas UML 1.4 standartas.

UML modelio ataskaitų kūrimo(dokumentavimo) priemonė - automatinis ataskaitų kūrimo mechanizmas leidžia greitai sukurti išsamią reikalavimų, dizaino ar kito tipo dokumentaciją HTML, PDF ir RTF formatais. Galima pasirinkti į ataskaitą įtraukiamas modelio dalis ir suformuoti norimą ataskaitų apipavidalinimą.

OO modelio analizės priemonė - "MagicDraw UML" kuria "package dependency" diagramas, sudaro paveldimumo medžius, leidžia sekti, kaip metodų, parametrų ar grąžinamų reikšmių kitimas turi įtakos UML modelio kitimui.

OO modelio kodo generatorius - generuoja programos kodą labiausiai paplitusiems modelio šablonams (design patterns) įskaitant ir GoF, Java, Junit modelio šablonus. Java ir Jpython pagalba galima kurti savus modelio šablonus išplečiant jau esamus modelio šablonus.

Duomenų bazių modeliavimo priemonė - galima sukurti DB struktūrą pasinaudojant UML klasių diagramų kūrimo priemonėmis ir sugeneruoti DDL kodą pagal sukurtą diagramą. Taip pat yra galimybė per JDBC užkrauti jau sukurtą DB ir matyti jos diagramą.

### **Zend Studio**

„Zend Studio“ – tai tūkstančių gamintojų visame pasaulyje pripažintas PHP priedų kūrimo būdas. „Zend Studio“ siūlo integruotą plėtros terpę ir įgalina neatidėliotiną priedų kūrimą. Vartotojams suteikiamos patikimos gamybos priemonės, patikrinimai ir priedų išskleidimai visose paplitusiose aplinkose (įskaitant Windows, Linux ir Mac) [17].

„Zend Studio“ yra integruota programavimo aplinka skirta tiek profesionaliems, tiek pradedantiesiems programuotojams. Šios programavimo aplinkos turėtų užtikrinti produkto kokybę, taip pat sistemos kūrimo proceso valdymą ir atlikimą laiku.

Šis produktas padidina produktyvumą su patikrinta programavimo aplinka, [18]į kurią įeina PHPį palaikymas, kodo analizavimas, lizdinis kodo užbaigimas, sintaksės išryškėjimas, projekto valdiklis, kodo redaktorius, grafinė derinimo programa ir daugybė įvairiausių vedlių. Šis įrankis taip pat palaiko ir versijavimą. Labai paprasta ir greita navigacija, jeigu reikia surasti tam tikrą kodo dalį, yra galimybė filtruoti pagal klases, funkcijas ir kintamuosius.

### **Subversion**

Subversion [19] - atviro kodo bylų versijų kontrolės sistema. Ši sistema tvarko failus, direktorijas, bei pakeitimus juose. Šiuolaikinės versijos kontrolės sistemos atlieka ir daugiau

funkcijų: sujungia keleto vartotojų vienu metu atliktus pakeitimus, kaupia duomenis apie tai, kas ir kaip keitė dokumentus, sukuria tekstus, glaustai parodančius, kaip pasikeitė dokumento turinys bei leidžia tinklo pagalba keisti šį turinį iš bet kurios pasaulio vietos. Versijų kontrolės sistema yra vienas pagrindinių laisvos programinės įrangos kūrimo įrankių, tačiau ji naudojama ir kitokiai techninei dokumentacijai rengti.

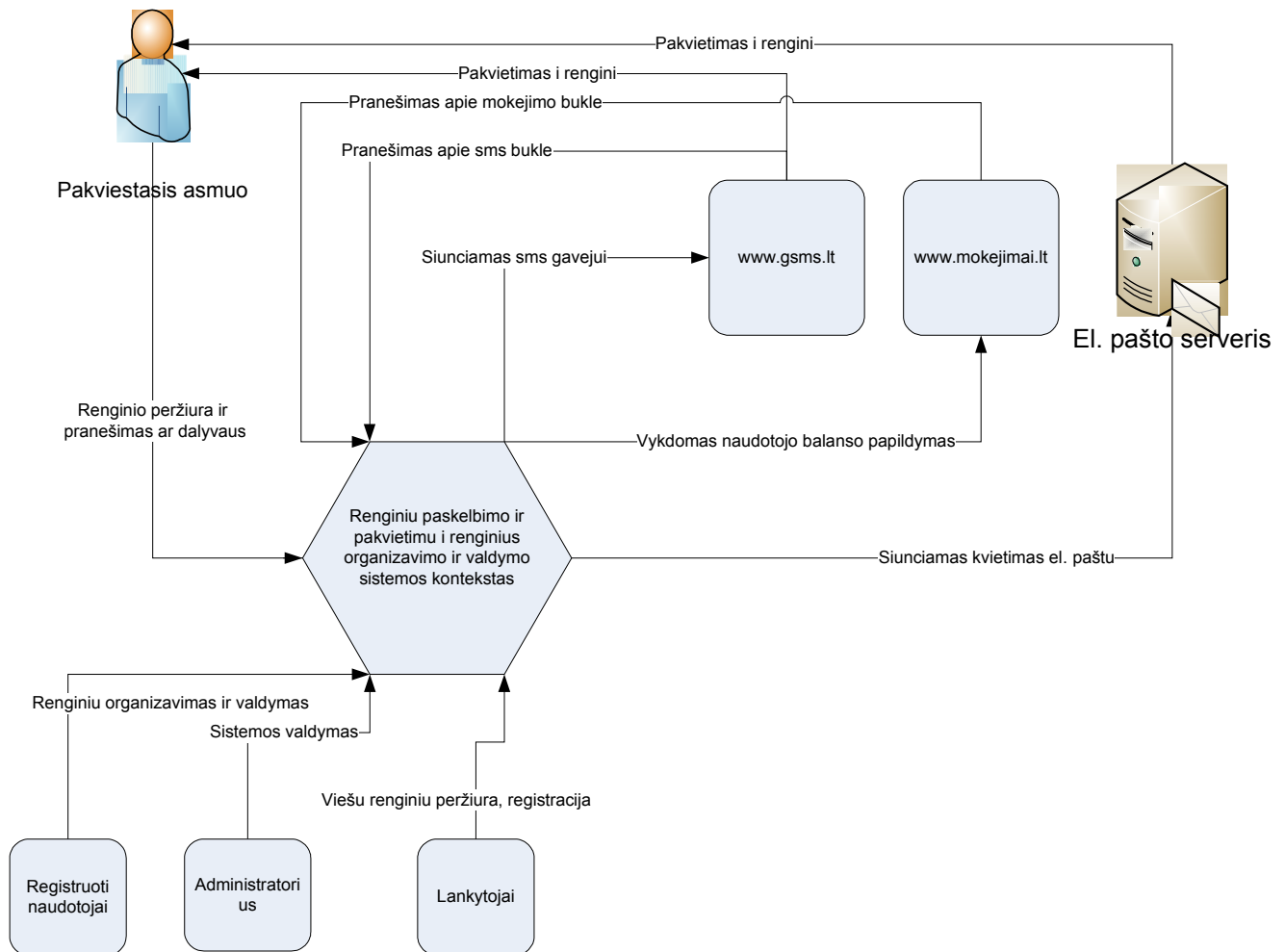
#### **1.4.2 Sistemos kontekstas**

Kuriama sistema bus įgyvendinta kaip tinklo programa (WEB application). Ją bus galima pasiekti iš bet kurio kompiuterio, kuris yra prijungtas prie interneto ryšio, adresu [www.invup.lt](http://www.invup.lt). Visa sistemos naudojama informacija bus saugojama duomenų bazėje. Sistema naudoja savo vartotojų autentifikaciją. Visi veiksmai kuriamoje sistemoje atliekami naudojant tą pačią vartotojo sąsają, skiriasi tik sąsajos panaudojimo galimybės, priklausomai nuo vartotojo teisių sistemoje. Sistema yra pritaikyta konkrečiai problemai spręsti ir pakartotinis jos panaudojimas yra įmanomas tik tam tikriems sistemos komponentams. Kuriama sistema yra priklausoma nuo kitų sistemų:

[www.mokejimai.lt](http://www.mokejimai.lt) – sistema per kuria atliekami piniginiai pervedimai.

[www.gsms.lt](http://www.gsms.lt) – sistema per kurią išsiunčiamos trumposios sms žinutės.

Sistemos projekto vizualus ryšių atvaizdavimas:



Pav. 1 Sistemos kontekstas

### 1.4.3 Vartotojo charakteristikos

Naudotojas norėdamas naudotis informacine sistema privalo mokėti naudotis internetine naršykle, bei turėti bendrą supratimą kaip naudotis teksto redaktoriumi.

Taip pat naudotojas turi turėti personalinį kompiuterį, kuris būtų prijungtas prie interneto. Naudotojo kompiuterio minimalūs parametrai turi būti tokie, kad palaikytų normalų šiuolaikinės internetinės naršyklės darbą. Kiekviena naršyklė turi skirtingus reikalavimus aparatinei įrangai. Mes siūlome naudotis Firefox internetine naršykle, kadangi ji veikia sparčiau, bei yra saugesnė, be to galima gauti įvairių priedų, kurie labai palengvina darbą internete, ir yra nemokami. Firefox naršyklės rekomenduojami reikalavimai kompiuteriui yra tokie:

- 500 Mhz procesorius
- 256 operatyviosios atminties
- 100 MB laisvos vietos diske

- Microsoft Windows XP operacinė sistema

#### **1.4.4 Vartotojo problemos**

Ši sistema yra skirta renginių planavimo ir organizavimo problemoms spręsti. Projekto tikslas yra užpildyti tokio pobūdžio sistemų trūkumo nišą. Sistema yra skirta ne tik Lietuvos rinkai. Sistemoje yra įdiegta daugiakalbystė. Projekto pradžioje yra numatyta įgyvendinti sistemą, kuri puikiai funkcionuotų dviem kalbomis – tai anglų ir lietuvių. Sistema yra orientuota į šias renginius organizuojančias žmonių grupes:

- Renginių organizavimo kompanijos.
- Verslo kompanijos.
- Fiziniai asmenys.
- Valstybinės institucijos.

Sistema yra aktuali visoms aukščiau išvardintoms žmonių grupėms, kuriuos organizuoja privačius arba viešus renginius. Sistemoje galima paskelbti tiek privačius, tiek viešus renginius, tačiau sistema yra labiau orientuota į asmenis, organizuojančius privačius renginius, kur yra svarbu pakviesti konkrečius asmenis į renginį.

#### **1.4.5 Vartotojo tikslai**

Produktas bus orientuotas į vyresniąją vartotojų kategoriją. Sistemos išvaizda turi būti solidi, patraukli akiai, tačiau ne per daug spalvota ir neapkrauta. Iš pirmo žvilgsnio sistemos išvaizda turi sudaryti pasitikėjimo ir ramumo jausmą. Išvaizdai keliamas pagrindinis reikalavimas, kad ji atrodytų paprasta ir elementari, lengva naudotis, tačiau tuo pačiu turi netrūkti komponentų, palaikančių

visišką sistemos funkcionalumą. Taip pat išvaizda turi sudaryti ir privatumo bei konfidencialumo jausmą, kad vartotojas, tik išvydęs dizainą, nesusidarytų pigaus ir nepatikimo puslapio įspūdžio. Tačiau to vien tik nepakanka, kad pritraukti naujų vartotojų dėmesį, privalėtų būti ir šokia tokia intriga, kad vartotojas būtų susidomėjęs užsiregistruoti renginių organizavimo sistemoje.

Kadangi vartotojo sąsaja yra vienas iš svarbiausių programos komponentų, todėl labai svarbu, kad ji pasižymėtų šiomis savybėmis:

- Patogumu;



- Lengva navigacija;
- Paprastumu;
- Neapkrauta aplinka;
- Lengvu įsisavinimu.

#### 1.4.6 Egzistuojantys sprendimai

Pirmiausiai reikėtų atkreipti dėmesį kokie yra organizatoriaus lūkesčiai, jam organizuojant tam tikrus renginius.

- Ar jiems paprasčiausiai reikia greito, nebrangaus ir lengvo kelio surinkti mokėjimams
- Ar jie planuoja organizuoti sudėtingus ir didelius renginius ar konferencijas.
- Ar jiems reikalingas kitoks apmokėjimų būdas.
- Ar jų svarbiausias poreikis yra svečių sąrašas, saugomas ir analizuojamas sistemos duomenų bazėje.

Pasaulyje yra keletas panašios paskirties projektų, tačiau dauguma jų skirti profesionaliems vartotojams. Šios sistemos turi begalę funkcijų, tačiau jos yra sudėtingos naudotis. Tam kad pilnai išnaudot jų funkcionalumą reikia būti patyrusiu kompiuterio vartotoju. Be to norint išbandyti kai kurias sistemas reikia parašyti vadybininkui, kad jis duotą priėjimą prie demonstracinės versijos. Taigi norint tarkim paskelbti renginį, reikia laukti kelias dienas kol galėsit tai padaryti.

Mano nuomone geriausia šias problemas išsprėdė sekančios sistemos. Žinoma kiekviena iš jų turi savo privalumų bei trūkumų, tačiau būtent šias sistemas mes laikome savo tiesioginiais konkurentais.

##### 1.4.6.1 Eventbrite

Adresas: [www.eventbrite.com](http://www.eventbrite.com)

##### **Privalumai:**

Ši sistema leidžia labai greitai paskelbti renginį ir sukviesti į jį svečius. Pakankamai aiškus vartotojo interfeisas leis net ir nepatyrusiam vartotojui gana lengvai sukurti renginį. Sistema kaip manome sukurta panaudojus Python programavimo kalbą, dėl ko funkcinis sistemos

veikimas yra greitas. Leidžiama tinkinti registracijos į renginį puslapį, šios funkcijos dėka renginio organizuotojui suteikiama vietos pasireikšti fantazijai, kad prijauktų kuo daugiau lankytojų į savo renginį. Taip pat organizuotojui leidžiama nurodyti Paypal sąskaitą į kurią tiesiogiai būtų pervesti pinigai už įsigytus bilietus į renginį.

**Trūkumai:**

Kadangi sistema yra pakankamai neseniai sukurta tai vis dar pasitaiko keletas klaidelių, taip pat nėra galimybės tinkinti elektroninio pašto laišku, siunčiamų kviečiamiesiems asmenims. Kalbant apie naudojamumą, kadangi funkcijų šioje sistemoje yra tikrai daug, ne visada pavyksta pilnai jas visas išnaudoti, nes paprasčiausiai sunku surasti kur ji yra, tam reikia truputėlį daugiau laiko kad priprasti prie funkcijų nuorodų išdėliojimo.

### 1.4.6.2 Eventsbot

Adresas: [www.eventsbot.com](http://www.eventsbot.com)

**Privalumai:**

Visų pirma lyginant su eventsbrite.com sistema šios sistemos dizainas yra malonesnis.

Renginių kūrimas taip pat yra nesudėtingas, joje daugiau iššokinėjančių langų, kurie leidžia neperkraunant puslapio įvesti renginio informaciją. Taip pat patogiu kad tik įvedus renginio pavadinimą ir laiką iš karto esate nuvedami į puslapį, kuriame matote kaip atrodys renginio registracijos puslapis. Šiame puslapyje galite keisti visus renginio, bei registracijos puslapio parametrus.

**Trūkumai:**

Truputėlį skurdokas renginio statistikos peržiūros langas, matomai puslapio kūrėjai nespėjo tinkamai jo išdirbti, labiau koncentravosi į renginio kūrimą.

### 1.4.6.3 Trumba.com

Adresas: [www.trumba.com](http://www.trumba.com)

**Privalumai:**

Kompanijos veikimas pagrįstas interaktyvių komponentų diegimu skirtingose svetainėse, pvz. Kompanija savo svetainėje nori turėti kalendorių, kuriame būtų atvaizduojamas renginių sąrašas su aprašymais. Šią paslaugą būtent ir tiekia Trumba. Kalendoriaus komponentas yra valdomas ir prižiūrimas Trumba korporacijos.

**Trūkumai:**

Taip pat neatitinka pagrindinė idėja mūsų projektuojamos sistemos. Trumba siekia patogiai atvaizduoti didelį kiekį renginių viename kalendoriaus komponente, o mūsų sistemos pagrindinis tikslas, padėti renginių organizatoriams išsiuntinėti ir valdyti pakvietimus. Brangus sistemos diegimas, palaikymas ir priežiūra. Sudėtinga pradinė išvaizda.

**1.4.6.4 Kiko**

Adresas: [www.kiko.com](http://www.kiko.com)

**Privalumai:**

Sistema yra Google Calendar analogas. Pagrindinis tikslas, internetinis asmeninis renginių kalendorius, prie kurio galima prijungti ir kitus – viešus kalendorius. Patogus kalendoriaus valdymas, nes sistema veikia Ajax technologijos pagrindu. Priminimai apie renginius el. paštu ir SMS žinutėmis.

**Trūkumai:**

Šia sistema negalima sukurti pakvietimų. Negalima parinkti kvietimų dizaino, bei turinio. Pagrindinė paskirtis kalendorius, o ne renginio organizatoriaus padėjėjas.

Rinkoje egzistuoja keletas panašaus pobūdžio sistemų, tačiau jų visų pagrindinis tikslas nesutampa su mūsų sistemos tikslu. Didelės kompanijos, turi savo viduje panašaus tikslo sistemas, tačiau viešai jų neskelbia, ir naudoja savo poreikiams. Paanalizuokime tas sistemas kurios yra viešai prieinamos atskirai:

**1.4.6.5 Google Calendar**

Adresas: [www.google.com/calendar](http://www.google.com/calendar)

**Privalumai:**

Ši sistema yra sparčiai populiarėjanti, nors šiuo metu yra tik beta stadijoje. Apie Google produktus labai greitai sužino visas pasaulis, nes jų paieškos sistema naudojasi milijonai žmonių. Įdėję nors ir nedidelį užrašą paieškos sistemoje apie naują produktą, jis kaip mat tampa populiarus. Taip pat Google Calendar yra susietas su el. pašto sistema Gmail, kas dar labiau išpopuliarina šią sistemą.

Sistemoje yra galimybė kurti renginius, į juos pakviesti savo draugus iš kontaktų sąrašo. Automatiškai išsiuntinėjami el. laiškai kviečiamiems asmenims. Taip pat, jei gavėjas turi Google

sistemoje įsivedęs savo mobilaus telefono. nr. Tai jis gauna ir trumpąją žinutę nemokamai, kurioje pranešama apie renginį. Suderinama su iCal.

### **Trūkumai:**

Pagrindinė idėja skiriasi nuo mūsų projektuojamos sistemos. Tai yra, Google Calendar pritaikytas daugiau privačiam sektoriui, pavieniems asmenims, kurie neorganizuoja oficialių renginių, kuriems nėra svarbus įvaizdis. Negalima redaguoti el. laiško turinio. Jį automatiškai sugeneruoja Google sistema. Negalima pasirinkti laiško dizaino, jis visada bus vienodas. Negalima nurodyti kokias būdais bus nusiųstas pakvietimas. Negalima nurodyti gavėjų, visada bus siunčiama visiems pakviestiesiems. Kompliciuotas gavėjų pasirinkimo būdas, reikia mintinai žinoti kviečiamų žmonių vardus ar pavardes, nes pasirinkimas atsiranda tik pradėjus vesti pirmas kontakto vardo raides.

## 2 PROJEKGINĖ DALIS

### 2.1 Reikalavimų specifikavimas

#### 2.1.1 Įvadas

Pagrindinė problema, kuriai išspręsti yra skirtas ši sistema – tai renginių paskelbimas, pakvietimų išsiuntinėjimas renginyje dalyvaujantiems asmenims, bei jų dalyvavimo registravimas. Pakvietimų išsiuntinėjimas turi būti supaprastintas iki minimumo. Dalyvių reakcijos užfiksavimas neturėtų kelti jokių sunkumų, kvietimą gavusiam asmeniui, net jei jo kompiuterinis raštingumas yra žemas. Kvietimą gavęs asmuo turėtų turėti galimybę pranešti organizatoriui apie dalyvavimo statusą, net nesinaudodamas kompiuteriu. Sistema turi būti patogi tiek kviečiamam asmeniui (kuris yra perspektyvus sistemos naudotojas kaip organizatorius), tiek ir renginių organizatoriui. Tai turėtų pasireikšti patogia ir aiškia vartotojo sąsaja, neįkyriu dizainu, neperkrautu informacijos kiekiu.

