

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

Brigita Bagdonaitė

Meno draugijos informacinė sistema

Magistro darbas

**Vadovas
doc. L. Nemuraitė**

KAUNAS, 2005

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

Meno draugijos informacinė sistema

Informacinių technologijų magistro baigiamasis darbas

**Kalbos konsultantė
dr. J. Mikelionienė**

**Vadovas
doc.L.Nemuraitė**

Recenzentas

**Atliko
IFM 9/3 gr. stud.
B.Bagdonaitė**

KAUNAS, 2005

TURINYS

1. ĮVADAS.....	8
2. MENO DRAUGIJOS INFORMACINĖS SISTEMOS KŪRIMO PAGRINDIMAS.....	11
2.1. VARTOTOJŲ POREIKIŲ IR ESAMŲ SISTEMŲ ANALIZĖ	11
2.2. DARBO TIKSLAS IR UŽDAVINIAI	14
2.3. PANAUDOJIMO ATVEJŲ MODELIS	14
2.4. NEFUNKCINIAI SISTEMOS REIKALAVIMAI	15
2.5. SISTEMOS REALIZAVIMO PRIEMONIŲ PASIRINKIMAS	16
2.5.1. <i>MS VisualStudio.NET</i>	16
2.5.2. <i>MS.NET Framework</i>	16
2.5.3. <i>Visual C#</i>	17
2.5.4. <i>ASP.NET</i>	17
2.5.5. <i>ADO.NET</i>	17
2.5.6. <i>MS SQL Server 2000</i>	18
2.6. ANALIZĖS IŠVADOS.....	18
3. MENO DRAUGIJOS SISTEMOS MODELIS IR FORMALI SPECIFIKACIJA	20
3.1. SISTEMOS REIKALAVIMŲ SPECIFIKACIJA	20
3.2. FORMALI SPECIFIKACIJA Ž KALBA	30
3.3. VARTOTOJŲ SAŠAJOS MODELIS	39
3.4. KLASIŲ APRAŠYMAS	40
3.5. DUOMENŲ BAZĖS SCHEMA	46
3.6. PROGRAMINĖS ĮRANGOS DIEGIMO ARCHITEKTŪRA.....	49
3.7. SISTEMOS REALIZACIJA IR VEIKIMO ANALIZĖ	50
4. VARTOTOJO DOKUMENTACIJA.....	62
4.1 SISTEMOS FUNKCINIS APRAŠYMAS	63
4.2 SISTEMOS VADOVAS	63
4.3 SISTEMOS INSTALIAVIMO DOKUMENTAS.....	64
5. PRODUKTO KOKYBĖS ĮVERTINIMAS.....	65
6. IŠVADOS.....	67
7. LITERATŪRA.....	68
8. PRIEDAI	69

9.1 PRIEDAS. STRAIPSNIS	69
IŠVADOS.....	73
LITERATŪROS SĀRAŠAS.....	73
9.2 PRIEDAS. DB SCHEMOS SPECIFIKACIJOS MS VISIO ATASKAITA.....	74

SUMMARY

In this work, the Internet Information System is created for the Artists' Association. It was planned to meet the specific needs of the Artists who wanted to let the people know about themselves and their works, and finally sell them. It also meets the needs of every Internet user who might be interested in information about the popular or perspective artists and who might purchase their works. The system was developed on the basis of .NET technologies: ADO.NET, ASP.NET and programmed with Visual C#. The data of the information system are stored in MS SQL Server 2000.

Finally the information system was implemented and thoroughly tested. It meets all the functional and non-functional requirements of end users: the Administrator, the Artist, the Registered User and Internet User. The system is safe, fast, and comfortable to use, has good look, and may be extended in future when the needs of its users would grow.

Lentelių sąrašas

1 LENTELĖ. MENO DRAUGIJŲ INFORMACINIŲ SISTEMŲ PALYGINIMAS	12
2 LENTELĖ. MENO DRAUGIJOS INFORMACINĖS SISTEMOS VARTOTOJAI	13
3 LENTELĖ. PA „PERŽIŪRĖTI INFORMACIJĄ APIE MENO KŪRINIUS, AUTORIUS, PARODAS“ SPECIFIKACIJA	20
4 LENTELĖ. PA „ATLIKTI PAIEŠKĄ“ SPECIFIKACIJA.....	21
5 LENTELĖ. PA „REGISTRUOTI NAUJĄ REGISTRUOTĄ VARTOTOJĄ“ SPECIFIKACIJA.....	23
6 LENTELĖ. PA „PRISIJUNGTI“ SPECIFIKACIJA	24
7 LENTELĖ. PA „UŽSISAKYTI MENO KŪRINIUS“ SPECIFIKACIJA.....	26
8 LENTELĖ. PA „ĮVESTI IR KOREGUOTI INFORMACIJĄ APIE MENO KŪRINIUS“ SPECIFIKACIJA	26
9 LENTELĖ. PA „ĮVESTI IR KOREGUOTI INFORMACIJĄ APIE PARODAS“ SPECIFIKACIJA.....	27
10 LENTELĖ. PA „TVARKYTI VARTOTOJŲ INFORMACIJĄ“ SPECIFIKACIJA.....	28
11 LENTELĖ. PA „TIKRINTI TEISES“ SPECIFIKACIJA.....	29
12 LENTELĖ. SISTEMOS KLASIŲ APRAŠAS.....	40
13 LENTELĖ. OPERACINĖS SISTEMOS.....	64
14 LENTELĖ. REIKALINGA ATMINTIS	64
15 LENTELĖ. PŪ KOKYBĖS ĮVERTINIMAS.....	65

Paveikslukų sąrašas

1 PAV. MENO DRAUGIJOS INFORMACINĖS SISTEMOS PANAUDOJIMO ATVEJŲ MODELIS....	15
2 PAV. VARTOTOJŲ SAŠAJOS MODELIS.....	40
3 PAV. DUOMENŲ BAZĖS STRUKTŪRA MS VISIO APLINKOJE.....	48
4 PAV. PROGRAMINĖS ĮRANGOS DIEGIMO ARCHITEKTŪRA.....	49
5 PAV. PRISIJUNGIMO PUSLAPIS	50
6 PAV. REGISTRACIJOS LANGAS	51
7 PAV. DEMONSTRACINĖ VERSIJA.....	52

8 PAV. ADMINISTRATORIAUS APLINKA.....	53
9 PAV. VISŲ VARTOTOJŲ PUSLAPIS.....	54
10 PAV. VISŲ MENININKŲ PUSLAPIS.....	55
11 PAV. VISŲ PAVEIKSLŲ PUSLAPIS.....	56
12 PAV. VISŲ UŽSAKYMŲ PUSLAPIS.....	56
13 PAV. VISŲ PARODŲ PUSLAPIS.....	56
14 PAV. MENININKO APLINKA.....	57
15 PAV. MENININKO PAVEIKSLO ĮKĖLIMO PUSLAPIS	58
16 PAV. MENININKO PARODŲ PUSLAPIS.....	58
17 PAV. REGISTRUOTO VARTOTOJO PUSLAPIS.....	59
18 PAV. REGISTRUOTO VARTOTOJO PAVEIKSLŲ PERŽIŪROS PUSLAPIS.....	60
19 PAV. REGISTRUOTO VARTOTOJO PUSLAPIS.....	61
20 PAV. REGISTRUOTO VARTOTOJO PARODŲ PUSLAPIS.....	62

1. ĮVADAS

Šiuolaikiniam žmogui yra labai svarbu laiku ir greitai susirasti visą jam reikiamą informaciją. Jis neturi laiko naršyti tarp netvarkingų, padrikų duomenų. Jis nori, kad viskas būtų susisteminta, aišku ir pateikta jam priimtina forma.

