



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

**Tomas Urbonas**

**DUOMENŲ BAZĖS KŪRIMAS PANAUDOJANT  
DUOMENŲ PILDYMO FORMŲ ŠABLONUS**

**Magistro darbas**

Darbo vadovas

doc. dr. R. Butkienė

Kaunas, 2011



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

**Tomas Urbonas**

**DUOMENŲ BAZĖS KŪRIMAS PANAUDOJANT  
DUOMENŲ PILDYMO FORMŲ ŠABLONUS**

**Magistro darbas**

Recenzentas

doc. dr. Tomas Blažauskas

2011-05-

Vadovas

doc. dr. R. Butkienė

2011-05-

Atliko

IFN-5/2 gr. stud.  
Tomas Urbonas

2011-05-27

Kaunas, 2011

# TURINYS

<b>1.</b>	<b>ĮVADAS</b> .....	<b>1</b>
<b>2.</b>	<b>BENDRA DUOMENŲ BAZIŲ BEI VALDYMO SISTEMŲ KŪRIMO PRIEMONIŲ APŽVALGA.</b>	<b>3</b>
2.1	PROGRAMINIŲ PRIEMONIŲ APŽVALGA .....	3
2.1.1	<i>Oracle Designer (Oracle Developer Suite)</i> .....	3
2.1.2	<i>Microsoft Access</i> .....	4
2.1.3	<i>Microsoft Visual FoxPro</i> .....	5
2.2	PROGRAMINIŲ PRODUKTŲ APŽVALGOS APIBENDRINIMAS.....	6
2.3	KAIP ŠI PROBLEMA SPRENDŽIAMA PASAULYJE?.....	8
2.4	TIKSLAS – DUOMENŲ BAZĖS BEI JOS VALDYMO SISTEMOS KŪRIMO AUTOMATIZAVIMAS .....	8
2.5	IŠVADOS .....	8
<b>3.</b>	<b>AUTOMATIZUOTO DUOMENŲ BAZĖS BEI JOS VALDYMO SISTEMOS KŪRIMO REALIZAVIMAS</b> .....	<b>9</b>
<b>4.</b>	<b>PROJEKTAS</b> .....	<b>15</b>
4.1	TIKSLAI IR UŽDAVINIAI.....	15
4.2	PROJEKTO KOKYBĖS KRITERIJŲ APIBRĖŽIMAS .....	15
4.3	SISTEMOS KŪRIMO PRIEMONIŲ PARINKIMAS .....	15
4.4	REIKALAVIMŲ SPECIFIKAVIMAS .....	17
4.4.1	<i>Bendrieji reikalavimai</i> .....	17
4.4.2	<i>Funkciniai reikalavimai</i> .....	17
4.4.3	<i>Nefunkciniai reikalavimai</i> .....	18
4.4.4	<i>Projekto apribojimai</i> .....	18
4.5	VARTOTOJAI.....	18
4.6	PANAUDOJIMO ATVEJAI.....	19
4.7	PROJEKTO IŠEIGA.....	21
4.7.1	<i>Atviri klausimai</i> .....	21
4.7.2	<i>Pagaminti komponentai kurie gali būti panaudoti</i> .....	21
4.7.3	<i>Įtaka jau įdiegtoms sistemoms</i> .....	22
4.7.4	<i>Reikalavimai esamų duomenų perkėlimui</i> .....	22
4.8	SISTEMOS PROJEKTAS .....	22
4.8.1	<i>Projekto apribojimai</i> .....	22
4.8.2	<i>Sistemos statinis vaizdas</i> .....	23
4.8.3	<i>Sistemos dinaminis vaizdas</i> .....	25
4.8.4	<i>Išdėstymo vaizdas</i> .....	28
4.8.5	<i>Duomenų vaizdas</i> .....	29
4.8.6	<i>Kokybė</i> .....	30
<b>5.</b>	<b>EKSPERIMENTINIS SISTEMOS TYRIMAS BEI KOKYBĖS ĮVERTINIMAS</b> .....	<b>31</b>
5.1	EKSPERIMENTINIO SISTEMOS TYRIMO TIKSLAI .....	31
5.2	REALIZUOTOS INFORMACINĖS SISTEMOS FUNKCINIS APRAŠAS .....	31
5.2.1	<i>Sistemos paskirtis</i> .....	31
5.2.2	<i>Sistemos atliekamos funkcijos</i> .....	31
5.2.3	<i>Bendri sistemos principai</i> .....	31
5.3	EKSPERIMENTINIO TYRIMO KOKYBĖS VERTINIMAS (TESTAVIMAS).....	34
5.3.1	<i>Testavimo planas</i> .....	34
5.3.2	<i>Testavimo strategija</i> .....	35
5.3.3	<i>Testavimo resursai ir rezultatai</i> .....	36
5.3.4	<i>Testavimo eiga</i> .....	37
5.3.5	<i>Testavimo išvados</i> .....	38
<b>6.</b>	<b>IŠVADOS</b> .....	<b>39</b>
<b>7.</b>	<b>LITERATŪRA</b> .....	<b>40</b>
<b>8.</b>	<b>SANTRAUKA ANGLŲ KALBA</b> .....	<b>41</b>
<b>9.</b>	<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS</b> .....	<b>42</b>
<b>10.</b>	<b>PRIEDAI</b> .....	<b>43</b>
10.1	1 PRIEDAS. DUOMENŲ BAZĖS LENTELIŲ APRAŠAI .....	43

10.2	2 PRIEDAS. SISTEMOS TESTAVIMO DOKUMENTACIJA .....	45
------	---	----

## 1. ĮVADAS

Šiais modernių kompiuterinių technologijų laikais kiekviena organizacija turi savo kompiuterinės bei programinės įrangos bazę. Didėjant kaupiamos informacijos kiekiui ir augant jos analizavimo pagal įvairius kriterijus poreikiui bei siekiant užtikrinti sklandų organizacijos darbą, veiksmingumą ir našumą atsiranda būtinybė turėti programinę įrangą skirtą informacijos apdorojimui. Kokią PĮ pasirinkti sprendžia pati organizacija, priklausomai nuo norimo apdoroti informacijos kiekio, PĮ vartotojų skaičiaus, PĮ kainos, organizacijoje naudojamos techninės įrangos bei naudojamos operacinės sistemos.

Galima rinktis jau gerai žinomus produktus. Jei organizacijai dėl kokių nors priežasčių (per didelė kaina; programoje yra modulių kurie nebus naudojami, bet už juos reikia mokėti; nėra kai kurių reikalingų funkcionalumų ir t.t.) netinka esamos duomenų apdorojimo programos, galima užsakyti individualią programinę įrangą, pritaikytą konkrečiai organizacijai, kurią gali pagaminti programavimo paslaugas teikiančios firmos ar pavieniai programuotojai.

Duomenų bazės projektuojamos pagal specifinius reikalavimus ir tai dažniausiai daroma rankiniu būdu. Suprojektuotos duomenų bazės valdymui kuriamos programinės priemonės. Žinoma, programinės priemonės pritaikytos tik tai duomenų bazei valdyti. Gautas rezultatas dažniausiai tenkina vartotoją, tačiau neilgai. Nuolat besikeičiantys reikalavimai duomenims didina atotrūkį tarp norimos apdoroti ir galimos apdoroti informacijos. Taigi išlaikyti duomenų bazę suderintą su poreikiais yra sudėtinga. Nuolat besikeičiantys poreikiai didina informacinės sistemos palaikymo sąnaudas. O tai smulkaus verslo įmonėms gali tapti nepakeliamu našta. Kyla poreikis pačiam sistemos vartotojui pakoreguoti ją taip, kad sistema tenkintų pasikeitusius reikalavimus duomenų apdorojimui.

Šiame darbe pristatomas duomenų bazės kūrimas naudojant vartotojo sukurtus duomenų pildymo šablonus. Pagal vartotojo sukurtus šablonus keičiasi ir informacinės sistemos vartotojo sąsaja skirta duomenims apdoroti.

Kuriama informacinė sistema bus orientuota į vartotojus, kurie neturi pakankamai žinių apie duomenų bazės projektavimą bei kūrimą. Šis įrankis skirtas pačiam sukurti duomenų bazę, viskas bus daroma pagal duomenų pildymo šablonus. Šabloną paruoš pats vartotojas surašydamas kokius duomenis bus kaupiami apie vieną ar kitą jam reikalingą objektą.

Vartotojas net nežinos kiek ir kokių lentelių bus duomenų bazėje, jam net nereikės kurti lentelių, tai darys pati sistema, kai tai bus būtina. Jam reikės tik nurodyti kokius duomenis norės saugoti. Duomenų pildymo šablonai kuriami dialogo pagalba: sistema pateiks keletą

klausimų, būtinų nustatyti tikslų duomenų aprašymą (reikšmės tipas, pasikartojančios ar nulinės reikšmės ir t.t.).

Kadangi kuriamai sistemai svarbus dinamiškumas (interaktyvumas), sistemą planuojama realizuoti interneto technologijomis. Taip bus palengvinamas vartotojo darbo vietos paruošimas, reikės tik teisingai sukongūruoti interneto naršyklę. Tinkamai pasirūpinus HTTP serverio saugumu būtų galima jį padaryti prieinamą ne tik iš vietinio tinklo. Tokiu būdu atsirastų puiki galimybė šią sistemą naudoti ne tik vienoje organizacijoje, bet nuo pagrindinės organizacijoje nutolusiuose filialuose.

Projektuojant sistemą pasirinktas komponentinio projektavimo metodas, kurio pagrindinis privalumas – pakartotinio komponentų panaudojimo galimybė. Dėl šios savybės padidėja sistemos kūrimo produktyvumas, tuo pačiu sutrumpėja projekto kūrimo ciklas. Komponentinio projektavimo principas leidžia ateityje atsiradus poreikiui minimaliomis sąnaudomis prijungti naujus modulius.

Informacinė sistema kuriama žingsniniu kūrimo metodu. Sistemos kūrimas išskaidytas į žingsnius, kuriuose realizuojamas vis kitas panaudojimo atvejis. Procesui prasidėjus, reikalavimų stengiamasi nekeisti, nors tai ne visada pavyksta [6].

Projekto dalyje identifikuojami sistemos projekto komponentai bei jų sąveikos. Pateikiamos komponentų specifikacijos, apibrėžiami kuriamos sistemos aktoriai, veiklos sudėtis, išskiriami panaudojimo atvejai. Pateikiami sistemos statinis, dinaminis bei išdėstymo vaizdai, loginė duomenų bazės struktūra. Sudaroma informacinės sistemos architektūra.

## 2. BENDRA DUOMENŲ BAZIŲ BEI VALDYMO SISTEMŲ KŪRIMO PRIEMONIŲ APŽVALGA

Duomenų bazės projektuojamos pagal specifinius reikalavimus ir tai dažniausiai daroma rankiniu būdu. Suprojektuotos duomenų bazės valdymui kuriamos programinės priemonės. Žinoma, programinės priemonės pritaikytos tik tai duomenų bazei valdyti. Gautas rezultatas dažniausiai tenkina vartotoją, tačiau neilgai. Nuolat besikeičiantys reikalavimai duomenims didina atotrūkį tarp norimos apdoroti ir galimos apdoroti informacijos. Taigi išlaikyti duomenų bazę suderintą su poreikiais yra sudėtinga. Nuolat besikeičiantys poreikiai didina informacinės sistemos palaikymo sąnaudas. O tai smulkaus verslo įmonėms gali tapti nepakeliama našta. Kyla poreikis pačiam sistemos vartotojui pakoreguoti ją taip, kad sistema tenkintų pasikeitusius reikalavimus duomenų apdorojimui.

### 2.1 Programinių priemonių apžvalga

Yra sukurta daugybė duomenų valdymo sistemų, realizuojančių įvairius sprendimus ir metodus, kuriuos galima pritaikyti skirtingo pobūdžio ir dydžio organizacijose. Vienose sistemose visiems veiksams atlikti naudojamos tiesiai programos lange rašomos SQL užklausos, kitose yra puikiai paruošta grafinė vartotojo sąsaja, kurioje nesunku vykdyti esamų duomenų bazių pildymą, ruošti ataskaitas, o SQL užklausas generuoja ir vykdo paruošti programų moduliai. Žinoma ir šiose sistemose paliekama galimybė pačiam rašyti SQL užklausas, taip galima gauti tikslesnį rezultatą nei naudojantis programos paruoštomis formomis.

Žemiau aprašomi keli programiniai sprendimai teikiantys galimybę:

- aprašyti DB struktūrą;
- sudaryti specialias formas, skirtas duomenų įvedimui, peržiūrai, modifikavimui;
- kurti įvairių lygių meniu sistemas;
- įvairiai manipuluoti duomenimis;
- apdoroti užklausas, pateikiamas SQL kalba.

Su šiais programiniais sprendimais teko dirbti vykdant kai kuriuos programinius sprendimus darbovietėje arba studijų metu.

#### 2.1.1 Oracle Designer (Oracle Developer Suite)

*Oracle Designer* pristatoma kartu su *Oracle Developer Suite*. Juo galima atlikti verslo procesų modeliavimą, sistemų analizę, programinės įrangos projektavimą ir kūrimą. *Oracle Designer* suteikia daugiaabonentį priėjimą prie saugyklos (*angl. multi-user repository*) ir yra glaudžiai susijęs su *Oracle Forms Developer* duomenų bazės valdymo sistemos kūrimo

priemone. Tokiu būdu *Oracle Designer* leidžia organizacijoms kurti ir greitai pristatyti kliento – serverio modeliu sukurtas sistemas.

Norint dirbti šia programine priemone reikia išmanyti kaip kuriamos ir valdomos duomenų bazės. DB kūrimą šios sistemos pagalba galime suskirstyti tokiais etapais[3]:

- Procesų diagramos braižymas. Joje abstrakčiai reikia nurodyti kokie duomenys kaip kelias bus DB, t.y. duomenų įvedimas, saugojimas saugyklose (lentelėse), skaitymas;
- Esybių – ryšių diagramos braižymas. Joje reikia nubraižyti kaip tarpusavyje sąveikauja anksčiau nurodytos esybės (lentelės), t.y. kokie atributai (stulpeliai) sudaro esybę ir kokių atributų pagalba esybės jungiamos tarpusavyje;
- Funkcijų hierarchijos diagramos braižymas. Ši diagrama automatiškai įtraukia visus procesus iš anksčiau nubraižytos procesų diagramos. Vartotojui reikia nurodyti kaip kiekvienoje funkcijoje naudojami duomenys (esybės ir jų atributai);
- Duomenų srautų diagramos braižymas. Joje nurodoma kokie duomenys kaip keliauja ir kas jais gali manipuliuoti (nurodomos išorinės esybės ir objektai);
- Automatizuotas duomenų lentelių projektavimas. Šiame etape vartotojas peržiūri kokias lenteles projektuojamoje DB ruošiasi kurti sistema (stulpelių sąrašas), vėliau sugeneruojamas scenarijus, kurį įvykdžius bus sukurta DB su visomis projektuojamomis lentelėmis;
- Loginės ir fizinės duomenų bazės kūrimas. Pagal sugeneruotą scenarijų sistema grafiškai atvaizduoja projektuojamos DB schemą (pavaizduojamos lentelės su ryšiais). Toliau generuojama fizinė duomenų bazė;

Privalumai:

- Realizuota galimybė generuoti grafinę DB valdymo programą.

Trūkumai:

- šis įrankis skirtas tik darbui su *ORACLE* duomenų bazių serveriu;
- Serveryje turi būti įdiegtas programinis paketas Oracle Designer Repository;
- Veikia tik MS Windows operacinėse sistemose;
- Didelė produkto kaina;

### **2.1.2 Microsoft Access**

*Microsoft Access* – reliacinė duomenų bazės valdymo sistema, skirta smulkaus verslo įmonėms.

*MS Access* yra pritaikyta dirbti viename kompiuteryje (*angl. desktop database system*),. Tačiau panaudojus ODBC prisijungimo būdą galima prijungti ir naudoti *MS Access* duomenų bazę iš kito kompiuterio. *MS Access* yra taikomųjų programų paketo *Microsoft Office* dalis.



Privalumai:

- Gali dirbti su keliais informacijos šaltiniais – *MS Excel* bylomis, kituose kompiuteriuose esančiomis *MS Access* duomenų bazėmis, *MS SQL Server* duomenų bazių serveriu.
- Į duomenų bazę galima įkelti įvairias bylas;
- Automatinis duomenų tipo nustatymas – nustato ar tai data, laikas ar bendras tipas (tekstas);
- Ataskaitų rengimas režimu „tai ką matote atitinka tai ką gausite“ (*angl. what you see is what you get*)

Trūkumai:

- Veikia tik MS Windows operacinėse sistemose;
- Didelė produkto kaina, nes parduodamas tik su pilnu *Microsoft Office* paketu;
- Duomenų bazės ribojamas dydis 2 GB.

### **2.1.3 Microsoft Visual FoxPro**

*Microsoft Visual FoxPro* duomenų bazių kūrimo sistema, kurios pagalba galima greitai sukurti didelio našumo vieno kompiuterio (*angl. desktop dabase system*), kliento – serverio modeliu ar internete veikiančią duomenų bazę. *FoxPro* įdiegtas galingas duomenų valdymo mechanizmas įgalina greitai ir patogiai apdoroti didelius duomenų kiekius iš duomenų bazės ar iš internete patalpintų XML failų. Objektinis programavimas įgalina pakartotinį komponentų panaudojimą.

Privalumai:

- Turi aukšto lygio programavimo kalbą;

Trūkumai:

- Veikia tik MS Windows operacinėse sistemose;
- Didelė produkto kaina;
- Duomenų bazės ribojamas dydis 2 GB.

