

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

**Tomas Stasiūnas**

**INTERNETINĖS DUOMENŲ APDOROJIMO  
SAŠAJOS KŪRIMAS: METODOLOGIJA IR  
ĮRANKIS**

Magistro darbas

**Vadovė  
doc. L. Nemuraitė**

**KAUNAS, 2005**

## **SUMMARY**

With increase of a web application popularity there was a necessity of simplification of their creation. In this paper ways of creation electronic data processing and opportunities of their simplification are investigated. Models of web application and tools for their creation are considered in the paper. Being based on this analysis the model which more all would approach for creation of the fast interface of data processing is offered. It is based on the user interface architecture separated from logic of activity. The offered model is realized by system of programs PHPForm on a platform Eclipse, the comparative analysis of the created and existing products is executed.

## TURINYS

|   |    |
|---|----|
| ĮVADAS .....  | 7  |
| 1. INTERNETO PROGRAMŲ KŪRIMAS .....   | 9  |
| 1.1. Technologijų analizė .....   | 9  |
| 1.1.1. Interneto programų technologijų ir architektūrų apžvalga .....                   | 9  |
| 1.1.2. Įrankių, skirtų interneto programoms kurti, apžvalga .....                       | 10 |
| 1.2. Darbo tikslai .....  | 11 |
| 1.3. Interneto programų kūrimo proceso palengvinimas.....                               | 12 |
| 1.3.1. Optimalus architektūros modelis greitam duomenų apdorojimo sąsajos kūrimui ..... | 12 |
| 1.3.2. IDE, realizuojanti pasirinktą interneto programos modelį .....                   | 13 |
| 2. INTERNETO PROGRAMOS MODELIS IR JŲ REALIZUOJANTIS ĮRANKIS .....                       | 15 |
| 2.1. Sistemos vartotojų grupės .....  | 15 |
| 2.2. Projekto apribojimai .....   | 15 |
| 2.3. Funkciniai reikalavimai .....  | 16 |
| 2.3.1. Konteksto diagrama.....  | 16 |
| 2.3.2. Veiklos įvykių sąrašas .....   | 17 |
| 2.3.3. Panaudojimo atvejų vaizdas .....   | 17 |
| 2.3.4. Panaudojimo atvejų aprašymas .....   | 18 |
| 2.4. Nefunkciniai reikalavimai .....  | 20 |
| 2.5. Programų sistemos architektūra .....   | 20 |
| 2.5.1. Architektūros pateikimas.....  | 21 |
| 2.5.2. Architektūros tikslai ir apribojimai .....                                       | 21 |
| 2.5.3. Loginis vaizdas.....   | 21 |
| 2.5.4. Paketų detalizavimas .....   | 21 |
| 2.5.5. Procesų vaizdas .....  | 25 |
| 2.5.6. Išdėstymo vaizdas .....  | 29 |
| 2.5.7. Duomenų vaizdas .....  | 30 |
| 3. PHPFORM PROGRAMŲ SISTEMOS KOKYBĖS TYRIMAS .....                                      | 31 |
| 3.1. Vartotojų pastebėtos klaidos .....   | 31 |
| 3.2. Sistemos vertinimas pagal kokybės faktorius .....                                  | 31 |
| 3.2.1. Korektiškumas.....   | 31 |
| 3.2.2. Patikimumas .....  | 31 |

|  |    |
|--|----|
|  | 4  |
| 3.2.3. Efektyvumas.....  | 32 |
| 3.2.4. Panaudojimas .....  | 32 |
| 3.2.5. Testavimas.....   | 32 |
| 3.2.6. Lankstumas.....   | 32 |
| 3.2.7. Pernešamumas .....  | 32 |
| 3.2.8. Pakartotinas panaudojimas.....  | 32 |
| 3.2.9. Sąsajos galimybės .....   | 33 |
| 3.3. Sistemos kokybės vertinimas remiantis vartotojų pastabomis.....   | 33 |
| 4. EKSPERIMENTINĖ DALIS .....  | 34 |
| 4.1. Interneto programos pavyzdys .....  | 34 |
| 4.2. Interneto programos kūrimas PHPForm sistema .....   | 36 |
| 4.3. Eksperimento rezultatų tyrimas.....   | 38 |
| IŠVADOS .....  | 40 |
| LITERATŪRA .....   | 41 |
| TERMINŲ IR SANTRUMPŲ ŽODYNAS .....   | 43 |
| 5. 1 PRIEDAS. Vaizdinės priemonės internetinei duomenų apdorojimo sąsajai kurti. Informacinės technologijos'05 ..... | 44 |
| 5.1. Įvadas .....  | 44 |
| 5.2. Pasaulinio tinklo programų architektūrų ir technologijų analizė .....   | 44 |
| 5.2.1. Pasaulinio tinklo programų modelių ir technologijų apžvalga.....  | 44 |
| 5.2.2. Įrankių, skirtų pasaulinio tinklo programoms kurti, apžvalga .....  | 46 |
| 5.3. Optimalus architektūros modelis greitam duomenų apdorojimo sąsajos kūrimui .....                                | 47 |
| 5.3.1. Pasirinktas pasaulinio tinklo programos modelis.....  | 47 |
| 5.3.2. IDE realizuojanti pasirinktą pasaulinio tinklo programos modelį .....   | 49 |
| 5.4. Įrankio realizacija Eclipse platformoje .....   | 49 |
| 5.5. Išvados.....  | 50 |
| 6. 2 PRIEDAS. PHPForm sistemos vartotojo dokumentacija .....   | 51 |
| 6.1. Sistemos funkcinis aprašymas.....   | 51 |
| 6.1.1. Sistemos paskirtis.....   | 51 |
| 6.1.2. PHPForm programų sistemos galimybės .....   | 51 |
| 6.2. Vartotojo atmintinė .....   | 51 |
| 6.3. Detali sistemos atmintinė .....   | 52 |
| 6.3.1. PHPForm perspektyva.....  | 52 |

|  |    |
|--|----|
|  | 5  |
| 6.3.2. Naujų PHPForm projektų, formų ir vartotojo komponentų kūrimas ..... | 54 |
| 6.3.3. PHPForm internetinių formų vartotojo sąsaja.....                    | 57 |
| 6.3.4. PHPForm veiklos logikos aprašymas.....                              | 61 |
| 6.3.5. DBExplorer galimybių aprašymas .....                                | 62 |
| 6.4. Sistemos instaliavimo dokumentas .....                                | 64 |
| 6.4.1. Reikalavimai.....   | 64 |
| 6.4.2. PHPForm programų sistemos instaliacija .....                        | 64 |
| 6.4.3. Instaliacijos patikrinimas.....                                     | 66 |
| 6.4.4. PHPForm programų sistemos pašalinimas.....                          | 67 |
| 6.5. Sistemos administratoriaus vadovas.....                               | 67 |
| 6.5.1. Programų sistemos sąveika su HTTP serveriu .....                    | 67 |
| 6.5.2. Programų sistemos sąveika su DBVS serveriu .....                    | 67 |

## Lentelių sąrašas

|  |    |
|--|----|
| 1.1 lentelė. Technologijų, skirtų interneto programoms kurti, palyginimas penkiabalėje sistemoje. .... | 10 |
| 1.2 lentelė. Įrankių, skirtų interneto programoms kurti, palyginimas penkiabalėje sistemoje. ....      | 11 |
| 2.1 lentelė. Veiklos įvykių sąrašas .....  | 17 |
| 3.1 lentelė. Vartotojų pastebėtos klaidos.....   | 31 |
| 3.1 lentelė. PHPForm sistemos kokybės vertinimas penkiabalėje sistemoje.....                           | 33 |
| 4.1 lentelė. PHPForm sistemos vertinimas penkiabalėje sistemoje .....                                  | 39 |

## Paveikslėlių sąrašas

|   |    |
|---|----|
| 1.1 pav. Interneto programos sukūrimo schema .....                              | 13 |
| 2.1 pav. Konteksto diagrama .....   | 16 |
| 2.2 pav. Panaudojimo atvejų diagrama .....                                      | 18 |
| 2.3 pav. Sistemos paketų diagrama .....   | 21 |
| 2.4 pav. Vartotojo sąsajos klasių diagrama .....                                | 22 |
| 2.5 pav. Veiklos logikos klasių diagrama.....                                   | 23 |
| 2.6 pav. Duomenų saugojimo klasių diagrama.....                                 | 23 |
| 2.7 pav. PHP klasių diagrama .....  | 24 |
| 2.8 pav. Projekto sukūrimo sekos diagrama.....                                  | 25 |
| 2.9 pav. Interneto programos surinkimo sekos diagrama.....                      | 25 |
| 2.10 pav. DB jungties kūrimo sekos diagrama .....                               | 26 |
| 2.11 pav. SQL konsolės sekos diagrama.....                                      | 26 |
| 2.12 pav. Komponento įkėlimo sekos diagrama.....                                | 27 |
| 2.13 pav. Klasės darbui su DB generavimo sekos diagrama.....                    | 27 |
| 2.14 pav. Klasės darbui su DB generavimo bendradarbiavimo diagrama.....         | 28 |
| 2.15 pav. Objekto savybės reikšmės keitimo sekos diagrama.....                  | 28 |
| 2.16 pav. Projekto būsenos diagrama.....  | 29 |
| 2.17 pav. Sistemos išdėstymo diagrama.....                                      | 29 |
| 2.18 pav. Duomenų struktūros.....   | 30 |
| 4.1 pav. Vartotojo formos vaizdas .....   | 34 |
| 4.2 pav. Vartotojo formos sąsajos aprašymas.....                                | 35 |
| 4.3 pav. Automatiškai sugeneruota vartotojo formos logiką aprašanti klasė ..... | 35 |
| 4.4 pav. Naujo projekto sukūrimo vedlys.....                                    | 36 |
| 4.5 pav. PHPForm pagrindinis langas.....  | 37 |
| 4.6 pav. Darbas su duomenų baze.....  | 37 |
| 4.7 pav. Sukurta internetinė forma.....   | 38 |

## IVADAS

Interneto programos (angl. *Web application*) tampa vis populiareesnės. Tai susiję su tuo, kad kai kuriais aspektais jos teikia gerokai daugiau galimybių negu tradicinės programos (angl. *desktop application*). Vienas iš pagrindinių interneto programos privalumų – prieinamumas, ja gali naudotis daug vartotojų. Be to, norint naudotis bet kokia interneto programa, nereikia diegti papildomos programinės įrangos, pakanka paprastos interneto naršyklės.

Interneto programų kūrimas yra sudėtingas, daug žinių ir patirties reikalaujantis procesas. Norint palengvinti programuotojo darbą, galima naudoti įvairias standartines bibliotekas ir įrankius (IDE), pritaikytus interneto programoms kurti. Tam skirta nemažai universalių komercinių produktų, tačiau jie turi nemažai trūkumų: brangūs, sudėtingi, reikalaujantys daug žinių. Šiame darbe apžvelgiami interneto programų modeliai ir instrumentinės priemonės joms kurti. Remiantis šia analize, pasiūlytas modelis, kuris geriausiai tinka greitam duomenų apdorojimo sąsajos kūrimui. Taip pat aprašomos sukurtos instrumentinės priemonės, realizuojančios pasiūlytą modelį. Sukurta sistema buvo iširta pagal kokybės faktorius, atliktas jos eksperimentinis įvertinimas.

Šio darbo tikslas yra išanalizuoti interneto programų kūrimo technologijas ir sukurti priemones, leidžiančias lengvai ir greitai kurti interneto programas, kuriomis apdorojami duomenys. Sukurta sistema bus sudaryta iš dviejų dalių:

- interneto programos modelio;
- instrumentinių priemonių, skirtų interneto programoms kurti.

Kuriama programų sistema pavadinta PHPForm.

Šiame darbe apžvelgiamos technologijos, pritaikytos kurti interneto programas. Analizuojamos ASP.NET, JSP ir PHP technologijos ir interneto programų architektūros: vieno sluoksnio, vartotojo sąsajos, atskirtos nuo veiklos logikos, ir MVC (*Model–View–Controller*), išskiriami jų privalumai ir trūkumai. Interneto programos modelis realizuojamas naudojant PHP technologiją ir vartotojo sąsajos, atskirtos nuo veiklos logikos, architektūrą. PHP technologija pasirinkta todėl, kad ji plačiai naudojama ir yra nemokama. Vartotojo sąsajos, atskirtos nuo veiklos logikos, architektūra labiausiai tinka interneto programoms, skirtoms duomenų apdorojimui kurti. Taip pat buvo remiamasi Terence Parr straipsnyje „Enforcing strict model-view separation in template engines“ [1] pateiktais argumentais.

Šiame darbe atliekama instrumentinių priemonių, skirtų interneto programoms kurti, analizė. Apžvelgiamos integruotos Visual Studio .NET, Macromedia DreamWeaver ir Eclipse interneto programų kūrimo aplinkos, jų privalumai bei trūkumai. Atlikus analizę, instrumentinėms priemonėms kurti

pasirinkta Eclipse platforma, nes ji skirta integruotoms vystymo aplinkoms (IDE) kurti. Eclipse platforma universali ir nepriklausoma nuo operacinės sistemos.

Antrame skyriuje pateikiami kuriamai programų sistemai keliami reikalavimai ir svarbiausi architektūros aspektai. Surinkti reikalavimai buvo suskirstyti į funkcinius, nefuncinius ir projektą apribojančius. Sistemos architektūra sukurta pagal RUP principus [18]. Ji išskaidyta į tris sluoksnius: vartotojo sąsajos, veiklos logikos ir duomenų. Architektūra aprašoma panaudojimo atveju, komponentų, klasių, procesų, sekų, būsenų ir išdėstymo diagramomis.

Sukūrus ir įdiegus programinę įrangą, atliktas jos panaudojimo tyrimas. Vartotojų pastebėtos klaidos, trūkumai ir pasiūlymai buvo registruojami. Į išsakytas pastabas stengtasi atsižvelgti toliau vystant PHPForm programų sistemą. Sukurta sistema įvertinta pagal kokybės kriterijus. Supažindinus vartotojus su sistema, atliktas eksperimentinis tyrimas, kuriuo buvo siekiama nustatyti, kaip sukurtas produktas palengvina interneto programų kūrimo procesą.

Šio magistro darbo tema skaitytas pranešimas tarpuniversitetinėje magistrantų ir doktorantų konferencijoje Informacinės technologijos'05 (straipsnis pateiktas pirmame priede).



# 1. INTERNETO PROGRAMŲ KŪRIMAS

## 1.1. Technologijų analizė

Šiame skyriuje apžvelgiamos ir analizuojamos interneto programų technologijos, architektūros ir įrankiai joms kurti.

### 1.1.1. Interneto programų technologijų ir architektūrų apžvalga

Interneto programoms kurti naudojamos ASP, ASP.NET, JSP, PHP, *Cold Fusion* technologijos. Viena populiariausių nemokamų priemonių, naudojamų kurti interneto programas, yra PHP (PHP: *Hypertext Preprocessor*). PHP kalba parašyti scenarijai yra interpretuojami tinklo serveryje. PHP palaiko objektinio programavimo paradigmą, yra panaši į C, C++, Java kalbas, lengvai įsisavinama, turi bibliotekas, funkcijų, skirtų dirbti su failais, XML, duomenų bazėmis ir kt. [2, 3].

Pats paprasčiausias interneto programų architektūros modelis yra vieno sluoksnio, kai vartotojo sąsaja ir veiklos logika aprašoma viename faile. Šis architektūros modelis yra greičiausiai realizuojamas, tačiau tinka tik mažoms programoms. Jis turi daug trūkumų, kurie išryškėja didėjant programos sudėtingumui. Sudėtingoje interneto programoje atlikti kokius nors pakeitimus labai sunku, nes ji yra nelengvai skaitoma, dažnai nedideli pakeitimai vienoje vietoje gali sugadinti visą programą. Norint atlikti vieną pakeitimą, gali prireikti perrašyti programą keliose vietose [1, 3]. Šis modelis gali būti realizuojamas PHP, ASP arba JSP technologija.

ASP.NET – „Microsoft“ sukurta technologija, skirta interneto programoms kurti. Tai nauja ASP versija, kuri skiriasi nuo savo pirmtakės, tačiau visiškai su ja suderinama. Ji ne tik suteikia galimybę kurti dinamines tinklalapius, kaip jos pirmtakė, bet ir aprūpina bendru interneto programų kūrimo modeliu, kuris suteikia galimybę kurti interneto programas didelėms įmonėms. Vartotojo forma sudaryta iš dviejų dalių: išvaizdos ir elgsenos. Išvaizda aprašoma .aspx faile naudojant HTML elementus ir specialiomis ASP.NET bibliotekoje arba vartotojo aprašytomis žymėmis. Puslapio formos elgsena aprašoma atskirame faile bet kokia .NET palaikoma kalba [4, 5].

„Java Servlet“ išplečia interneto serverio galimybes ir leidžia prieiti prie egzistuojančių verslo sistemų. Tai kaip įsiuvas, kuris veikia serverio pusėje, tačiau neturi vartotojo sąsajos. Informaciją į išorę galima perduoti *System.print* funkcija [6, 7]. Ši technologija skirta veiksmams serverio pusėje atlikti, o ne vartotojo sąsajai kurti. Tam naudojama JSP (angl. *JavaServer Pages*) technologija, skirta vykdyti į HTML

puslapius įterptą Java kodą. Vykdam JSP serveryje, iš jo sugeneruojamas „Java Servlet“. HTML kodas įkeliamas į *System.print* funkciją [8].

Kitas interneto programų architektūros modelis yra MVC (modelis–vaizdas–valdiklis). Modelis yra programos veiklos logika, vaizdas – tai duomenų pateikimo vartotojui priemonė (HTML puslapis), o valdiklis – programos dalis, nustatanti, kokius veiksmus atlikti ir kokį vaizdą pateikti vartotojui. Šis modelis pritaikytas kurti dideles interneto programas. Remdamasis juo, vartotojas sukuria struktūrizuotą, lengvai skaitomą interneto programos kodą [9]. Egzistuoja kelios bibliotekos, realizuojančios šį architektūros modelį. Viena iš jų yra nemokama *Struts* biblioteka, realizuota naudojant JSP. Pagrindinis *Struts* realizacijos trūkumas – viena interneto programa turi tik vieną valdiklį. Be to, reikia nemažai laiko norint perprasti, kaip sukurti paprasčiausią interneto programą [10].

### 1.1 lentelė. Technologijų, skirtų interneto programoms kurti, palyginimas penkiabalėje sistemoje

| Kriterijus/Vertinimas   | ASP.NET | JSP Struts | PHP (vieno sluoksnio) |
|-------------------------|---------|------------|-----------------------|
| Ar lengvai panaudojamas | 4       | 2          | 5                     |
| Programos plečiamumas   | 4       | 5          | 3                     |
| Kaina                   | 3       | 5          | 5                     |

#### 1.1.2. Įrankių, skirtų interneto programoms kurti, apžvalga

Gana populiarius interneto svetainių kūrimo įrankis yra „Macromedia DreamWeaver“. Juo galima vizualiai kurti sudėtingą vartotojo sąsają, taip pat jis turi patogų kodo redaktorių. Tai universalus įrankis, kuriuo galima kurti įvairiausių tipų interneto programas, naudojančias ASP.NET, PHP, Cold Fusion, JSP technologijas. Jo trūkumai:

- brangus – viena kopija kainuoja 479 JAV dolerius;
- apkrauta vartotojo sąsaja – turi daug funkcijų, kurių retai reikia vartotojui;
- naudojant vizualų redaktorių, sugadinamas HTML kodo stilius, HTML žymės naudojamos neteisingai arba pasikartoja [11].

Microsoft Visual Studio .NET produktas skirtas kurti interneto programas. Jis paremtas ASP.NET technologija. Kuriant programas, naudojami RAD (angl. *rapid application development*) principai. Tai universalus, daug galimybių turintis įrankis. Jo trūkumai:

- brangus – viena kopija kainuoja 1799 JAV dolerius;
- veikia tik „Windows 2000“, XP operacinėje sistemoje;

- sukurtos interneto programos veikia tik „Microsoft IIS“ serveryje [12, 13].

Eclipse yra nemokama platforma, pritaikyta kurti integruotas vystymo aplinkas (IDE). Jos gali būti naudojamos kurti taikomąsias ir interneto programas, įvairių tipų projektus, pavyzdžiui, Java, JSP, PHP. Pagrindinė Eclipse savybė yra išplečiamumas, tam naudojama *plug-in* technologija. Eclipse *plug-in* – mažiausias Eclipse platformos funkcijos vienetas. Eclipse turi specialų įrankių rinkinį *Plug-in Development Environment* (PDE), kuriuo galima padidinti Eclipse platformos funkcionalumą. Eclipse sukurta naudojant Java technologiją, todėl veikia daugelyje operacinių sistemų („Windows“, „Linux“ bei Mac OS) [14, 15]. Egzistuoja keli įrankiai, skirti interneto programoms kurti Eclipse platformoje, tačiau jie yra komerciniai, ribotų galimybių ir nelabai paplitę.

### 1.2 lentelė. Įrankių, skirtų interneto programoms kurti, palyginimas penkiabalėje sistemoje

| Kriterijus\Vertinimas        | Visual Studio .NET | Macromedia DreamWeaver | Eclipse |
|------------------------------|--------------------|------------------------|---------|
| Ar lengvai panaudojamas      | 4                  | 4                      | 4       |
| Vartotojo sąsajos apkrovimas | 4                  | 3                      | 5       |
| Veikimas keliose platformose | 2                  | 4                      | 5       |
| Naudojami resursai           | 3                  | 4                      | 4       |
| Kaina                        | 2                  | 3                      | 5       |

## 1.2. Darbo tikslai

Šio darbo tikslas yra sukurti priemones, kuriomis galima lengviau ir greičiau kurti internetinę duomenų apdorojimo sąsają. Produktas turi būti sudarytas iš dviejų dalių:

- interneto programos modelio;
- instrumentinių priemonių, skirtų interneto programoms kurti.

Interneto programos modelis turi būti orientuotas į paprastą, lengvai realizuojamą vartotojo sąsajos aprašymą, skirtą duomenims apdoroti. Kuriant internetines programas ir internetinę duomenų apdorojimo sąsają, instrumentinės priemonės privalo kiek galima maksimaliau išnaudoti vizualizavimo ir automatizavimo galimybes. Pagrindinis jų uždavinys:

- vizualus vartotojo formų kūrimas;
- vartotojo veiksmų automatizavimas.

### 1.3. Interneto programų kūrimo proceso palengvinimas

Interneto programų kūrimo procesas palengvinamas tokiais būdais:

- Naudojant standartines bibliotekas. Kuriant interneto programas, dažnai susiduriama su uždaviniais, pavyzdžiui, formų validavimo, būsenos išsaugojimo, vartotojų identifikavimo, kuriuos, kuriant naujas sistemas, tenka spręsti iš naujo. Norint išvengti pakartotino kodo rašymo, kuriamos standartinės bibliotekos, skirtos interneto programoms kurti.
- Naudojant integruotas vystymo aplinkas (IDE), programų sistemos kūrimo procesą galima pagreitinti naudojant vaizdinius įrankius, veikiančius RAD principu. Naudojant vedlius, automatinį kodo generavimą ir vaizdines programų kūrimo priemones, interneto programų kūrimo procesas tampa greitesnis ir paprastesnis.