Toks yra pagrindinis apibendrintas reikalavimas visoms informacinėms sistemoms. Kai interneto vartotojui prireikia specifinės informacijos ar paslaugos, jis ieško sistemos, kuri atitiktų jo poreikius. Interneto paslaugų skalė labai plati, tačiau kai kurie reikalavimai yra bendri: patogumas, greitis, saugumas, grįžtamoji reakcija, laukiamo rezultato gavimas.

Šio darbo tyrimo sritis yra interneto sistemų kūrimas, o konkretus tyrimo objektas – Meno draugijos informacinė sistema. Tiek teorinė analizė, tiek praktiniai rezultatai, pateikti šiame darbe, nagrinėja konkrečius ir specifinius meno draugijos informacinės sistemos vartotojų poreikius. Darbo pradžioje atliktas teorinis tyrimas išskiria konkrečius tokios informacinės sistemos vartotojus, detalus informacinės sistemos panaudojimų atvejų įvardinimas ir analizė atvaizduoja, ko konkrečiai vartotojas nori iš sistemos bei kas vyksta sistemoje, kai vartotojas imasi atitinkamos veiklos. Įvardinus sistemoje vykstančius procesus natūralia kalba, pateikiama Z specifikacija, kuri detaliai atvaizduoja, ką kuriama sistema atliks ir nors neatskleidžia, kaip tai bus atlikta, vis tik yra svarbi informacinės sistemos kūrimo dalis, kadangi įgalina sumažinti galimų klaidų sistemoje skaičių bei išvengti nesusipratimų ir dviprasmybių, kurie gali kilti dėl skirtingų interpretacijų ar reikšmingų, tačiau „savaiame suprantamų“ faktų praleidimo. Vėliau yra smulkiai aprašomos technologijos, kurių pagalba sistema realizuojama. Tai naujos technologijos, kurias siūlo MS VisualStudio.NET paketas. Jos suteikia informacinės sistemos programavimui didesnes galimybes, o sistemos vartotojams komforto jausmą naudojantis sistema.

Meno draugijos informacinė sistema yra daugiau informacinio nei komercinio pobūdžio, tačiau sistemos vartotojai gali užsisakyti jiems patikusį kūrinį. Tokiu būdu yra sudaromos patogios sąlygos vartotojams ne tik susipažinti su įvairiais meno draugijai priklausančiais menininkais ir jų kūriniais, tačiau ir įsigyti jų darbų nesitraukiant iš namų ar darbo vietos.

Meno draugijos informacinei sistemai labai svarbus ir dizainas, kadangi estetiškai pateikta informacija yra patrauklesnė, labiau sudomina. Sistema skirta meno žmonėms, todėl jų estetinis poreikis nepriekaištingam dizainui yra santykinai svarbesnis nei kitose informacinėse sistemose. Meno draugijos atveju gražus dizainas ir laukiamų paslaugų suteikimas – du svarbiausi žmogiškieji

faktoriai, kuriuos svarbu įvertinti, kad vartotojai, kartą užsukę į sistemą, sugrįžtų ir taptų tikrais jos nariais.

Kadangi daug informacinių sistemų yra sukurta, gali iškilti natūralus klausimas, kam reikalinga dar viena. Pirmiausia, Lietuvos vartotojams reikia sistemos, kuri atspindėtų jų specifinius kultūros poreikius, teiktų informaciją apie šalia gyvenančius menininkus, leistų menininkams patiems tvarkyti savo informacinius puslapius. Kitaip tariant, reikalinga sistema tam tikrai vietinei socialinei bendruomenei, kurią vienija potraukis menui.

Antra, dabartinis visuomenės narių užimtumas ir gyvenimo tempas dažnai neleidžia skirti pakankamai laiko kultūriniais poreikiams. Sukurtoji sistema turėtų bent dalinai užpildyti šią spragą ir leisti patekti į meno pasaulį didesniai visuomenės narių skaičiui kad ir trumpos pietų pertraukos metu.

Meno sistemos, skirtos užsienio vartotojams, yra gerai apgalvotos ir suprojektuotos, dažnai atitinka daugelį poreikių, tačiau Lietuvoje paprastai yra apsiribojama tiesiog interneto puslapiu. Be abejo, interneto puslapis gali suteikti labai daug informacijos, tačiau jis neteikia paslaugų, kurių reikia galutiniam vartotojui.

Šiame darbe sukurta sistema yra kol kas tik prototipas. Praktikoje įdiegtos tokios sistemos sėkmė priklauso ne tik nuo gerai apgalvotos programinės įrangos, bet ir nuo ją palaikančios žmonių grupės, kuri užpildo tokią sistemą informaciniu turiniu. Naudojama tokia sistema turėtų būti plečiama, atsižvelgiant į naujai atsiradusius poreikius. Buvo siekiama sukurti tokią įrangą, kuri būtų nesunkiai modifikuojama ir greitai nepasentų, todėl buvo pasirinktos pažangiausios technologijos, sukurta nesudėtinga ir patogi sąsaja.

Darbą rašant buvo nagrinėjami tokių autorių darbai kaip Ivaro Jacobsono[2], kurio sukurti panaudojimo atvejai labai paplito ir puikiai tinka sistemų modeliavimui. J. Spivey darbas suteikė informacijos kuriant Z psecifikaciją[5]. V. Sekliuckio, S.Gudo, G. Garšvos vadovėlis pravertė kuriant duomenų bazę su MS Visio[4]. Microsoft puslapis apie Visual Studio.NET buvo pagrindis informacijos apie .NET technologijas šaltinis[3].

Pagrindinė dėstomoji dalis susideda iš trijų dalių: 1. Meno draugijos informacinės sistemos kūrimo pagrindimo, 2. Meno draugijos sistemos modelio ir formalios specifikacijos, 3. Vartotojo dokumentacijos. Pirmoji dalis prasideda nuo vartotojų poreikių ir esamų sistemų analizės. Vėliau suformuluojamas darbo tikslas bei uždaviniai, sudaromas panaudojimo atvejų modelis, nustatomi

nefunkciniai sistemos reikalavimai, išvardinamos numatomos naudoti technologijos: MS Visual Studio.NET, MS.NET Framework, Visual C#, ASP.NET, ADO.NET, MS SQL Server 2000.

Antroji projektinė dalis susideda iš sistemos reikalavimų specifikacijos, formalios specifikacijos Z kalba, vartotojų sąsajos modelio, esybių ryšių diagramos, programinės įrangos diegimo architektūros, sistemos realizacijos bei jos testavimo.

Trečiojoje dalyje aprašoma vartotojo dokumentacija bei produkto kokybės įvertinimas.

Šio darbo tema buvo parengtas straipsnis, išspausdintas 10-os tarpuniversitetinės magistrantų ir doktorantų konferencijos leidinyje[1].