## 2.2 Programinių produktų apžvalgos apibendrinimas

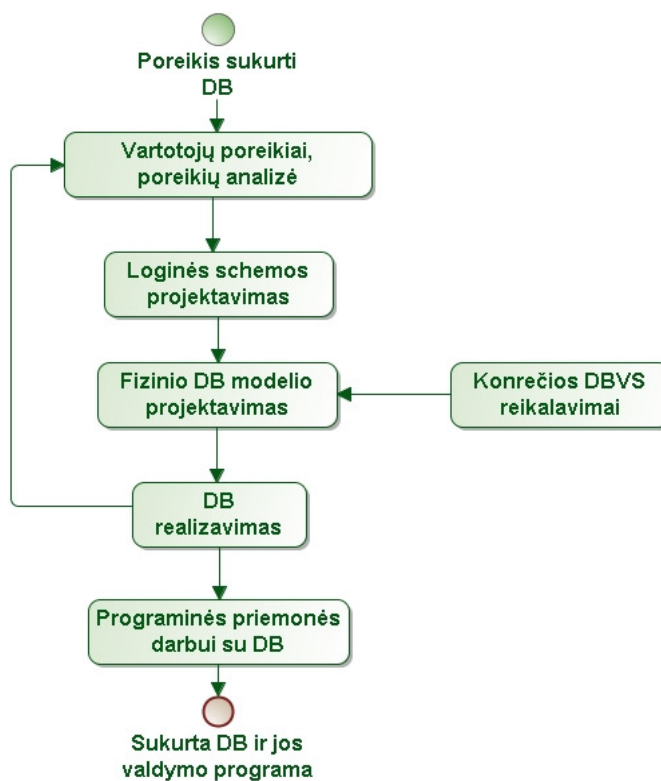
Lentelė Nr. 1 Programų apibendrinimas

Produktas	Oracle Designer (Oracle Developer Suite)	MS Access	FoxPro
Kaina	15477 LTL	Parduodamas su MS Office paketus. Paketo kaina 499 USD	Nuo 495 USD
DBVPG <sup>1</sup>	Taip	Taip	Taip
Valdymo programos formų keitimas darbo metu	Ne	Ne	Ne
DB automatinis generavimas	Taip	Ne	Ne
DBKSDM <sup>2</sup> koregavimas sistemos darbo metu	Ne	Taip	Taip
Duomenų bazės dydis	Neribojamas	2 GB	2 GB

Apžvelgtuose produktuose naudojamas tradicinis duomenų bazės kūrimo metodas, kuomet duomenų bazės gyvavimo ciklas susideda iš duomenų rinkimo (siekiant nustatyti vartotojų poreikius), duomenų bazių elementų kūrimo – projektavimo (loginės struktūros), DBVS pasirinkimas duomenų bazės palaikymui, programinių priemonių parinkimas darbui su duomenų baze, pakartotinis poreikių peržiūrėjimas sukurtos duomenų bazės kontekste. Tradicinis duomenų bazės kūrimo principas pateikiamas 1 pav.

<sup>1</sup> Duomenų bazių valdymo programų generavimas

<sup>2</sup> Duomenų bazės koregavimas sistemos darbo metu



**1 pav. Tradicinis duomenų bazės kūrimas**

Norint šiuo metodu sukurti duomenų bazę nepakanka žinoti, kad duomenys saugomi dvimatėje duomenų lentelėje, kurios stulpeliai (atributai) nurodo kokia informacija juose saugoma ir kad visi įrašai saugomi eilutėse. Teoriškai duomenų bazė galėtų būti viena didelė duomenų lentelė. Bet tuomet tokioje lentelėje atsirastų nereikalingų duomenų dubliavimų o kiti lentelės laukeliai gali likti neužpildyti. Tai ap sunkintų duomenų įvedimą, paiešką bei atnaujinimą. Siekiant išvengti duomenų pertekliškumo lentelė skaidoma į kelias mažesnes duomenų lenteles tai vadinama normalizavimu.

Norint sukurti programinę priemonę duomenų bazės valdymui reikia išmanyti, kaip duomenys iš duomenų bazės imami vienu iš kelių lentelių arba kaip vienu metu įrašomi į kelias skirtingas lenteles.

Tačiau ką daryti vartotojui kuriam reikia programinio sprendimo, bet jis neturi pakankamai žinių sukurti duomenų bazę bei jos valdymui skirtą programinę priemonę? Žinoma, vartotojas gali įsigyti jau paruoštą produktą arba pasinaudoti koku nors nemokamu jo poreikius atitinkančiu produktu. Bet ką daryti kai įsigytas produktas nebetenkins poreikių? Modifikuoti turimą ar ieškoti naujo poreikius atitinkančio produkto? Suradus naują produktą iškyla problema su sukauptų duomenų perkėlimu. Vėl reikia pagalbos iš tos srities specialistų.

### **2.3 Kaip ši problema sprendžiama pasaulyje?**

Programinių priemonių kūrėjams visada naudingas pagamintų programinių priemonių tobulinimas, nes tai jų pragyvenimo šaltinis.

Strateginis duomenų bazių planavimas – bandymas numatyti informacinius poreikius ilgesniam laiko periodui ne visada pasiteisina. Gali keistis vartotojo poreikiai, įstatyminė bazė. Tai gali turėti įtakos esamai informacinei sistemai. Dažniausi pakeitimai duomenų bazėje: pakeisti duomenų lauko tipą (plotį), įterpti papildomą stulpelį lentelėje ar pašalinti nereikalingą. Tokiu atveju turimai sistemai atnaujinti reikalingos papildomos sąnaudos:

- Gaištamas darbuotojų laikas pageidaujiamiems pakeitimams aprašyti;
- Negalima tinkamai naudotis informacine sistema;
- Mokamos programavimo paslaugos;

Net ir norint pakeisti tokią smulkmeną kaip nulinių ar unikalių reikšmių įvedimas reikia keisti ne tik duomenų lentelės struktūrą bet ir duomenų manipuliavimo formos struktūrą. Formose įvestos reikšmės tikrinamos dar prieš įrašant juos į duomenų lentelę. Taip daroma siekiant padaryti patogią vartotojo sąsają, kurioje apie aptiktas klaidas pranešama aiškiai ir vienareikšmiai suprantamu pranešimu.

### **2.4 Tikslas – duomenų bazės bei jos valdymo sistemos kūrimo automatizavimas**

Sukurti algoritmą duomenų bazės bei jos valdymo sistemos kūrimo automatizavimui. Pagal šį algoritmą realizuota sistema vartotojas, neišmanantis apie duomenų bazių kūrimą, galėtų sukurti duomenų bazę bei jos valdymo sistemą pritaikytą savo reikmėms. Galėtų bet kada modifikuoti duomenų bazės struktūrą – pridėti, pašalinti stulpelius. Duomenų bazės struktūra koreguojama pagal šablonus. Šablonų sudėtis saugoma duomenų bazės lentelėse. Pagal aprašytą algoritmą vartotojas galėtų sukurti duomenų įvedimo šabloną, kuris jam atrodytų patogus ir suprantamas.

#### **Darbo uždaviniai:**

- Sukurti algoritmą skirtą duomenų bazės lentelių kūrimui, kurios tenkintų reliacinėms duomenų bazėms taikomas normalines formas.
- Suprojektuoti informacinę sistemą pagal sukurtą algoritmą;
- Realizuoti informacinę sistemą;
- Atlikti sistemos pagalba sukurtos duomenų bazės įvertinimą – jos atitikimą normalinėms formoms.

### **2.5 Išvados**

Šiuolaikinės duomenų bazių valdymo sistemos pritaikytos sukurti numatytos struktūros duomenų bazę bei jos valdymo sistemą. Duomenų bazės struktūrą galima keisti veikiančios

programos pagalba, tačiau formos pritaikyti prie pakeistos duomenų lentelės pritaikyti neįmanoma.

Pasirinktas duomenų bazės kūrimo pagal šablonus principas. Šablonas skirtas saugoti informacijai apie konkretų objektą<sup>3</sup>.

### 3. AUTOMATIZUOTO DUOMENŲ BAZĖS BEI JOS VALDYMO SISTEMOS KŪRIMO REALIZAVIMAS

Vartotojas gali sukurti duomenų įvedimo šabloną tokį, koks jam atrodys patogus. Tarkime vartotojo sukurto šablono pagalba sukurta ir užpildyta lentelė atrodys kaip pateikta pavyzdyje Lentelė Nr. 1.

Lentelė Nr. 2 Duomenų lentelės pavyzdys

Tabelio numeris	Darbuotojas	Pareigos	Atlikta užduotis	Klientas
517	R.Žymeklis	Inžinierius	Kondicionieriaus remontas	UAB „Be pavadinimo“
519	G.Rašiklis	Meistras	Kondicionieriaus įrengimas	UAB „Pavadinimas“
519	G.Rašiklis	Meistras	Kondicionieriaus remontas	UAB „Pavadinimas“
517	R.Žymeklis	Inžinierius	Filtro keitimas	UAB „Nauja UAB“

Dėl neteisingo duomenų bazės projektavimo gali kilti sekančios problemos, dar vadinamos anomalijomis:

**Duomenų perteklius.** Pavyzdyje kartojasi darbuotojai, pareigos, atliktos užduotys bei klientai.

**Atnaujinimo anomalija.** Jeigu reikėtų pakeisti kliento pavadinimą, tai reikėtų keisti visuose įrašuose. Bent viename įrašė nepakeitus pavadinimo atsiras duomenų neatitikimas.

**Pašalinimo anomalija.** Tarkime vartotojas sugalvojo pašalinti vartotoją R.Žymeklis, tokiu atveju pašalinus įrašą, bus prarasta informacija apie atliktas užduotis ir klientus.

**Įvedimo anomalija.** Pavyzdinėje lentelėje norint įvesti priimtą darbuotoją reikia iš kart priskirti ir užduotį bei nurodyti klientą. Vadinasi darbuotoją bus galima įvesti tik tada kai jis bus atlikęs bent vieną užduotį.

Šių problemų galima išvengti projektuojant normalizuotą duomenų bazę. Bet kaip tai padaryti vartotojui neišmanančiam duomenų bazės kūrimo?

Normalizavimo procesas yra ne kas kita kaip „skaidyk ir valdyk“ principas. Planuojama panaudojant tokį principą siekti sukurti normalizuotą duomenų bazę.

#### Šablono kūrimas

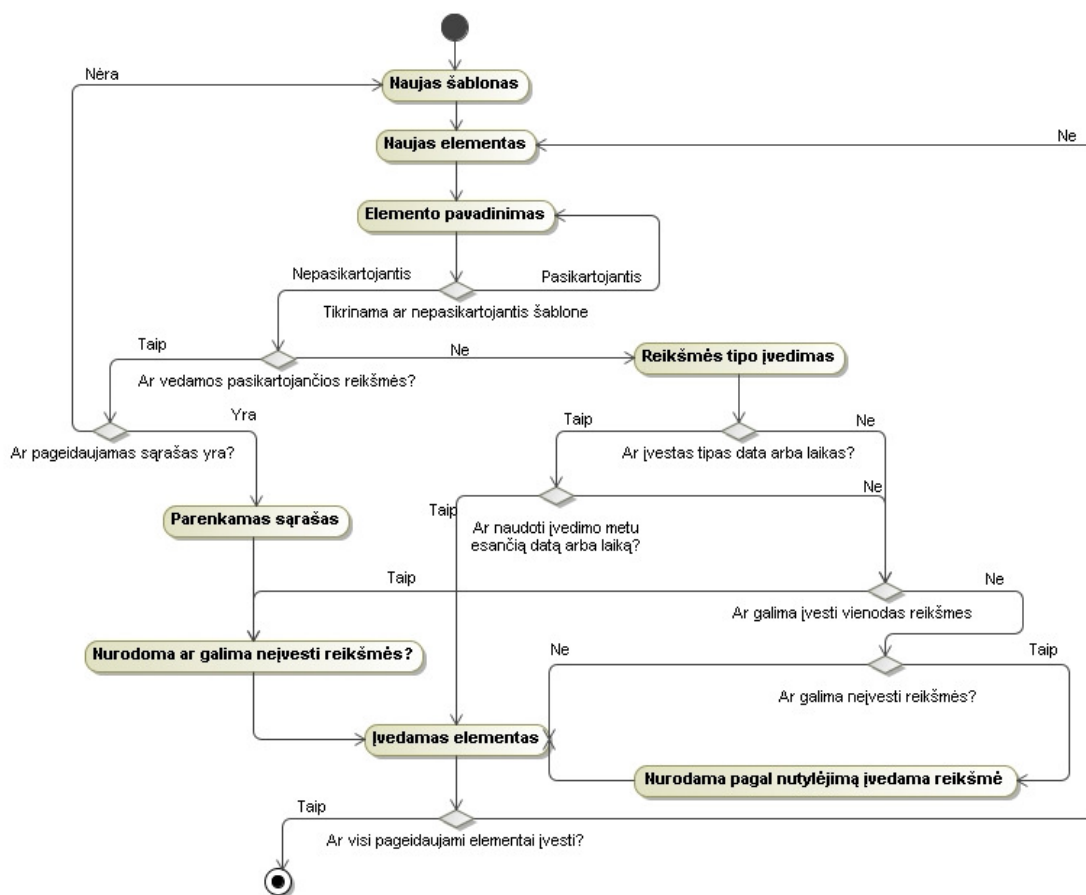
<sup>3</sup> Objektas – kiekvienas daiktas, dalykas, reiškiny [7]

Siekiant, kad duomenų bazė būtų normalizuota, duomenų lentelėje negali būti pasikartojančių reikšmių, kaip kad pavyzdyje Lentelė Nr. 1. Tuo tikslu vartotojui norint į šabloną įtraukti naują elementą pateikiamas klausimas *Ar laukeliuose bus saugomos pasikartojančios reikšmės?* Jei vartotojo atsakymas teigiamas vartotojui siūloma įvedamų duomenų reikšmes įvesti ne rankiniu būdu, bet pasirinkti iš sąrašo. Siūloma rinktis sąrašus jau iš esančių sistemoje sukurtų šablonų. Jei tokio nėra siūloma pirmiau sukurti pageidaujamų reikšmių sąrašą apibūdinantį šabloną. Kiekvieno šablono pagalba vedamiems duomenims saugoti sukuriama atskira duomenų lentelė.

Jeigu vartotojo atsakymas į klausimą (*Ar laukeliuose bus saugomos pasikartojančios reikšmės?*) neigiamas, vartotojui siūloma rinktis vieną iš kelių numatytų duomenų tipų. Vartotojui pasirinkus *Data* arba *Laikas* duomenų tipą siūloma galimybė šiame įrašė išsaugoti *įrašo kūrimo metu esančią datą arba laiką*. Vartotojui pasirinkus variantą kai saugoma *įrašo metu esanti data ar laikas* dialoge daugiau klausimų nepateikiama ir galima įtraukti elementą į šabloną. Jeigu pasirinktas kitoks duomenų tipas dialogo lange pateikiamas klausimas *ar galima įvesti vienodas reikšmes* ir *ar galima kuriant įrašą neįvesti reikšmės*. Tuo atveju kai vartotojas nusprendė, kad *galima neįvesti reikšmės* ir prieš tai nurodė, kad *galima įvesti vienodas reikšmes*, jam siūloma įvesti reikšmę, kuria bus užpildomi laukeliai lentelėje, jeigu vartotojas nenurodys reikšmės.

Kad vartotojas kurdamas šablono elementus nesuklystų ir neįvestų tokiu pačiu pavadinimu kelių elementų tikrinamas elemento pavadinimo unikalumas konkrečiame šablone. Skirtinguose šablonuose gali būti tokie patys elementų pavadinimai.

Bendra šablono kūrimo metu vartotojo atliekamų veiksmų schema pateikiama 2. pav.



2 pav. Bendra vartotojų veiksmų schema kuriant šabloną

### Duomenų lentelių pirminiai ir išoriniai raktai

Kiekvienoje lentelėje, kuriamoje pagal duomenų pildymo šabloną, automatiškai sukuriama laukelis *ID* ir pirminis raktas *ID*. Vartotojui kuriant šablonus ir nurodant, kad reikšmės bus imamos iš sąrašo, lentelėje sukuriamas išorinis raktas susietas ryšiu vienas prie daug su sąrašo informaciją saugančios lentelės pirminiu raktu.

### Šablono elementų šalinimas

Vartotojui inicijavus elemento šalinimo veiksmą tikrinama ar elementas nėra naudojamas kituose šablonuose ir ar nėra įvestų duomenų pagal elementą. Jeigu elementas naudojamas kitame šablone perspėjama, kad elemento pašalinti negalima, kol jis bus naudojamas kituose šablonuose, t.y. šalinant elementus taikomas režimas *ON DELETE RESTRICT*. Jeigu pageidaujamas pašalinti šablono elementas nenaudojamas kituose šablonuose, tačiau į duomenų lentelę yra pagal jį suvestų duomenų, vartotojui pranešama, kad ištrinus elementą bus prarasti ir duomenys. Vartotojui sutikus pašalinti elementą duomenų bazėje vykdoma komanda `ALTER TABLE `lentele` DROP `stulpelis``. Jeigu šalinamas elementas yra naudojantis duomenis iš kitos duomenų lentelės tuo pačiu pašalinamas ir išorinis raktas.

## **Šablono šalinimas**

Vartotojui inicijavus šablono šalinimo veiksmą tikrinama ar šablono elementai nėra naudojami kituose šablonuose ir ar nėra įvestų duomenų pagal šabloną. Jeigu šablono elementai naudojami kitame šablone, perspėjama, kad šablono pašalinti negalima, kol jo elementai naudojami kituose šablonuose, t.y. šalinant šablonus taikomas režimas *ON DELETE RESTRICT*. Jeigu pageidaujamas pašalinti šablono elementai nenaudojami kituose šablonuose, tačiau į duomenų lentelę yra pagal jį suvestų duomenų, vartotojui pranešama, kad ištrynus šabloną bus prarasti ir duomenys. Vartotojui sutikus pašalinti šabloną duomenų bazėje vykdoma komanda *DROP TABLE `lentele`*. Jeigu šalinamas šablonas naudojo elementus iš kitų šablonų tai tuo pačiu pašalinamas ir išorinis raktas.