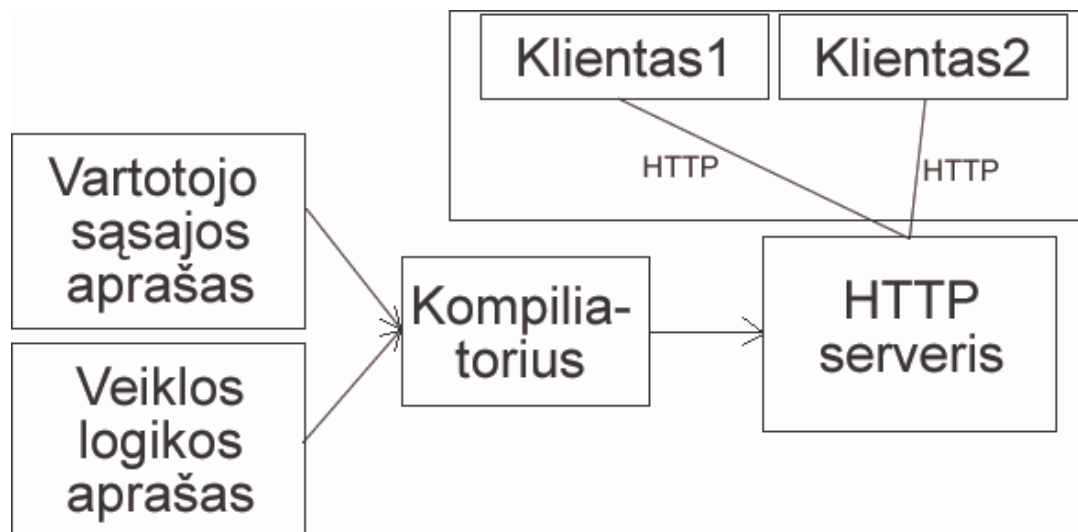
#### 1.3.1. Optimalus architektūros modelis greitam duomenų apdorojimo sąsajos kūrimui

Interneto programos modelis yra orientuotas į greitai kuriamą duomenų apdorojimo sąsają. Remiantis atlikta interneto programų technologijų analize, pasirinktas vartotojo sąsajos, atskirtos nuo veiklos logikos, modelis. Vieno sluoksnio architektūra netinka, nes yra sunkiai plečiama, o MVC modelis per daug sudėtingas šiam uždaviniui. Taip pat šiam pasirinkimui turėjo įtakos Terence Parr straipsnyje „Enforcing strict model-view separation in template engines“ [1] pateikti argumentai. Todėl pasirinkta architektūra modelis–vaizdas (angl. *model–view*). Šiame architektūros modelyje daugiau dėmesio skiriama programos veiklos logikai, mažiau – vartotojo sąsajos dizainui. Interneto programos modelis realizuojamas PHP technologija.

Kuriant vartotojo sąsają, programuotojui neturi rūpėti tokios problemos, kaip formos būsenos išsaugojimas ar atlikto veiksmo apdorojimas (angl. *event handling*). Šių uždavinių sprendimu turi pasirūpinti vaizdinių komponentų biblioteka. Ji realizuoja vaizdinius komponentus, skirtus vartotojo sąsajai kurti. Jų paskirtis – pagreitinti programos kūrimo procesą. Programuotojui nebereikia pakartotinai rašyti kodo. Vaizdinis komponentas susideda iš dviejų dalių: išvaizdos ir veiklos logikos. Vaizdiniai komponentai skirstomi į standartinius ir vartotojo. Standartiniai komponentai yra žymenos (angl. *label*), teksto įvedimo laukeliai (angl. *textbox*), mygtukai (angl. *button*), sąrašai (angl. *listbox*). Vartotojo komponentai – tai HTML kalboje atitiktoms neturintys vartotojo sąsajos elementai. Pavyzdžiui, duomenų tinklėlis (angl. *DataGrid*), kurio nėra tarp standartinių HTML elementų, tačiau jis realizuotas kaip vartotojo komponentas.

Komponentų biblioteką pasirinkta realizuoti PHP technologija, nes ji yra nemokama, šiuo metu labai populiari, plačiai naudojama Lietuvoje ir pasaulyje. Vartotojo sąsaja aprašoma HTML kalba, nes vartotojams, mokantiems ją, nereikia papildomai nieko mokytis. Vartotojo sąsajos aprašas papildytas žymomis, skirtomis vaizdiniais vartotojo komponentams įterpti. Vartotojo sąsajos aprašas turi būti gerai suformuotas (angl. *well formed*) XML dokumentas. Buvo pasirinkta, kad iš vartotojo sąsajos aprašo būtų galima sugeneruoti jos veiklos logiką aprašančią PHP klasę. Į automatiškai sugeneruotą klasę programuotojui reikės įrašyti tik formos veiklos logiką. Formos inicializavimas bus sugeneruotas automatiškai.

Interneto programa sudaryta iš tarpusavyje susijusių internetinių formų. Internetinė forma sudaryta iš vartotojo sąsajos aprašo ir jos veiklos logiką aprašančios PHP klasės. 1.1 paveiksle pavaizduota interneto programos sukūrimo schema.



1.1 pav. Interneto programos sukūrimo schema

Iš vartotojo sąsajos ir veiklos logikos aprašymų kompiliatorius sugeneruos vykdomą PHP failą, kuris saugomas HTTP serveryje. Dėl šio architektūrinio sprendimo interneto programos vykdomumas (angl. *performance*) bus toks pat, kaip ir vieno sluoksnio PHP interneto programos. Klientai internetinę formą galės pasiekti internetu, naudodamiesi HTTP klientu.

### 1.3.2. IDE, realizuojanti pasirinktą interneto programos modelį

Kuriant programų sistemas, svarbu įvertinti instrumentines priemones. Pavyzdžiui, kuriant Ada programavimo kalbą, projekto tikslas buvo sukurti aukšto lygio programavimo kalbą. Dalis projekto skirta realizuoti jai pritaikytą programavimo aplinką (angl. *Ada programming environment*), tačiau šiai projekto daliai skirta ne tiek daug dėmesio, kiek pačios programavimo kalbos kūrimui. Todėl programavimo kalba

Ada neturi tokios programavimo sąsajos, kuri galėtų prilygti kitų sistemų aplinkoms, pavyzdžiui, *Visual Studio*. Vartotojui sunkiau pasinaudoti nauja technologija, nes, norint ją įvaldyti, reikia skirti daugiau pastangų ir laiko [16, 17].

Realizuojant šią sistemą, buvo nuspręsta sukurti instrumentines priemones, įgyvendinančias pasirinkto interneto programų kūrimo modelį. Jo paskirtis – palengvinti vartotojui įsisavinti siūlomą įrankį ir automatizuoti kūrimo procesą. Reikalavimai įrankiui:

- vizualus dizaineris vartotojo formoms kurti;
- automatinis formos veiklos logiką aprašančios klasės generavimas iš formos aprašo;
- PHP ir HTML kodo redaktorius;
- vedlių naudojimas atliekant sudėtingesnius veiksmus.

Įrankio kūrimui pasirinkta Eclipse platforma, nes ji turi visas pagrindines savybes, reikalingas IDE: projektų kūrimas, failų redagavimas, integruota failų versijų kontrolė. Kuriant sistemą, nebereikia realizuoti standartinių IDE funkcijų, todėl visas dėmesys skiriamas sukurtam interneto programos modeliui realizuoti. Naudojant Eclipse, orientuojamasi į platų vartotojų ratą, nes ji veikia daugelyje operacinių sistemų.

## 2. INTERNETO PROGRAMOS MODELIS IR JI REALIZUOJANTIS ĮRANKIS

Šiame skyriuje aprašomi kuriamai programinei įrangai keliami reikalavimai ir programų sistemos architektūra. Išskiriami sistemos vartotojai, projekto apribojimai, funkciniai ir nefunkciniai reikalavimai. Programų sistemos architektūra aprašoma UML diagramomis.

### 2.1. Sistemos vartotojų grupės

Išskiriamos dvi programų sistemos vartotojų grupės: programuotojas ir dizaineris.

#### 1. Programuotojas

| Vartotojo kategorija                       | Programuotojas   |
|--|--|
| Vartotojo sprendžiami uždaviniai           | Programos veiklos logikos programavimas                        |
| Patirtis dalykinėje srityje                | Naujokas arba pažengęs   |
| Patirtis informacinių technologijų srityje | Programavimo pagrindų ir darbo su duomenų bazėmis patirtis     |
| Papildomos vartotojo charakteristikos      | Turi būti susipažinęs su interneto programų veikimo principais |
| Vartotojo prioritetas                      | Labai svarbus  |

#### 2. Dizaineris

| Vartotojo kategorija                       | Dizaineris   |
|--|--|
| Vartotojo sprendžiami uždaviniai           | Internetinės vartotojo sąsajos išvaizdos kūrimas           |
| Patirtis dalykinėje srityje                | Naujokas arba pažengęs                                     |
| Patirtis informacinių technologijų srityje | Darbo grafiniu redaktoriumi pradmenys ir geros HTML žinios |
| Papildomos vartotojo charakteristikos      |  |
| Vartotojo prioritetas                      | Svarbus  |

### 2.2. Projekto apribojimai

Šiame skyriuje aprašomi programų sistemai keliami apribojimai.

#### Sprendimų apribojimai

Pagrindinis sistemos apribojimas – ji turi būti įgyvendinta nemokamomis priemonėmis, t.y., jos neturi sudaryti komerciniai komponentai.

#### Diegimo aplinka

Sistema neturi priklausyti nuo konkrečios operacinės sistemos. Turi būti realizuota galimybe dirbti Windows, Linux ir UNIX šeimos operacinėje sistemoje ir naudotis pagrindinėmis duomenų bazių valdymo sistemomis (DBVS), pavyzdžiui, MS SQL. Sukurta internetinė sąsaja turi būti suderinta su populiariausiomis interneto naršyklėmis: Internet Explorer ir Netscape.

### Numatoma darbo vietos aplinka

Darbo vietos aplinkai nekeliama jokie specifiniai reikalavimai. Tai turi būti patogi vartotojui darbo vieta, atitinkanti ergonominius reikalavimus.

### Svarbūs faktai ir prielaidos

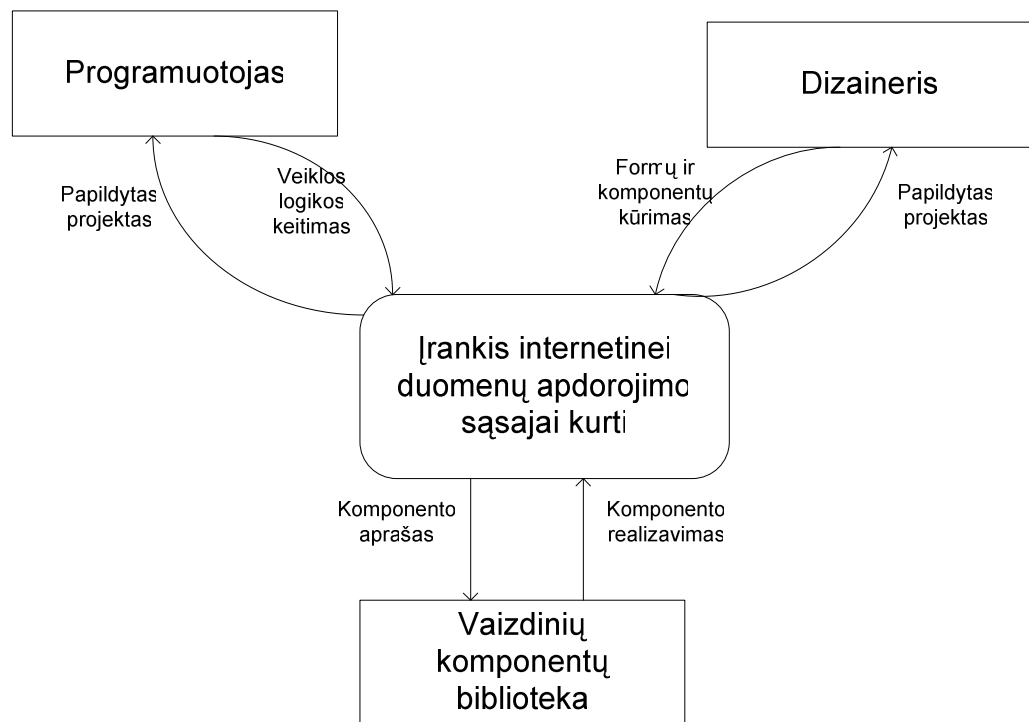
Kuriant sistemą, svarbu efektyviai išnaudoti kompiuterio resursus. Interneto programos naudoja daug serverio resursų net labai nedideliems pakeitimams padaryti, pavyzdžiui, norint gauti iš DB serverio informaciją apie vartotojo pasirinktą objektą, tenka iš naujo generuoti visą puslapį, nors pakeitimai gali būti nedideli. Vykdamt projektą, svarbu atkreipti dėmesį į šį faktą ir siekti sumažinti serverio apkrovą.

## 2.3. Funkciniai reikalavimai

Šiame skyriuje aprašomi programų sistemai keliami funkciniai reikalavimai.

### 2.3.1. Konteksto diagrama

2.1 paveiksle pavaizduota programų sistemos konteksto diagrama.



2.1 pav. Konteksto diagrama



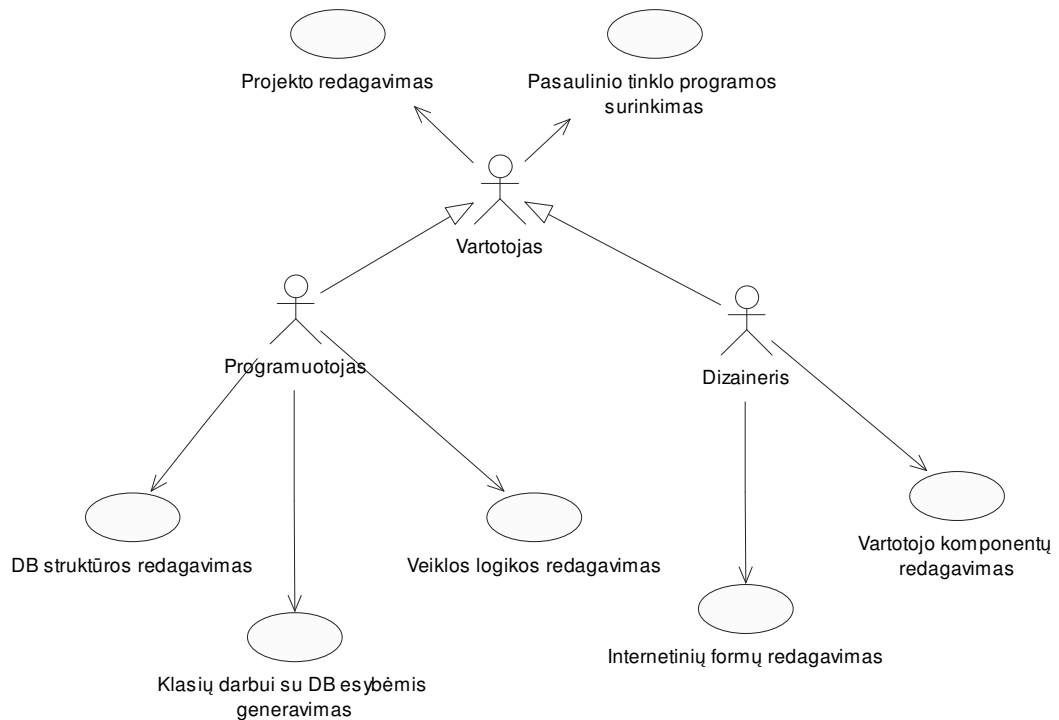
### 2.3.2. Veiklos įvykių sąrašas

2.1 lentelė. Veiklos įvykių sąrašas

| <b>Eil. nr.</b> | <b>Įvykio pavadinimas</b>                | <b>Įeinantys/išeinantys informacijos srautai</b>                       |
|-----------------|--|--|
| 1               | Projekto redagavimas                     | Sukurta projekto struktūra (OUT)                                       |
| 2               | Interneto programos surinkimas           | Paruošta vykdyti interneto programa (OUT)                              |
| 3               | DB redagavimas                           | Duomenys apie DB struktūrą ir duomenis (IN)                            |
| 4               | Klasių darbui su DB esybėmis generavimas | DB struktūra (IN)/Klasė, skirta darbui su DB lentelės įrašais (OUT)    |
| 5               | Veiklos logikos redagavimas              | Programos logika (IN)  |
| 6               | Internetinių formų redagavimas           | Duomenys apie internetinę formą (IN)/Internetinė formos išvaizda (OUT) |
| 7               | Vartotojo komponentų redagavimas         | Duomenys apie vartotojo komponentą (IN)/Vartotojo komponentas (OUT)    |

### 2.3.3. Panaudojimo atvejų vaizdas

2.2 paveiksle pavaizduota kuriamos programų sistemos panaudojimo atvejų diagrama.



2.2 pav. Panaudojimo atvejų diagrama

### 2.3.4. Panaudojimo atvejų aprašymas

#### 1. Projekto redagavimas

|                          |   |
|--------------------------|---|
| <b>Tikslas</b>           | Sukurti interneto programos projekto aprašymą                   |
| <b>Aktoriai</b>          | Programinės įrangos vartotojas                                  |
| <b>Prieš sąlygos</b>     | nėra  |
| <b>Sužadinimo sąlyga</b> | Vartotojas pasirinko meniu punktą <i>sukurti naują projektą</i> |
| <b>Po sąlyga</b>         | Sukurtas projekto aprašymas                                     |

#### 2. Interneto programos surinkimas

|                          |  |
|--------------------------|--|
| <b>Tikslas</b>           | Iš aprašymų surinkti interneto programą  |
| <b>Aktoriai</b>          | Programinės įrangos vartotojas   |
| <b>Prieš sąlygos</b>     | Sukurtas projektas   |
| <b>Sužadinimo sąlyga</b> | Vartotojas pasirinko meniu punktą <i>surinkti</i> , išsaugojo veiklos logiką arba vartotojo sąsają aprašantį failą |
| <b>Po sąlyga</b>         | Iš projekto aprašymo surinkti interneto programos failai   |

#### 3. DB struktūros redagavimas

|                          |  |
|--------------------------|--|
| <b>Tikslas</b>           | Redaguoti duomenų bazės struktūrą ir dirbti su DB duomenimis |
| <b>Aktoriai</b>          | Programuotojas   |
| <b>Prieš sąlygos</b>     | Sukurtas projektas ir jungtis prie DBVS                      |
| <b>Sužadinimo sąlyga</b> | Pasirinktas meniu punktas <i>redaguoti duomenų bazę</i>      |
| <b>Po sąlyga</b>         | Pakeista DB struktūra ir duomenys                            |

#### 4. Klasių darbui su DB esybėmis generavimas

|                          |   |
|--------------------------|---|
| <b>Tikslas</b>           | Sugeneruoti klasę, skirtą darbui su DB lentelės įrašais                           |
| <b>Aktoriai</b>          | Programuotojas  |
| <b>Prieš sąlygos</b>     | Egzistuoja DB lentelės aprašymas  |
| <b>Sužadinimo sąlyga</b> | Programuotojas pasirinko iš DB lentelės kontekstinio meniu <i>generuoti klasę</i> |
| <b>Po sąlyga</b>         | Sukurta klasė, skirta darbui su DB lentelės įrašais                               |

#### 5. Veiklos logikos redagavimas

|                          |   |
|--------------------------|---|
| <b>Tikslas</b>           | Sukurti ir pakeisti veiklos logiką                          |
| <b>Aktoriai</b>          | Programuotojas  |
| <b>Prieš sąlygos</b>     | Sukurta internetinė forma arba vartotojo komponentas        |
| <b>Sužadinimo sąlyga</b> | Programuotojas pasirinko meniu punktą <i>redaguoti kodą</i> |
| <b>Po sąlyga</b>         | Pakeistas veiklos logiką aprašantis kodas                   |

#### 6. Internetinių formų redagavimas

|                          |  |
|--------------------------|--|
| <b>Tikslas</b>           | Internetinių formų redagavimas   |
| <b>Aktoriai</b>          | Dizaineris   |
| <b>Prieš sąlygos</b>     | Sukurtas projektas   |
| <b>Sužadinimo sąlyga</b> | Vartotojo sąsajos redagavimo aplinkoje XML kalba nurodoma, kaip turi būti išdėstyti standartiniai arba vartotojo komponentai ir jų savybės |
| <b>Po sąlyga</b>         | Pakeistas internetinės formos aprašas  |

#### 7. Vartotojo komponentų redagavimas

|                      |                                  |
|----------------------|----------------------------------|
| <b>Tikslas</b>       | Vartotojo komponentų redagavimas |
| <b>Aktoriai</b>      | Dizaineris                       |
| <b>Prieš sąlygos</b> | Sukurtas projektas               |

|                          |  |
|--------------------------|--|
| <b>Sužadinimo sąlyga</b> | Pasirinktas meniu punktas <i>redaguoti komponento vartotojo sąsają</i> |
| <b>Po sąlyga</b>         | Pakeistas komponento aprašas   |

## 2.4. Nefunkciniai reikalavimai

Šiame skyriuje aprašomi programų sistemai keliami nefunkciniai reikalavimai.

### Reikalavimai sistemos išvaizdai

Bendri reikalavimai vartotojo sąsajai:

- nesudėtingas naudojimas;
- galimybė naudotis vedliu atliekant sudėtingesnius veiksmus;
- atitinka kitus vartotojo naudojamus produktus (pavyzdžiui, MS Visual Studio, Borland Builder).

### Reikalavimai naudojimui

Panaudojimo paprastumas, kuris gali būti vertinamas konkrečiais kriterijais:

- paprastas naudoti programuotojams ir dizaineriams, dirbantiems su panašiomis sistemomis;
- veiklos našumo padidėjimas dėl sistemos diegimo.

### Reikalavimai vykdymo charakteristikoms

Kuriama sistema yra sudėtinga, todėl reikia užtikrinti, kad jos vykdymui užtektų kompiuterio resursų. Taip pat sukurta programų sistema turi būti patikima, t.y., vartotojas, dirbdamas ja, neturi patirti nepatogumų.

### Teisiniai reikalavimai

Sukurtą produktą galima laisvai platinti, nes jis bus sukurtas naudojant nemokamas priemones. Jis yra nepriklausomas nuo komercinių produktų.