2. MENO DRAUGIJOS INFORMACINĖS SISTEMOS KŪRIMO PAGRINDIMAS

Šiame skyriuje aptariama probleminė sritis – interneto sistemų kūrimas ir tyrimo objektas – Meno draugijos informacinė sistema. Nagrinėjamas uždavinys turi svarbią kultūrinę ir socialinę reikšmę, jis skirtas vaizdinės ir pažintinės informacijos skleidimui bei žmogaus kultūrinių poreikių patenkinimui. Dabartinis visuomenės narių užimtumas ir gyvenimo tempas dažnai neleidžia skirti pakankamai laiko kultūriniais poreikiams. Sukurtoji sistema turėtų bent dalinai užpildyti šią spragą ir leisti patekti į meno pasaulį kad ir pietų pertraukos metu. Lietuvoje tokių sistemų nėra daug ir jos neteikia paslaugų, kurios galėtų tenkinti meno vartotojų bendrijos lūkesčius.

2.1. Vartotojų poreikių ir esamų sistemų analizė

Informacinės sistemos naudojamumas priklauso nuo to, ar bus atsižvelgta į visų ir visus vartotojų reikalavimus. Tarkim darant internetinį meno draugijos puslapį, geriausiu atveju bus išklausomi draugijai priklausančių žmonių reikalavimai ir pageidavimai sistemai. Tačiau juk prieiga per internetą yra skirta ne pačiai draugijos administracijai ar jos nariams, o bet kuriam vartotojui, kurį tikimasi sudominti menu ar draugijos nariais. Tarkim kad internetinis puslapis gali patenkinti bet kurio vartotojo informacinius poreikius, bet šiandieninis interneto vartotojas yra daug labiau išlepintas: jis gali internete užsisakyti picą, knygas, įvairias prekes ir paslaugas neišeidamas iš namų, netgi nesivargindamas pakelti telefono ragelio. Radęs internete kažką tinkamo, jis čia pat užsisako ir su juo susisiekiama bei prekė jam pristatoma. Jei toks žmogus susidomės meno kūriniais interneto puslapyje, kur vienintelė galimybė įsigyti tai, kas jam patiko, yra skambintis į draugiją duotais telefonais ar ieškotis konkretaus menininko kitais puslapyje pateiktais būdais, jis gali tiesiog atidėti užsakymą vėlesniam patogesniam laikui, o galiausiai net ir pamiršti apie savo sumanymą. Tokiu būdu interneto vartotojo dėmesį labai lengva prarasti, o norint jį išsaugoti tenka mąstyti iš tokio vartotojo pozicijų ir išskirti tokius jo reikalavimus, kurie garantuotų pilną vartotojo komfortą bei jo dėmesį.

Tačiau reiktų pradėti nuo to, ką šiai dienai siūlo užsienio ir Lietuvos meno draugijos kaip informacines sistemas savo nariams bei interneto vartotojams. Toliau pateikiama lentelė, kurioje atvaizduoti įvairių meno informacinių sistemų privalumai bei trūkumai. Šiais trumpiniais bus įvardinamos tokios meno draugijos: EAAA – Stebuklinių kūrinių meno draugija (*Enchanted Artworks Artist's Association*), IAA - Tarptautinė Menininkų Asociacija (*International Artist*

Association), AA – Absoliutaus meno draugija (*Absolute Arts*), LFS – Lietuvos Fotomenininkų Sąjunga, MMD – mano siūloma meno draugijos informacinė sistema.

1 lentelė. Meno draugijų informacinių sistemų palyginimas

	EAAA	IAA	AA	LFS	MMD
Išsami informacija	Taip	Taip	Taip	Taip	Taip
Kontaktai	Taip	Ne	Taip	Taip	Taip
Vartotojų registracija internete	Taip	Taip	Taip	Ne	Taip
Paveikslų galerija	Taip	Taip	Taip	Taip	Taip
Meniškas dizainas	Taip	Taip	Taip	Taip	Taip
Darbų užsakymas internete	Taip	Ne	Taip	Ne	Taip
Prisijungimas prie sistemos tik menininkams	Taip	Ne	Ne	Taip	Ne
Paieška	Taip	Taip	Taip	Ne	Taip

Atlikta analizė, kurios rezultatai dalinai atsispindi lentelėje, parodė, kad užsienio meno draugijos veikia aktyviau ar bent jau labiau rūpinasi savo įvaizdžiu, nariais bei interneto vartotojais. Jos plačiau naudoja naujas technologijas bei labiau analizuoja vartotojų poreikius, savo prezentacinius puslapius nuolat atnaujina ir tobulina, prisitaikydamos prie situacijos ir galimybių. Tarkim, registracija internete yra suprantamas dalykas, kai kurių draugijų netgi skatinamas įvairiais prisiregistravusių konkursais, kurių prizai meno kūriniai ir pan., o Lietuvos Fotomenininkų Sąjunga nevykdo registracijos internete. Fotomenininkai gali prisiregistruoti ir prisijungti prie sistemos su iš anksto turimais prisijungimo vardais bei slaptažodžiais, matyt, suteikiamais įstojus į draugiją. Interneto vartotojai nėra skatinami dalyvauti draugijos veikloje, jiems nėra suteikiama galimybė įsigyti patikusią fotografiją čia pat internete. Tarptautinė Menininkų Asociacija ir Lietuvos Fotomenininkų Sąjunga neleidžia meno kūrinių užsisakyti internetu. Tarptautinė Menininkų Asociacija pateikia daugybę menininkų darbų, tačiau dėl aiškiai informacinio pobūdžio komercija neužsiima. Idealiausiai atitinkančios įvairius reikalavimus yra Stebuklinių kūrinių meno draugija ir

Absoliutaus Meno Draugija. Jos abi yra tiek informacinio pobūdžio, tiek ir komercinio, viskas yra sutvarkyta vartotojo patogumui ir komfortui.

Mano siūlomas meno draugijos informacinės sistemos projektas yra sutvarkytas taip, kad meno draugijos veikloje aktyviai galėtų dalyvauti ne tik patys menininkai, bet ir interneto vartotojai. Šioje sistemoje vartotojas ne tik susipažins su meno draugijos nariais, jų biografijom, pasiekimais, kūriniais, buvusiomis ir netolimoj ateity rengiamomis parodomis, bet ir leis užsisakyti jam patikusį kūrinį. Registracija bus vykdoma čia pat internete, tokiu būdu interneto vartotojas sutaupys laiko ir iškart galės prisijungti prie sistemos. .NET technologijos leis vartotojui saugiai ir produktyviai naršyti informacinėje sistemoje.

Tam kad būtų galima kalbėti apie specifinius konkrečių vartotojų reikalavimus, reikia pradėti nuo pačių vartotojų įvardinimo. Toliau pateiktoje lentelėje vartotojai ne tik įvardinami, bet ir šiek tiek plačiau aptariami:

2 lentelė. Meno draugijos informacinės sistemos vartotojai

Interneto vartotojas	Tai bet koks vartotojas, kuris naršydamas internete galėtų rasti informacinę sistemą.
Registruoti vartotojai	Žmonės, kurie mėgsta meną, domisi menininkais, jų parodomis, kūriniais. Informacinė sistema yra daroma tam, kad aprūpintų juos visa reikalinga informacija per kuo trumpesnę laiką bei suteiktų galimybę užsisakyti patikusį kūrinį.
Menininkai	Tai draugijos nariai, kurie talpina sistemoje savo biografijos faktus, kūrinčius, įvairią informaciją susijusią su jų kūrybine veikla.
Meno draugija (Administracija)	Tai draugija, kurios nariai yra menininkai – žmonės, kurie visą savo laiką arba dalį jo skiria tapybai, skulptūrai ir pan. Menininkų draugijos tikslas yra suburti kuo daugiau menininkų bei reklamuoti juos ir jų

	darbus, suteikti pilną informaciją potencialiems klientams – registruotiems vartotojams. Informacinėje sistemoje jos pareigos apsiriboja sistemos priežiūra.
--	--

Vartotojai jau įvardinti telieka išskirti ir apibrėžti jų reikalavimus sistemai. Reikalavimai bus apibrėžti naudojant Jacobsono pasiūlytą būdą - panaudojimo atvejų modelį bei patikslinami Z specifikacijos.