## **Šablono elementų redagavimas**

Šablono elementui leidžiama pakeisti sekančius parametrus:

- Pavadinimą
- naudoti kūrimo metu esančią datą arba laiką
- ar galima įvesti vienodas reikšmes
- ar galima kuriant įrašą neįvesti reikšmės
- įrašo reikšmę, kuria bus užpildomi laukeliai lentelėje

Duomenų tipų bei įvedimo būdo keisti negalima.

## **Suvestų pagal šablonus duomenų redagavimas**

Redagavimo formos yra dviejų tipų:

- redaguojamas iš sąrašo pasirinktas įrašas
- redaguojami vienu metu visi įrašai tinklėlyje (*angl. grid*)

Jeigu redaguojamas įrašas yra naudojamas pagal kitus šablonus suvestuose duomenyse, tai prie įrašo rodomas simbolis (tarkime raudona žvaigždutė *\**) reiškiantis, kad atnaujinta informacija pasikeis visuose įrašuose kuriuose panaudota įvesta reikšmė, t.y. naudojamas išorinio rakto režimas *ON UPDATE NO ACTION*.

## **Suvestų pagal šablonus duomenų šalinimas**

Jeigu pageidaujamas šalinti įrašas yra naudojamas pagal kitus šablonus suvestuose duomenyse, tai šio įrašo pašalinti neleis, t.y. naudojas rakto režimas *ON DELETE RESTRICT*.

## **Virtualių lentelių – vaizdų (*angl. view*) naudojimas**

Norint supaprastinti kelių lentelių naudojimą vienu metu patogiu naudoti virtualias lenteles – vaizdus. Tuo atveju kai kuriamas šablono elementas yra naudojantis informaciją iš kito šablono sistema sukuria redaguojamo šablono lentelei vaizdą. Vaizdas atnaujinamas kiekvieną kartą, kai pašalinamas arba pridedamas šablono elementas. Vaizdai naudojami patogesniai duomenų pateikimui vartotojui. Iškvietus informacinės sistemos formą duomenų



peržiūrai, sistema nustato iš kur reikia imti duomenis – duomenų lentelės ar vaizdo. Šis parametras saugomas duomenų lentelėje *MDB\_SABLONAI*. Vaizdo stulpelių pavadinimai tokie pat kaip ir duomenų lentelės, kuriai yra sukurtas vaizdas.

Duomenų lentelių pavyzdžiai pateikiami lentelėse Nr. 3, Nr. 4 ir Nr. 5

**Lentelė Nr. 3 Duomenų lentelės „MDB\_LENTELE\_2“ pavyzdys**

ID	Laukelis_2	Laukelis_3	Laukelis_4	Laukelis_5
1	Raudonas	Žymeklis	1	2010-03-02
2	Geltonas	Rašiklis	2	2010-03-02
3	Žalias	Pieštukas	4	2010-03-02

**Lentelė Nr. 4 Duomenų lentelės „MDB\_LENTELE\_1“ pavyzdys**

Id	Laukelis_1
1	Inžinierius
2	Meistras
3	Sekretorius
4	Direktorius

**Lentelė Nr. 5 Virtualios duomenų lentelės „MDB\_VAIZDAS\_2“ pavyzdys**

ID	Laukelis_2	Laukelis_3	Laukelis_4	Laukelis_5
1	Raudonas	Žymeklis	Inžinierius	2010-03-02
2	Geltonas	Rašiklis	Meistras	2010-03-02
3	Žalias	Pieštukas	Direktorius	2010-03-02

Lentelių pavadinimų sudarymas:

Lentelės pavadinimas *MDB\_LENTELE\_2*. Indeksas „2“ yra įrašo „Darbuotojai“, esančio duomenų lentelėje *MDB\_SABLONAI*, identifikatorius.

Stulpelio pavadinimas Laukelis\_3. Indeksas „3“ yra įrašo „Pavardė“, esančio duomenų lentelėje *MDB\_SABLONU\_ELEMENTAI*, identifikatorius.

Virtualios duomenų lentelės *MDB\_VAIZDAS\_2* struktūra atitinka duomenų lentelės *MDB\_LENTELE\_2* lentelės struktūrą, tik jos laukeliai yra užpildyti pagal sekantį algoritmą:

Į vaizdo sudėtį įeina visi lentelės, kuriai konstruojamas vaizdas, nesudėtiniai laukeliai, t.y. tokie kurių reikšmės įvedamos rankiniu būdu, po jų prijungiami laukeliai naudojami iš kitų lentelių. Kadangi vartotojas gali pažymėti kelias reikšmes laukelio pasirenkamo iš sąrašo atvaizdavimui, tai reikšmei sudaryti naudojama *SQL* funkcija *CONCAT(laukelis\_x, ' ', laukelis\_y, ...)*. Pasirenkamos iš sąrašo reikšmės sudedamosios dalys saugomos duomenų lentelėje *MDB\_ELEMENTU\_SUDETIS*. Lentelės apjungiamos papildant lentelės, kuriai

kuriamas vaizdas reikšmės, t.y. naudojamas *MDB\_LENTELE\_2 LEFT JOIN MDB\_LENTELE\_1*. Reikšmės jungiamos naudojant sudėtinio stulpelio reikšmes lyginant su prijungiamos lentelės stulpelio ID reikšmėmis.

Virtualiai duomenų lentelei sukurti skirtos SQL užklauskos pavyzdys

```
CREATE VIEW MDB_VAIZDAS_2
AS
SELECT
    c0.id, c0.laukelis_2, c0.laukelis_3, c0.laukelis_5,
    CONCAT (c1.laukelis_1) AS laukelis_4
FROM
    MDB_LENTELE_2 LEFT JOIN MDB_LENTELE_1
ON
    c0.laukelis_4 = c1.id
```

## **4. PROJEKTAS**

### **4.1 Tikslai ir uždaviniai**

Tikslas – suprojektuoti organizacijos informacinę sistemą, skirtą kaupti ir apdoroti informaciją pagal darbuotojų sukurtus duomenų įvedimo bei koregavimo šablonus.

Uždaviniai:

- apibrėžti sistemos aktorius;
- apibrėžti veiklos sudėtį bei išskirti panaudojimo atvejus;
- sudaryti sistemos koncepcinį modelį;
- pateikti sistemos statinį, dinaminį, išdėstymo vaizdą bei loginę duomenų struktūrą;
- sumodeliuoti informacinės sistemos architektūrą.

### **4.2 Projekto kokybės kriterijų apibrėžimas**

Kadangi išnagrinėtos sistemos apimties bei funkciniu požiūriu pripažintos netinkamomis, nuspręsta kurti naują sistemą, kuo tiksliau tenkinančią numatytus reikalavimus.

Kokybės kriterijai, leisiantys nustatyti kuriamos sistemos tinkamumą:

- Sistemoje galima kurti duomenų įvedimo šablonus;
- Sistemoje galima redaguoti esamus duomenų įvedimo šablonus;
- Pašalinti šablono elemento negalima kol jis naudojamas kitame šablone duomenų įvedimui;
- Pašalinti šablono negalima kol jo elementai naudojami kituose šablonuose duomenų įvedimui;
- Sistemoje galima suvesti pageidaujamus duomenis pagal sukurtus šablonus;
- Sistemoje galima redaguoti suvestus duomenis;
- Įrašo negalima pašalinti kol jis naudojamas kitų duomenų lentelių įrašuose;
- Duomenis galima redaguoti tinklelio formoje (angl. grid);
- Įrašo, kuris naudojamas kitų duomenų lentelių įrašuose, redagavimo metu turi būti perspėjimas apie duomenų pasikeitimą visuose susijusiuose įrašuose.

Detalesni kriterijai aprašyti skyriuje „Reikalavimų specifikavimas“.

### **4.3 Sistemos kūrimo priemonių parinkimas**

Kuriamą sistemą nutarta projektuoti kaip internetinę sistemą, kurią sudaro klientas (interneto naršyklė) ir serveris (HTTP serveris su PHP palaikymu bei duomenų bazių serveris).

Internetinės svetainės gali būti statinės arba dinaminės. Statinės svetainės, tai paprastų tarpusavyje susietų HTML failų rinkinys. Svetainę nesudėtinga sukurti, užima mažai vietos, joje yra nedaug informacijos ir ta informacija atnaujinama gana retai.

Dinaminės svetainės gali atnaujinti informaciją pagal duomenis imamus iš duomenų bazės, t.y. tokios svetainėse informacija nėra įrašyta HTML failuose, o dinamiškai keičiama puslapių kūrimo scenarijų pagalba.

Kadangi kuriama sistema bus skirta darbui su duomenų baze, statinė interneto svetainė šiai sistemai netinka. Kuriama sistema bus dinaminė interneto svetainė.

Dinaminių svetainių kūrimui bei veikimui yra naudojamos serverio ir kliento technologijos. Prie serverių priskiriamos specializuotos programos, kurias valdo HTTP serveriai.

Veiksniai nulemiantys programavimo priemonių pasirinkimą:

- Nemokama priemonė;
- Nepriklauso nuo serverio sisteminės įrangos;
- Konfigūravimo paprastumas.

Populiariausios HTTP serverių programavimo kalbos: Perl, PHP, Python [2]

Populiariausi HTTP serveriai: Apache HTTP Server, Microsoft IIS, Zeus (tolesnis programos vystymas sustabdytas) [5].

**Lentelė Nr. 6 Kūrimo priemonių pasirinkimo kriterijai**

Kriterijus	Nemokama priemonė	Nepriklauso nuo serverio sisteminės įrangos	Nesudėtingas konfigūravimas
Programinė priemonė			
Programavimo kalba Perl	Taip	Taip	Taip
Programavimo kalba PHP	Taip	Taip	Taip
Programavimo kalba Python	Taip	Taip	Ne
HTTP serveris Apache	Taip	Taip	Taip
HTTP serveris Microsoft IIS	Ne	Ne	Taip
HTTP serveris Zeus	Ne	Ne	Taip

HTTP serveriai Microsoft IIS ir Zeus atkrenta dėl kainos ir priklausomumo nuo platformos. Programavimo kalba Python atkrenta dėl sudėtingos jos konfigūravimo. Iš likusių programavimo kalbų pasirinkta PHP programavimo kalba.

Šiame projekte naudojamos programos – Apache 2.2.3 (Win32) Server, PHP 5.1.6, MySQL 5.0.24a, phpMyAdmin 2.9.0.1. Apache 2.2.3 (Win32) Server, PHP 5.1.6 ir MySQL 5.0.24a naudojami programinės įrangos kūrimui bei vykdymui, o phpMyAdmin 2.9.0.1 naudojamas sukurti pradines lenteles bei sukurtos programinės įrangos generuojamų lentelių bei duomenų patikrinimui, t.y. peržiūrėjimui.

Šios programos yra nemokamos jei jos nenaudojamos pelno siekimui. Kadangi projektas vykdomas moksliniais tikslais, projekto vykdytojas nesiekia gauti jokio pelno, nebus pažeistos šių programų naudojimo licencijos.

#### **4.4 Reikalavimų specifikavimas**

Šio skyriaus tikslas – nustatyti ir pateikti reikalavimus kuriamai sistemai.

##### **4.4.1 Bendrieji reikalavimai**

**Produkto apibrėžimas:** informacinė sistema turi saugoti informaciją apie vartotojų sukurtus duomenų įvedimo formų šablonus bei duomenis suvestus pagal sukurtus šablonus.

**Vartotojų apibūdinimas:** Vartotojai moka naudotis kompiuteriu bei interneto naršyklėmis.

**Vartotojų tikslai ir poreikiai:** vartotojų tikslas yra operatyvumas ir visapusiškas informacijos apdorojimas, laiko sąnaudų mažinimas informacijos apdorojimui bei lengvai suprantamas informacinės sistemos naudojimas.

##### **4.4.2 Funkciniai reikalavimai**

Informacinė sistema turi vykdyti organizacijoje pageidaujamų kaupiamų duomenų saugojimą bei kitokį apdorojimą.

Sistema turi leisti:

- Sukurti duomenų įvedimo šablonus.
- Redaguoti duomenų įvedimo šablonus.
  - o Šablonų redagavimo metu duomenys nenaikinami jeigu jie naudojami kito šablono pildymo metu.
  - o Šablonų elementus, nenaudojamus kitų šablonų pildymo metu, galima naikinti: vartotojas perspėjamas apie kartu panaikinamus suvestus duomenis.
- Suvesti duomenis pagal sistemoje sukurtus duomenų įvedimo šablonus.
- Redaguoti duomenis suvestus pagal duomenų įvedimo šablonus.

- Patogiai ir greitai vykdyti duomenų paiešką pagal vartotojo nurodytus paieškos kriterijus.

Sistema turi neleisti:

- Duomenų šalinimo jeigu tie duomenys naudojami kituose šablonuose.

#### **4.4.3 Nefunkciniai reikalavimai**

##### **Reikalavimai panaudojamumui**

Informatyvi, lengvai skaitoma, neperkrauta vartotojo sąsaja. Spalvos turi būti suderintos ir neįkyrios, tinkamai parinktas šrifto dydis. Išmokti naudotis sistema turi būti nesudėtinga. Vartotojams nereikėtų mokymo kursų darbui su sistema.

Nesudėtingas iki dviejų lygiu meniu, kad vartotojui būtų paprasta orientuotis sistemoje.

Sistema turi padėti vartotojui išvengti galimų klaidų įvedant ar koreguojant duomenis. Sistema turi neleisti išsaugoti įrašų neteisingai užpildžius įvedimo laukelius ir perspėti vartotoją apie neteisingai suvestas reikšmes.

Sistema turi būti draugiška vartotojui – įvykus klaidai, turi būti pateikiamas aiškus ir konkretus pranešimas.

##### **Reikalavimai vykdymo charakteristikoms**

Efektyvus resursų panaudojimas – sistema turi efektyviai naudoti resursus ir netrikdyti kitų sistemų darbo.

Efektyvus užduočių vykdymas – sistema turi atlikti užduotis kaip galima per trumpesnę laiką.

Sistema, turi būti suprojektuota taip, kad būtų lengva išplėsti jos galimybes.

Sistemos veikimas turi būti stabilus.

##### **Kiti nefunkciniai reikalavimai**

Sistemoje turi būti naudojama taisyklinga lietuvių kalba.

#### **4.4.4 Projekto apribojimai**

Sistema turi būti pasiekama per interneto naršyklę.

Sistema turi būti realizuota modeliu klientas – serveris.

Panaudoti esamus resursus (kompiuterius, programinę įrangą).

### **4.5 Vartotojai**

Sukurtos sistemos vartotojai gali būti moksleiviai ar studentai tiriantys panašias temas.

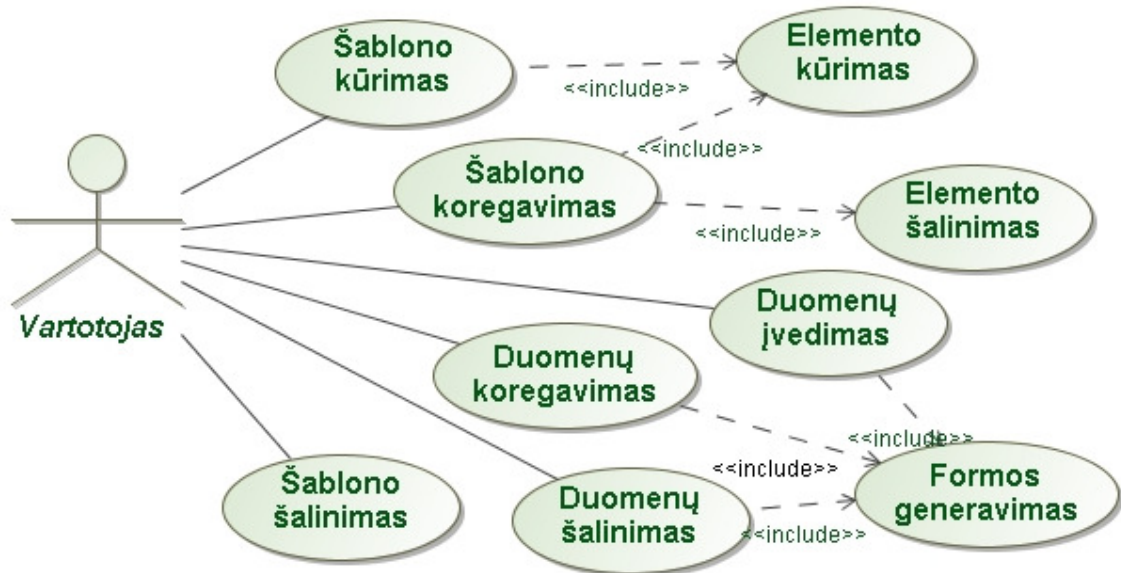
Vartotojo sprendžiami uždaviniai:

- Šablonų kūrimas, koregavimas, šalinimas;
- Dalinis įvedimo formos projektavimas – duomenų įvedimo elementai keičiami vietomis (galima perkelti žemyn, aukštyn, į kairę ar į dešinę)

- Duomenų vedimas, peržiūra, koregavimas.

#### 4.6 Panaudojimo atvejai

Atsižvelgiant į tai, kokie veiksmai turi būti galimi kuriamoje informacinėje sistemoje, sudaroma sistemos panaudojimo atvejų diagrama, kuri pateikta 3 paveikslėlyje.



3 pav. Panaudojimo atvejai

1. Panaudojimo atvejis: šablono kūrimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu sistemoje registruojamas naujas šablonas.

