

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

**Donatas Ražanskas**

**PASŲ IR VIZŲ IŠDAVIMO INFORMACINĖ  
SISTEMA NAUDOJANT MVC ARCHITEKTŪRĄ**

Magistro darbas

**Vadovas  
doc. dr. R. Butleris**

**KAUNAS, 2006**

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

**PASŲ IR VIZŲ IŠDAVIMO INFORMACINĖ  
SISTEMA NAUDOJANT MVC ARCHITEKTŪRĄ**

Magistro darbas

Vadovas

doc. dr. R. Butleris

Recenzentas

doc. dr. A. Riškus

Atliko

FM 0/4 gr. stud.

D. Ražanskas

**KAUNAS, 2006**

## SUMMARY

D.Ražanskas. Passports and visas issue information system using MVC architecture. Master work in science of informatics. Supervisor dr. doc. R.Butleris. Department of Information Systems Engineering, Faculty of Informatics.

This master work seeks to create and to install passports and visas issue information system and to make MVC framework which may be used in other information systems creation process.

Passports and visas issues information system was designed in order to automate the process of giving passports and visas. To create this information system was programmed MVC framework. Mentioned framework is able to call modules methods from template files. This ability makes programming process more speedy.

This master work consists of such steps: reviewing similar passports and visas issue information systems, analyzing MVC framework, choosing templates class, programming languages and according to MVC framework was created passports and visas issue web application.

It will be observed that passports and visas issue information system is installed and used in practically and the experience of exploitation is 0.5 years.

# TURINYS

Įvadas .....	6
1. PIVI IS, architektūros, šablonų apdorojimo klasių ir technologijų analizė .....	7
1.1. Pasų ir vizų išdavimo internetinių svetainių palyginamoji analizė .....	7
1.2. Programavimo technologijų analizė .....	8
1.3. Kuriamos internetinės svetainės architektūra .....	10
1.3.1. Paprastas internetinės svetainės kūrimas .....	10
1.3.2. Internetinių svetainių kūrimas naudojant MVC modelį .....	11
1.3.3. Panašūs architektūriniai sprendimai .....	13
1.3.4. Kuriamos MVC architektūros privalumas .....	16
1.3.5. Šablonų apdorojimo klasių analizė .....	18
1.4. Analizės išvados .....	20
2. PIVI IS ir MVC architektūra .....	21
2.1. PIVI IS veiklos sfera .....	21
2.2. Vystymo etapai .....	22
2.3. Pagrindiniai IS ir MVC architektūros kūrimo etapai .....	23
2.4. Pagrindinės IS funkcijos .....	25
2.5. IS keliami reikalavimai .....	25
2.5.1. Funkciniai reikalavimai .....	25
2.5.2. Nefunkciniai reikalavimai .....	26
2.6. Duomenų bazės schema .....	28
3. PIVI IS ir MVC realizacija .....	31
3.1. Sistemos pasiskirstymas .....	31
3.2. Klasių diagramos .....	32
3.2.1. MVC architektūros klasių diagrama .....	32
3.2.2. PIVI IS klasių diagrama .....	33
3.3. PIVI IS realizacijos aprašymas .....	34
3.3.1. Pateikimo sąsaja pasų ir vizų užsakovams .....	34
3.3.2. Vadybininko sąsaja .....	34
3.4. PIVI IS įdiegimas .....	35
3.4.1. Įdiegimas .....	36
3.4.2. Konfigūravimas .....	37
4. MVC architektūros tyrimas .....	39
4.1. Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn .....	39
4.1.1. Algoritmo gilyn tyrimo eiga .....	39
4.1.2. Algoritmo gilyn laikiniai rezultatai .....	40
4.2. Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn .....	41
4.2.1. Algoritmo platyn tyrimo eiga .....	41
4.2.2. Algoritmo platyn laikiniai rezultatai .....	42
4.3. Algoritmų gilyn ir platyn palyginimas .....	43
4.4. Parametrų ir modulių išskleidimo palyginimas .....	43
4.5. Tyrimo išvados .....	44
5. Išvados .....	45
6. Literatūra .....	46
7. Terminų ir santrumpų žodynas .....	47
Priedai .....	48
1 priedas. UML diagramos ir specifikacijos .....	49
2 priedas. Instrukcija vartotojams .....	65

## PAVEIKSLAI

<b>1 pav.</b>	Sistemos skirstymas į lygmenis.....	11
<b>2 pav.</b>	MVC modelio skirstymas į lygmenis.....	12
<b>3 pav.</b>	MVC modelio ryšiai.....	12
<b>4 pav.</b>	MVC modelio ryšiai, kai galima kreiptis į valdiklį iš šablonų bylų .....	17
<b>5 pav.</b>	Veiklos kontekstas.....	21
<b>6 pav.</b>	Duomenų bazės diagrama .....	28
<b>7 pav.</b>	Duomenų bazės diagramos tęsinys .....	29
<b>8 pav.</b>	Sistemos paskirstymo diagrama .....	31
<b>9 pav.</b>	MVC modelio klasių diagrama .....	32
<b>10 pav.</b>	PIVI IS klasių diagrama .....	33
<b>11 pav.</b>	PIVI IS vartotojo sąsajos pavyzdys.....	34
<b>12 pav.</b>	PIVI IS vadybininko sąsajos pavyzdys .....	35
<b>13 pav.</b>	PIVI IS įdiegimo diagrama .....	36
<b>14 pav.</b>	Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn .....	39
<b>15 pav.</b>	Algoritmo gilyn vykdymo laikas.....	40
<b>16 pav.</b>	Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn .....	41
<b>17 pav.</b>	Algoritmo platyn vykdymo laikas.....	42
<b>18 pav.</b>	Algoritmų gilyn ir platyn palyginimas .....	43
<b>19 pav.</b>	Parametrų ir modulių išskleidimas .....	44

## LENTELĖS

Panašių projektų palyginimas.....	7
Programavimo technologijų charakteristikos.....	8
Programavimo technologijų įvertinimas .....	9
Šabloninių failų elementų keitimo laikas, kai naudojama 20 kintamųjų ir 10 iteracijų.....	18
Šabloninių failų elementų keitimo laikas, kai naudojami 5 kintamieji ir 10 iteracijų .....	19
Šabloninių failų elementų keitimo laikas, kai naudojama 100 kintamųjų ir 5 iteracijos .....	19
Šabloninių failų ciklų vykdymo laikas, kai teksto dydis 10kb., 100 ciklų, 10 iteracijų.....	20
Vystymo etapai ir jiems atlikti reikalinga techninė bei programinė įranga .....	22
MVC architektūros kūrimo etapai.....	23
PIVI IS kūrimo etapai .....	24
DB lentelių aprašymas .....	30
Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn .....	40
Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn .....	42
Parametrų išskleidimas, kai modulių kiekis pastovus.....	43
Modulių išskleidimas, kai parametrų kiekis pastovus.....	44
Terminai ir santrumpos .....	47

## IVADAS

Vis labiau plintantis informacinių technologijų naudojimas yra universalus reiškinys visuose verslo ir administracijos lygiuose. Naujos informacijos apdorojimo technologijos kuria naujas verslo sritis, o tai leidžia aplenkti konkurentus - pateikti savo prekes bei paslaugas, rasti naujų verslo partnerių arba klientų[1].

Šiuolaikinių organizacijų veiklos pobūdis ir verslo aplinkos savybės (konkurencija pasaulio rinkose, vadyba ir kontrolė globalioje rinkoje, žiniomis ir informacija grindžiama ekonomika, globalinės darbo grupės, atsiribojimas nuo geografinės padėties ir kt.) reikalauja informacinių technologijų paslaugų. Šiandien vienos organizacijos yra labiau kompiuterizuotos, kitos – mažiau. Tai priklauso nuo organizacijos tikslų bei informacinės sistemos (IS) svarbos suvokimo. Taigi yra aktualu kalbėti apie organizacijos funkcijų kompiuterizavimą, nes kompiuterizuotos IS leidžia organizacijoms efektyviau atsakyti į aplinkos poreikius.

Kalbant apie pasų ir vizų išdavimo (PIVI) procesą, esminis poreikis, kurį suformuluoja to proceso dalyviai (dokumentų užsakovai – klientai, dokumentų išdavimo tarpininkai – įmonės ir institucijos – ambasados, išduodančios pasus ir vizas), yra šiam dalykui skiriamo laiko trumpinimas. Kadangi pasų ir vizų išdavimo procese pagrindinis darbo krūvis tenka dokumentų išdavimo tarpininkui, todėl tikslingiausia optimizuoti jo darbo eigą. To galima pasiekti automatizuojant sąsajas tarp užsakovo užpildyto klausimyno ir ambasadoms pateikiamos paraiškos dokumentams gauti.

*Tyrimo sritis* - elektroninės paslaugos, elektroninė komercija.

*Darbo objektas* – pasų ir vizų išdavimo informacinė sistema (PIVI IS).

*Tikslas:*

1. Sukurti pasų ir vizų išdavimo informacinę sistemą
2. Sukurti MVC architektūrą (angl. Framework), kurią naudojant bus galima kurti ir kitas informacines sistemas

Tikslui pasiekti keliami tokie *uždaviniai*:

1. Išanalizuoti kompiuterizavimo projekto aplinką
2. Išanalizuoti panašias IS sistemas
3. Išanalizuoti panašių modelių sistemas
4. Parinkti programavimo priemones
5. Sudaryti programų sistemos projektą
6. Testuoti, įvertinti, bandyti praktinį programos veikimą
7. Paruošti dokumentaciją
8. Įdiegti programą

# 1. PIVI IS, ARCHITEKTŪROS, ŠABLONŲ APDOROJIMO KLASIŲ IR TECHNOLOGIJŲ ANALIZĖ

## 1.1. Pasų ir vizų išdavimo internetinių svetainių palyginamoji analizė

Siekiant pasinaudoti panašių sistemų kūrėjų patirtimi buvo atlikta kelių projektų analizė, kuri pateikiama 1 lentelėje.

1 lentelė

Panašių projektų palyginimas

Savybės	GoToRussia[2]	PassportsAndVisas[3]	PassportVisaExpress[4]
<i>Platforma</i>	J2EE. Naudojama 8081 jungtis (angl. port). Vartotojams, kuriems uždarytos nestandartinės jungtys, negalės naudotis IS.	ASP. Reikalinga Windows OS. Nors jau yra realizuotas ASP palaikymas ir UNIX sistemose, tačiau nestabiliai veikiantis, mažiau funkcionalus	PHP.
<i>Vartotojų tipai</i>	<p><i>Paprastas</i> – nereikalaujama prisijungti prie sistemos. Užsisakant vizas būtina atsakyti į pasirinktos valstybės klausimus. Įvykdžius užsakymą pateikiama nuoroda į svetainę, kurią įvedus galima matyti užsakymo būseną ir administratoriaus pranešimus.</p> <p><i>Kolektyvinis</i> (corporate) – administratoriaus įregistruotas vartotojas (rašomas el. paštas administratoriui). Skiriasi paslaugų kainos, gali matyti chronologine tvarka išdėstytus užsakymus, apkomėjimus.</p> <p><i>Administratorius</i> – valdo IS, keičia parametrus (kainas, laikotarpius ir pan.), kontaktuoja su ambasados darbuotojais pasų ir vizų užsakymui, keičia užsakymų būsenas, rašo pranešimus užsakovams. Vizų paraiškos, pateikiamos ambasados darbuotojams, turi būti užpildytos tam tikrame formate (kiekvienai valstybei pateikiami šabloninės PDF tipo bylos). Administratorius užpildo PDF tipo bylas pagal kiekvieno užsakovo pateiktus atsakymus. Atspausdinus PDF bylą ji pristatoma ambasados darbuotojams. Pasai ir vizos užsakovui pristatomi pagal užsakyme nurodytą būdą.</p>	<p><i>Paprastas</i> – nereikia prisijungti prie sistemos. Užsisakant vizas reikia atsisiųsti tos valstybės klausimus (PDF formate) ir juos atsakius nusiųsti el. paštu. Įvykdžius užsakymą pateikiama nuoroda į svetainę, kurią įvedus galima matyti užsakymo būseną ir administratoriaus pranešimus. Neįvykdžius galutinio užsakymo, tačiau nusiuntus savo duomenis viza ar pasas yra išduodamas, bet negalima stebėti užsakymo būsenos.</p> <p><i>Kolektyvinis</i> (corporate) - administratoriaus įregistruotas vartotojas (rašomas el. paštas administratoriui). Skiriasi paslaugų kainos, gali matyti chronologine tvarka išdėstytus užsakymus, mokesčius.</p> <p><i>Tarpininkas</i> (affiliate) – vartotojas nesusijęs su pasų ir vizų užsakymais. Jis tik nukreipia vartotojus reklamuodamas pasų ir vizų išdavimo IS. Už kiekvieną nukreiptą ir įvykdytą užsakymą gaunama sutartą procentą nuo užsakymo sumos. Prisijungus prie sistemos matomi užsakovų duomenys, komisiniai pinigai.</p> <p><i>Administratorius</i> – bendrauja el. paštu su užsakovais, kontaktuoja su ambasados darbuotojais pasų ir vizų užsakymui. Ambasados darbuotojams pateikiami atspausdintos PDF tipo bylos, kuriuos užpildo ir atsiunčia el.</p>	<p><i>Paprastas</i> – būtina prisijungti prie sistemos. Įdiegtas krepšelis į kurį galima įtraukti pasus ir vizas. Įvykdant užsakymą krepšelyje esančios prekės matomos administratoriams, užsakymų sąrašas. Užsakovas gali matyti chronologine tvarka išdėstytus užsakymus ir pranešimus.</p> <p><i>Kolektyvinis</i> – panašus į paprastą vartotoją. Šias teises suteikia administratorius. Pirmą kartą prisijungus prie sistemos reikalaujama atsakyti į tam tikrą klausimų sąrašą. Įtraukiant vizas į krepšelį nereikalaujama atsakinėti į tos valstybės klausimus. Jei klausimai sutampa atsakymai imami iš prieš tai užpildytų klausimų. Skiriasi paslaugų kainos.</p> <p><i>Pareigybinis</i> (official) – panašus į kolektyvinį vartotoją. Skiriasi paslaugų kainos.</p> <p><i>Agentas</i> (agent) – administratorius, gavęs pasų ar vizų užsakymo paraišką faksu, sukuria šį klientą. Užsakovas prisijungęs prie sistemos gali matyti užsakymus, pranešimus.</p> <p><i>Vadybininkai</i> - valdo IS, keičia parametrus (kainas kiekvienam tipų vartotojui, laikotarpius, naujų valstybių ir klausimų įtraukimus ir kt.), kontaktuoja su ambasados darbuotojais pasų ir vizų užsakymui, keičia užsakymų būsenas, rašo pranešimus užsakovams. PDF tipo bylos sukuriamos automatiškai pagal</p>

Savybės	GoToRussia[2]	PassportsAndVisas[3]	PassportVisaExpress[4]
		paštu užsakovai.	užsakovų pateiktus atsakymus. <i>Administratorius</i> – tai vadybininko tipo vartotojas, turintis didžiausias teises (gali naudotis visu IS funkcionalumu). Gali kurti naujus vartotojus, suteikti jiems teises naudoti parinktus modulius.
<i>Pagalbininkas</i>	Realizuota garsinis (Macromedia Flash) ir tekstinis pagalbininkas.	Tekstinis	Tekstinis
<i>Paieškos indeksacija</i>	Kiekviename puslapyje naudojamos kitokios antraštės (headers) siekiant pagerinti pozicijas paieškos serverių rezultatuose. Antraštes keičia programuotojai.	Kiekviename puslapyje naudojamos kitokios antraštės. Antraštes keičia programuotojai.	Kiekviename puslapyje naudojamos kitokios antraštės. Antraštes keičia vadybininkai.
<i>Trūkumai</i>	Kadangi nėra realizuotas prekių krepšelis, galima užsisakyti tik vieną vizą ar pasą. PDF tipo bylas reikia užpildyti administratoriams.	Nėra prekių krepšelio. PDF tipo bylas reikia užpildyti užsakovams.	
<i>Kitos savybės</i>	Administratoriai gali naudotis priminimais, kuriuos valdo patys (nurodo dieną ir pranešimo tekstą).		Galima stebėti duomenų statistiką (įrašų, šiukšlių kiekį). Vadybininkai gali naudotis priminimais, kuriuos valdo patys (nurodo dieną ir pranešimo tekstą). Įtraukiant naujas visas, pasus, šalis, klausimus galima nustatyti testavimo tipą. Tada klausimai bus matomi tik vadybininkams.

## 1.2. Programavimo technologijų analizė

Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP, .NET ir Java technologijos. Išstudijavus literatūrą [5, 6, 7] buvo apibendrinti technologijų privalumai ir trūkumai, kurie pateikti 2 lentelėje.

2 lentelė

### Programavimo technologijų charakteristikos

	PHP	.NET	Java
Vidiniai tarnybinės stoties skriptai HTML generavimui	PHP	ASP	JSP
Palaikomos kalbos	PHP	Palaiko apie 25 skirtingas programavimo kalbas ( <i>JScript, VBScript</i> ir t. t.)	Java



	<b>PHP</b>	<b>.NET</b>	<b>Java</b>
Programavimo principai	Objektinis arba procedūrinis programavimas, taip pat dažnas abiejų principų mišinys	Priklausomai nuo programavimo kalbos, taikomi ir objektinio, ir procedūrinio programavimo principai	Objektinis programavimas
Priklausomybė nuo platformos	Pritaikyta beveik kiekvienai platformai	Geriausiai pritaikyta „Microsoft“ operacinėms sistemoms	Nepriklausoma
Funkcionalumas	Pasiekiamas per atskirus modulius (grafikos įrankius, susijungimo su DB tvarkyklės ir t. t.)	Pasiekiamas per esančius arba paties sukurtus COM objektus	Sąlygoja pakartotinio panaudojimo komponentai ir žymos
Atviras kodas	Taip	Ne, dauguma paketų priklauso „Microsoft“ korporacijai	Taip
Patikimumas	Dėl atviro kodo aptiktos klaidos greitai ir efektyviai pataisomos	Kadangi tai ne atviro kodo produktas, programų paketo klaidų negali pataisyti nepriklausomi programuotojai	Patikima
Sparta	Interpretuojama programavimo kalba, sparta nusileidžia daugumai kompiliuojamųjų	Dauguma paketo programavimo kalbų kompiliuojamos	Tik iš dalies kompiliuojama (programos kodas iš pradžių kompiliuojamas į baitinį kodą, o paskui virtuali mašina kodą interpretuoja)
Prisijungimas prie DB	MySQL	ADO	JDBC

3 lentelėje pateikiami kriterijai ir įvertinimai[8]. Įvertinimas 1 nurodo aukščiausią įvertinimą, 3 – žemiausią.

**3 lentelė**

<b>Palyginimo kriterijai</b>	<b>Įvertinimas</b>		
	<b>Java</b>	<b>PHP</b>	<b>.NET</b>
Paprastumas	1	2	3
Lankstumas	3	2	1
API lankstumas	1	3	2
Programavimo aplinkos	1	2	3
Paprastumas mokytis	2	1	3
Sparta	2	1	3

Palyginimo kriterijai	Įvertinimas		
	Java	PHP	.NET
Vidutinis įvertinimas	1,66	1,88	2,50

Remiantis vidutiniais įvertinimais, matosi, kad labiausiai vertėtų pasirinkti Java. Tačiau užsakovas yra pateikęs reikalavimą, kad būsima IS kiek galima greičiau funkcionuotų ant PIII 500MHz. tipo kompiuterio, todėl didžiausią dėmesį pasirenkant programavimo kalbą, skirsiu spartumui.

Kaip matosi iš 3 lentelėje pateiktų duomenų, sparta labiausiai pasižymi PHP programavimo kalba, nors ji yra tik interpretatorius. Šiai spragai užpildyti jau yra sukurta nemokama priemonė, kuri išeities tekstus paverčia į baitinį kodą. Todėl galima teigti, kad PHP išeities kodas yra dalinai interpretuojamas, o tai, pasak kūrėjų[9], paspartina veikimą iki 40%.

Pasirinkta duomenų bazės valdymo sistema – MySQL, kadangi ji labai gerai suderinama su PHP. Taip pat labai svarbus MySQL pasirinkimo kriterijus - programinės įrangos kaina, MySQL, kaip ir PHP, yra nemokama DBVS.

### 1.3. Kuriamos internetinės svetainės architektūra

#### 1.3.1. Paprastas internetinės svetainės kūrimas

Kuriant internetines svetaines, dažniausiai užtenka į tam tikrą dizaino formą įkelti reikalingą informaciją. Kaip pavyzdį galima naudoti tokią puslapio generavimo formą:

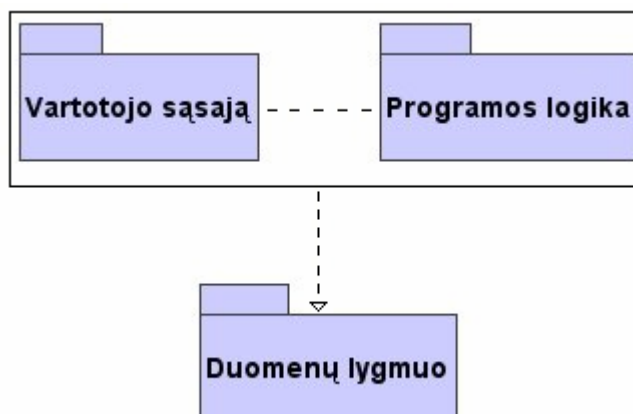
1. Įtraukti puslapio viršų (statinė dalis)
2. Įtraukti puslapio meniu (statinė dalis)
3. Įtraukti puslapio pagrindinę dalį (dinaminė dalis)

Pagrindinė duomenų kaita vyksta dinaminėje dalyje. Visumoje realizuojant šiuos uždavinius galima naudoti vieną bylą (skriptą) kuri išvestų statinę informaciją, o pagal pateiktus parametrus išvestų ir kintamą informaciją. Pvz., sukuriame index.php bylą, kurioje išdėstomas statinės dalies dizainas HTML kalba, o reikalingose bylos vietose panaudojame PHP kalbos struktūras dinaminės dalies vaizdavimui. Pvz., pasinaudojus GET metodu ir pagal perduotą kintamąjį "page\_name" įterpti nurodomą puslapį. Čia puslapis gali būti tiek statinis (HTML), tiek dinaminis (PHP).

Pagrindiniai trūkumai:

- Dizainas neatskirtas nuo logikos
- Beveik neįmanomas pakartotinis panaudojamumas
- Sistema mažai struktūrizuota, sunkiai plečiama.

Bendru atveju šią sistemą galima būtų suskirstyti į 2 lygmenis (1 pav.): viename vartotojo sąsaja ir programos logikos lygmuo, kitame - duomenų lygmuo. Duomenų lygmenį sudaro DBVS ir duomenų bazės schema.

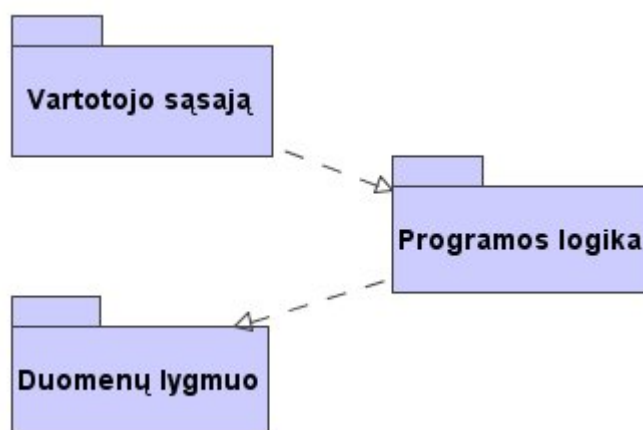


1 pav. Sistemos skirstymas į lygmenis

### 1.3.2. Internetinių svetainių kūrimas naudojant MVC modelį

Kadangi apsiribojant paprastais puslapiais, į kuriuos įdedami skriptai, dažnai veiklos logika pradeda kartotis, o sistemos palaikymas bei pratęsimas darosi sudėtingas, tai šią pasikartojančią logiką galima iškelti į atskirą komponentą ir sudaryti bendradarbiaujančių komponentų/modulių rinkinius. Tai pašalina perteklišumą, bet padidėja sprendimo sudėtingumas. Sudėtingumui valdyti įvairiose srityse, tame tarpe ir programavime, naudojami šablonai. Bendro naudojimo objektinio projektavimo šablonai aprašyti [10], kur pateikti 32 šablonai, bet jų yra ir daugiau. Pavyzdžiui, yra atskira šablonų grupė, taikoma interneto sistemų kūrimui.

Klasikinis interneto sistemoms taikomas modelis yra Modelis-Vaizdiklis-Valdiklis (angl. Model-View-Controller). Tai senas modelis, kuris atlaikė laiko testą ir atskyrė vaizdavimo logiką nuo veiklos logikos. Nors šis modelis nėra naujas, projektuotojui jo reikšmė yra labai didelė, kadangi sukurti moduliai gali būti panaudojami pakartotinai, o vaizdavimo logika atskirta nuo veiklos logikos. Programuotojai PHP dažnai naudoja greitam kūrimui, todėl neteikia didelės reikšmės programos architektūrai tobulinti. Tuo galima paaiškinti tą faktą, kad PHP kalboje mažai naudojami šablonai, paplitę Java, C, .NET programavimo technologijose. Tačiau ši kalba duoda ne mažesnes galimybes naudoti šablonus ir sukurti gerą programos architektūrą. Šio modelio skirstymas į lygmenis pavaizduotas 2 pav.