2.2. Darbo tikslas ir uždaviniai

Darbo tikslas yra sukurti internete veikiančią Meno draugijos informacinę sistemą, kuri: kiekvienam interneto vartotojui leistų peržiūrėti informaciją apie menininkus, jų kūrinis, dalyvavimą parodose, atlikti paiešką, registruotis ir tapti sistemos vartotoju; menininkui – įvesti ir koreguoti informaciją apie save ir savo kūrybą; registruotam vartotojui – užsisakyti patikusius kūrinius; administratoriui (Meno draugijos atstovui) – tvarkyti informaciją, suteikti teises, prižiūrėti sistemą.

Prisijungę vartotojai galės atlikti tik jiems leidžiamus veiksmus, kuriuos nusako jų teisės.

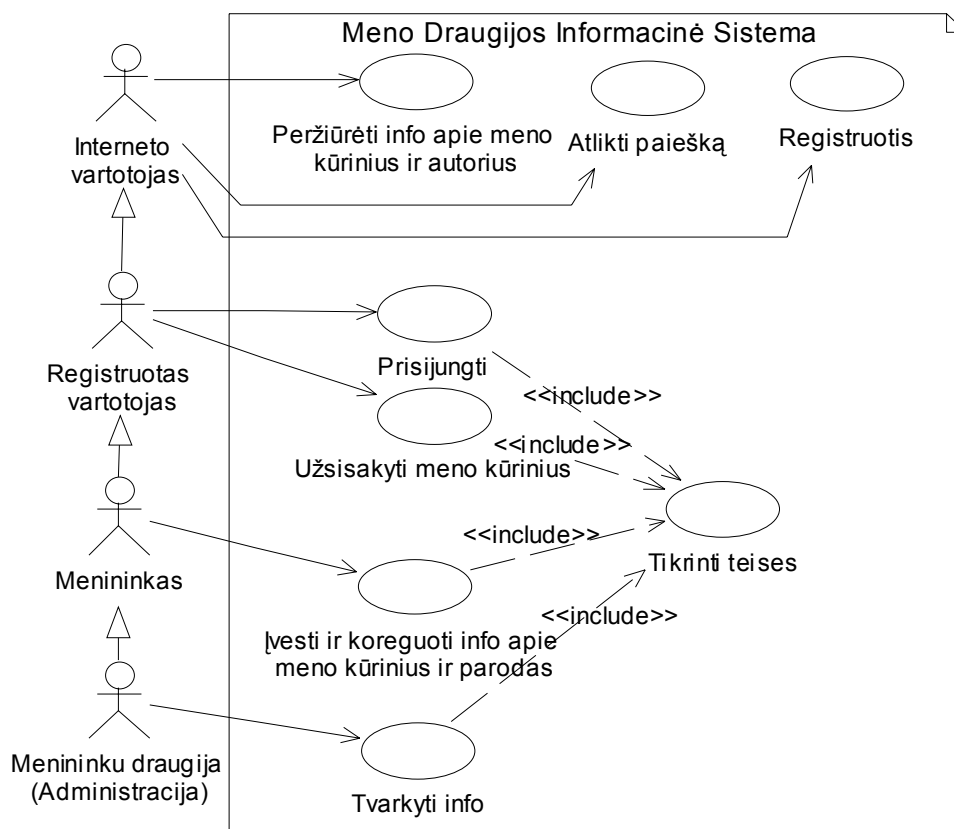
Siekiamą sistemą turi būti:

- Patogi naudotis (lengva navigacija, patogiai išdėliota ir susisteminta informacija);
- Funkcionali (vykdyti tai, ko iš jos tikisi vartotojas);
- Estetiška (vartotojas turi džiaugtis ne tik komfortu, bet ir maloniais akiai vaizdais, atitinkančiais sistemos tematiką);
- Moderni (padaryta naujausių technologijų pagrindu, šiuo atveju MS.NET technologijomis).

2.3. Panaudojimo atvejų modelis

Kaip jau buvo kalbėta straipsnyje „Panaudojimo atvejų atvaizdavimas Z specifikacija“, panaudojimo atvejis pradedamas sudarinėti nuo to, kad yra apibrėžiama sistema, toliau yra išskiriamas aktorius (sistemos vartotojas) bei jo veiksmai sistemoje. Tiek sistema, tiek aktoriai, tiek veiksmai yra tarpusavyje susiję ir sudaro pagrindą informacinės sistemos kūrimui. Konkrečiu menų

draugijos informacinės sistemos atveju yra keturių tipų vartotojai: interneto vartotojas, registruotas vartotojas, menininkas ir meno draugijos administracija. Kiekvienas jų atlieka skirtingus veiksmus sistemoje, taip pat daro ir skirtingą poveikį sistemai, pavyzdžiui, informacinės sistemos administratorius gali pilnai administruoti sistemoje esančius duomenis (keisti, trinti, įterpti ir t.t.), o interneto vartotojas jokio poveikio sistemoje esantiems duomenims nedaro (nebent nuspręstų prisiregistruoti) (1 pav.):



1 pav. Meno draugijos informacinės sistemos panaudojimo atvejų modelis

2.4. Nefunkciniai sistemos reikalavimai

Saugumas - svarbu, kad sistema būtų saugi, kad bet kuris vartotojas negalėtų keisti sistemos struktūros ar redaguoti duomenų. Tai turi būti išimtinai administratoriaus teisė.

IS išplėtimo reikalavimai - IS turi būti lengvai išplečiama prijungiant prie jos papildomus kompiuterius ar įrenginius (jei to reikia).

Taikomųjų programų suderinamumas - taikomosios programos turi būti tarpusavy suderintos.

Reikalavimai servisui - sistema, jos veikimas turi būti reguliariai tikrinamas, o radus gedimus ar neteisingą veikimą, gedimai ir trūkumai turi būti nedelsiant pašalinti.

Estetinė išvaizda – labai svarbu, kaip sistema atrodo, kadangi ji skirta išlavintą menininių skonį turintiems žmonėms.

Greitis – jį užtikrina MS. NET platforma

Panaudojamumas – sąsaja patogi naudoti.

Išplečiamumas – turi būti galimybė plėsti ir programinę sistemos dalį.

2.5. Sistemos realizavimo priemonių pasirinkimas

Kaip jau buvo minėta darbo uždavinių aptarime, labai svarbus yra sistemos modernumas – sistema turi būti kuriama kuo naujesnėmis technologijomis. Tokiu būdu yra užtikrinamas ne tik vartotojo komfortas dirbant su sistema, bet ir pačios sistemos efektyvumas, našumas, saugumas bei funkcionalumas. Visa tai atitinka Microsoft MS.NET technologijos, apie kurias šiek tiek plačiau kalbama tolesniuose skyreliuose.

2.5.1. MS VisualStudio.NET

VisualStudio.NET yra lengvai įvaldoma priemonė tobulesnių taikomųjų programų kūrimui. Ji palengvina programuotojo darbą ir padidina jo efektyvumą. Jos dėka galima greitai perprasti įvairias MS.NET siūlomas technologijas ir rezultatyviai jas taikyti taikomųjų programų kūrime. MS VisualStudio.NET apima VB.NET, VC.NET, C# bei ASP.NET. taip pat joje yra integruoti tokie internetiniai standartai kaip HTML ir XML.