**Prieš sąlyga:** veikianti sistema

**Sužadinimo sąlyga:** vartotojas nori sukurti naują šabloną ir pasirinko meniu punktą „Sukurti naują šabloną“

**Po sąlyga:** sukurtas naujas šablonas, duomenų bazėje sukurta nauja lentelė

2. Panaudojimo atvejis: šablono koregavimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu redaguojamas sistemoje esantis šablonas

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas

**Sužadinimo sąlyga:** vartotojas nori redaguoti šabloną ir pasirinko meniu punktą „Redaguoti šabloną“

**Po sąlyga:** pakoreguotas šablonas, atitinkamai modifikuota duomenų lentelė

3. Panaudojimo atvejis: šablono šalinimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu šalinamas sistemoje esantis šablonas

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas. Šablono elementai neturi būti naudojami kito šablono pildymui

**Sužadinimo sąlyga:** vartotojas nori pašalinti šabloną ir pasirenko meniu punktą „Pašalinti šabloną“

**Po sąlyga:** pašalintas šablonas, panaikinta duomenų lentelė

4. Panaudojimo atvejis: elemento kūrimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu kuriamas naujas šablono elementas

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas

**Sužadinimo sąlyga:** vartotojas nori pridėti elementą ir pasirenko meniu punktą „Pridėti elementą“

**Po sąlyga:** šablone sukurtas naujas elementas, atitinkamai modifikuota duomenų lentelė

5. Panaudojimo atvejis: elemento šalinimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu šalinamas šablono elementas

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas ir nors vienas elementas. Elementas neturi būti naudojamas kito šablono pildymui

**Sužadinimo sąlyga:** vartotojas nori pašalinti elementą ir pasirenko meniu punktą „Pašalinti elementą“

**Po sąlyga:** šablono elementas pašalintas, atitinkamai modifikuota duomenų lentelė

6. Panaudojimo atvejis: duomenų vedimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu vedami nauji duomenys

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas ir nors vienas elementas

**Sužadinimo sąlyga:** vartotojas nori įvesti naują įrašą ir pasirenko meniu punktą „Naujas įrašas“

**Po sąlyga:** sukurtas įrašas duomenų lentelėje

7. Panaudojimo atvejis: duomenų koregavimas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu koreguojami esantys duomenys

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas, nors vienas elementas ir įvestas



nors vienas įrašas

**Sužadavimo sąlyga:** vartotojas nori koreguoti įrašą ir pasirinko meniu punktą „Redaguoti“

**Po sąlyga:** pakoreguotas įrašas duomenų lentelėje

8. Panaudojimo atvejis: duomenų šalinamas

**Vartotojas/Aktorius:** vartotojas, sistema

**Aprašas:** apima procesą, kurio metu šalinami esantys duomenys

**Prieš sąlyga:** turi būti sukurtas nors vienas šablonas, nors vienas elementas ir įvestas nors vienas įrašas

**Sužadavimo sąlyga:** vartotojas nori pašalinti įrašą ir pasirinko meniu punktą „Pašalinti įrašą“

**Po sąlyga:** iš duomenų lentelės pašalintas įrašas

9. Panaudojimo atvejis: formos generavimas

**Vartotojas/Aktorius:** sistema

**Aprašas:** apima procesą, kurio metu pagal sukurtą šabloną generuojama forma ir pateikiama vartotojui

**Prieš sąlyga:** turi būti sukurtas bent vienas šablonas ir bent vienas elementas

**Sužadavimo sąlyga:** vartotojas pasirinko vieną iš meniu punktų: „Duomenų įvedimas“, „Duomenų koregavimas“, „Duomenų šalinimas“

**Po sąlyga:** sugeneruota ir vartotojui pateikta duomenų manipuliavimo forma

## 4.7 Projekto išeiga

### 4.7.1 Atviri klausimai

Sukurti paprastą tinklalapį nesunku. Daug sunkiau pasiekti gero svetainės apipavidalinimo. Tam gali padėti pakopiniai stilių šablonai (CSS), tinklapių išvaizdos pagrindas [4]. Kaskadinių stilių pagalba galima daug tiksliau ir paprasčiau išdėstyti svetainės turinį. Naudojant šiuos šablonus kyla grėsmė, kad pasitaikys vartotojo sąsajos iškreipimų, nes ne visos naršyklės (net ir tos pačios naršyklės skirtingos versijos) nevienodai interpretuoja tą patį CSS stilių.

### 4.7.2 Pagaminti komponentai kurie gali būti panaudoti

Filter – JavaScript scenarijus, skirtas į tekstinius laukelius įvedamos informacijos tikrinimui. Galima nurodyti, kad vartotojui būtų leidžiama įvesti vien skaičius, skaičius su dešimtainiu kableliu ir t.t. Modulio autorius Joyce Babu.

### **4.7.3 Įtaka jau įdiegtoms sistemoms**

Diegiamos sistemos sėkmingam darbui reikalingas HTTP serveris su PHP palaikymu bei MySQL duomenų bazių serveris. HTTP serveryje vienu metu gali veikti kelios panašaus pobūdžio sistemos. Diegiama sistema gali naudoti jau esančią duomenų bazę MySQL duomenų bazių serveryje arba sukurti naują. Kiekviena diegiamos sistemos lentelė turi savo priešdėlį, pagal kurį galima atskirti, kad lentelės priklauso naujai įdiegtai sistemai tuo atveju jei bus pasirinktas diegimas panaudojant jau esančią duomenų bazę.

### **4.7.4 Reikalavimai esamų duomenų perkėlimui**

Į sukurta informacinę sistemą neplanuojama kelti jokių duomenų iš esančių duomenų bazių ar kitų informacijos saugojimo priemonių.

## **4.8 Sistemos projektas**

Sistemos projektavimo metu kuriamas sistemos statinis ir dinaminis vaizdai. Jie pateikiami UML diagramų pavidalu:

- sistemos statinis vaizdas atvaizduojamas sistemos klasių diagrama;
- sistemos dinaminis vaizdas – būsenų ir veiklos diagramomis.

Šio skyriaus tikslas – surinkti ir pateikti svarbius architektūrinius sprendimus, kuriuos galima atlikti kuriamoje informacinėje sistemoje.

Komponentai yra klasifikuojami. Aprašoma jų struktūra, resursai, sąveika, tikslai, apribojimai ir sąsajos. Realizuojant sistemą tai padės geriau suprasti architektūrinius sprendimus.

Sistemos architektūros specifikacija pateikiama:

- Statinio sistemos vaizdo diagrama (suskaidymas į paketus, paketų klasių diagramos);
- Dinaminio sistemos vaizdo diagrama (veiklos diagramos, bendradarbiavimo diagramos, sekų diagramos);
- Išdėstymo diagrama;
- duomenų bazės modelio diagrama.

### **4.8.1 Projekto apribojimai**

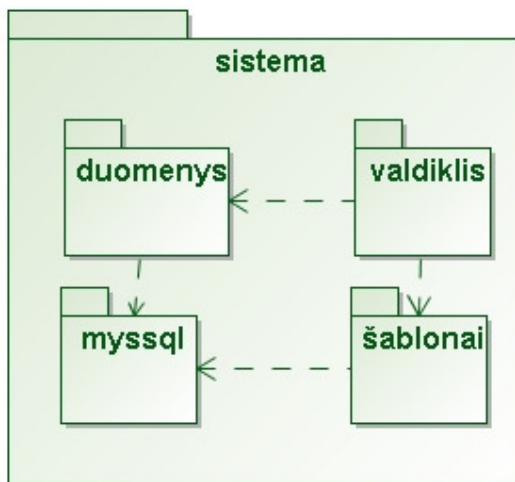
Architektūros sprendimus įtakojantys reikalavimai bei apribojimai:

- Sistema turi būti realizuota modeliu klientas – serveris.
- Duomenų bazė turi būti realizuota MySQL duomenų bazės pagrindu.
- Sistemos architektūra turi būti suprojektuota taip, kad būtų lengva išplėsti jos galimybes.
- Sistemoje gali dirbti tik autorizuoti vartotojai.

- Sudarant sistemos architektūrą, turi būti atsižvelgta į informacinės sistemos charakteristikas, kurios apibrėžtos reikalavimų specifikacijoje (skyrius „Reikalavimų specifikavimas“).

#### 4.8.2 Sistemos statinis vaizdas

Informacinė sistema kuriama kaip atskira sistema. Kuriamos sistemos realizacijos paketų struktūra turėtų atrodyti taip, kaip pateikta 4 paveikslėlyje.