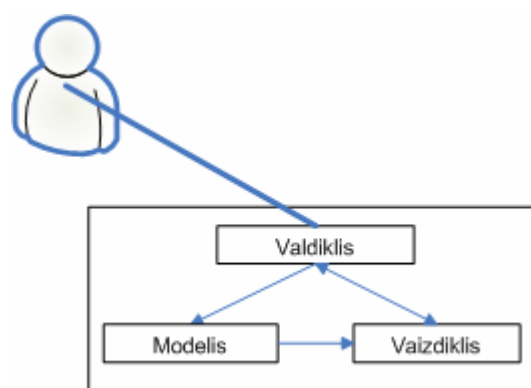
**2 pav.** MVC modelio skirstymas į lygmenis

Bendrajai prasme MVC, tai lyg programavimo modelis, kuris yra skirtas kurti, diegti ir vykdyti tinklo taikomąsias programas, kurios bendrauja per tinklą tokiais standartiniais protokolais, kaip SOAP, XML ir HTTP. Šie protokolai leidžia integruoti taikomąją programą su naujos kartos taikomosiomis programomis ir paslaugomis. Taip pat šis programinis modelis leidžia išspręsti didelio patikimumo ir plečiamumo taikomųjų programų diegimo ir valdymo problemas.

Pagrindiniai MVC modelio privalumai:

- 1) kodas pilnai atskirtas nuo dizaino (programinėje dalyje nevykdomas duomenų išvedimas)
- 2) lengva pakartotinai iškviesti modulį
- 3) galima autorizaciją kiekvienam komponento metodui
- 4) sistemai suteikia tvarkingumo ir struktūriškumo
- 5) pagerėja testavimas

MVC modelio objektų ryšiai pavaizduoti 3 pav.



**3 pav.** MVC modelio ryšiai

### **1.3.3. Panašūs architektūriniai sprendimai**

Siekiant pasinaudoti panašių sistemų kūrėjų patirtimi buvo atlikta kelių projektų analizė. Toliau pateikiama PHP-Nuke, php.MVC architektūrų (angl. Frameworks) apžvalga.

#### **1.3.3.1. PHP-Nuke sistemos architektūra**

Pasak PHP-Nuke kūrėjų, tai sistema skirta internetinių svetainių kūrimui ir plečiama bet kokiam panaudojimui, palaikanti 33 kalbas ir 9 DBVS [11].

Išorinį sistemos vaizdą sudaro trys stulpeliai, viršutinis kolontitulas (header) ir apatinis kolontitulas (footer). Kiekviena puslapį struktūriškai sudaro blokai ir moduliai. Blokai - nuorodos/apklauso/bet kuri kita informacija, kuri rodoma visuose svetainės puslapiuose, kairiame ir/ar dešiniame stulpelyje. Moduliai - sistemos "šerdis" - juos galima laikyti konkrečiais puslapiais, pvz. moduliai skirti naujienoms, paieškai, apklausai ir pan.

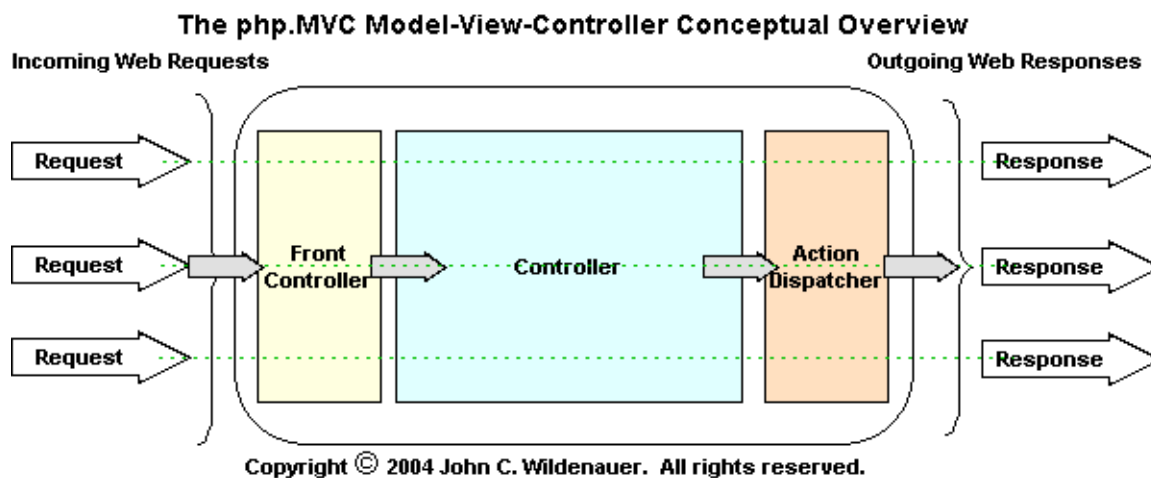
Visi modulių failai yra tvarkomi trimis pagrindiniais failais - index.php (pirmas puslapis), modules.php (įtraukia vykdomą modulį), admin.php (administratoriaus aplinka). Taip pat šakniniame sistemos kataloge yra failai, susiję su autorizacija, dizainu ir bendromis sistemos funkcijomis.

Kiekvienas modulis - tai kataloge "/modules/" esantis index.php failas. Pvz. norint išvesti modulį "Topics", reiktų kreiptis į <http://domeinas/modules.php?name=Topics>, tuomet modules.php iškvieštų /modules/Topics/index.php failą. Tiesiogiai šio failo peržiūrėti negalima, dėl saugumo, o taip pat ir dėl bendro dizaino palaikymo. Kiekvieno modulio kūrimui yra keletas taisyklių, kurių reikia laikytis, kad nesugadinti sistemos, bet praktiškai kiekvienas modulis tai yra atskiras failas, kurio pradžioje įtraukiamos bendros sistemos funkcijos, dizaino elementai, o duomenų išvedimui naudojamos funkcijos iš sistemos bibliotekų. Pvz. atlikus duomenų bazės užklausą atidarom lentelę (opentable()), nustatom jos pavadinimą, išvedam duomenis (kiekvieną eilutę panaudojant list() funkciją ir uždaram lentelę (closetable())).

Taigi, visa sistema yra paremta iš anksto paruoštomis funkcijomis, kurių pagalba informacija aplyejama dizainu. Pati sistema tobulinama apie 7 metus, todėl funkcijų yra gana daug. Realizuotas puslapių dizaino lengvas keitimas ("skins"). Yra daug kartu su sistema pateikiamų modulių, taigi ji yra paruošta "plug-n-play" principu - užtenka parsisiųsti, paleisti instaliacijos skriptus ir ja galima naudotis. Sistemos trūkumas yra tai, kad prastai dokumentuotas modulių kūrimas, paliekama daug laisvės programuotojui - t.y. sistema yra ganėtinai palaida ir nestruktūrizuota - mano nuomone modulių realizavimui geriau naudoti objektinį projektavimą, o ne paprastą funkcinį programavimą ir paprasčiausią failų įtraukimą (include), tam, kad juos įvykdyti.

### 1.3.3.2. php.MVC sistemos architektūra

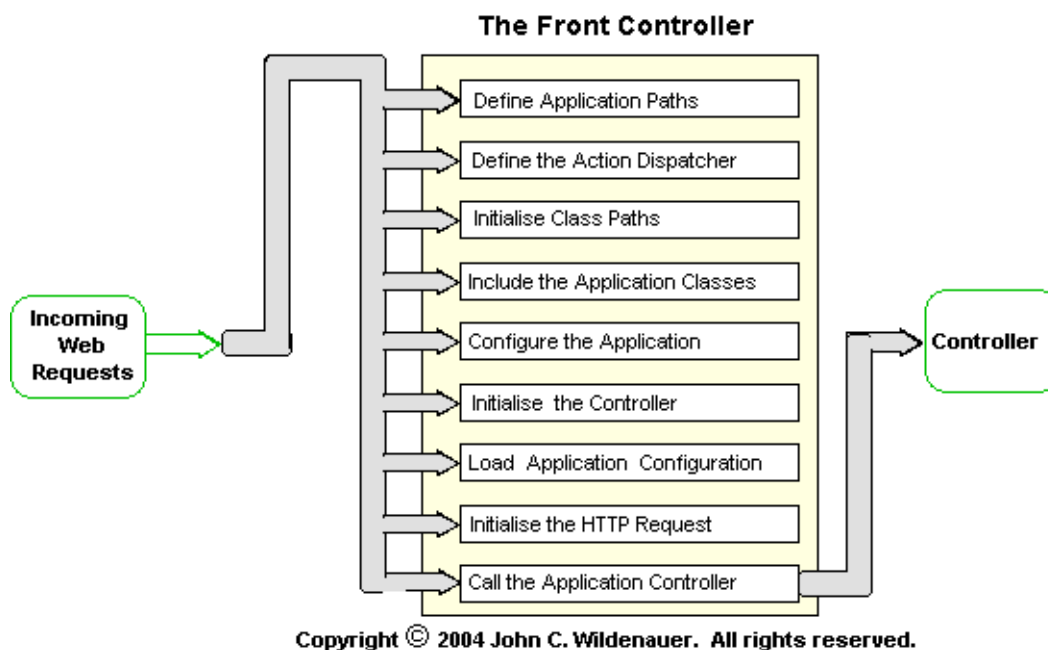
php.MVC tai Modelio-Valdiklio-Vaizdiklio architektūra (angl. Framework) skirta internetinių svetainių kūrimui[12]. Šios architektūros konceptualus modelis pavaizduotas 4 pav.



4 pav. php.MVC konceptualus modelis

Iš konceptualaus modelio matosi, kad php.MVC architektūra susideda iš trijų pagrindinių dalių: priekinio valdiklio (angl. Front controller), valdiklio (angl. Controller) ir veiksmų valdiklio (angl. Action Dispatcher).

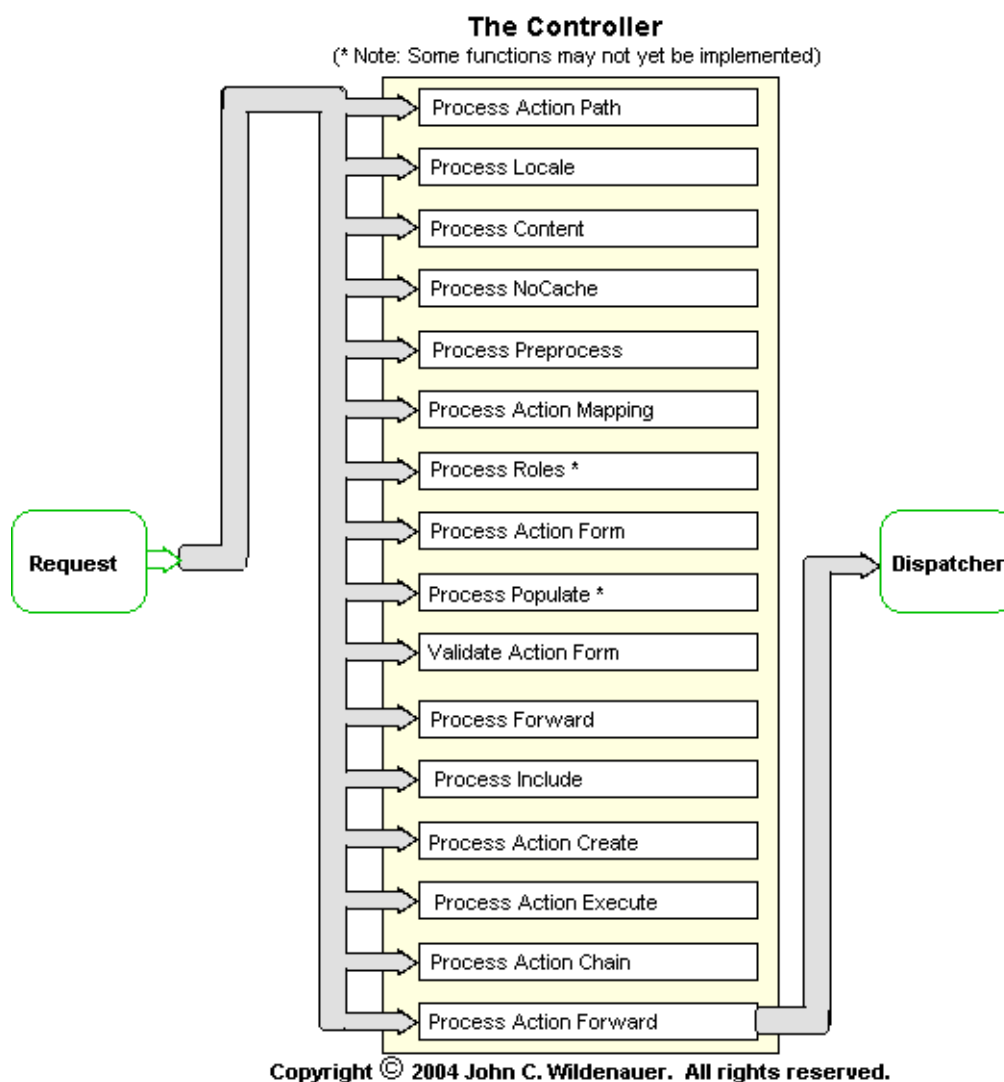
Priekinis valdiklis atsakingas už programos nustatymus, kai vykdoma užklausa. Šio valdiklio procesų seka pavaizduota 5 pav.



5 pav. php.MVC priekinis valdiklis

Priekinio valdiklio funkcijos: nustatyti programos kelius, priskirti veiksmų valdiklį, inicijuoti klasių kelius, įtraukti klases, sukonfigūruoti programą, inicijuoti valdiklį, užkrauti programos konfigūraciją, inicijuoti HTTP užklausa, iškviešti valdiklį.

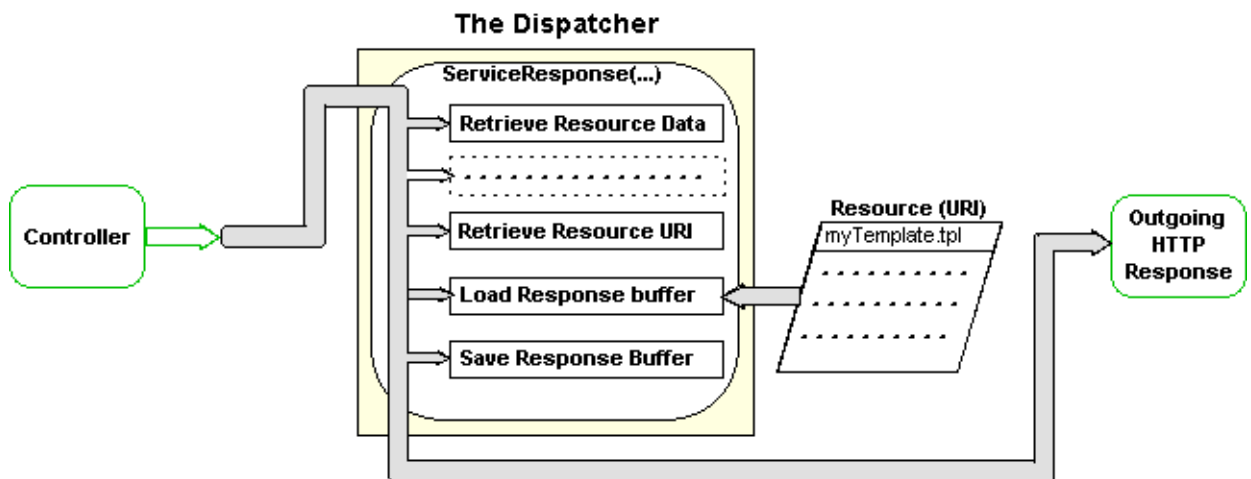
Priekiniam valdikliui baigus vykdymą, procesas persiduoda valdikliui. Valdiklio procesų seka pavaizduota 6 pav.



6 pav. php.MVC valdiklis

Valdiklio funkcijos: parinkti veiksmo kelią, parinkti kalbos parametrus, parinkti turinį, parinkti puslapio atsargos parametrus, parinkti vyksmo klasę, parinkti vyksmo surišimo kelius, parinkti roles, parinkti vyksmo formą, patikrinti formos duomenis, parinkti vyksmo persiuntimą, parinkti įtraukimo bylas, sukurti veiksmo klasę, įvykdyti veiksmo klasę, įvykdymo veiksmo klases nurodytas veiksmų eilėje, perduoti eigą veiksmų valdikliui.

Veiksmų valdiklis atsakingas už biznio logikos vykdymą ir vaizdavimą. Šio valdiklio procesų seka pavaizduota 7 pav.



Copyright © 2004 John C. Wildenauer. All rights reserved.

7 pav. php.MVC veiksmų valdiklis

Veiksmų valdiklis iškviečia biznio logikos metodą. Šis metodas pateikia duomenis, kurie apipavidalinami šabloniniu failu (angl. Template). Taip sugeneruojamas HTML dokumentas kuris perduodamas interneto naršyklei.

### 1.3.4. Kuriamos MVC architektūros privalumas

Kuriant architektūras (angl. Frameworks) naudojant PHP programavimo kalbą nėra kreipiamas dėmesys į metodų iškvietimą iš šablonų failų. Realizavus šį funkcionalumą galima paspartinti programavimo procesą, nes tai leistų lankčiau naudoti modulius – juos iškviešti būtų galima ne tik iš PHP kodo, bet ir iš šablonų bylų HTML kodo (naudojant papildomas žymas (angl. tags)).

Realizuojant metodų iškvietimą iš šablono failų bylų, pravartu sukurti ir internetinius kreipinius URL, kuriuose būtų galima nurodyti į kokį modulį kreipiamasi, kokius parametrus jam perduodame.

Galima naudoti tokį kreipinį:

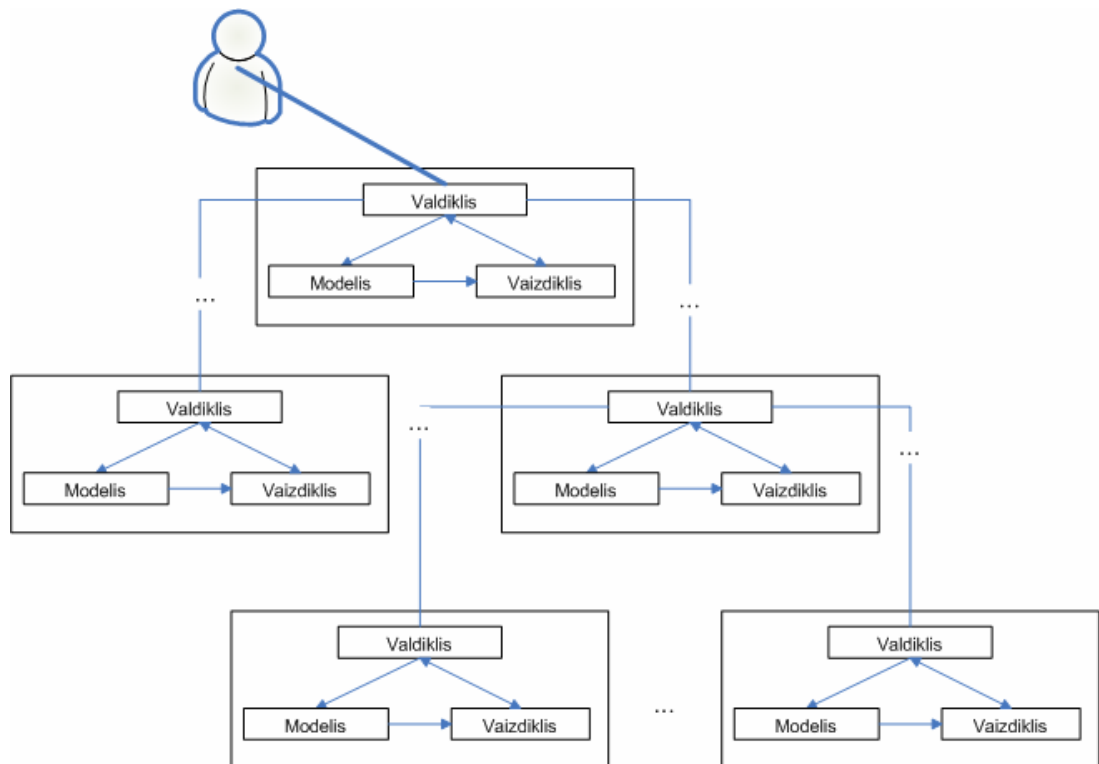
```
.../index.php?ModulioVardas1,ParametroVardas1.ParametroReikšmė1,
ParametroVardas2.ParametroReikšmė2;ModulioVardas2,ParametroVardas1.ParametroReikšm
ė1, ParametroVardas2.ParametroReikšmė2
```

O PHP programavimo kalba šį kreipinį galima paversti į toką struktūrą:

```
[ModulioVardas1] = array(ParametroVardas1 => ParametroReikšmė1
                        ParametroVardas2 => ParametroReikšmė2),
[ModulioVardas2] = array(ParametroVardas1 => ParametroReikšmė1
                        ParametroVardas2 => ParametroReikšmė2)
```

Realizavus šį funkcionalumą MVC architektūros objektų ryšius galima pavaizduoti taip, 8 pav.





8 pav. MVC modelio ryšiai, kai galima kreiptis į valdiklį iš šablonų bylų

Taikymo pavyzdys:

Bylos vardas	<i>showLoggedUserInfo.html</i>
Bylos turinys	Prisijungusio vartotojo duomenys: {module:Users,action.showUserInfo,userId:{currentUserId}}
Atlikti veiksmai	Neatlikta jokių veiksmų

Bylos vardas	<i>showUserInfo.html</i>
Bylos turinys	Vardas: {userFirstName} Pavardė: {userLastName}
Atlikti veiksmai	Neatlikta jokių veiksmų

Bylos vardas	<i>Users.mod.php</i>
Klasės vardas	Users
Klasės turinys	<pre> Class Users {     ...     Function showUserInfo() {         \$currentUserId = \$this-&gt;getLoggedUserId();         \$mData = array('currentUserId' =&gt; \$currentUserId);         return \$mData;     }      Function showLoggedUserInfo() {         \$userId = \$this-&gt;getParam('userId');         \$userInfo = \$this-&gt;getUserInfoById(\$userId);         \$mData = array(             'userFirstName' =&gt; \$userInfo['firstName'],             'userLastName' =&gt; \$userInfo['lastName']         );         return \$mData;     } } </pre>

Vartotojui kreipiantis nuoroda `index.php?Users,action.showLoggedInUserInfo` MVC modelio valdiklis iškviečia *Users* klasėje esantį metodą *showLoggedInUserInfo()*. Šis metodas grąžina prisijungusio vartoto *id*, o vaizdiklis pakeičia *showLoggedInUserInfo.html* šablone esantį elementą *{currentUserId}* į 5. Bylos *showLoggedInUserInfo.html* turinys keičiasi į:

```
Prisijungusio vartotojo duomenys:
{module:Users,action.showUserInfo,userId.5}
```

Valdiklis nagrinėja bylos *showLoggedInUserInfo.html* turinį ir tikrina ar nereikia iškviešti kito valdiklio. Randant iškvietimo žymę: *'module:'* iškviečia valdiklį ir perduoda parametrus. Valdiklis pagal perduotus parametrus kreipiasi į *Users* klasėje esantį metodą *showUserInfo()*. Šis metodas grąžina vartotojo duomenis (vardą – *firstName* ir pavardę – *lastName*), o vaizdiklis pakeičia *showUserInfo.html* šablone esančius elementus. Šios bylos turinys keičiasi į:

```
Vardas: Vardas
Pavardė: Pavardaitis
```

Atlikus šiuos veiksmus, valdymas perduodamas pirminiam valdikliui, kuris pakeičia Bylos *showLoggedInUserInfo.html* turinį:

```
Prisijungusio vartotojo duomenys:
Vardas: Vardas
Pavardė: Pavardaitis
```

### 1.3.5. Šablonų apdorojimo klasių analizė

Šablonų apdorojimo klasė glaudžiai susijusi su MVC modelio taikymu, nes ji leidžia atskirti programinį kodą nuo dizaino. Todėl reikia pasirinkti optimalią šablonų (templates) apdorojimo klasę. Atsižvelgdamas į testų duomenis pasirinkau Easy Template System šablonų apdorojimo klasę, kadangi jos vidutiniai spartos duomenys geriausi [13].

Šablonų apdorojimo klasių palyginimai atlikti su Pentium IV, 2.6GHz., 1GB RAM tipo kompiuteriu ir pavaizduoti 4,5,6,7 lentelėse.