2.5.2. MS.NET Framework

NET.Framework yra nauja platforma kurti integruotoms, į paslaugų sferą orientuotoms taikomosioms programoms, kurios atitinka visus šiandieninio internetinio verslo kriterijus, ir kurios panaudojamos reikalingai informacijai internete surinkti bei norimą produktą užsisakyti (nepaisant to, kokiomis kalbomis jos parašytos).

Šios platformos plusai yra daugkartinis kodo panaudojimas, resursų vadyba, daugiau nei vienos kalbos panaudojimo galimybės tobulinimas, saugumas ir administravimas. Be to, MS.NET nėra susietas vien su Windows operacinių sistemų šeima.

2.5.3. Visual C#

C# (C-Sharp) yra objektiškai orientuota programavimo kalba, Microsoft naudojama su .NET Framework. C# buvo sukurtas specialiai tam, kad būtų galima realizuoti plataus masto taikomas programas, naudojantis .NET Framework. Savo sintakse ji panaši tiek į C++, tiek į Java ir Microsoft ją laiko išsivysčiusia iš C ir C++ kalbų. C# sukūrė Anders Hejlsberg (Turbo Pascal'io autorius ir Delphi architektas), Scot Wiltamuth ir Peter Golde.

2.5.4. ASP.NET

ASP.NET sujungia produktyvumą su puikiu veikimu, patikimumu ir tam tikru failų išdėstymu.

Produktyvumas:

Lengvai programuojamas modelis. ASP.NET leidžia daug lengviau kurti taikomas realaus pasaulinio tinklo programas. ASP.NET serverio kontrolieriai leidžia HTML stiliaus programavimą, kuris įgalina kurti puikius puslapius naudojant daug mažiau kodo nei klaiskiniame ASP. Duomenų pateikimas, įvedimas, failų patalpinimas – viskas padaroma daug paprasčiau. Be to, ASP.NET puslapiai veikia visose naršyklėse – įskaitant ir Netscape, Opera, AOL bei Internet Explorer. ASP.NET leidžia programuoti pasirinktinai bet kokia iš daugiau nei 25 kalbų (tame tarpe VB.NET, Jscript.NET ir t.t.).

Taikomųjų programų savybės anksčiau buvo sunkiai įdiegiamos arba reikalavo papildomo komponento, o dabar tai padaroma tik keliomis papildomomis programos kodo eilutėmis (naudojant .NET Framework).

2.5.5. ADO.NET

ADO.NET suteikia priėjimą prie tokių duomenų šaltinių kaip Microsoft SQL Server. Duomenimis besikeičiančios vartotojo taikomosios programos naudoja ADO.NET prisijungti prie šių duomenų šaltinių ir reikiamus duomenis išrinkti, valdyti bei papildyti.

ADO.NET valdo .NET Framework duomenis, kurie leidžia prisijungti prie duomenų bazės, vykdyti joje komandas ir pateikti rezultatus. Rezultatai yra arba iškart atspausdinami, arba patalpinami į ADO.NET DataSet objektą tam, kad būtų vartotojui pateikti kai prireiks. ADO.NET DataSet objektas gali būti panaudotas ir atskirai nuo .NET Framework duomenų tiekėjo tam, kad valdyti vietinius taikomosios programos duomenis ar paimtus iš XML.

2.5.6. MS SQL Server 2000

SQL Server 2000 yra vertas dėmesio pirmiausia dėl išsamios duomenų analizės ir duomenų gavybos (data mining) galimybių, kurios yra integruotos kartu su įprastomis taikomosiomis programomis, tokiomis kaip Microsoft Office. Dėl susiderinimo (self-tuning) ir dinaminio savikonfigūravimo yra optimizuojamas duomenų bazės veikimas, o vadybos įrenginiai automatizuoja įprastus veiksmus. Grafiniai įrankiai ir vedliai (wizards) palengvina duomenų bazės dizainą, nustatymus, veikimo monitoringą, kuris leidžia duomenų bazės administratoriui susikonsultuoti ties strateginiais verslo poreikiais. Ne taip kaip jo konkurentai, SQL Server 2000 suteikia galingą ir lengvai suprantamą duomenų valdymo platformą. Kiekviena programinės įrangos licenzija turi ir platų valdymo ir tobulinimo įrangos asortimentą. Turi ir naujas galimybes, tokias kaip informavimo (notifikavimo) paslaugas. SQL Server 2000 yra tinkamiausia priemonė duomenims saugoti, puikiai veikianti .NET Framework aplinkoje.

2.6. Analizės išvados

Prieš kuriant meno draugijos informacinę sistemą, buvo atlikta esančių meno draugijų informacinių sistemų analizė, kuri parodė, kad Lietuvoje veikiančios panašios sistemos netenkina vartotojų poreikių. Todėl buvo suformuluotas uždavinys sukurti Meno draugijos informacinę sistemą. Tuo tikslu:

- Įvardinti būsimos sistemos vartotojai: meno draugijos administracija, menininkai, registruoti vartotojai, interneto vartotojai.
- Išanalizavus vartotojų reikalavimus sistemai, buvo nutarta realizuoti svarbiausias sistemai reikalingas funkcijas: informacijos pateikimą, peržiūrą, registraciją, užsakymų priėmimą.
- Sudarytas panaudojimo atvejų modelis
- Įvardinti nefunkciniai reikalavimai: saugumas, reikalavimai IS išplėtimui, funkcionalumas, rezultatyvumas, suderinamumas, reikalavimai servisui.
- Sistema bus realizuojama: Visual Studio.NET priemonėmis, naudojant ADO.NET, ASP.NET, MS SQL Server 2000 technologijas bei programuojant C#.

Sistema bus kuriama pagal apibrėžtus kokybinius kriterijus: geras estetinis vaizdas, intuityvus naudojimas ir patogumas vartotojui, korektiškas veikimas.

Tolesniuose skyriuose pateikiamas sistemos projektas ir realizacijos aprašymas, nagrinėjamos sistemos savybės.

3. MENO DRAUGIJOS SISTEMOS MODELIS IR FORMALI SPECIFIKACIJA

Panaudojimo atvejai, kuriuos pasiūlė I. Jacobson kaip priemonę sistemai modeliuoti[2], puikiai tinka ir Meno draugijos informacinės sistemos modeliui (šis modelis yra pavaizduotas 1pav.). Šiame skyriuje bus plačiau kalbama apie panaudojimo atvejus (jie įvardinami natūralia kalba) bei pateikiama sistemos Z specifikacija. Plačiau apie metodą kaip transformuoti panaudojimo atvejus į Z specifikaciją galima paskaityti straipsnyje “Panaudojimo atvejų atvaizdavimas Z specifikacija”[1].

3.1. Sistemos reikalavimų specifikacija

Pirmoje dalyje 1 paveiksle pavaizduotas panaudojimo atvejų modelis, kuris buvo sudarytas išanalizavus Meno draugijos informacinės sistemos vartotojų poreikius. Pilnai modelio analizei kiekvienas panaudojimo atvejis natūralia kalba aptariamas atskiroje lentelėje:

3 lentelė. PA “Peržiūrėti informaciją apie meno kūrinius, autorius, parodas“ specifikacija

Panaudojimo atvejis	Peržiūrėti info apie meno kūrinius, autorius, parodas.
Numeris	PA1
Aktorius	Interneto vartotojas, Registruotas vartotojas, Menininkas, Administratorius
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Vartotojai yra įtraukti į menininkų draugijos vartotojų duomenų bazę arba ne (jei tai Interneto vartotojas).
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pažymėti eilutę su dominančiu kūriniu.	1. Sistema pakeičia žymimos eilutės spalvą, parodydama, kad eilutė pažymėta.
2. Paspausti paveikslą peržiūros mygtuką	2. Sistema pateikia paveikslą pavyzdį.
Po sąlyga	Sistemoje niekas nepasikeičia.