### Vartotojo dokumentacija ir mokymas

Vartotojo dokumentacija turi būti sudaryta iš tokių dalių:

- komponentų bibliotekos aprašymas;
- sukurto įrankio dokumentacija ir integruotas žinynas;
- paaiškinimai, kaip pažingsniui atlikti sudėtingesnes operacijas.

## 2.5. Programų sistemos architektūra

Šiame skyriuje aprašoma programų sistemos loginė architektūra, fizinis išdėstymas ir naudojamos duomenų struktūros. Loginė architektūra sudaryta remiantis RUP metodika: suskirstyta į vartotojo sąsajos, veiklos logikos ir duomenų sluoksnius.

### 2.5.1. Architektūros pateikimas

Sistemos architektūros pateikimui naudojami keturių rūšių vaizdai:

- panaudojimo atvejų;
- loginis;
- procesų;
- išdėstymo.

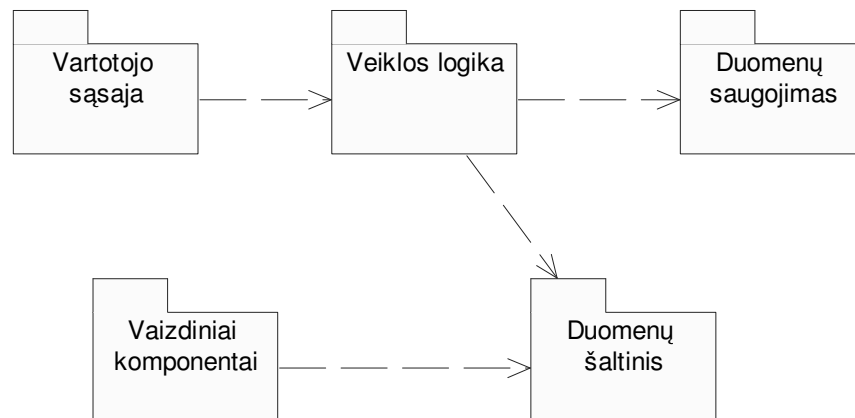
Panaudojimo atvejų vaizde pateikiami sistemos vartotojai ir galimi sistemos naudojimo atvejai bei trumpas jų apibūdinimas. Loginis vaizdas parodo, kaip į paketus yra suskirstytos sistemos klasės ir kaip jos susietos tarpusavyje. Procesų vaizde nurodomos svarbiausios sąveikos ir būsenos diagramos. Išdėstymo vaizde pateikiamas sistemos fizinis išdėstymas.

### 2.5.2. Architektūros tikslai ir apribojimai

Architektūra yra apribota reikalavimų specifikacijoje apibrėžtais apribojimais. Pagrindinis apribojimas yra Eclipse technologijos pasirinkimas įgyvendinant integruotą kūrimo aplinką.

### 2.5.3. Loginis vaizdas

Kuriamos programinės įrangos architektūra pagal RUP [18] metodologiją suskaidyta į penkis paketus, kaip pavaizduota 2.3 paveiksle:



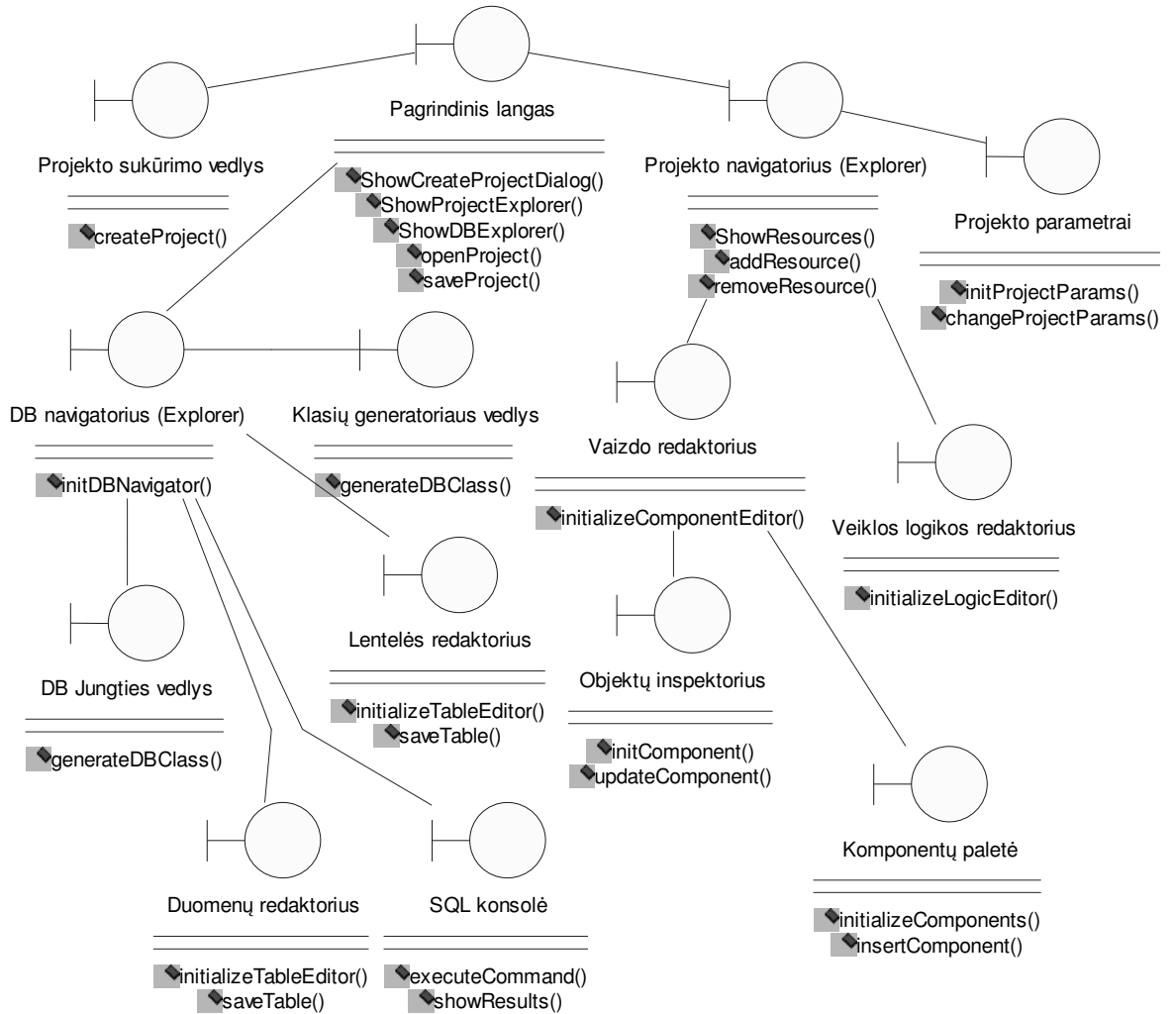
2.3 pav. Sistemos paketų diagrama

### 2.5.4. Paketų detalizavimas

Šiame skyriuje pateikiamos detalizuotos klasių diagramos.

### 2.5.4.1. Vartotojo sąsajos

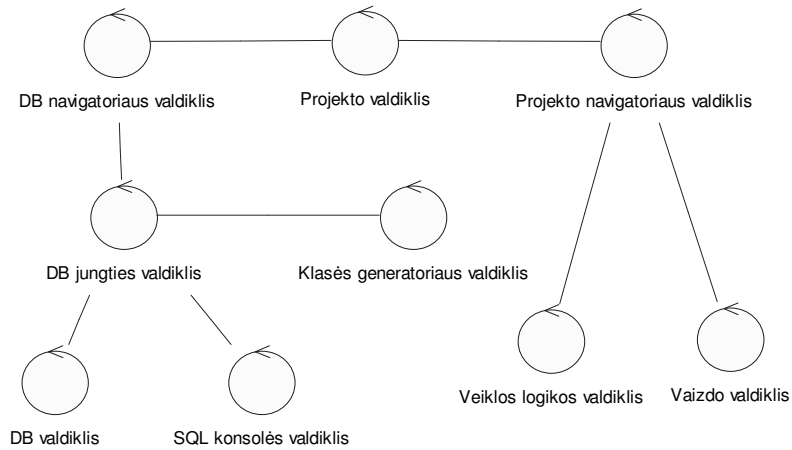
Ši paketą sudaro klasės, skirtos vartotojo sąsajai realizuoti. Vartotojo sąsajos klasės ir ryšiai tarp jų pavaizduoti 2.4 pav.:



2.4 pav. Vartotojo sąsajos klasių diagrama

### 2.5.4.2. Veiklos logikos

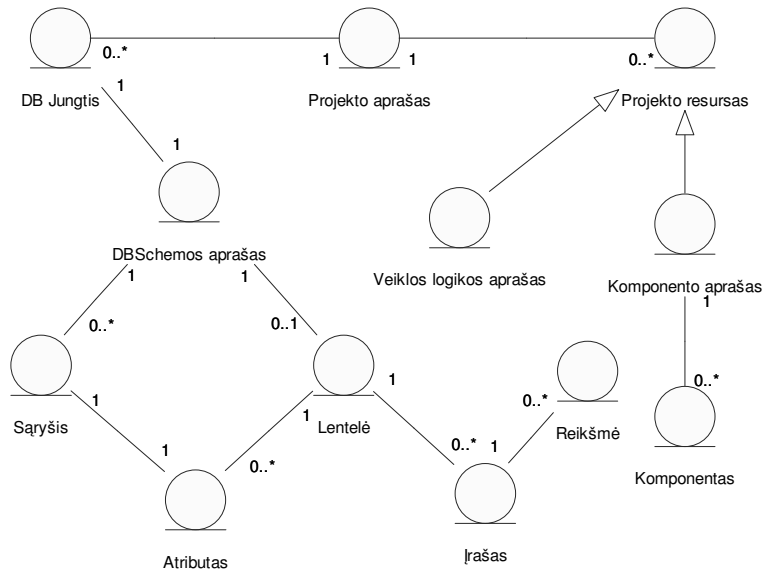
Ši paketą sudaro klasės, skirtos sistemos veiklos logikai aprašyti. Veiklos logikos klasės ir ryšiai tarp jų pavaizduoti 2.5 paveiksle:



2.5 pav. Veiklos logikos klasių diagrama

**2.5.4.3. Duomenų saugojimo**

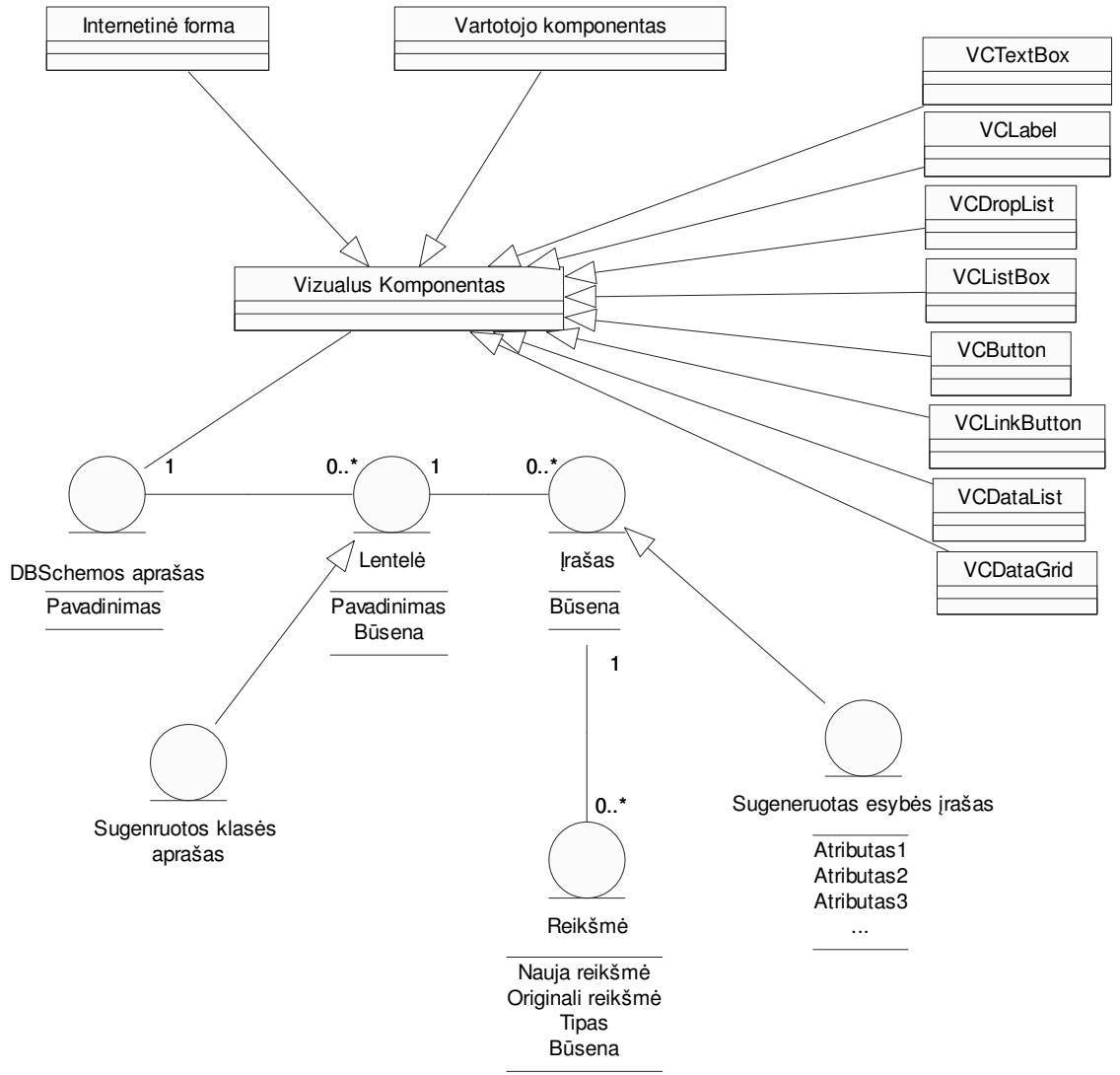
Ši paketą sudaro klasės, skirtos sistemos duomenims saugoti. Duomenų saugojimo klasės ir ryšiai tarp jų pavaizduoti 2.6 paveiksle:



2.6 pav. Duomenų saugojimo klasių diagrama

**2.5.4.4. PHP klasių diagrama**

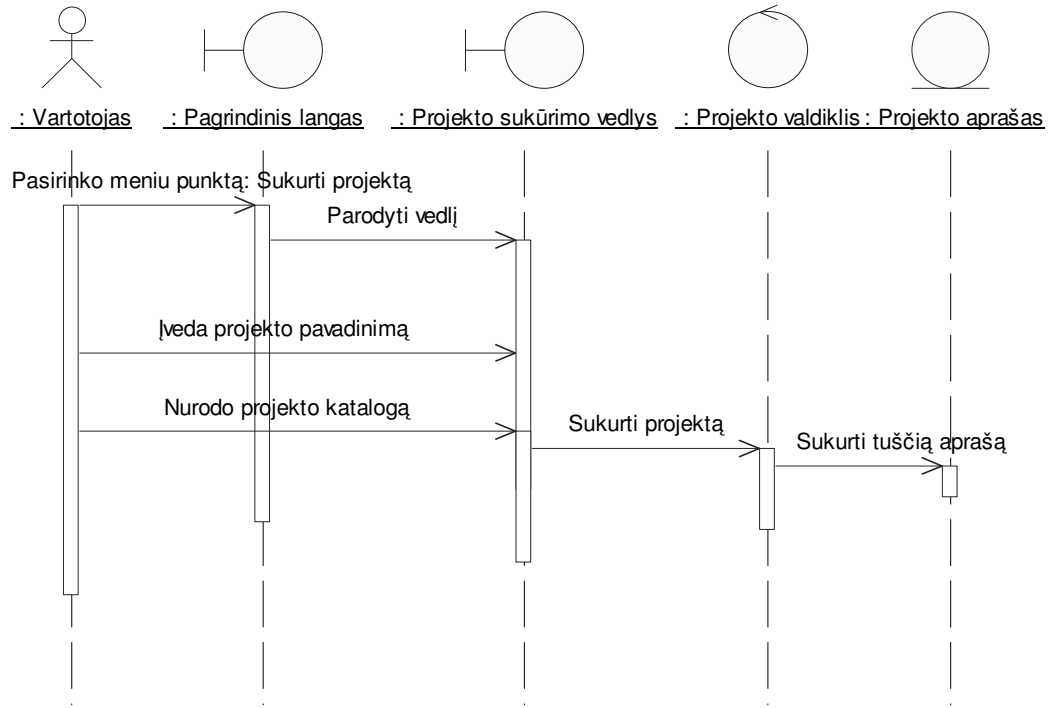
Ši paketą sudaro klasės, realizuojančios vaizdinę komponentų biblioteką. PHP bibliotekos klasės ir ryšiai tarp jų pavaizduoti 2.7 paveiksle:



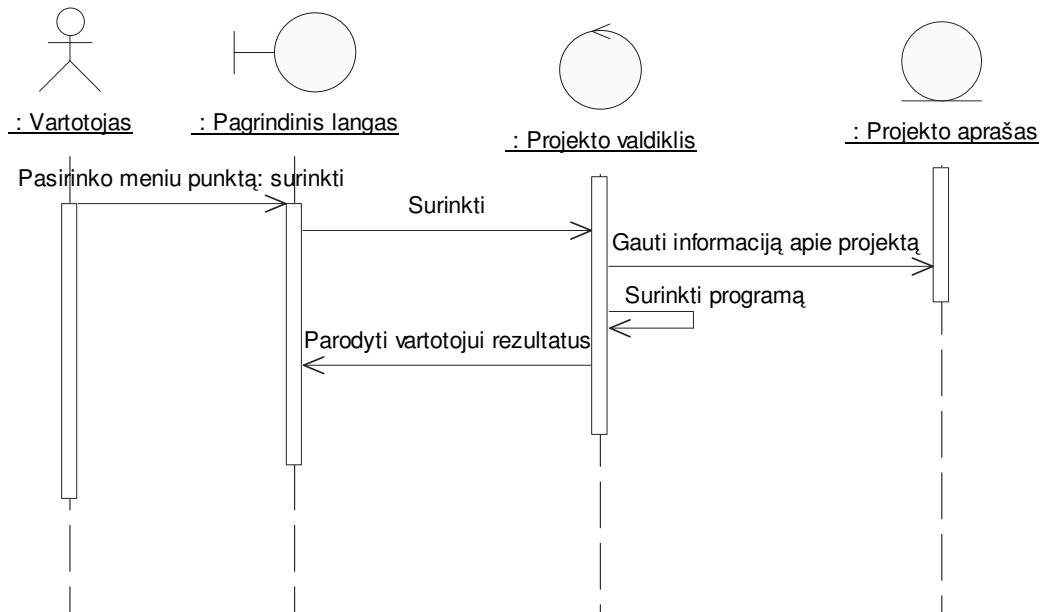
2.7 pav. PHP klasių diagrama



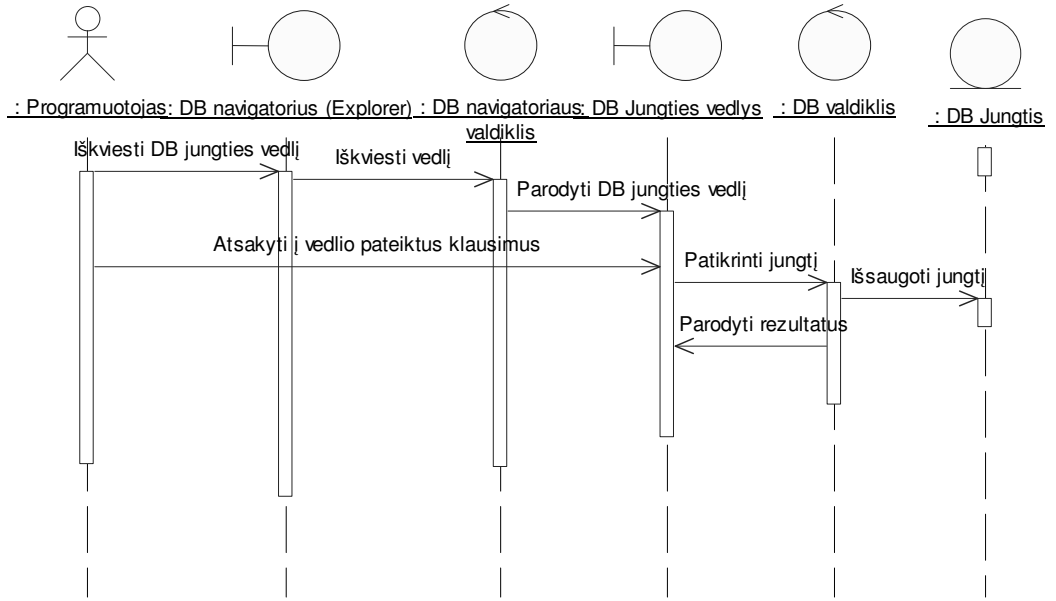
### 2.5.5. Procesų vaizdas



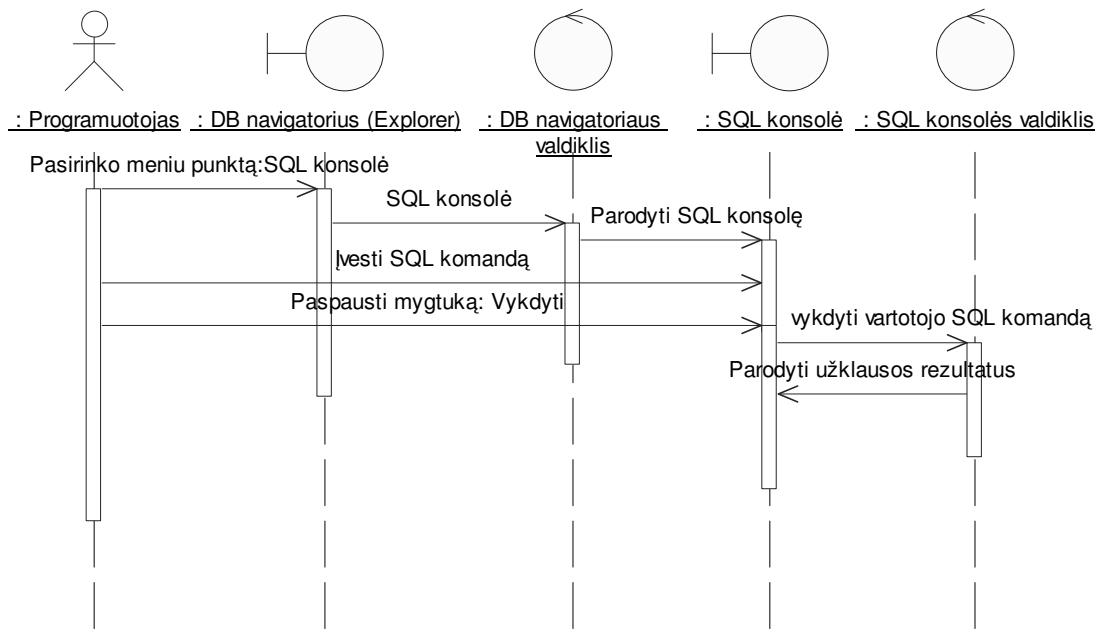
2.8 pav. Projekto sukūrimo sekos diagrama