**4 lentelė**

**Šabloninių failų elementų keitimo laikas, kai naudojama 20 kintamųjų ir 10 iteracijų**

Šablonų apdorojimo klasės pavadinimas	Laikas	Vidutinis iteracijos įvykdymo laikas
Easy Template System	1ms	0.0013762
phemplate	2ms	0.0016225
bugi	2ms	0.0021003
heyes class	2ms	0.0022084
Smarty	2ms	0.0024240
BrainyTemplate	3ms	0.0025108

<b>Šablonų apdorojimo klasės pavadinimas</b>	<b>Laikas</b>	<b>Vidutinis iteracijos įvykdymo laikas</b>
XTemplate	3ms	0.0026261
phplib tpl	3ms	0.0031540
TemplatePower v3	6ms	0.0062897
vTemplate	7ms	0.0071323

5 lentelė

**Šabloninių failų elementų keitimo laikas, kai naudojami 5 kintamieji ir 10 iteracijų**

<b>Šablonų apdorojimo klasės pavadinimas</b>	<b>Laikas</b>	<b>Vidutinis iteracijos įvykdymo laikas</b>
bugi	1ms	0.0013688
heyes class	2ms	0.0015025
phemplate	2ms	0.0020379
Easy Template System	2ms	0.0021019
BrainyTemplate	2ms	0.0023741
XTemplate	3ms	0.0027502
phplib tpl	3ms	0.0032715
Smarty	3ms	0.0033965
TemplatePower v3	6ms	0.0062556
vTemplate	7ms	0.0067409

6 lentelė

**Šabloninių failų elementų keitimo laikas, kai naudojama 100 kintamųjų ir 5 iteracijos**

<b>Šablonų apdorojimo klasės pavadinimas</b>	<b>Laikas</b>	<b>Vidutinis iteracijos įvykdymo laikas</b>
phemplate	1ms	0.0021070
BrainyTemplate	1ms	0.0024312
Easy Template System	1ms	0.0029406
XTemplate	2ms	0.0030840
Smarty	2ms	0.0041580

Šablonų apdorojimo klasės pavadinimas	Laikas	Vidutinis iteracijos įvykdymo laikas
heyes class	3ms	0.0051260
bugi	3ms	0.0054288
TemplatePower v3	4ms	0.0079734
vTemplate	4ms	0.0083752
phplib tpl	5ms	0.0096274

7 lentelė

Šabloninių failų ciklų vykdymo laikas, kai teksto dydis 10kb., 100 ciklų, 10 iteracijų

Šablonų apdorojimo klasės pavadinimas	Laikas	Vidutinis iteracijos įvykdymo laikas
Easy Template System	1ms	0.0007465
Smarty	3ms	0.0032585
vTemplate	10ms	0.0098080
Phemplate	10ms	0.0099458
Bugi	14ms	0.0140821
BrainyTemplate	24ms	0.0238421
TemplatePower v3	29ms	0.0289323
phplib tpl	81ms	0.0811702
XTemplate	121ms	0.1213947
heyes class	139ms	0.1392510

#### 1.4. Analizės išvados

1. Renkantis kūrimo platformą projekto realizavimui, svarstyta keletas alternatyvų: PHP, .NET ir Java technologijos. Pasirinkta PHP programavimo kalba, kadangi užsakovo pagrindiniai kriterijai – PIVI IS veikimo sparta ir mažas resursų naudojimas.
2. PIVI IS realizacijai bus kuriama MVC architektūra (ang. Framework), galinti iš šablonų bylų kreiptis į valdiklį.
3. Atlikus šablonų (templates) apdorojimų klasių analizę pasirinkta Easy Template System klasė, kuri, atlikus palyginimus su kitomis klasėmis, parodė geriausias spartos rezultatus.

## 2. PIVI IS IR MVC ARCHITEKTŪRA

Šiame skyriuje pateikiami esminiai PIVI IS ir MVC architektūros projektavimo etapai.

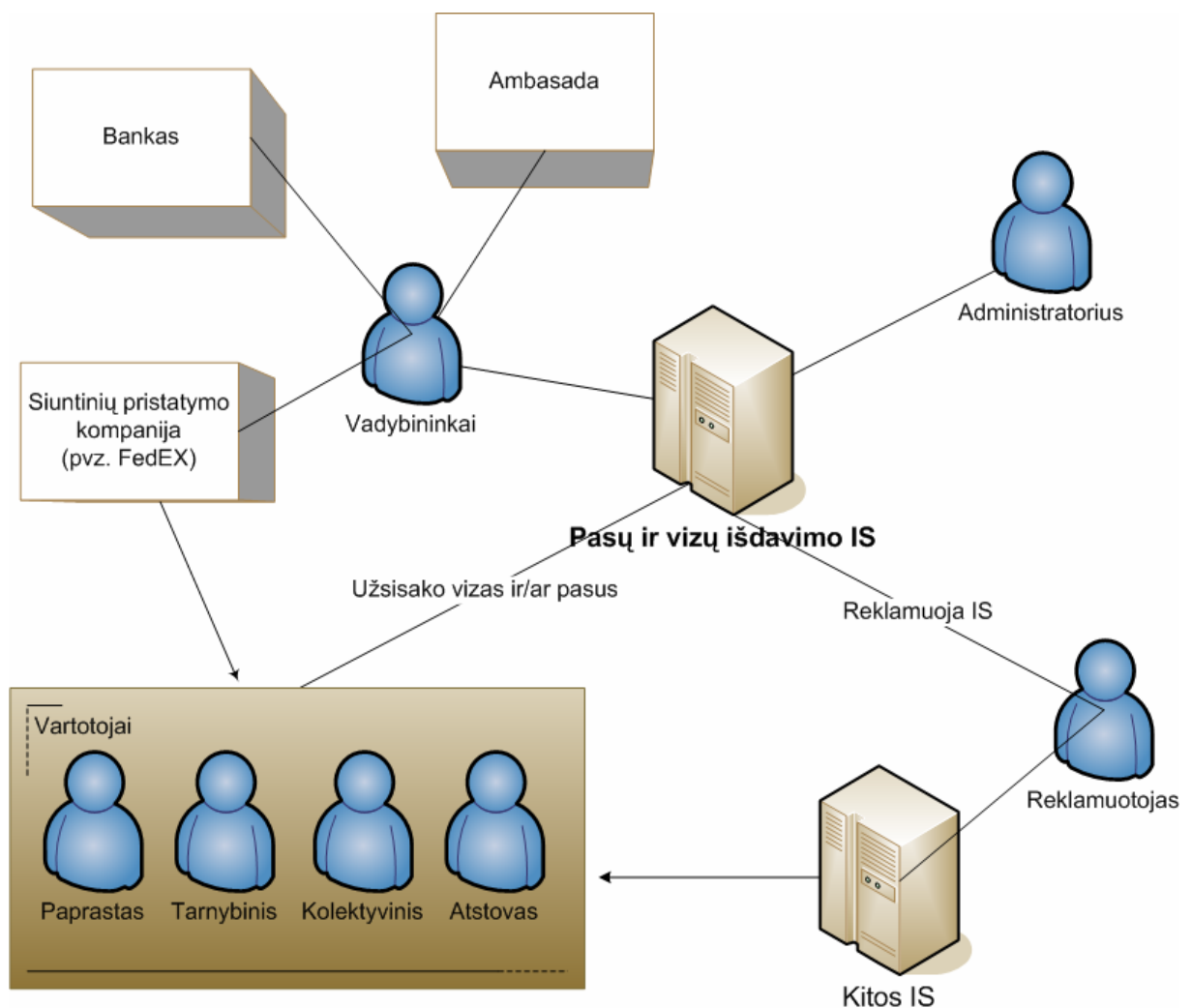
### 2.1. PIVI IS veiklos sfera

Kaip ir daugelyje veiklos sferų, pagrindinis dėmesys skiriamas vartotojui (užsakovui). Šioje IS bus realizuoti 4 užsakovų tipai: paprastas, pareigybinis, kolektyvinis ir agentas. Visi jie gali užsisakyti vizas ar pasus, tačiau reikalavimai išdavimui ir kainos gali skirtis.

Vartotojai užsakinėja vizas ar pasus naudojant pasų ir vizų išdavimo IS.

Vadybininkai, kurie gali būti išsidėstę įvairiuose miestuose, stebi užsakymus. Priklausomai nuo užsakymo tipo, vadybininkai kontaktuoja su tos šalies ambasadamis. Iškilus neaiškumams, vadybininkai skambina užsakovui, rašo el. paštu arba palieka žinutę PIVI IS. Vadybininkui gavus vizas/pasus iš ambasados, išskaičiuojami pinigai iš banko, o pasai ir vizos grąžinami siuntinių pristatymo kompanijos pagalba (pvz. FedEx).

Grafiškai veiklos kontekstas pavaizduotas 9 pav.



9 pav. Veiklos kontekstas

## 2.2. Vystymo etapai

Vystymo etapams reikalinga techninė ir programinė įranga pavaizduota 8 lentelėje.

8 lentelė

Vystymo etapai ir jiems atlikti reikalinga techninė bei programinė įranga

Eil. Nr.	Etapo pavadinimas	Darbams atlikti reikalinga	
		Techninė įranga	Programinė įranga
1.	Reikalavimų specifikavimas	kompiuteris	MS Word
2.	Architektūros specifikavimas	kompiuteris	MS Word, MS Visio, Magic Draw UML
3.	Duomenų bazės schemas sudarymas	kompiuteris, tarnybinė stotis	MySQL DBVS
4.	Sistemos administratoriaus sąsajos realizacija	kompiuteris, tarnybinė stotis	Macromedia Dreamweaver, Zend Development Enviroment. Tarnybinėje stotyje esanti programinė įranga, MySQL DBVS, Apache, PHP
5.	Vartotojo sąsajos realizacija	kompiuteris, tarnybinė stotis	Macromedia Dreamweaver, Zend Development Enviroment. Tarnybinėje stotyje esanti programinė įranga, MySQL DBVS, Apache, PHP
6.	Sistemos testavimas	kompiuteris, tarnybinė stotis	MS Internet Explorer
7.	Klaidų taisymas	kompiuteris, tarnybinė stotis	Macromedia Dreamweaver, Zend Development Enviroment, MySQL DBVS
8.	Sistemos įdiegimas bandomajam naudojimui	kompiuteris, tarnybinė stotis	MySQL DBVS, Apache, PHP
9.	Klaidų taisymas	kompiuteris, tarnybinė stotis	Macromedia Dreamweaver, Zend Development Enviroment. Tarnybinėje stotyje esanti programinė įranga, MySQL DBVS, Apache
10.	Sistemos pristatymas užsakovui	kompiuteris, tarnybinė stotis	Microsoft PowerPoint
11.	Vartotojo dokumentacijos paruošimas	kompiuteris	Microsoft Word
12.	Sistemos įdiegimas	kompiuteris, tarnybinė stotis	MySQL DBVS, Apache, PHP

## 2.3. Pagrindiniai IS ir MVC architektūros kūrimo etapai

Kadangi kuriant internetinę svetainę realizuosime tiek MVC architektūrą, tiek taikomojo uždavinio logiką, tai sistemos kūrimas susidės iš dviejų etapų: MVC architektūros kūrimo etapo ir internetinės svetainės kūrimo etapo. Detalūs MVC architektūros ir PIVI IS kūrimo etapai pavaizduoti 9,10 lentelėse.

9 lentelė

MVC architektūros kūrimo etapai

Darbo pavadinimas	Atliekami darbai	Rezultatas
Sistemos analizė ir projektavimas	<ul style="list-style-type: none"> <li>Sistemos logikos analizė ir projektavimas</li> <li>Sistemos komponentų analizė ir projektavimas</li> </ul>	<ul style="list-style-type: none"> <li>Projekto dokumentacija</li> </ul>
Pagrindinės struktūros (framework) kūrimas	<ul style="list-style-type: none"> <li>MVC modelis</li> <li>Klaidų apdorojimo klasė</li> </ul>	<ul style="list-style-type: none"> <li>Realizuota sistema, leidžianti vykdyti skirtingus modulius, užtikrinanti modulių tarpusavio bendradarbiavimą.</li> <li>Leidžiama naudoti šablonus (templates)</li> <li>Užtikrinamas klaidų surinkimas/perdavimas iš skirtingų modulių</li> </ul>
Parametrų failų lygyje, apdorojimo funkcijų kūrimas	<ul style="list-style-type: none"> <li>Parametrų failai</li> <li>Parametrų failų analizatorius</li> </ul>	<ul style="list-style-type: none"> <li>Parametrai skaitomi iš tekstinių failų</li> <li>Parametrai analizuojami (parsing) ir suvedami į tam tikrą struktūrą</li> </ul> <p>Pvz:</p> <pre>[first_section]   one = 1 [dirs]   path = usr/ [dirs2]   path = {dirs:path}local // → usr/local</pre>
DB tvarkymo modulio kūrimas	<ul style="list-style-type: none"> <li>Klasė skirta atlikti DB lygio funkcijoms</li> <li>Klasė skirta atlikti dažnai naudojamiems veiksams su DB</li> </ul>	<ul style="list-style-type: none"> <li>Užklausų siuntimas į DB</li> <li>Duomenų saugojimas</li> <li>Transakcijų saugomų garantavimas, t.y. lentelių rakinimas (locking)</li> <li>Klaidų apdorojimas</li> <li>Užtikrina nesudėtingą kitos DBVS naudojimą sistemoje (t.y. nesudėtinga kodo adaptacija prie kitos DBVS)</li> </ul>
Sesijų tvarkymo modulio kūrimas	<ul style="list-style-type: none"> <li>Sesijų laikymas DB lygyje</li> <li>Sesijos kintamųjų laikymas DB lygyje</li> <li>Sesijos duomenų tvarkymas</li> </ul>	<ul style="list-style-type: none"> <li>Kiekvienam prisijungusiam vartotojui sukuriama nauja sesija</li> <li>Kiekvienai sesijai saugomi tik jai skirti kintamieji (DB lygyje)</li> <li>Sesijos kintamųjų skaitymas/rašymas</li> <li>Pasenusios neregistruotos (t.y. vartotojas neautorizuotas) sesijos trinamos iš DB, jei praėjo laikas T1</li> <li>Pasenusios registruotos sesijos trinamos iš DB, jei praėjo laikas T2</li> </ul>
Funkcijų, klasių kūrimas	<ul style="list-style-type: none"> <li>JavaScript funkcijų paketas</li> <li>Duomenų kontrolės klasė</li> <li>Parametrų analizės klasė</li> <li>El. laiškų siuntimo klasė</li> <li>PDF bylų tvarkymo klasė</li> </ul>	<ul style="list-style-type: none"> <li>JavaScript funkcijos - pateikti užklausas vartotojui, jas apdoroti, atidaryti langus</li> <li>Atliekama duomenų kontrolė – konvertuojami duomenys, kad nebūtų galima įvykdyti SQL injection pažeidimų</li> <li>Parametrai (failų, URL) suvedami į tam tikrą struktūrą. (pvz.: a,b.1,c.2;c,d.1  <pre>array([a] =&gt; array(   ([b] =&gt; 1, [c] =&gt; 2)), [d] =&gt; 1)</pre> </li> </ul>

Darbo pavadinimas	Atliekami darbai	Rezultatas
		<ul style="list-style-type: none"> <li>• Siunčiamas el. laiškas, atsiradus klaidai</li> <li>• PDF bylų kūrimas</li> </ul>

10 lentelė

**PIVI IS kūrimo etapai**

Darbo pavadinimas	Atliekami darbai	Rezultatas
Sistemos analizė ir projektavimas	<ul style="list-style-type: none"> <li>• Sistemos komponentų projektavimas</li> <li>• Sistemos logikos projektavimas</li> <li>• Sistemos architektūros kūrimas</li> </ul>	<ul style="list-style-type: none"> <li>• Projekto dokumentacija</li> </ul>
Administruojančių vartotojų sąrašų kūrimas	<ul style="list-style-type: none"> <li>• Vartotojų sąrašai, jų tvarkymas</li> <li>• Teisių sistemos architektūros sukūrimas</li> </ul>	<ul style="list-style-type: none"> <li>• Vartotojų sąrašas – atvaizdavimas ekrane</li> <li>• Naujo vartotojo kūrimas</li> <li>• Vartotojų trynimas</li> <li>• Vartotojo redagavimas</li> <li>• Rikiavimas pagal ID</li> <li>• Vartotojo redagavimo langas</li> <li>• Paskutinio vartotojo prisijungimo registravimas (laikas, IP)</li> <li>• Galimybė saugoti personalius kintamuosius ilgam laikui</li> <li>• Prisijungusių vartotojų tvarkymas (atjungimas, teisių perkrovimas)</li> </ul>
Vartotojų sąrašų kūrimas:	<ul style="list-style-type: none"> <li>• Vartotojų sąrašai, jų tvarkymas</li> </ul>	<ul style="list-style-type: none"> <li>• Vartotojų sąrašas – atvaizdavimas ekrane</li> <li>• Naujo vartotojo kūrimas</li> <li>• Vartotojų trynimas</li> <li>• Vartotojo redagavimas</li> <li>• Rikiavimas pagal ID</li> <li>• Vartotojo redagavimo langas</li> <li>• Paskutinio vartotojo prisijungimo registravimas</li> <li>• Slaptažodžio siuntimas el. paštu</li> </ul>
Autentifikacijos ir autorizacijos sistemos kūrimas:	<ul style="list-style-type: none"> <li>• Autentifikacija</li> <li>• Autorizacija</li> </ul>	<ul style="list-style-type: none"> <li>• Prisijungimas registruotam vartotojui prie sistemos <ul style="list-style-type: none"> <li>○ Vadybininkui</li> <li>○ Paprastam vartotojui</li> <li>○ Kolektyviniam (corporate) vartotojui</li> <li>○ Pareigybiniam (official) vartotojui</li> <li>○ Agentui (agent)</li> </ul> </li> <li>• Prisijungimų registravimas DB</li> <li>• Teisės nuskaitomas iš DB</li> <li>• Teisių algoritmas (ką vartotojas gali naudoti/atlikti)</li> <li>• Funkcijos gauti atsakymą, ar gali atlikti vartotojas vykdyti veiksmą ar ne</li> </ul>
Klausimyno kūrimas	<ul style="list-style-type: none"> <li>• Klausimų sąrašas</li> <li>• Atskiras modulis – panaudojamas keliose vietose</li> <li>• Klausimų atvaizdavimo analizatorius (parser)</li> </ul>	<ul style="list-style-type: none"> <li>• Naujo klausimo sukūrimas</li> <li>• Klausimo redagavimas</li> <li>• Klausimo trynimas: jei naudojamas – žymima kaip D, kitu atveju trinamas iš DB</li> <li>• Klausimų sąrašo filtras:</li> <li>• Rikiavimas</li> <li>• Klausimų redagavimo langas</li> <li>• Funkcijos klausimų nuskaitymui iš DB</li> <li>• Funkcijos klausimo HTML teksto formavimui</li> </ul>
Prekių krepšelio kūrimas	<ul style="list-style-type: none"> <li>• Prekių krepšelis</li> <li>• Apmokėjimas</li> </ul>	<ul style="list-style-type: none"> <li>• Galima įdėti naują prekę į krepšelį</li> <li>• Prekę galima ištrinti</li> <li>• Skaičiuojama apmokėjimo suma</li> <li>• Prekių krepšelis aktyvus tik veikiant konkrečiai sesijai</li> </ul>



## 2.4. Pagrindinės IS funkcijos

Sukurta svetainė turės užtikrinti šias funkcijas:

- PIVI IS užsakovams:
  - 1) Pasų ir vizų užsakymų registravimas ir kaupimas
  - 2) Taikyti pasų ir vizų išdavimo kainas pagal vartotojo tipą
  - 3) Leisti stebėti pasų ir vizų išdavimo eigą (eigą aprašo vadybininkas)
- PIVI IS vadybininkams:
  - 1) IS administravimas
  - 2) Užsakymų pateikimas PDF formate
  - 3) Pranešimų pateikimas užsakovams
  - 4) Priminimų sąrašų sudarymas
  - 5) HTML antraščių (angl. headers) keitimas

## 2.5. IS keliami reikalavimai

Šiame poskyryje išvardinti sistemai keliami funkciniai ir nefunkciniai reikalavimai.

### 2.5.1. Funkciniai reikalavimai

- Užsakymų vykdymas:
  - Sistemoje turi būti numatyta galimybė įdėti į krepšelį pasus ir vizas
  - Taikyti pasų ir vizų išdavimo kaunas pagal vartotojo tipą
  - Galimybė stebėti vykdomą užsakymo eigą
- Vartotojų administravimas:
  - Sistemoje turi būti numatyta galimybė sukurti/redaguoti/trinti vartotojus
  - Sistemoje turi būti numatyta galimybė kiekvieno tipo vartotojams suteikti priėjimo prie pasirinktų modulių teises
  - Sistemoje turi būti numatyta galimybė priskirti vizą ar pasą atstovo (angl. agent) tipo vartotojams
- Pasų ir vizų tipų kūrimas:
  - Sistemoje turi būti numatyta galimybė sukurti/redaguoti/trinti pasų ir vizų tipus
  - Sudaryti kainas pagal užsakovų tipą (paprastam vartotojui, kolektyviniam ir kt.)
  - Leisti sukurti/redaguoti/trinti pasų ir vizų išdavimo reikalavimus
- Pagalbos dokumentų peržiūra:
  - Galimybė vartotojui peržiūrėti paruoštus pagalbos dokumentus

- Sistemoje turi būti numatyta galimybė vadybininkams keisti HTML puslapių antraštes (angl. headers)
- Užsakymų valdymas:
  - redaguoti, trinti užsakymus
  - redaguoti atsakymus, šalinti vizas iš užsakymo
  - peržiūrėti ir atspausdinti vizų užsakymus PDF formate
  - šalinti pasus iš užsakymo
  - sudaryti priminimo sąrašus

### 2.5.2. Nefunkciniai reikalavimai

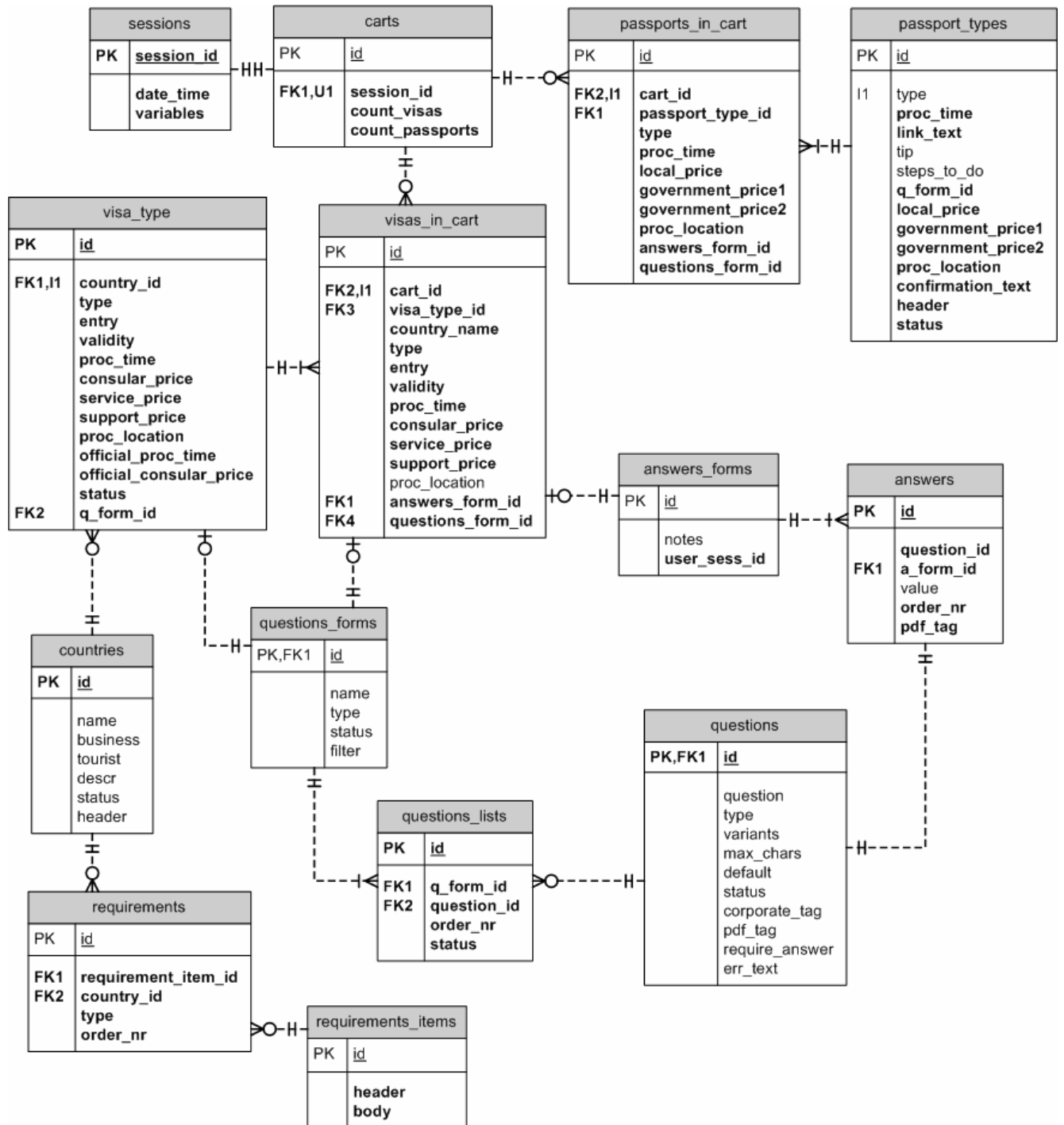
Kuriamai sistemai taip pat yra keliama visa eilė ir nefunkcinių reikalavimų:

- Reikalavimai sistemos išvaizdai:
  - Vartotojo sąsaja turi būti neperkrauta, paprasto dizaino, veiksmų pasirinkimo meniu aiškus ir tinkamai išdėstytas
- Reikalavimai panaudojamumui:
  - Sistema turi būti paprasta, naudotis vartotojui susipažinusiam su naudojimo instrukcija
  - Sistemos sąsaja turi būti vienos kalbos: anglų
- Reikalavimai pasiekiamumui:
  - Nutolę vartotojai turi turėti galimybę naudotis sistema bet kuriuo metu
- Reikalavimai vykdymo charakteristikoms:
  - Vartotojo komandos vykdomos realiaame laike, t.y. vartotojas neturi ilgai laukti, kol bus pradėta apdoroti jo komanda
  - Vartotojas bet kuriuo paros metu turi turėti galimybę prisijungti prie sistemos
- Reikalavimai veikimo sąlygoms:
  - Vartotojas turi galėti naudotis sistema naudodamasis *Microsoft Internet Explorer 5.0* ar naujesnės versijos naršykle;
  - Sistema turi veikti greitai ir stabiliai, kai serverio dalies kompiuteris PIII-500 MHz.
- Reikalavimai saugumui:
  - Vartotojo slaptažodis neturi būti matomas kitiems asmenims
  - Vartotojui įvedant duomenis naudoti saugų HTTPS protokolą
  - Sistema neturi perduoti asmeninių vartotojo duomenų su sistemos palaikymų nesusijusiems asmenims