Pagrindinis sistemos tikslas padėti renginių organizatoriams išspręsti organizavimo metu iškilusias problemas, sistemingai išanalizuoti praėjusį renginį, padėti padaryti išvadas. Suteikti patogų pakvietimų į renginius organizavimo ir valdymo metodą. Norint įsigilinti į organizavimo kylančias problemas, buvo bendraujama su renginių organizatoriais ir padarytos išvados. Renginių galima suskirstyti į etapus:

#### 1. Renginio planavimas

- laiko parinkimas;
- vietos parinkimas;
- renginį organizuojančių žmonių parinkimas;
- dienotvarkės sudarymas;
- į renginį kviečiamų asmenų sąrašo sudarymas;
- asmenų pakvietimų būdų į organizuojamą renginį parinkimas;
- pakvietimų išsiuntinėjimas
- asmenų sutikimo arba atsisakymo dalyvauti organizuojamame renginyje registravimas.

#### 2. Renginio pravedimas

- atėjusiųjų į renginį asmenų registravimas;
- vedamo renginio prezentavimas;
- renginio dienotvarkės pravedimas;
- renginio pabaiga.

### 3. Renginio įvertinimas

- renginio pravedimo kaštų analizė;
- renginyje dalyvavusiųjų asmenų reakcijos aptarimas;
- renginio nauda organizatoriams.

#### **2.1.2 Sistemos apribojimai**

Sistema veikia visuose kompiuteriuose, kurie turi priejimą prie interneto, nepriklausomai nuo kompiuteriuose esančios operacinės sistemos. Būtina sąlyga, kad kompiuteryje, turinčiame priejimą prie interneto, būtų įdiegta interneto naršyklė. Interneto naršyklės pagalba vartotojas turės galimybę pasiekti Renginių organizavimo sistemą. Rekomenduotina atviro kodo interneto naršyklė „Mozilla Firefox“, kuri yra greita ir patogi, tačiau sistema puikiai funkcionuos ir su kitomis interneto naršyklėmis: MS Internet Explorer, Opera, Netscape. Tam, kad sistema funkcionuotų reikia, kad naudojama naršyklė palaikytų XHTML 1.0 žymėjimo standartą, Javascript programavimo kalbą (be to naršyklėje turi būti nustatyta, kad būtų galima vykdyti Javascript komandų interpretavimą).

Prie sistemos galės prisijungti naudotojas, susikūres anketą sistemoje. naudotojas galės prisijungti tikrai turėdamas priejimą prie interneto. Prisijungimas bus patvirtinamas prisijungimo vardu ir slaptažodžiu, kuriuos žinos tik užsiregistravęs naudotojas. Naudotojų skaičius nebus ribojimas ir jų skaičius neįtakos sistemos funkcionavimo.

Renginių organizavimo sistema internete yra Internetinė programa skirta renginių paskelbimui ir organizavimui. Kontaktus galima įkelti iš failo tik pagal tam tikrus nustatytus standartus. Kontaktus išsisaugoti į failą galima pagal tam tikrus standartus. Pildymo formose nustatyti laukai, kai kurie privalomi, kiti pasirinktini.

Sistema užtikrins dalyvių pakvietimą nustatytu būdu ir laiku. Pakvietimo laikas priklausys nuo pasirinktos šalies laiko juostos nustatymų.

Sistema turės apsaugą nuo naudotojo klaidingai įvestų duomenų. Taip pat bus apribojimai tyčiniams klaidingų duomenų įvedimams. Naudotojui bus paaiškinta ir nurodyta kokio tipo duomenys yra priimtini.

Kad būtų galima teikti paslaugas, reikia turėti serverį arba užsisakyti tinklapių talpinimo paslaugas. Serveryje turėtų būti veikiantis Apache servisas, įdiegta MySQL duomenų bazės valdymo sistema ir taip pat įdiegtas PHP kalbos palaikymas, be kurių neįmanomas dinaminės sistemos veikimas, ši programinė įranga pasirinkta todėl, kad ji labai puikiai sąveikauja tarpusavyje, o tuo pačiu yra nemokama.

Sistema užtikrina naudotojų duomenų konfidencialumą apie jų turimus kontaktus ir organizuojamus renginius. Sistemos veikimas stebimas ir prižiūrimas, tai padės greitai reaguoti į iškilusias klaidas. Taip pat, php kalbos ir serverio apsaugų pagalba bus pasirūpinta ir apsauga nuo išorinių įsilaužėlių. Užtikrintas sistemos veikimas ir priėjimas prie jos 24 valandas per parą ir 7 dienas per savaitę.

### 2.1.3 Funkciniai reikalavimai

Reikalavimo Nr.:	FR-3	Reikalavimo tipas:	3	Įvykio/panaudos atvejo Nr.:	PA-1, VI-1
Aprašymas:	Registruoti sistemos naudotojai gali kurti renginius.				
Loginis paaiškinimas:	Registruotas naudotojas gali turėti neribotą kiekį renginių.				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

Lentelė Nr. 2 FR-3 Sukurti naują renginį

Reikalavimo Nr.:	FR-4	Reikalavimo tipas:	3	Įvykio/panaudos atvejo Nr.:	PA-14, VI-2
Aprašymas:	Leisti vartotojams ieškoti renginių.				
Loginis paaiškinimas:	Vartotojai gali ieškoti renginių pagal nurodytą datos intervalą, arba pagal įvestą tekstą				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

Lentelė Nr. 3 FR-4 Surasti renginį

Reikalavimo Nr.:	FR-5	Reikalavimo tipas:	3	Įvykio/panaudos atvejo Nr.:	PA-3, VI-1
Aprašymas:	Leisti vartotojams keisti renginio matomumą				
Loginis paaiškinimas:	Vartotojai gali keisti tik savo sukurtų renginių matomumą.				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

Lentelė Nr. 4 FR-5 Paskelbti renginį

Reikalavimo Nr.:	FR-6	Reikalavimo tipas:	3	Įvykio/panaudos atvejo Nr.:	PA-2, VI-1
Aprašymas:	Keisti renginio duomenis				
Loginis paaiškinimas:	Vartotojai turi turėti galimybę keisti savo sukurtų renginių duomenis				
Sukūrė:	Mantas Žvironas				

<i>Atitikimo kriterijus:</i>			
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>	
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>	
<i>Kiti info šaltiniai:</i>			
<i>Istorija:</i>			

#### Lentelė Nr. 5 FR-6 Tvarkyti renginį

<i>Reikalavimo Nr.:</i>	<b>FR-7</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-9, VI-10</b>
<i>Aprašymas:</i>	<b>Naujo kontakto sukūrimas</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai turi turėti galimybę sukurti neribotą kiekį kontaktų</b>				
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>				
<i>Atitikimo kriterijus:</i>					
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>			
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>			
<i>Kiti info šaltiniai:</i>					
<i>Istorija:</i>					

#### Lentelė Nr. 6 FR-7 Sukurti naują kontaktą

<i>Reikalavimo Nr.:</i>	<b>FR-8</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-10, VI-10</b>
<i>Aprašymas:</i>	<b>Kontaktų importavimo galimybė</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai gali pasirinkti koku būdu importuoti kontaktus</b>				
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>				
<i>Atitikimo kriterijus:</i>					
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>			
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>			
<i>Kiti info šaltiniai:</i>					
<i>Istorija:</i>					

#### Lentelė Nr. 7 FR-8 Importuoti kontaktus

<i>Reikalavimo Nr.:</i>	<b>FR-9</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-8, VI-10</b>
<i>Aprašymas:</i>	<b>Kontaktų eksportavimo galimybė</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai gali eksportuoti savo sukurtus kontaktus</b>				
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>				
<i>Atitikimo kriterijus:</i>					
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>			
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>			
<i>Kiti info šaltiniai:</i>					
<i>Istorija:</i>					

#### Lentelė Nr. 8 FR-9 Eksportuoti kontaktus

<i>Reikalavimo Nr.:</i>	<b>FR-10</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-11, VI-10</b>
<i>Aprašymas:</i>	<b>Kontaktų paieškos galimybė</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai gali ieškoti tarp savo sukurtų kontaktų pagal vardą, pavardę, email, telefoną</b>				
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>				
<i>Atitikimo kriterijus:</i>					
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>			
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>			
<i>Kiti info šaltiniai:</i>					
<i>Istorija:</i>					

#### Lentelė Nr. 9 FR-10 Surasti kontaktus

<i>Reikalavimo Nr.:</i>	<b>FR-11</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-11, VI-10</b>
<i>Aprašymas:</i>	<b>Kontaktų valdymo galimybė</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai gali keisti kontaktų duomenis, juos ištrinti, perkelti iš vienos grupės į kitą</b>				
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>				
<i>Atitikimo kriterijus:</i>					
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>			
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>			
<i>Kiti info šaltiniai:</i>					
<i>Istorija:</i>					

#### Lentelė Nr. 10 FR-11 Tvarkyti savo kontaktus

<i>Reikalavimo Nr.:</i>	<b>FR-12</b>	<i>Reikalavimo tipas:</i>	<b>3</b>	<i>Įvykio/panaudos atvejo Nr.:</i>	<b>PA-11, VI-10</b>
<i>Aprašymas:</i>	<b>Kontaktų grupės sukūrimo galimybė</b>				
<i>Loginis paaiškinimas:</i>	<b>Vartotojai gali turėti neribotą kiekį kontaktų grupių</b>				



Sukūrė:	Mantas Žvironas		
Atitikimo kriterijus:			
Tinkamumas klientui:		Netinkamumas klientui:	
Prioritetas:	1	Konfliktai:	
Kiti info šaltiniai:			
Istorija:			

#### Lentelė Nr. 11 FR-12 Sukurti naują kontaktų grupę

Reikalavimo Nr.:	FR-13	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	PA-11, VI-10
Aprašymas:	Kontaktų grupės paieškos galimybė				
Loginis paaiškinimas:	Vartotojai kontaktų grupės gali ieškoti pagal jos pavadinimą				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

#### Lentelė Nr. 12 FR-13 Surasti kontaktų grupę

Reikalavimo Nr.:	FR-14	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	PA-11, VI-10
Aprašymas:	Kontaktų grupių valdymo galimybė				
Loginis paaiškinimas:	Vartotojai gali keisti kontaktų grupių sudėtį, jas ištrinti, keisti jos pavadinimą				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

#### Lentelė Nr. 13 FR-14 Tvarkyti kontaktų grupes

Reikalavimo Nr.:	FR-15	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	PA-11, VI-10
Aprašymas:	Kontaktų grupės eksportavimo galimybė				
Loginis paaiškinimas:	Vartotojai gali išeportuoti visą kontaktų grupę į failą				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

#### Lentelė Nr. 14 FR-15 Eksportuoti kontaktų grupę

Reikalavimo Nr.:	FR-16	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	VI-13
Aprašymas:	Vartotojo duomenų keitimo galimybė				
Loginis paaiškinimas:	Vartotojai gali pilnai keisti savo duomenis t.y. vardą, pavardę, šalį, slaptažodį. Email adreso keisti negalima				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

#### Lentelė Nr. 15 FR-16 Tvarkyti savo duomenis

Reikalavimo Nr.:	FR-17	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	VI-6, PA-5
Aprašymas:	Sąskaitos balanso padidinimo galimybė				
Loginis paaiškinimas:	Vartotojai jiems patogiu apmokėjimo būdu gali lengvai papildyti savo sąskaitą.				
Sukūrė:	Mantas Žvironas				
Atitikimo kriterijus:					
Tinkamumas klientui:		Netinkamumas klientui:			
Prioritetas:	1	Konfliktai:			
Kiti info šaltiniai:					
Istorija:					

#### Lentelė Nr. 16 FR-17 Papildyti sąskaitą

Reikalavimo Nr.:	FR-18	Reikalavimo tipas:	3	Ivykio/panaudos atvejo Nr.:	VI-9, PA-12
Aprašymas:	Registravimosi į renginį galimybė				

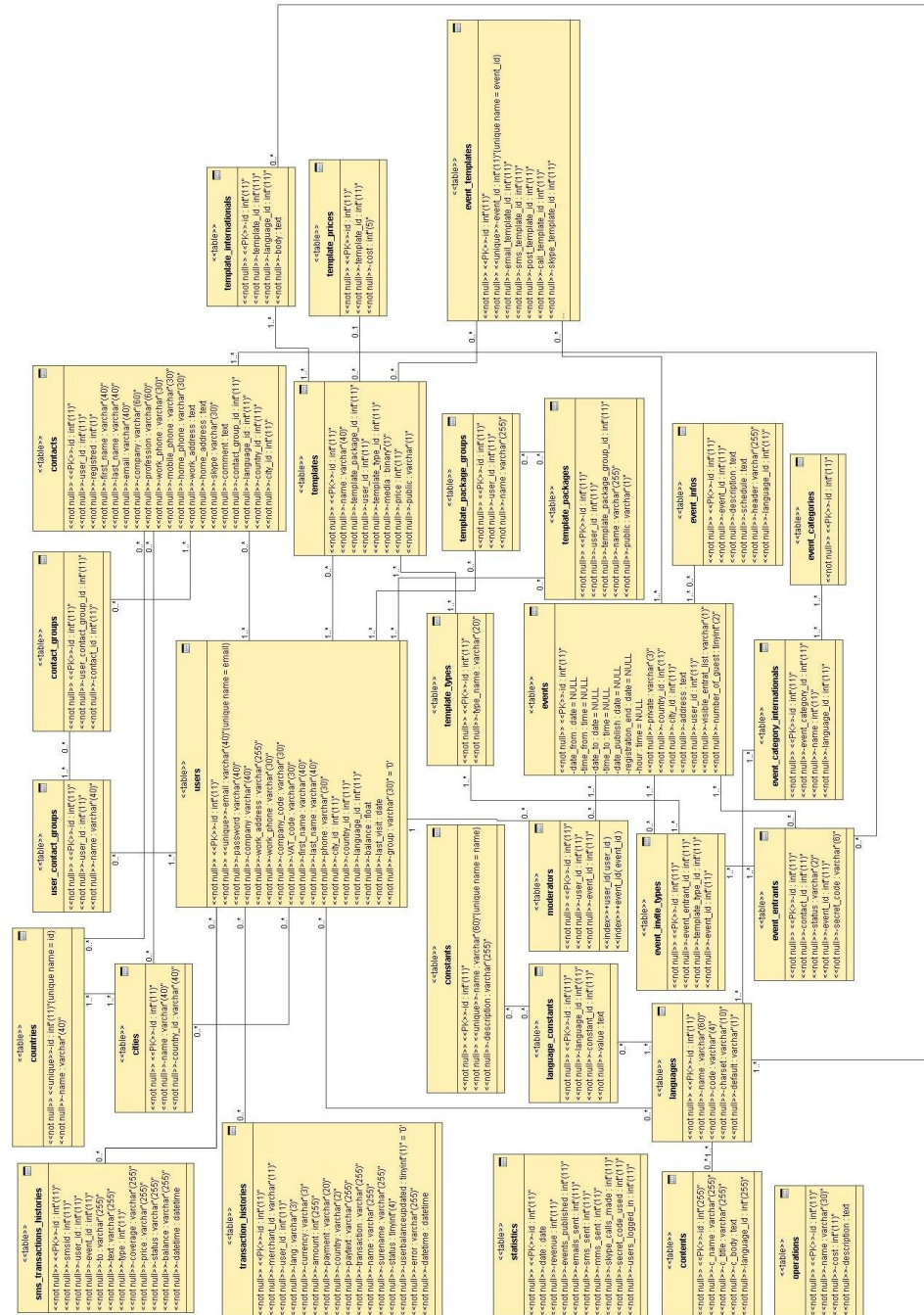
<i>Loginis paaiškinimas:</i>	<b>Bet kuris sistemos lankytojas, gavęs pakvietimo į renginį kodą, gali užsiregistruoti į renginį.</b>		
<i>Sukūrė:</i>	<b>Mantas Žvironas</b>		
<i>Atitikimo kriterijus:</i>			
<i>Tinkamumas klientui:</i>		<i>Netinkamumas klientui:</i>	
<i>Prioritetas:</i>	<b>1</b>	<i>Konfliktai:</i>	
<i>Kiti info šaltiniai:</i>			
<i>Istorija:</i>			

**Lentelė Nr. 17 FR-18 Registruotis į kviečiamą renginį**

### ***2.1.3.1.1 Duomenų reikalavimai***

Visi sistemos duomenys ir rezultatai yra saugomi MySQL duomenų bazės valdymo sistemoje (tik laikiniems duomenims saugoti yra naudojami atskiri failai). Dalis duomenų bazės lentelių yra sistemos lentelės, kitos lentelės yra sukurtos siekiant išgauti reikalingą funkcionalumą pagal užsakovo poreikius.

Duomenų bazės loginę schemą pateikiu Pav. 2 Duomenų bazės schema.



Pav. 2 Duomenų bazės schema

Toliau pateikiu esminių duomenų bazės lentelių aprašimus

### 2.1.3.1.1.1 Lentelė contact\_groups

Aprašymas: Skirta saugoti, kuriai vartotojo kontaktų grupei koks kontaktas priklauso.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK	Sveikasis skaičius 11	Kontaktų grupės numeris

				skaitmenų	
user_contact_group_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Vartotojo kontaktų grupės numeris
contact_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Kontakto numeris

**Lentelė Nr. 18 DB lentelės “contact\_groups” aprašymas**

### 2.1.3.1.1.2 Lentelė contacts

Aprašymas: Skirta saugoti informacijai apie kontaktą. Kiekvienas kontaktas turi savo unikalų numerį.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK	Sveikasis skaičius 11 skaitmenų	Kontakto numeris
user_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Vartotojo numeris
registered	int(1)	Taip		Sveikasis skaičius 1 skaitmens	Ar kontaktas registruotas. 1- taip, 0-ne
first_name	varchar(40)	Taip		Daugiausia 40 simbolių	Kontakto vardas
last_name	varchar(40)	Taip		Daugiausia 40 simbolių	Kontakto pavardė
email	varchar(40)	Taip		Daugiausia 40 simbolių	Email adresas
company	varchar(60)	Taip		Daugiausia 60 simbolių	Kompanijos pavadinimas
profession	varchar(60)	Taip		Daugiausia 60 simbolių	Profesija
work_phone	varchar(30)	Taip		Daugiausia 30 simbolių	Darbo telefonas
Mobile_phone	varchar(30)	Taip		Daugiausia 30 simbolių	Mobilaus telefono numeris
home_phone	varchar(30)	Taip		Daugiausia 30 simbolių	Namų telefono numeris
work_address	text	Taip			Darbo adresas
home_address	text	Taip			Namų adresas
skype	varchar(30)	Taip		Daugiausia 30 simbolių	Skype vardas
comment	text	Taip			Komentaras apie kontaktą
contact_group_id	int(11)	Taip		Sveikasis	Kontaktų grupės

				skaičius 11 skaitmenų	numeris
language_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Kalbos numeris
country_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Šalies numeris
city_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Miesto numeris

Lentelė Nr. 19 DB lentelės “contacts” aprašymas

#### 2.1.3.1.1.3 Lentelė event\_entrants

Aprašymas: Skirta saugoti kiekvieno renginio dalyvių sąrašui.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK		Renginio dalyvio numeris
contact_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Kontakto numeris
status	varchar(2)	Taip		Daugiausia 2 simbolių, galimos reikšmės: 0-neatsakė, 1-dalyvaus, 2-neapsisprendė, 3-nedalyvaus	Dalyvavimo požymis
event_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Renginio numeris
secret_code	varchar(6)	Taip	Unikalus	Daugiausia 6 simbolių	Dalyvio slaptažodis

Lentelė Nr. 20 DB lentelės “event\_entrants” aprašymas

#### 2.1.3.1.1.4 Lentelė events

Aprašymas: Skirta saugoti informacijai apie renginį.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK	Sveikasis skaičius 11 skaitmenų	Renginio numeris
date_from	date	Taip		Tik data	Renginio pradžios data

time_from	time	Taip		Tik laikas	Renginio pradžios laikas
date_to	date	Taip		Tik data	Renginio pabaigos data
time_to	time	Taip		Tik laikas	Renginio pabaigos laikas
date_publish	date	Taip		Tik data	Paskelbimo data
registration_end	date	Taip		Tik data	Registracijos pabaigos data
private	varchar(1)	Taip		Daugiausia 1 simbolių	Ar renginys privatus? 0-ne, 1-taip
country_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Šalies numeris
city_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Miesto numeris
address	text	Taip			Adresas
user_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Vartotojo numeris
visible_entrat_list	varchar(1)	Taip		Daugiausia 1 simbolis	Ar renginio dalyvių sąrašas matomas kitiems? 0-ne 1-taip

Lentelė Nr. 21 DB lentelės “events” aprašymas

#### 2.1.3.1.1.5 Lentelė user\_contact\_groups

Aprašymas: Skirta saugoti vartotojo kontaktų grupėms.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK	Sveikasis skaičius 11 skaitmenų	Vartotojo kontaktų grupės numeris
user_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Vartotojo numeris
name	varchar(40)	Taip		Daugiausia 40 simbolių	Grupės pavadinimas

Lentelė Nr. 22 DB lentelės “user\_contact\_groups” aprašymas

#### 2.1.3.1.1.6 Lentelė users

Aprašymas: Skirta saugoti informaciją apie vartotoją.