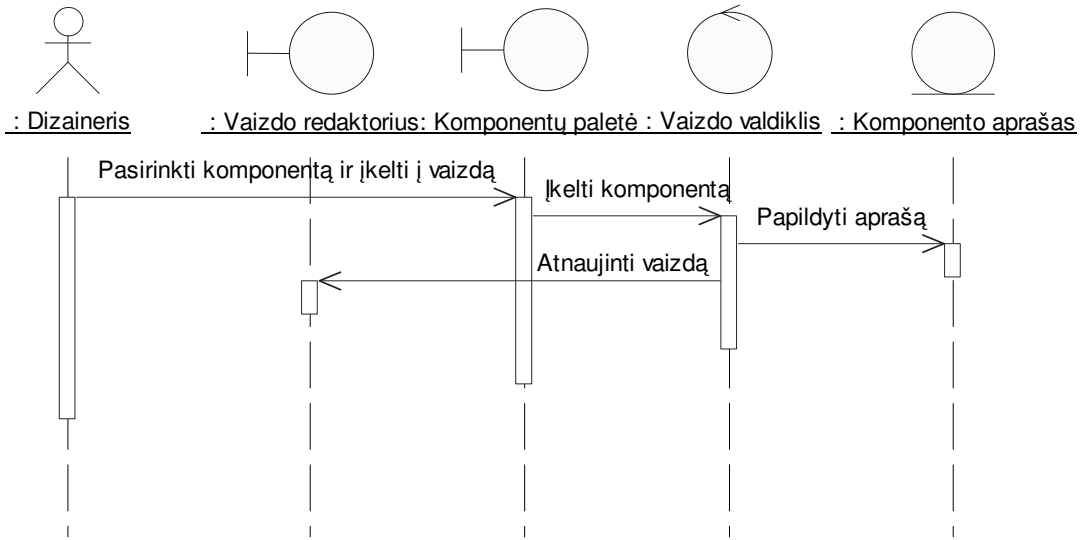
2.9 pav. Interneto programos surinkimo sekos diagrama



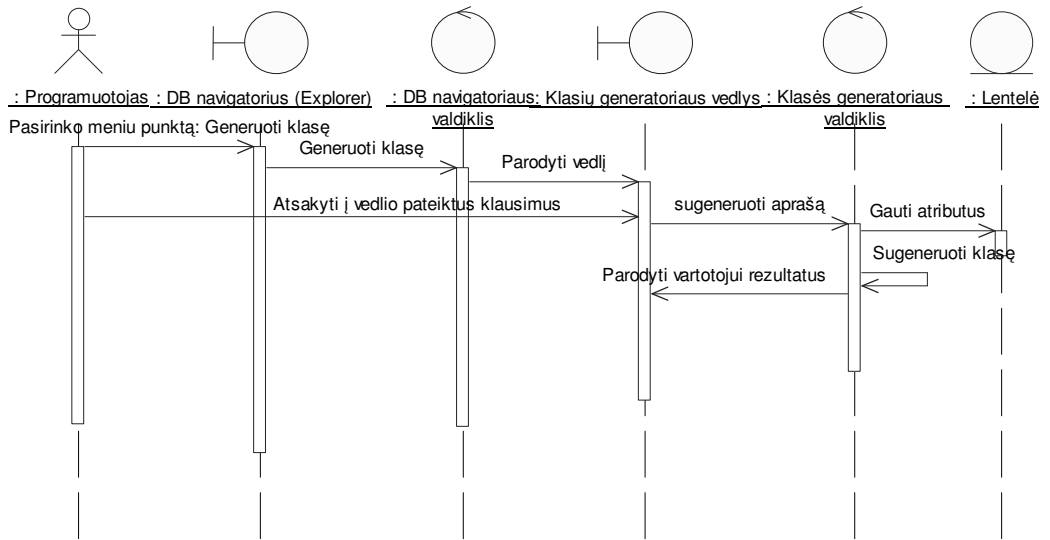
2.10 pav. DB jungties kūrimo sekos diagrama



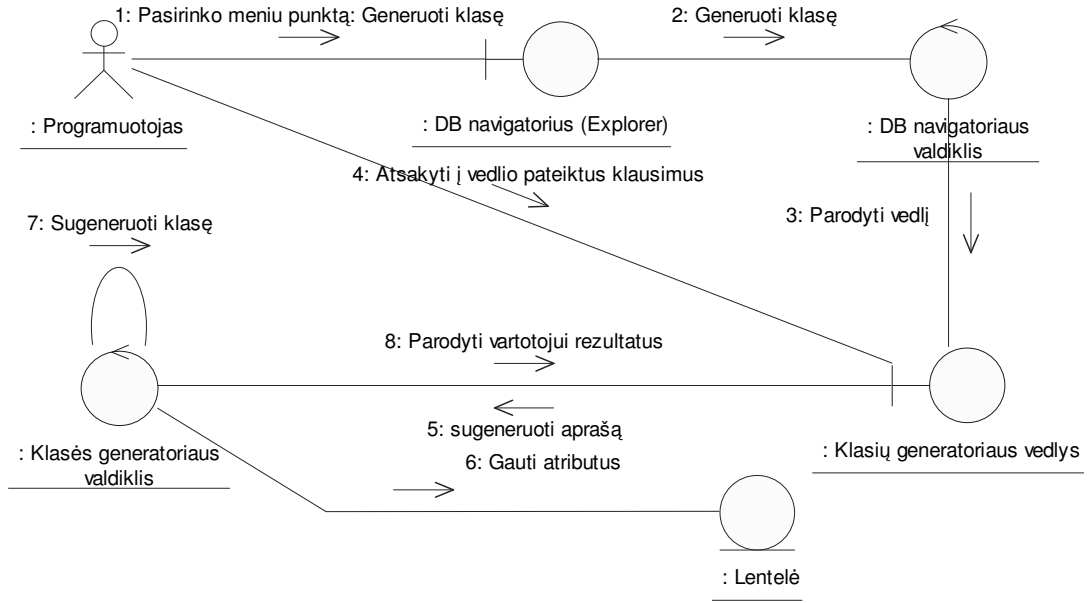
2.11 pav. SQL konsolės sekos diagrama



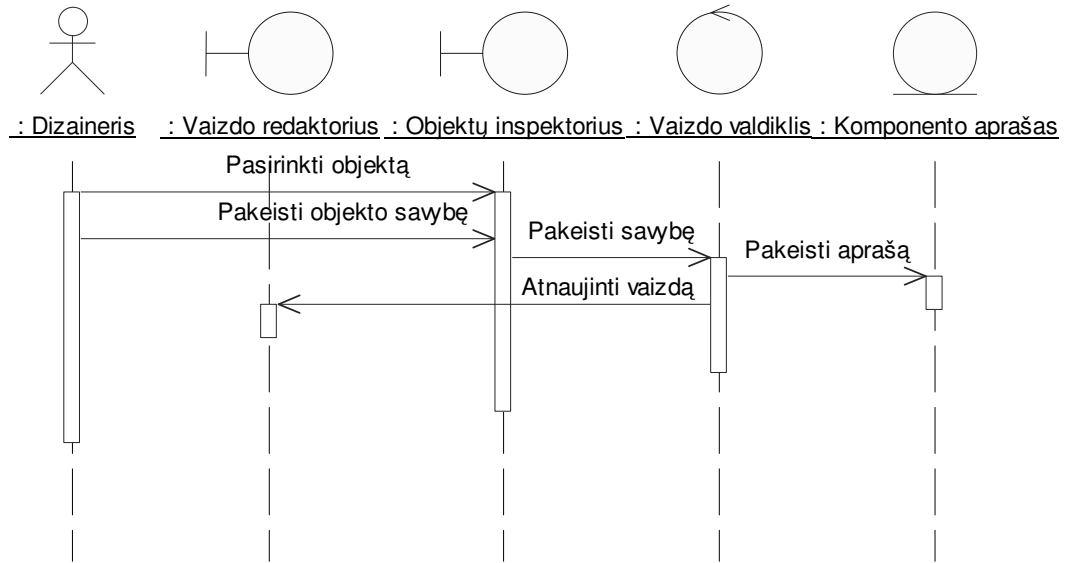
2.12 pav. Komponento įkėlimo sekos diagrama



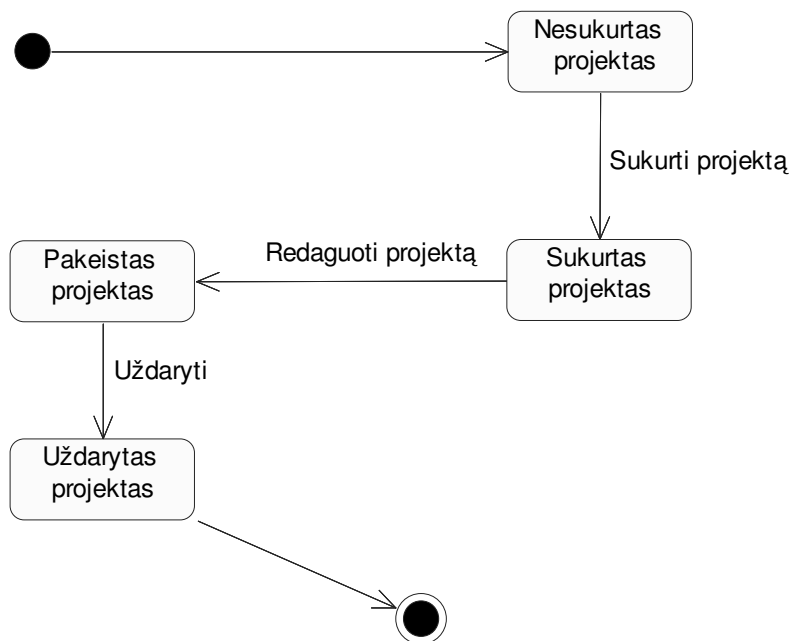
2.13 pav. Klasės darbui su DB generavimo sekos diagrama



2.14 pav. Klasės darbui su DB generavimo bendradarbiavimo diagrama



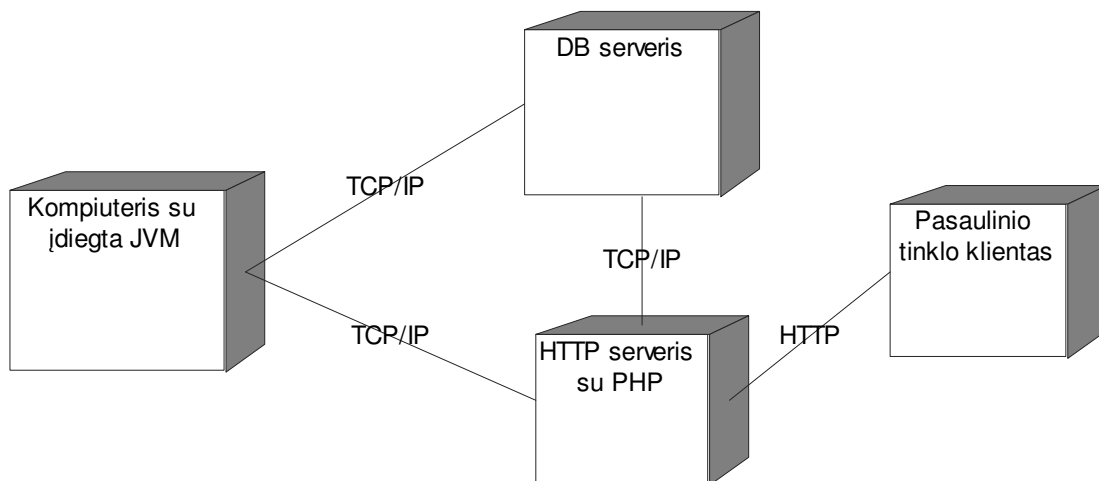
2.15 pav. Objekto savybės reikšmės keitimo sekos diagrama



2.16 pav. Projekto būsenos diagrama

### 2.5.6. Išdėstymo vaizdas

Internetinės duomenų apdorojimo sąsajos kūrimo įrankis veikia Eclipse platformoje. Jis nepriklausomas nuo operacinės sistemos, nes Eclipse yra paremta Java technologija.



2.17 pav. Sistemos išdėstymo diagrama

#### Techninės įrangos reikalavimai:

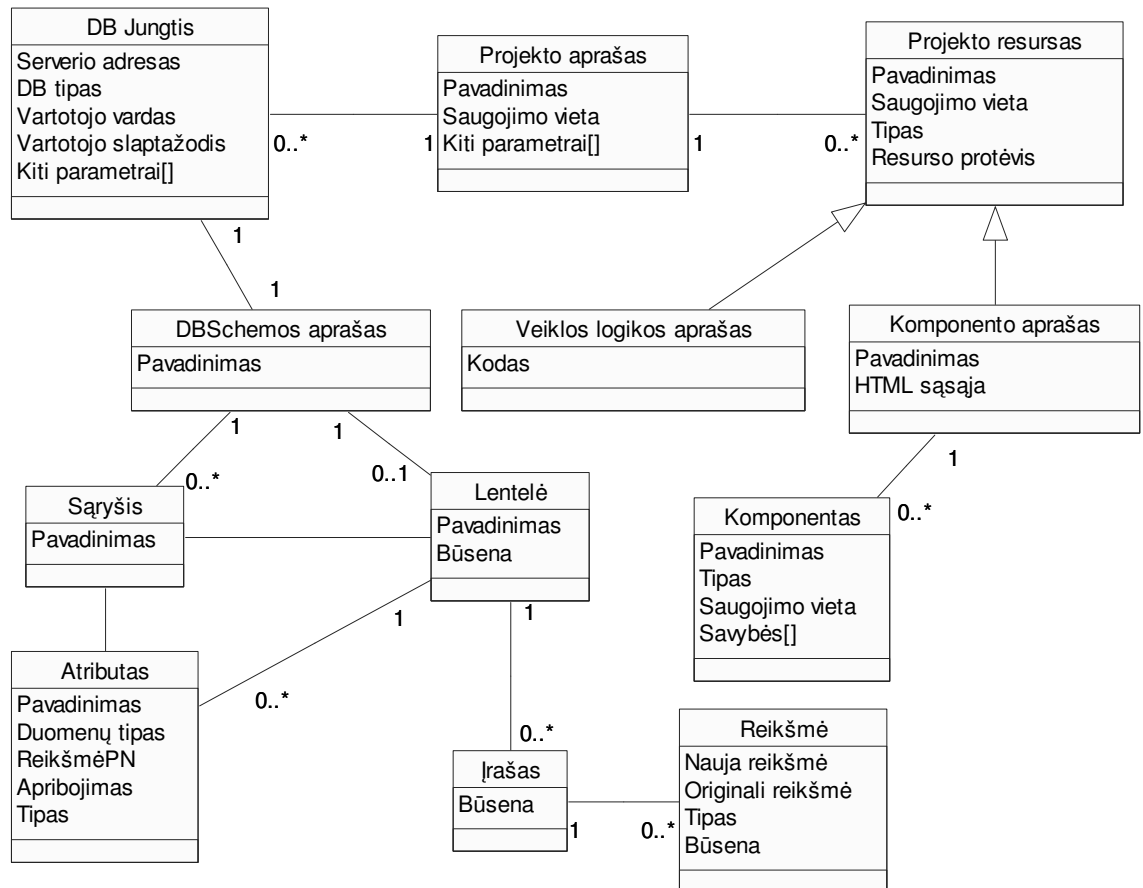
- minimalūs: P3 600 Mhz - 256 MB/s;
- rekomenduojami: P3 1000 MHz - 512 MB/s.

### Sistemos konfigūracija:

- JVM 1.4;
- Windows / Linux (su Mozilla 1.4 GTK2);
- Eclipse 2.1.x arba 3.0 M1 – M8.

### 2.5.7. Duomenų vaizdas

Sistemos duomenys bus saugomi tekstinėse HTML, PHP ir JS rinkmenose. Programos vykdymo metu duomenys saugomi operatyviojoje kompiuterio atmintyje, kurios struktūra pateikta diagramoje:



2.18 pav. Duomenų struktūros

### 3. PHPFORM PROGRAMŲ SISTEMOS KOKYBĖS TYRIMAS

Įdiegus sukurta programinę įrangą, buvo atliktas naudojimo tyrimas. Užregistruotos vartotojų pastebėtos klaidos, trūkumai ir pasiūlymai. Į išsakytas pastabas buvo stengiamasi atsižvelgti toliau plėtojant PHPForm programų sistemą.

#### 3.1. Vartotojų pastebėtos klaidos

Programų sistemos testuotojai:

- Lina Nemuraitė, magistrinio darbo vadovė;
- Mindaugas Naujokaitis, PHP programuotojas, potencialus PHPForm vartotojas.

#### 3.1 lentelė. Vartotojų pastebėtos klaidos

| Nr. | Pastabos, klaidos   | Sprendimas   |
|-----|---|--|
| 1   | Vizualiaame dizaineryje negalima matyti vartotojo formos komponentų sąrašo.                             | Pasirinkus dizainerio režimą standartiniame Eclipse <i>Outline</i> lange, parodomas vartotojo formos vaizdinių komponentų sąrašas. Vartotojui pasirinkus komponentą, <i>Properties</i> lange parodomas komponento savybės. |
| 2   | Vizualiaame dizaineryje pažymėjus vartotojo komponentą, <i>Properties</i> lange neparodomas jo savybės. | Tai klaida, susijusi su panaudotu komponentu <i>Ekit</i> , kuris nėra šio darbo dalis. Šiai klaidai ištaisyti kreiptasi į šio komponento kūrėjus.  |
| 3   | Automatinio kodo baigimas pateikia vieną galimą pasirinkimą, nors jį galėtų automatiškai įterpti.       | Šis pakeitimas įgyvendintas naujoje sistemos versijoje.  |

#### 3.2. Sistemos vertinimas pagal kokybės faktorius

Šiame skyriuje pateikiamas sukurto programų sistemos įvertinimas pagal kokybės faktorius.

##### 3.2.1. Korektiškumas

Sistemos korektiškumą galima patikrinti palyginus realizuotą sistemą su technine užduotimi. Kuriam PHPForm programų sistema yra sudėtinga, o jai keliami reikalavimai kintantys ir sunkiai apibrėžiami, todėl sukurta programų sistema nežymiai skiriasi nuo pradiniuose reikalavimuose jai iškeltų reikalavimų. Padaryti pakeitimai suderinti su užsakovu, taigi šis kokybės kriterijus yra patenkintas.

##### 3.2.2. Patikimumas

Kuriant programų sistemą buvo kreipiamas dėmesys į patikimumą. Svarbu, kad vartotojas, dirbdamas šia sistema, neprarastų duomenų. Eclipse platformoje įdiegta CVS technologija, kuria galima apsaugoti nuo duomenų praradimo, saugant juos saugykloje. Taigi šis kokybės kriterijus yra patenkintas.

### 3.2.3. Efektyvumas

Vienas iš tikslų kuriant sistemą, buvo kuo efektyviau panaudoti resursus. Kadangi IDE sukurta naudojant Eclipse, buvo stengiamasi resursams keliamus reikalavimus išlaikyti tokius pačius, kaip ir Eclipse platformai. Testuojant sukurta programų sistemą, ji veikė greitai ir stabiliai. Jai pakako minimalius reikalavimus atitinkančio kompiuterio resursų greitaveikai išlaikyti, taigi programos efektyvumo kriterijų galima laikyti patenkintu.

### 3.2.4. Panaudojimas

Kuriamos programų sistemos tikslas yra palengvinti interneto programų kūrimą, todėl vartotojui turi būti patogiu ja dirbti. Sukurta programų sistema pasižymi sąsajos intuityvumu. Redaguojant internetinę formą, galima naudotis grafiniu dizaineriu, kuris veikia panašiai kaip „Microsoft Word“ redaktorius. Naujų komponentų įkėlimas labai paprastas, tiesiog reikia iš įrankių juostos pasirinkti norimą komponentą ir ant jo spustelti pelyte. HTML ir PHP redaktoriai pasižymi spalviniu kodo išskyrimu ir automatinio užbaigimo savybe. Taigi programos panaudojamumo kriterijų galima laikyti patenkintu.

### 3.2.5. Testavimas

Programų sistema lengvai testuojama naudojant į Eclipse platformą integruotą automatinį testavimo įrankį JUnit, kuris buvo naudojamas atliekant sistemos testavimą. Sukūrus ir įvykdžius testą, JUnit įrankis parodo, ar jis pavyko. Jeigu nepavyko, programa nurodo, kuris testavimo atvejis suveikė klaidingai.

### 3.2.6. Lankstumas

Programų sistemos konfigūracija yra labai minimali, tačiau tolimesniuose sistemos vystymo etapuose numatyta plėsti konfigūracijos galimybes pagal vartotojų poreikius.

### 3.2.7. Pernešamumas

Sukurta programų sistema yra lengvai pernešama. Ją galima naudoti „Windows“ ir „Linux“ operacinėje sistemoje, nes programų sistema naudojama Java technologija.

### 3.2.8. Pakartotinas panaudojimas

Sistemos veiklos logika yra suprogramuota naudojant objektinės orientacijos programavimo paradigmą ir Java programavimo kalbą. Todėl veiklos logiką galima lengvai panaudoti dar kartą, kuriant programų sistemą, naudojantis kitomis, ne Eclipse, technologijomis. Be to, naudojant Java technologiją, ši



kodą galima vykdyti įvairiose operacinėse sistemose. Taigi sukurta programų sistema pagal šį kokybės faktorių yra patenkinta.

### 3.2.9. Sąsajos galimybės

PHPForm bendrauja su HTTP ir DBVS serveriais. PHPForm kompiliacijos metu kopijuoja paruoštus vykdyti PHP failus į nurodytą HTTP serverio dokumentų katalogą. Programų sistema gali nukopijuoti failus tik į lokalių arba tinkle esančio kompiuterio katalogą. Prieiga per FTP serverį negalima.

Šiuo metu sukurta programų sistema dirba tik su MySQL ir MS SQL server DBVS, vėliau palaikomų DBVS sąrašas bus papildytas. Eclipse platformoje prie duomenų bazės jungiamasi naudojant JDBC. Vaizdinių komponentų bibliotekos realizacijoje į duomenų bazę kreipiamasi naudojant standartines PHP bibliotekas.