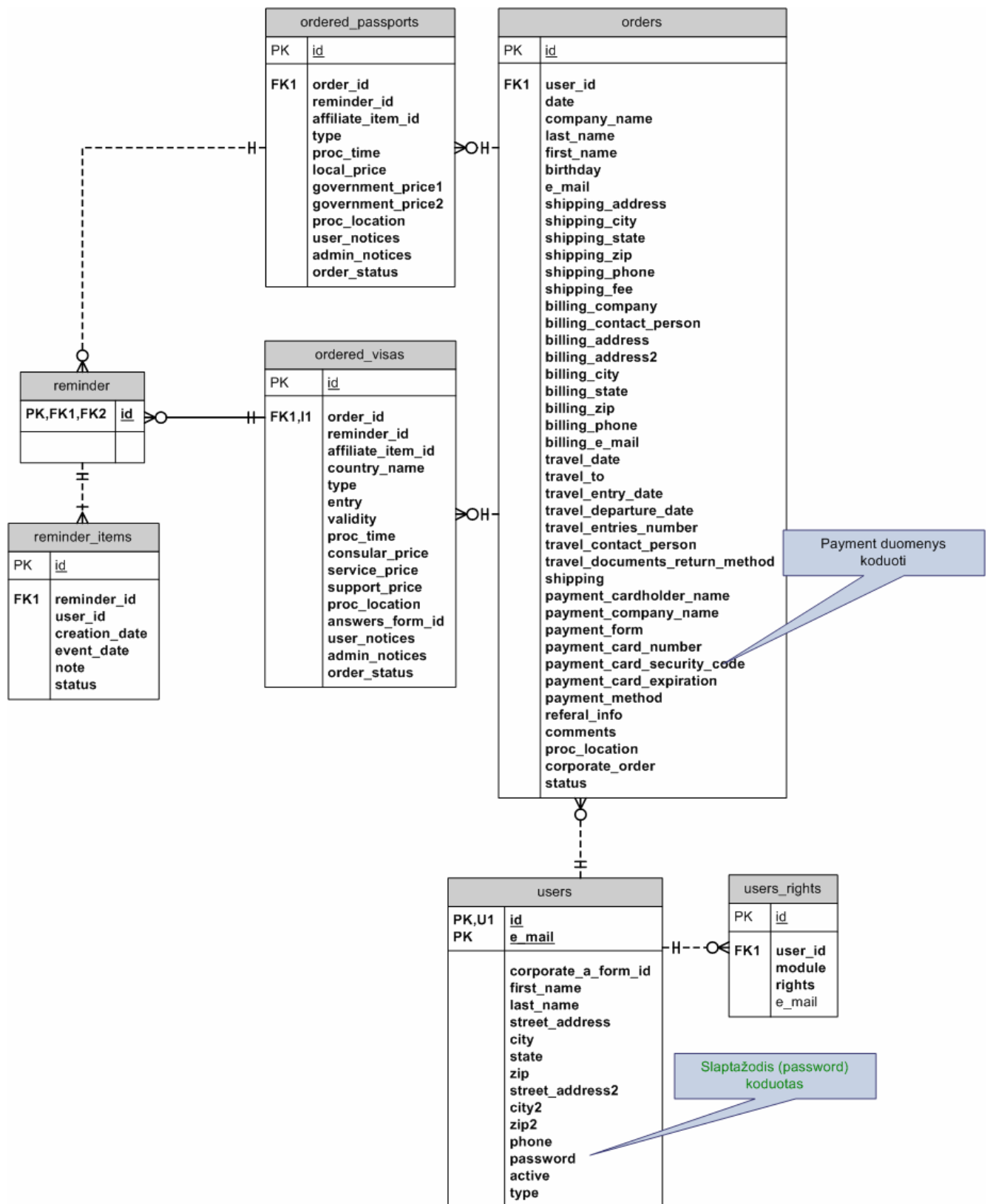
- Prie sistemoje esančių užsakymų duomenų, galimybę naudotis turi turėti tik duomenų savininkas ir vadybininkai
- Kultūriniai – politiniai reikalavimai:
  - Sistemoje negali būti naudojamos frazės, galinčios įžeisti kitos tautybės ar religinių įsitikinimų asmenis
- Teisiniai reikalavimai:
  - Sistema turi tenkinti autorinių ir gretutinių teisių įstatymus ir kitus teisės aktus, t.y. sistema neturi kopijuoti ar kitaip panaudoti kitų autorių sukurtus produktus be jų sutikimo. Taip pat sistema ar jos dalys neturi būti kopijuojama ar kitaip panaudojama be sistemos autoriaus sutikimo
  - Sistema turi tenkinti asmens duomenų apsaugos įstatymą

## 2.6. Duomenų bazės schema

Duomenų bazės diagrama pavaizduota 10,11 pav.



10 pav. Duomenų bazės diagrama



11 pav. Duomenų bazės diagramos tęsinys

Duomenų bazės lentelių (angl. tables) aprašymai pateikti 11 lentelėje.

11 lentelė

**DB lentelių aprašymas**

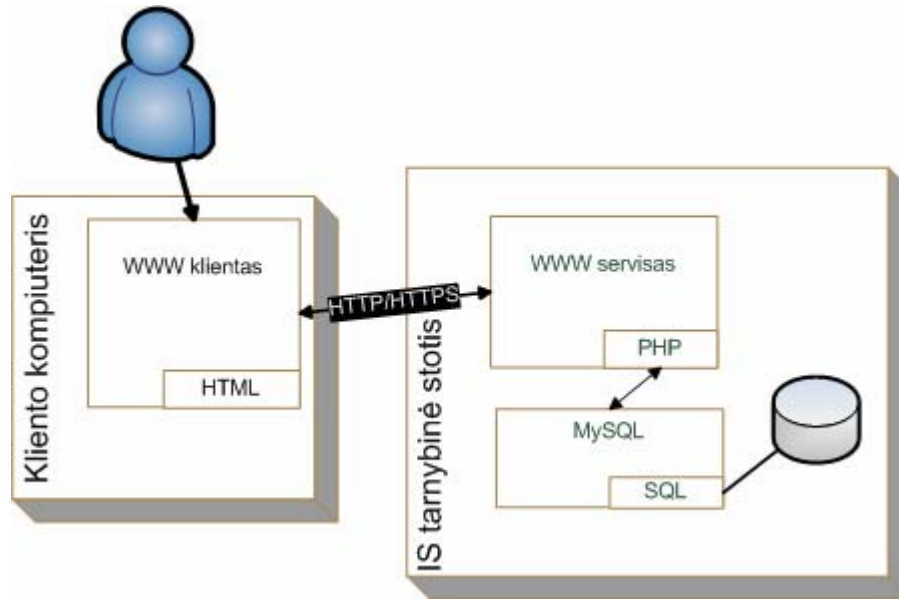
<b>Lentelė</b>	<b>Trumpas aprašas</b>
<i>sessions</i>	Saugojami sesijos duomenys
<i>carts</i>	Krepšelis, saugojami duomenys apie užsakomus pasus ir vizas
<i>passports in cart</i>	Duomenys apie pasus, esančius krepšelyje
<i>passport types</i>	Pasų tipų lentelė
<i>visas in cart</i>	Duomenys apie vizas, esančias krepšelyje
<i>visa type</i>	Vizų tipų lentelė
<i>countries</i>	Šalys, kurioms išduodamos vizos
<i>requirements</i>	Reikalavimų susiejimo lentelė tarp <i>countries</i> ir <i>requirements items</i>
<i>requirements items</i>	Reikalavimai vizų išdavimui
<i>questions</i>	Klausimai
<i>questions list</i>	Klausimų susiejimo lentelė tarp <i>questions</i> ir <i>questions form</i>
<i>questions form</i>	Klausimai pasirinktai šaliai
<i>answers form</i>	Atsakymų susiejimo lentelė tarp <i>users</i> ir <i>answer</i>
<i>answers</i>	Vartotojo atsakymai į pateiktus šalies klausimus
<i>orders</i>	Užsakymo duomenys
<i>ordered passports</i>	Užsakyti pasai
<i>ordered visas</i>	Užsakytos vizos
<i>reminder</i>	Priminimų susiejimo lentelė tarp <i>orderis passport</i> , <i>ordered visas</i> ir <i>reminder</i>
<i>reminder items</i>	Priminimai
<i>users</i>	Vartotojai
<i>users rights</i>	Vartotojo teisės

### 3. PIVI IS IR MVC REALIZACIJA

Šiame skyrelyje trumpai apžvelgtas realizuotos sistemos ypatumai.

#### 3.1. Sistemos pasiskirstymas

Žemiau esančiame paveikslėlyje (12 pav.) pateikiamas sistemos programinių komponentų pasiskirstymas techninei įrangai.



12 pav. Sistemos pasiskirstymo diagrama

Ji sudaryta iš šių sudėtinių dalių:

- Vartotojo, kuris prisijungia prie sistemos pasinaudodamas tinklo naršykle. Vartotojas jungiasi iš bet kurio kompiuterio
- Minimalūs reikalavimai darbo vietai būtų šie:
  - „Microsoft Internet Explorer 5.0“ ar naujesnė „Microsoft Internet Explorer“ tinklo naršyklė
  - personalinis kompiuteris turi turėti išėjimą į internetą
  - vartotojo kompiuteryje turi būti įjungtas slapukų palaikymas institucijos serveriui
- Į visas vartotojo užklausas atsako serveris, sudarytas iš tokių sudėtinių dalių:
  - Apache WWW serveris, apdorojantis vartotojo užklausas, gautas HTTP/HTTPS protokolais, taip pat interpretuoja PHP kodą
  - MySQL DBVS apdorojanti SQL užklausas
  - Duomenys saugomi MySQL serveryje esančioje duomenų bazėje

Vartotojui jungiantis prie sistemos papildoma programinė įranga į vartotojo kompiuterį nediegiama.

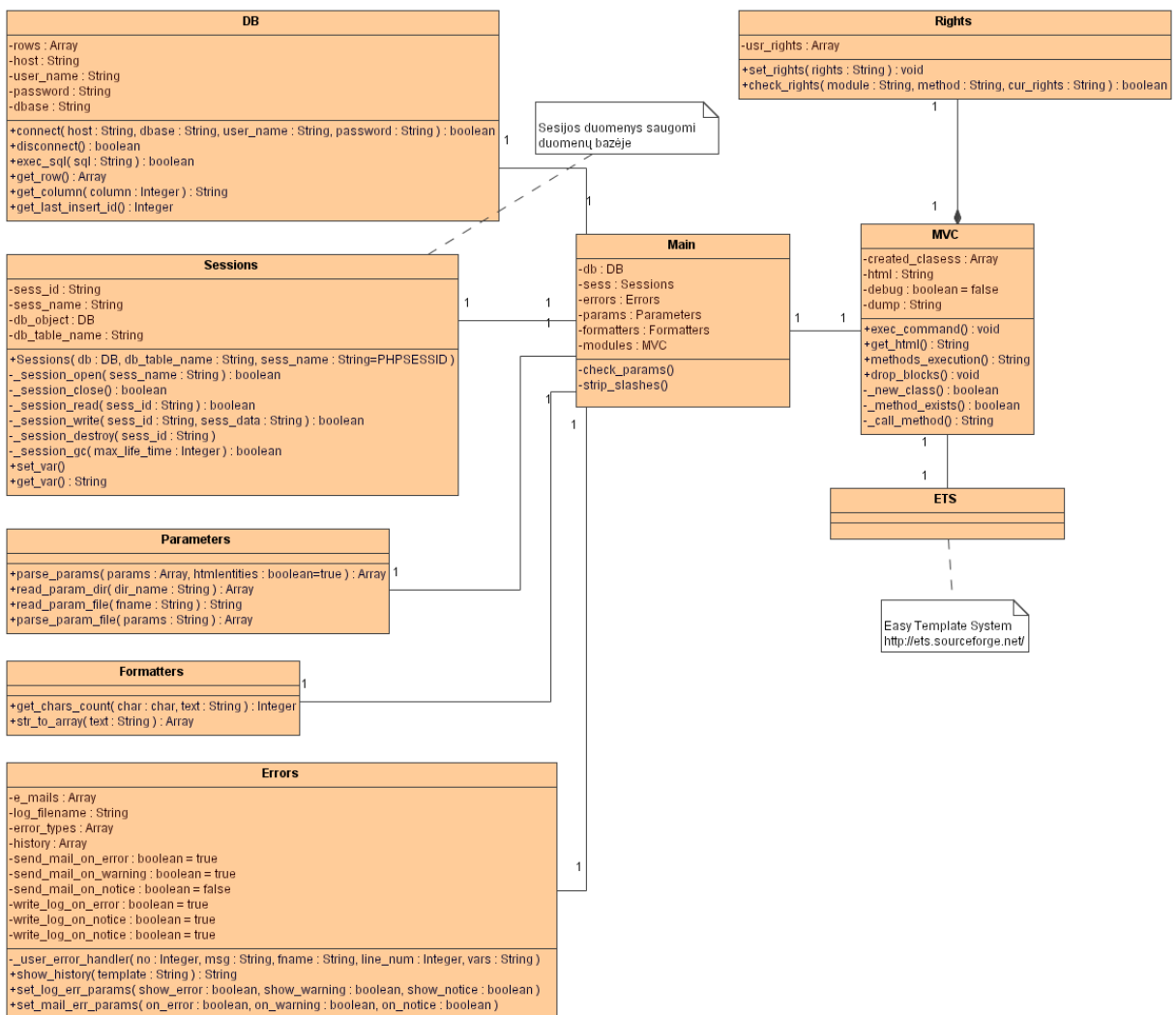
## 3.2. Klasių diagramos

### 3.2.1. MVC architektūros klasių diagrama

Pilnai MVC architektūrai funkcionuoti reikalingos 9 klasės:

- 1) DB – duomenų bazės tvarkymo klasė
- 2) Sessions – sesijų tvarkymo klasė
- 3) Parameters – parametrų tvarkymo klasė
- 4) Formatters – formatų tvarkymo klasė
- 5) Errors – klaidų apdorojimo klasė
- 6) Rights – teisių tvarkymo klasė
- 7) MVC – Model View Controller klasė
- 8) ETS – šablonų apdorojimo klasė
- 9) Main – pagrindinė klasė, apdorojanti kitas klases

Klasių diagrama pavaizduota 13 pav.



13 pav. MVC modelio klasių diagrama



### 3.2.2. PIVI IS klasių diagrama

PIVI IS funkcionavimui reikalingos 8 pagrindinės klasės:

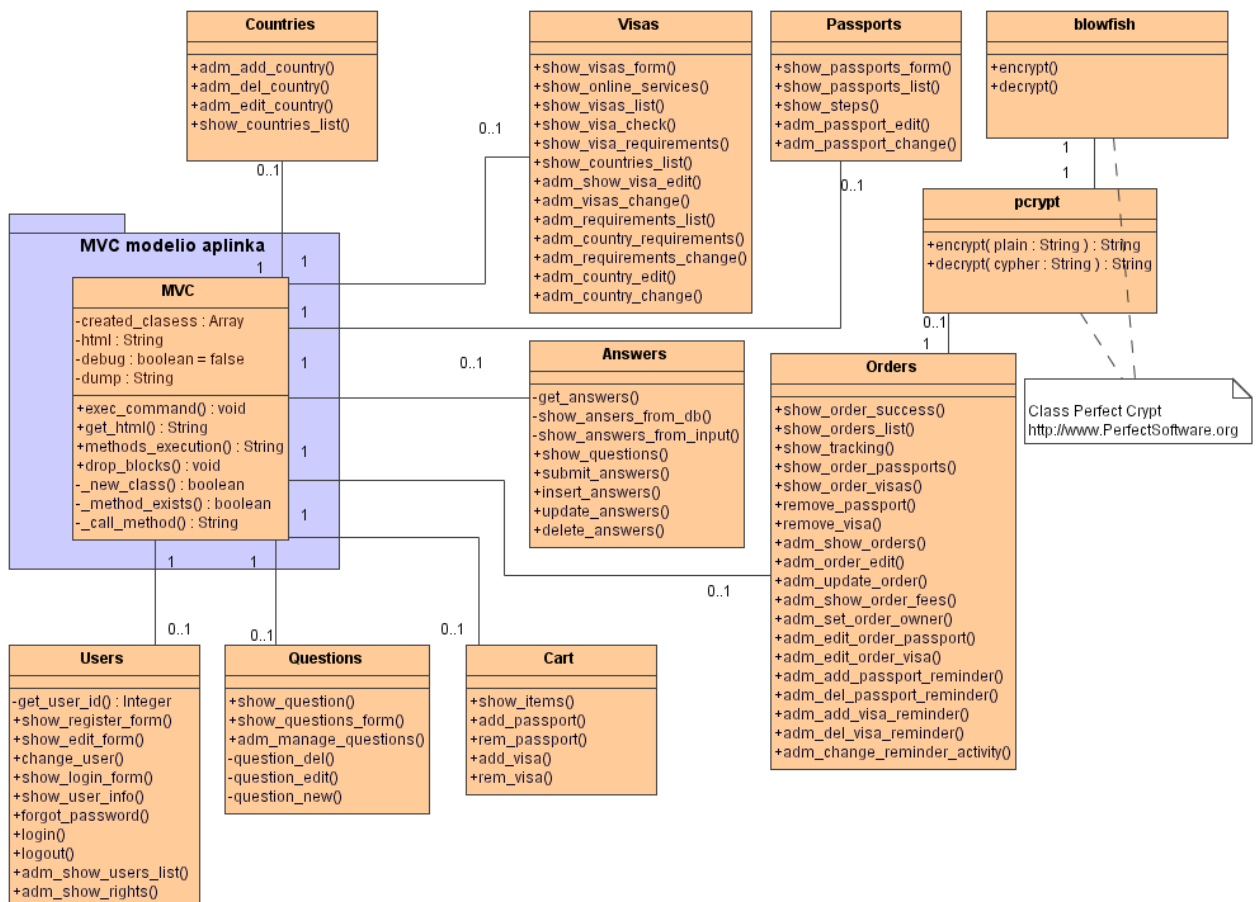
- 1) Visas – vizų tvarkymo klasė
- 2) Passports – pasų tvarkymo klasė
- 3) Questions – klausimų tvarkymo klasė
- 4) Answers – atsakymų tvarkymo klasė
- 5) Countries – valstybių tvarkymo klasė
- 6) Cart – krepšelio tvarkymo klasė
- 7) Orders – užsakymų apdorojimo klasė
- 8) Users – vartotojų tvarkymo klasė

Svarbių duomenų (vizitinių kortelių duomenys, slaptažodžiai) kodavimui naudojamos 2 Perfect Software duomenų kodavimo klasės[14]:

- 1) pcrypt – abstrakčių metodų klasė, kuri iškviečia skirtingus algoritmus
- 2) blowfish – blowfish[15] kodavimo algoritmo klasė

Taip pat naudojama MVC architektūros klasių paketas (plačiau poskyris „3.2.1. MVC architektūros klasių diagrama“).

PIVI IS klasių diagrama pavaizduota 14 pav.



14 pav. PIVI IS klasių diagrama

### 3.3. PIVI IS realizacijos aprašymas

Šiame skyriuje aprašomos pagrindinės naudojamos vartotojų sąsajos ir pateikiami šių sąsajų pavydžiai. Plačiau vartotojų sąsajos aptartos 2 priede „Instrukcija vartotojams“.

#### 3.3.1. Pateikimo sąsaja pasų ir vizų užsakovams

Sąsajos pagalba galima:

- Įtraukti vizas į krepšelį
- Įtraukti pasus į krepšelį
- Vykdyti užsakymą
- Stebėti užsakymo eigą, vadybininko pranešimus
- Keisti prisijungimo vartotojo duomenis

The screenshot displays the Passport Visa Express website interface. At the top, there is a navigation menu with links: Home, Passport Services, Visa Services (highlighted), Customer Service, Travel Partners, Corporate Clients, and Visa Support. Below the navigation, a breadcrumb trail shows: Visa Check, Visa Requirements, Visa Application, Order Form, and Order Confirmation. The main content area features a heading: "You are applying for a visa to Pakistan". Below this, there is a table with the following data:

Purpose of visit	Visa required?
Business:	Yes
Tourist:	Yes
Notes:	This information is for US citizens only. If you're not a US citizen, <a href="#">click here</a> for more information.

Below the table, there are three dropdown menus for selection:

- Visa Type: Tourist single
- Validity: 3 months
- Processing time: 2 days

Below these selections, the pricing is displayed:

- Consular price: \$ 120.00
- Service price: \$ 39.00
- Total: \$ 159

At the bottom left, there is a link: "Need assistance? [Click here](#)". At the bottom right, there is a red "continue" button. The footer contains the text: "Copyright © 2004 PassportVisaExpress.com, Inc. All rights reserved."

15 pav. PIVI IS vartotojo sąsajos pavyzdys

#### 3.3.2. Vadybininko sąsaja

Sąsajos pagalba galima:

- Prižiūrėti vartotojus:
  - sukurti, redaguoti, trinti vartotoją
  - keisti vartotojo tipą ir aktyvumą
  - užsakyti pasą ar vizą, pasirinktam vartotojui
  - suteikti teises naudotis parinktais moduliais

- priskirti vizą ar pasą atstovo (angl. agent) tipo vartotojams
- Prižiūrėti vizų sąrašus:
  - sukurti, redaguoti ir trinti valstybę
  - sukurti, redaguoti ir trinti valstybei klausimyną
  - sukurti, redaguoti, trinti, priskirti pasirinktai valstybei reikalavimus
  - sukurti, trinti, redaguoti pasirinktai valstybei vizas
  - keisti HTML puslapių antraštes (headers)
- Prižiūrėti pasų sąrašus:
  - sukurti, trinti, redaguoti pasų kategorijas
  - sukurti, redaguoti, trinti, priskirti pasirinktai pasų kategorijai reikalavimus
  - keisti HTML puslapių antraštes (headers)
- Administruoti užsakymus:
  - redaguoti, trinti užsakymus
  - redaguoti atsakymus, šalinti vizas iš užsakymo
  - peržiūrėti ir atspausdinti vizų užsakymus PDF formate
  - šalinti pasus iš užsakymo
  - sudaryti priminimo sąrašus
  - rašyti pranešimus užsakovui

#	ID	DATE	STATUS	FILE NR.	ITEMS	CLIENT	OWNER	ACTIONS
3325		2005-05-28 21:22:22	Approved		China Approved	sukotto_merced@hotmail.com Carranza, Scott Jason	None	Edit Delete
3324		2005-05-28 16:41:27	Approved		China Approved	eswapp@mac.com Swapp, Eric	Daria Ma	Edit Delete
3323		2005-05-28 13:27:27	In progress		Brazil In progress	vreale@rdg.boehringer- ingelheim.com Reale, Virginia Danielle	Borodin Alex	Edit Delete
3322		2005-05-28 12:29:03	Pending		Kenya Kenya Pending Pending	mdoak@prodjay.net Doak, Mary	None	Edit Delete
3321		2005-05-28 09:19:31	Approved		Vietnam China Cambodia Approved Approved Approved	jdunn94@aol.com Waligory, Jennifer Lynn	Andreas C. Kansil	Edit Delete
3320		2005-05-28 06:38:14	Approved	43	China Approved	joysmith3@yahoo.com SMITH, JOCELYN ELAINE	Andreas C. Kansil	Edit Delete
3319		2005-05-28 01:44:15	Approved	26	Vietnam Approved	matt.elkourie@gmail.com Elkourie, Matthew Bryan	Andreas C. Kansil	Edit Delete

Pages: [ 1 ]

16 pav. PIVI IS vadybininko sąsajos pavyzdys

### 3.4. PIVI IS įdiegimas

Šiame skyriuje aprašomi PIVI IS diegimo ir konfigūravimo etapai.

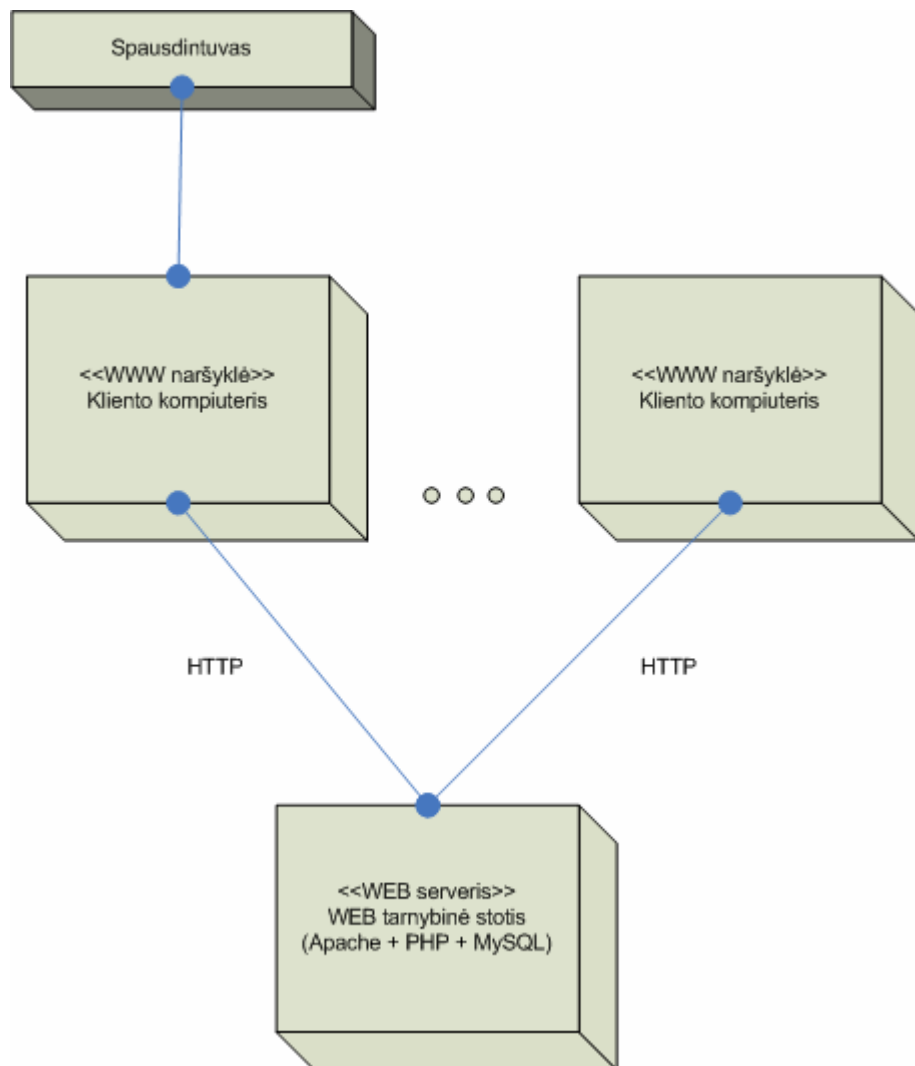
### 3.4.1. Įdiegimas

Sistemos įdiegimui didžiausi reikalavimai skiriami serverio dalies pusei, kurioje turi būti įdiegtas Apache su PHP palaikymo moduliu ir MySQL duomenų bazė. Internetinės svetainės įdiegimui reikia perkelti į serverį vykdomuosius failus ir sukonfigūruoti sistemą (plačiau skyriuje „3.4.2. Konfigūravimas“)

Kliento dalies pusei (pasų ir vizų užsakovams) reikia turėti *Microsoft Internet Explorer 5.0* ar naujesnės versijos naršyklę. Jungiantis prie sistemos papildoma programinė įranga į vartotojo kompiuterį nediegiama.