Lauko vardas	Lauko tipas	Ar būtinas	Raktas ir indeksas	Reikšmių apribojimai, trigeriai	Aprašymas
<i>id</i>	int(11)	Taip	PK	Sveikasis skaičius 11 skaitmenų	Vartotojo numeris
<b>email</b>	varchar(40)	Taip		Daugiausia 40	Vartotojo email

				simbolių	adresas
password	varchar(40)	Taip		Daugiausia 40 simbolių	Vartotojo slaptažodis
company	varchar(40)	Taip		Daugiausia 40 simbolių	Kompanijos pavadinimas
work_address	varchar(255)	Taip		Daugiausia 255 simbolių	Darbo adresas
work_phone	varchar(30)	Taip		Daugiausia 30 simbolių	Darbo telefonas
company_code	varchar(30)	Taip		Daugiausia 30 simbolių	Bendrovės kodas
VAT_code	varchar(30)	Taip		Daugiausia 30 simbolių	PVM kodas
first_name	varchar(40)	Taip		Daugiausia 40 simbolių	Vardas
last_name	varchar(40)	Taip		Daugiausia 40 simbolių	Pavardė
phone	varchar(30)	Taip		Daugiausia 30 simbolių	Telefonas
city_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Miesto numeris
country_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Šalies numeris
language_id	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Kalbos numeris
balance	int(11)	Taip		Sveikasis skaičius 11 skaitmenų	Balansas
last_visit	date	Taip		Tik data ir laikas	Paskutinio apsilankymo data
group	varchar(30)	Taip	0	Daugiausia 40 simbolių	Vartotojo grupė

Lentelė Nr. 23 DB lentelės “users” aprašymas

## 2.1.4 Nefunkciniai reikalavimai

### 2.1.4.1 Sistemos išvaizda

Produktas orientuotas į vyresniąją vartotojų kategoriją. Sistemos išvaizda yra solidi, patraukli akiai, tačiau ne per daug spalvota ir neapkrauta. Iš pirmo žvilgsnio sistemos išvaizda turi sudaryti pasitikėjimo ir ramumo jausmą. Išvaizdai keliamas pagrindinis reikalavimas, kad ji atrodytų paprasta ir elementari, lengva naudotis, tačiau tuo pačiu turi netrūkti komponentų, palaikančių

visišką sistemos funkcionalumą. Taip pat išvaizda turi sudaryti ir privatumo bei konfidencialumo jausmą, kad vartotojas, tik išvydęs dizainą, nesusidarytų pigaus ir nepatikimo puslapio įspūdžio. Tačiau to vien tik nepakanka, kad pavyktų pritraukti naujų vartotojų dėmesį, privalėtų būti ir šiokia tokia intriga, kad vartotojas būtų susidomėjęs užsiregistruoti renginių organizavimo sistemoje.

Kadangi vartotojo sąsaja yra vienas iš svarbiausių programos komponentų, todėl labai svarbu, kad ji pasižymėtų šiomis savybėmis:

- Patogumu;
- Lengva navigacija;
- Paprastumu;
- Neapkrauta aplinka;
- Lengvu įsisavinimu.

#### **2.1.4.2 Panaudojamumas ir prieinamumas**

Sistema turi būti pasiekiamą visiems interneto vartotojams (įskaitant ir neigaliuosius ir sveikuosius), nepaisant kokią naršymo technologiją jie naudoja. Visiems prieinamas tinklalapis gali turėti didelės naudos tiek pačiam tinklalapiui, tiek ir verslui.

Internetinė svetainė turi funkcionuoti su skirtingomis naršymo technologijomis. Pirma ir viena iš svarbiausių prieinamumo taisyklių yra, kad ne visi vartotojai naudoja naujausios versijos Internet Explorer naršyklę, su visais įmanomais atnaujinimais ir papildiniais. Kaip pavyzdžiui, galima pateikti keletą skirtingo tipo interneto naršymo technologijų:

- Lynx naršyklė – tai yra tik tekstinė naršyklė, kuri nepalaiko lentelių, css, paveiksliukų, JavaScript, Flash, audio ar video turinio.
- Naršymas per televizorių – 560 taškų pločio ir nėra horizontalios slintkies juostos.
- Ekranų skaitytojas – tai programa, balsu skaitanti puslapio turinį tokia tvarka, kaip jis atsiranda HTML dokumente
- Delninkas – labia mažas ekranas su ribotomis JavaScript ir didelių paveiksliukų galimybėmis.
- Ekranų didintuvas – vienu metu gali būti rodomi tik 3 – 4 žodžiai.
- Lėtas interneto ryšys – vartotojai gali išjungti paveiksliukus, Flash technologijas



- Plačiaekraniai vaizdukliai.

Visuomet turi būti alternatyvos paveiksliukams, JavaScript, Flash technologijoms, audio ir video turiniui. Taip pat atkreipti dėmesį ir į tai kaip svetainė atrodo be CSS. Taip pat ir užpildymo formos turi būti prieinamos visiems interneto vartotojams. Visiems vartotojams turėtų būti lengva ir greitai pasiekti turinį tinklalapyje. Patartina naudoti antraštes, nuorodas, paryškinius, numeravimus, taip pat konkrečius aprašymus, kaip pavyzdžiai: niekada nepavadinti nuorodos – „spausti čia“.

Patartina atskirti puslapio struktūrą nuo pateikimo. Struktūra nusako kaip dokumentas yra išdėstytas, dažniausiai naršymo meniu elementai, antraštės, pastraipos, sąrašai ir nuorodos. Puslapio pateikimas nusako kaip visa tai yra pateikta vartotojui. Pagrindinis principas yra naudoti CSS, o ne lentelių tipo išdėstymą. Taip pat patartina pasitelkti į pagalbą ir interaktyvumą, kaip pavyzdžiui: leisti vartotojams keisti teksto dydį, perspėti juos, kai nuoroda atsidarys kitame naršyklės lange ir pan.

Aukšto prieinamumo web paslaugos gali būti pasiektos, kuriant plataus naudojimo sistemas. Tačiau tokias sistemas sukurti yra brangu, ir tai gali įtakoti mažesnes kompanijas atsisakyti šių reikalavimų.

### **2.1.4.3 Našumas**

Web technologijų sparta yra matuojama pralaidumu, gaišties laiku, vykdymo laiku ir operacijos laiku. Pralaidumas yra skaičius, parodantis pateiktų užklausų skaičių per tam tikrą laiką. Gaištis yra laikas, per kurį yra nusiunčiama užklausa ir gaunamas atsakas. Vykdyto laikas yra laikas, skirtas sistemai atlikti visą jai skirtą veiklą seką. Operacijos laikas parodo Web sistemos užtrunkamą laiką, per kurį yra įvykdoma viena pilna operacija. Kuo didesnis pralaidumas, mažesnis gaištis, vykdymo laikas ir kuo greitesnis operacijos laikas, tuo web sistema greičiau veikia.

Visa Web sistemos sparta priklauso nuo programos logikos, tinklo ir labiausiai nuo esančių viduje žinučių siuntinėjimo ir duomenų perdavimo protokolų, tokių kaip SOAP ir HTTP.

Nepaisant SOAP, reikia atkreipti dėmesį ir į XML apdorojimo žingsnius. Keli iš būdų, kaip optimizuoti XML apdorojimo žingsnius:

- Naudoti veiksmingus ir lengvus analizatorius

- Efektyvus XML validavimo naudojimas kūrimo stadijoje
- Suspausto XML naudojimas, siunčiant žinutes per tinklą
- Web paslaugų padėjimas į podelį
- Kiek galima paprastesnių duomenų tipų naudojimas SOAP žinutėse.

#### **2.1.4.4 Veikimo reikalavimai**

Patikimumas yra vienas iš svarbiausių kriterijų, vertinant Web sistemų kokybę. Skaičius sutrikusių funkcionavimų per dieną, per savaitę ar per metus yra svarbiausias dydis, parodantis sistemos patikimumą. Šiuo metu Web technologijos pasikliauja tokiais duomenų perdavimo protokolais, kaip HTTP. Tačiau tai dar negarantuoja, kad žinutė bus pristatyta į vietą. Vienas iš būdų sumažinti riziką yra asinchroninės žinučių eilės, kitas būdas būtų paveldėjimas naujų patikimų duomenų perdavimo protokolų, tokių kaip HTTPR, REST, BEEP.

Patikimumas apibrėžia ar sistema bus prieinama tam tikru laiko momentu. Su patikimumu yra susijusi TTR (angl. Time-to-Repair) technologija. TTR atstovauja laiką, kuris bus sugaištas, tvarkant tam tikrą Web paslaugą.

Kurti web paslaugas, kurios net ir įvykus tam tikrai klaidai veiktų toliau, yra brangu. Kompanijos, pasirinkusios šį kelią, susiduria su skirtingose vietose išdalintų technologijų problemomis. Kyla tokie klausimai, kaip:

- Ar tik ne vienas iš pagrindinių serverių tapo nepasiekimas?
- Ar sistema netapo per daug apkrauta?
- Kodėl užklaustos trunka taip ilgai?

#### **2.1.4.5 Palaikomumas**

Kol kuri ir programuoji web sistemą, struktūra, kurią naudoji, puslapio išdėstymas ir viskas, kas susiję tarpusavyje tampa gerai žinoma ir pažįstama. Tačiau praeina tam tikras laiko tarpas, pridedi vienur šiek tiek, pridedi kitur, ir po kurio laiko sistema tampa didžiulė. Po kurio laiko atmintis išblėsta ir ilgainiui tampa nebeaišku, kodėl vienas dalykas yra čia, o ne ten, kam yra skirtas šitas principas. Dabar, jeigu norėsi ką nors pakeisti, yra didelė galimybė, kad keičiant vieną dalyką, gali nustoti funkcionavę kiti. Kad to pavyktų išvengti, reikia būti nuolatos pastoviam ir naudoti vieną pasirinktą metodologiją ir principą, nedaryti tų pačių dalykų skirtingais būdais. Verta naudoti komentarus. Yra blogai manoma, kad komentarai skirti tik

keistuoliams, kad jie galėtų suprasti ar net pasisavinti jūsų metodus. Nes labiausiai tikėtina, kad tas keistuolis jūs pats ir būsite, kuris mokinsis iš komentarų.

Patartina naudoti pastovią tiek pačio tinklapio, tiek pačios sistemos projektavimo struktūrą. Vienas iš būdų įvesti sistemingą struktūrą yra MVC (modelis-vaizdas-valdiklis) naudojimas. Tai tam tikra loginė bei praktinė metodologija, atskirianti veiksmus su duomenų bazėmis, išskirianti sistemos objektų klases, skirtas kontroliuoti sistemos vaizdą ir duomenų srautus.

#### **2.1.4.6 Saugumas**

Didėjant web technologijų paklausai, kurios yra prieinamos per viešą internetą, auga susirūpinimas dėl saugumo. Web kūrėjai vis dažniau susiduria su nuolatos didėjančiu spragų kiekiu, kurios tik dar labiau apsunkina darbą. Web paslaugų saugumas reiškia konfidencialių duomenų perdavimą, tik įgaliojtiems asmenims, žinučių šifravimą, ir priėjimo kontrolės tiekimą. Web sistemoje gali būti skirtingų lygių konfidenciali informacija, kurią yra būtina apsaugoti. Su saugumu susijusios problemos negali būti paliktos nuošalyje, sistemos saugumas privalo užtikrinti vartotojų pasitikėjimą.

Vienas iš didžiausių rizikos šaltinių organizacijoms yra Web serveriai. Įsilaužimas į serverius gali padaryti labai didelių problemų organizacijoms. Šias pažeidžiamas spragas galima suskirstyti į keletą kategorijų:

**Standartinė konfigūracija.** Web serveriai yra dažnai įdiegiami tiesiog su standartine konfigūracija, kuri gali būti nesaugi. Pažeidžiamos vietos gali būti standartiniai pavyzdžiai, šablonai, administravimo įrankiai, nuspėjamos paslaugų programų vietos. Be atitinkamo rizikos apsisaugojimo tai gali nuvesti į keletą skirtingo tipo įsilaužimų, po kurių piktaivaliai vartotojai gali perimti pilną Web serverio valdymą

**Vartotojo įvedamų duomenų tikrinimas.** Internetinės svetainės turi būti interaktyvios, kad būtų naudingos vartotojui. Neapsisaugojus nuo vartotojo įvedamų duomenų, gali kilti daugybė saugumo problemų, kuriomis gali pasinaudoti piktaivalis vartotojas ir įsilaužti į Web serverį bei prieiti prie duomenų, esančių duomenų bazėse.

**Duomenų užšifravimas.** Kad ir kaip bebūtų gaila, nors ir dabartiniai šifravimo algoritmai virtualiai yra nenulaužiami, tačiau jie nėra pakankamai naudojami. Taip yra dėl didelio kainos ir kokybės skirtumo tarp procesorių ir specializuotų šifravimo greitintuvų. Beveik visas šiuo metu tinklo srautas keliauja nešifruotas ir gali būti lengvai perimtas piktaivalių vartotojų

**Duomenų laikymo apsaugojimas.** Dauguma konfidencialių failų yra saugomi viešai prieinamuose serveriuose, vietoj to, kad būtų talpinami specialiai apsaugotuose serveriuose. Kai kurios programos skiria per mažai dėmesio laikinų failų išvalymui, palikdamos svarbią informaciją lengvai prieinamą.

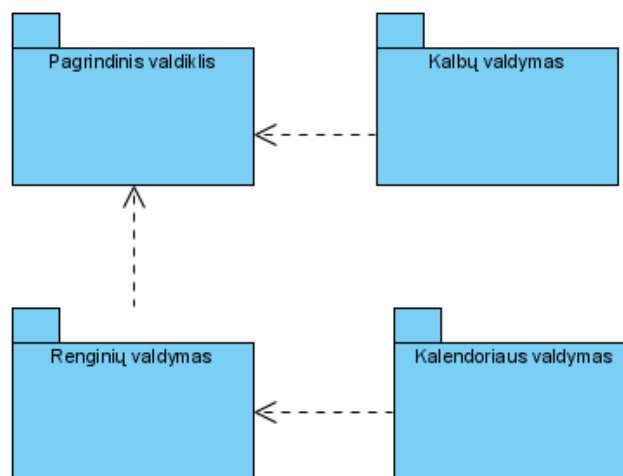
**Sesijos valdymas.** Dauguma svetainių per mažai dėmesio skiria unikalių sesijų valdymui. Tai gali būti ir naudojami silpni autentifikavimo metodai, prastas sistemos sausainiukų (angl. cookies) valdymas, nesugebėjimas nustatyti sesijos neveiksnumo laiko ir panašios sesijų silpnybės. Tai dažnai veda į sesijų užgrobimus, kur užgrobėjas gali perimti visus vartotojo duomenis.

## 2.2 Sistemos projektavimas

### 2.2.1 Sistemos objektinis modelis

#### 2.2.1.1 Apžvalga

Pateikiamas sistemos išskaidymas į paketus. Detalesnis paketų paskirties paaiškinimas pateikiamas kitame skyrelyje.