### 3.3. Sistemos kokybės vertinimas remiantis vartotojų pastabomis

Vartotojai susipažinę su PHPForm sistema, ją įvertino penkiabalėje sistemoje, pagal tokius kokybės kriterijus:

#### 3.2 lentelė. PHPForm sistemos kokybės vertinimas penkiabalėje sistemoje

| Vartotojas\Vertinimas | Korektiškumas | Efektyvumas | Panaudojamumas | Patikimumas |
|-----------------------|---------------|-------------|----------------|-------------|
| Mindaugas Naujokaitis | 5             | 5           | 5              | 5           |
| Marijus Novikas       | 5             | 4           | 4              | 4           |
| Paulius Penkauskas    | 5             | 5           | 5              | 4           |
| Tomas Urniežius       | 5             | 3           | 4              | 4           |

Remiantis lentelėje pateiktais vertinimais, galima teigti, kad sukurta programų sistema atitinka daugelį kokybės reikalavimų, tačiau dar turi būti tobulinama atsižvelgiant į vartotojų reikalavimus. Šią programų sistemą galima naudoti kaip įrankį, palengvinantį internetinės duomenų apdorojimo sąsajos kūrimą.

## 4. EKSPERIMENTINĖ DALIS

Šioje dalyje atliekamas sukurtos PHPForm programinės įrangos eksperimentinis tyrimas. Šio tyrimo tikslai:

- nustatyti, kaip vartotojai išmoksta dirbti PHPForm sistema;
- kokios vartotojo patirties užtenka, kad jis galėtų efektyviai dirbti sukurta sistema;
- kaip palengvinamas interneto programų kūrimo procesas;
- kokio tipo internetines programas galima sukurti šiuo įrankiu;
- kokios šiuo įrankiu sukurtų interneto programų charakteristikos;
- kiek laiko trunka sukurti tokią pat interneto programą skirtingais įrankiais.

Eksperimentas buvo atliekamas tokiais etapais:

- Pirmame etape vartotojai buvo supažindinami su PHPForm interneto programos modeliu ir įrankiu, skirtu kurti interneto programas. Interneto programos pavyzdys pateiktas 4.1 skyriuje. Taip pat jiems buvo pateiktas vartotojo vadovas, esantis šio darbo antrame priede.
- Antrame etape buvo atliekamas praktinis susipažinimas su sistema. Sukurta minimali interneto programa, demonstruojanti PHPForm galimybes. Interneto programos kūrimas PHPForm sistema aprašytas 4.2 skyriuje.
- Trečiame etape vartotojai bandė kurti interneto programas, skirtas duomenims apdoroti.
- Ketvirtame etape vartotojams buvo pateikta užduotis – sukurti internetinę duomenų apdorojimo sąsają, skirtą vartotojams administruoti.

Po visų etapų vartotojai įvertino PHPForm programų sistemą. Apklausos rezultatai buvo sukaupiti, juos apdorojus suformuluotos išvados apie PHPForm programų sistemą.

### 4.1. Interneto programos pavyzdys

4.1 paveiksle pavaizduota vartotojo forma sudaryta iš įvedimo laukelio ir patvirtinimo mygtuko.

4.2 paveiksle pateiktas šios formos vartotojo sąsajos aprašymas.



The image shows a simple web form. It consists of a single-line text input field with the placeholder text "Enter name" and a rectangular button to its right labeled "Submit".

4.1 pav. Vartotojo formos vaizdas

```

<html>
<head>
</head>
<body>
  <vcform name="userEdit" class="UserEdit" >
  <vctextbox name="textbox1" text="Enter name"/>
  <vcbutton name="button1" text="Submit"/>
</vcform>
</body>
</html>

```

Vartotojo formos aprašas

Vartotojo formos elementai

4.2 pav. Vartotojo formos sąsajos aprašymas

Formos elementai rašomi tarp `<vcform>` žymės. Forma pavadinta `userEdit`, o jos veiklos logika aprašoma PHP klasėje `UserEdit`. Pavaizduota forma susideda iš teksto įvedimo laukelio `textbox1`, kurio reikšmė pagal nutylėjimą *Enter name* ir mygtuko formos informacijai patvirtinti `button1`, kurio tekstas *Submit*.

Aprašytos vartotojo sąsajos veiklos logiką aprašanti klasė `UserEdit` pavaizduota 4.3 paveiksle:

```

<?php
include_once("inc/vcform.php");
class UserEdit extends VCForm
{
  // Auto generated vars
  var $textbox1;
  var $button1;
  // Auto generated vars
  function init($form, $parentId, $name)
  {
    parent::init($form, $parentId, $name);
    // Auto generated init method
    $this->textbox1 = new VCTextBox();
    $this->textbox1->text = 'Enter name';
    $this->textbox1->init($this->form, $this->id, 'textbox1');
    $this->button1 = new VCButton();
    $this->button1->text = 'Submit';
    $this->button1->init($this->form, $this->id, 'button1');
    // Auto generated init method
  }

  // User logic
  // ...
}
?>

```

Automatiškai sugeneruotas kodas, kuris aprašo formos elementų kintamuosius

Automatiškai sugeneruotas kodas, kuris inicializuoja formos elementus

Vartotojo atliktų veiksmų apdorojimas

4.3 pav. Automatiškai sugeneruota vartotojo formos logiką aprašanti klasė

Vartotojo formos klasė yra paveldėta iš vaizdinių komponentų bibliotekos klasės `VCForm`. Ji automatiškai generuojama iš vartotojo sąsajos aprašymo.

## 4.2. Interneto programos kūrimas PHPForm sistema

Šiame skyriuje aprašoma, kaip naudojantis PHPForm instrumentinėmis priemonėmis sukurti interneto programą. Vartotoja, susipažinęs su šiomis instrukcijomis, turi pats pamėginti dirbti PHPForm sistema ir pateikti atsiliepimus, iš kurių bus galima padaryti išvadas apie PHPForm sistemą.

4.4 paveiksle pavaizduotas naujo projekto sukūrimo vedlys. Užpildžius jį bus sukurtas naujas interneto programos projektas.

**New Project**

**Create new PHPForm Project**

This wizard creates a new PHPForm Project.

Project name:

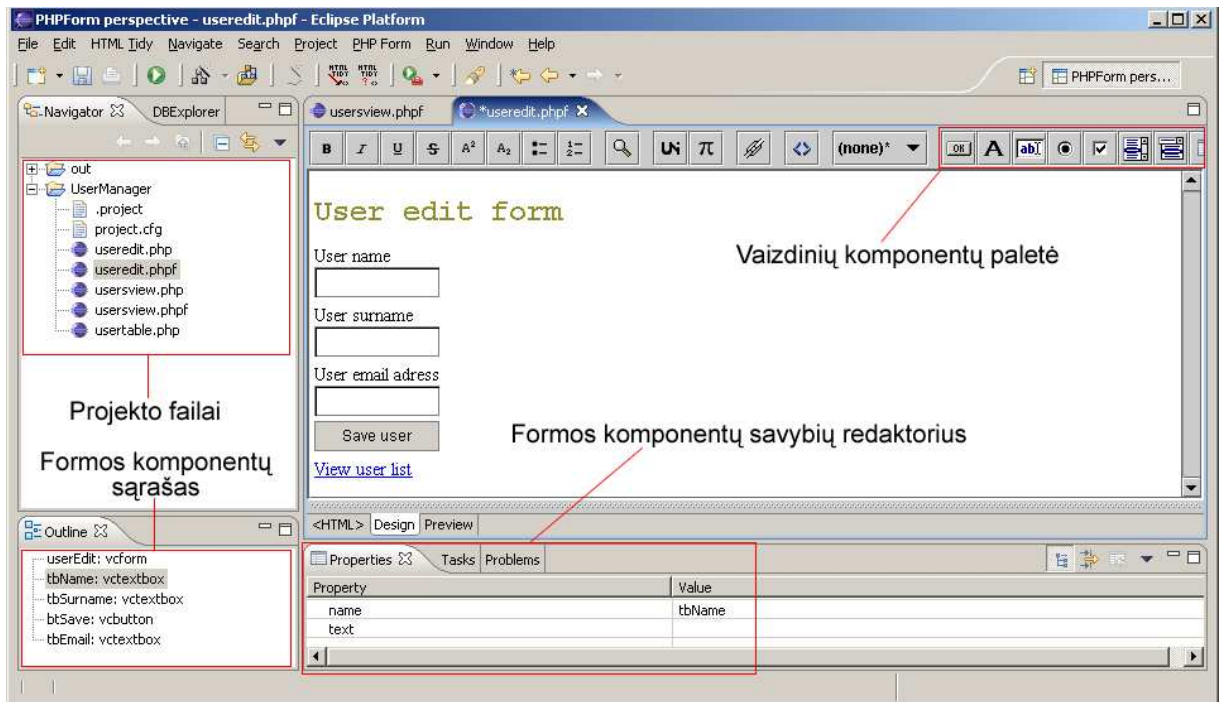
Deploy location:

Project location:

Site URL:

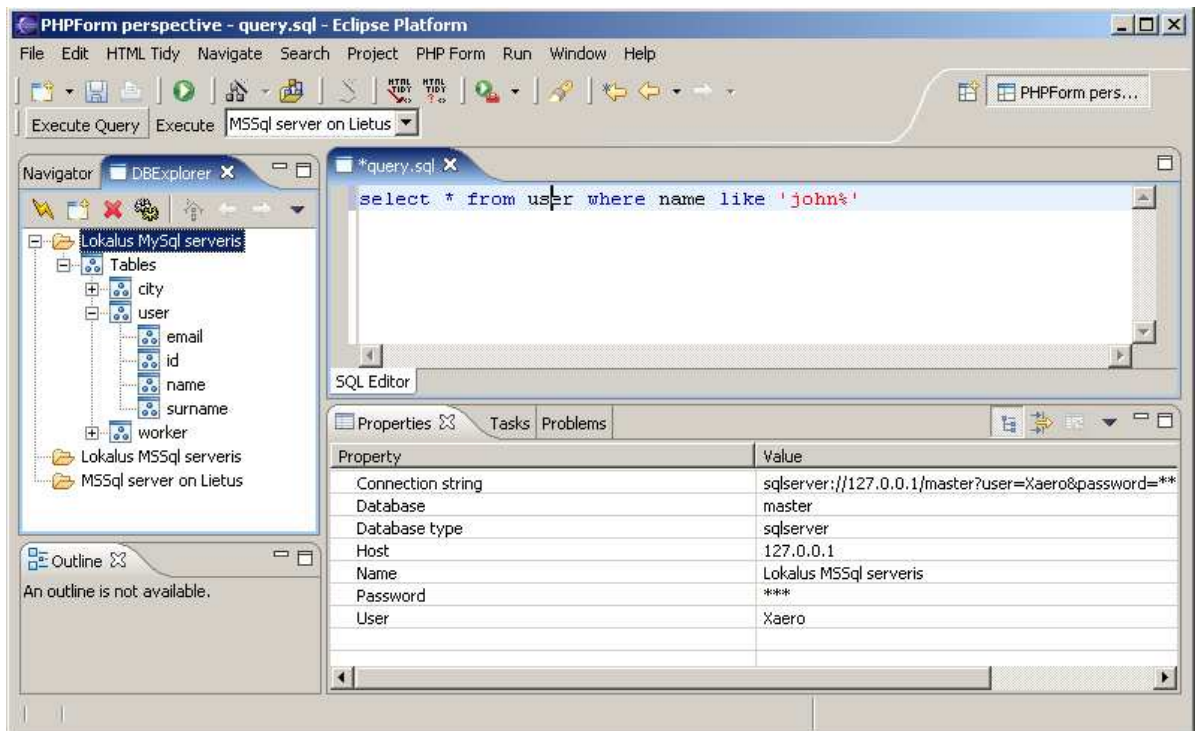
4.4 pav. Naujo projekto sukūrimo vedlys

Sukūrus interneto programos projektą, atsidaro 4.5 paveiksle pavaizduotas PHPForm pagrindinis langas. Jame pavaizduota projektų failų navigatorius, vizualus internetinių formų dizaineris, internetinės formos komponentų sąrašas ir komponentų savybių redaktorius.



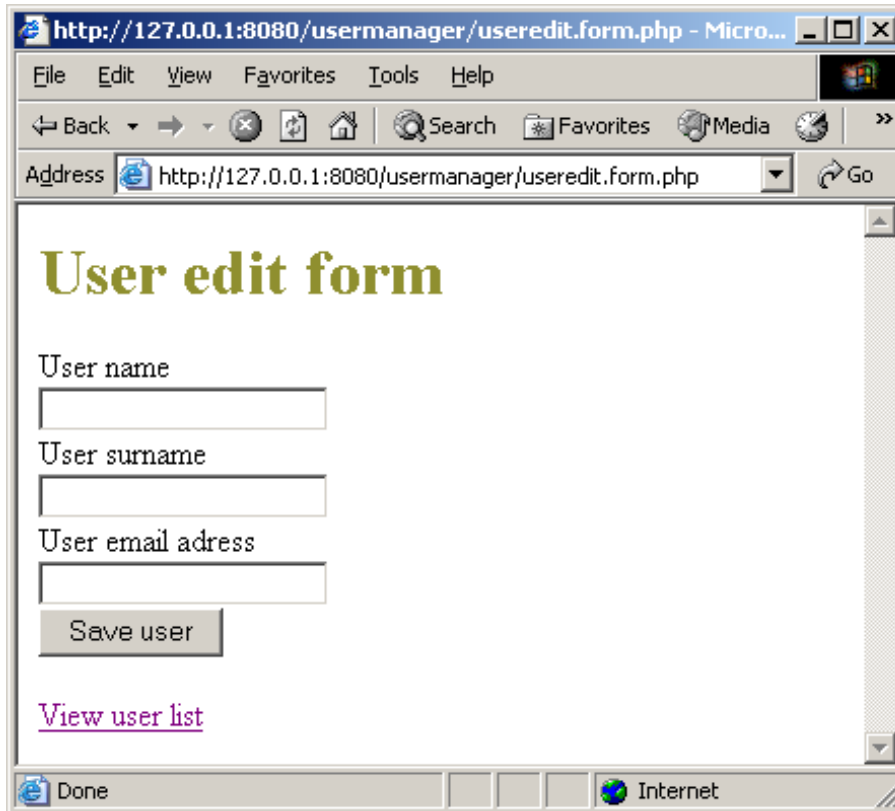
4.5 pav. PHPForm pagrindinis langas

Pasirinkęs *DBExplorer* langą, vartotojas gali prisijungti prie duomenų bazių valdymo sistemos (DBVS) ir atlikti veiksmus su duomenų baze, kaip pavaizduota 4.6 paveiksle (plačiau žiūrėti vartotojo vadove, kuris pateiktas antrame priede).



4.6 pav. Darbas su duomenų baze

Sukūrus internetinės formos vartotojo sąsają ir aprašius jos veiklos logiką, interneto programą galima vykdyti serveryje. Tam reikia projektų failų navigatoriuje pasirinkti norimą formą ir meniu punktą *PHPForm->Open in Browser*. PHPForm programų sistema automatiškai sugeneruos vykdomą PHP kodą ir nusiųs jį į HTTP serverį, kuriame bus vykdoma interneto programa. Tada bus atidaromas interneto naršyklės langas su pasirinkta internetine forma, kaip pavaizduota 4.7 paveiksle.



4.7 pav. Sukurta internetinė forma

### 4.3. Eksperimento rezultatų tyrimas

Vartotojai, susipažinę su PHPForm sistema, ją įvertino penkiabalėje sistemoje pagal tokius kriterijus:

- įsisavinimas – kaip vartotojui sekėsi mokytis dirbti PHPForm sistema;
- funkcionalumas – kaip kuriant internetines programas buvo panaudotas PHPForm sistemos funkcionalumas;
- našumo pagerinimas – kaip PHPForm sistema pagreitino ir palengvino internetinės duomenų apdorojimo sąsajos kūrimo procesą;
- programos charakteristikos – kaip vertina sukurtų interneto programų charakteristikas;
- naudingumas – ar PHPForm sistema naudinga vartotojui.

**4.1 lentelė. PHPForm sistemos vertinimas penkiabalėje sistemoje**

| <b>Vartotojas\Vertinimas</b> | <b>Funkcion<br/>alumas</b> | <b>Išsavinimas</b> | <b>Našumo<br/>pagerinimas</b> | <b>Programos<br/>charakteristikos</b> | <b>Naudin<br/>gumas</b> |
|------------------------------|----------------------------|--------------------|-------------------------------|---------------------------------------|-------------------------|
| Mindaugas Naujokaitis        | 5                          | 4                  | 5                             | 5                                     | 5                       |
| Marijus Novikas              | 5                          | 4                  | 5                             | 4                                     | 4                       |
| Paulius Penkauskas           | 4                          | 5                  | 5                             | 5                                     | 4                       |
| Tomas Urniežius              | 3                          | 3                  | 4                             | 4                                     | 3                       |

Vartotojai, susipažinę su PHPForm sistema, buvo apklausiami. Apklausos rezultatai buvo sukaupti ir apdoroti. Suformuluotos išvados apie PHPForm programų sistemą. Iš gautų atsiliepimų padarytos išvados:

- PHPForm sistema yra nesunkiai išsavinama;
- nereikalauja didelės vartotojo patirties;
- palengvina ir pagreitina programuotojo darbą:
  - dėl interneto programos modelio savybių;
  - kūrimo procese užtenka vieno įrankio galimybių;
- sukurtą interneto programą lengva modifikuoti;
- šiuo įrankiu patogiausia kurti nedideles ir vidutines interneto programas, skirtas duomenims apdoroti.

## IŠVADOS

Magistriniame darbe apžvelgti interneto programų modeliai ir aptartos instrumentinės interneto programų kūrimo priemonės. Atlikus analizę, pasiūlytas interneto programos modelis, skirtas internetinei duomenų apdorojimo sąsajai kurti. Jis paremtas vartotojo sąsajos, atskirtos nuo veiklos logikos, architektūra. Palyginti su MVC ir Microsoft ASP.NET, pasiūlytas modelis yra paprastesnis ir greičiau realizuojamas. Vartotojo sąsaja yra atskirta nuo logikos, todėl atlikti pakeitimai neturi įtakos kitoms programos vietoms. Kadangi vartotojo formos išvaizda ir kodas generuojamas į vykdomą PHP scenarijų, tai našumas panašus į vieno sluoksnio architektūros programos.

Sukurtos PHPForm instrumentinės priemonės, palyginti su egzistuojančiomis sistemomis, turi daug pranašumų. Jomis galima greitai ir paprastai kurti internetines formas, skirtas duomenims įvesti ir redaguoti. Išmokti dirbti sukurta sistema nėra sudėtinga. Daugelį veiksmų galima atlikti naudojant vedlius arba vizualų dizainerį.

Sukurta sistema nėra universali, ji orientuota tik į duomenų apdorojimo formų kūrimą. Todėl sistema nėra apkrauta nereikalingomis funkcijomis ir veikia greičiau, nei šiame darbe apžvelgtos panašios sistemos.

Apibendrintos išvados:

- apžvelgti interneto programų modeliai ir instrumentinės priemonės joms kurti;
- sukurtas interneto programos modelis, paremtas vartotojo sąsajos, atskirtos nuo logikos, architektūra;
- sukurtas įrankis, realizuojantis pasiūlytą modelį, kuriuo galima kurti internetines formas, skirtas duomenims įvesti ir redaguoti;
- sukurta programų sistema palengvina ir pagreitina interneto programų kūrimą;
- sukurta sistema galima tobulinti išplečiant vartotojo komponentų kūrimo galimybes;
- šio darbo tematika skaitytas pranešimas tarpuniversitetinėje magistrantų ir doktorantų konferencijoje Informacinės technologijos'05.



## LITERATŪRA

1. TERENCE P. *Enforcing strict model-view separation in template engines*. International World Wide Web Conference, 2004, ACM Press, New York.
2. MEHDI A., F. Betz. *PHP Manual* [interaktyvus]. [žiūrėta 2005 m. kovo 10]. Prieiga internete <<http://www.php.net/manual/en/>>, 2005.
3. RATSCHILLER T., GERKEN T. *Web Application Development with PHP 4.0*. New Riders, 2000.
4. Microsoft. *Introduction to ASP.NET* [interaktyvus]. [Žiūrėta 2005 m. kovo 15]. Prieiga internete <<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconintroductiontoasp.asp> >
5. Microsoft. *Introduction to Web Forms Pages*. [interaktyvus]. [žiūrėta 2005 m. kovo 22]. Prieiga internete <<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconIntroductionToWebForms.asp>>
6. MURRAY G., JOHNSON M. *Web - Tier Application Framework Design. Designing Enterprise Applications with the J2EETM Platform* [interaktyvus]. [žiūrėta 2003 m. vasario 12]. Prieiga internete <[http://java.sun.com/blueprints/guidelines/designing\\_enterprise\\_applications\\_2e/DEA2eTOC.html](http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/DEA2eTOC.html) >
7. PERRONE P. *Building Java Enterprise Systems with J2EE*. Sams. 2000.
8. Sun Microsystems. *JavaServer Pages* [interaktyvus]. [Žiūrėta 2004 m. sausio 27]. Prieiga internete <<http://java.sun.com/products/jsp/>>
9. Rational software. *Web Architecture Patterns*. [interaktyvus]. [žiūrėta 2004 m. sausio 20]. Prieiga internete <[http://www.rational.com/products/rup/tryit/eval/RationalUnifiedProcess/process/workflow/ana\\_desi/](http://www.rational.com/products/rup/tryit/eval/RationalUnifiedProcess/process/workflow/ana_desi/)>
10. DAVIS M. *Struts, an open-source MVC implementation* [interaktyvus]. 2001. [žiūrėta 2005 m. gegužės 10]. Prieiga internete <<http://www-106.ibm.com/developerworks/java/library/j-struts/>>
11. Macromedia. *Macromedia DreamWeaver* [interaktyvus]. [žiūrėta 2005 m. vasario 25]. Prieiga internete <<http://www.macromedia.com>>
12. Microsoft. *Microsoft Visual Studio .NET* [interaktyvus]. [žiūrėta 2005 m. kovo 15]. Prieiga internete <<http://h20276.www2.hp.com/drc/topics/Net.jsp>>

13. Microsoft. *Microsoft Visual Studio .NET* [interaktyvus]. [žiūrėta 2005 m. kovo 16]. Prieiga internete <<http://msdn.microsoft.com/vstudio/howtobuy/pricing.aspx>>
14. Object Technology International. *Eclipse Platform Technical Overview* [interaktyvus]. 2003 [žiūrėta 2005 m. gegužės 10]. Prieiga internete <<http://www.eclipse.org/whitepapers/eclipse-overview.pdf>>
15. Object Technology International. *Eclipse platform* [interaktyvus]. [žiūrėta 2005 kovo 12]. Prieiga internete < <http://www.eclipse.org> >
16. BOEKHOUDT Caspar. *The Big Bang Theory of IDEs*. Queue, Volume 1. 2003.
17. THORSTEINSON P., OBERG R. *NET Architecture and Programming Using Visual C++*. Prentice Hall 2002, 29 p.
18. Rational software. *Rational Unified Process* [interaktyvus]. [žiūrėta 2005 kovo 18]. Prieiga internete <<http://www-306.ibm.com/software/awdtools/rup/>>

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

- HyperText Transfer Protocol (HTTP) – protokolas, skirtas hipertekstui perduoti internetu;
- Hyper Text Markup Language (HTML) – kalba, naudojama internetiniams puslapiams aprašyti;
- Interneto programa (angl. *web application*) – programa, kuri kaip pagrindinį komunikavimo protokolą naudoja HTTP ir perduoda informaciją vartotojui HTML kalba;
- Vaizdinis komponentas – komponentas, sudarytas iš:
  - vartotojo sąsajos (išvaizdos);
  - serverio pusėje vykdomo kodo, kuris apdoroja vartotojo sužadintus įvykius;
  - kliento pusėje vykdomo kodo, kuris apdoroja vartotojo sužadintus įvykius.