Vadybininkai, prižiūrintys IS, papildomai turi turėti *Adobe Acrobat Reader* paketą, kuris leidžia peržiūrėti ir atspausdinti PDF tipo bylas.

PIVI IS įdiegimo diagrama pavaizduoja 17 pav.



17 pav. PIVI IS įdiegimo diagrama

### 3.4.2. Konfigūravimas

Pagrindinis konfigūravimo failas randasi *etc/params.conf.php* byloje. Toliau pateikiamas parametų failo turinys su trumpais paaiškinimais:

```
#MySQL duomenų bazės parametrai.
[db_class]

    #MySQL tarnybinės stoties adresas.
    host=localhost:3306

    #MySQL vartotojo vardas.
    user=root

    #MySQL vartotojo slaptažodis.
    password=test123

    #MySQL duomenų bazės pavadinimas.
    dbname=pve

#Keliai.
[root_path]

    dir=/home/www/passportvisaexpress.com/
    www=http://www.passportvisaexpress.com/
    www_ssl=https://www.passportvisaexpress.com/
    modules=modules/
    tpl=tpl/
    css=css/
    logs=logs/
    templates=templates/
    pdf_templates=(root_path:www)pdf_templates/
    fdf_temp=temp/
    htmls=htmls/
    skins=skins/

#Klaidų apdorojimo klasės parametrai.
[err_class]

    #Klaidų saugojimo failas.
    log_fname=(root_path:logs)

    #El. pašto nustatymai.
    mail_to_on_error=info@passportvisaexpress.com
    mail_subject=[www.passportvisaexpress.com] Error report!

    #Siūsti el. paštą iškilus nesklandumams (įvykus klaidai, išpėjimui, pastabai).
    mail_on_error=1
    mail_on_warning=1
    mail_on_notice=1

    #Išsaugoti pranešimą jei įvyko nesklandumas (klaida, išpėjimas, pastaba).
    log_on_error=1
    log_on_warning=1
    log_on_notice=1

    #Sustabdyti veikimą jei įvyko nesklandumas (klaida, išpėjimas, pastaba).
    halt_on_error=0
    halt_on_warning=0
    halt_on_notice=0

    #Rodyti pranešimą vartotojui jei įvyko nesklandumas (klaida, išpėjimas, pastaba).
    show_on_error=1
    show_on_warning=1
    show_on_notice=1

#Modulių parametrai.
[modules]

    #Klasė pagal nutylėjimą.
    default=users
```

```
#Šablonų parametrai.
[templates]

    #Šablonų bylų išplėtimas.
    extension=.htm

[password]

    #Minimalus simbolių skaičius, įvedant slaptažodį.
    min_chars=4

#SMTP serverio parametrai.
[smtp]

    #SMTP tarnybinės stoties adresas.
    host_name=localhost

    #SMTP vartotojo vardas.
    user_name=

    #SMTP vartotojo slaptažodis.
    password=

    #El. pašto adresas, iš kurio siunčiami el. laiškai.
    from_email=staff@passportvisaexpress.com

    #Rodyti el. pašto siuntimo eigą.
    debug=0

#Slapukų parametrai.
[cookies]

    #Slapukų gyvavimo laikas sekundėmis.
    live_time=2592000

#Apipavidalinimo parametrai.
[skins]

    #Apipavidalinimo konfigūracinio failo išplėtimas.
    conf_files_ext = .skin.conf.php

    #Apipavidalinis pagal nutylėjimą.
    default_skin = s_home

#Puslapavimo parametrai.
[paging]

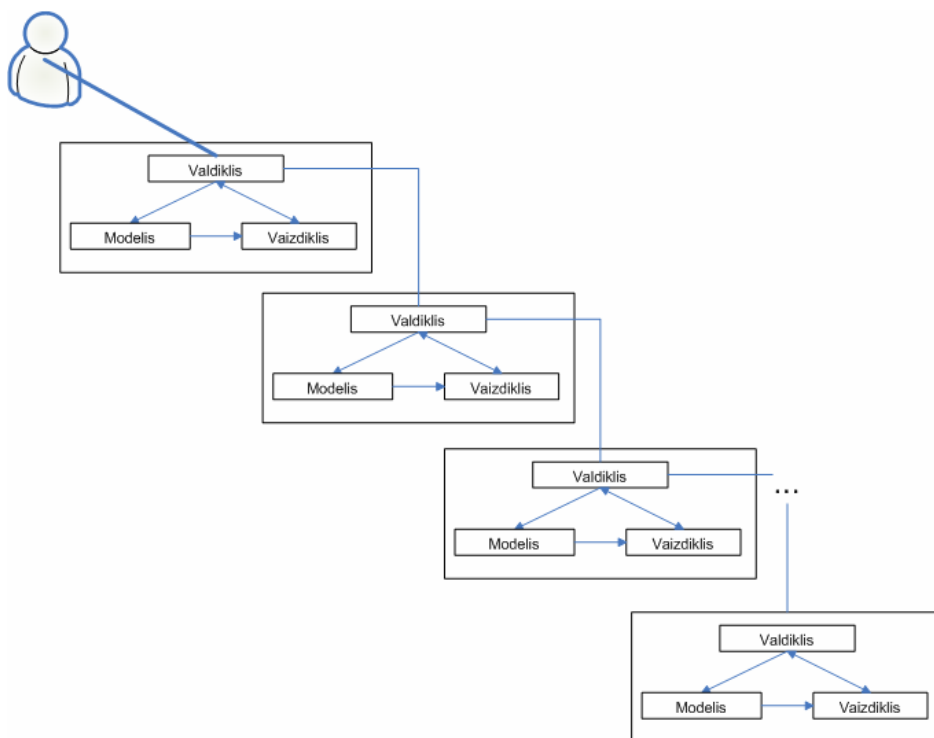
    #Maksimalus įrašų kiekis lape (vartotojų, užsakymų sąrašai).
    show_items_per_page=30
```

## 4. MVC ARCHITEKTŪROS TYRIMAS

Šiame darbo skyriuje pateikta realizuotos MVC architektūros, metodų iškvietimo iš šabloninių bylų, spartos tyrimas. Tyrimas atliekamas naudojant Pentium IV, 2.6GHz., 1GB RAM tipo kompiuterį.

### 4.1. Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn

Algoritmo gilyn veikimo principas pavaizduotas 18 pav.



18 pav. Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn

#### 4.1.1. Algoritmo gilyn tyrimo eiga

Kreipiamasi adresu (URL): [http://localhost/magistras/experiment/index.php?experiment,action.alg\\_deep,start.5](http://localhost/magistras/experiment/index.php?experiment,action.alg_deep,start.5)

Pagal adreso pateiktus parametrus, MVC architektūros valdiklis, iškviečia metodą *act\_alg\_deep* iš modulio *experiment.mod.php*

Modulio bylą:	experiment.mod.php
Metodas:	<pre>function act_alg_deep() {     \$start = \$this-&gt;get('start');     \$action = '';     if (\$start &gt; 0) {         \$action = '{module:experiment,action.alg_deep,start.'.(\$start-1).'}';     }     \$mData = array(         'action' =&gt; \$action,         'start' =&gt; \$start,     );     return \$mData; }</pre>

Šis metodas yra susijęs su šablonine byla *alg\_deep.htm*

Šablono byla:	<i>alg_deep.htm</i>
Bylos turinys	{action}start={start}

Įvykdžius *act\_alg\_deep* metodą, šabloninės bylos *alg\_deep.htm* turinys pasikeičia į:

Šablono byla:	<i>alg_deep.htm</i>
Bylos turinys	{module:experiment,action.alg_deep,start.4} start=5

Išanalizavus šią byla, randama žyma ({module:experiment,action.alg\_deep,start.4}), kuri iškviečia metodą *act\_alg\_deep* iš modulio *experiment.mod.php*, tik dabar parametro start reikšmė – 4. Įvykdžius metodą šabloninės bylos *alg\_deep.htm* turinys pasikeičia į:

Šablono byla:	<i>alg_deep.htm</i>
Bylos turinys	{module:experiment,action.alg_deep,start.3} start=4

Algoritmo gilyn vykdymas sustoja, kai parametro start reikšmė tampa 0, o *alg\_deep.htm* bylos turinys pasikeičia į:

Šablono byla:	<i>alg_deep.htm</i>
Bylos turinys	start=5 start=4 start=3 start=2 start=1

#### 4.1.2. Algoritmo gilyn laikiniai rezultatai

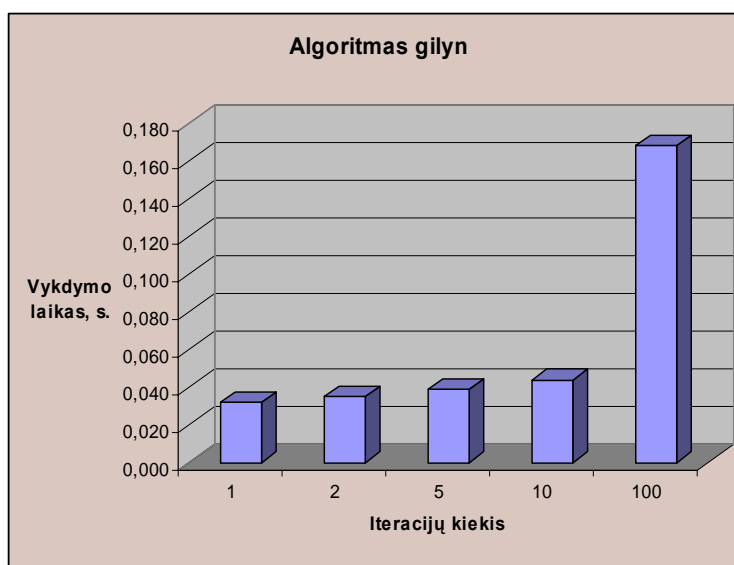
Algoritmo gilyn laikiniai rezultatai pavaizduoti 12 lentelėje.

12 lentelė

Metodų iškvietimas iš šablono bylos naudojant algoritmą gilyn

	Iteracijos				
Vykdymo laikas	1	2	5	10	100
	0.032 s.	0.035 s.	0.039 s.	0.044 s.	0.168 s.

Algoritmo gilyn laikiniai rezultatai grafiškai pavaizduoti 19 pav.

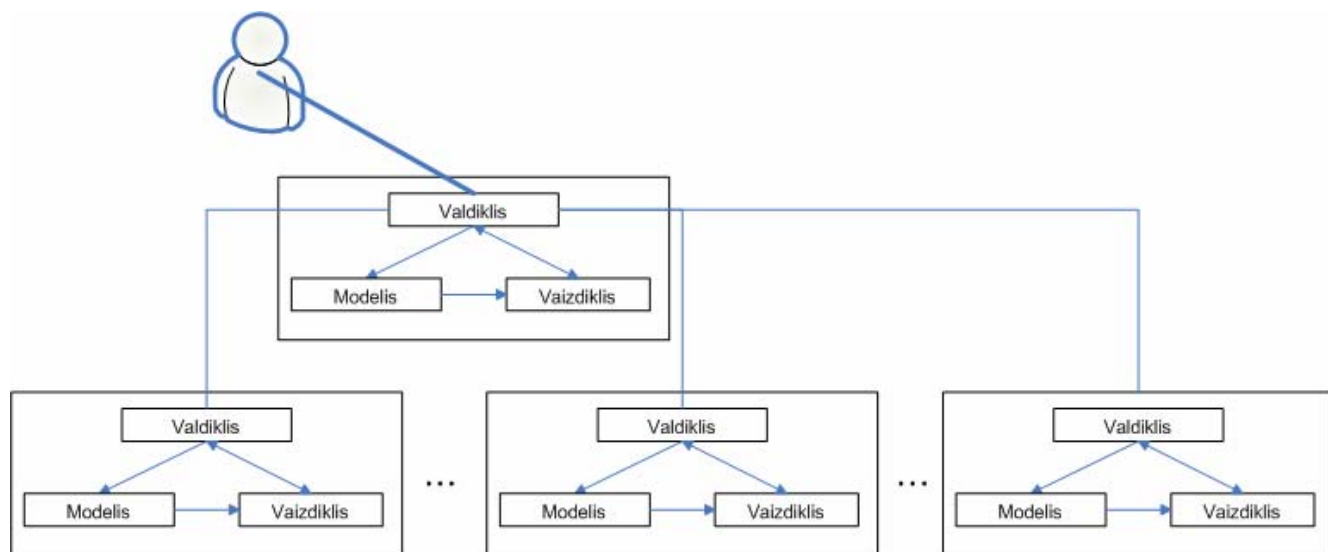


19 pav. Algoritmo gilyn vykdymo laikas



## 4.2. Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn

Algoritmo platyn veikimo principas pavaizduotas 20 pav.



20 pav. Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn

### 4.2.1. Algoritmo platyn tyrimo eiga

Kreipiamasi adresu (URL): [http://localhost/magistras/experiment/index.php?experiment,action.alg\\_width](http://localhost/magistras/experiment/index.php?experiment,action.alg_width)

Pagal adreso pateiktus parametrus, MVC architektūros valdiklis, iškviečia metodą *act\_alg\_width* iš modulio *experiment.mod.php*

Modulio bylą:	<code>experiment.mod.php</code>
Metodai:	<pre>function act_alg_width() { }  function act_alg_width_msg() {     \$start = \$this-&gt;get('start');     \$mData = array('start' =&gt; \$start);     return \$mData; }</pre>

Šis metodas yra susijęs su šablonine byla *alg\_width.htm*

Šablono bylą:	<i>alg_width.htm</i>
Bylos turinys	<pre>{module:experiment,action.alg_width_msg,start.1} {module:experiment,action.alg_width_msg,start.2} ... {module:experiment,action.alg width msg,start.5}</pre>

Išanalizavus šią bylą, randamos žymos (`{module:experiment,action.alg_width_msg,start.1}` ir kt.), kurios iškviečia metodą *act\_alg\_width\_msg* iš modulio *experiment.mod.php* ir perduoda atitinkamas parametro *start* reikšmes – 1,2..5. Metodas *act\_alg\_width\_msg* susijęs su *alg\_width\_msg.htm* byla:

Šablono bylą:	<i>alg_width_msg.htm</i>
Bylos turinys	<code>start={start}</code>

Įvykdžius metodą *act\_alg\_width\_msg* gražinamas *alg\_width\_msg.htm* bylos turinys su pakeistu *{start}* parametru. Algoritmo platyn vykdymas sustoja, kai iš *alg\_width.htm* bylos iššaukiami visi metodai. Tada *alg\_width.htm* bylos turinys pasikeičia į:

Šablono bylą:	<i>alg_deep.htm</i>
Bylos turinys	start=1 start=2 start=3 start=4 start=5

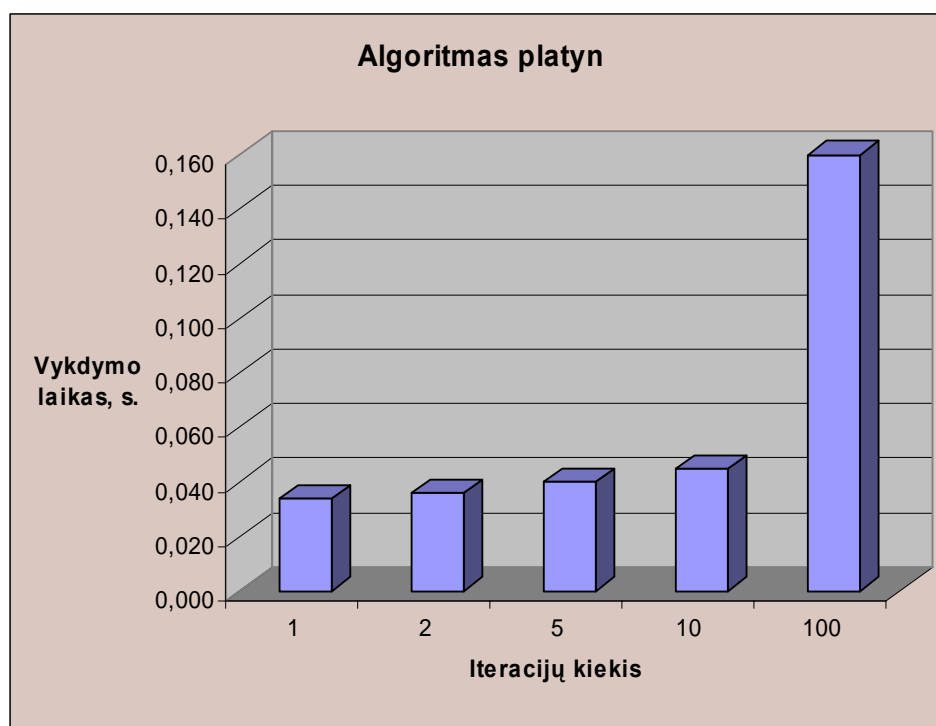
#### 4.2.2. Algoritmo platyn laikiniai rezultatai

Algoritmo gilyn laikiniai rezultatai pavaizduoti 13 lentelėje.

13 lentelė

		Metodų iškvietimas iš šablono bylos naudojant algoritmą platyn				
		Iteracijos				
Vykdymo laikas	1	2	5	10	100	
		0.034 s.	0.036 s.	0.040 s.	0.045 s.	0.160 s.

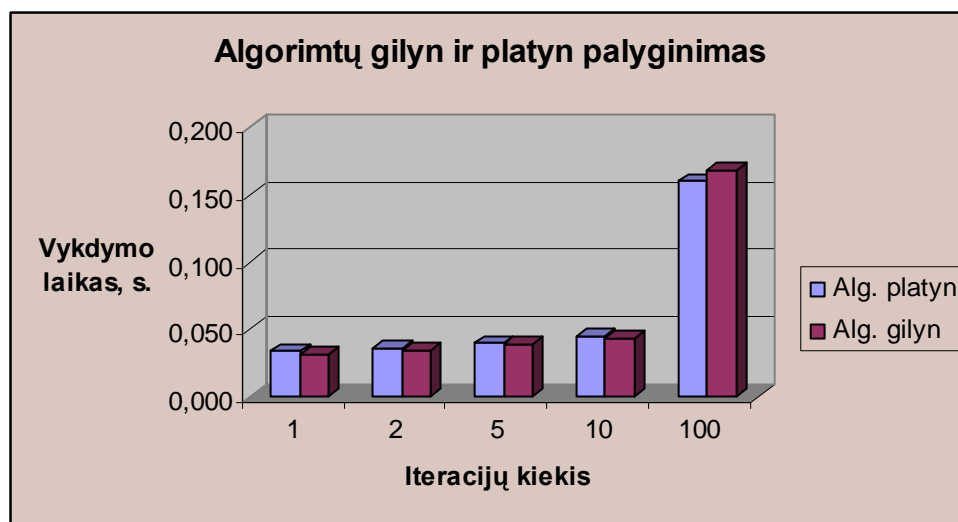
Algoritmo platyn laikiniai rezultatai grafiškai pavaizduoti 21 pav.



21 pav. Algoritmo platyn vykdymo laikas

### 4.3. Algoritmų gilyn ir platyn palyginimas

Algoritmų gilyn ir platyn palyginimas pavaizduotas 22 pav.



22 pav. Algoritmų gilyn ir platyn palyginimas

Paveikslėlyje matosi, kad esant iteracijų kiekiui iki 100, algoritmas gilyn veikia greičiau, tačiau padidėjus iteracijų kiekiui, greičiau veikia algoritmas platyn. Algoritmas platyn, esant ne dideliui iteracijų kiekiui, veikia lėčiau, nes šabloninės bylos analizė užtrunka ilgiau.

### 4.4. Parametrų ir modulių išskleidimo palyginimas

Šiame skyrelyje apteiktas parametrų ir modulių išskleidimo tyrimas. Tarkime turime tokią eilutę: Users,action.doLogin,FName.Petras,LName.Petraitis;Updater,action.updateLoginTime,now.2006-01-03

Šioje eilutėje galima išskirti 2 modulius: *Users* ir *Updater*. Iš modulio *Users* galima išskirti tokius parametrus: action = doLogin, FName = Petras, LName = Petraitis, o iš modulio *Updater* galima išskirti tokius parametrus: action = updateLoginTime, now = 2006-01-03.

Atliekame parametrų išskleidimo tyrimą, kai modulių kiekis 1000, o parametrų kiekis kinta nuo 1 iki 100. Laikiniai rezultatai pavaizduoti 14 lentelėje.

14 lentelė

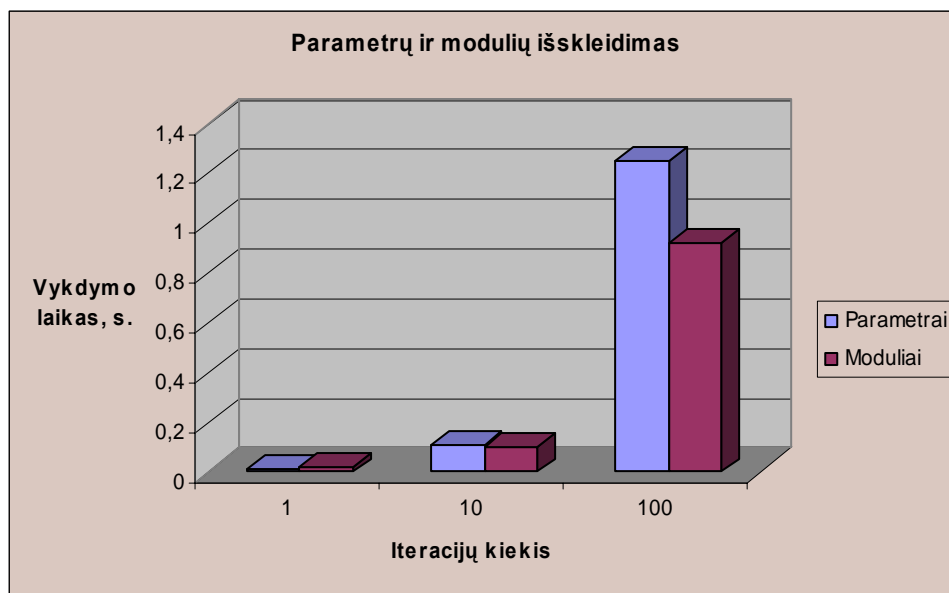
Parametrų išskleidimas, kai modulių kiekis pastovus				
1000 modulių		Parametrų kiekis		
		1	10	100
Vykdymo laikas, s.		0.018	0.101	0.922

Atliekame parametrų išskleidimo tyrimą, kai parametrų kiekis 1000, o modulių kiekis kinta nuo 1 iki 100. Laikiniai rezultatai pavaizduoti 15 lentelėje.

**Modulių išskleidimas, kai parametų kiekis pastovus**

1000 parametų	Modulių kiekis		
	1	10	100
Vykdomo laikas, s.	0.010	0.106	1.248

Parametų ir modulių išskleidimo laikiniai rezultatai grafiškai pavaizduoti 23 pav.



**23 pav.** Parametų ir modulių išskleidimas

#### 4.5. Tyrimo išvados

Šio tyrimo metu buvo siekiami ištirti ir palyginti algoritmų gilyn ir platyn veikimo spartas, patikrinti ar parametų ir modulių išskleidimas ženkliai įtakoja laikinius rezultatus.

Įvertinus gautus rezultatus galima teigti:

1. MVC architektūros algoritmų gilyn ir platyn laikiniai rezultatai labai panašūs
2. Algoritmas platyn veikia lėčiau, nes reikalauja daugiau laiko šabloninės bylos turinio analizei
3. Parametų ir modulių išskleidimas laikinius rezultatus ženkliai įtakoja tik esant dideliame modulių ir parametų kiekiui, o tai praktikoje pasitaiko labai retai

## 5. IŠVADOS

1. Darbo tikslas išnagrinėti pasų ir vizų išdavimo informacinės sistemas, atlikti MVC architektūros analizę ir realizaciją, parinkti šablonų apdorojimo klasę, programavimo priemones, realizuoti pasų ir vizų išdavimo internetinę svetainę MVC architektūros pagrindu.
2. Atlikus programavimo priemonių analizę buvo pasirinkta PHP kalba ir MySQL duomenų bazių valdymo sistema. Darbe aprašyti PHP privalumai prieš kitas technologijas ir geras ryšys su MySQL DBVS nulėmė šį pasirinkimą.
3. Šio darbo naujumas - sukurta MVC architektūra, leidžianti iš šablonų bylų, algoritmais gilyn ir platyn iškviešti (kreipiantis į valdiklį) pasirinkto modulio metodus. Metodų iškvietimui naudojamos papildomos HTML žymos.
4. Atlikus tyrimą paaiškėjo, kad algoritmų gilyn ir platyn laikiniai rezultatai panašūs, o parametrų ir modulių išskleidimas beveik neįtakoja laikinių rezultatų.
5. MVC architektūra ir sukurti moduliai gali būti panaudojami kuriant kitas internetines svetaines, o atlikus nedidelius modifikavimus galima lengvai realizuoti panašaus funkcionalumo informacinės sistemas.
6. Magistrinio darbo metu buvo suprojektuota, realizuota, įdiegta ir palaikoma PIVI IS. Jau yra eksploatavimo patirtis - 0.5 m.
7. Realizuotai PIVI IS trūksta automatinio transakcijų atlikimo su banku.