4 lentelė. PA „Atlikti paiešką“ specifikacija

Panaudojimo atvejis	Atlikti paiešką
Numeris	PA2
Aktorius	Analogiška PA1.
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Analogiška PA1.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Interneto vartotojas gali :	
A. Pagal bet kurio duomenų stulpelio ieškomą reikšminį žodį susirasti pilną su tuo žodžiu susijusią informaciją.	1.A.1. Sistema gavusi užklausą suranda duomenų eilutę, turinčią vartotojo įvestą reikšminį žodį.
B. Peržiūrėti pasirinktą paveiksluką.	3.C.1. Sistema parodo prašomą paveiksluką.
2. Vartotojas Registruotas vartotojas gali:	
A. Pagal bet kurį reikšminį duomenų stulpelio žodį rasti duomenis tiek apie patį menininką, tiek apie jo kūrybą, parodas	2.A.1. Sistema pateikia visą su menininku susijusią informaciją.
3. Vartotojas Menininkas gali:	
A. Pagal bet kurį reikšminį duomenų stulpelio žodį rasti duomenis tiek apie patį menininką, tiek apie jo kūrybą, parodas.	3.A.1. Sistema pateikia visą su menininku susijusią informaciją.
B. Įkelti naują savo paveikslą.	3.B.1. Sistema įrašo paveiksluką į meno draugijos duomenų bazę.
C. Peržiūrėti pasirinktą paveiksluką.	3.C.1. Sistema parodo prašomą paveiksluką.
4. Vartotojas Menininkų Draugijos administracijos darbuotojas gali::	
A. Pagal bet kurį reikšminį duomenų stulpelio žodį rasti visus duomenis apie registruotą vartotoją ar menininką.	4.A.1. Sistema pateikia duomenis apie surastą vartotoją.
Po sąlyga	Vartotojas gali peržiūrėti surastą informaciją.
Veiklos taisyklės	Vartotojas turi teisingai įvesti paieškos žodį.
Specialūs reikalavimai (nefunkciniai)	Kiekvienam vartotojui prieinama ne visa, o tik tam tikra informacija, pagal jo

	statusui priskiriamas privilegijas.
--	-------------------------------------

5 lentelė. PA „Registruoti naują Registruotą vartotoją“ specifikacija

Panaudojimo atvejis	Registruotis
Numeris	PA3
Aktorius	Registruotas vartotojas
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Vartotojas dar nėra tapęs Registruotu vartotoju, todėl nei jo prisijungimo vardo, nei slaptažodžio Meno draugijos informacinės sistemos duomenų bazėje nėra, tačiau jis gali užsiregistruoti čia pat internete, pateikęs savo asmens duomenis ir tapti Registruotu vartotoju.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti pasirinktą vartotojo vardą.	
2. Įvesti ir pakartoti pasirinktą slaptažodį.	
3. Įvesti savo pavardę.	
4. Įvesti savo gimimo datą.	
5. Įvesti savo namų adresą.	
6. Įvesti savo telefono numerį.	
7. Išsaugoti įvestus duomenis.	1-7. Jei vartotojas suveda visus duomenis ir slaptažodžių laukai sutampa, jis yra įtraukiamas į Meno draugijos informacinę sistemą kaip Registruotas vartotojas.
Po sąlyga	Duomenų bazėje išsaugoti naujo Registruoto vartotojo duomenys.
Alternatyvos (nesėkmės atvejai)	1-7a. Jei sistema randa neužpildytus registracijos formos laukus arba slaptažodžių laukai nesutampa, vartotojas neištraukiamas kaip naujas Registruotas vartotojas ir sistema praneša apie įvykusią klaidą.
Veiklos taisyklės	Naujas Registruotas vartotojas privalo užpildyti visus laukus bei į slaptažodžių laukus suvesti identiškus slaptažodžius.

6 lentelė. PA „Prisijungti“ specifikacija

Panaudojimo atvejis	Prisijungti
Numeris	PA4
Aktorius	Meno draugijos informacinės sistemos vartotojas (Registruotas vartotojas, Menininkas ar Meno draugijos administracija).
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Meno draugijos informacinės sistemos vartotojas (Interneto vartotojas, Registruotas vartotojas, Menininkas, Meno draugijos informacinės sistemos administracija) yra įtrauktas į sistemos vartotojų duomenų bazę. Meno draugijos administracijos darbuotojai yra įtraukti kaip administratoriai, o Menininkai, kaip turintys mažiau teisių nei administratoriai, bet daugiau nei Registruoti vartotojai. Registruotų vartotojai turi teisę matyti savo ir tam tikrus menininkų duomenis bei užsisakinėti meno kūrinius. Neregistruotų Interneto vartotojų teisės yra labiausiai ribotos.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti prisijungimo vardą.	
2. Įvesti slaptažodį.	
3. Patvirtinti įvestus duomenis.	<p>3.1. Pirmiausia sistemoje yra tikrinamas vartotojo prisijungimo vardas ir jį atitinkantis slaptažodis. Jei šie sutampai su esamais duomenų bazėje, tuomet vartotojas yra prijungiamas prie sistemos:</p> <ul style="list-style-type: none"> -jeigu vartotojas – Interneto vartotojas, tuomet užkraunamas parodomasis interfeisas; - jeigu vartotojas – Registruotas vartotojas, tuomet užkraunamas Registruoto vartotojo interfeisas; - jeigu vartotojas – Menininkas, užkraunamas Menininko interfeisas; - jeigu vartotojas – Meno draugijos administracijos darbuotojas, užkraunamas Meno draugijos administracijos darbuotojo interfeisas; <p>Jeigu prisijungimo duomenys neteisingi, sistema grąžinama į 1 žingsnį ir perspėja apie įvykusią klaidą ir jos priežastį.</p>
Po sąlyga	Vartotojas patenka į sistemą.
Alternatyvos (nesėkmės atvejai)	Neteisingas vartotojo prisijungimo vardas ir/arba slaptažodis. Sistema grįžta į 1 žingsnį.
Veiklos taisyklės	Vartotojas turi teisingai įvesti prisijungimo duomenis.
Specialūs reikalavimai (nefunkciniai)	Priėjimas prie sistemos turi būti saugus, t.y. ja naudotis gali tik tam tikri užregistruoti asmenys, pagal sistemos jiems suteikiamas teises.

7lentelė. PA„Užsisakyti meno kūriniai“ specifikacija

Panaudojimo atvejis	Užsisakyti meno kūriniai
Numeris	PA5
Aktorius	Registruotas vartotojas
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Registruotas vartotojas yra įtrauktas į Registruotų vartotojų sąrašą, kuris patalpintas Meno draugijos vartotojų duomenų bazėje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pažymėti duomenų lentelėje norimą paveikslą.	1.1 Sistema pakeičia pažymėtos eilutės spalvą, parodydama, kad ji pažymėta.
2. Paspausti užsakymo mygtuką.	2.1 Sistema patikrina, ar šis paveikslas nėra užsakytas, jei ne, tai įveda užsakymą į meno draugijos duomenų bazę.
2. Peržiūrėti savo meno kūrinių užsakymus.	2.1. Sistema atpažįstą konkretų registruotą vartotoją ir pateikia jo užsisakytų meno kūrinių sąrašą.
Po sąlyga	Užsakytas meno kūrinys.
Veiklos taisyklės	Vartotojas turi teisingai įvesti duomenis.