Į komponentą galima įdėti kitą komponentą.

- Standartinis komponentas – su kuriama sistema pateikiami komponentai, pavyzdžiui, Textbox, Button, Listbox;
- Vartotojo komponentas – vartotojo sukurtas komponentas;
- Internetinė forma – vartotojo sukurta forma, sudaryta iš standartinių ir vartotojo komponentų, iš kurių galima sugeneruoti interneto programą;
- Web client – interneto klientas (internetu naršyklė);
- PHP – Hypertext Preprocessor;
- MVC (modelis – vaizdas – valdiklis);
- DB – duomenų bazė;
- DBVS – duomenų bazių valdymo sistema;
- FTP – File Transfer Protocol;
- JDBC – Java Database Connectivity;
- Unified Modeling Language (UML) – universali modeliavimo kalba;
- Rational Unified Process (RUP) – apibendrintas projektavimo metodas;
- Integrated development environment (IDE) – integruota vystymo aplinka;
- Event handling – veiksmų apdorojimo mechanizmas;
- Eclipse perspektyva – nusako, kokie langai bus matomi ir kokius veiksmus bus galima atlikti Eclipse platformos lange;
- Eclipse plug-in – mažiausias Eclipse platformos funkcijos vienetas.

## **5. 1 PRIEDAS. Vaizdinės priemonės internetinei duomenų apdorojimo sąsajai kurti. Informacinės technologijos'05**

*KTU Informatikos fakultetas, Studentų g. 50, Kaunas*

Sparčiai didėjant pasaulinio tinklo programų populiarumui, atsirado poreikis palengvinti jų kūrimą. Šiame straipsnyje nagrinėjamos internetinės duomenų apdorojimo sąsajos kūrimo priemonės ir būdai, kaip galima palengvinti pasaulinio tinklo programų kūrimo procesą. Straipsnyje apžvelgiami pasaulinio tinklo programų modeliai ir instrumentinės priemonės joms kurti. Remiantis šia analize, pasiūlytas modelis, kuris geriausiai tiktų greitam duomenų apdorojimo sąsajos kūrimui. Jis paremtas vartotojo sąsajos, atskirtos nuo veiklos logikos, architektūra. Pasiūlytas modelis realizuotas PHPForm programų sistema Eclipse platformoje, atlikta sukurto ir egzistuojančių produktų lyginamoji analizė.

### **5.1. Įvadas**

Pasaulinio tinklo programos (angl. Web application) tampa vis populiarsnės. Tai susiję su tuo, kad kai kuriais aspektais jos teikia gerokai daugiau galimybių negu tradicinės programos (angl. desktop application). Vienas iš pagrindinių pasaulinio tinklo programos privalumų – prieinamumas, ja gali naudotis daug vartotojų. Be to, norint naudotis bet kokia pasaulinio tinklo programa, nereikia diegti papildomos programinės įrangos, užtenka paprastos interneto naršyklės.

Pasaulinio tinklo programų kūrimas yra sudėtingas, daug žinių ir patirties reikalaujantis procesas. Programuotojo darbui palengvinti, galima naudoti įvairias standartines bibliotekas ir įrankius (IDE), skirtus pasaulinio tinklo programoms kurti. Joms kurti yra skirta nemažai universalių komercinių produktų, tačiau jie turi nemažai trūkumų: didelė kaina, sudėtingi, reikalaujantys daug žinių. Straipsnyje apžvelgiami pasaulinio tinklo programų modeliai ir instrumentinės priemonės joms kurti. Remiantis šia analize, pasiūlytas modelis, kuris geriausiai tinka greitam duomenų apdorojimo sąsajos kūrimui. Taip pat aprašomos sukurtos instrumentinės priemonės, realizuojančios pasiūlytą modelį, bei atliekamas sukurtos sistemos palyginimas su jau egzistuojančiomis sistemomis.

### **5.2. Pasaulinio tinklo programų architektūrų ir technologijų analizė**

#### **5.2.1. Pasaulinio tinklo programų modelių ir technologijų apžvalga**

Pasaulinio tinklo programų kūrimui naudojamos ASP, ASP.NET, JSP, PHP, „Cold Fusion“ technologijos. Viena populiariausių nemokamų priemonių, naudojamų kuriant pasaulinio tinklo programas, yra PHP (PHP: Hypertext Preprocessor). PHP kalba parašyti scenarijai yra interpretuojami tinklo serveryje. PHP palaiko objektinio programavimo paradigmą, yra panaši į C, C++, Java kalbas, lengvai įsisavinama, turi funkcijų, skirtų dirbti su failais, XML, duomenų bazėmis ir kt., bibliotekas.

Pats paprasčiausias pasaulinio tinklo programų architektūros modelis yra vieno sluoksnio, kai vartotojo sąsaja ir veiklos logika aprašoma viename faile. Šis modelis yra greičiausiai realizuojamas,

tačiau tinka tik mažoms programoms. Jis turi daug trūkumų, kurie išryškėja didėjant programos sudėtingumui. Sudėtingoje pasaulinio tinklo programoje atlikti kokius nors pakeitimus labai sunku, nes ji yra sunkiai skaitoma ir dažnai maži pakeitimai vienoje vietoje gali sugadinti visą programą. Norint atlikti vieną pakeitimą, gali prireikti perrašyti programą keliose vietose [1]. Šis modelis gali būti realizuojamas PHP, ASP arba JSP technologijomis.

ASP.NET – „Microsoft“ sukurta technologija, skirta pasaulinio tinklo programų kūrimui. Tai yra nauja ASP versija, kuri skiriasi nuo savo pirmtakės, tačiau visiškai su ja suderinama. Ji ne tik suteikia galimybę kurti dinaminis tinklalapius, kaip jos pirmtakė, bet ir aprūpina unifikuotu pasaulinio tinklo programų kūrimo modeliu, kuris suteikia galimybę kurti pasaulinio tinklo programas didelėms įmonėms. Vartotojo forma sudaryta iš dviejų dalių: išvaizdos ir elgsenos. Išvaizda aprašoma .aspx faile panaudojant HTML elementus ir specialiomis ASP.NET bibliotekoje arba vartotojo aprašytomis žymėmis. Puslapio formos elgsena aprašoma atskirame faile bet kuria iš .NET palaikomų kalbų [2].

„Java Servlet“ išplečia pasaulinio tinklo serverio galimybes ir leidžia prieiti prie egzistuojančių verslo sistemų. „Java Servlet“ galima laikyti kaip įsiuvą, kuris veikia serverio pusėje, tačiau neturi vartotojo sąsajos. Informaciją į išorę galima išvesti System.print funkcija. Ši technologija skirta daugiau veiksmams serverio pusėje atlikti, nei vartotojo sąsajos kūrimui. Tam naudojama JSP (angl. JavaServer Pages) technologija, skirta į HTML puslapius įterpti Java kodo vykdymui. Serveryje vykdant JSP puslapį, iš jo sugeneruojamas „Java Servlet“, HTML kodas įkeliamas į System.print funkciją.

Kitas pasaulinio tinklo programų architektūros modelis yra MVC (modelis – vaizdas – valdiklis). Modelis yra programos veiklos logika, vaizdas – tai duomenų pateikimo vartotojui priemonė (HTML puslapis), o valdiklis – programos dalis, nustatanti, kokius veiksmus atlikti ir kokį vaizdą pateikti vartotojui. Šis modelis skirtas didelių pasaulinio tinklo programų kūrimui. Laikydamasis jo, vartotojas sukuria struktūrizuotą, lengvai skaitomą pasaulinio tinklo programos kodą. Egzistuoja kelios bibliotekos, realizuojančios šį architektūros modelį. Viena iš jų nemokama „Struts“ biblioteka, kuri realizuota naudojant JSP. Pagrindinis „Struts“ realizacijos trūkumas – viena pasaulinio tinklo programa turi tik vieną valdiklį. Be to, reikia nemažai laiko norint perprasti, kaip sukurti paprasčiausią pasaulinio tinklo programą.

1 lentelė. Technologijų, skirtų pasaulinio tinklo programoms kurti, palyginimas penkiabalėje sistemoje.

| Kriterijus\ Vertinimas  | ASP.NET | JSP Struts | PHP (vieno sluoksnio) |
|-------------------------|---------|------------|-----------------------|
| Ar lengvai panaudojamas | 4       | 2          | 5                     |
| Programos plečiamumas   | 4       | 5          | 3                     |
| Kaina                   | 3       | 5          | 5                     |

### 5.2.2. Įrankių, skirtų pasaulinio tinklo programoms kurti, apžvalga

Gana populiarius interneto svetainių kūrimo įrankis yra „Macromedia DreamWeaver“. Juo galima vizualiai sukurti sudėtingą vartotojo sąsają, taip pat jis turi patogų kodo redaktorių. Tai universalus įrankis, kuriuo galima kurti įvairiausių tipų pasaulinio tinklo programas, naudojančias ASP.NET, PHP, Cold Fusion, JSP technologijas. Jo trūkumai:

- brangus – viena jo kopija kainuoja 479 JAV dolerius;
- apkrauta vartotojo sąsaja – turi daug funkcijų, kurių retai reikia vartotojui;
- naudojant vizualų redaktorių, sugadinamas HTML kodo stilius, HTML žymės naudojamos neteisingai arba kartojasi [3].

Microsoft Visual Studio .NET produktas skirtas pasaulinio tinklo programoms, paremtoms ASP.NET technologija, kurti. Kuriant programas, naudojami RAD (angl. rapid application development) principai. Tai universalus, daug galimybių turintis įrankis. Jo trūkumai:

- brangus – viena jo kopija kainuoja 1799 JAV doleriu;
- veikia tik Windows 2000, XP operacinėse sistemose;
- sukurtos pasaulinio tinklo programos veikia tik „Microsoft IIS“ serveryje.

Eclipse yra nemokama platforma, pritaikyta integruotų vystymo aplinkų (IDE) kūrimui, kurios gali būti naudojamos kurti taikomąsias ir pasaulinio tinklo programas. Ja galima kurti įvairių tipų projektus, pavyzdžiui, Java, JSP, PHP. Pagrindinė Eclipse savybė yra išplečiamumas, tam naudojama plug-in technologija. Eclipse plug-in – mažiausias Eclipse platformos funkcijos vienetas. Eclipse turi specialų įrankių rinkinį Plug-in Development Environment (PDE), kuriuo galima išplėsti Eclipse platformos funkcionalumą. Eclipse sukurta naudojant Java technologiją, todėl veikia daugelyje operacinių sistemų („Windows“, „Linux“ bei Mac OS) [4]. Egzistuoja keli įrankiai, skirti pasaulinio tinklo programų kūrimui Eclipse platformoje, tačiau jie yra komerciniai, ribotų galimybių ir nelabai paplitę.

2 lentelė. Įrankių, skirtų pasaulinio tinklo programoms kurti, palyginimas penkiabalėje sistemoje.

| Kriterijus \ Vertinimas      | Visual Studio .NET | „Macromedia DreamWeaver“ | „Eclipse“ |
|------------------------------|--------------------|--------------------------|-----------|
| Ar lengvai panaudojamas      | 4                  | 4                        | 4         |
| Vartotojo sąsajos apkrovimas | 4                  | 3                        | 5         |
| Veikia keliose platformose   | 2                  | 4                        | 5         |
| Naudojami resursai           | 3                  | 4                        | 4         |
| Kaina                        | 2                  | 3                        | 5         |

### 5.3. Optimalus architektūros modelis greitam duomenų apdorojimo sąsajos kūrimui

Pasaulinio tinklo programų kūrimo procesas palengvinamas tokiais būdais:

- Naudojant standartines bibliotekas – kuriant pasaulinio tinklo programas, dažnai susiduriama su uždaviniais (pvz., formų validavimo, būsenos išsaugojimo, vartotojų identifikavimo), kuriuos, kuriant naujas sistemas, tenka spręsti iš naujo. Norint išvengti pakartotino kodo rašymo, kuriamos standartinės bibliotekos, skirtos pasaulinio tinklo programoms kurti.
- Naudojant vaizdinius įrankius (IDE) – programų sistemos kūrimo procesą galima pagreitinti naudojant vaizdinius įrankius, veikiančius RAD principu. Naudojant vedlius, automatinį kodo generavimą ir vaizdines programų kūrimo priemones, pasaulinio tinklo programų kūrimo procesas tampa greitesnis ir paprastesnis.

#### 5.3.1. Pasirinktas pasaulinio tinklo programos modelis

Pasaulinio tinklo programos modelis yra orientuotas į greitai kuriamą duomenų apdorojimo sąsają. Remiantis atlikta pasaulinio tinklo programų analize, pasirinktas vartotojo sąsajos, atskirtos nuo veiklos logikos, modelis. Jis pasirinktas todėl, kad vieno sluoksnio architektūra netinka, nes yra sunkiai plečiama, o MVC modelis yra per daug sudėtingas šiam uždaviniui. Todėl pasirinkta tarpinė: modelis – vaizdas (angl. model – view) architektūra. Šiame modelyje visas dėmesys sutelkiamas į programos veiklos logiką. Vartotojo sąsajos dizainui skiriama mažiau dėmesio.

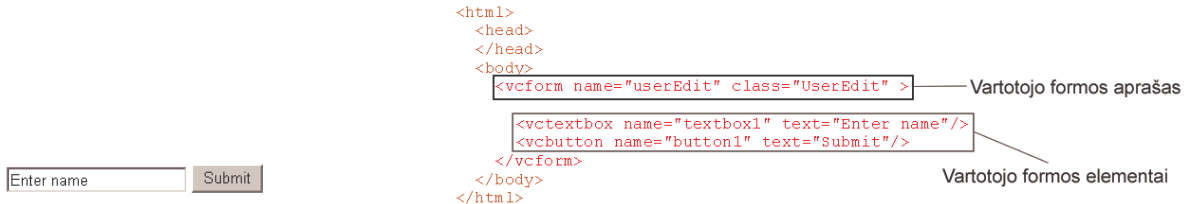
Kuriant vartotojo sąsają, programuotojui neturi rūpėti tokios problemos, kaip formos būsenos išsaugojimas ar atlikto veiksmo apdorojimas (angl. event handling). Šių uždavinių sprendimu turi pasirūpinti vaizdinių komponentų biblioteka. Ji realizuoja vaizdinius komponentus, skirtus vartotojo sąsajai kurti. Vaizdinių komponentų paskirtis – pagreitinti programos kūrimo procesą ir išlaisvinti programuotoją nuo pakartotino kodo rašymo. Vaizdinis komponentas susideda iš dviejų dalių: išvaizdos ir veiklos logikos. Vaizdiniai komponentai skirstomi į standartinius ir vartotojo. Standartiniai komponentai yra: užrašai (angl. label), teksto įvedimo laukeliai (angl. textbox), mygtukai (angl. button), sąrašai (angl. listbox). Vartotojo komponentai – tai HTML kalboje atitiktams neturintys vartotojo sąsajos elementai.

Pavyzdžiui, duomenų tinklelis (angl. DataGrid), kurio nėra tarp standartinių HTML elementų, tačiau jis realizuotas kaip vartotojo komponentas.

Komponentų biblioteką pasirinkta realizuoti PHP technologija, nes PHP yra nemokama, šiuo metu labai populiaru ir plačiai naudojama Lietuvoje ir pasaulyje. Vartotojo sąsają pasirinkta aprašyti HTML kalba, todėl vartotojams, mokantiems HTML, nereikės nieko papildomai mokytis. Vartotojo sąsajos aprašas papildytas žymomis, skirtomis vaizdiniam vartotojo komponentams įterpti. Vartotojo sąsajos aprašas turi būti gerai suformuotas (angl. well formed) XML dokumentas. Taip buvo pasirinkta, kad iš vartotojo sąsajos aprašo būtų galima sugeneruoti jos veiklos logiką aprašančią PHP klasę. Į automatiškai sugeneruotą klasę programuotojui reikės įrašyti tik formos veiklos logiką, formos inicializavimas bus sugeneruotas automatiškai.

### 5.3.1.1. Pasaulinio tinklo programos pavyzdys

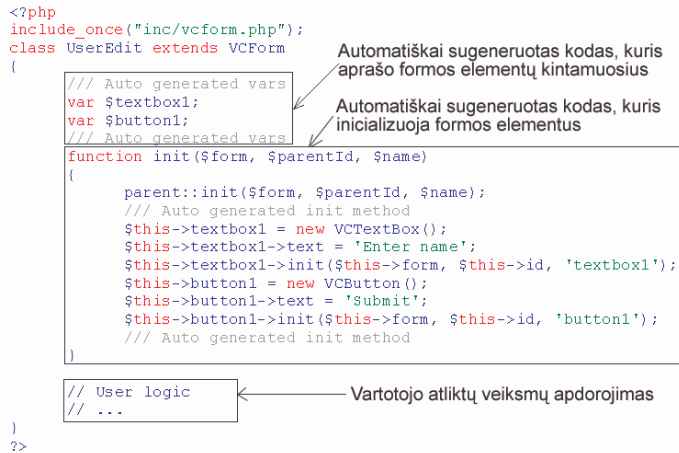
1 paveiksle pavaizduota vartotojo forma sudaryta iš įvedimo laukelio ir patvirtinimo mygtuko. 2 paveiksle pavaizduotas šios formos vartotojo sąsajos aprašymas. Formos elementai rašomi tarp <vcform> žymės. Forma pavadinta userEdit, o jos veiklos logika aprašoma PHP klasėje UserEdit. Pavaizduota forma susideda iš teksto įvedimo laukelio textbox1, kurio reikšmė pagal nutylėjimą: „Enter name“ ir mygtuko formos informacijai patvirtinti button1, kurio tekstas: „Submit“.



1 pav. Vartotojo formos vaizdas.

2 pav. Vartotojo formos sąsajos aprašymas.

Aprašytos vartotojo sąsajos veiklos logiką aprašanti klasė UserEdit pavaizduota 3 paveiksle:





### **3 pav. Automatiškai sugeneruota vartotojo formos logiką aprašanti klasė.**

Vartotojo formos klasė yra paveldėta iš vaizdinių komponentų bibliotekos klasės *VForm*. Ji yra automatiškai generuojama iš vartotojo sąsajos aprašymo.

### **5.3.2. IDE realizuojanti pasirinktą pasaulinio tinklo programos modelį**

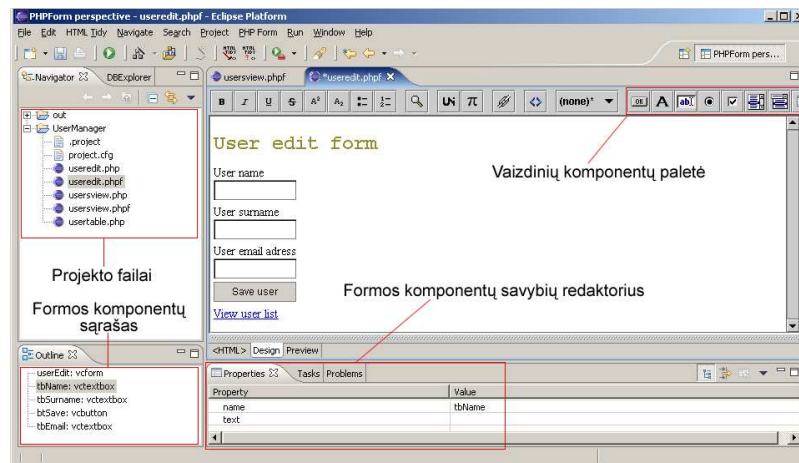
Kuriant programų sistemas, svarbu įvertinti instrumentines priemones. Pavyzdžiui, kuriant Ada programavimo kalbą, projekto tikslas buvo sukurti aukšto lygio programavimo kalbą. Dalis projekto buvo skirtas realizuoti jai skirtą programavimo aplinką (angl. Ada programming environment), tačiau šiai projekto daliai buvo skirta ne tiek daug dėmesio, kiek kuriant pačią programavimo kalbą. Todėl programavimo kalba Ada neturi tokios programavimo sąsajos, kuri galėtų prilygti kitų sistemų aplinkoms pavyzdžiui, Visual Studio. Vartotojui tampa sunkiau pasinaudoti nauja technologija, nes, norint ją įvaldyti, reikia įdėti daugiau pastangų ir laiko [5, 6].

Kuriant šią sistemą, buvo nuspręsta sukurti instrumentines priemones, realizuojančias pasirinkto pasaulinio programų kūrimo modelį. Jo paskirtis – palengvinti vartotojui įsisavinti siūlomą įrankį ir automatizuoti kūrimo procesą. Reikalavimai įrankiui:

- vizualus dizaineris vartotojo formoms kurti;
- automatinis formos veiklos logiką aprašančios klasės generavimas iš formos aprašo;
- PHP ir HTML kodo redaktorius;
- vedlių panaudojimas atliekant sudėtingesnius veiksmus.

### **5.4. Įrankio realizacija Eclipse platformoje**

Įrankio kūrimui pasirinkta Eclipse platforma. Pasirinkimą lėmė tai kad, Eclipse turi visas pagrindines savybes, reikalingas IDE, t.y projektų kūrimas, failų redagavimas, integruota failų versijų kontrolė. Kuriant sistemą nebereikia realizuoti standartinių IDE funkcijų, todėl visas dėmesys skiriamas sukurti pasaulinio tinklo programos modelio realizavimui. Taip pat naudojant Eclipse, orientuojamasi į platų vartotojų ratą, nes ji veikia daugelyje operacinių sistemų. Įrankio realizacijos PHPForm langas pateiktas 4 paveiksle.



4 pav. Vizualaus vartotojo formų dizainerio vaizdas.