## 6. LITERATŪRA

1. Valerijus Bogunas, *Elektroniniai kontaktai internete* [interaktyvus, žiūrėta 2004-12-02]. Prieiga per internetą: <<http://www.esecurity.lt/article/1068.html>>.
2. *Go To Russia Travel* Amerikos kompanijos pasų ir vizų išdavimo IS. [interaktyvus, žiūrėta 2004-03-17]. Prieiga per internetą: <<http://GoToRussia.com>>
3. *Passports and Visas.com, LLC*. Amerikos kompanijos pasų ir vizų išdavimo IS. [interaktyvus, žiūrėta 2004-03-17]. Prieiga per internetą: <<http://www.PassportsAndVisas.com>>
4. *Passport Visa Express, Inc.* Amerikos kompanijos pasų ir vizų išdavimo IS. [interaktyvus, žiūrėta 2004-03-17]. Prieiga per internetą: <<http://www.PassportVisaExpress.com>>
5. *ServerSideScripting techniques analysis* [interaktyvus, žiūrėta 2004-12-22]. Prieiga per internetą: <<http://www.b2bsim.de/documents/wewior/main.html>>.
6. *.Net vs Coldfusion, PHP, JAVA* [interaktyvus, žiūrėta 2004-12-22]. Prieiga per internetą: <<http://www.promoteware.com/Module/Article/ArticleView.aspx?id=10>>.
7. *Integro, Technologies comparision (PHP, J2EE, .NET)* [interaktyvus, žiūrėta 2004-12-22]. Prieiga per internetą: <<http://icm.integro.com/Documents/TechnologyComparison.pdf>>
8. *A comparison of several server-side technologies* [interaktyvus, žiūrėta 2004-12-22]. Prieiga per internetą: <[http://onlinehome.us/docs/Masters\\_Report.doc](http://onlinehome.us/docs/Masters_Report.doc)>
9. *Zend Optimizer for PHP* [interaktyvus, žiūrėta 2004-12-24]. Prieiga per internetą: <<http://www.zend.com/store/products/zend-optimizer.php>>
10. *Design Patterns* [interaktyvus, žiūrėta 2004-03-17]. Prieiga per internetą: <<http://www.javacamp.com>>
11. *PHP Nuke architektūra* [interaktyvus, žiūrėta 2005-03-18]. Prieiga per internetą: <<http://www.phpnuke.org>>
12. *php.MVC architektūra* [interaktyvus, žiūrėta 2005-04-05]. Prieiga per internetą: <<http://www.phpmvc.net/docs/guides/guidesIdx.php?doc=phpmvc-101>>
13. *Šablonų apdorojimo klasių palyginimas* [interaktyvus, žiūrėta 2005-04-26]. Prieiga per internetą: <[http://pukomuko.esu.lt/phemplate/benchmark/tpl\\_benchmark.zip](http://pukomuko.esu.lt/phemplate/benchmark/tpl_benchmark.zip)>
14. Igor Ribeiro de Assis, *Perfect Software* pcrypt ir blowfish klasės. [interaktyvus, žiūrėta 2005-05-25]. Prieiga per internetą: <<http://www.PerfectSoftware.org/index.php?page=cat&cat=Security>>
15. Bruce Schneier, *The Blowfish Encryption Algorithm*. [interaktyvus, žiūrėta 2005-05-25]. Prieiga per internetą: <<http://www.schneier.com/blowfish.html>>

## 7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

16 lentelė

Terminai ir santrumpos

Terminas	Paaškinimas
IS	Informacinė sistema
PIVI IS	Pasų ir vizų išdavimo informacinė sistema
MVC	Modelis, pagrįstas vartotojo sąsajos, programos logikos ir duomenų lygmens atskyrimu ( <i>Model View Controller</i> )
WWW	Pasaulinis kompiuterinis tinklas ( <i>World Wide Web</i> )
DB	Duomenų bazė ( <i>Data Base</i> )
DBVS	Duomenų bazių valdymo sistema
SQL	Struktūrinė užklausų kalba ( <i>Structured Query Language</i> )
SQL injection	SQL instrukcijų įterpimas į WWW tinklapio parametrus taip, kad tos instrukcijos būtų įvykdytos SQL serveryje
JSP	Serverio pusėje vykdomų programų kalba ( <i>Java Server Pages</i> )
PHP	Serverio pusėje vykdomų programų kalba ( <i>Personal Home Page</i> )
HTML	Kalba naudojama tinklalapių atvaizdavimui, vykdoma kliento pusėje ( <i>Hyper Text Markup Language</i> )
XML	Išplėstinė žymių kalba - duomenų formatas struktūrizuotų dokumentų keitimuisi tinkle ( <i>eXtensible Markup Language</i> )
PDF	Portatyvus dokumento formatas. Adobe Systems kompanijos sukurtas formatas, kuris leidžia iš anksto suformuotus puslapius platinti kompaktiniuose diskuose arba kompiuterių tinklais. Apie jį galima paskaityti internete <a href="http://www.adobe.com/">http://www.adobe.com/</a> ( <i>Portable Document Format</i> )
JDBC	Java duomenų bazių jungtis ( <i>Java Data Bases Connection</i> )
ADO	ActiveX duomenų objektai ( <i>ActiveX Data Objects</i> )
MySQL	Atvirojo kodo duomenų bazių valdymo sistema (DBVS)
UML	Universali modeliavimo kalba ( <i>Unified Modeling Language</i> )
URL	Interneto išteklių adresas ( <i>Universal Resource Locator</i> )
Cookie	Slapukas, mažas dokumentas, kuris išsaugomas vartotojo diske. Jame interneto naršyklė gali rašyti įvairia tarnybinę informaciją
COM	Komponentinis objektų modelis ( <i>Component Object Model</i> )
CASE	Programinė įranga arba programų paketai, skirti supaprastinti programų kūrimą ir palaikymą ( <i>Computer Aided-Software Engineering</i> )
API	Taikomųjų programų programavimo sąsaja ( <i>Application Programming Interface</i> )

## **PRIEDAI**

Žemiau pateikiami dokumentą lydintys priedai.

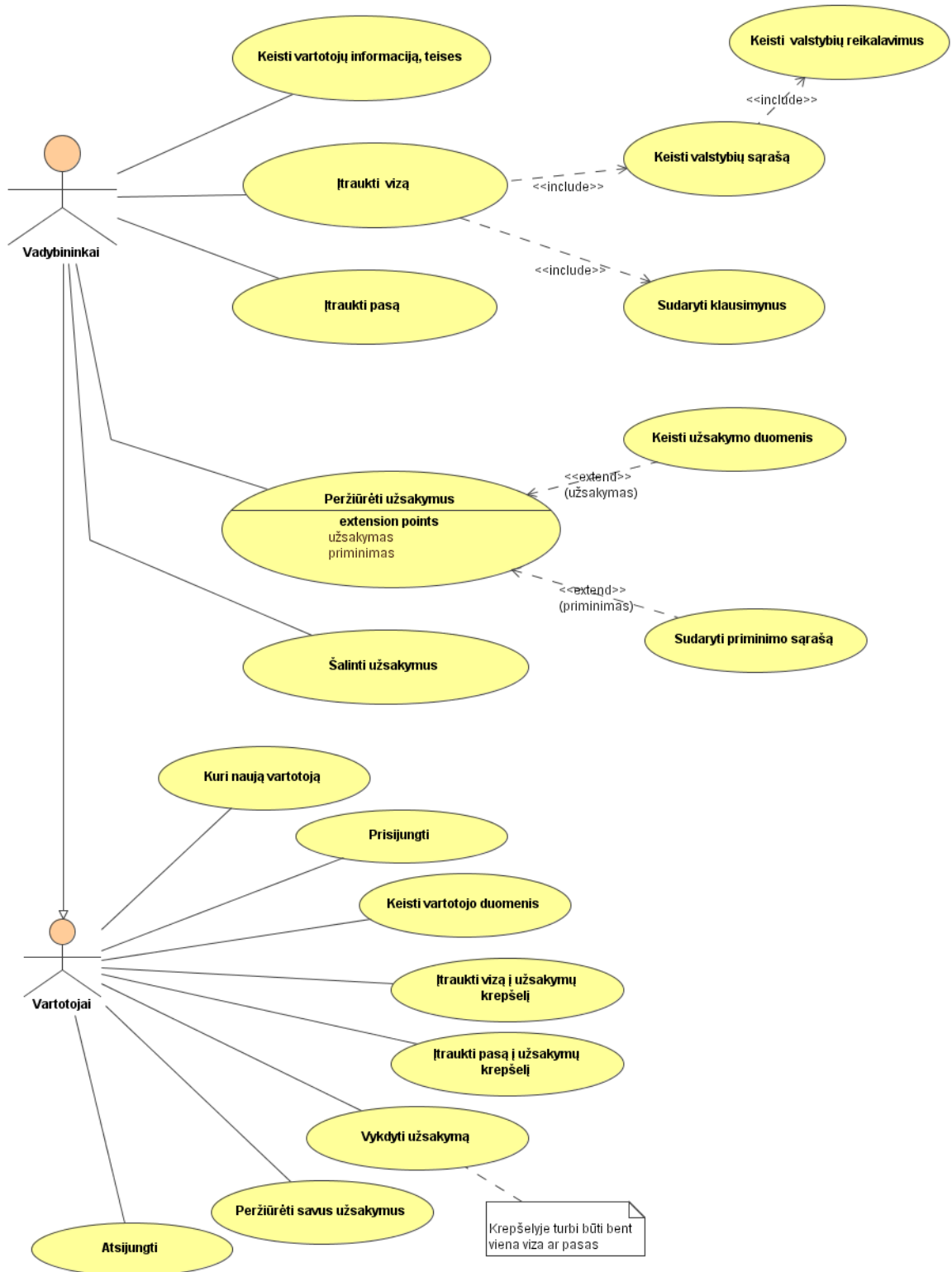


# 1 PRIEDAS. UML DIAGRAMOS IR SPECIFIKACIJOS

## 1. Pasų ir vizų išdavimo IS UML diagramos

### 1.1. Panaudojimo atvejų diagrama

Panaudojimo atvejų diagrama pavaizduota 1 pav.



1 pav. Veiklos panaudojimo atvejų diagrama

## 1.2. Panaudojimo atvejų specifikacijos

### Panaudojimo atvejo “Kurti naują vartotoją” specifikacija

<b>Prieš sąlyga</b>	Vartotojas neužregistruotas duomenų bazėje.
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas atsako į pateiktus klausimus	1.1. Sistema tikrina, ar visi privalomi laukeliai užpildyti, jei ne pranešama ir pereinama į 1 žingsnį. 1.2. Sistema tikrina, ar įvesti slaptažodžiai sutampa, jei ne praneša apie nesutapimą ir pereina į 1 žingsnį. 1.3. Sistema tikrina ar įvesto slaptažodžio ilgis už privalomą, jei ne praneša ir pereina į 1 žingsnį.
2. Vartotojui pranešama apie sėkmingą vartotojo sukūrimą	
<b>Po sąlyga:</b>	Vartotojo duomenys įrašomi į duomenų bazę.

### Panaudojimo atvejo “Keisti vartotojo duomenis” specifikacija

<b>Prieš sąlyga</b>	Sukurto vartotojo duomenys saugojami duomenų bazėje.
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas atsako į pateiktus klausimus	1.1. Sistema tikrina, ar visi privalomi laukeliai užpildyti, jei ne pranešama ir pereinama į 1 žingsnį. 1.2. Sistema tikrina, ar įvesti slaptažodžiai sutampa, jei ne praneša apie nesutapimą ir pereina į 1 žingsnį. 1.3. Sistema tikrina ar įvesto slaptažodžio ilgis už privalomą, jei ne praneša ir pereina į 1 žingsnį.
2. Vartotojui pranešama apie sėkmingą duomenų keitimą	
<b>Po sąlyga:</b>	Asmeniniai vartotojo duomenys įrašomi į duomenų bazę.

### Panaudojimo atvejo “Įtraukti vizą į užsakymų krepšelį” specifikacija

<b>Prieš sąlyga</b>	
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas pasirenka šalį	1.1. Sistema tikrina, ar duomenų bazėje yra tokia šalis. Jei tokios šalies nėra, sistema baigia darbą.
2. Vartotojas pasirenka vizos tipą	2.1. Jei tokio vizos tipo nėra, sistema baigia darbą, kitu atveju rodoma šios paslaugos kaina.
3. Vartotojas supažindinamas su reikalavimais	3.1. Jei vartotojas nesutinka su reikalavimais, sistema baigia darbą.
4. Vartotojas atsako į pateiktus klausimus	4.1. Sistema tikrina ar visi privalomi laukeliai užpildyti. Jei užpildyti, sistema išsaugo duomenis (šalį, vizos tipą, atsakymus į tos šalies klausimus) ir pereina į 5 žingsnį. Jei ne visi laukeliai užpildyti, sistema pereina į 4 žingsnį.

5. Vartotojui pateikiama ataskaita apie užsakytą vizą	5.1. Jei vartotojas pasirenka klausimų redagavimą, sistema pereina į 4 žingsnį.
<b>Po sąlyga:</b>	Duomenų bazėje išsaugoma užsakyta visa su vartotojo ir šalies identifikatoriais.

### Panaudojimo atvejo “Įtraukti pasą į užsakymų krepšelį” specifikacija

<b>Prieš sąlyga</b>	
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Pasirenkamas paso tipas	1.1. Sistema tikrina, ar duomenų bazėje yra toks tipas. Jei nėra, sistema baigia darbą.
2. Pasirenkamas paso pagaminimo laikotarpis	2.1. Sistema tikrina ar pasirinktam paso tipui yra toks pagaminimo laikotarpis. Jei nėra, sistema baigia darbą.
3. Vartotojas supažindinamas su reikalavimais	3.1. Jei vartotojas nesutinka su reikalavimais, sistema baigia darbą.
4. Patvirtina	4.1. Sistema išsaugo užsakyto paso duomenis.
<b>Po sąlyga:</b>	Duomenų bazėje išsaugomas užsakytas pasas su vartotojo identifikatoriumi.

### Panaudojimo atvejo “Vykdėti užsakymą” specifikacija

<b>Prieš sąlyga</b>	Krepšelyje yra visa ar pasas.
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas prisijungia prie sistemos (šis žingsnis apima panaudojimo atvejį “Prisijungti”)	
2. Vartotojas vykdo užsakymą	2.1. Sistema tikrina ar vartotojas prisijungęs prie sistemos. Jei ne, nukreipia į prisijungimo langą (šis žingsnis apima panaudojimo atvejį “Prisijungti”) 2.2. Sistema tikrina, ar krepšelyje yra bent viena vizą ar pasas. Jei nėra, sistema baigia darbą.
3. Vartotojui pateikiami klausimai	3.1. Sistema tikrina ar naudojamas saugus kanalas (SSL) tarp kliento ir serverio. Jei ne, sistema nukreipia į saugų kanalą ir pereina į antrą žingsnį.
4. Vartotojas atsako į pateiktus klausimus	4.1. Sistema tikrina ar visi pageidaujami laukeliai užpildyti. Jei užpildyti, sistema išsaugo duomenis (užsakymo, vizų ir/ar pasų) ir pereina į 5 žingsnį. Jei ne visi laukeliai užpildyti, sistema pereina į 3 žingsnį.
5. Vartotojui pateikiama užsakymo ataskaita	
<b>Po sąlyga:</b>	Duomenų bazėje išsaugomas užsakymas su vartotojo identifikatoriumi. Taip pat išsaugomi užsakyti pasai ir/ar vizos su užsakymo identifikatoriumi.

### Panaudojimo atvejo "Prisijungti" specifikacija

<b>Prieš sąlyga</b>	Vartotojas įtrauktas į duomenų bazę.
<b>Įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas įveda prisijungimo vardą ir slaptažodį	1.1. Sistema tikrina, ar duomenų bazėje yra toks vartotojas. Jei vartotojas yra ir jis aktyvus, sistema pereina į 1.2 žingsnį, kitu atveju atspausdina atitinkamą pranešimą ir pereina į 1 žingsnį. 1.2. Sistema tikrina ar reikia nukreipti puslapį. Jei taip, nukreipiama į nurodytą puslapį. Jei ne, nukreipiama į pagal nutylėjimą nurodytą puslapį.
<b>Po sąlyga:</b>	Vartotojo duomenys įrašyti į sesiją. Vartotojas gali vykdyti užsakymą.

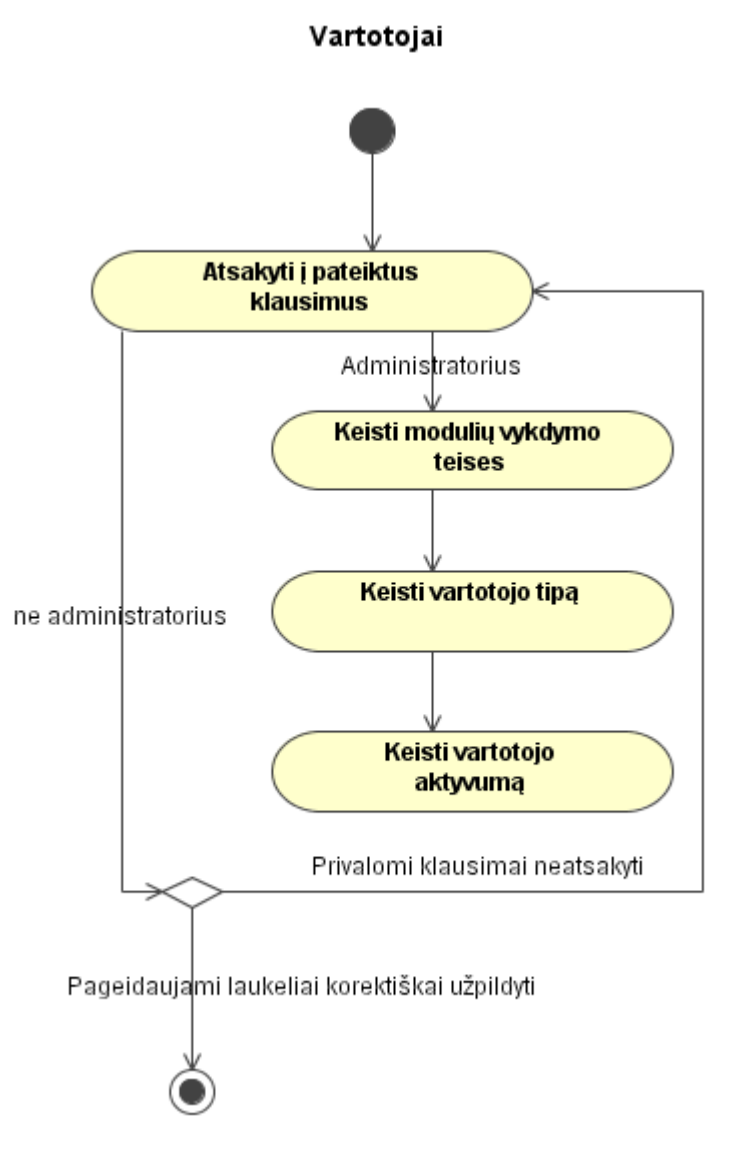
### 1.3. Būsenų diagramos

#### 1.3.1. Keisti vartotojo duomenis

Keičiant vartotojo duomenis rodomi klausimai, kurie skirstomi į privalomus ir neprivalomus. Neatsakius į privalomus klausimus, vartotojų modulis praneša apie klaidą ir prašo užpildyti privalomus laukelius.

Jei prisijungęs vartotojas vadybininkas, jis gali keisti vartotojo aktyvumą (aktyvus/neaktyvus), vartotojo tipą (vadybininkas, paprastas vartotojas, pareigybinis, kolektyvinis, atstovas). Taip pat gali suteikti kiekvienam moduliui vykdymo teises: r – read (nuskaityti), w – write (rašyti).

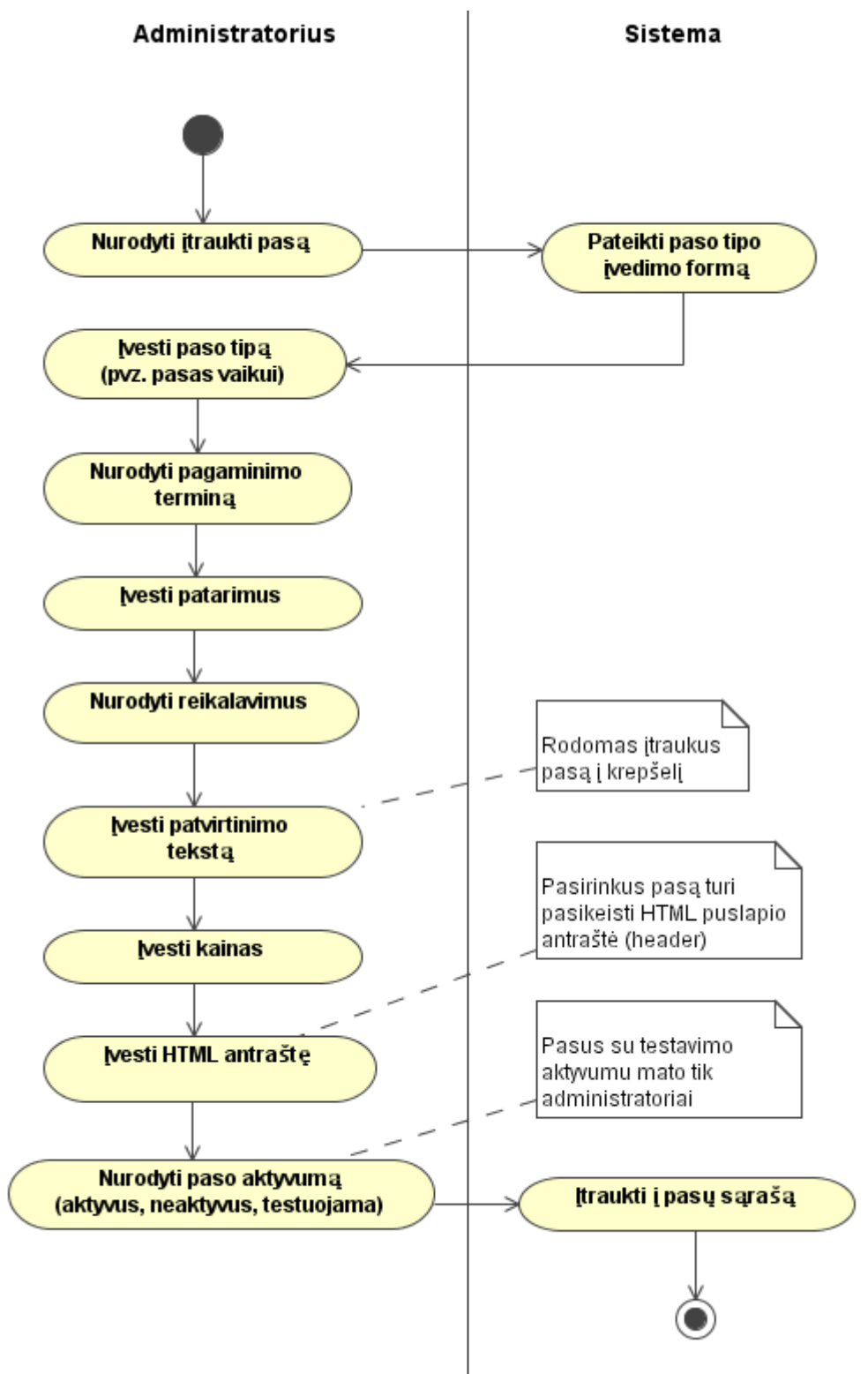
Būsenų diagrama, kai vartotojas keičia asmeninius duomenis pavaizduota 2 pav.