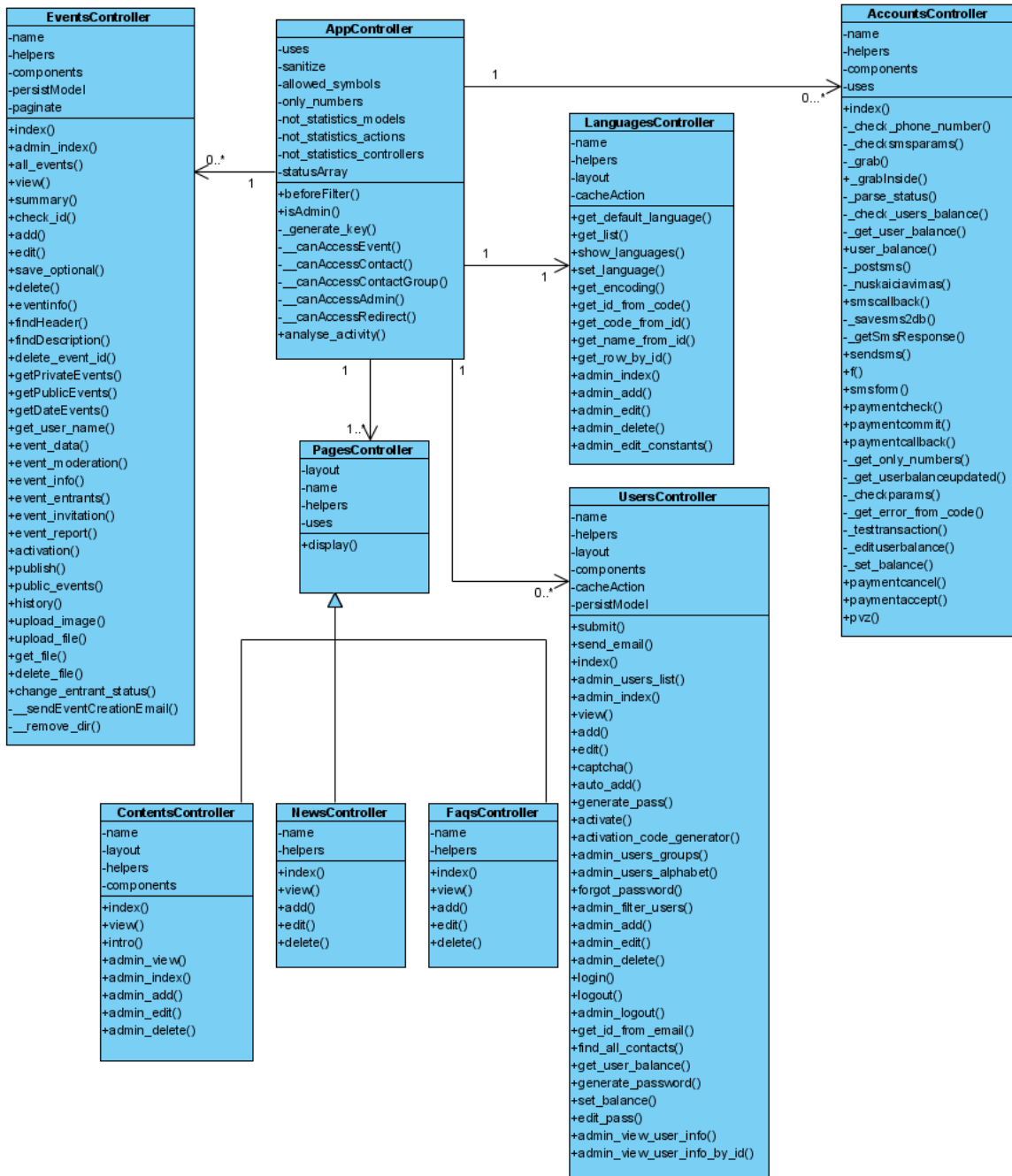


Pav. 3 Klasių išskaidymas į paketus

#### 2.2.1.2 Paketų detalizavimas

##### 2.2.1.2.1 Pagrindinio valdiklio klasių paketas

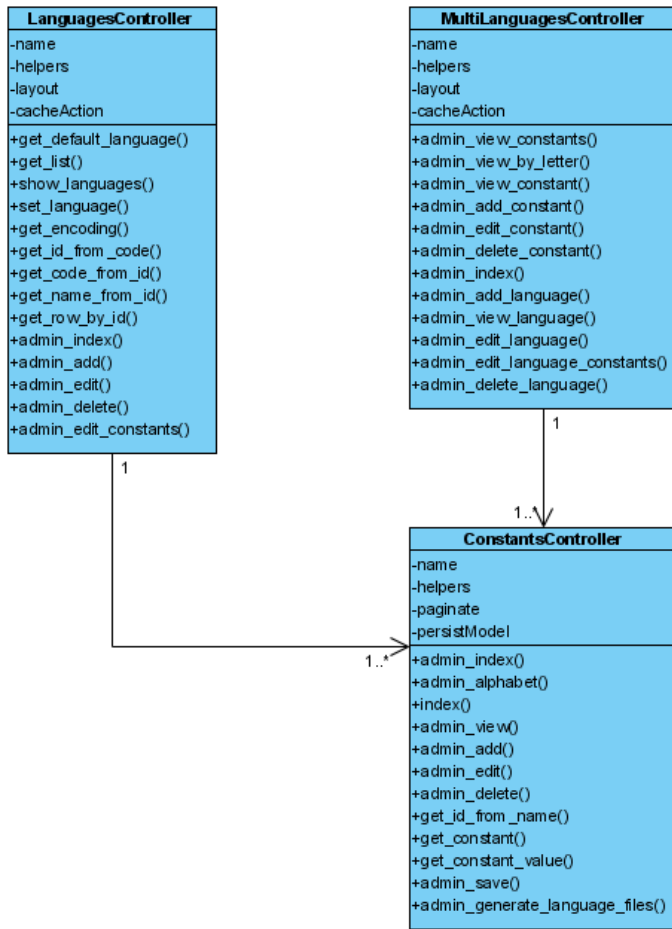
Šis valdiklis valdo visus kitus klasių objektus, nuo jo priklauso sistemos veikiamumas, funkcionalumas, priėjimas. Šiam paketui taip pat priklauso ir vartotojų klasė, vartotojų sąskaitų klasė ir pagalbinių puslapių klasės. Taip pat šiame pakete pavaizduoti pagrindinio valdiklio sąryšiai su renginių ir kalbų klasėmis.



Pav. 4 Pagrindinio valdiklio klasių paketas

### 2.2.1.2.2 Kalbų valdiklio klasių paketas

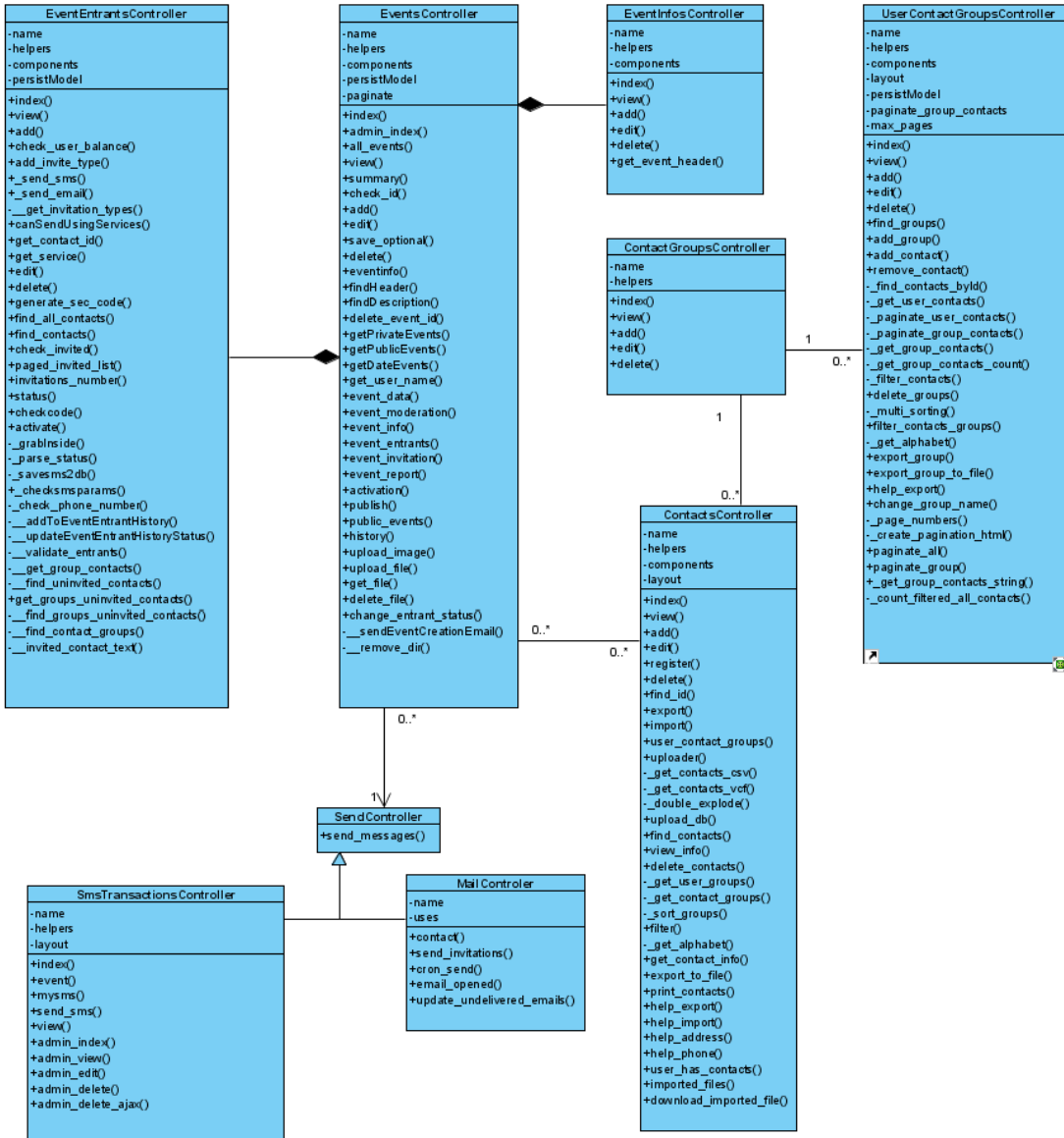
Šiame pakete yra sistemos kalbų valdiklio klasė, klasė atsakinga už daugiakalbiškumo tvarkymą ir taip pat yra kalbų konstantų klasė, sąryšis su pagrindiniu valdikliu nurodytas 2.2.1.2.1 skyrelyje



Pav. 5 Kalbų valdiklio klasių paketas

### 2.2.1.2.3 Renginių valdiklio klasių paketas

Šiame pakete yra pateiktos išsamios klasės, susijusios su renginių tvarkymu ir valdymu. Į šį klasių paketą taip pat yra prijungtos ir kontaktų, kontaktų grupių bei pranešimų siuntinėjimo klasės. Sąryšis su pagrindiniu valdikliu nurodytas 2.2.1.2.1 poskyryje.

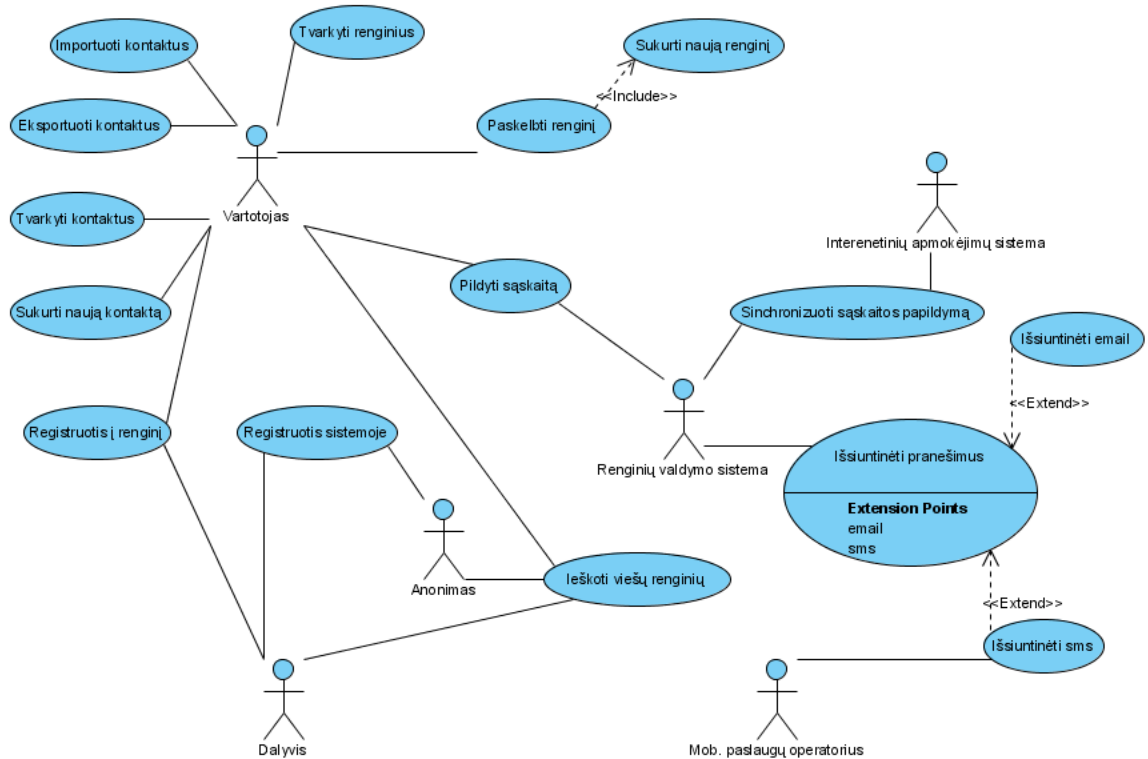


Pav. 6 Renginių valdiklio klasių paketas

## 2.2.2 Sistemos dinaminis modelis

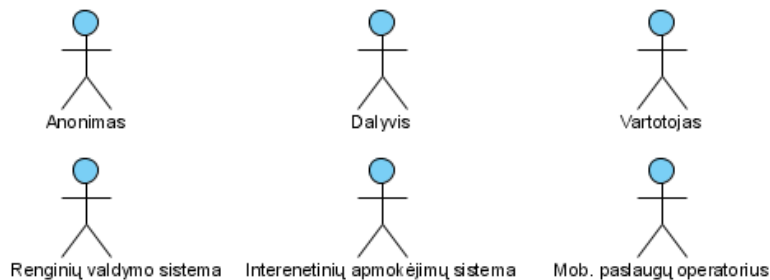
### 2.2.2.1 Panaudojimo atvejų vaizdas

Pateikiami esminiai panaudojimo atvejai:



Pav. 7 panaudojimo atvejų diagrama

Aktorių funkcijos (aprašomas kiekvienas aktorius).



Pav. 8 Sistemos aktoriai

**Vartotojas** – tai registruotas sistemos vartotojas. Jis gali naudotis visomis sistemos funkcijomis, kurios skirtos sistemos vartotojams.

**Anonimas** – tai yra tinklalapio lankytojas, kuris nėra nei registruotas nei pakviestas į renginį. Jis gali tik peržiūrėti viešus renginius bei užsiregistruoti sistemoje.



**Dalyvis** – tas pats kas anonimas, tačiau jis turi slapta kodą su kuriuo gali užsiregistruoti į renginį nesiregistruodamas sistemoje.

**Renginių valdymo sistema** – tai sistema, kuri atlieka pagrindinį vaidmenį, tai yra leidžia vartotojams valdyti savo sukurtus renginius ir kontaktus

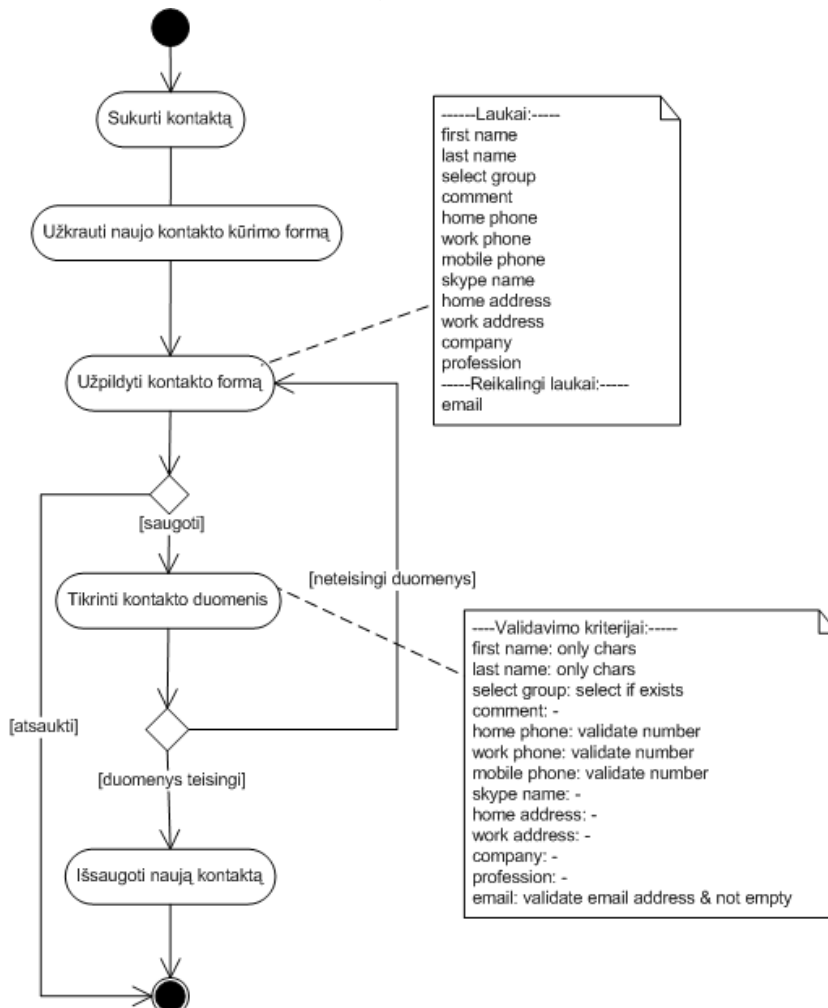
**Internetinių apmokėjimų sistema** – jos pagalba renginiu valdymo sistemos vartotojai gali pasipildyti savo sąskaitą, pasinaudodami vienu iš kelių būdų, tam kad galėtų skelbti renginius bei išsiuntinėti pakvietimus sms žinutėmis.

**Mob. Paslaugų operatorius** – jo pagalba renginių valdymo sistemos vartotojai gali išsiųsti pakvietimą į renginį sms žinute.

## 2.2.2.2 Veiklos diagramos

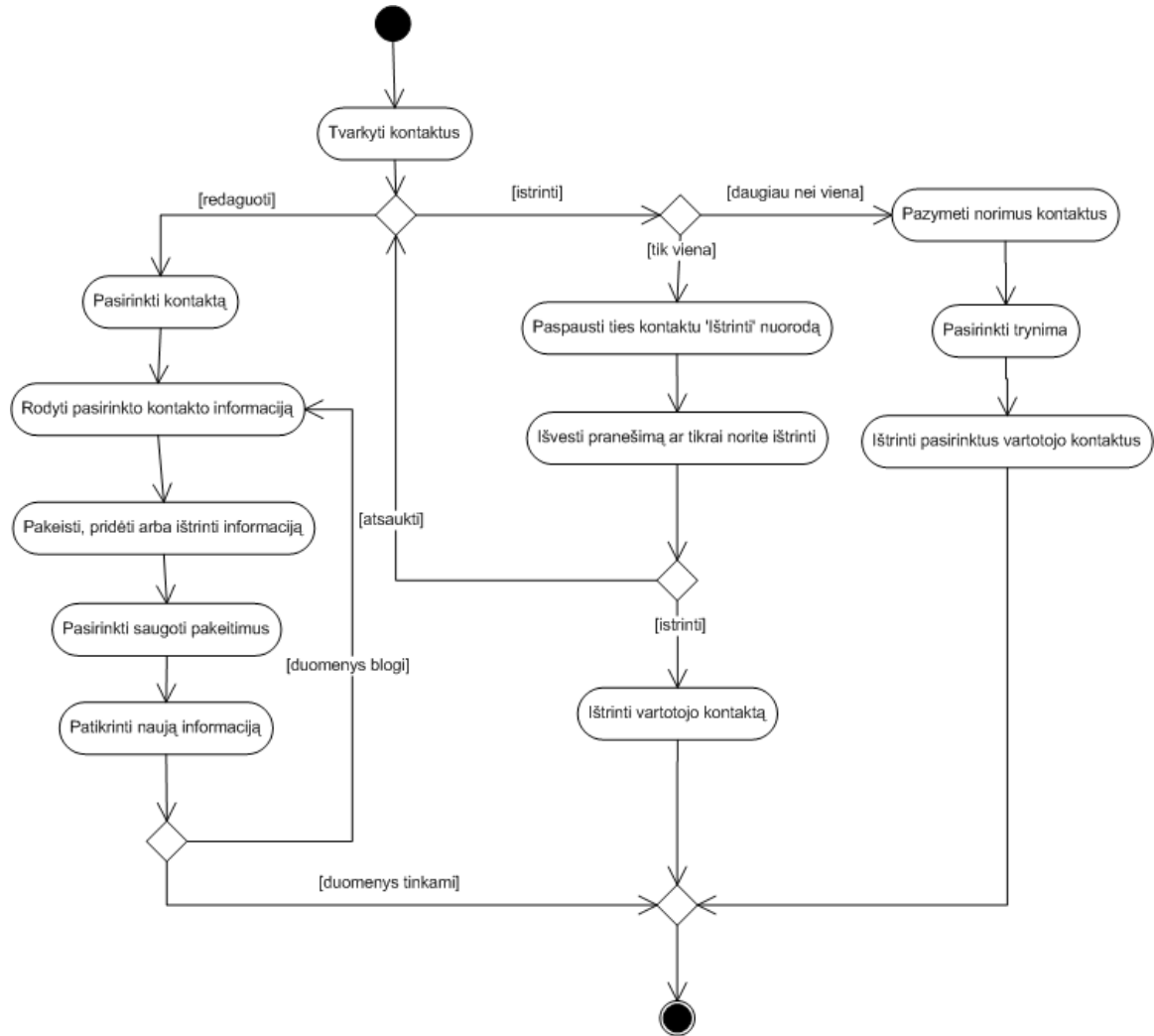
Šiame skyriuje pateikiu pagrindines sistemos veiklos diagramas