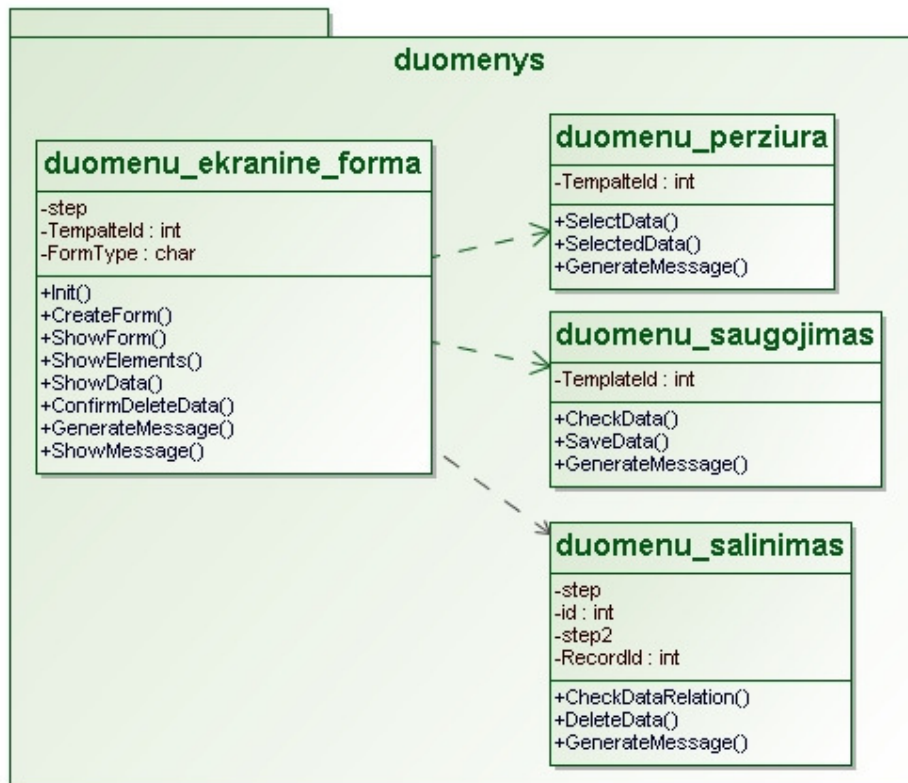


4 pav. Sistemos išskaidymas į paketus

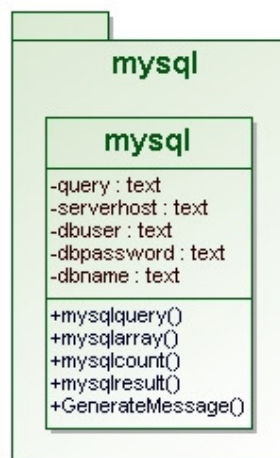
#### Paketų detalizavimas



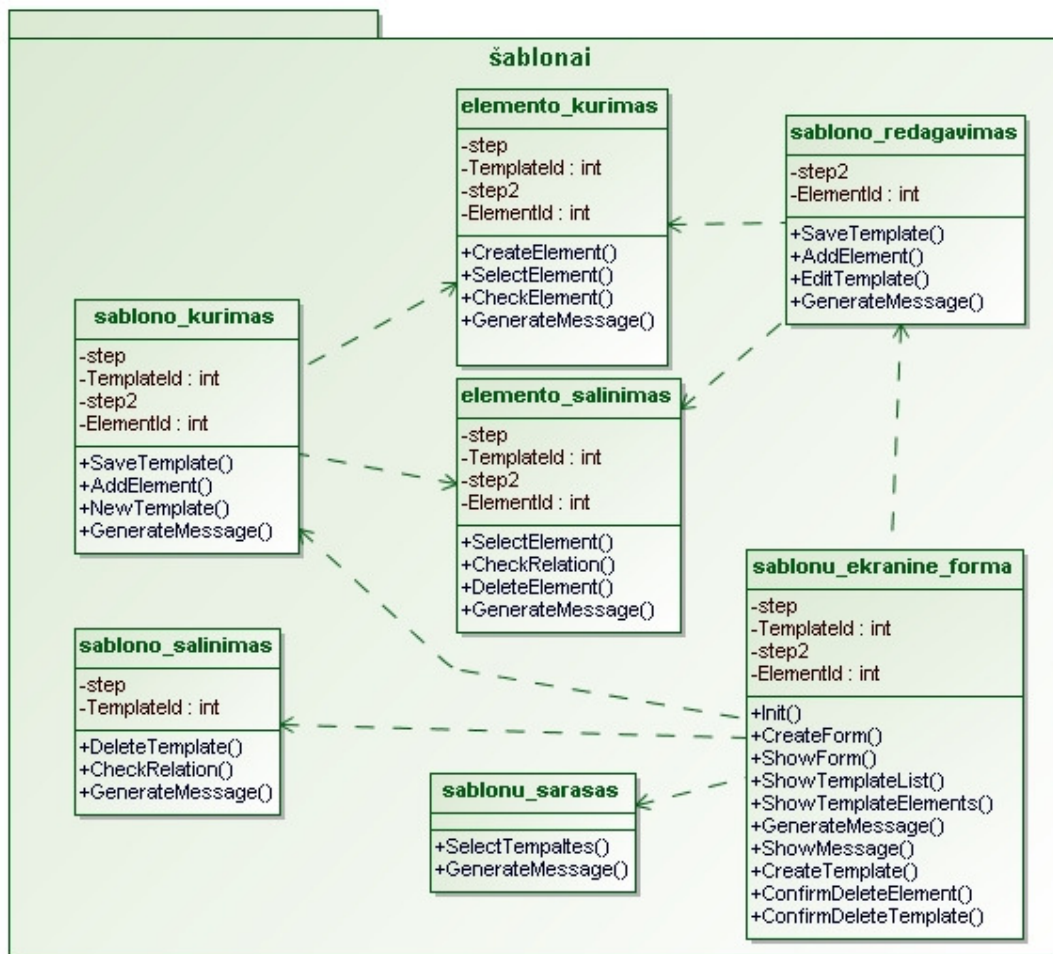
5 pav. Paketo „valdiklis“ struktūra



6 pav. Paketo „duomenys“ struktūra



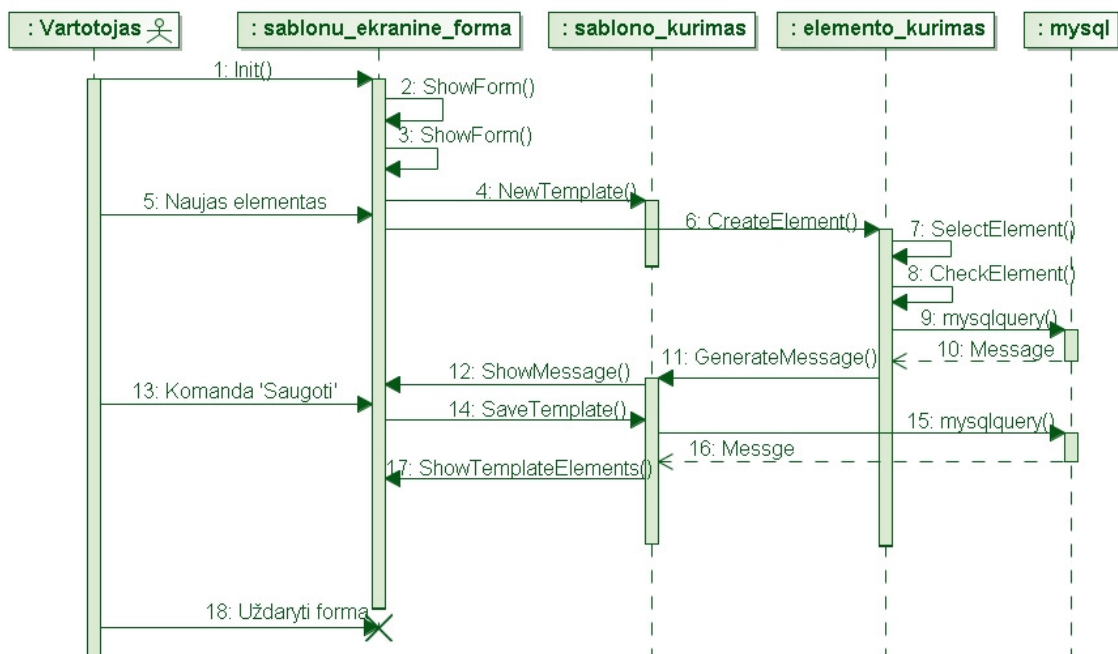
7 pav. Paketo „mysql“ struktūra



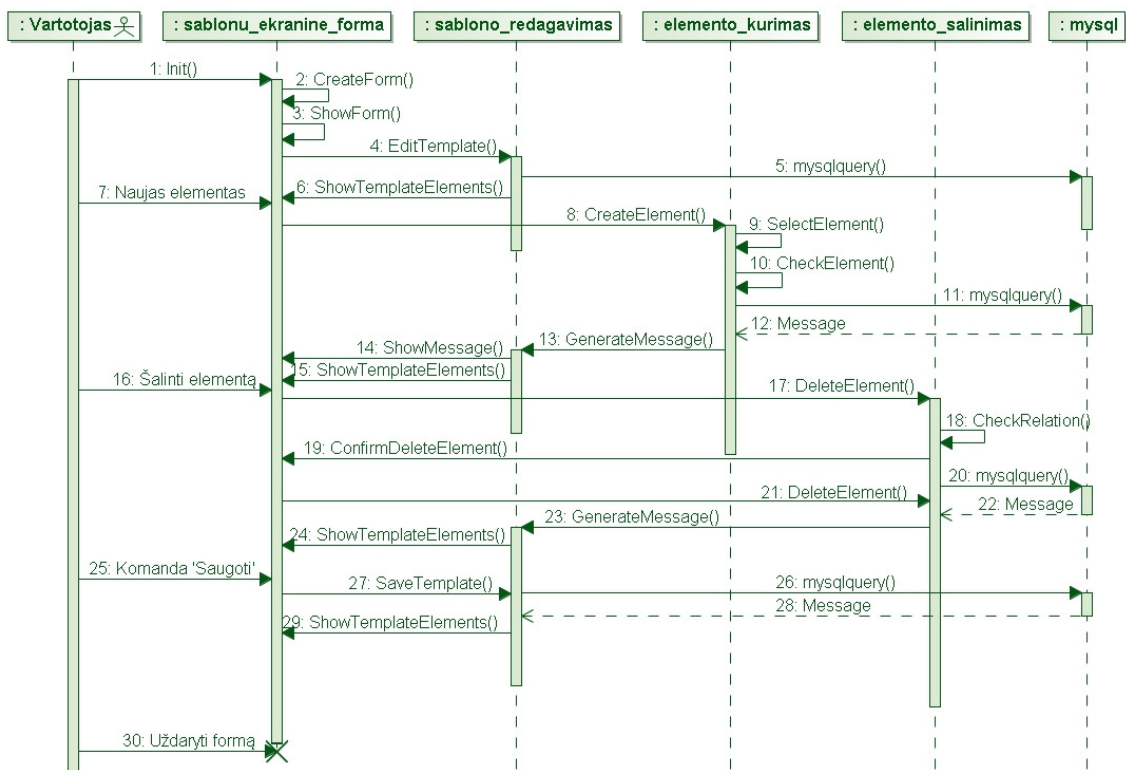
8 pav. Paketo „šablonai“ struktūra

### 4.8.3 Sistemos dinaminis vaizdas

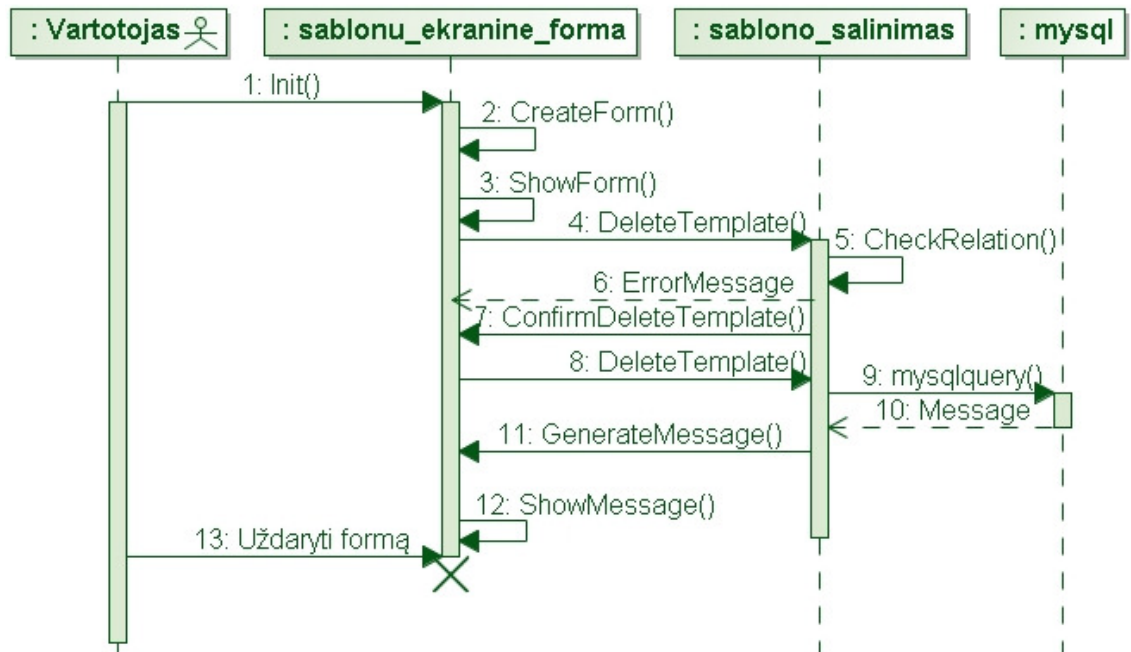
- Sekų diagramos



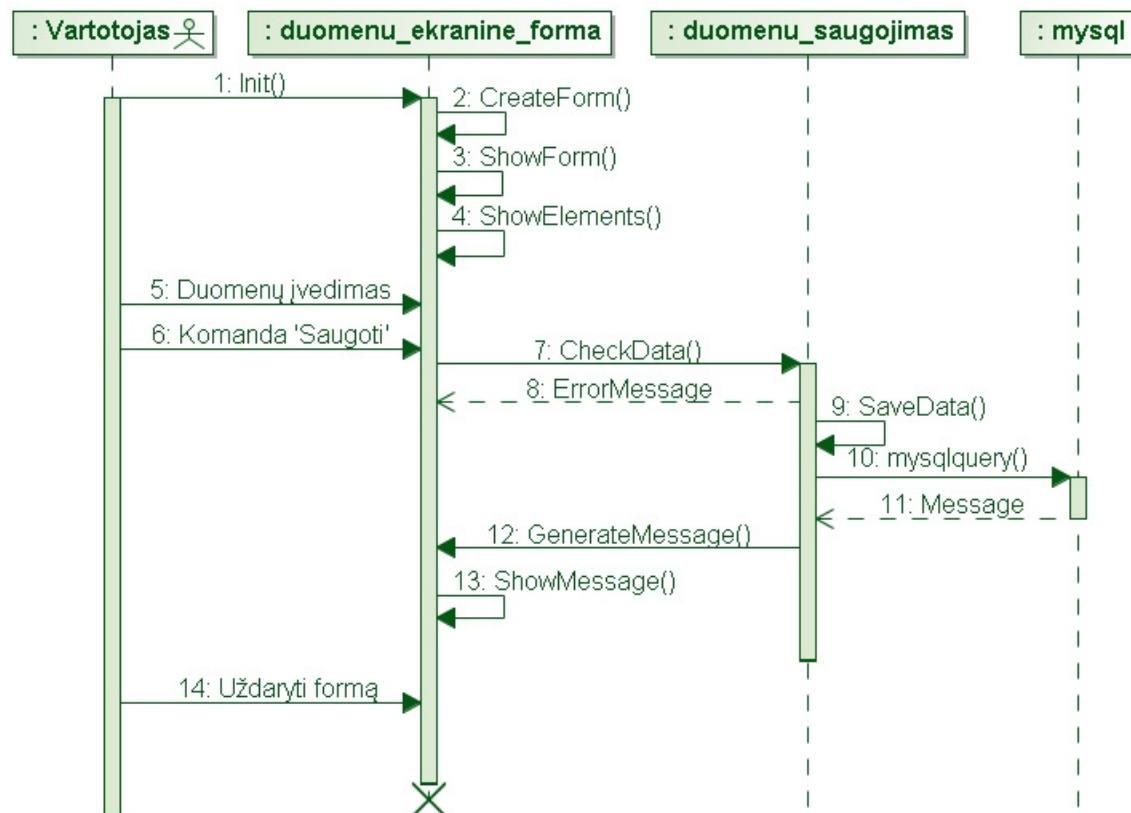
9 pav. Panaudojimo atvejo „Šablono kūrimas“ sekų diagrama



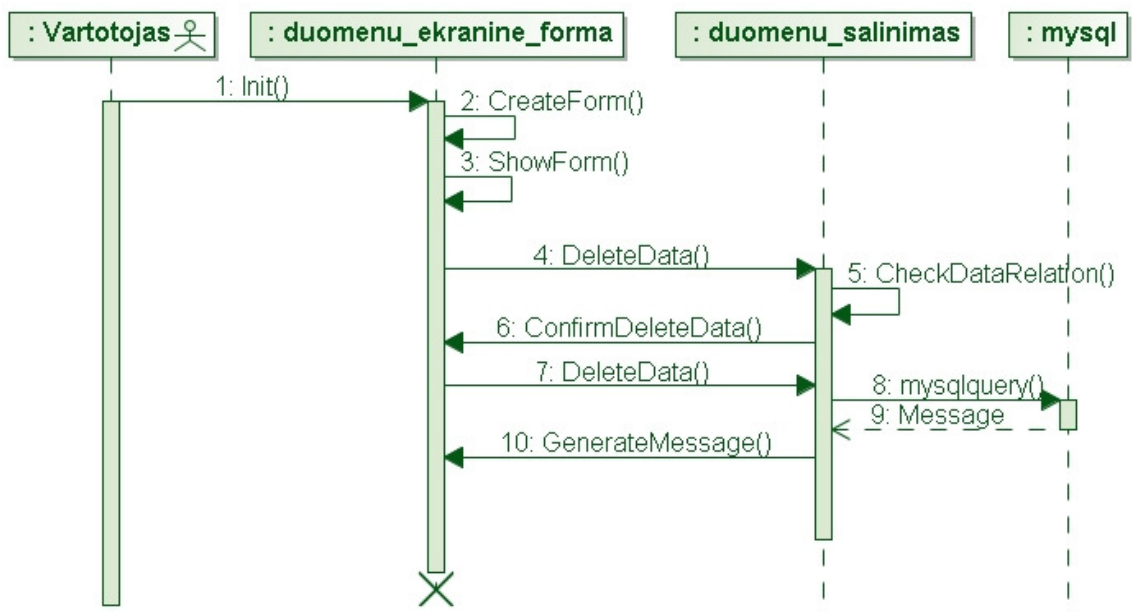
10 pav. Panaudojimo atvejo „Šablono koregavimas“ sekų diagrama



11 pav. Panaudojimo atvejo „Šablono šalinimas“ sekų diagrama



12 pav. Panaudojimo atvejo „Duomenų koregavimas“ sekų diagrama



13 pav. Panaudojimo atvejo „Duomenų šalinimas“ sekų diagrama

#### 4.8.4 Išdėstymo vaizdas

##### - Serveris

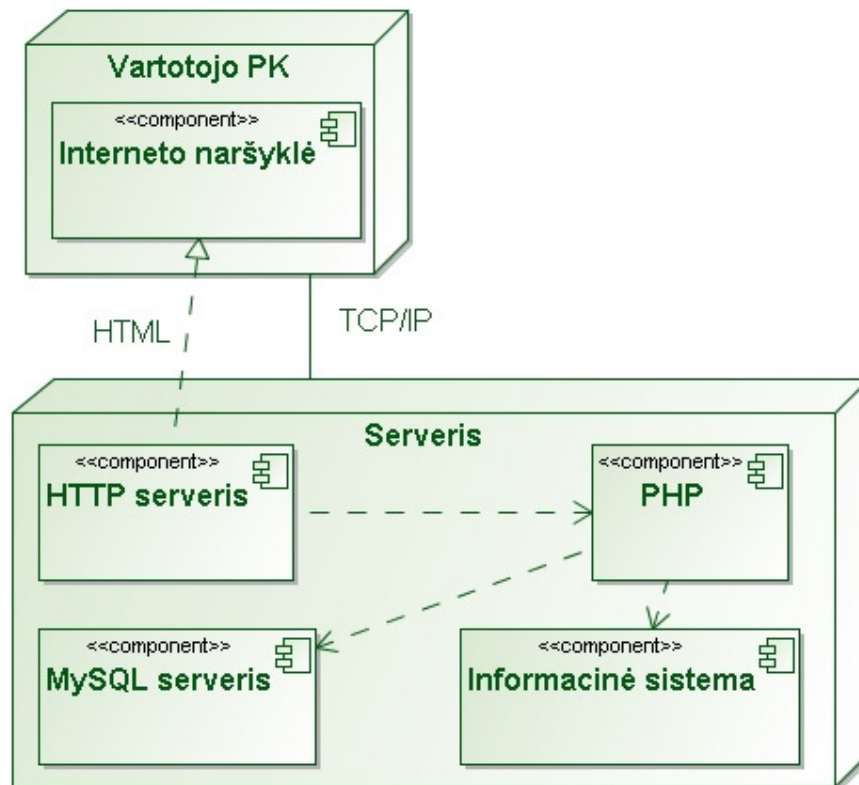
Kompiuteryje, kuriame yra įdiegta Microsoft XP operacinė sistema, bus įdiegtas HTTP serveris su PHP palaikymu, MySQL duomenų bazių serveris bei sukurta informacinė sistema. Serveris turi būti prijungtas prie organizacijos vietinio kompiuterių tinklo.

##### - Kliento PK

Kliento kompiuteryje veikia MS Windows XP operacinė sistema. Sistemai pasiekti naudojama interneto naršyklė palaikanti JavaScript. Kompiuteris turi būti prijungtas prie organizacijos vietinio kompiuterių tinklo.

Atsižvelgiant į aukščiau išdėstytą informaciją sudaromas sistemos komponentų išdėstymo fiziniame lygmenyje vaizdas.





14 pav. Išdėstymo vaizdas

#### 4.8.5 Duomenų vaizdas

Duomenų bazei saugoti yra pasirinktas MySQL duomenų bazių serveris.

Pradinėje duomenų bazėje esančių lentelių sąrašas pateikiamas lentelėje Nr. 4.

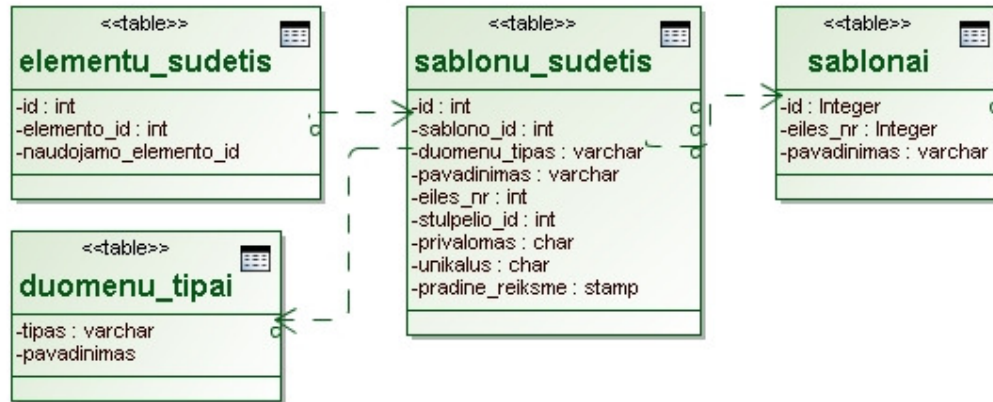
Pradinėje duomenų bazėje informacinės sistemos lentelių priešdėlis yra MDB.

Lentelė Nr. 7 Duomenų bazės lentelės

Eil. Nr.	Lentelė	Aprašymas
1.	MDB_SABLONAI	Lentelė skirta pagrindinės informacijos apie sistemoje sukurtus šablonus saugojimui.
2.	MDB_SABLONU_ELEMENTAI	Lentelė skirta informacijos apie sistemoje sukurtų šablonų sudėtį saugojimui.
3.	MDB_DUOMENU_TIPAI	Lentelė skirta saugoti informaciją apie duomenų tipus
4.	MDB_ELEMENTU_SUDETIS	Lentelė skirta saugoti informaciją apie sudėtinius šablonų elementus, kurie imami iš kitų lentelių

Detalesni lentelėje Nr. 3 pateiktų duomenų bazės lentelių aprašymai pateikiami 1 priede „Duomenų bazės lentelių aprašai“.

Duomenų bazės modelis pateikiamas 39 paveikslėlyje.



15 pav. Duomenų bazės modelis

#### 4.8.6 Kokybė

Aprašytos informacinės sistemos kūrimo strategija paremta komponentiniu kūrimu – informacinė sistema kuriama iš atskirų modulių. Tai leis informacinėje sistemoje esančius modulius lengvai patobulinti bei papildyti sistemą naujais moduliais.

Informacinės sistemos realizavimas klientas – serveris architektūra įgalina lengvai atlikti pakeitimus sistemoje. Taip išvengiama klientų programinės įrangos modifikavimo.

Sistemos pasiekiamumas panaudojant internetinę naršyklę daro patogesnę kliento dalies diegimą bei leidžia lengvai atlikti informacinės sistemos programinės dalies pakeitimus.

Jei sistemą reikėtų perkelti į kitą, ne MS Windows, operacinę sistemą tai jokios įtakos informacinei sistemai neturės.

## **5. EKSPERIMENTINIS SISTEMOS TYRIMAS BEI KOKYBĖS ĮVERTINIMAS**

### **5.1 Eksperimentinio sistemos tyrimo tikslai**

Kuriant eksperimentinę programinę įrangą siekiama, kad produktas turėtų kiek įmanoma mažesnę klaidų bei defektų kiekį. Pristatomo tyrimo tikslas — atskleisti ir ištaisyti kaip galima daugiau sukurtos informacinės sistemos klaidų. Taip pat siekiama patikrinti, ar sukurta programinė įranga atitinka vartotojų reikalavimus bei specifikaciją.

### **5.2 Realizuotos informacinės sistemos funkcinis aprašas**

Skyriuje trumpai aprašoma realizuota Duomenų bazės kūrimo panaudojant duomenų pildymo formų šablonus informacinė sistema. Aprašomi vartotojo sąsajos elementai bei teikiami funkcionalumai.

#### **5.2.1 Sistemos paskirtis**

Informacinė sistema turi kaupti informaciją apie vartotojų sukurtus duomenų įvedimo formų šablonus bei duomenis suvestus pagal sukurtus šablonus.