2 pav. Vartotojo asmeninių duomenų keitimo būsenų diagrama

### 1.3.2. Įtraukti pasą į sąrašus

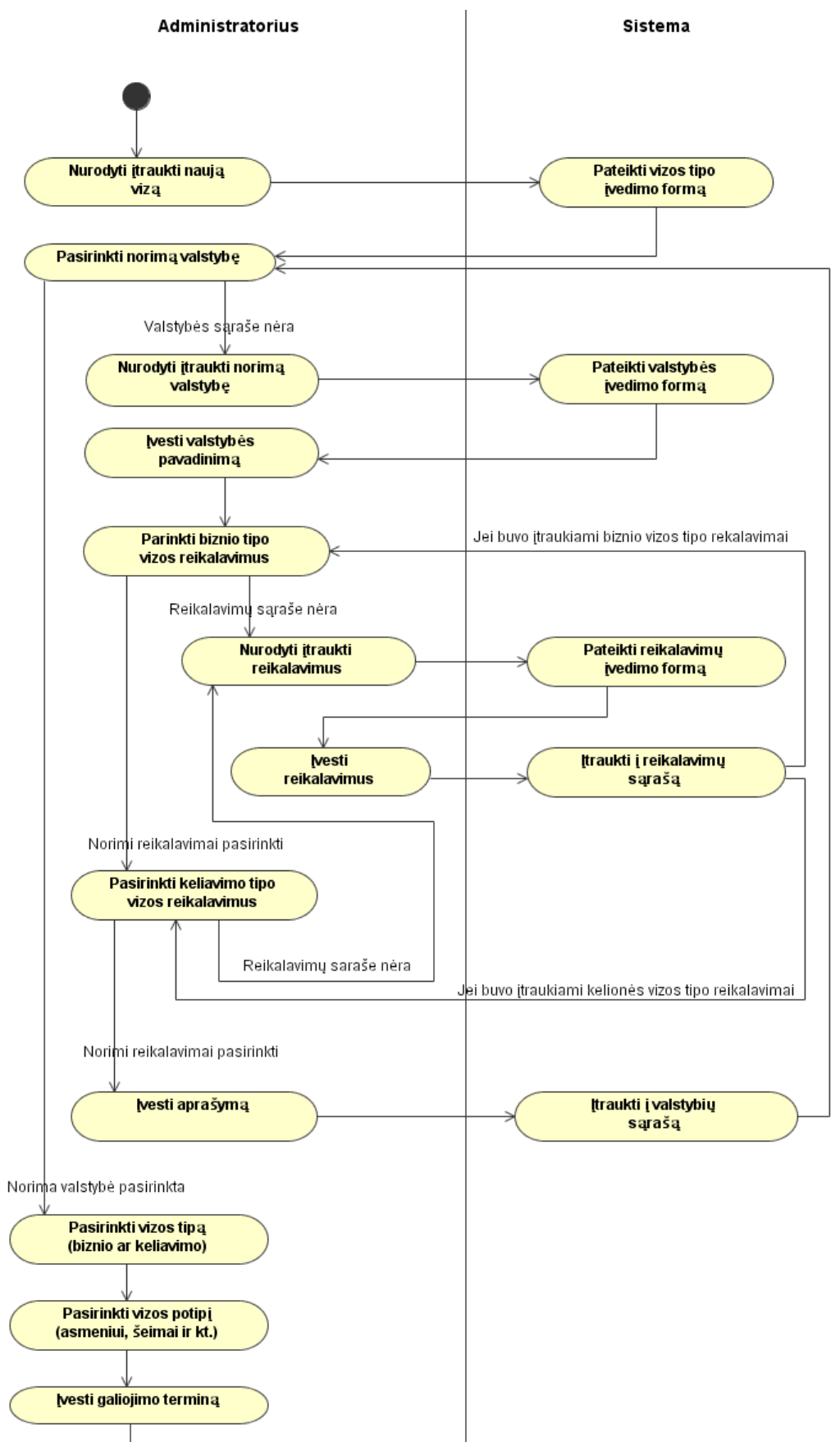
Būsenų diagrama, kai administratorius įtraukia naujus pasus pavaizduota 3 pav.



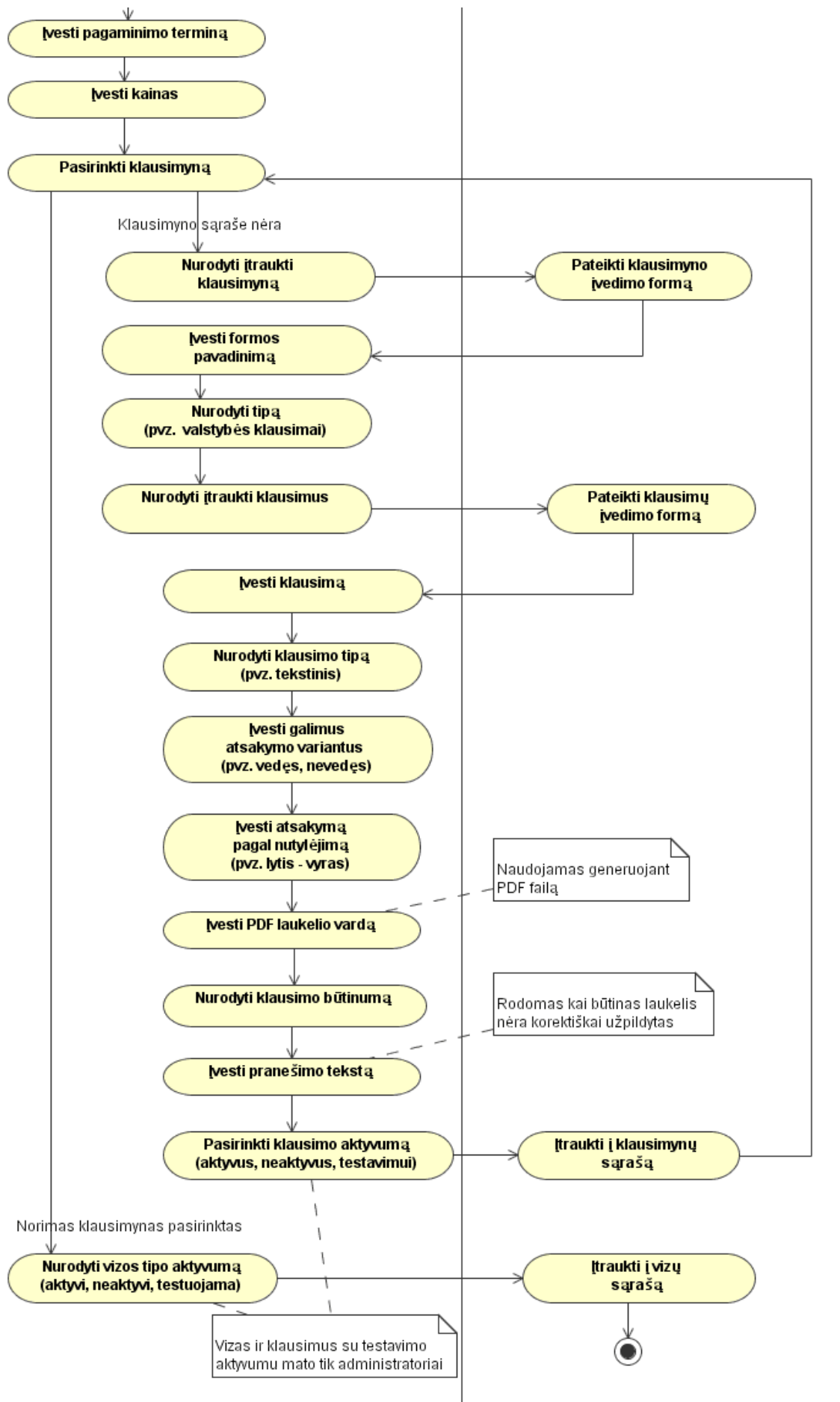
3 pav. Pasų įtraukimo būsenų diagrama

### 1.3.3. Įtraukti vizą į sąrašus

Būsenų diagrama, kai administratorius įtraukia naujus vizų tipus pavaizduota 4,5 pav.



4 pav. Vizų įtraukimo būsenų diagrama

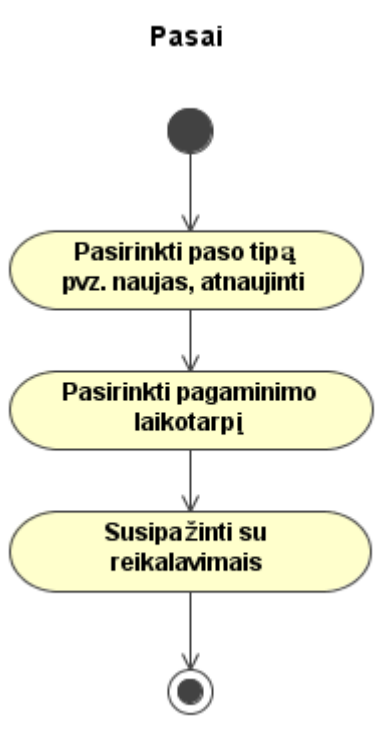


5 pav. Vizų įtraukimo būsenų diagrama tęsinys



### 1.3.4. Įtraukti pasą į užsakymų krepšelį

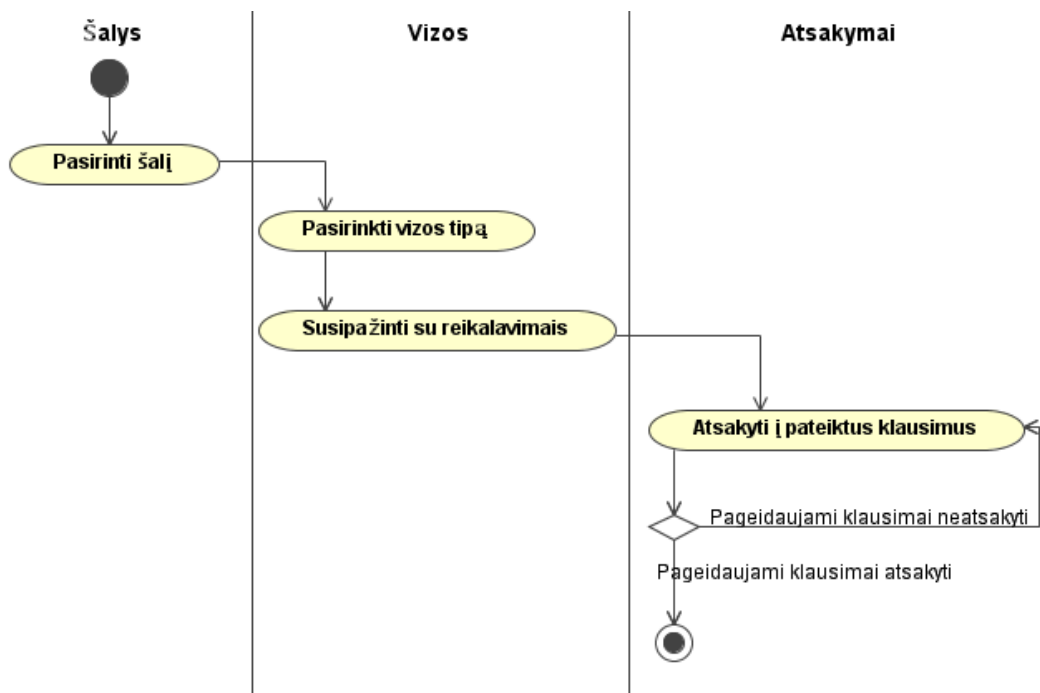
Būsenų diagrama, kai vartotojas įtraukia pasą į užsakymų krepšelį pavaizduota 6 pav.



6 pav. Pasų įtraukimo į užsakymų krepšelį būsenų diagrama

### 1.3.5. Įtraukti vizą į užsakymų krepšelį

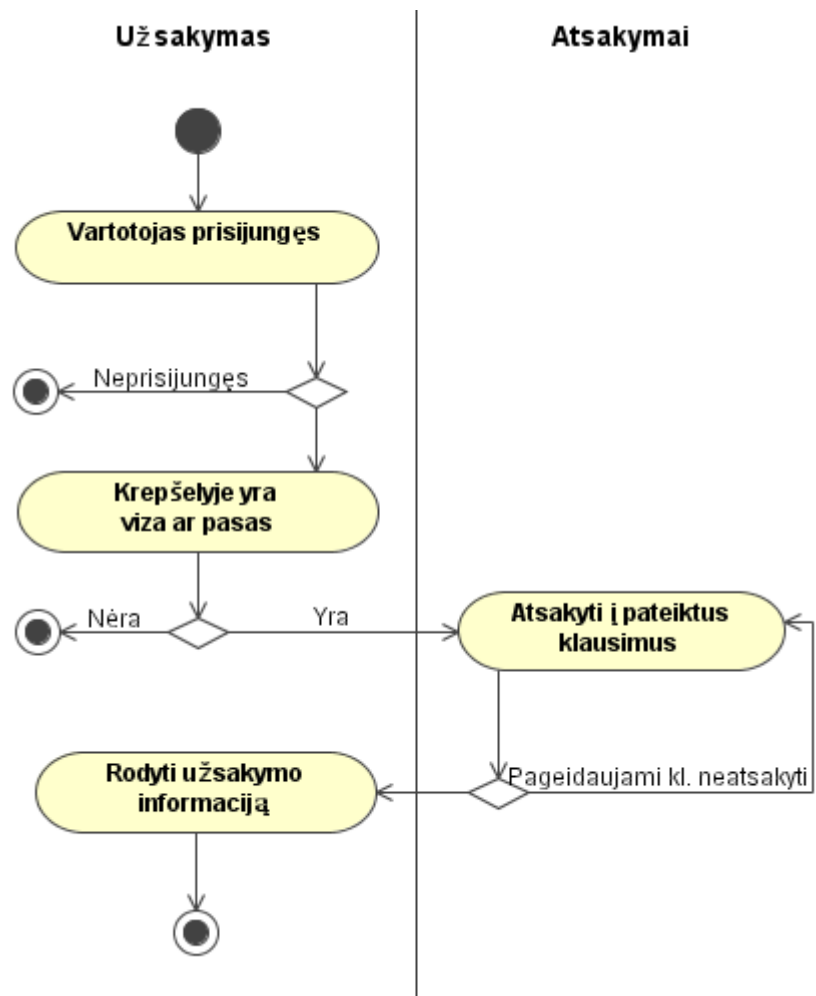
Būsenų diagrama, kai vartotojas įtraukia vizą į užsakymų krepšelį pavaizduota 7 pav.



7 pav. Vizų įtraukimo į užsakymų krepšelį būsenų diagrama

### 1.3.6. Vykdyti užsakymą

Būsenų diagrama, kai vartotojas vykdo užsakymą pavaizduota 8 pav.

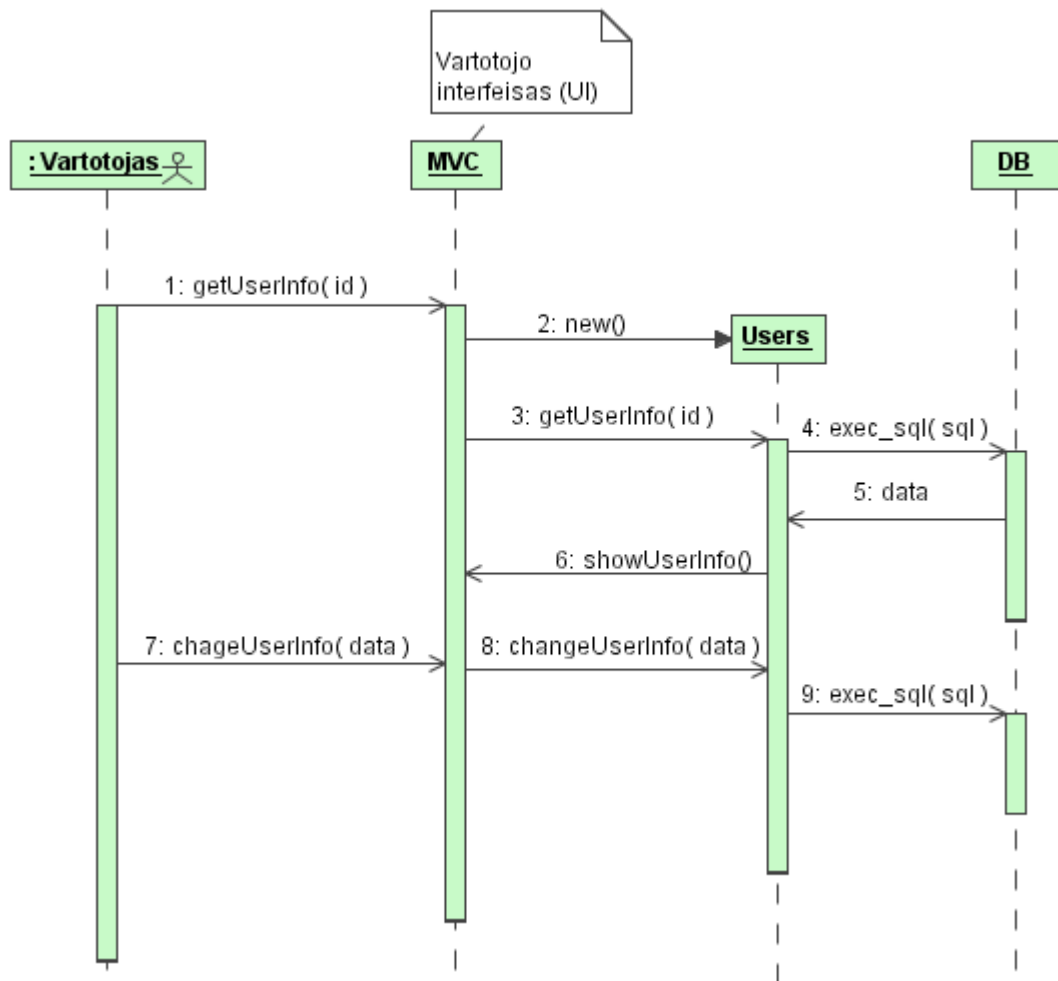


8 pav. Užsakymo vykdymo būsenų diagrama

## 1.4. Sekų diagramos

### 1.4.1. Keisti asmeninius duomenis

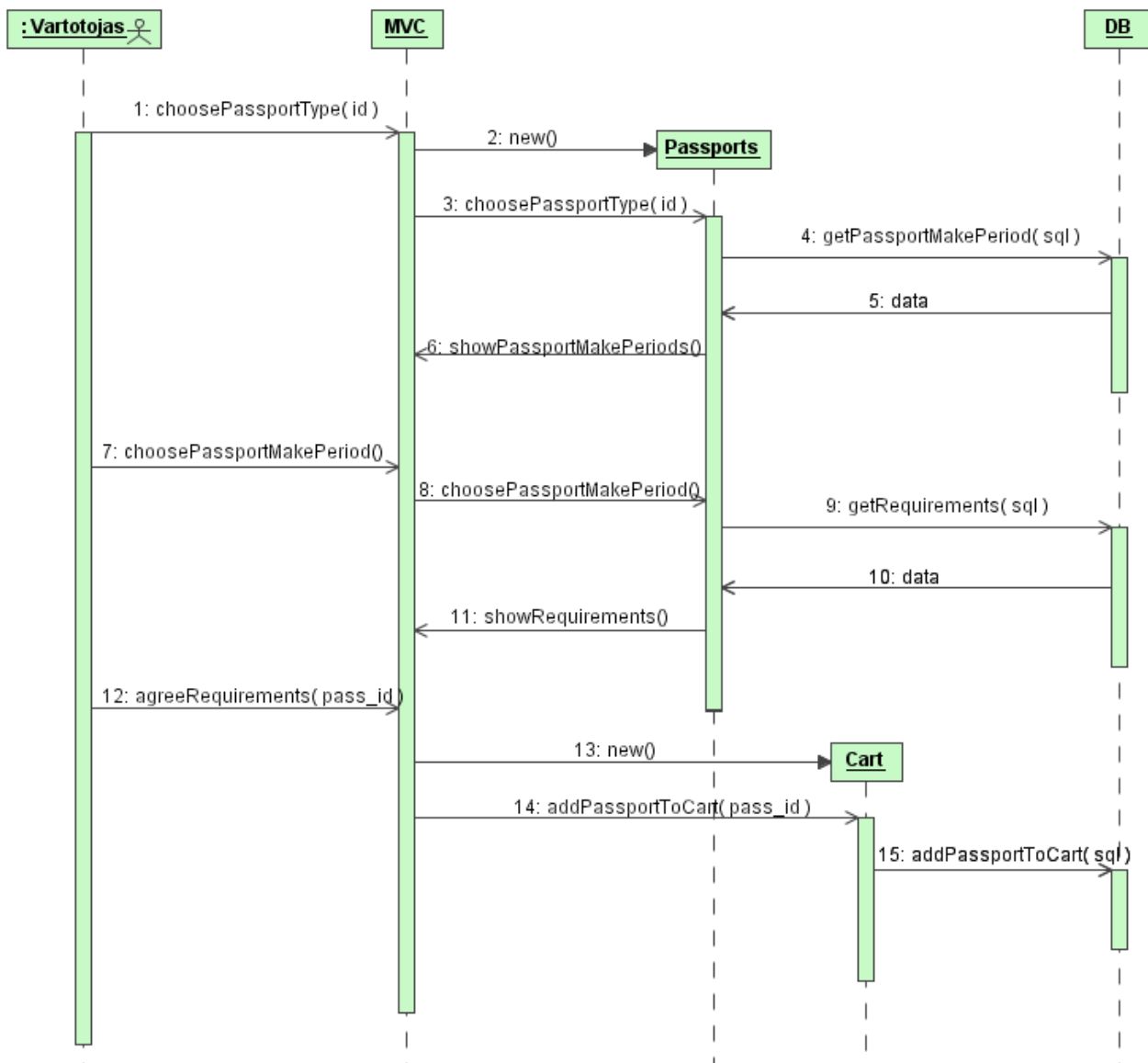
Sekų diagrama kai vartotojas keičia asmeninius duomenis pavaizduota 9 pav.



9 pav. Vartotojo asmeninių duomenų keitimo sekų diagrama

### 1.4.2. Įtraukti pasą į užsakymų krepšelį

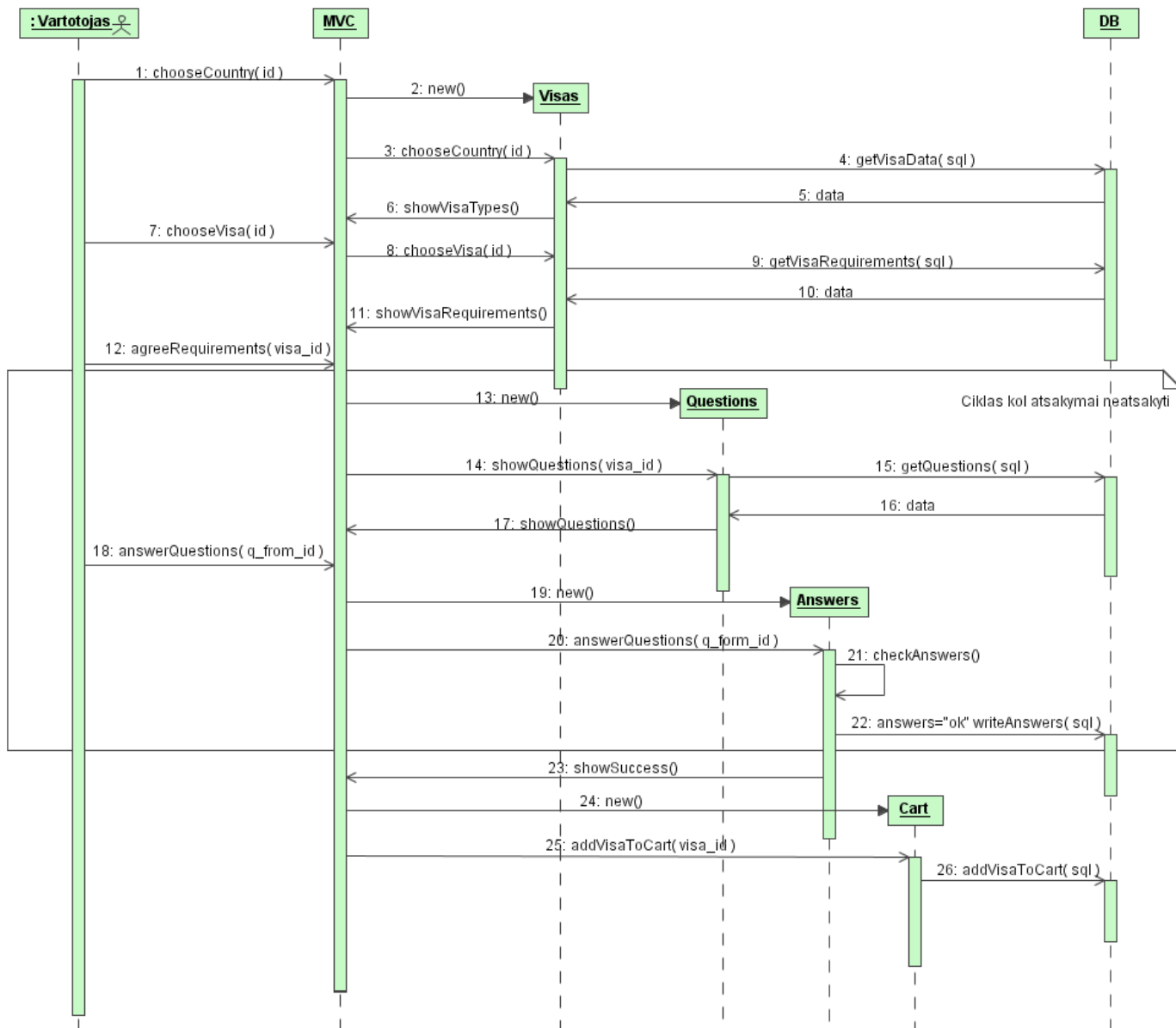
Sekų diagrama, kai vartotojas įtraukia pasą į užsakymų krepšelį pavaizduota 10 pav.



10 pav. Pasų įtraukimo į užsakymų krepšelį sekų diagrama

### 1.4.3. Įtraukti vizą į užsakymų krepšelį

Sekų diagrama, kai vartotojas įtraukia vizą į užsakymų krepšelį pavaizduota 11 pav.