## 5.5. Išvados

Straipsnyje apžvelgti pasaulinio tinklo programų modeliai. Taip pat aptartos pasaulinio tinklo programų kūrimo instrumentinės priemonės. Pasiūlytas modelis, palyginti su MVC ir Microsoft ASP.NET, yra paprastesnis ir greičiau realizuojamas. Kadangi vartotojo formos išvaizda ir kodas generuojamas į vykdomą PHP scenarijų, tai vykdomumas (angl. performance) panašus į vieno sluoksnio architektūros programos. Vaizdas yra atskirtas nuo logikos, todėl atlikti pakeitimai neturi įtakos kitos programos vietoms. PHPForm instrumentinės priemonės, palyginti su egzistuojančiomis sistemomis, turi daug pranašumų. Ja galima greitai ir paprastai kurti internetines formas, skirtas duomenims įvesti ir redaguoti. Išmokti dirbti sukurta sistema nėra sudėtinga. Daugelį veiksmų galima atlikti naudojant vedlius arba vizualų dizainerį. Tik formos veiklos logikos redagavimui reikalingas kodo redaktorius. Sukurta sistema nėra universali, o orientuota tik į duomenų apdorojimo formų kūrimą. Todėl ji nėra apkrauta nereikalingomis funkcijomis ir veikia greičiau, nei prieš tai apžvelgtos sistemos. Sukurtą pasaulinio tinklo programos modelį galima tobulinti, plečiant vartotojo komponentų galimybes. Taip pat PHPForm instrumentinės priemonės reikia papildyti komponento įvykio apdorojimo metodo įterpimu. Pasirinkus vaizdinio komponento įvykį, turėtų būti sudaryta galimybė automatiškai įterpti metodą, skirtą to įvykio apdorojimui.

## **6. 2 PRIEDAS. PHPForm sistemos vartotojo dokumentacija**

### **6.1. Sistemos funkcinis aprašymas**

#### **6.1.1. Sistemos paskirtis**

PHPForm programų sistema skirta internetinei duomenų apdorojimo sąsajai kurti. Ji yra orientuota į vartotoją, siekiantį mažiausiomis laiko sąnaudomis sukurti pasaulinio tinklo programą, pritaikytą duomenims apdoroti.

Sukurta sistema susideda iš dviejų dalių:

- komponentų bibliotekos, skirtos internetinei vartotojo sąsajai realizuoti;
- IDE (Integrated Development Environment), kuria galima vizualiai kurti duomenų apdorojimo sąsają, panaudojant komponentų biblioteką.

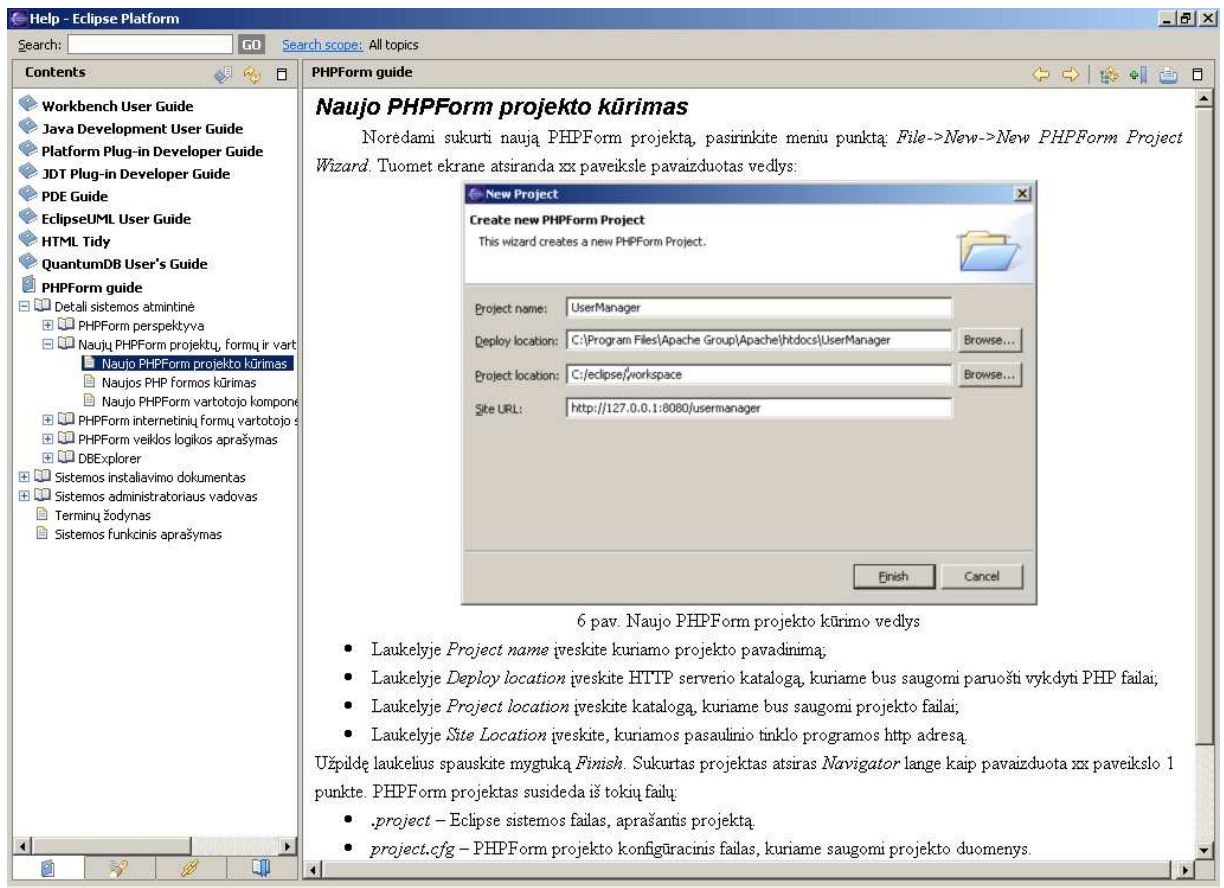
#### **6.1.2. PHPForm programų sistemos galimybės**

Pagrindinės PHPForm funkcijos:

- pasaulinio tinklo programų projektų kūrimas;
- vaizdinis ir tekstinis internetinių formų redagavimas;
- internetinių formų veiklos logikos redagavimas;
- aprašytų internetinių formų sąsajų realizavimas;
- duomenų bazių peržiūrėjimas ir PHP klasių darbui su DB generavimas.

### **6.2. Vartotojo atmintinė**

PHPForm programų sistema turi į Eclipse platformą integruotą vartotojo žinyną. Sistemos vartotojo žinynas atidaromas pasirinkus meniu punktą *Help -> Help Contents*, jis pavaizduotas 1 paveiksle:



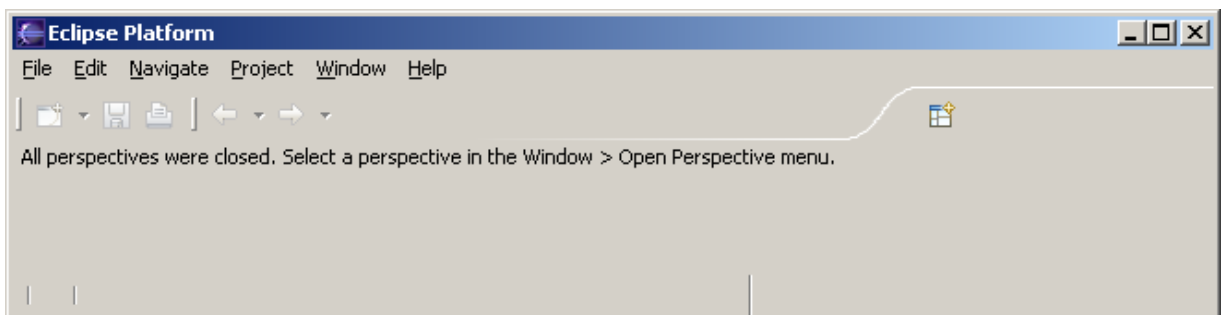
1 pav. PHPForm sistemos žinynas

Kaip naudotis šiuo žinynu skaitykite kartu su Eclipse platforma platinamą dokumentaciją.

## 6.3. Detali sistemos atmintinė

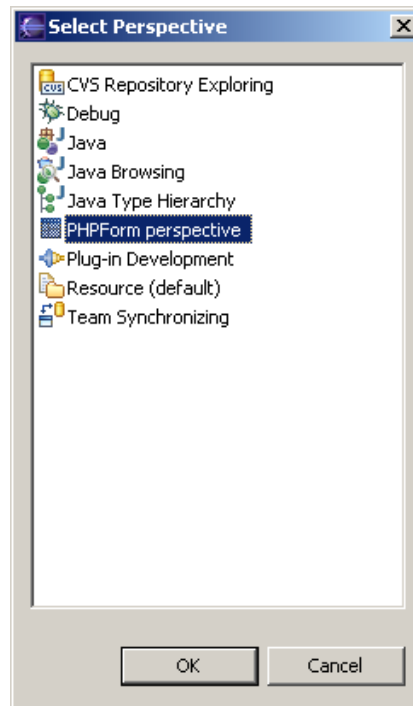
### 6.3.1. PHPForm perspektyva

Kai paleidžiama Eclipse platforma, atsiranda 2 paveiksle pateiktas vaizdas.



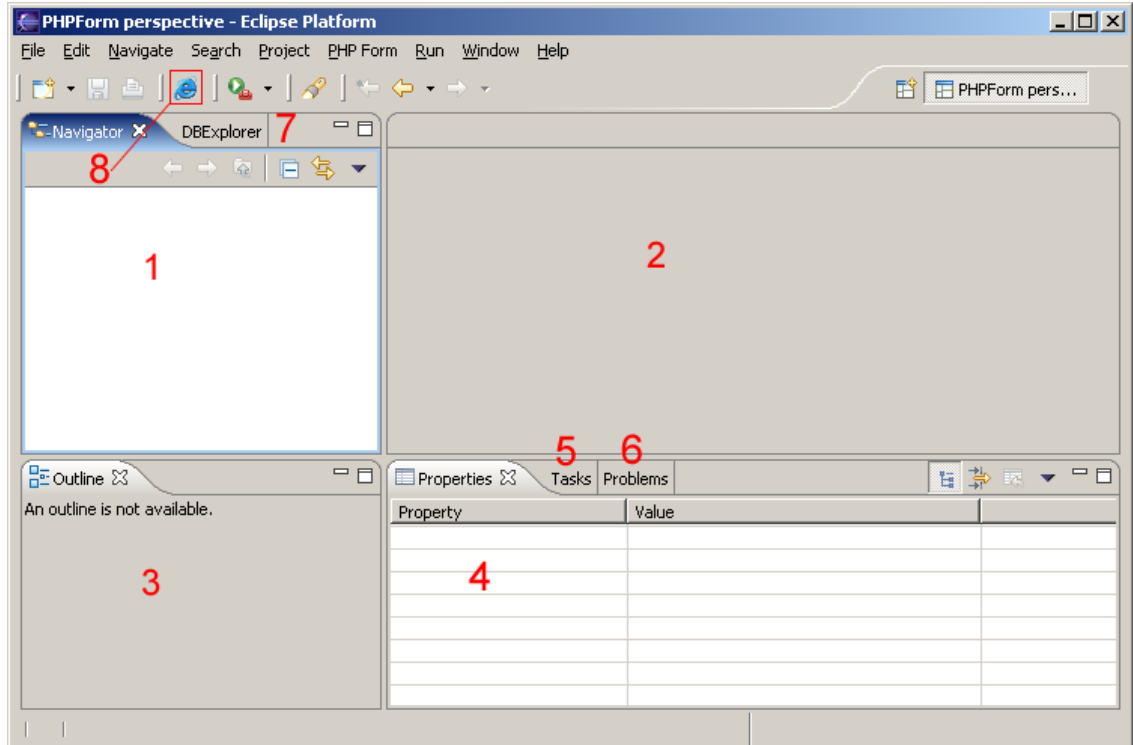
2 pav. Eclipse platformos vaizdas

Pasirinkus meniu punktą *Window->Open Perspective*, 3 paveiksle pateiktas langas:



3 pav. Perspektyvų pasirinkimo langas

Pateiktame perspektyvų sąrašė pasirinkite *PHPForm perspective* ir paspauskite *OK*. Pasirodys 4 paveiksle pateiktas PHPForm perspektyvos vaizdas:



4 pav. PHPForm perspektyvos vaizdas

PHPForm perspektyva susideda iš šių vaizdų:

1. *Navigator* lange pateikiamas projektų ir juos sudarančių failų sąrašas.
2. Vieta HTML, PHP arba kitiems redaktoriams.
3. *Outline* lange pateikiama redaguojamo dokumento santrauka.
4. *Property* lange pateikiamos ir redaguojamos pažymėto perspektyvos elemento savybės.
5. *Tasks* lange pateikiamas užduočių, susijusių su konkrečiu dokumentu arba projektu, sąrašas.
6. *Problems* lange pateikiamas problemų, susijusių su konkrečiu dokumentu arba projektu, sąrašas.
7. *DBExplorer* darbui su duomenų bazėmis;
8. *Navigator* lange pažymėtos PHPForm formos atidarymas interneto naršyklėje.

### 6.3.2. Naujų PHPForm projektų, formų ir vartotojo komponentų kūrimas

Nauji PHPForm projektai, formos ir vartotojo komponentai kuriami naudojant vedlius.

#### 6.3.2.1. Naujo PHPForm projekto kūrimas

Norėdami sukurti naują PHPForm projektą, pasirinkite meniu punktą: *File->New->New PHPForm Project Wizard*. Tuomet ekrane atsiranda 5 paveiksle pavaizduotas vedlys:

5 pav. Naujo PHPForm projekto kūrimo vedlys

- Laukelyje *Project name* įveskite kuriamo projekto pavadinimą;
- Laukelyje *Deploy location* nurodykite HTTP serverio katalogą, kuriame bus saugomi paruošti vykdyti PHP failai;
- Laukelyje *Project location* nurodykite katalogą, kuriame bus saugomi projekto failai;
- Laukelyje *Site Location* įveskite kuriamos pasaulinio tinklo programos interneto adresą.

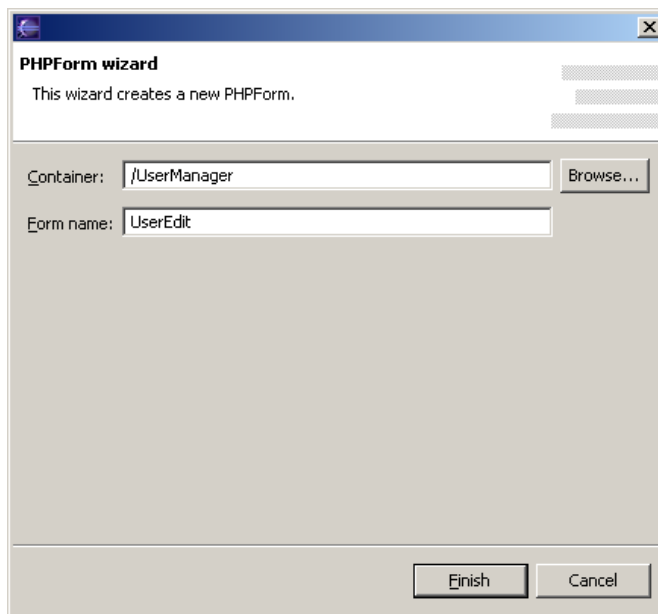
Užpildę laukelius, spauskite mygtuką *Finish*. Sukurtas projektas atsiras *Navigator* lange. PHPForm projektas susideda iš tokių failų:

- *.project* – Eclipse sistemos failas, aprašantis projektą.
- *project.cfg* – PHPForm projekto konfigūracinis failas, kuriame saugomi projekto duomenys.
- *\*.phpf* – Failai su plėtiniu *phpf*, naudojami aprašyti internetinių formų išvaizdą.
- *\*.phpc* – Failai su plėtiniu *phpc*, naudojami aprašyti vartotojo komponentų išvaizdą.
- *\*.php* – Failai su plėtiniu *php*, naudojami aprašyti internetinių formų ir vartotojo komponentų logiką.

**Pastaba:** Naujame projekte automatiškai sukuriamas *phpform1* internetinė forma.

### 6.3.2.2. Naujos PHP formos kūrimas

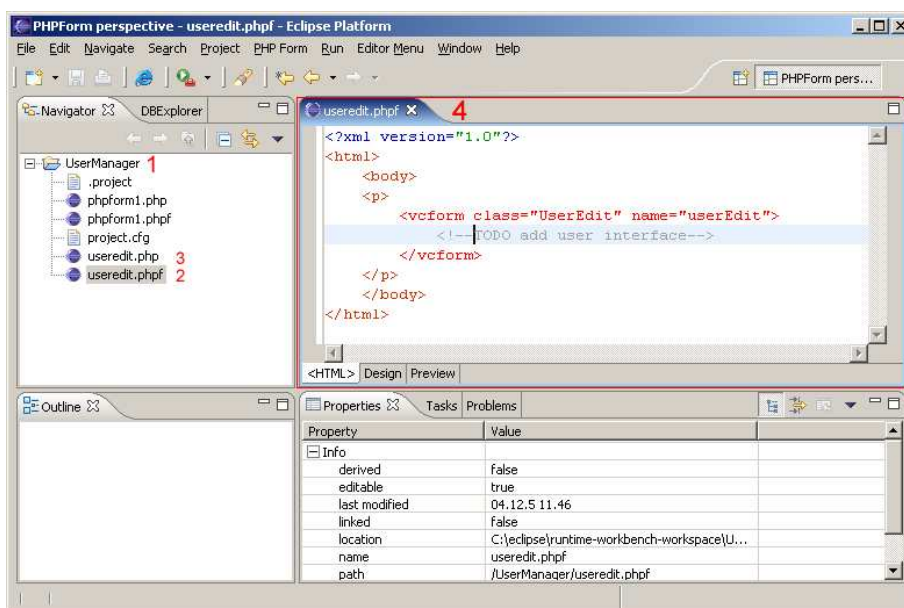
Norėdami sukurti naują internetinę formą, pasirinkite meniu punktą: *File->New->New PHPForm Wizard*. Tuomet ekrane atsiranda 6 paveiksle pavaizduotas vedlys:



6 pav. Naujos internetinės formos kūrimo vedlys

- Laukelyje *Container* nurodykite kuriamos internetinės formos konteinerį. Tai gali būti projektas arba jo katalogas. Jį galite pasirinkti iš sąrašo paspaudę mygtuką *Browse*.
- Laukelyje *Form name* įveskite kuriamos internetinės formos pavadinimą.

Užpildę laukelius, spauskite mygtuką *Finish*. Vedlys nurodytame konteineryje sukurs internetinę formą, kuri atsiras *Navigator* lange, ir ją automatiškai atidarys vartotojo sąsajos redaktoriuje. Sukurta internetinė forma pavaizduota 7 paveiksle:



7 pav. Naujos internetinės formos redaktoriaus vaizdas



1. *Navigator* lange rodomas projektas *UserManger*.
2. Internetinės formos vartotojo sąsajos aprašas *useredit.phpf*.
3. Internetinės formos veiklos logikos aprašas *useredit.php*.
4. Internetinės formos vartotojo sąsajos redaktorius.

### 6.3.2.3. Naujo PHPForm vartotojo komponento kūrimas

Norėdami sukurti naują vartotojo komponentą, pasirinkite meniu punktą: *File->New->New PHPForm User Component Wizard*. Šio vedlio veikla analogiška 3.2.2 skyriuje aprašytam internetinės formos kūrimo vedliui.

PHPForm vartotojo komponentas, kaip ir internetinė forma, susideda iš dviejų dalių:

- vartotojo komponento išvaizdos;
- vartotojo komponento veiklos logikos.

### 6.3.3. PHPForm internetinių formų vartotojo sąsaja

#### 6.3.3.1. PHPForm vartotojo sąsajos aprašymas

PHPForm internetinės formos vartotojo sąsaja aprašoma HTML kalba su išplėstu PHPForm žymų rinkiniu. PHPForm vartotojo sąsajos aprašas verčiamas į vykdomą PHP kodą. Kompiliacijos metu PHPForm žymos keičiamos į HTML žymas. Vietoje vienos PHPForm žymos, aprašančios vaizdinį komponentą, galutiniame vartotojo sąsajos variante atsiranda daugiau HTML žymų, kurios ne tik atvaizduoja PHPForm vaizdinį komponentą, bet ir saugoja papildomą informaciją apie jį. Taip sukuriama galimybė programuotojui nerašant papildomo kodo išsaugoti vaizdinio komponento būseną. Pavyzdžiui, vartotojui užpildžius teksto įvedimo laukelį, komponentų biblioteka automatiškai atsimeina įvestą tekstą.

Vartotojo sąsajos aprašas turi būti gerai suformuotas (well formed) XML dokumentas. Kompiliacijos metu rastos klaidos parodomos 4.6 paveiksle pavaizduotame *Problems* sąrašė. Ištaisius kompiliatoriaus rastas klaidas ir išsaugojus dokumentą, jos išnyksta iš *Problems* sąrašo.

#### 6.3.3.1.1. PHPForm žymų rinkinys

Vartotojo informacijai įvesti ir atvaizduoti naudojamas PHPForm žymų rinkinys:

- *vcform* – PHP formą aprašanti žymė;
- *vcomponent* – vartotojo komponentą aprašanti žymė;

- `vcbutton` – formos patvirtinimo mygtukas;
- `vctextbox` – teksto įvedimo laukelis;
- `vclabel` – tekstinės informacijos vaizdavimui skirtas komponentas;
- `vccheckbox` – vieno arba kelių elementų pasirinkimo mygtukas;
- `vcradiobutton` – vieno iš kelių pasirinkimo mygtukas;
- `vclistbox` – elementų sąrašo komponentas;
- `vccombobox` – vieno elemento pasirinkimo iš kelių komponentas;
- `vclistitem` – sąrašo elemento aprašas;
- `vdatagrid` – duomenų lentelės vaizdavimo komponentas;
- `vccolumn` – duomenų tinklelio stulpelio aprašas.
- `vccommand` – duomenų tinklelio komandos aprašas.

### 6.3.3.1.2. PHPForm vartotojo sąsajos aprašymo pavyzdys

```
<html>
  <head>
  </head>
  <body>
    <vcform name="userEdit" class="UserEdit" >
      <vctextbox name="textbox1" text="Enter name" />
      <vcbutton name="button1" text="Submit" />
    </vcform>
  </body>
</html>
```

1 pvz. PHPForm vartotojo sąsajos aprašymas

Šiame pavyzdyje pateikta PHPForm internetinė forma pavadinimu *userEdit*. Šios formos veiklos logika aprašoma PHP klasėje *UserEdit*. Internetinė forma susideda iš teksto įvedimo laukelio *textbox1*, kurio reikšmė pagal nutylėjimą: „*Enter name*“ ir mygtuko formos informacijai patvirtinti *button1*, kurio tekstas: „*Submit*“. Aprašytos vartotojo sąsajos vaizdas interneto naršyklėje pavaizduotas 8 paveiksle:


 A screenshot of a web browser displaying a simple form. It consists of a single-line text input field with the placeholder text "Enter name" and a rectangular button labeled "Submit" positioned to the right of the input field. The browser's address bar and other interface elements are partially visible at the top and bottom.