8 lentelė. PA „Įvesti ir koreguoti informaciją apie meno kūriniai“ specifikacija

Panaudojimo atvejis	Įvesti ir koreguoti informaciją apie meno kūriniai.
Numeris	PA6
Aktorius	Menininkas
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Menininkas yra įtrauktas į Menininkų sąrašą, kuris patalpintas Meno draugijos vartotojų

	duomenų bazėje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti naują meno kūrinį	1.1. Sistema atpažįsta menininką ir priskiria konkrečiam meno kūriniai autorių.
A. Užpildyti visus prašomus laukus.	1.A.1. Sistema tikrina ar užpildyti visi būtini laukai. Įvykus klaidai, praneša apie tai menininkui.
2. Ištrinti esamą meno kūrinį.	
A. Pasirinkti kūrinį duomenų peržiūros lentelėje.	
B. Naikinti.	2.B.1. Sistema tikrina, ar paveikslas nėra užsakytas. Jei nėra, tuomet paklausia vartotojo, ar jsi tikrai nori ištrinti įrašą ir tik jam patvirtinus ištrina.
3. Redaguoti meno kūrinio pavadinimą:	
A. Pasirinkti kūrinį duomenų peržiūros lentelėje.	
B. Sukatyvinti jo laukelį ir pakeisti įrašą.	
C. Išsaugoti.	3.C.1. Sistema išsaugo pakeistą meno kūrinio pavadinimą.
Po sąlyga	1. Išsaugotas naujas meno kūrinys. 2. Panaikintas meno kūrinys. 3. Redaguotas meno kūrinio pavadinimas.
Alternatyvos (nesėkmės atvejai)	1.A.1.a. Neužpildomi visi būtini laukai. 2.B.1.a. Meno kūrinys yra užsakytas.
Veiklos taisyklės	Menininkas turi užpildyti visus sistemos reikalaujamus laukus.
Ryšiai su kitais PA	Apima panaudojimo atvejį "Tikrinti teises"

9 lentelė. PA „Įvesti ir koreguoti informaciją apie parodas“ specifikacija

Panaudojimo atvejis	Įvesti ir koreguoti informaciją apie parodas.
Numeris	PA6
Aktorius	Menininkas
Sistema	Menininkų Draugijos informacinė sistema

Prieš sąlyga	Menininkas yra įtrauktas į Menininkų sąrašą, kuris patalpintas Menininkų Draugijos vartotojų duomenų bazėje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti naują meno parodą.	1.1. Sistema atpažįsta menininką ir priskiria naujiems duomenims autorių.
A. Užpildyti visus sistemos prašomus laukus.	1.2 Sistema tikrina ar visi laukai suvesti.
B. Išsaugoti duomenis.	1.B. Sistema tikrina, ar teisiškai užpildyti visi būtini laukai. Jei viskas tvarkoje paroda įtraukiama į meno draugijos duomenų bazę.
2. Redaguoti parodos duomenis.	
A. Pasirinkti parodą duomenų peržiūros lentelėje.	
B. Atlikti parodos duomenų redagavimą.	
C. Išsaugoti.	3.C.1. Sistema išsaugo pakeistus parodos duomenis.
Po sąlyga	1. Itraukti naujos parodos duomenis. 2. Redaguoti esami parodos duomenys.
Alternatyvos (nesėkmės atvejai)	1.A. Ne visi būtini laukai užpildyti.
Veiklos taisyklės	Menininkas turi suvesti teisingus duomenis apie parodą.
Ryšiai su kitais PA	Apima panaudojimo atvejį "Tikrinti teises"

10 lentelė. PA „Tvarkyti vartotojų informaciją“ specifikacija

Panaudojimo atvejis	Tvarkyti vartotojų informaciją
Numeris	PA7
Aktorius	Sistemos administratorius
Sistema	Meno draugijos informacinė sistema

Prieš sąlyga	Sistemos administratorius yra įtrauktas į sistemos administratorių sąrašą.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti naujo sistemos vartotojo duomenis.	1. Analogiška PA3
2. Redaguoti sistemos vartotojo duomenis:	2.1. Sistema pateikia sistemos vartotojų sąrašą.
A. Pasirinkti vartotoją, kurio duomenis koreguos.	
B. Atlikti korekcijas.	
C. Patvirtinti duomenis.	2.C.1. Analogiška PA7 I.E.1.
3. Naikinti sistemos vartotojo duomenis:	3.1. Analogiška PA7 2.1.
A. Pasirinkti vartotoją, kurio duomenis naikins.	
B. Patvirtinti naikinimą.	3.B.1. Tikrina ar vartotojas neturi užsakytų arba nėra užsakęs meno kūrinų. Sistema prašo patvirtinti naikinimo operaciją.
C. Patvirtinti naikinimo operaciją.	3.C.1. Sistema naikina vartotojo duomenis.
Po sąlyga	1. Užregistruotas naujas sistemos vartotojas. 2. Redaguoti egzistuojančio sistemos vartotojo duomenys. 3. Panaikintas sistemos vartotojas
Alternatyvos (nesėkmės atvejai)	Analogiška PA3 1-7a
Veiklos taisyklės	Sistemos administratorius turi teisingai įvesti vartotojo duomenis. Neturi teisės savavališkai keisti ar naikinti vartotojų duomenų.
Ryšiai su kitais PA.	Yra agreguotas panaudojimo atvejo "Tvarkyti informaciją". Apima panaudojimo atvejį "Tikrinti teises".

11 lentelė. PA „Tikrinti teises“ specifikacija

Panaudojimo atvejis	Tikrinti teises
Numeris	PA8

Aktorius	Meno draugijos informacinės sistemos vartotojas (Registruotas vartotojas, Menininkas ar Meno draugijos administracija).
Sistema	Meno draugijos Informacinė Sistema
Prieš sąlyga	Analogiška PA1 prieš sąlygai.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka vykdyti kokį nors veiksmą	1.1 Sistema tikrina, ar vartotojas turi teises vykdyti pasirinktą veiksmą.
Po sąlyga	Vartotojas gali vykdyti pasirinktą veiksmą.
Alternatyvos (nesėkmės atvejai)	Vartotojas neturi reikiamų teisių, todėl vykdyti veiksmo negali – tai jam uždraudžia sistema.