#### **5.2.2 Sistemos atliekamos funkcijos**

##### **Šablonų valdymas**

Sistemos vartotojas kuria naujus arba koreguoja esamus duomenų pildymo šablonus. Sukurtame šablone gali pridėti naują arba pašalinti esamą elementą. Rikiuoti elementų pateikimo tvarką pildymo formoje. Gali pakeisti nustatymą rodyti ar nerodyti pagrindiniame meniu tiesioginę nuorodą į pildymo formą.

##### **Duomenų pagal sukurtus šablonus valdymas**

Sistemos vartotojas į konkretų šabloną gali vesti naujus arba redaguoti esamus duomenis. Sistema pagal sukurtą šabloną generuoja duomenų pildymo formą skirtą duomenų manipuliavimui.

#### **5.2.3 Bendri sistemos principai**

Darbas su realizuota informacine sistema vyksta interneto naršyklės pagalba. Sistemai pasiekti interneto naršyklės adreso laukelyje reikia įrašyti informacinės sistemos serverio adresą. Teisingai surinkus adresą atveriamas pagrindinis sistemos langas.

##### **Pagrindinis sistemos langas**

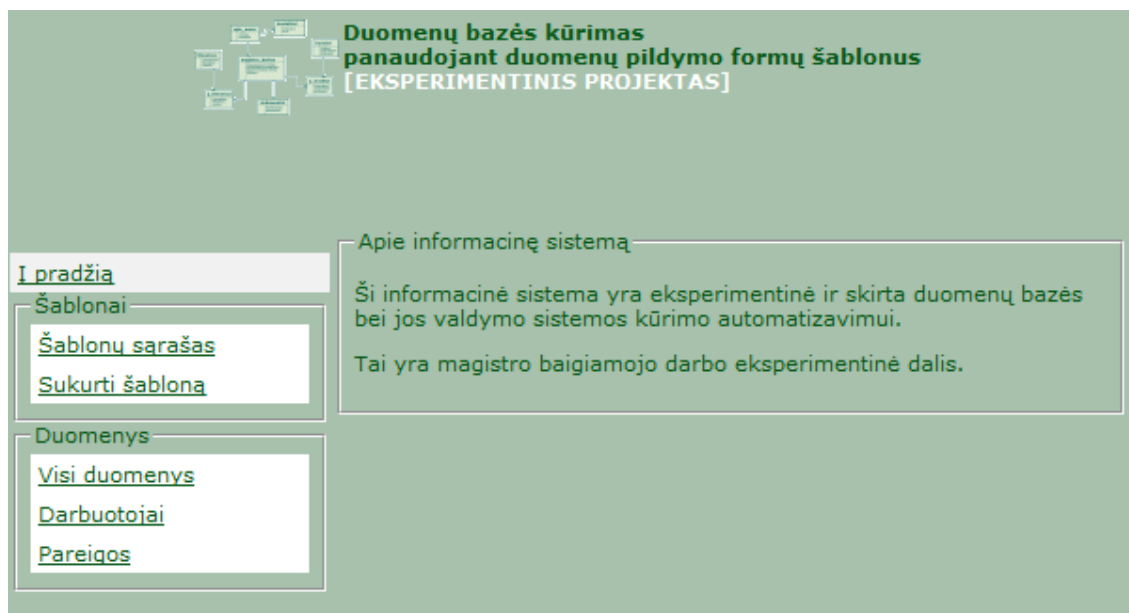
Pagrindiniai lango elementai:

- Informacinės sistemos pavadinimas bei logotipas
- Pagrindinis meniu.
- Centrinė lango dalis.

Pagrindinio informacinės sistemos lango centrinėje dalyje pateikiama trumpa informacija apie informacinę sistemą.

Informacinės sistemos pavadinimas bei logotipas ir pagrindinis meniu išlieka kiekviename informacinės sistemos lange. Centrinėje lango dalyje pateikiama forma pagal vartotojo pasirinktą meniu punktą.

Bendra lango schema ir funkcionalumas pateikiami 16 pav.



16 pav. Pagrindinis informacinės sistemos langas

### Pagrindinis meniu

Pagrindinis informacinės sistemos meniu yra kairėje pagrindinio langė pusėje. Sistema pagrindinį meniu generuoja automatiškai. Meniu punktai „Į pradžia“, „Šablonų sąrašas“, „Sukurti šabloną“ ir „Visi duomenys“ rodomi visada. Papildomai meniu dalyje „Duomenys“ rodomi punktai nukreipiantys tiesiai į duomenų pildymo formas, kurių šablonams vartotojas nurodė požymi „Rodyti pagrindiniame meniu“.

### Langas „Šablonų sąrašas“

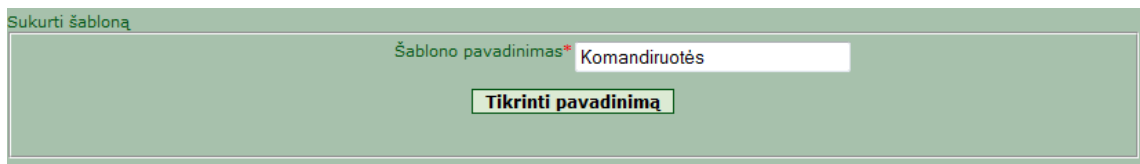
Į šį langą (žr. 17 pav.) galima patekti pasirinkus punktą „Šablonų sąrašas“. Jame vartotojui pateikiami informacinėje sistemoje sukurti duomenų pildymo šablonai. Vartotojas gali pasirinkti kuriuos šablonus pateikti pagrindiniame meniu, gali keisti jų pateikimo eiliškumą, koreguoti šabloną bei pašalinti. Šiame lange yra meniu punktas „Naujo šablono kūrimas“, kurį paspaudus išsikleidžia pradinį duomenų apie duomenų pildymo šabloną forma.



17 pav. Langas „Sukurti šablona“

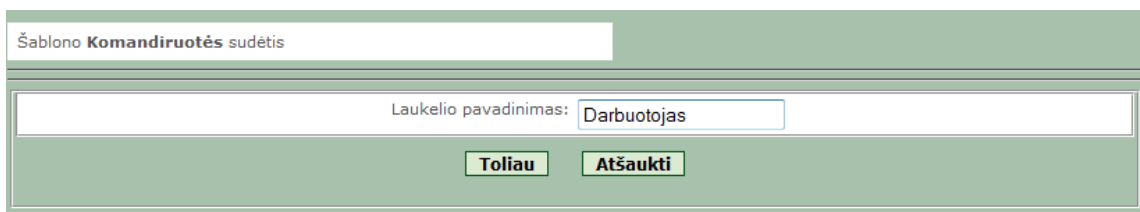
### Langas „Naujo šablono kūrimas“

Formoje (žr. 18 pav.) reikia nurodyti duomenų pildymo šablono pavadinimą „Tikrinti pavadinimą“. Paspaudus mygtuką „Tikrinti pavadinimą“ atliekamas pavadinimo tikrinimas. Jei šablono vardas unikalus pereinama į kitą formą.



18 pav. Langas „Naujo šablono kūrimas“

Atsidariusioje formoje (žr. 19 pav.) vartotojas privalo įvesti naujo šablono elemento pavadinimą. Paspaudus mygtuką „Toliau“ tikrinamas elemento pavadinimo unikalumas šablone.



19 pav. Langas „Naujo šablono kūrimas“

Toliau vartotojui pateikiamas klausimas „Ar laukeliuose bus saugomos pasikartojančios reikšmės?“. Jei vartotojas tiksliai žino, kad tos reikšmės bus pasikartojančios jam siūloma sukurti atskirą reikšmių sąrašą – žinyną arba pasirinkti iš jau esančių sistemoje. Pasirinkęs žinyną iš sąrašo, vartotojas turi nurodyti kuriuos laukelius sistema turi atvaizduoti vedant duomenis (20 pav.). Taip pat vartotojas turi nurodyti ar galima įvesti vienodas reikšmes bei ar galima palikti tuščią reikšmę. Suvedus duomenis ir paspaudus mygtuką „Toliau“ sukuriamas šablono elementas.

Šablono **Komandiruotės** sudėtis

Laukelio pavadinimas:

Ar laukeliuose bus saugomos pasikartojančios reikšmės?  Ne. Nežinau  
 Tikrai taip

Pasikartojančias reikšmes rekomenduojama suvesti į atskirą sąrašą.  
Iš kurio sąrašo bus imamos reikšmės:  
(Jei sąrašo dar nėra sukurkite ir suveskite duomenis)

Vardas  Pavardė  Pareigos  Priėmimo data

Ar galima įvesti vienodas reikšmes?  Taip  
 Ne

Ar privaloma įvesti reikšmę?  Taip, privaloma  
 Galima palikti tuščią laukelį

20 pav. Langas „Naujo šablono kūrimas“

### Langas „Naujas įrašas“

Kai vartotojas užpildo laukelius ir paspaudžia mygtuką „Išsaugoti“ sistema atlieka duomenų patikrinimą. Duomenų tikrinimas atliekamas pagal tai ką nurodė vartotojas kurdamas šabloną. Kadangi kuriant šabloną „Pareigos“ buvo nurodyta, kad reikšmės negali kartotis sistema rodo klaidą. (pav. 21)

Pareigos

Laukelyje **Pareigos** nurodyta reikšmė jau yra! Reikšmės negali kartotis

21 pav. Langas „Naujas įrašas“

## 5.3 Eksperimentinio tyrimo kokybės vertinimas (testavimas)

Testavimo objektas – duomenų bazės kūrimo panaudojant duomenų pildymo formų šablonus informacinė sistema.

Testavimo procesas turi keletą tikslų:

- Surasti ir pašalinti neatitikimus reikalavimams;
- Surasti pakankamai mažą klaidų skaičių, kuris parodytų sukurtos priemonės stabilumą, veikimo teisingumą bei darbo našumą;
- Užtikrinti, kad būtų pateikta tinkamai veikianti programinė priemonė.

### 5.3.1 Testavimo planas

Testavimo planas apibrėžia sistemos testavimo strategiją, nenurodant sistemos testavimo smulkmenų.

Testavimas susideda iš kelių etapų:

- Bendras sistemos testavimas:
  - o Vienetų testavimas.
  - o Sistemos integracijos testavimas.
  - o Sistemos sąsajos testavimas.
- Priėmimo testavimas.
- Aukšto lygio testavimas.

Sistemos dalių testavimas – programinės įrangos dalių, jų susietumo testavimas, baigus programuoti sistemos dalis bei apjungus jas į sistemą. Šis testavimas atliekamas prieš bandomąjį testavimą pas užsakovą. Pagrindiniai bendrojo testavimo uždaviniai yra patikrinti/užtikrinti sistemos bei jos dalių tinkamumą naudoti bei vykdyti jų tolesnį testavimą.

Priėmimo testavimas turi užtikrinti, kad sistema atitinka vartotojo poreikius, aprašytus sistemos reikalavimų specifikacijoje. Remiamasi funkciniais bei nefunkciniais sistemos reikalavimais.

Aukšto lygio testavimas atliekamas pas užsakovą, įdiegiant į užsakovo kompiuterį sukurtą programinę priemonę. Čia yra tikrinami tokie nefunkciniai reikalavimai, kaip panaudojamumas, sistemos išvaizda, vykdymo savybės, programinės priemonės veikimo reikalavimai, reikalavimai sistemos priežiūrai ir t.t.

### **5.3.2 Testavimo strategija**

Visų aprašytų testų strategija – kuo efektyvesnis ir kokybiškesnis programinės priemonės ištestavimas. Kur įmanoma, naudojamos testų automatizavimo priemonės.

Nuo ribotų testuotojo išteklių priklauso kiek tinkamai bus apgalvota ir efektyvi testavimo strategija. Tai gali nulemti projekto pabaigą.

Toliau yra aprašomos vienetų, sistemos integracijos, priėmimo bei aukšto lygio testavimo strategijos.

#### **Vienetų testavimas**

Atskirų vienetų testavimui bus naudojami struktūrinis ir funkcinis testavimo metodai.

Testuojant struktūriniu būdu atsižvelgiama į algoritmo struktūrą („baltos dėžės“ principas) ir tikrinamas vieneto veikimas kada kiekvienas vieneto sakiny yra įvykdomas bent po vieną kartą.

Testuojant funkcinio būdu tikrinama ar pagal metodui perduotus pradinis duomenis gaunami laukiami rezultatai. Duomenys parenkami neatsižvelgiant į metodo struktūrą („juodos dėžės“ principas). Testinių duomenų rinkiniai sudaromi pagal sistemos specifikaciją.

#### **Integravimo testavimas**

Atlikus vienetų testavimą, jie apjungiami dalimis ir testuojama jų sąveika. Naudojamas stambinant (angl. „bottom-up“) integracinis testavimas. Ištestuoti žemiausio lygio vienetai

apjungiami į posistemas (modulius). Vėliau sąveikaujantys moduliai apjungiami tarpusavyje. Apjungimas užbaigiamas „valdymo“ moduliu (apjungia visas sistemos dalis).

### **Priėmimo testavimas**

Priėmimo testavimas atliekamas pagal „juodos dėžės“ principą. Programinė priemonė bus nuodugnai testuojama pagal reikalavimų specifikaciją, taip siekiant nustatyti ar sistema atitinka vartotojo poreikius. Pastebėjus, kad programinė priemonė neatitinka vartotojo poreikių, apibrėžtų reikalavimų specifikacijoje, registruojama klaida, kuri šalinama artimiausiu metu. Jei testavimo metu išaiškėja nauji vartotojo poreikiai, neįtraukti į reikalavimų specifikaciją, registruojami programinės priemonės tobulinimo darbai ir keičiama reikalavimų specifikacija.

### **Aukšto lygio testavimas**

Užbaigtos programinės priemonės testavimas, siekiant išsiaiškinti jos kokybę. Tam naudojami tokie testavimo metodai:

#### **- Atsistatymas**

Šio testavimo metu bus tikrinama ar turi įtakos netikėtas kompiuterio, kuriame įdiegta programinė priemonė, išjungimas teisingam programinės priemonės veikimui. Bus siekiama parodyti, kad programinė priemonė gali, po tokių situacijų, tęsti darbą be klaidų ir duomenų praradimo.

#### **- Našumas**

Šio testavimo metu bus stebima kaip programinė priemonė veikia, kai kompiuteris, kuriame įdiegta programinė priemonė, stipriai apkrautas, t.y. kai kompiuteris dirba kelis darbus, kurie naudoja daug procesoriaus ir kompiuterinio tinklo resursų.

Kiekvienam testavimo atvejui buvo parenkami realūs testavimo duomenys, atsižvelgiant į reikalavimų specifikaciją.

### **5.3.3 Testavimo resursai ir rezultatai**

Testavimo techninė bei programinė bazė:

- kompiuteris:
  - o Procesorius – AMD Athlon XP 2500+, 1.83 GHz
  - o Operatyvioji atmintis – 1 GB
  - o Operacinė sistema – MS Windows XP Pro SP2
- kompiuteris:
  - o Procesorius – Intel Core 2, 2 GHz
  - o Operatyvioji atmintis – 1 GB
  - o Operacinė sistema – MS Windows XP Pro SP2
- serveris:



- Procesorius – Intel Core 2, 2 GHz
- Operatyvioji atmintis – 1 GB
- Operacinė sistema – MS Windows XP Pro SP2
- HTTP serveris – Apache 2.2.3 (Win32) Server su PHP palaikymu
- Duomenų bazių serveris – MySQL 5.0.24a

Gauti testų rezultatai kaupiami lentelėje (žr. lentelė Nr. 4).

**Lentelė Nr. 8 Testavimo rezultatų lentelė**

Testas	Laukiamas rezultatas	Pavyko/Nepavyko

### 5.3.4 Testavimo eiga

Testavimo metu buvo siekiama įvertinti realizuotos informacinės sistemos teikiamą funkcionalumą, rasti bei ištaisyti sistemos klaidas, pašalinti trūkumus. Informacinės sistemos vienetų testavimas buvo vykdomas dviem etapais. Pirmasis vienetų testavimas etapas buvo vykdomas programavimo metu. Buvo aptikta ir pašalinta didžioji dalis informacinės sistemos klaidų. Realizavus informacinę sistemą, buvo atliktas antrasis vienetų testavimo etapas bei įvykdytas integravimo testavimas. Testavimo metu buvo laikomasi apibrėžtų metodikų. Tuo tikslu buvo sukurtas testavimo planas ir testavimo strategija. Visose testavimo procedūrose aptikti defektai buvo sėkmingai pašalinti. Testavimo medžiaga pateikiama 2 priede „Sistemos testavimo dokumentacija”.

Priėmimo testavimo metu buvo patikrinta ar sistema atitinka reikalavimų specifikaciją. Šio testavimo metu buvo išbandyti informacinės sistemos teikiami funkcionalumai ir patikrinta ar šie funkcionalumai atitinka specifikaciją. Vykdamas priėmimo testavimą, testavimo dalyviai turėjo užpildyti lentelę (žr. lentelė Nr. 4) su pateiktais klausimais apie informacinę sistemą. Apklausos anketoje reikėjo pažymėti įvertinimą balais skalėje nuo 1 (žemas įvertinimas) iki 5 (aukštas įvertinimas). Apklausoje dalyvavo 8 Šiaulių AVMI Informatikos skyriaus darbuotojai.