### 2.2.2.2.1 Sukurti kontaktą



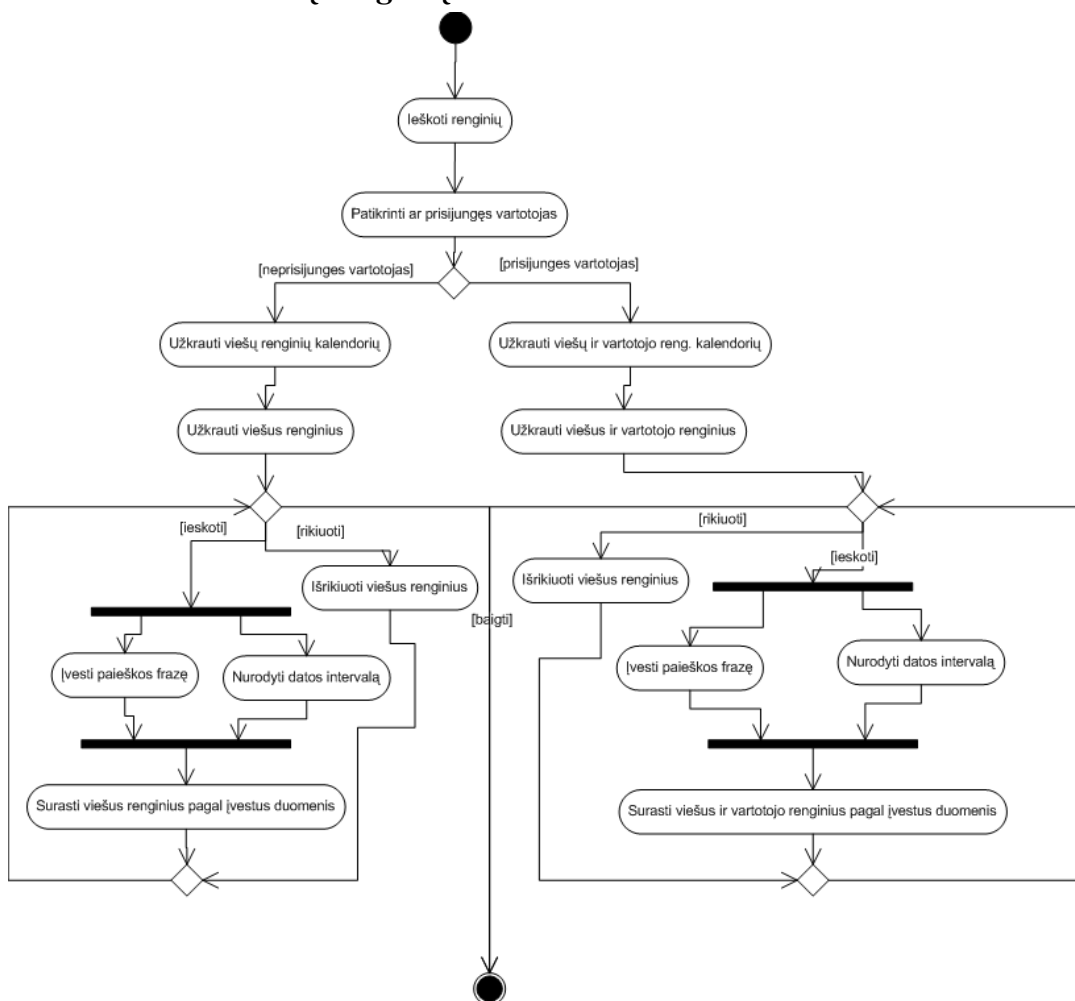
Pav. 9 Sukurti kontaktą veiklos diagrama

### 2.2.2.2.2 Tvarkyti kontaktus



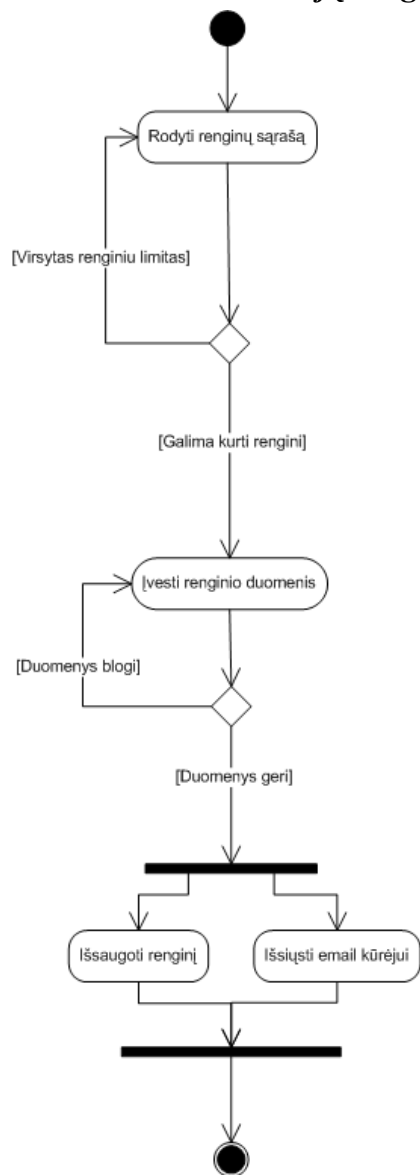
Pav. 10 Tvarkyti kontaktus veiklos diagrama

### 2.2.2.2.3 Ieškoti viešų renginių



Pav. 11 Ieškoti viešų renginių veiklos diagrama

#### 2.2.2.2.4 Sukurti naują renginį



Pav. 12 Sukurti naują renginį veiklos diagrama



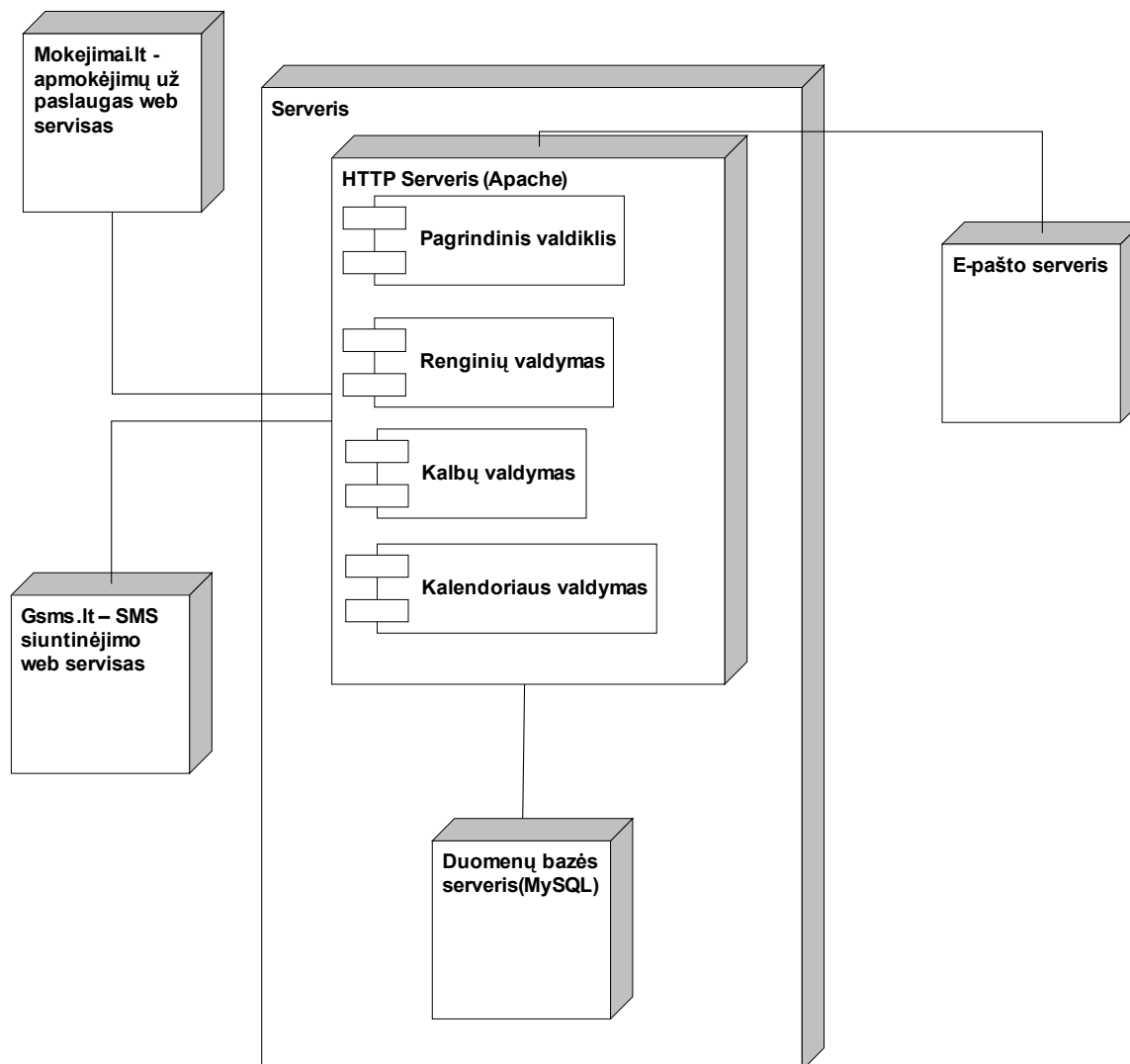
### 2.2.3 Išdėstymo vaizdas

Kuriama sistema bus įgyvendinta kaip tinklo programa (WEB application). Ją galima bus pasiekti iš bet kurio kompiuterio, kuris yra prijungtas prie interneto ryšio, adresu [www.invup.lt](http://www.invup.lt). Visa sistemos naudojama informacija bus saugojama duomenų bazėje. Sistema naudoja savo vartotojų autentifikaciją. Visi veiksmai kuriamoje sistemoje atliekami naudojant tą pačią vartotojo sąsają, skiriasi tik sąsajos panaudojimo galimybės, priklausomai nuo vartotojo teisių sistemoje. Sistema yra pritaikyta konkrečiai problemai spręsti ir pakartotinis jos panaudojimas yra įmanomas tik tam tikriems sistemos komponentams. Kuriama sistema yra priklausoma nuo kitų sistemų:

[www.mokejimai.lt](http://www.mokejimai.lt) – sistema per kuria atliekami piniginiai pervedimai.

[www.gsms.lt](http://www.gsms.lt) – sistema per kurią išsiuntinėjamos trumposios sms žinutės.

E-pašto serveris – skirtas el. Pašto laiškų išsiuntinėjimui



Pav. 14 Sistemos išdėstymo vaizdas

## 3 Eksperimentinė dalis

### 3.1 Sistemos probleminė sritis

Naudojantis sistema buvo pastebėta, kad dalyvių valdymo lange, pateiktame Pav. 15, iškyla problemų kuomet kontaktų ir dalyvių skaičius pasiekia apie du šimtus. Taip pat šio lango atliekamos funkcijos tenkino ne visus reikalavimų specifikacijoje išvardintus reikalavimus.

Esant nedideliam kontaktų bei dalyvių skaičiui, šis langas yra pakankamai suprantamas ir funkcijos atliekamos greitai. Tačiau kuomet dalyvių ar kontaktų skaičius išaugdavo iki dviejų šimtų buvo jaučiamas:

- ryškus sistemos atsako laiko pailgėjimas, kviečiant penkiasdešimt ir daugiau kontaktų vienu metu, arba šalinant penkiasdešimt ir daugiau dalyvių vienu metu
- Pailgėjęs dalyvių valdymo lango užkrovimas

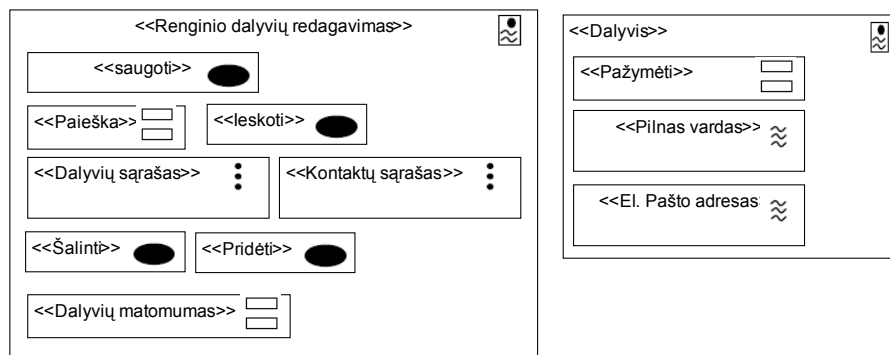
Taip pat šio lango funkcionalumas neatitiko reikalavimų specifikacijoje išvardintų naujų reikalavimų:

- Galimybė sukurti naują kontaktą, nepaliekant šio lango, bei iš karto naująjį kontaktą pridėti prie pakviestųjų sąrašo
- Galimybė importuoti kontaktus iš šio lango, bei sugrįžti atgal, po sėkmingo kontaktų importavimo.
- Galimybė pažymėti visus pakviestuosius asmenis šalinimui iš renginio.
- Pakviestųjų bei kontaktų sąrašus pateikti taip, kad juose kontaktai būtų suskirstyti į grupes
- Galimybė išskleisti bei sutraukti grupes, norint pamatyti jose esančius kontaktus.
- Puslapiuoti kontaktų sąrašą grupėje, jei kontaktų yra daugiau nei dvidešimt penki.
- Galimybė žymėti kontaktus, esančius viename kontaktų grupės puslapyje
- Vaikstant per kontaktų grupės sąrašo puslapius, atsiminti kokie kontaktai buvo pažymėti, bei automatiškai pažymėti kontaktus jei jie buvo pažymėti.

- Žymint kontaktus iš karto parodyti kiek dar galima pakviesti kontaktų bei kiek jau bus pakviesta kuomet pažymėti kontaktai ar dalyviai bus atitinkamai pakviesti į renginį ar pašalinti iš renginio.
- Galimybė pažymėti visą kontaktų grupę perkėlimui.

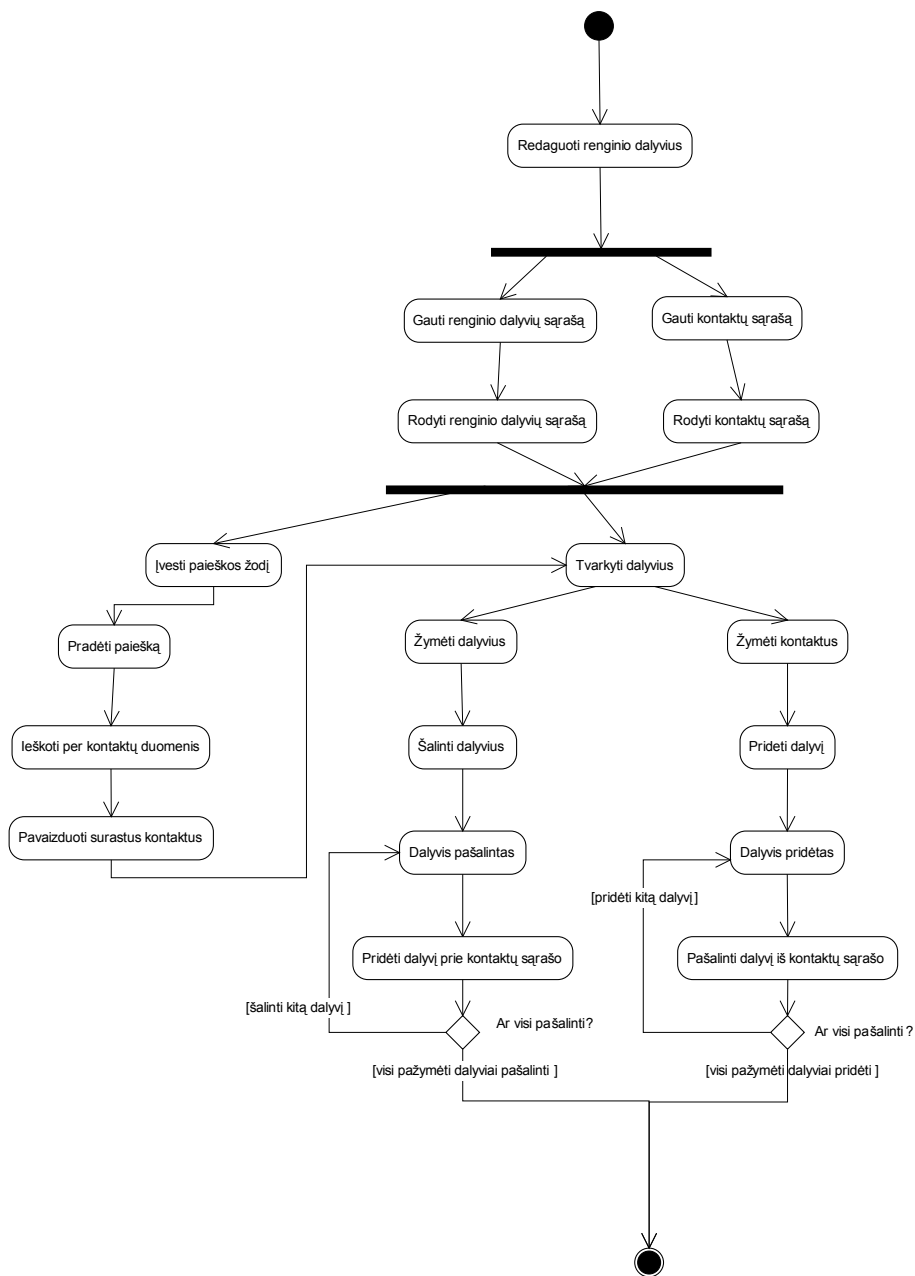
### 3.1.1 Pradinis konceptualinis modelis

Siekiant geriau suprasti renginio dalyvių valdymo lango veikimą buvo padarytas pradinis konceptualus modelis, pateiktas Pav. 16 ir Pav. 15. Realus pradinis šio lango vaizdas yra pateiktas prieduose esančiame paveiksluke Pav. 23. Šis modelis yra realizuotas kaip veiklos diagrama ir pateikimo modelio objektas, kuris buvo aptartas skyrelyje 1.1.1. Modelio tikslas yra atvaizduoti renginio dalyvių valdymo funkcionalumą.



Pav. 15 Pradinio konceptualinio modelio dalyvių valdymo lango pateikimo objektas





Pav. 16 Pradinio konceptualinio modelio veiklos diagrama

### 3.1.1.1 Aprašymas

Pirmiausia sistemos naudotojas būdamas renginio redagavime, pasirenka redaguoti renginio dalyvius. Tuomet sistema lygiagrečiai pradeda krauti redaguojamo renginio dalyvių sąrašą bei prisijungusio naudotojo kontaktų sąrašą. Kontaktų sąraše rodomi tik tie kontaktai, kurie dar nėra pakviesti į renginį. Kuomet baigiami krauti abu šie sąrašai, jie yra parodomi naršyklės lange ir naudotojas gali tęsti darbą.

Jis gali pasirinkti atlikti paiešką tarp kontaktų. Tam jis turi įvesti paieškos žodį į teksto laukelį, bei paspausti „Enter“. Jei bus surastas nors vienas kontaktas atitinkantis paieškos kriterijų, naudotojas galės surastus(-ą) kontaktus(-ą) pažymėti ir padaryti jį renginio dalyviu.

Renginio dalyvius ir kontaktus naudotojas gali žymėti ir iš karto po to, kai užkraunamas langas. Pažymėjęs norimus kontaktus ar dalyvius, naudotojas paspaudžia „Šalinti“ arba „Pridėti“, priklausomai nuo to ar jis padaro kontaktus dalyviais, ar šalina kontaktus iš renginio. Jei naudotojas daro dalyvių pašalinimą iš renginio, tuomet kai jis paspaudžia mygtuką „Šalinti“, sistema pradeda kiekvieno dalyvio atskirai šalinimo procesą, ir pašalinus dalyvį jis atsiranda kontaktų sąrašė. Panašūs veiksmai vyksta ir su kontaktų pridėjimu prie dalyvių sąrašo. Naudotojui paspaudus mygtuką „Pridėti“, kiekvienas pažymėtas kontaktas yra perkeliamas į dalyvių sąrašą. Sistemai baigus vykdyti veiksmus naudotojas gali tęsti darbą su sistema.