11 pav. Vizų įtraukimo į užsakymų krepšelį sekų diagrama

## 1.5. Klasių diagrama

PIVI IS funkcionavimui reikalingos 6 pagrindinės klasės:

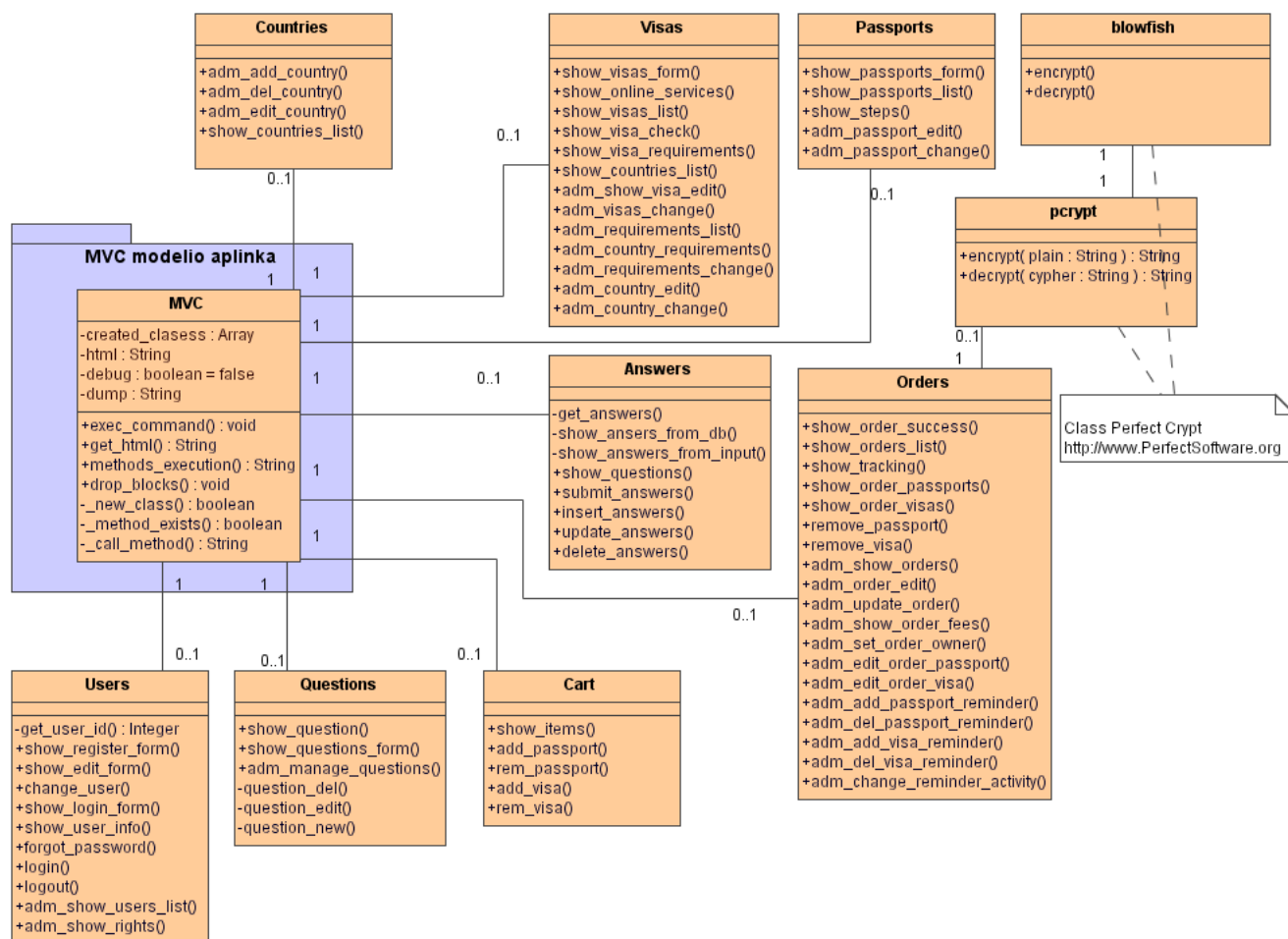
- 1) Visas – vizų tvarkymo klasė
- 2) Passports – pasų tvarkymo klasė
- 3) Answers – atsakymų tvarkymo klasė
- 4) Cart – krepšelio tvarkymo klasė
- 5) Orders – užsakymų apdorojimo klasė
- 6) Users – vartotojų tvarkymo klasė

Svarbių duomenų (vizitinių kortelių duomenys, slaptažodžiai) kodavimui naudojamos 2 Perfect Software duomenų kodavimo klasės[14]:

- 1) pcrypt – abstrakčių metodų klasė, kuri iškviečia skirtingus algoritmus
- 2) blowfish – blowfish[15] kodavimo algoritmo klasė

Taip pat naudojama MVC architektūros klasių paketas (plačiau „2.2. klasių diagrama“).

PIVI IS klasių diagrama pavaizduota 12 pav.



12 pav. PIVI IS klasių diagrama

## 2. MVC architektūros UML diagramos

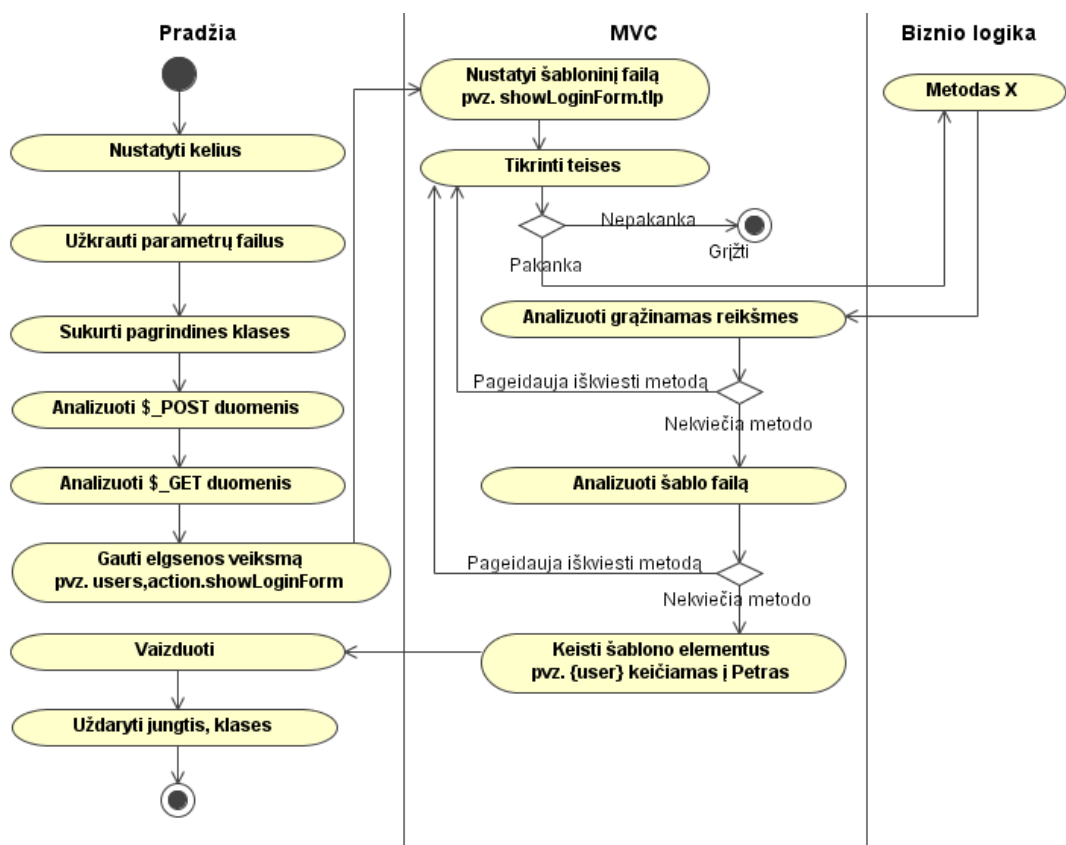
### 2.1. Būsenų diagrama

Pirmiausia yra nustatomi keliai kur randasi parametrų bylos. Visos bylos, esančiame kataloge analizuojamos ir užkraunamos. Atlikus šiuos veiksmus sukuriamos pagrindinės klasės: DB, sesijų valdymo, klaidų apdorojimo, duomenų analizavimo, MVC architektūros.

Sukūrus šias klases perinama į valdiklio klasę, kuri pagal vartotojo pateiktus parametrus nustato šabloninį failą. Prieš iškviečiant metodą tikrinama ar leidžiama jį iškviešti. Neturint tokių teisių šablono turinys pakeičiamas į: „Priėjimas uždraustas!“, o nurodytas metodas neiškviečiamas. Jei vartotojas turėjo teisių metodą iškviešti, tai jis yra iškviečiamas. Gražinamos reikšmės yra analizuojamos ir tikrinama ar nereikia iškviešti kito metodo. Jei taip jis yra iškviečiamas, ir taip vyksta tol, kol nereikia nieko iškviešti. Iškvietus visus metodus analizuojamas šablonų failas. Tikrinama ar jame nėra metodo iškvietimo elementų. Jei taip, metodai iškviečiami. Iškvietus visus metodus, šabloninio failo elementai yra keičiami į kintamųjų reikšmes ir išeinama iš MVC modelio veiklos.

Pagrindinė programa išveda pakeistą šablono failo turinį ir uždaro jungtis su DB, panaikina sukurtas klases.

MVC architektūros būsenų diagrama pavaizduota 13 pav.



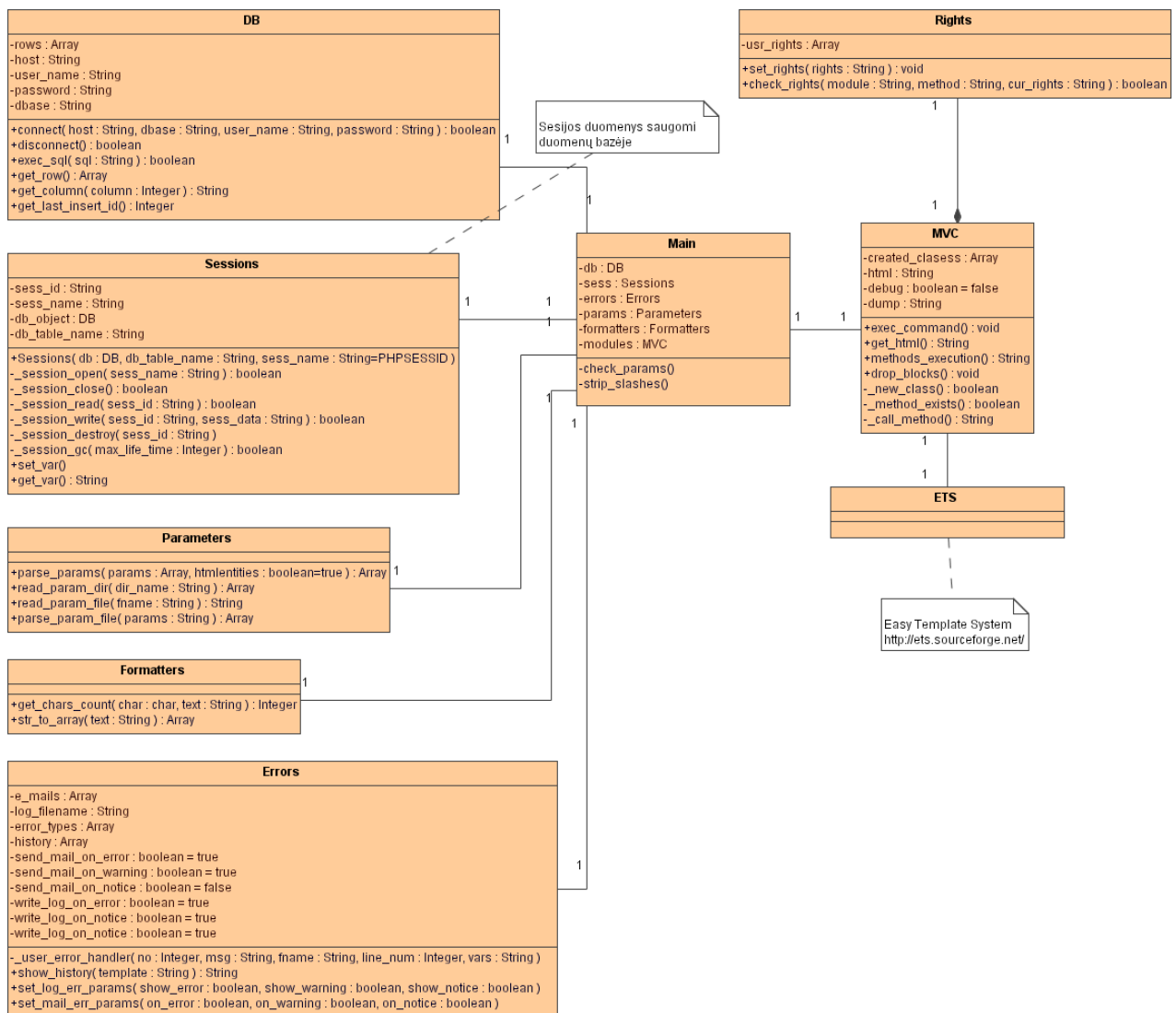
13 pav. MVC architektūros būsenų diagrama

## 2.2. Klasių diagrama

Pilnam MVC modelio funkcionavimui reikalingos 9 klasės:

- 10) DB – duomenų bazės tvarkymo klasė
- 11) Sessions – sesijų tvarkymo klasė
- 12) Parameters – parametrų tvarkymo klasė
- 13) Formatters – formatų tvarkymo klasė
- 14) Errors – klaidų apdorojimo klasė
- 15) Rights – teisių tvarkymo klasė
- 16) MVC – Model View Controller klasė
- 17) ETS – šablonų apdorojimo klasė
- 18) Main – pagrindinė klasė

MVC architektūros klasių diagrama pavaizduota 14 pav.



14 pav. MVC architektūros klasių diagrama



## 2 PRIEDAS. INSTRUKCIJA VARTOTOJAMS

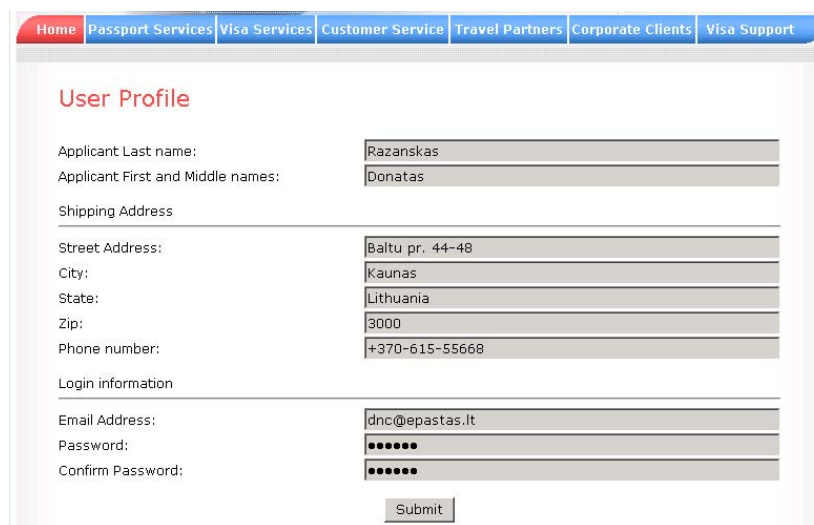
### 1. Instrukcija vizų užsakovams

Šiame skyrelyje pateikiamos instrukcijos šių tipų vartotojams:

- Paprastiems
- Kolektyviniams (Corporate)
- Pareigybiniais (Official)
- Atstovams (Agent)

#### 1.1. Naujo vartotojo registracija

Prisijungimo lange paspaudus mygtuką <Register> pateikiama registracijos forma. Korektiškai<sup>1</sup> užpildžius privalomus laukelius vartotojas prijungiamas prie sistemos ir pranešama apie sėkmingą įregistravimą. Naujo vartotojo registravimo pavyzdys pavaizduotas 1 pav.



The screenshot shows a web browser window with a navigation menu at the top containing: Home, Passport Services, Visa Services, Customer Service, Travel Partners, Corporate Clients, and Visa Support. The main content area is titled "User Profile" and contains a registration form with the following fields and values:

Applicant Last name:	Razanskas
Applicant First and Middle names:	Donatas
Shipping Address	
Street Address:	Baltu pr. 44-48
City:	Kaunas
State:	Lithuania
Zip:	3000
Phone number:	+370-615-55668
Login information	
Email Address:	dnc@epastas.lt
Password:	•••••
Confirm Password:	•••••

At the bottom of the form is a "Submit" button.

1 pav. Naujo vartotojo registracija

#### 1.2. Prisijungimas prie IS

Atsidarykite tinklalapį adresu <http://www.PassportVisaExpress.com>. Prisijunkite įvesdami į atitinkamus laukelius savo vartotojo vardą (Username) ir slaptažodį (Password) ir paspauskite

<sup>1</sup> Slaptažodžių laukeliai sutampa, o slaptažodžio raidžių kiekis ne mažesnis kaip 5 simboliai

mygtuką <Login>. Prisijungimas prie IS pavaizduotas 2 pav. Jei vartotojo vardo dar neturite, t. y. pirmą kartą jungiatės prie IS, nuspauskite mygtuką <Register>.

Prisijungus prie IS leidžiama:

- Įvykdyti užsakymą
- Stebėti užsakymų sąrašą, jų būsenas ir vadybininkų pranešimus

Išimty: vartotojas nebus prijungtas prie sistemos jei administratorius nustatė aktyvumo parametą į neaktyvus (Inactive). Jungiantis vartotojui pranešama, kad jis yra neaktyvus ir siūlo susisiekti su IS administratoriumi.



2 pav. Prisijungimas prie PIVI IS

### 1.3. Slaptažodžio priminimas

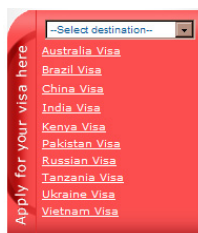
Pamiršus prisijungimo vartotojo slaptažodį paspauskite mygtuką <Forgot>. Atverčiamas langas, kuriame įvedus savo vartotojo vardą ir paspaudus <Submit> naujai sugeneruotas slaptažodis išsiunčiamas į el. pašto dėžutę. Slaptažodžio priminimo forma pavaizduota 3 pav.



3 pav. Prisijungimas prie PIVI IS

### 1.4. Vizos įtraukimas į krepšelį

Norint užsisakyti vizas, pirmiausia reikia pasirinkti valstybę. Valstybių pasirinkimo sąrašas pavaizduotas 4 pav.



4 pav. Valstybių sąrašas

Pasirinkus valstybę yra pateikiamas vizų tipų pasirinkimo sąrašas, pagal vizos tipą, galiojimo laiką ir pagaminimo laikotarpį. Šis sąrašas pavaizduotas 5 pav.

The screenshot shows a web interface for applying for a visa to Vietnam. At the top, there is a navigation menu with 'Home', 'Passport Services', 'Visa Services', 'Customer Service', 'Travel Partners', 'Corporate Clients', and 'Visa Support'. Below this is a breadcrumb trail: 'Visa Check', 'Visa Requirements', 'Visa Application', 'Order Form', and 'Order Confirmation'. The main heading reads 'You are applying for a visa to Vietnam'. A table indicates that visas are required for both Business and Tourist purposes. A note states that the information is for citizens of all countries, except for those of Vietnamese origin. Below the note are three dropdown menus: 'Visa Type' (set to 'Tourist single'), 'Validity' (set to '1 month'), and 'Processing time' (set to '3 days'). A price summary shows a consular price of \$85.00, a service price of \$89.00, and a total of \$174. A 'continue' button is located at the bottom right, and a 'Need assistance? Click here' link is at the bottom left.

5 pav. Vizų tipų pasirinkimo sąrašas

Pasirinkus norimą variantą ir paspaudus mygtuką <continue>, atidaromas reikalavimų vizos išdavimui sąrašas, 6 pav.

The screenshot shows the requirements for a visa to Vietnam. The navigation and breadcrumb trail are the same as in the previous screenshot. The main heading reads 'You are applying visa to Vietnam'. Below this is a section titled 'Please Follow and Complete Instructions 1 - 3'. The first instruction is 'Valid United States Passport', which states that the passport must be signed, valid, and have sufficient blank pages, with an expiration date greater than six months from the date of entry. The second instruction is 'Passport-Style Photographs', which requires one identical passport photograph. The third instruction is 'Non-US citizens', which requires a photo copy of a green card or proof of current US visa status. A 'continue' button is at the bottom right, and a 'Need assistance? Click here' link is at the bottom left.

6 pav. Reikalavimų vizos išdavimui sąrašas

Įsidėmėjus reikalavimus ir paspaudus mygtuką <continue> atidaroma klausimų forma, 7 pav. Užpildžius privalomus laukelius ir paspaudus mygtuką <submit> viza yra įtraukiama į krepšelį.

Home Passport Services **Visa Services** Customer Service Travel Partners Corporate Clients Visa Support

Visa Check Visa Requirements **Visa Application** Order Form Order Confirmation

[Need assistance? Click here](#)

\*Last name Razanskas

\*First and Middle names Donatas

\*Gender  Male  Female

\*Date of Birth 07 January 1981

\*Purpose of your Journey  Tourism  Business  Private

\*Place of Birth Kaunas

7 pav. Klausimų forma

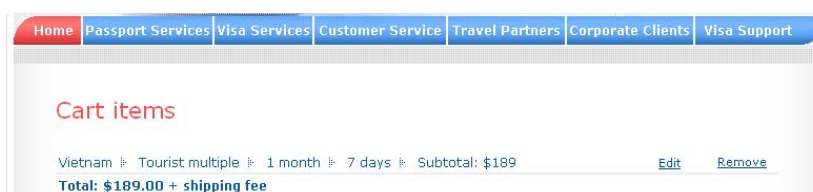
## 1.5. Vizos klausimų redagavimas

Užsakymo krepšelyje, 8 pav., paspaudus mygtuką <Edit items> pateikiamas krepšelyje esančių pasų ir vizų sąrašas.



8 pav. Užsakymų krepšelis

Krepšelyje esančių pasų ir vizų sąrašas pavaizduotas 9 pav.



9 pav. Užsakymų krepšelyje esančios vizos

Prie pasirinktos vizos (šiuo atveju viza į Vietnamą) paspaudus mygtuką <Edit> atverčiama klausimų forma, 7 pav., kurioje pakoregavus įvestus duomenis ir paspaudus mygtuką <Submit> duomenys yra išsaugomi.

## 1.6. Paso įtraukimas į krepšelį

Norint užsisakyti pasą, pirmiausia reikia pasirinkti paso tipą. Pasų pasirinkimo sąrašas pavaizduotas 10 pav.



10 pav. Pasų tipų sąrašas

Pasirinkus pasą pagal tipą (naujas, atnaujinimas, vaikams ir pan.) yra pagaminimo terminų sąrašas, 11 pav.

Home Passport Services Visa Services Customer Service Travel Partners Corporate Clients Visa Support

### New us passport

Passport processing is determined by your date of departure from the United States.  
**When is your departure date from the United States?**

**Urgent Processing:**  
I would like my passport in 24 hours or less!  
\$145 Passport Visa Express fee + \$145 government fee  
(Select if your departure date is within 2 days or within 10 days AND you require a visa)

**Express Processing:**  
I would like my passport in 4 days or less!  
\$105 Passport Visa Express fee + \$145 government fee  
(Select if your departure date is within 4 days or within 10 days AND you require a visa)

**Rush Processing:**  
I would like my passport within 5 to 10 days!  
\$60 Passport Visa Express fee + \$145 government fee  
(Select if your departure date is within 5 to 10 days)

**Regular Processing:**  
I would like my passport in 11 to 30 days!  
\$40 Passport Visa Express fee + \$145 government fee  
(Select if your departure date is within 11 to 30 days)

Need assistance? [Click here](#)

11 pav. Paso pagaminimo laikotarpiai

Pasirinkus pagaminimo laikotarpį, atverčiami paso išdavimo reikalavimai, 12 pav. Juos įsidėmėjus ir paspaudus mygtuką <Proceed> pasas įtraukiamas į krepšelį.

Home Passport Services Visa Services Customer Service Travel Partners Corporate Clients Visa Support

### Urgent New US Passport Service

If your scheduled departure is in less than 2 days or within 10 days AND you require a visa, then you are eligible for our Urgent Passport Processing Service.

**WHO CAN APPLY?**

- Applying for a passport for the first time at age 16 & over
- The prior passport was issued at age 15 or under
- The prior passport was issued more than 15 years ago
- The current & valid passport is **damaged**. YOU MUST WRITE A DETAILED STATEMENT EXPLAINING WHY OR HOW YOUR CURRENT PASSPORT WAS DAMAGED. YOUR ORIGINAL SIGNATURE MUST BE ON THE LETTER.

**FIRST TIME PASSPORT APPLICATION INSTRUCTIONS**

In order to apply and receive a United States passport you must follow the instructions as outlined below. To make the process easier to complete we suggest you print this page and follow the steps in the order presented.

**Step 1**

**Letter of Authorization**

- Please print and complete two (2) copies of the [Letter of Authorization](#).

**Step 2**

**Proof of Citizenship** - You must provide ONE of the following:

12 pav. Paso išdavimo reikalavimai

## 1.7. Paso ar vizos šalinimas iš krepšelio

Užsakymo krepšelyje, 13 pav., paspaudus mygtuką <Edit items> pateikiamas krepšelyje esančių pasų ir vizų sąrašas.



13 pav. Užsakymų krepšelis

Krepšelyje esančių pasų ir vizų sąrašas pavaizduotas 14 pav.



14 pav. Užsakymų krepšelyje esantys pasai ir vizos

Pasas ar viza iš krepšelio pašalinami paspaudus <Remove> mygtuką.

## 1.8. Užsakymo vykdymas

Vykdyti užsakymą galima jei krepšelyje yra bent viena viza ar pasas. Paspaudus mygtuką <Order> nukreipiama į užsakymo formą. Užpildžius privalomus laukelius pranešama apie sėkmingą užsakymą, prekių krepšelis išvalomas. Tačiau šiuo žingsniu viskas nesibaigia. Užsakovas laikas nuo laiko turėtų patikrinti užsakomo būseną ar nėra pranešimų (pvz. kad trūksta vienokių ar kitokių duomenų).

Užsakymo forma pavaizduota 15 pav.

15 pav. Užsakymų krepšelis