Apibendrinus anketų duomenis nustatyta, kad visi reikalavimai išpildyti ir vartotojus realizuota informacinė sistema tenkina. Apibendrinti rezultatai pateikiami lentelėje Nr. 5

**Lentelė Nr. 9 Apibendrinti apklausos rezultatai**

Nr.	Charakteristika	Vidurkis
1.	Ar lengvai suprantama vartotojo sąsaja?	4,75
2.	Ar lengvai suprantami informaciniai pranešimai?	4,625
3.	Ar tenkina sistemos teikiamas funkcionalumas?	4,5

4.	Įvertinkite sistemos dizainą	4
5.	Sistemos teikiama nauda	4,25
6.	Įvertinkite sistemos darbo greitį	4,875

### 5.3.5 Testavimo išvados

- Realizuotos informacinės sistemos testavimas patvirtino, kad minėta sistema realizuota kokybiškai. Testavimo metu aptiktos klaidos buvo sėkmingai pašalintos.
- Informacinės sistemos vartotojų apklausa parodė, kad realizuota informacinė sistema tenkina juos tenkina.
- Atlikus priėmimo testavimą bei sistemos kokybės vertinimą nustatyta, kad visos specifikacijoje apibrėžtos funkcijos yra realizuotos. Nuspręsta, jog duomenų bazės kūrimo panaudojant duomenų pildymo šablonus informacinė sistema tinkama naudojimui.
- Sąlyginai žemas sistemos dizaino įvertinimas rodo, kad reikia atlikti papildomus darbus siekiant pagerinti šią sistemos charakteristiką.

## 6. IŠVADOS

1. Atlikta programinių priemonių analizė, skirta nustatyti jų tinkamumą iškelto problemos sprendimo eksperimentiniam realizavimui. Iš egzistuojančių programinių priemonių nė viena netiko sprendimui realizuoti. Nuspręsta eksperimentinę programą realizuoti PHP programavimo kalba panaudojant MySQL duomenų bazių serverį.
2. Sukurtas algoritmas, kuris realizuotas dialogo pagalba leidžia kurti tam tikros formos duomenų pildymo šabloną, pagal kurį kuriamos duomenų bazės lentelės atitinka pirmąją normalinę formą.
3. Pagal sukurtą algoritmą sukurta informacinė sistema, kurios pagalba projektuojama ir kuriama duomenų bazė bei interaktyviai atnaujinamos duomenų manipuliavimo formos.
4. Parinkta komponentinė sistemos architektūra kilus poreikiui leis minimaliomis sąnaudomis prijungti naujus modulius.
5. Pasirinktas projektavimo principas ir sistemos kūrimo priemonės pilnai pasiteisino vykdant iškelto uždavinio. Suprojektuota sistema atitinka projekto specifikaciją bei atlieka numatytas funkcijas.
6. Prie sukurtos sistemos prijungus papildomus modulius, kuriuose būtų aprašytos SQL instrukcijos kitoms duomenų bazėms, sukurtą sistemą galima naudoti ne tik darbui su MySQL duomenų bazėmis.
7. Sukurtos informacinės sistemos testavimas parodė, kad sistema tenkina visus jai keliamus reikalavimus. Vartotojai sistemos darbą įvertino teigiamai.
8. Kadangi informacinėje sistemoje transakcijos nėra sudėtingos ir ilgai vykdomos serverio apkrovimas informacinės sistemos darbui įtakos neturi.
9. Į suprojektuotą ir sukurtą informacinę sistemą papildomai įdiegus vartotojų autentifikavimą ir sistemą patalpinus internetu pasiekiamame serveryje galima naudoti ne vienoje organizacijoje, bet ir jos filialuose.
10. Sukurta informacinė sistema nepriklauso nuo naudojamos serverio ar kliento PK operacinės sistemos, reikalinga naršyklė kuri dirba su JavaScript scenarijų kalba.

## 7. LITERATŪRA

1. *CHARONWARE internetinė svetainė*. [interaktyvus] [žiūrėta 2006-11-03]. Prieiga internetu: <<http://www.casestudio.com/enu/>>
2. *Programming Language Popularity* [interaktyvus] Prieiga internetu: <<http://www.langpop.com/>>
3. BUTKIENĖ, R. *Reikalavimų programoms analizė: paskaitų konspektas*. [interaktyvus] [žiūrėta 2006-11-03]. Prieiga internetu: <<ftp://isd.ktu.lt/Isd/Butkiene/T120M015/>>
4. RUTKAUSKAITĖ, R.; NEMURAITĖ, L. *Šablonų naudojimas kuriant duomenų apdorojimo sistemas internete*. Informacinės technologijos [interaktyvus] 2004: Konferencijos pranešimų medžiaga, X sekcija. Kaunas, 2004, p. 45-51. [žiūrėta 2006-11-03]. Prieiga internetu: <[http://www.ktu.edu/lt/apie\\_renginius/konferencijos/2006/k6\\_02/IT2004/Sekc10.pdf](http://www.ktu.edu/lt/apie_renginius/konferencijos/2006/k6_02/IT2004/Sekc10.pdf)>
5. *Web Server Software Usage: 10 Most Popular* [interaktyvus] Prieiga internetu: <http://itmanagement.earthweb.com/entdev/article.php/3673386/Web-Server-Software-Usage-10-Most-Popular.htm>
6. TARGAMADZĖ A. *Programų inžinerija: paskaitų skaidrės* [interaktyvus] [žiūrėta 2011-05-20]. Prieiga internetu: <[http://oras.if.ktu.lt/moduliai/T120M026/Skaidres/PIP\\_IV\\_dalis\\_2005.pdf](http://oras.if.ktu.lt/moduliai/T120M026/Skaidres/PIP_IV_dalis_2005.pdf)>
7. Vikipedija – laisvoji enciklopedija [interaktyvus] [žiūrėta 2011-05-20] Prieiga internetu: <http://lt.wiktionary.org/wiki/objektas>

## **8. SANTRAUKA ANGLŲ KALBA**

### **SUMMARY**

Databases are designed according to specific requirements, and in most cases it is carried out manually. Having database already designed, software tools to manage the particular database are developed. Usually the result meets user's needs, but just on a short term basis. Since the requirements keep changing constantly, gap between information to be processed and information possible to process in practice is growing bigger, so it is complicated to keep the database updated according to user's needs. Shifting needs have significant influence on information system support input, which can become a challenge to minor businesses. Therefore particular system users have to modify it themselves to meet the changing requirements of data processing.

This final thesis analyses the development of database using data filling templates designed by the user. According to the templates designed by the user, information system user interface for data processing keeps changing as well.

The purpose is automatic design of database and its management system, and development of algorithm to automate database and its management system. In case the system is based on this algorithm, its user having very little knowledge of database development still is able to design a database and to shape its management system according to his needs. It is possible to modify database structure – to add or to remove columns – at any time. Database structure is adjusted by the templates kept in database tables. Based on the mentioned algorithm, the user is able to develop acceptable and easy to use data entry template.

## 9. TERMINŲ IR SANTRUMPŲ ŽODYNAS

PĮ – programinė įranga;

HTTP - (trumpinys nuo *angl. HyperText Transfer Protocol*) – pagrindinis metodas pasiekti informaciją pasauliniame tinkle;

IS – informacinė sistema;

DFD – (trumpinys nuo *angl. Data Flow Diagram*) – duomenų srautų diagrama

UML – (trumpinys nuo *angl. The Unified Modeling Language*) – Vieninga modeliavimo kalba;

ODBC - standartizuota taikomosios programinės įrangos programavimo sąsaja prisijungimui prie duomenų bazių (*angl. Open Database Connectivity*)

DBVS – duomenų bazių valdymo sistema;

HTML - (trumpinys nuo *angl. Hyper text Markup Language*) – tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete;

Perl – aukšto lygio dinaminė interpretuojama programavimo kalba;

PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba, sukurta 1997 m. ir specialiai pritaikyta interneto svetainių kūrimui;

Python – yra interpretuojama, interaktyvi programavimo kalba sukurta Guido van Rossumo 1990 metais;

JavaScript – objektiškai orientuota scenarijų programavimo kalba, besiremianti prototipų principu. Dažniausiai kalba naudojama internetinių puslapių interaktyvumo realizacijai;

IT – informacinės technologijos;

PK – personalinis kompiuteris;

Apache – tai atviro kodo HTTP serveris skirtas šiuolaikinėms operacinėms sistemoms, įskaitant UNIX ir Windows.

## 10. PRIEDAI

### 10.1 1 priedas. Duomenų bazės lentelių aprašai

Pateikiami duomenų bazės lentelių, pateiktų skyriuje „Duomenų vaizdas“ aprašymai

Lentelė Nr. 10 Duomenų bazės lentelės MDB\_DUOMENU\_TIPAI struktūra

Pavadinimas	Reikšmės tipas	Privaloma įvesti	Apibūdinimas
Tipas	VARCHAR(10)	Taip	Saugomos reikšmės tipas
Pavadinimas	VARCHAR(50)	Taip	Saugomos reikšmės tipo pavadinimas
<b>Raktai</b>	<b>Tipas</b>	<b>Pirminis</b>	

Lentelė Nr. 11 Duomenų bazės lentelės MDB\_SABLONAI struktūra

Pavadinimas	Reikšmės tipas	Privaloma įvesti	Apibūdinimas
Id	INT(11)	Taip	Saugomas šablono identifikatorius
Tipas	CHAR(1)	Taip	Saugoma informacija apie tai ar duotai lentelei yra sukurta loginė lentelė – vaizdas
Pavadinimas	VARCHAR(50)	Taip	Saugomas šablono pavadinimas
Komentaras	VARCHAR(100)	Taip	Saugomas šablono komentaras
Vartotojo_id	INT(11)	Taip	Vartotojo, sukūrusio šabloną, identifikatorius
<b>Raktai</b>	<b>Id</b>	<b>Pirminis</b>	

Lentelė Nr. 12 Duomenų bazės lentelės MDB\_SABLONU\_ELEMENTAI struktūra

Pavadinimas	Reikšmės tipas	Privaloma įvesti	Apibūdinimas
Id	INT(11)	Taip	Saugomas įrašo identifikatorius
Sablono_id	INT(11)	Taip	Saugomas šablono identifikatorius
Eiles_nr	INT(11)	Taip	Saugoma šablono sudedamųjų dalių rikiavimo tvarka
Stulpelio_nr	INT(11)	Taip	Saugoma šablono sudedamųjų dalių rikiavimo tvarka
Pavadinimas	VARCHAR(50)	Taip	Saugomas šablono laukelio pavadinimas

Tipas	VARCHAR(15)	Ne	Saugomas šablono laukelio duomenų tipas
Saltinio_id	INT(11)	Ne	Saugomas šablono identifikatorius. Jeigu duomenys imami iš kitos duomenų lentelės užpildytos pagal kitą šabloną
Privaloma	VARCHAR(8)	Taip	Saugoma informacija ar vartotojas veddamas duomenis pagal šabloną privalo užpildyti konkretų laukelį
Pradine_reiksme	VARCHAR(100)	Ne	Saugoma reikšmė, kurią sistema turi įrašyti į laukelį jei numatyta, kad vartotojas gali kai kuriems laukeliams nenurodyti pradinės reikšmės
Vienodos_reiksmes	VARCHAR(1)	Taip	Saugoma informacija apie tai ar galima vesti pasikartojančias reikšmes į nurodytą duomenų lentelės stulpelį
<b>Raktai</b>	<b>Id</b>	<b>Pirminis</b>	
	<b>Sablono_id</b>	<b>Išorinis</b>	

Lentelė Nr. 13 Duomenų bazės lentelės MDB\_ELEMENTU\_SUDETIS struktūra

<b>Pavadinimas</b>	<b>Reikšmės tipas</b>	<b>Privaloma įvesti</b>	<b>Apibūdinimas</b>
Id	INT(11)	Taip	Įrašo identifikatorius
Elemento_id	INT(11)	Taip	Elemento identifikatorius
Naudojamo_elemento_id	INT(11)	Taip	Naudojamų elementų identifikatoriai
<b>Raktai</b>	<b>Id</b>	<b>Pirminis</b>	
	<b>Elemento_id</b>	<b>Išorinis</b>	
	<b>Naudojamo_elemento_id</b>	<b>Išorinis</b>	



## 10.2 2 priedas. Sistemos testavimo dokumentacija

### 1. Testavimo tikslai ir objektai

Kuriant programinę įrangą siekiama sukurti ir pateikti produktą, kuris turėtų kuo mažiau klaidų bei defektų. Testavimo tikslas — aptikti kaip galima daugiau sistemos klaidų, kad jas būtų galima pašalinti iki informacinės sistemos diegimo. Tai leidžia pateikti produktą su kiek galima mažesniu klaidų kiekiu. Kitas ne mažiau svarbus testavimo tikslas yra įvertinti ar sukurta programinė įranga atitinka vartotojų reikalavimus ir specifikaciją.

Šiame dokumente pateikiama bendra testavimo apžvalga, testavimo planas bei testavimo eiga. Pateikiami testavimo atvejai leis užtikrinti kaip galima mažesnę informacinės sistemos klaidų kiekį ir kaip galima aukštesnę kuriamos sistemos kokybę.

### 2. Testavimo planas

Šiame skyriuje pateikiamos testavimo procedūros, kurių pagalba testuojant informacinę sistemą galima pasiekti užsibrėžtus tikslus.

Pateikiamos testuojamos sistemos formos (langai):

**Pradinis langas** – skirtas atvaizduoti informacinės sistemos pavadinimą bei prisijungti prie sistemos. Prisijungimo formoje būtina įvesti vartotojo vardą bei slaptažodį. Paspaudus mygtuką „Prisijungti“, tikrinama ar vartotojas suvedė teisingus prisijungimo parametrus ir ar vartotojui nėra uždėta galiojimo pabaiga. Suvedus teisingus prisijungimo parametrus vartotojui, atsižvelgiant į turimas teises bei roles sistemoje, generuojamas ir pateikiamas pagrindinis sistemos langas. Jeigu vartotojas neteisingai suvedė prisijungimo parametrus sistema praneša apie nesėkmingą bandymą prisijungti.

**Sistemos meniu** – skirtas sistemos valdymui, perėjimui iš vieno lango į kitą.

**Langas „Šablonai“** – skirtas duomenų pildymo formų šablono sąrašo peržiūrai.

**Langas „Naujo šablono kūrimas“** – skirtas duomenų pildymo šablono pavadinimui bei numanomam duomenų laukelių skaičiui įvesti.

**Langas „Naujo šablono kūrimas. 2 žingsnis“** – skirtas kuriamo šablono komentaro, laukelių pavadinimų, duomenų tipų ir kitoms reikšmėms nurodyti. Jei yra teisingai užpildytas nors vienas įrašas apie duomenų pildymo šablono sudedamąją dalį ir yra paliktų tuščių laukelių „Laukelio pavadinimas“, tai paspaudus mygtuką „Sukurti šabloną“ sistema ignoruos tuščius laukelius. Į jį galima pateikti tik iš lango „Naujo šablono kūrimas“.

**Langas „Šablono redagavimas“** – skirtas duomenų pildymo šablono redagavimui. Į jį galima pateikti tik iš lango „Šablonai“. Testavimo metu atidaromas šablono „pareigos2“ koregavimo langas.

**Pagrindinis sistemos langas** – skirtas vartotojui pateikti informaciją apie sistemoje jam suteiktas teises bei roles.

**Langas „Naujas įrašas“** – skirtas duomenų vedimui pagal sukurtą duomenų pildymo šabloną. Į jį galima pateikti tik iš pagrindinio lango paspaudus mygtuką „Naujas įrašas“ esantį šalia šablono pavadinimo. Testavimo metu nauji įrašai vedami pagal duomenų pildymo šabloną „pareigos2“.

**Langas „Duomenų peržiūra“** – skirtas duomenims suvestiems pagal duomenų pildymo šabloną peržiūrai. Į jį patenkama iš pagrindinio lango paspaudus mygtuką „Peržiūra“.

**Langas „Duomenų redagavimas“** – skirtas duomenų redagavimui pagal sukurtą duomenų pildymo šabloną. Į jį galima pateikti tik iš pagrindinio lango paspaudus mygtuką „Peržiūra“ esantį šalia šablono pavadinimo. Testavimo metu duomenys redaguojami pagal duomenų pildymo šabloną „pareigos2“.

### 3. Testavimo resursai bei rezultatai

Jokių papildomų techninių ar programinių resursų sistemos testavimui nereikia. Naudojami turimi techniniai ir programiniai resursai:

Serveris:

- Procesorius – Intel Core 2, 2 GHz
- Operatyvioji atmintis – 1 GB
- Operacinė sistema – MS Windows XP Pro SP2
- HTTP serveris – Apache 2.2.3 (Win32) Server su PHP palaikymu
- Duomenų bazių serveris – MySQL 5.0.24a

Kompiuteris:

- Procesorius – AMD Athlon XP 2500+, 1.83 GHz
- Operatyvioji atmintis – 1 GB
- Operacinė sistema – MS Windows XP Pro SP2

Kompiuteris:

- Procesorius – Intel Core 2, 2 GHz
- Operatyvioji atmintis – 1 GB
- Operacinė sistema – MS Windows XP Pro SP2

Testų rezultatai bus kaupiami lentelėje:

Lentelė Nr. 14 Testų rezultatų saugojimo lentelė

Testas	Laukiamas rezultatas	Pavyko/Nepavyko
--------	----------------------	-----------------

--	--	--

#### 4. Testavimo procedūra

Skyriuje pateikiamos testavimo procedūros, kurios bus naudojamos atliekant programinės įrangos testavimą.

##### Vienetų testavimas

Žemiau lentelėse pateikiami atskirų vienetų testavimo atvejai bei rezultatai.