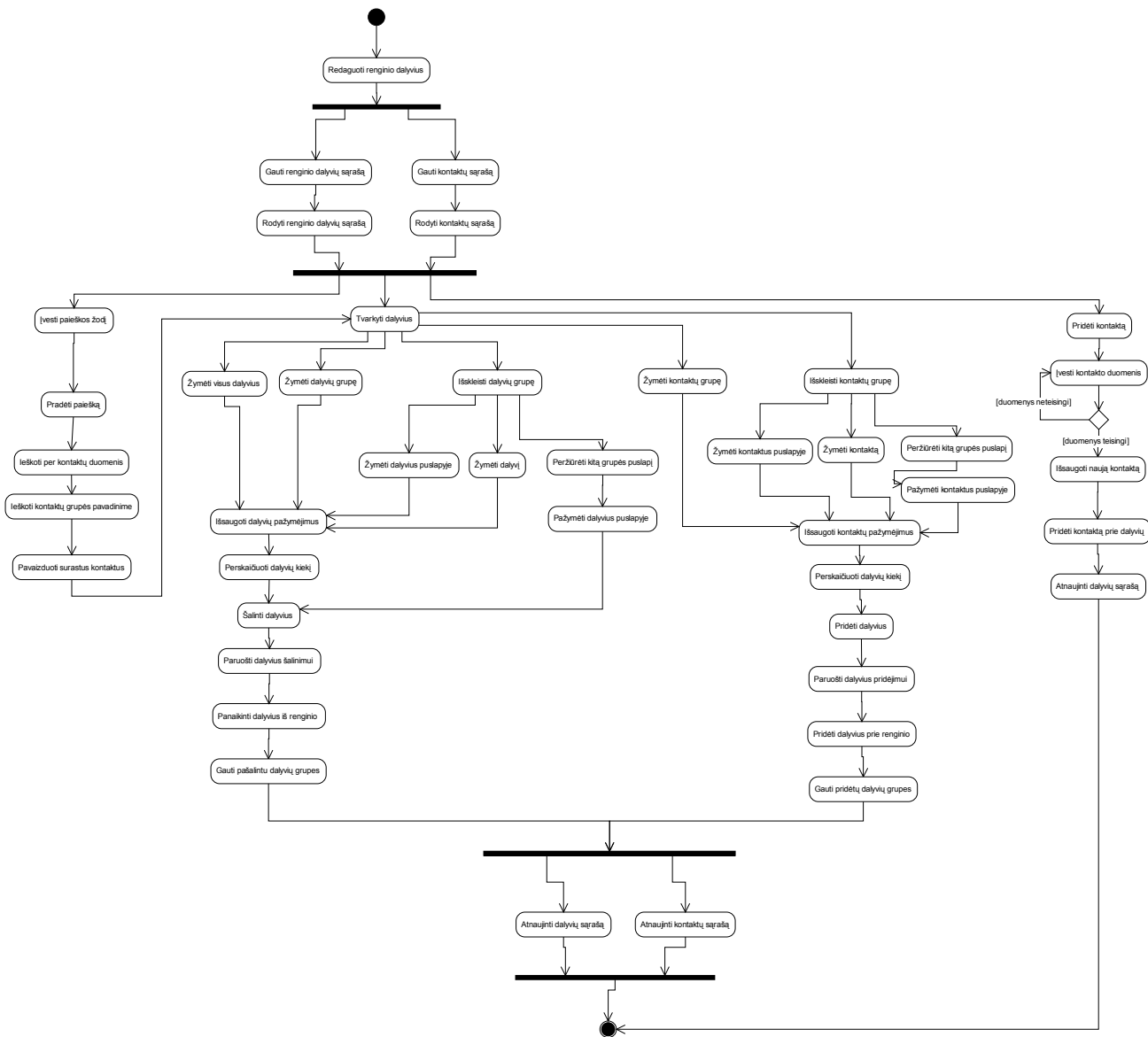
Dalyvių lango pateikimo objektas Pav. 15, parodo schematinį vartotojo sąsajos elementų išdėstymą. Pirmiausia pateikiamas mygtukas „Saugoti“, skirtas išsaugoti renginio dalyvių matomumo nustatymams. Žemiau yra du elementai skirti atlikti kontaktų paiešką pagal įvestą paieškos žodį. „Dalyvių sąrašas“ ir „Kontaktų sąrašas“ yra objektų rinkiniai sudaryti iš „Dalyvis“ tipo objektų. Objektas „Dalyvis“, savo ruožtu yra sudarytas iš formos elemento, skirto pažymėti dalyvį, ir dviejų teksto elementų kurie parodo informaciją apie šį dalyvį: pilna vardą ir el. pašto adresą. Nors šie sąrašai iš pirmo žvilgsnio atrodo identišškai, tačiau funkcinio požiūriu, dalyvių sąrašė yra rodomi kontaktai, kurie yra pakviesti į renginį. O kontaktų sąrašė rodomi kontaktai, kurie nėra pakviesti į šį renginį.

Toliau einantys du mygtukai „Šalinti“ ir „Pridėti“ yra skirti perkelti pažymėtus kontaktus ar dalyvius atitinkamai iš kontaktų sąrašo į dalyvių sąrašą. Ir paskutinis elementas yra dalyvių matomumo nustatymo forma, kurios pagalba naudotojas gali nustatyti ar dalyvių sąrašas bus matomas ar ne.

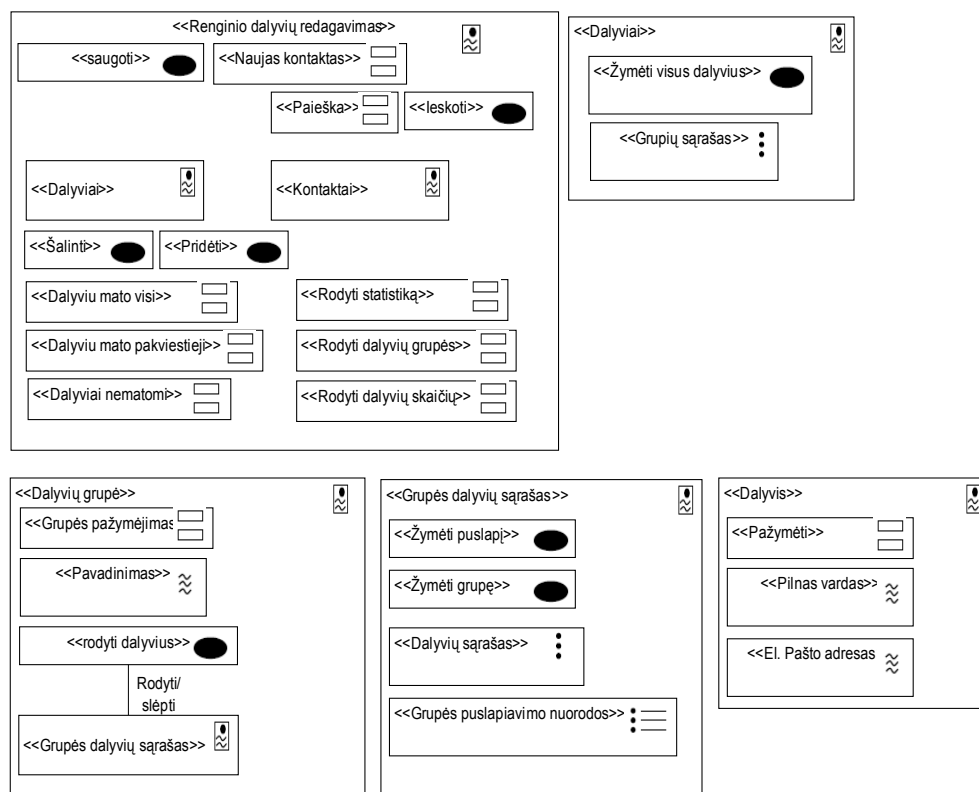
### **3.2 Konceptualinio modelio transformacija**

Skyrelyje 3.1.1 buvo pateiktas pirminis konceptualinis modelis, kurio tikslas buvo aprašyti pradinę sistemos probleminę sritį, t.y. renginio dalyvių valdymo langą. Šiame skyriuje pateikiu paskirties konceptualinį modelį. Kaip ir pradinis konceptualinis modelis jis yra pateiktas veiklos diagrama Pav. 17 ir pateikimo modelio objektas . Kaip galima pastebėti pradiniam konceptualiniam modeliui buvo atlikta homogeninė horizontalioji transformacija, t.y. abstrakcijos lygmuo ir modelio pavaizdavimo kalba išliko tie patys, tačiau pats modelis tapo labiau

detalizuotas, siekiant atvaizduoti atsiradusius naujus funkcinis bei nefunkcinis renginio dalyvių valdymo probleminės srities reikalavimus. Taip pat verta paminėti, kad transformacija buvo atlikta rankiniu būdu, kadangi transformacijos pagrindas buvo nauji funkciniai bei nefunkciniai reikalavimai, todėl siekiant kuo tiksliau juos pavaizduoti buvo pasinaudota esama projektavimo patirtimi. Realus galutinis renginio dalyvių valdymo lango vaizdas yra pateiktas prieduose esančiame paveiksliuke Pav. 24.



Pav. 17 Paskirties konceptualinio modelio veiklos diagrama



Pav. 18 Paskirties konceptualinio modelio dalyvių valdymo lango pateikimo objektas

### 3.2.1 Aprašymas

Pirmiausia apžvelgsiu konceptualinių modelių veiklos diagramų skirtumus. Kaip galite pastebėti pirmosios penkios veiklos tiek pradiniam tiek paskirties modeliuose yra tos pačios. Iš esmės šita dalis yra atsakinga už patį pradinį renginio dalyvio lango duomenų užkrovimą, ir šios dalies veiklos prasme nebuvo galima labiau optimizuoti. Einant toliau, pirmieji pakeitimai, kuriuos galima pastebėti yra šakoje kurioje atlieka paieška pagal įvestą paieškos žodį, šioje dalyje atsirado viena nauja t.y. „Ieškoti per kontaktų grupės pavadinimą“. Šios veiklos metu yra ieškoma įvesto žodžio kontaktų grupės pavadinime. Didžiausi pokyčiai įvyko dalyvių tvarkymo šakoje. Atsirado nemažai naujų veiklų. „Žymėti visus dalyvius“, kuri skirta iš karto pažymėti visus esamus renginio dalyvius, „Žymėti dalyvių grupę“, kurios metu galima pažymėti visą dalyvių grupę. Taip pat, kadangi dalyviai yra pateikiami suskirstyti į grupes, kuriose jie yra netgi puslapiuojami, atsirado veiklos skirtos tai pavaizduoti. Pirmiausia reikia paminėti veiklą „Išskleisti dalyvių grupę“, kuri skirta išskleisti vieną ar kelias dalyvių grupes, kad būtų galima peržiūrėti jose esančius dalyvius. Po to eina veiklos skirtos veiksmams su dalyviais esančiais pirmame dalyvių grupės puslapyje. „Žymėti dalyvius puslapyje“, skirta pažymėti tik tuos dalyvius kurie yra rodomame dalyvių grupės puslapyje. Žinoma taip pat galima žymėti dalyvius ir po vieną, tam skirta veikla yra „Žymėti dalyvį“. Vienas iš pagrindinių reikalavimų buvo

puslapiuoti grupėje esančius dalyvius. Tam skirta veikla yra „Peržiūrėti kitą grupės puslapį“. Kuomet sistemos naudotojas vaikšto per dalyvių grupės puslapius, kiekvieną kartą užkrovus puslapį, sistema automatiškai pažymi šiame puslapyje esančius dalyvius, žinoma jeigu juos anksčiau buvo pažymėjęs sistemos naudotojas. Nesvarbu kokių būdu naudotojas žymės dalyvius, sistema šiuos žymėjimus išsaugo, ir automatiškai perskaičiuoja ir pateikia realiu laiku dalyvių kiekį, esantį renginyje. Kuomet naudotojas baigia žymėti dalyvius, jis gali pašalinti juos iš renginio. Kuomet jis pasirenka šį veiksma, sistema iš karto gauna šalinamų dalyvių sąrašą, ir pašalina juos iš renginio. Po šių veiksmų yra gaunama pašalintų dalyvių grupės, ir atnaujinami renginio dalyvių bei kontaktų sąrašai.

Praktiškai identiški veiksmai yra atliekami su kontaktų žymėjimais, modelyje aprašyti kontaktų žymėjimo galimybes skiriasi tik tuo, kad negalima pažymėti visų kontaktų vienu metu.

Dar vienas iš naujų funkcionalumų yra greitas naujo kontakto sukūrimas iš renginio dalyvių valdymo lango. Naudotojas paprasčiausiai atlieka „Pridėti kontaktą“ veiksma, po kurio įveda naujo kontakto duomenis, ir jeigu šie duomenis yra teisingi, naujas kontaktas automatiškai yra pridodamas prie renginio dalyvių sąrašo.

Žvelgiant į paskirties konceptualaus modelio pateikimo objektą, galima pastebėti, kad jis yra žymiai sudėtingesnis. Pirmiausia, „Renginio dalyvių redagavimo“ objekte atsirado forma „Naujas kontaktas“, kuri yra skirta greitam naujo kontakto sukūrimui, taip pat atsirado formos skirtos renginio informacijos rodomumo nustatymui. Jų pagalba naudotojas gali nustatyti kas matys dalyvius (visi, pakviestieji, niekas), ar rodyti renginio statistiką (kiek kartų buvo peržiūrėtas kvietimas, kiek dalyvių dalyvauja, kiek dalyvių nedalyvauja ir kt.), ir ar rodyti bendrą renginio dalyvių skaičių. Visi šitie nustatymai įtakoja atitinkamo informacinio bloko rodymą renginio kvietimo peržiūros lange. Taip pat galima pastebėti, kad pateikimo objektai „Dalyviai“ ir „Kontaktai“ tapo daug sudėtingesni. Kadangi šie objektai iš esmės atvaizduoja tokio pačio tipo informaciją t.y. naudotojo kontaktus, Pav. 18 išdėstyti pateikimo objektai aprašantys dalyvių valdymo vartotojo sąsają.

Taigi objektą „Dalyviai“ sudaro mygtukas „Žymėti visus dalyvius“, kuris yra skirtas visų dalyvių pažymėjimui. Žemiau yra pateikiamas visas grupių sąrašas, kurį sudaro „Dalyvių grupė“ tipo objektai. „Dalyvių grupė“ pateikimo objektą sudaro forma, skirta visų dalyvių esančių grupėje pažymėjimui, tekstinis objektas „Pavadinimas“, skirtas parodyti grupės pavadinimą,

mygtukas „Rodyti dalyvius“, kurio pagalba galima parodyti arba paslėpti grupės dalyvių sąrašą. „Grupės dalyvių sąrašo“ objektą sudaro tokie vartotojo sąsajos elementai:

- mygtukas „Žymėti puslapį“, skirtas pažymėti šiuo metu peržiūrimo puslapio dalyvius,
- mygtukas „Žymėti grupę“, pažymintis visus grupės dalyvius,
- objektų rinkinys „Dalyvių sąrašas“, kuris liko nepakitęs nuo pradiniam konceptualiniame modelyje pateikto objekto,
- nuorodų rinkinys „Grupės puslapiavimo nuorodos“, kurių pagalba galima pereiti iš vieno dalyvių grupės puslapio į kitą (jeigu grupėje esančių dalyvių skaičius viršija dvidešimt penkis)

### **3.3 Transformacijos įtaka sistemos architektūrai**

Kadangi mano atliktos transformacijos abstrakcijos lygmuo išliko nepakitęs, be to transformacijai realizuoto dėl laiko stokos buvo pasirinktos anksčiau sistemoje naudotos technologijos bei įrankiai, sistemos architektūra nepakito. Esminiai pakitimai įvyko žemesniame sistemos lygmenyje. Pagrindinius pakeitimus apžvelgsiu sekančiuose eksperimentinės dalies skyreliuose

### **3.4 Renginio dalyvių parinkimo tyrimas**

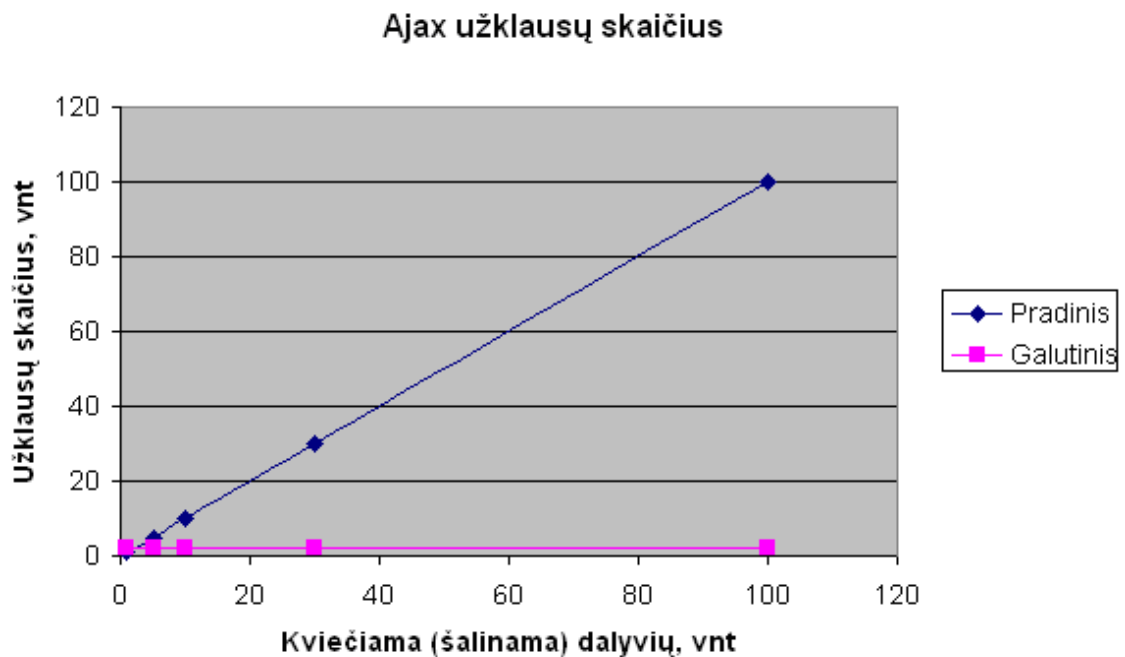
Šiame skyrelyje apžvelgsiu kokie pagrindiniai veiksmai buvo atlikti siekiant pagerinti, renginio dalyvių parinkimo procesą.

#### **3.4.1 Ajax užklausų skaičiaus mažinimas**

Siekiant gauti kuo daugiau informacijos apie internetinio puslapio krovimą, labai daug padėjo Firebug derinimo įrankis. Šis įrankis yra labai populiarios naršyklės Firefox įskiepis, kuris labai gražiai integruojasi į šia naršyklę ir galima matyti tokią informaciją kaip: įvairių failų, sudarančių puslapį, t.y. javascript, css, paveikslukai, krovimosi laikus, aptikti javascript vykdymo klaidas, bet viena iš pagrindinių privalumų yra matyti ajax užklausų vykdymo trukmes, užklausimo ir atsako antraštes, bei siųstus parametrus ir visą atsako tekstą gautą iš serverio.

Pradiniam sistemos variante, nors dalyvių valdymo sąsaja ir nebuvo sudėtinga, tačiau dalyvių kvietimas ir šalinimas sueikvodavo daug resursų. Pagrindinė to priežastis buvo tai jog kiekvieno dalyvio šalinimas (pakvietimas) reikalavė atskiros užklausos į serverį. O kaip žinia,

kiekviena užklausa, nesvarbu ar bus siunčiamas statinis failas pvz., javascript ar css, ar bus vykdomi kažkokie skaičiavimai, užklausa vis tiek truks nuo 100 milisekundžių paprastiem skaičiavimams ir nedideliems statiniams failams (iki 200-300 KB), ir iki 500 milisekundžių ir daugiau, sudėtingesniems skaičiavimams bei didesniems statiniams failams.



**Pav. 19 Ajax užklausių skaičius, kviečiant (šalinant) dalyvius**

Aukščiau esančiame paveiksliuke Pav. 19 pateikiu ajax užklausių skaičiaus palyginimą, pradinėje versijoje ir galutinėje versijoje.