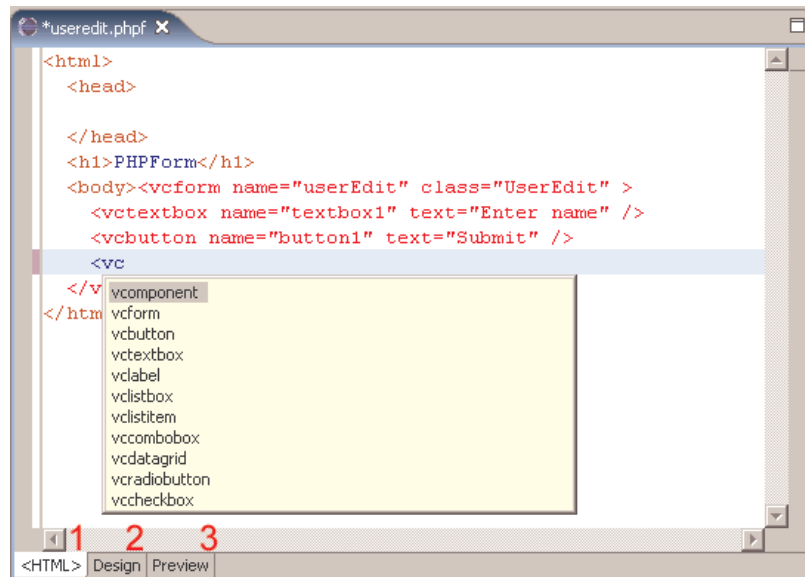
8 pav. Pavyzdyje aprašyta vartotojo sąsaja

### 6.3.3.2. PHPForm internetinių formų vartotojo sąsajos redaktorius

PHPForm internetinių formų vartotojo sąsajos redaktorius susideda iš trijų dalių:

- HTML redaktorius;
- vizualaus dizainerio;
- Preview lango.

Jis pavaizduotas 9 paveiksle:



9 pav. PHPForm internetinių formų vartotojo sąsajos redaktorius

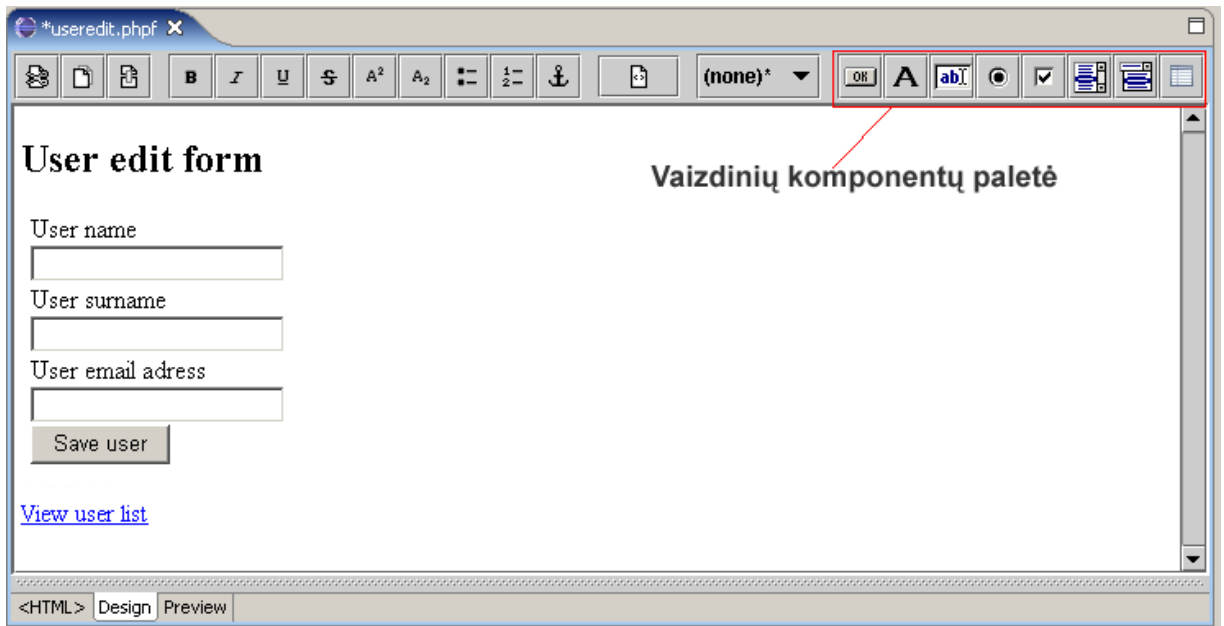
Redagavimo režimą perjungti galima 9 pav. pavaizduotų 1, 2 ir 3 jungikliais.

### 6.3.3.2.1. HTML redaktorius

HTML redaktorius yra tekstinis redaktorius, skirtas failams su plėtiniu *phpf* ir *phpc* redaguoti. Jis vaizduoja HTML dokumento sintaksę skirtingomis spalvomis (syntax highlighting). Taip pat šis redaktorius turi automatišką kodo užbaigimo galimybę (syntax completing). Parašius žymės atidarymo simbolį: “<” arba paspaudus *ctrl + space* klavišų kombinaciją, vartotojui pateikiamas galimas žymų baigčių sąrašas. HTML redaktoriaus vaizdas pateiktas 9 paveiksle.

### 6.3.3.2.2. Vizualus dizaineris

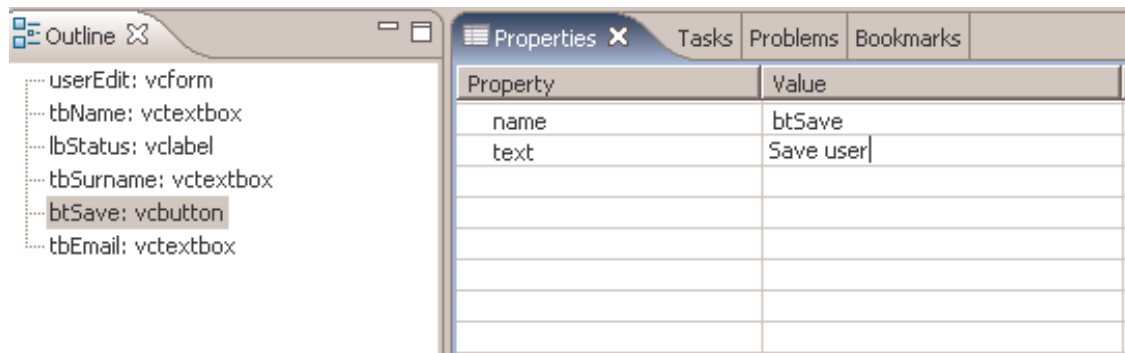
Vartotojo sąsajos redagavimą palengvina grafinis redaktorius. Jame vartotojas mato panašų į galutinį, redaguojamos sąsajos vaizdą, kurį gali redaguoti, kaip pvz.: „Microsoft Word“ dokumentą. Vizualaus dizainerio langas pateiktas 10 pav:



10 pav. Vizualaus dizainerio langas

Pasirinkus norimą vaizdinį komponentą iš vaizdinių komponentų paletės ir ant jo paspaudus pelę, dokumentas bus papildytas pasirinktu vaizdiniu komponentu.

*Outline* lange pateiktas redaguojamos vartotojo sąsajos vaizdinių komponentų sąrašas. *Outline* lange pažymėjus norimą komponentą, jo savybės bus parodytos *Properties* lange, kaip pavaizduota 11 pav.



11 pav. *Outline* ir vaizdinių komponentų savybių redagavimo vaizdas

Pasirinktos komponento savybės reikšmę galima pakeisti įvedus naują reikšmę ir paspaudus *Enter* klavišą.

### 6.3.3.2.3. Vartotojo sąsajos peržiūra

Redaguojant vartotojo sąsają patogu greitai pažiūrėti, kaip formos vaizdas atrodys interneto naršyklėje. Pasirinkus *Preview* režimą, vartotojo sąsaja bus pavaizduota kaip 12 pav. pateiktame vartotojo sąsajos peržiūros lange.



12 pav. Vartotojo sąsajos peržiūros langas

### 6.3.4. PHPForm veiklos logikos aprašymas

PHPForm formos arba vartotojo komponento veiklos logika aprašoma PHP klasėje, kurios bazinė klasė atitinkamai yra VCForm arba VComponent. Ši klasė yra automatiškai sugeneruojama sukūrus naują formą arba vartotojo komponentą. Kaip sukurti naują formą arba vartotojo komponentą, skaitykite 3.2.2 ir 3.2.3 skyriuose.

Papildant vartotojo sąsajos aprašą ir jį išsaugant, veiklos logikos klasė automatiškai atnaujinama. Veiklos logiką aprašanti klasė susideda iš dviejų automatiškai atnaujinamų blokų:

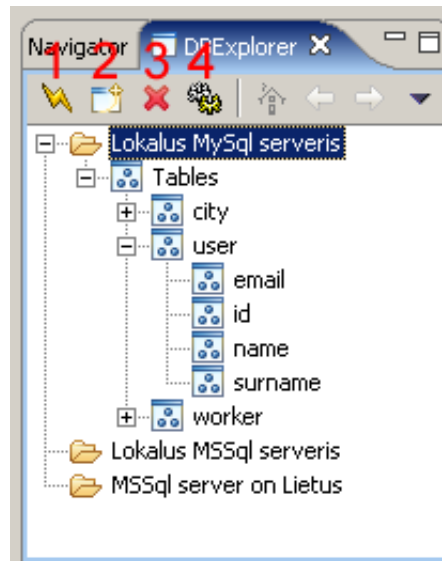
- sąsajos komponentų aprašo, išskiriama: „`/// Auto generated vars`“ tekstu;
- komponentų inicializavimo, išskiriama: „`/// Auto generated init method`“ tekstu.

#### 6.3.4.1. PHPForm formos veiklos logikos aprašymo pavyzdys

Antrame pavyzdyje, pateiktam *UserEdit* formos vartotojo sąsajos aprašymui, automatiškai sugeneruota *UserEdit* klasė, aprašanti šios formos veiklos logiką:

```
<?php
include_once("inc/vcform.php");
class UserEdit extends VCForm
{
    /// Auto generated vars
    var $textbox1;
    var $button1;
    /// Auto generated vars
    function init($form, $parentId, $name)
    {
        parent::init($form, $parentId, $name);
        /// Auto generated init method
        $this->textbox1 = new VCTextBox();
        $this->textbox1->text = 'Enter name';
        $this->textbox1->init($this->form, $this->id, 'textbox1');
        $this->button1 = new VCButton();
    }
}
```





14 pav. DBExplorer langas

Šia naršykle programuotojas gali prisijungti prie skirtingų gamintojų DBVS. Šiuo metu palaikomos Microsoft SQL server ir MySQL DBVS. Prisijungęs prie duomenų bazės, vartotojas gali peržiūrėti jos turinį. Integruotas DBExplorer pagreitina pasaulinio tinklo programų kūrimą, nes nereikia dirbti keliomis skirtingomis programomis vienu metu. Pagrindinis DBExplorer privalumas – automatinis PHP klasių generavimas darbui su nurodyta lentelės struktūra.

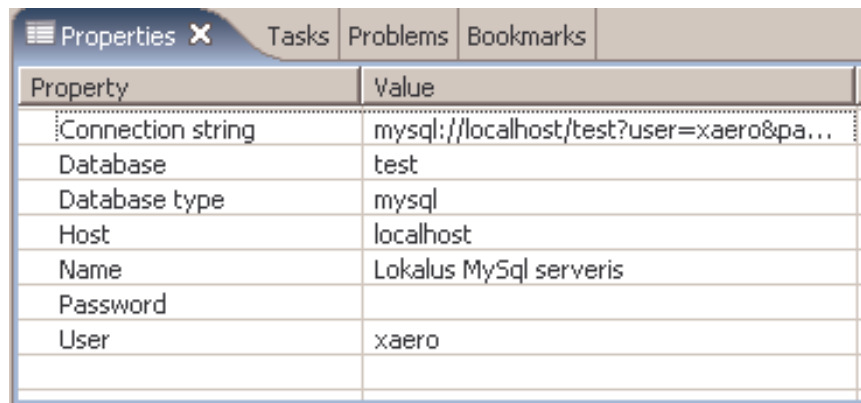
### 6.3.5.1. DB jungties sukūrimas ir pašalinimas

Norėdami sukurti naują DB jungtį, *DBExplorer* lange paspauskite 14 paveiksle 2 pažymėtu mygtuką, užpildykite atsiradusį dialogą ir spauskite *OK* mygtuką. Sukurta DB jungtis atsiras *DBExplorer* lange. Norėdami prisijungti prie duomenų bazės, pasirinkite norimą DB jungtį ir paspauskite 14 paveiksle 1 pažymėtu mygtuką.

Norėdami ištrinti DB jungtį, pažymėkite ją *DBExplorer* lange ir paspauskite 14 paveiksle 3 pažymėtu mygtuką.

### 6.3.5.2. DB jungties redagavimas

Norėdami pakeisti DB jungties parametrus, pažymėkite ją *DBExplorer* lange ir atidarykite *Property* langą, kaip pavaizduota 15 paveiksle:



| Property          | Value                                   |
|-------------------|---|
| Connection string | mysql://localhost/test?user=xaero&pa... |
| Database          | test                                    |
| Database type     | mysql                                   |
| Host              | localhost                               |
| Name              | Lokalus MySql serveris                  |
| Password          |   |
| User              | xaero                                   |

15 pav. DB jungties savybių redagavimo langas

Pasirinkite norimą DB jungties savybę ir, pakeitę nauja reikšmę, spauskite *enter* klavišą.

**Pastaba:** Padaryti pakeitimai bus panaudoti kitą kartą jungiantis prie duomenų bazės.

### 6.3.5.3. PHP klasių generavimas

Norėdami sugeneruoti klasę darbui su *DBExplorer* lange pažymėta lentelė, paspauskite 14 paveiksle 4 pažymėtą mygtuką ir, užpildę atsiradusį vedlį, spauskite *Finish* mygtuką. Sukurta PHP klasė atsiras *Navigator* lange.

## 6.4. Sistemos instaliavimo dokumentas

### 6.4.1. Reikalavimai

Minimalūs reikalavimai kompiuteriui:

- 500 MHz Pentium III procesorius;
- 128 MB operatyviosios atminties;
- 500 MB vietos kietajame diske.

Reikalavimai programinei įrangai:

- Windows 2000, XP arba Linux šeimos OS;
- HTTP serveris (pvz.: Apache 1.3);
- DBVS (MySQL, Microsoft SQL Server);
- Eclipse 3.0 platforma.

### 6.4.2. PHPForm programų sistemos instaliacija

**Išpėjimas:** prieš pradėdami instaliaciją, uždarykite veikiančias Eclipse platformos versijas.

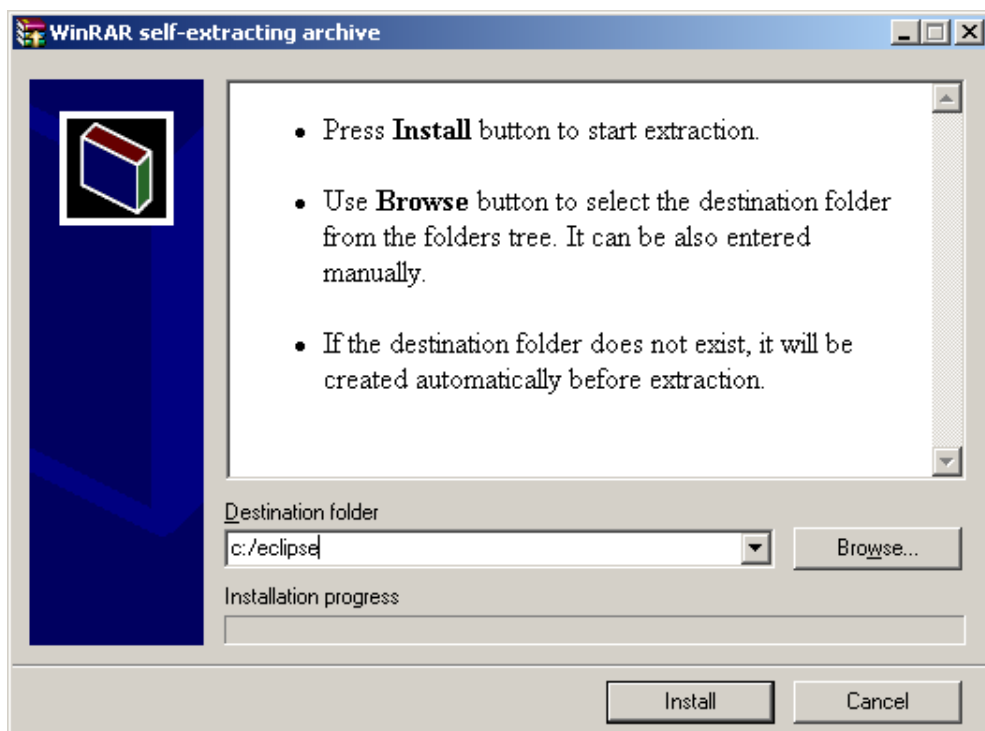
Instaliuoti PHPForm programų sistemą galima keliais būdais:



- naudojant instaliacinį failą (tik Windows OS);
- išarchyvuojant programų sistemą į Eclipse platformos katalogą.

### 6.4.2.1. Instaliacija Windows OS

Instaliacija, skirta Windows OS, pateikiama savaime išsipakuojančiame RAR archyve `phpform.exe`. Jį galima parsisiųsti iš projekto informacinės sistemos adresu: <http://soften.ktu.lt/~s79940/phpform>. Instaliacijos failą nukopijuoti į kompiuterio kietąjį diską. Paleidus vykdyti, pasirodys 16 paveiksle pateiktas vaizdas:



16 pav. Sistemos instaliavimo langas

Laukelyje *destination folder* įveskite visą kelią iki katalogo, kuriame suinstaliuota Eclipse platforma. Pagal nutylėjimą pateikiama reikšmė: “*c:/eclipse*”. Nurodę kelią, spauskite mygtuką *Install* ir PHPForm programų sistema bus instaliuota nurodytame kataloge. Jei norite atšaukti instaliaciją, spauskite mygtuką *Cancel*.

### 6.4.2.2. Instaliacija iš archyvo

Instaliacija pateikiama ZIP archyve: `phpform.zip`. Jį galima parsisiųsti iš projekto informacinės sistemos adresu: <http://soften.ktu.lt/~s79940/phpform>. Šį archyvą reikia išarchyvuoti į šakninį Eclipse platformos katalogą.

Eclipse katalogų struktūra:

Eclipse\ ← šakninis Eclipse platformos katalogas

configuration\

features\

plugins\ ←

Eclipse platformos plug-ins katalogas, į kurį reikia nukopijuoti instaliacinę paketinę PHPForm

readme\

workspace\

### 6.4.3. Instaliacijos patikrinimas

Paleiskite Eclipse platformą. Pasirinkite meniu punktą *Help -> About Eclipse Platform -> Plug-in details*. Turi pasirodyti panašus į 17 paveiksle pateiktą langą, kuriame rodomas suinstaliuotų plug-in sąrašas.

| Provider        | Plug-in Name                 | Version | Plug-in Id                       |
|-----------------|------------------------------|---------|----------------------------------|
| Eclipse.org     | Plug-in Development Envir... | 3.0.1   | org.eclipse.pde.runtime          |
| Eclipse.org     | Plug-in Development UI       | 3.0.1   | org.eclipse.pde.ui               |
| Eclipse.org     | R21 Presentation Plug-in     | 3.0.1   | org.eclipse.ui.presentations...  |
| Eclipse.org     | Refactoring Core             | 3.0.1   | org.eclipse.ltk.core.refactoring |
| Eclipse.org     | Refactoring UI               | 3.0.1   | org.eclipse.ltk.ui.refactoring   |
| Eclipse.org     | Search Support               | 3.0.1   | org.eclipse.search               |
| Eclipse.org     | Standard Widget Toolkit      | 3.0.1   | org.eclipse.swt                  |
| Eclipse.org     | Standard Widget Toolkit f... | 3.0.1   | org.eclipse.swt.win32            |
| Eclipse.org     | Team Support Core            | 3.0.0   | org.eclipse.team.core            |
| Eclipse.org     | Team Support UI              | 3.0.1   | org.eclipse.team.ui              |
| Eclipse.org     | Text                         | 3.0.1   | org.eclipse.text                 |
| Eclipse.org     | Text Editor Framework        | 3.0.1   | org.eclipse.ui.workbench.te...   |
| Eclipse.org     | Tomcat Wrapper               | 4.1.30  | org.eclipse.tomcat               |
| Eclipse.org     | Views                        | 3.0.0   | org.eclipse.ui.views             |
| Eclipse.org     | Win32 fragment               | 3.0.1   | org.eclipse.ui.win32             |
| Eclipse.org     | Workbench                    | 3.0.1   | org.eclipse.ui.workbench         |
| Eclipse.org     | Workbench Compatibility      | 3.0.0   | org.eclipse.ui.workbench.co...   |
| Tomas Stasiūnas | DBExplorer Plug-in           | 1.0.0   | dbexplorer                       |
| Tomas Stasiūnas | Htmleditor Plug-in           | 1.0.0   | htmleditor                       |
| Tomas Stasiūnas | Phpeditor Plug-in            | 1.0.0   | phpeditor                        |
| Tomas Stasiūnas | PHPForm Plug-in              | 1.0.0   | phpform                          |

17 pav. Suinstaliuotų plug-in sąrašas

PHPForm programų sistema susideda iš keturių plug-in:

- Dbexplorer;
- Htmleditor;
- Phpeditor;

- Phpform.

Jeigu šių plug-in nėra pateiktame sąrašė, įsitikinkite, ar tikrai instaliacijos metu gerai nurodėte Eclipse platformos kelią.

#### 6.4.4. PHPForm programų sistemos pašalinimas

**Įspėjimas:** prieš pradėdami PHPForm programų sistemos pašalinimą, uždarykite veikiančias Eclipse platformos versijas.

Atidarykite Eclipse platformos plugins katalogą ir ištrinkite iš jo šiuos katalogus:

- dbexplorer\_x.x.x
- htmleditor\_x.x.x
- phpeditor\_x.x.x
- phpform\_x.x.x

### 6.5. Sistemos administratoriaus vadovas

PHPForm programų sistema tiesiogiai bendrauja su HTTP ir DBVS sistemomis.

#### 6.5.1. Programų sistemos sąveika su HTTP serveriu

PHPForm programų sistema kompiliacijos metu kopijuoja paruoštus vykdyti PHP failus į nurodytą HTTP serverio dokumentų katalogą. Programų sistema gali nukopijuoti failus tik į lokalų arba tinkle esančio kompiuterio katalogą, priėjimas per FTP serverį yra negalimas.

#### 6.5.2. Programų sistemos sąveika su DBVS serveriu

Šiuo metu PHPForm programų sistema dirba tik su MySQL ir MS SQL server DBVS, vėliau palaikomų DBVS sąrašas bus papildytas. Eclipse platformoje prie duomenų bazės jungiamasi naudojant JDBC. Vaizdinių komponentų bibliotekos realizacijoje į duomenų bazę kreipiamasi naudojant standartines PHP bibliotekas. Susidūrę su problemomis dirbdami su nurodytomis DBVS naudokitės su jomis platinama dokumentacija.