Lentelė Nr. 15 Sistemos meniu

Testas	Laukiamas rezultatas	Testo rezultatas
Paspaudžiamas meniu punktas „Į pradžia“	Vartotojui pateikiamas pagrindinis sistemos langas	Pavyko
Paspaudžiamas meniu punktas „Šablonų sąrašas“	Vartotojui pateikiamas langas su sistemoje sukurtų duomenų pildymo šablonų sąrašu	Pavyko
Paspaudžiamas meniu punktas „Sukurti šabloną“	Vartotojui pateikiamas langas skirtas kurti naują šabloną	Pavyko

Lentelė Nr. 16 Langas „Šablonai“

Testas	Laukiamas rezultatas	Testo rezultatas
Pasirenkamas punktas „Šablonų sąrašas“	Pateikiamas sistemos duomenų pildymo šablonų sąrašas	Pavyko
Paspaudžiamas mygtukas „Naujo šablono kūrimas“	Vartotojui pateikiama naujo šablono kūrimo forma	Pavyko
Pateiktame šablonų sąraše spragtelima ant pasirinkto šablono	Vartotojui pateikiamas to šablono, ant kurio įrašo spragtelta, koregavimo langas	Pavyko

Lentelė Nr. 17 Langas „Naujo šablono kūrimas“

Testas	Laukiamas rezultatas	Testo rezultatas
Neužpildžius nė vieno laukelio paspaudžiamas mygtukas „Sukurti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir išvardina laukelius, kurie neužpildyti.	Pavyko
Užpildomas laukelis „Pavadinimas“. Laukelis „Laukelių skaičius“ paliekamas neužpildytas ir paspaudžiamas	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąraše pateikia laukelį „Laukelių skaičius“	Pavyko

Testas	Laukiamas rezultatas	Testo rezultatas
mygtukas „Sukurti“		
Užpildomas laukelis „Laukelių skaičius“. Laukelis „Pavadinimas“ paliekamas neužpildytas ir paspaudžiamas mygtukas „Sukurti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Pavadinimas“	Pavyko
Užpildomi visi laukeliai. Laukelyje „Laukelių skaičius“ įrašoma reikšmė „A“ ir paspaudžiamas mygtukas „Sukurti“	Sistema pateikia pranešimą, kad neteisingai nurodytas laukelių skaičius	Pavyko
Užpildomi visi laukeliai. Laukelyje „Laukelių skaičius“ įrašoma reikšmė „0“ ir paspaudžiamas mygtukas „Sukurti“	Sistema pateikia pranešimą, kad neteisingai nurodytas laukelių skaičius	Pavyko
Užpildomi visi laukeliai. Laukelyje „Pavadinimas“ įrašoma reikšmė „pareigos“ ir paspaudžiamas mygtukas „Sukurti“	Sistema praneša, kad duomenų pildymo šablono pavadinimas „pareigos“ jau registruotas sistemoje	Pavyko
Laukelyje „Pavadinimas“ įrašoma reikšmė „pareigos2“, laukelyje „Laukelių skaičius“ įrašoma reikšmė „1“ ir paspaudžiamas mygtukas „Sukurti“	Atidaromas langas „Naujo šablono kūrimas. 2 žingsnis“	Pavyko

Lentelė Nr. 18 Langas „Naujo šablono kūrimas. 2 žingsnis“

Testas	Laukiamas rezultatas	Testo rezultatas
Neužpildžius nė vieno laukelio paspaudžiamas mygtukas „Sukurti šabloną“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir išvardina laukelius, kurie neužpildyti.	Pavyko
Užpildomas laukelis „Laukelio pavadinimas“, laukelyje „Tipas“ paliekama reikšmė „Pagal	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Komentaras“. Taip pat pranešime rašoma,	Pavyko

Testas	Laukiamas rezultatas	Testo rezultatas
pavyzdį“. Laukelis „Komentaras“ paliekamas neužpildytas ir paspaudžiamas mygtukas „Sukurti šabloną“	kad neteisingai nurodytas laukelio duomenų pavyzdys.	
Laukelyje esančiame šalia mygtuko „ pridėti“ įrašoma reikšmė „2“ ir paspaudžiamas mygtukas pridėti	Atidarytame lange papildomai sukuriamos dvi eilutės skirtos dviejų papildomų duomenų pildymo šablono sudedamųjų dalių aprašymui	Pavyko
Užpildomi laukeliai nurodant sekančias reikšmes: Laukelyje „Komentaras“ – „Bandome įvairius tipus“, laukelyje „Laukelio pavadinimas 1“ – „Sveikas skaičius“, laukelyje „Laukelio pavadinimas 2“ – „Realus skaičius“, laukelyje „Laukelio pavadinimas 3“ – „Tekstas“, laukelyje „Pavyzdys 1“ – „123456“, laukelyje „Pavyzdys 2“ – „123.456“, laukelyje „Pavyzdys 3“ – „1a2b3c4d5e6f“.	<p>Sistema sukuria duomenų pildymo šabloną, t.y.:</p> <p>1. duomenų lentelėje „MDB_SABLONAI“ sukuriamas įrašas su šablono pavadinimu „pareigos2“, komentaru „Bandome įvairius tipus“, prisijungusio prie sistemos vartotojo ID ir sukūrimo data.</p> <p>2. duomenų lentelėje „MDB_SABLONU_SUDETIS“ sukuriami trys įrašai:</p> <ul style="list-style-type: none"> <li>- su laukelio pavadinimu „Sveikas skaičius“, tipas „INT“, privaloma „Taip“, vienodos reikšmės „Taip“, bei prisijungusio prie sistemos vartotojo ID ir sukūrimo data.</li> <li>- su laukelio pavadinimu „Realus skaičius“, tipas „FLOAT“, privaloma „Taip“, vienodos reikšmės „Taip“, bei prisijungusio prie sistemos vartotojo ID ir sukūrimo data.</li> <li>- su laukelio pavadinimu „Tekstas“, tipas „VARCHAR(12)“, privaloma „Taip“, vienodos reikšmės „Taip“, bei prisijungusio prie sistemos vartotojo ID ir sukūrimo data.</li> </ul> <p>3. sukuriama duomenų lentelė „MDB_TABLE_7 (7 – šablono įrašo identifikatorius duomenų lentelėje „MDB_SABLONAI“). Lentelę sudaro 4 stulpeliai:</p> <ul style="list-style-type: none"> <li>- ID – įrašo identifikatorius. Duomenų tipas INT. Pirminis raktas.</li> <li>- Field_23 (23 – šablono sudedamosios dalies</li> </ul>	Pavyko

Testas	Laukiamas rezultatas	Testo rezultatas
	<p>įrašo identifikatorius duomenų lentelėje „MDB_SABLONU_SUDETIS“). Duomenų tipas INT, privalomas „Taip“, vienodos reikšmės „Taip“</p> <ul style="list-style-type: none"> <li>- Field_24. Duomenų tipas FLOAT, privalomas „Taip“, vienodos reikšmės „Taip“</li> <li>- Field_25. Duomenų tipas VARCHAR(12), privalomas „Taip“, vienodos reikšmės „Taip“.</li> </ul>	

**Lentelė Nr. 19 Langas „Šablono redagavimas“**

Testas	Laukiamas rezultatas	Testo rezultatas
Atidarytoje formoje laukelyje esančiame šalia mygtuko „ pridėti“ įrašoma reikšmė „1“ ir paspaudžiamas mygtukas pridėti	Atidarytame lange papildomai sukuriama eilutė skirta papildomos duomenų pildymo šablono sudėtinės dalies aprašymui.	Pavyko
Užpildomi papildomą sudedamąją šablono dalį aprašantys laukeliai nurodant sekančias reikšmes: laukelyje „Laukelio pavadinimas 4“ – „Data“ laukelyje „Tipas 4“ parenkame reikšmę „Data“	<p>Sistema išsaugo duomenų pildymo šablono pakeitimus, t.y.:</p> <ul style="list-style-type: none"> <li>- duomenų lentelėje „MDB_SABLONU_SUDETIS“ sukuriamas įrašas: <ul style="list-style-type: none"> <li>- su laukelio pavadinimu „Data“, tipas „DATE“, privaloma „Taip“, vienodos reikšmės „Taip“, bei prisijungusio prie sistemos vartotojo ID ir sukūrimo data.</li> </ul> </li> <li>4. duomenų lentelė „MDB_TABLE_7 (7 – šablono įrašo identifikatorius duomenų lentelėje „MDB_SABLONAI“) papildoma vienu stulpeliu: <ul style="list-style-type: none"> <li>- Field_26 (26 – šablono sudedamosios dalies įrašo identifikatorius duomenų lentelėje „MDB_SABLONU_SUDETIS“). Duomenų tipas DATE, privalomas „Taip“, vienodos reikšmės „Taip“.</li> </ul> </li> </ul>	Pavyko

**Lentelė Nr. 20 Pagrindinis sistemos langas**

Testas	Laukiamas rezultatas	Testo rezultatas

Testas	Laukiamas rezultatas	Testo rezultatas
Paspaudžiamas mygtukas „Naujas įrašas“ esantis šalia šablono pavadinimo	Pateikiamas langas „Naujas įrašas“	Pavyko
Paspaudžiamas mygtukas „Duomenų peržiūra“ esantis šalia šablono pavadinimo	Pateikiamas langas „Duomenų peržiūra“	Pavyko

**Lentelė Nr. 21 Langas „Naujas įrašas“**

Testas	Laukiamas rezultatas	Testo rezultatas
Neužpildžius nė vieno laukelio paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir išvardina laukelius, kurie neužpildyti.	Pavyko
Užpildomi laukeliai „Realus skaičius“, „Tekstas“ ir „Data“. Laukelis „Sveikas skaičius“ paliekamas neužpildytas ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Sveikas skaičius“	Pavyko
Užpildomi laukeliai „Sveikas skaičius“, „Tekstas“ ir „Data“. Laukelis „Realus skaičius“ paliekamas neužpildytas ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Realus skaičius“	Pavyko
Užpildomi laukeliai „Sveikas skaičius“, „Realus skaičius“ ir „Data“. Laukelis „Tekstas“ paliekamas neužpildytas ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Tekstas“	Pavyko
Užpildomi laukeliai „Sveikas skaičius“, „Realus skaičius“ ir „Tekstas“. Laukelis „Data“ paliekamas neužpildytas ir paspaudžiamas mygtukas	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Data“	Pavyko

Testas	Laukiamas rezultatas	Testo rezultatas
„Įrašyti“		
Užpildomi visi laukeliai. Laukelyje „Sveikas skaičius“ įrašoma reikšmė „2000-11“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Sveikas skaičius reikšmės tipas! Turi būti sveikas skaičius!“	Pavyko
Užpildomi visi laukeliai. Laukelyje „Realus skaičius“ įrašoma reikšmė „2000-11“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Realus skaičius reikšmės tipas! Turi būti realus skaičius!“	Pavyko
Užpildomi visi laukeliai. Laukelyje „Data“ įrašoma reikšmė „2000“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Data reikšmės tipas! Turi būti data!“	Pavyko
Teisingai užpildomi visi laukeliai ir paspaudžiamas mygtukas „Įrašyti“	Sukuriamas naujas įrašas ir sistema pateikia pranešimą „Duomenys išsaugoti“	Pavyko

**Lentelė Nr. 22 Langas „Duomenų peržiūra“**

Testas	Laukiamas rezultatas	Testo rezultatas
Atidaromas langas „Duomenų peržiūra“	Sistema pateikia suvestų pagal duomenų pildymo šabloną duomenų sąrašą.	Pavyko

**Lentelė Nr. 23 Langas „Duomenų redagavimas“**

Testas	Laukiamas rezultatas	Testo rezultatas
Atidarytoje formoje išvalomi visi laukeliai ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir išvardina laukelius, kurie neužpildyti.	Pavyko
Atidarytoje formoje išvalomas laukelis „Sveikas skaičius“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Sveikas skaičius“	Pavyko
Atidarytoje formoje išvalomas	Sistema pateikia pranešimą, kad neužpildyti būtini	Pavyko



Testas	Laukiamas rezultatas	Testo rezultatas
laukelis „Realus skaičius“ ir paspaudžiamas mygtukas „Įrašyti“	laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Realus skaičius“	
Atidarytoje formoje išvalomas laukelis „Tekstas“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Tekstas“	Pavyko
Atidarytoje formoje išvalomas laukelis „Data“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą, kad neužpildyti būtini laukai ir neužpildytų laukelių sąrašė pateikia laukelį „Data“	Pavyko
Atidarytoje formoje išvalomas laukelyje „Sveikas skaičius“ įrašoma reikšmė „2000-11“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Sveikas skaičius reikšmės tipas! Turi būti sveikas skaičius!“	Pavyko
Atidarytoje formoje išvalomas laukelyje „Realus skaičius“ įrašoma reikšmė „2000-11“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Realus skaičius reikšmės tipas! Turi būti realus skaičius!“	Pavyko
Atidarytoje formoje išvalomas laukelyje „Data“ įrašoma reikšmė „2000-11“ ir paspaudžiamas mygtukas „Įrašyti“	Sistema pateikia pranešimą „Neteisingas laukelio Data reikšmės tipas! Turi būti data!“	Pavyko
Teisingai užpildomi visi laukeliai ir paspaudžiamas mygtukas „Įrašyti“	Duomenų pakeitimai išsaugomi ir sistema pateikia pranešimą „Duomenys išsaugoti“	

### **Integracinis testavimas**

Atlikus vienetų testavimą, ištestuoti vienetai apjungiami į bendrą sistemą ir toliau testuojama jų sąveika.

Integracinio testavimas vykdomos sekančia tvarka:

Administratorius veiksmai:

- Sukuriamas duomenų pildymo šablonas „pareigos“ su sekančia struktūra:

Lentelė Nr. 24 Šablono „pareigos“ struktūra

Komentaras	Darbuotojų pareigos				
Laukelio pavadinimas	Tipas	Pavyzdys	Privaloma	Pradinė reikšmė	Vienodos reikšmės
Pareigos	Pagal pavyzdį	Direktorius	Privaloma		Negali būti

- Sukuriamas duomenų pildymo šablonas „darbuotojai“ su sekančia struktūra:

Lentelė Nr. 25 Šablono „darbuotojai“ struktūra

Komentaras	Organizacijos darbuotojai				
Laukelio pavadinimas	Tipas	Pavyzdys	Privaloma	Pradinė reikšmė	Vienodos reikšmės
Vardas	Pagal pavyzdį	Spalvotas	Privaloma		Gali būti
Pavardė	Pagal pavyzdį	Pieštukas	Privaloma		Gali būti
Pareigos	Iš kitos lentelės	Pareigos.Pareigos	Privaloma		Gali būti
Priėmimo data	Data		Privaloma		Gali būti
Atleidimo data	Data		Neprivaloma		Gali būti

- sukuriamas vartotojas „sekretore“;
- vartotojui „sekretore“ suteikiama rolė „Visų duomenų skaitytojas“;
- vartotojui „sekretore“ į duomenų pildymo šabloną „Pareigos“ suteikiama teisė „Duomenų tvarkytojas“;
- vartotojui „sekretore“ į duomenų pildymo šabloną „Darbuotojai“ suteikiama teisė „Duomenų tvarkytojas“;

Vartotojo „sekretore“ veiksmai:

- Pagal duomenų pildymo šabloną „pareigos“ sukurti sekančius įrašus:  
Direktorius  
Sekretorė  
Konsultantas  
Vyr. konsultantas
- Pagal duomenų pildymo šabloną „darbuotojai“ sukurti sekančius įrašus:

Lentelė Nr. 26 Pagal šablona „darbuotojai“ vedami duomenys

Vardas	Pavardė	Pareigos	Priėmimo data
Spalvotas	Pieštukas	Direktorius	2000-11-11
Raudonas	Žymeklis	Sekretorė	2000-11-11
Mėlynas	Šratinukas	Konsultantas	2000-11-11
Lipnus	Lapelis	Vyr. konsultantas	2000-11-11

Administratoriaus veiksmai:

- sukuriamas vartotojas „direktorius“;
- vartotojui „direktorius“ suteikiamos rolės „Vartotojų tvarkytojas“, „Šablonų tvarkytojas“, „Visų duomenų tvarkytojas“, „Visų duomenų skaitytojas“;

Vartotojo „direktorius“ veiksmai:

- sukuriamas vartotojas „konsultantas“;
- vartotojui „konsultantas“ suteikiama rolė „Visų duomenų skaitytojas“;

Vartotojo „konsultantas“ veiksmai:

- peržiūri duomenis suvestus pagal duomenų pildymo šablonus „Darbuotojai“ ir „Pareigos“;

#### **Aukšto lygio testavimas**

Užbaigtos programinės priemonės testavimas, siekiant išsiaiškinti jos kokybę. Tam naudojami tokie testavimo metodai:

##### **- Atsistatymas**

Kol organizacijos darbuotojai vykdė aukščiau išvardintus veiksmus, kompiuteryje kuriame įdiegta informacinė sistema buvo atliekamas atsistatymo tikrinimas. Imituojant netikėtą kompiuterio išjungimą buvo nutrauktas Apache serverio bei MySQL serverio darbas. Darbui nutraukti buvo panaudota nemokama programa rkill.exe. Ji vykdoma komandinių eilučių pagalba (Command Prompt). Šios programos pagalba stabdomos programinės priemonės neatlieka jokių darbų, kurie atliekami įprastai išjungiant programines priemones. Iš naujo įjungus Apache serverį ir MySQL serverį duomenų praradimo nepastebėta.

##### **- Našumas**

Našumo patikrinimui keliuose kompiuteriuose iš serverio, kuriame įdiegta informacinė sistema, buvo užkrauti HTML puslapiai su niekada nesibaigiančiu JavaScript scenarijų vykdymu. Pačiame serveryje buvo užduotas didelio kiekio mažų failų archyvavimas bei užkrautas minėtas puslapis su begaliniu JavaScript scenarijų vykdymu.

Kituose kompiuteriuose vartotojai tęsė darbą su informacine sistema ir nepastebėjo jokių trukdžių. Kadangi informacinėje sistemoje transakcijos nėra sudėtingos ir ilgai vykdomos serverio apkrovimas informacinės sistemos darbui įtakos neturi.