Iš karto galime atkreipti dėmesį į ryškų skirtumą tarp užklausių kiekių pradinėje ir galutinėje versijoje. Pradinėje versijoje užklausių skaičius buvo tiesiškai priklausomas nuo kviečiamų, šalinamų dalyvių skaičiaus, o štai galutinėje versijoje užklausių skaičius reikalingas pašalinti (pakviesti) dalyvius visada išlieka pastovus ir lygus 2. Taip yra todėl, kad galutinėje versijoje visi sistemos naudotojo atliekami kontaktų ar dalyvių žymėjimai yra saugomi pas naudotoją naršyklėje esančiuose javascript masyvuose, ir kuomet naudotojas atlieka pažymėtų dalyvių ar kontaktų pakvietimą ar šalinimą, su pirmąja užklausa per parametrus yra paduodamas visas masyvas, kuris vėliau apdorojamas serveryje PHP pagalba, o antroji užklausa yra skirta gauti pasikeitimams, kurie po to javascript pagalba atvaizduojami naršyklėje.

### **3.4.2 MySQL užklausių optimizavimas, indeksų dėliojimas**

MySQL optimizavimo metu buvo naudoti keli įrankiai. Tai CakePHP integruota duomenų bazės užklausių derinimo priemonė, kurios pagalba galima matyti kokios užklaustos buvo

įvykdytos, kraunant puslapį. Na, o darbui su MySQL lentelėmis, jų indeksų valdymu, bei SQL užklausų vykdymui buvo naudotas phpMyAdmin įrankis.

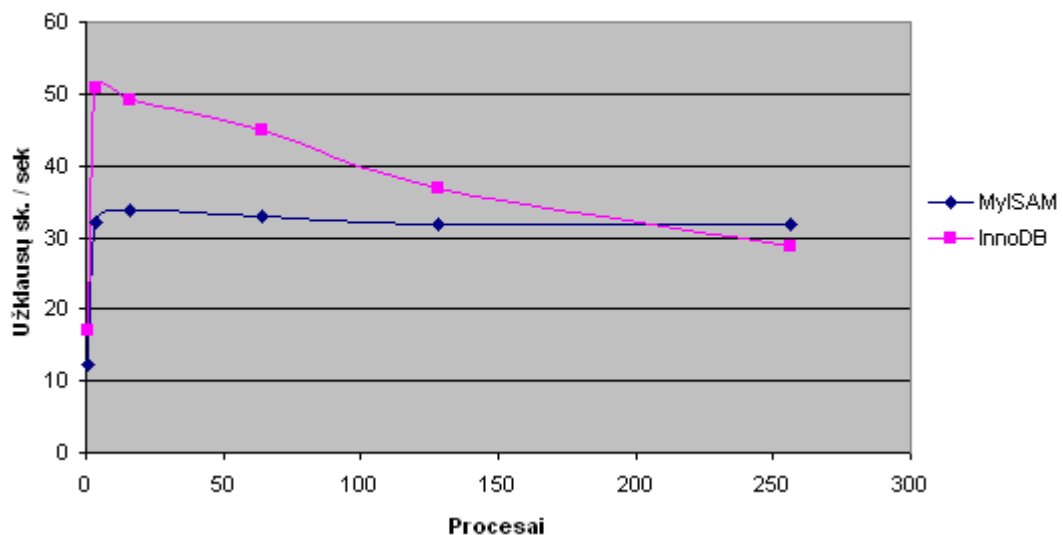
Kalbant apie MySQL optimizavimą vienas iš pradinių žingsnių kurių buvo imtasi tai visų duomenų bazės lentelių konvertavimas iš MyISAM tipo į InnoDB tipą. Tai buvo padaryta dėl to jog InnoDB yra pranašesnis prieš MyISAM dėl to, kad InnoDB palaiko lentelės blokavimą eilučių lygmenyje, o MyISAM dirbdamas su lentele užblokuoja visą lentelę, dėl ko perkrovos metu gali atsirasti duomenų bazės klaidų. Nors, reikia paminėti, kad kalbant apie įrašymo greitį InnoDB yra šiek tiek lėtesnis palyginus su MyISAM.

Viena pagrindinių užklausų, kurią reikėjo optimizuoti buvo tokia:

```
SELECT * FROM `contacts` AS `Contact` LEFT JOIN `contact_groups` AS `ContactGroup` ON `ContactGroup`.`contact_id` = `Contact`.`id` LEFT JOIN `user_contact_groups` AS `UserContactGroup` ON `ContactGroup`.`user_contact_group_id` = `UserContactGroup`.`id` WHERE ( `Contact`.`user_id` = '138' ) ORDER BY `Contact`.`full_name` ASC
```

Ši užklausa naudodama LEFT JOIN tipo sakinius, iš trijų duomenų bazės lentelių gauna duomenis apie kontaktus, bei grupes kurioms jie priklauso. Vėliau šios užklausos rezultatai, norint juos pateikti vartotojo dar bus apdoroti su PHP. Žemiau esančiame paveiksluke Pav. 20, pateikiu šios užklausos palyginimą, kuomet lentelės yra MyISAM ir InnoDB tipo. Testuojama buvo su lentelėmis, kuriose buvo 200 000 – 250 000 įrašų.

#### Užklausos vykdymo laikai



Pav. 20 Kontaktų išrinkimo užklausos naudojant MyISAM ir InnoDB vykdymo laikai

Kaip galima pastebėti, InnoDB šiuo atveju yra apie 58% geresnis, palyginus su MyISAM, kuomet procesų skaičius yra keturi. Tačiau procesų skaičiui didėjant šis pranašumas išnyksta,



matyt dėl to, kad yra kažkokių problemų su procesų valdymu. Tačiau mano atveju realus procesų skaičius nebus didelis, todėl InnoDB naudojimas pilnai pasiteisina.

Kitas svarbus dalykas, kurį būtina padaryti, tai yra sudėlioti lentelėms indeksus. Dažnai kai pamatote jog duomenų bazė veikia lėtai, užklausų eilė vis ilgėja, problema būna tame, jog nėra nurodyti indeksai arba jie nurodyti blogai, ir tai pataisius situacija kartais labai pagerėja. Kaip žinia, indeksas tai atskiras failas, kuris yra rikiuojamas, ir jame yra tik tie lentelės stulpeliai, kurie jums yra reikalingi (kurie naudojami užklausos where arba order by dalyje). Prisimenant anksčiau nagrinėta užklausa galite pastebėti jog where dalis yra tokia: WHERE ( `Contact`.`user\_id` = '138' ). Tai yra atrenkami tik tie kontaktai iš kontaktų lentelės, kuriuos sukūrė vartotojas su id 138.

Indeksas padeda MySQL varikliui greičiau surasti jums reikalingus duomenis. Indeksą galima palyginti su knygos turiniu. Pavyzdžiui jeigu jūs turėtumėte penkių šimtų lapų knygą apie MySQL, ir jums reikėtų rasti skyrelį apie MySQL optimizavimą, tai jeigu nebūtų turinio, jums tektų perskaityti visą knygą, na o turinį galima perskaityti per kelias sekundes ir rasti reikalingą skyrelį.

Norint pamatyti kokie indeksai naudojami užklausos vykdymo metu reikia prieš užklausa prirašyti žodį EXPLAIN.

Taigi anksčiau nagrinėta užklausa būtų tokia:

```
EXPLAIN SELECT * FROM `contacts` AS `Contact` LEFT JOIN `contact_groups` AS `ContactGroup` ON `ContactGroup`.`contact_id` = `Contact`.`id` LEFT JOIN `user_contact_groups` AS `UserContactGroup` ON `ContactGroup`.`user_contact_group_id` = `UserContactGroup`.`id` WHERE ( `Contact`.`user_id` = '138' ) ORDER BY `Contact`.`full_name` ASC
```

Ir jos rezultatas pateikiamas žemiau esančioje lentelėje Lentelė Nr. 24

table	type	possible keys	key	key len	Ref	rows	Extra
Contact	ALL	NULL	NULL	NULL	NULL	200000	Using where; Using filesort
ContactGroup	ref	contact id	contact id	4	invup.Contact.id	12	
UserContactGroup	eq_ref	PRIMARY	PRIMARY	4	invup.ContactGroup.user_contact_group_id	1	

**Lentelė Nr. 24 Explain užklausos rezultatas**

Ką gi reiškia šis rezultatas?

- Table stulpelyje, pateikiamos visos lentelės, kurios buvo naudotas užklausiai įvykdyti. Šiuo atveju matome jog buvo naudotos trys lentelės.
- Type stulpelis yra labai svarbus, nes parodo kokio tipo sujungimas buvo atliktas. Einant eilės tvarka nuo geriausio iki blogiausio tipai yra tokie: system, const, eq\_ref, ref, range, index, all. Kaip matome contact lentelei yra naudojamas all tipas, kas yra blogiausias įmanomas variantas.
- Possible\_keys parodo kokie galimi indeksai atitinkamai lentelei
- Key, parodo kuris indeksas buvo naudotas
- Key\_len parodo kokio ilgio indeksas buvo naudotas, kuo trumpesnis tuo geriau
- Ref parodo kuris stulpelis buvo naudotas
- Rows, tai eilučių kieki, kurį kaip mano MySQL reikės išnagrinėti norint gauti rezultatą
- Extra, papildoma informacija, labai blogai jeigu yra įrašai tokie kaip “using temporary” ar “using filesort”

Taigi ši užklausa tikrai yra katastrofa. Kaip matome kontaktų lentelėje nėra jokių indeksų, taigi MySQL teko pereiti per visus du šimtus tūkstančių įrašų, norint surasti atitinkamo vartotojo kontaktus. Norint išspręsti šią problemą, reikia uždėti indeksą ant contact lentelės user\_id stulpelio.

table	type	possible keys	key	key len	Ref	rows	Extra
Contact	ref	User id	User id	NULL	const	993	Using where; Using filesort
ContactGroup	ref	contact id	contact id	4	invup.Contact.id	12	
UserContactGroup	eq_ref	PRIMARY	PRIMARY	4	invup.ContactGroup.user_contact_group id	1	

**Lentelė Nr. 25 Explain užklauskos rezultatas, uždėjus indeksą**

Kaip matome dabar norint gauti rezultata, MySQL teko pereiti tik apie tūkstantį įrašų. Ir visa tai dėl to kad uždėjome indeksą ant contact lentelės user\_id stulpelio. Na žinoma ir užklausos vykdymo laikas sutrumpėjo nuo 0.2 sekundės iki 0.03 sekundės.

### 3.4.3 Javascript naudojimas

Kuriant sistemą buvo naudotas patogi ir galinga jQuery javascript biblioteka. Kurios pagalba galima nesudėtingai, praktiškai parašius vieną eilutę, surasti html elementą tinklapyje, ir atlikti su juo eilę veiksmų, nuo klasės pridėjimo iki visiško ištrynimo. Tačiau kaip ir su bet kokia biblioteka ar karkasu, šie veiksmai bus atliekami lėčiau negu su gryno javascript metodais. Nepaisant to, jeigu laikysimės jQuery greičio hierarchijos principų, skirtumas tarp jQuery ir javascript bus nereikšmingas. „Greičio hierarchija“ pradant nuo greičiausio iki lėčiausio atrodo taip:

1. ID: `$(, #elemento-id)`
2. Elementas: `$(, div)`
3. Klasė: `$(, .elemento-klase)`

Pirmajame variante, kuomet elemento išrinkimas atliekamas pagal id, jQuery naudoja gimtąjį Javascript metodą `getElementById()`. Tai yra pats greičiausias būdas surasti elementą dokumento objektų modelyje (DOM).

Norint rasti elementą pagal klasės vardą, kaip parodyta trečiame įrašė, jQuery eina per visą dokumento objektų modelį. Kita vertus jeigu mums reikia rasti elementą, yra naudojamas metodas javascript metodas `getElementsByTagName()`, taip apribojant elementų skaičių, per kuriuos reikia pereiti. Taigi žinant kokiam elemente yra naudojama tam tikra klasė galima parašyti taip

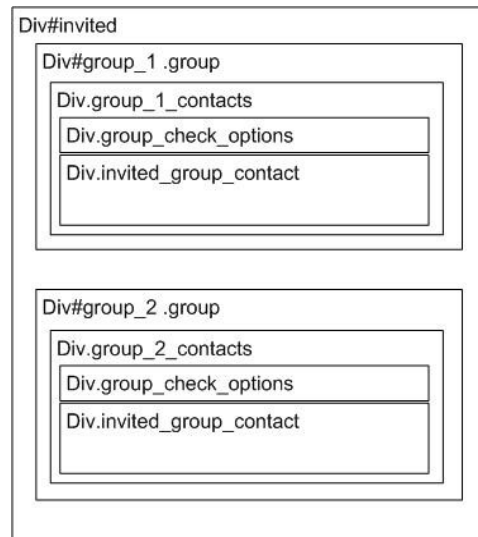
`$( 'div .contact_group' )` , o ne `$( '.contact_group' )`

Panašiai galima padaryti ir su elementų id, jeigu yra žinomas id elemento, kuriame yra ieškamos klasės

`$( '#contacts .contact_group' )` , o ne `$( '.contact_group' )`

Kadangi dalyvių parinkimo langas yra labai sudėtingas iš vartotojo sąsajos pateikimo ir javascript atliekamų veiksmų pusės, siekiant pagerinti naudotojo darbo su šiuo langu kokybę, vartotojo sąsaja buvo suskirstyta į tokius dokumento objekto modelio elementus, kurie leistų

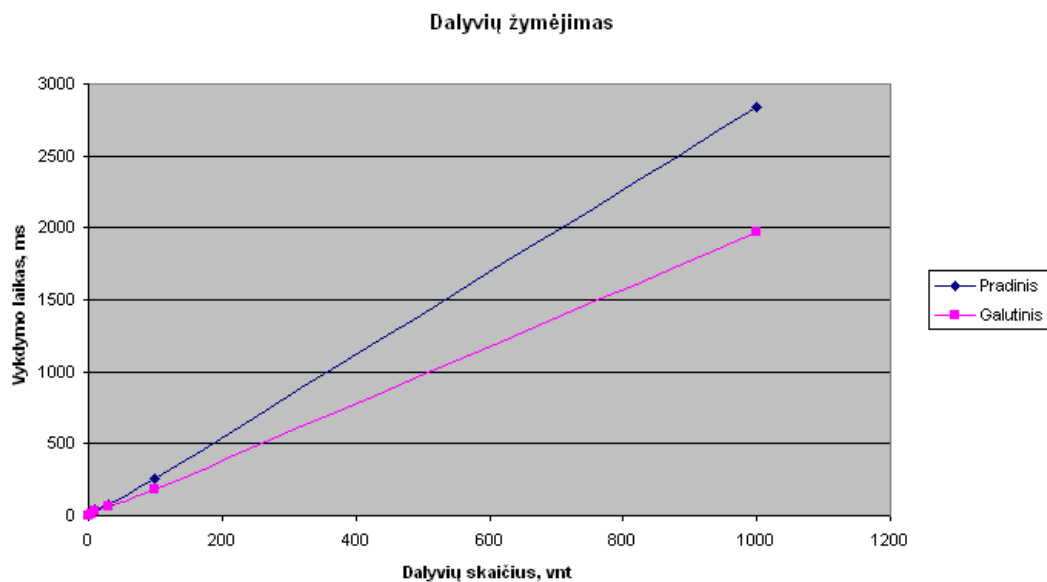
greitą įvairiapusišką elementų išrinkimą, pagal dalyvių grupes, aktyvius grupių puslapius, dalyvius tuose puslapiuose. Žemiau esančiame paveikslėlyje Pav. 21, pateiktas dalyvių valdymo lango pakviestųjų sąrašo dokumento objektų modelio elementų vaizdas.



**Pav. 21 Dalyvių valdymo lango, pakviestųjų sąrašo DOM elementai**

Elementų pavadinimai šiame paveiksluke pavaizduoti naudojant jQuery sintaksę. Tekstas esantis prieš „#“ ar „.“ yra html elementas, na o tekstas einantis po „#“ yra to html elemento id atributas, o tekstas einantis po „.“ yra to elemento class atributas.

Žemiau esančiame grafike, pateikiu dalyvių žymėjimų trukmių palyginimus, pradinėje ir galutinėje versijoje. Kaip matome, atlikus pakeitimus, dalyvių žymėjimas paspartėjo apie 30%.



**Pav. 22 Dalyvių žymėjimo greičio palyginimas pradiniam ir galutiniam variante**

Vykdymo laikas buvo skaičiuojamas pasinaudojus javascript Date objekto metodu getTime(). Pradžioje buvo išsaugomas laikas prieš žymėjimo vykdymą, o atlikus vykdymą buvo suskaičiuojamas laiko skirtumas, kurį po to išvesdavau javascript alert() metodo pagalba.

### 3.4.4 Kodo eilučių pokytis

Nors savo eksperimente aš tyrinėjau tik nedidelę visos sistemos atliekamų funkcijų dalį, tačiau pokyčiai kodo eilučių lygmenyje, ypač kalbant apie javascript kodą, buvo akivaizdūs. Kodo eilučių pokyčiams palyginti, aš pasinaudojau Araxis Merge įrankiu, kurio pagalba galima gauti įvairios statistikos apie failų palyginimą.

Failas	Eilučių skaičius				
	Nepakito	Pakito	Įterpta	Ištrinta	Pokytis
<b>Javascript</b>					
jq_controller_events.js	565	3714	1678	71	86,80 %
jq_controller_contacts.js	473	30	27	0	5,96 %
<b>PHP</b>					
events_controller.php	1439	836	433	183	36,75 %
event_entrants_controller.php	276	1350	679	43	83,03 %
contacts_controller.php	1839	53	49	0	2,80 %
<b>Šablonų failai</b>					
events/add.ctp	34	13	8	2	27,66 %
events/event_entrants.ctp	4	521	260	170	99,24 %
event_entrants/contact.ctp	12	437	187	176	97,33 %
event_entrants/invitedlist.ctp	34	426	120	78	92,61 %
event_entrants/paged_invited_list.ctp	0	39	39	0	100,00 %
contacts/import.ctp	63	32	13	4	33,68 %

Lentelė Nr. 26 Failų pokyčiai

Aukščiau pateiktoje lentelėje Lentelė Nr. 26, surašyti labiausiai pasikeitę failai. Lentelėje pateikiamas palyginimas tarp pradinės versijos ir galutinės versijos failų. Pokyčiai pateikiami vertinant pagal keturis parametrus: kiek eilučių nepakito, kiek pakito, kiek buvo įterpta ir kiek ištrinta. Taip pat failai atskirti į tris grupes: tai javascript failai, PHP failai, bei šablonų (html) failai. Kaip galite pastebėti, labiausiai kito javascript failai ypač jq\_controller\_events.js, kuriame yra renginio dalyvių parinkimo vartotojo sąsajos veiksmų logikos realizacija. Iš php failų taip pat keitėsi failai, atsakingi už renginio dalyvių valdymą. Iš jų labiausiai keitėsi failas event\_entrants\_controller.php, kuris yra atsakingas už dalyvių šalinimo bei kvietimo valdymą, bei informacijos pateikimą. Na, o labiausiai keitėsi failai, skirti atvaizduoti vartotojo sąsają. Dauguma jų buvo praktiškai pilnai perrašyti, tam kad būtų tenkinami nauji reikalavimai.

## IŠVADOS

Informacinė visuomenė vis didėja ir plečiasi, internetinės paslaugos tampa kasdienybe, žmonės supranta, kad internetu problemas galima išspręsti daug greičiau ir paprasčiau.

Todėl sukurtoji sistema bus aktuali daugeliui žmonių grupių, kuriuos organizuoja privačius arba viešus renginius. Sistema suteikia galimybę paskelbti tiek privačius, tiek viešus renginius, kviešti į juos asmenis bei iš karto tų asmenų reakcijas į pakvietimus.

Eksperimentinio tyrimo metu buvo atlikta pradinio konceptualinio modelio homogeninė horizontalioji transformacija, kurios dėka buvo galima aiškiau suprasti būtinausius analizuojamos sistemos dalies optimizavimo etapus. Taip pat eksperimento metu buvo išanalizuota eilė būdų kaip santykinai nesudėtingai galima ženkliai pagerinti internetinių sistemų veikimą.

Visi atlikti pakeitimai buvo atlikti rankiniu būdu, todėl kad kuriant šią sistemą naudotos technologijos atsirado palyginus neseniai. Ir modeliavimo įrankiai nesuteikia galimybių automatizuoti tokių sistemų kūrimo. Tačiau po truputį yra judama šia linkme, štai kad ir sistemos kūrime naudotas CakePHP karkasas, suteikia galimybę iš esamos duomenų bazės struktūros sugeneruoti paprastą internetinę programą, atliekančią kūrimo-skaitymo-atnaujimo-trynimo (angl. CRUD) funkcijas,. Ateityje galbūt bus sukurti įrankiai, kurie galėtų tai daryti ir iš UML modelių.

# LITERATŪRA

- [STUI ] V. Štūkys, R.Damaševičius. Modelių ir programų abstrakčios transformacijos, 2008
- [OSCA] Óscar Pastor, Juan Carlos Molina. Model-Driven Architecture in Practice– A Software Production Environment Based on Conceptual Modeling, 2007.
- [BAUM] Hubert Baumeister, Nora Koch, and Luis Mandel. Towards a UML Extension for Hypermedia Design [žiūrėta 2009 05 02] Prieiga per internetą: <http://www.pst.informatik.uni-muenchen.de/projekte/forsoft/pubs/uml99.pdf>
- [1] Vince Barnes, Web Site Maintainability [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.htmlgoodies.com/introduction/intro/article.php/3473641>
- [2] Michele Neylon, Sending bulk email without spamming [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.mneylon.com/blog/archives/2005/01/17/mass-emailing-dos-and-donts/>
- [3] George Dillon, HTML e-mail is STILL evil!!! [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.georgedillon.com/web/html\\_email\\_is\\_evil.shtml](http://www.georgedillon.com/web/html_email_is_evil.shtml)
- [4] Mark Wood, SMS Bulk Messaging, the problem and the solution [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.ceasa-int.org/library/7\\_sms\\_mass\\_messaging\\_problems\\_V1-2.doc](http://www.ceasa-int.org/library/7_sms_mass_messaging_problems_V1-2.doc)
- [5] Baliani, Multilanguage Web Design [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.baliani.com/servicedetail.asp?sid=27&bg=20>
- [6] Eionet, Benefits of a multilingual website [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.eionet.europa.eu/software/design/multilinguality/benefits>
- [7] Grady Booch, The architecture of Web applications [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.ibm.com/developerworks/ibm/library/it-booch\\_web/](http://www.ibm.com/developerworks/ibm/library/it-booch_web/)
- [8] Top 10 PHP MVC frameworks, [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.mustap.com/phpzone\\_post\\_73\\_top-10-php-mvc-frameworks](http://www.mustap.com/phpzone_post_73_top-10-php-mvc-frameworks)
- [9] PHP karkasai [žiūrėta 2007 10 28] Prieiga per internetą: <http://pixel.lt/php-frameworkai.html>
- [10] Model View Controller [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.phpwact.org/pattern/model\\_view\\_controller](http://www.phpwact.org/pattern/model_view_controller)
- [11] What is SOA? - SOA and Web Services Explained, [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.indiawebdevelopers.com/articles/SOA.asp>
- [12] Roger L. Costello, Building Web Services the REST Way [žiūrėta 2007 10 28]
- [13] Sparx systems, The Unified Modeling Language [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.sparxsystems.com.au/platforms/uml.html>
- [14] Scott W. Ambler, Introduction to the Diagrams of UML 2.0 [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.agilemodeling.com/essays/umlDiagrams.htm>
- [15] Doug Hughes, What Is UML? [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.alagad.com/go/blog-entry/what-is-uml>
- [16] No Magic, Inc, Key Benefits [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.magicdraw.com/main.php?ts=navig&cmd\\_show=1&menu=programmer&](http://www.magicdraw.com/main.php?ts=navig&cmd_show=1&menu=programmer&)
- [17] Dan Wellman, Product Review: Zend Studio 4 [žiūrėta 2007 10 28] Prieiga per internetą: <http://webhosting.devshed.com/c/a/Web-Hosting-News/Product-Review-Zend-Studio-4/>
- [18] Zend Studio, Zend Studio Description [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.php-editors.com/review/zend-studio.htm>
- [19] Ben Collins-Sussman, Brian W Fitzpatrick, Version control with Subversion [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.itwriting.com/subversionreview.php>
- [20] Gary Pollic, Test before you code [žiūrėta 2007 10 28] Prieiga per internetą: <http://www-128.ibm.com/developerworks/rational/library/4929.html>
- [21] James Bach, Good Practice Hunting [žiūrėta 2007 10 28] Prieiga per internetą: [http://www.satisfice.com/articles/good\\_practice\\_hunting.pdf](http://www.satisfice.com/articles/good_practice_hunting.pdf)
- [22] Ram Chillarege, Software Testinf Best Practices [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.chillarege.com/authwork/TestingBestPractice.pdf>
- [23] Phil Zoio, Testing 1,2,3... [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.oracle.com/technology/oramag/oracle/05-jul/o45testing.html>
- [24] Gopal Gupta, Language Based Software Engineering [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.utdallas.edu/~gupta/alp.pdf>
- [25] Patrick Freudens7.2, Jan Buck, Martin Nussbaumer ir Martin Gaedke, Model-driven Construction of Workflow-based Web Applications with Domain-specific Languages [žiūrėta 2007 10 28] Prieiga per internetą: <http://wise.vub.ac.be/MDWE2007/downloads/freudens7.2.pdf>  
Prieiga per internetą: <http://www.xfront.com/REST-Web-Services.html>

- [26] Pros and Cons to Web 2.0, [žiūrėta 2007 10 28] Prieiga per internetą:  
<http://www.ajaxwith.com/Pros-and-Cons-to-Web-20.html>
- [27] Tim O'Reilly, What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software [žiūrėta 2007 10 28] Prieiga per internetą:  
<http://www.oreillynet.com/lpt/a/6228>
- [28] Web Services Activity [žiūrėta 2007 10 28] Prieiga per internetą:  
<http://www.w3.org/2002/ws/>
- [29] Web Services Description Language [žiūrėta 2007 10 28] Prieiga per internetą:  
<http://en.wikipedia.org/wiki/WSDL>
- [30] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, Web Services Description Language (WSDL) 1.1 [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.w3.org/TR/wsdl>
- [31] Nicholas Quaine, SOAP Basics [žiūrėta 2007 10 28] Prieiga per internetą:  
<http://www.soapuser.com/basics1.html>
- [32] What is SOA? - SOA and Web Services Explained [žiūrėta 2007 10 28] Prieiga per internetą: <http://www.indiawebdevelopers.com/articles/SOA.asp>
- [33] Mashup (web application hybrid) [žiūrėta 2007 10 28] Prieiga per internetą:  
[http://en.wikipedia.org/wiki/Mashup\\_\(web\\_application\\_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
- [34] [žiūrėta 2007 10 28] Prieiga per internetą: <http://pipes.yahoo.com/pipes/docs>
- [35] [žiūrėta 2007 10 28] Prieiga per internetą: [www.php.net](http://www.php.net)
- [36] Dabrovolskas D., Python [interaktyvus] [žiūrėta 2007 10 30] Prieiga per internetą:  
<http://uosis.mif.vu.lt/~dado1945/blog/tech/python.html>
- [37] PHP karkasai [žiūrėta 2007 10 30] Prieiga per internetą:  
<http://pixel.lt/php-frameworkai.html>
- [38] [žiūrėta 2007 10 30] Prieiga per internetą: <http://www.cakephp.org>
- [39] [žiūrėta 2007 10 30] Prieiga per internetą: <http://www.symfony-project.com/about>
- [40] [žiūrėta 2007 10 30] Prieiga per internetą: <http://www.djangoproject.com/>
- [41] [žiūrėta 2007 10 30] Prieiga per internetą:  
[www.w3schools.org/asp/asp\\_ref\\_server.php](http://www.w3schools.org/asp/asp_ref_server.php)
- [42] Lisa Radcliffe, The Pros and Cons of ASPs [žiūrėta 2007 10 30] Prieiga per internetą: [http://www.unitedwaynyc.com/technews/v5\\_n2\\_a2.html](http://www.unitedwaynyc.com/technews/v5_n2_a2.html)
- [43] Justin Silverton, Top 5 javascript frameworks [žiūrėta 2007 10 30] Prieiga per internetą: <http://www.whenpenguinsattack.com/2007/04/24/top-5-javascript-frameworks/>
- [44] Brian Reindel, How to choose a JavaScript framework [žiūrėta 2007 10 30] Prieiga per internetą: <http://blog.reindel.com/2007/10/30/how-to-choose-a-javascript-framework/>



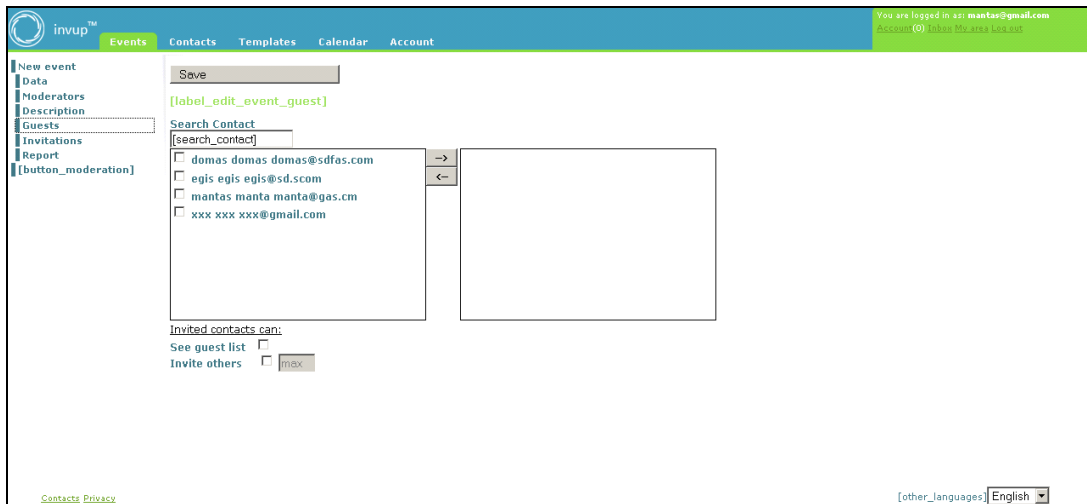
## TERMINŲ IR SANTRAUPŲ ŽODYNAS

Sąvoka	Apibrėžimas
<b>Apache</b>	Tai yra atviro kodo tinklo serverio programa skirta įvairioms operacinėms sistemoms, kad palaikytų tinklapių veikimą fiziniuose serveriuose.
<b>Darbinis pagrindas (framework)</b>	Darbinis pagrindas yra pakartotinio panaudojimo programinė įranga, skirta programinės įrangos sistemoms kurti. Darbinis pagrindas gali turėti pagalbinių programų, kodo bibliotekų, skriptingo kalbų ar kitokių pagalbinių įrankių, kurių pagalba galima plėtoti ir sudėti skirtingus komponentus į vieną pilnai veikiančią sistemą.
<b>DUK</b>	Dažnai užduodami klausimai
<b>Hostingas</b>	Hostingas - internetinių paslaugų grupė, apimanti puslapių talpinimą, elektroninio pašto servisą ir kita. Hostingas reiškia tiek paslaugą (tu atveju, kai tai atlieka firma, kuri teikia hostingo paslaugas), tiek patį procesą, tai fizinė patalpa, serveris (kompiuteris) ir interneto puslapio priežiūra.
<b>HTML</b>	Hiperteksto programavimo kalba: žymių ir taisyklių aibė skirta statinių www tinklapių kūrimui.
<b>Internetinė naršyklė</b>	Naršyklė (angl. browser) yra programa, skirta sklaidyti tinklo serverių (angl. web server) pateikiamą informaciją, dažniausiai naršant žiniatinklyje arba vidiniuose įmonės tinkluose (intranet). Interneto puslapiai gali turėti nuorodas į kitus puslapius ir naudotojas gali greitai ir lengvai pasiekti informaciją sekdamas šiomis nuorodomis.
<b>MySQL</b>	MySQL - viena iš reliacinių duomenų bazių valdymo sistemų (liet. santrumpa RDBVS, angl. - RDBMS), palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu. MySQL yra atviro kodo programinė įranga.
<b>MVV (Model-view-controller)</b>	Model-view-controller (MVC) yra architektūrinis šablonas naudojamas programinės įrangos kūrimui ir plėtojimui. Sudėtingose sistemose, kuriuos išvedinėja didžiulį duomenų kiekį vartotojams, dažnai norima atskirti duomenis (modelį) nuo vartotojo sąsajos (vaizdo). Toks išskyrimas neįtakoja sistemos valdomų duomenų, norint pakeisti vartotojo sąsają. Šiuos du komponentus susieja valdiklis, kuris yra tarsi "smegenys".
<b>Operacinė sistema</b>	Operacinė sistema (OS) - tai speciali programinė įranga, abstrahuojanti naudotojo bei programų darbą. Moderniausios operacinės sistemos sudaro galimybę dirbti daugeliui vartotojų vienu metu daugialypėje aplinkoje, užtikrina bylų (failų) apsaugą, turi daug kitų naudingų savybių. Dauguma operacinių sistemų yra pirma programinė įranga, kurią pradeda vykdyti įjungtas kompiuteris.
<b>Panaudos atvejis</b>	Panaudos atvejis yra išorinis sistemos vaizdas, kuris parodo tam tikrus veiksmus, kuriuos naudotojas turi atlikti, norėdamas įvykdyti tam tikrą užduotį. Kiekvienas panaudos atvejis yra sudarytas iš scenarijų. Kiekvienas scenarijus yra žingsnių seka, kuri nusako tarpusio sąveiką tarp naudotojo ir sistemos.
<b>PHP</b>	PHP - plačiai paplitusi dinaminė interpretuojama programavimo kalba,

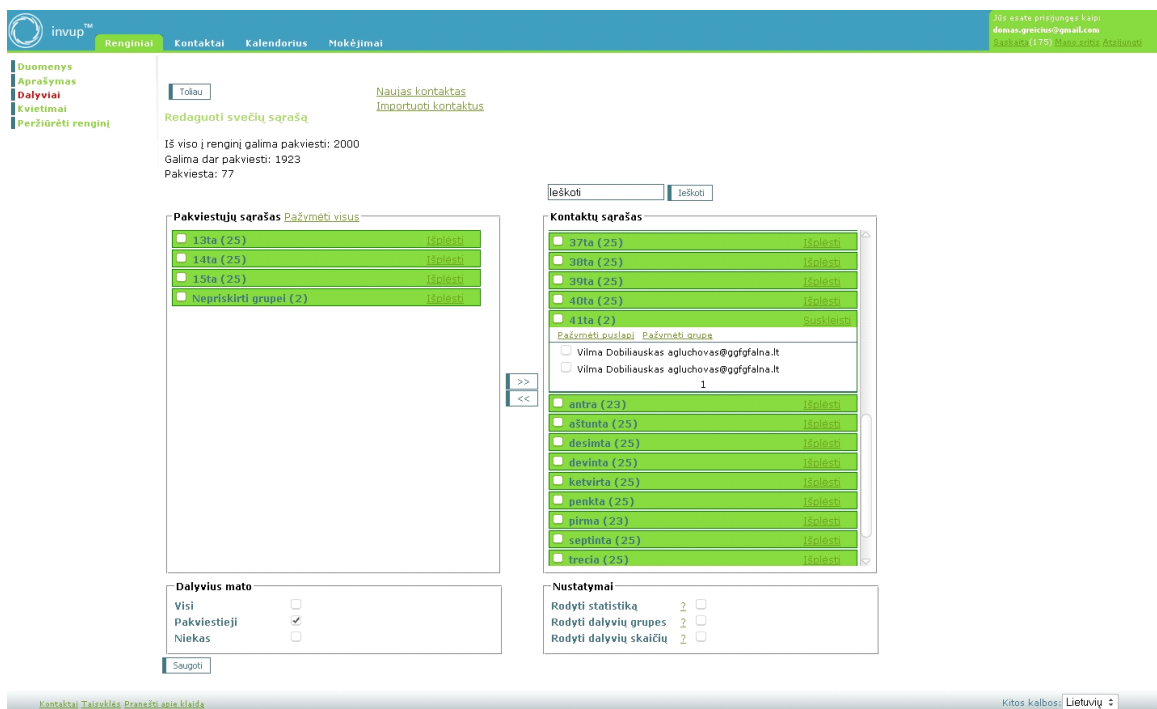
	sukurta 1997 m. ir specialiai pritaikyta svetainių kūrimui. PHP sintaksė panašumi į daugelį struktūrinių kalbų, ypatingai į C bei Perl. PHP kalba yra atviro kodo ir tai yra viena priežasčių, dėl ko kalba yra nors ir nesudėtinga, bet gana lanksti - veikia daugumoje operacinių sistemų, palaiko nemažai reliacinių duomenų bazių bei veikia su dauguma interneto serverių - CGI, FastCGI, ISAPI ir kitais protokolais.
<b>Programinė įranga</b>	Programinė įranga tai kompiuterio vykdomų instrukcijų seka, skirta tam tikriems veiksmams atlikti. Dažniausiai tokia įranga parašoma naudojant programavimo kalbas, o vėliau kompiliuojant ar interpretuojant parašytą kodą.
<b>Serveris</b>	Specialios paskirties kompiuteris, skirtas kitų kompiuterių aptarnavimui (neskirtas asmeniniam naudojimui). Dažniausiai išskiriamos kelios serverių rūšys (pagal paskirtį): failų serveriai (FTP, SMB, etc.), pašto serveriai (POP3, SMTP, IMAP), tinklalapių serveriai (angliškai: web server [host]) (HTTP), duomenų bazių serveriai (SQL), aplikacijų serveriai, ir pan. Daugeliu atvejų tas pats kompiuteris gali atlikti visas šias funkcijas. Serverių naudojimas pagrįstas client-server architektūra: klientas (kompiuteris ar programa) prisijungia prie serverio, skirto tam tikrai užduočiai, ir siunčia pastarajam įsakymus, kurie vykdomi.
<b>Sistema</b>	Pakvietimų į organizuojamus renginius planavimo ir valdymo informacinė sistema.
<b>Techninė įranga</b>	Kompiuterių techninė įranga apima visas fizines kompiuterio dalis, bet ne programinę įrangą, valdančią šias dalis.
<b>WWW (World Wide Web)</b>	Žiniatinklis, pasaulinis tinklas (angl. <i>World Wide Web</i> arba <i>WWW</i> ) – interneto dalis, resursai, kuriuos internete galima pasiekti naudojant URL (Vieningus Resursų Identifikatorius). Dėl savo naudojimo platumo jis dabar yra neretai supainiojamas su visu internetu apskritai, tačiau tai yra tik interneto poaibis.
<b>UML</b>	(Unified Modeling Language, <i>Vieninga modeliavimo kalba</i> ) – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus.

Lentelė Nr. 27 Terminai ir sutrumpinimai

# PRIEDAI



Pav. 23 Pradinis renginio dalyvių valdymo langas



Pav. 24 Pakeistas renginio dalyvių langas

## SISTEMOS ĮDIEGIMO PATVIRTINIMO AKTAS

2009 m. gegužės mėn. 15 d.

Vilnius

Inify, UAB, įmonės kodas 300655539, buveinės adresas Maumedžių g. 19-17, LT – 08420 Vilnius, pagal įstatus atstovaujama direktoriaus Manto Žvirono, toliau vadinama „Užsakovu“, patvirtina, kad Egidijus Gegeckas ir Domas Greičius, toliau vadinami „Autoriais“, sėkmingai įdiegė ir perdavė Renginių organizavimo ir valdymo sistemą **Užsakovui**, kurią galima pasiekti internete adresu: [www.invup.lt](http://www.invup.lt)

### Užsakovas

Inify, UAB

Įm.k. 300655539

Maumedžių g. 19-17, LT – 08420 Vilnius

Direktorius Mantas Žvironas

  
(parašas)