3.2. Formali specifikacija Z kalba

Formalūs metodai padeda išvengti nesusipratimų ir dviprasmybių. Jie ypač taikomi kritinėse sistemose, kur bet koks netinkamai interpretuotas reikalavimas gali sukelti didelę problemą. Daugumai sistemų formalūs metodai nėra būtini, tačiau jei norima parašyti visiškai aiškia sistemą ar tam tikrą jos dalių specifikaciją tenka jų imtis. Prieš tai išsamiai aprašius vartotojo reikalavimus natūralia kalba, juos formalizuoti nėra sudėtinga (plačiau apie tai straipsnyje „Panaudojimo atvejų atvaizdavimas Z specifikacija“[1]). Vienas formalių metodų yra specifikavimas Z kalba, kuri yra paremta Zermelo-Fraenkel aibių teorija bei predikatų logika. Toliau yra pateikiama meno draugijos administracinės dalies Z specifikacija:

MenininkuDraugijosSistema

[PAV_KODAS, login, slapt, adresas, pavarde, telefonas]

// PAV_KODAS - tai paveikslo kodu (id) aibe

// login - vartotoju loginu aibe

// slapt - vartotojams suteiktu slaptazodziu aibe

// adresas - vartotoju adresu aibe

// pavarde - vartotoju pavardziu aibe

// telefonas - vartotoju telefonu aibe

state

rezervuoti: PAV_KODAS login

available: PAV_KODAS

kataloginiai: PAV_KODAS

menininkai: login slapt

registruoti: login slapt

visi_vart: login slapt

tel: login telefonas

```
adr: login  adresas
pav: login  pavarde
```

```
kataloginiai  dom rezervuoti  available
visi_vart  menininkai  registruoti
```

```
// kataloginiai - tai visi patalpinti paveikslai
// available - tai tie paveikslai, kuriuos galima rezervuoti
// rezervuoti - tai tokie paveikslai, kuriuos jau rezervavo(uzsisake vartotojas)
init
```

```
rezervuoti'
available'
kataloginiai'
menininkai'
registruoti'
```

```
// NaujasKataloginis schema apraso, kaip i sistema yra ivedamas naujas kataloginis
paveikslas
```

```
NaujasKataloginis
```

```
state
state'
new_pav_kodas?: PAV_KODAS
log?: login
zinute!: string
```

```
pre ((new_pav_kodas? kataloginiai) (
new_pav_kodas? available) (
new_pav_kodas? dom rezervuoti))
log? dom menininkai
kataloginiai' kataloginiai new_pav_kodas?
zinute! "atlikta_sekmingai"
```

```
// NaujasKataloginisKlaida rodo del koku priezasciu operacija gali buti nesekminga
NaujasKataloginisKlaida
```

```
state
state'
new_pav_kodas?: PAV_KODAS
log?: login
zinute!: string
```

```
pre (new_pav_kodas? kataloginiai) (
new_pav_kodas? available) (
new_pav_kodas? dom rezervuoti) (
log? dom menininkai)
zinute! "neatlikta"
```

```
NaujasKataloginiaiIvestas (NaujasKataloginis NaujasKataloginisKlaida)  
op AddNaujasKataloginis  
NaujasKataloginiaiIvestas
```

```
// NaujasAvailable schema nusako, kaip kataloginis paveikslas tampa available  
NaujasAvailable
```

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? available) (  
pav_kodas? dom rezervuoti) (  
log? dom menininkai)  
available' available pav_kodas?  
zinate! "paveikslas sekmingai ivestas"
```

```
// kokiais atvejais ivyksta klaida  
NaujasAvailableKlaida
```

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? available) (  
pav_kodas? dom rezervuoti) (  
log? dom menininkai)  
zinate! "ivedant paveiksla ivyko klaida"
```

```
NaujasAvailableItrauktas (NaujasAvailable NaujasAvailableKlaida)  
op AddNaujasAvailable  
NaujasAvailableItrauktas
```

```
// paveikslas is available tampa rezervuotu  
NaujasRezervuotas
```

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string
```



```
pre (pav_kodas? kataloginiai) (
pav_kodas? available) (
pav_kodas? dom rezervuoti) ((
log? dom menininkai) (
log? dom registruoti))
dom rezervuoti' dom rezervuoti pav_kodas?
zinate! "paveikslas sekmingai ivestas"
```

```
// ivyksta klaida
NaujasRezervuotasKlaida
```

```
state
state'
pav_kodas?: PAV_KODAS
log?: login
zinate!: string
```

```
pre (pav_kodas? kataloginiai) (
pav_kodas? available) (
pav_kodas? dom rezervuoti) ((
log? dom menininkai) (
log? dom registruoti))
zinate! "ivedant paveiksla ivyko klaida"
```

```
NaujasRezervuotasItrauktas (NaujasRezervuotas NaujasRezervuotasKlaida)
op AddNaujasRezervuotas
NaujasRezervuotasItrauktas
```

```
// kataloginiai paveikslai istrinami
IstrintiKataloginiai
```

```
state
state'
pav_kodas?: PAV_KODAS
log?: login
zinate!: string
```

```
pre (pav_kodas? kataloginiai) (
pav_kodas? available) (
pav_kodas? dom rezervuoti) (
log? dom menininkai)
kataloginiai' kataloginiai pav_kodas?
zinate! "paveikslas sekmingai istrintas"
```

```
// istrinant katalogini paveiksla ivyko klaida
IstrintiKataloginiaiKlaida
```

```
state
```

```
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? available) (  
pav_kodas? dom rezervuoti) (  
log? dom menininkai)  
zinate! "ivyko klaida. paveikslas neistrintas"
```

```
KataloginiaiIstrinti (IstrintiKataloginiai IstrintiKataloginiaiKlaida)  
op IstrintiKataloginis  
KataloginiaiIstrinti
```

```
// rezervuota paveiksla nupirkus, jis istrinamas is sistemos
```

```
IstrintiRezervuota
```

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? dom rezervuoti) (  
log? dom menininkai)  
dom rezervuoti' dom rezervuoti pav_kodas?  
zinate! "paveikslas sekmingai istrintas"
```

```
// istrinant rezervuota paveiksla ivyko klaida
```

```
IstrintiRezervuotaKlaida
```

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? dom rezervuoti) (  
log? dom menininkai)  
zinate! "ivyko klaida. paveikslas neistrintas"
```

```
RezervuotiIstrinti (IstrintiRezervuota IstrintiRezervuotaKlaida)
```

```
op RezervuotiIstrinti
```

```
RezervuotiIstrinti
```

```
// paveikslas keicia statusa (is available i rezervuota), paveiksla uzsisakius
// login cia reikalingas tam, kad butu aisku, kuris vartotojas uzsisake kuri paveiksla
PakeisitiStatusaIsAvail
```

```
state
state'
pav_kodas?: PAV_KODAS
log?: login
zinute!: string

pre (pav_kodas? kataloginiai) (
pav_kodas? dom rezervuoti) (
pav_kodas? available) (
log? dom visi_vart)
kataloginiai' kataloginiai
available' available pav_kodas?
rezervuoti' rezervuoti (pav_kodas?, log?)
zinute! "statusas pakeistas sekmingai"
```

```
// ivyksta klaida
PakeisitiStatusaIsAvailKlaida
```

```
state
state'
pav_kodas?: PAV_KODAS
log?: login
zinute!: string

pre (pav_kodas? kataloginiai) (
pav_kodas? dom rezervuoti) (
pav_kodas? available) (
log? dom visi_vart)
kataloginiai' kataloginiai
zinute! "statusas nepakeistas"
```

```
AvailStatusoKeitimas (PakeisitiStatusaIsAvail PakeisitiStatusaIsAvailKlaida)
op KeiciaAvailStatusa
AvailStatusoKeitimas
```

```
// paveikslo statusas keiciamas is rezervuoto i available (vartotojui atsisakius rezervuoto
paveikslo)
```

```
PakeisitiStatusaIsRezerv
state
state'
pav_kodas?: PAV_KODAS
log?: login
zinute!: string
```

```
pre (pav_kodas? kataloginiai) (  
pav_kodas? dom rezervuoti) (  
pav_kodas? available) (  
log? dom visi_vart)  
kataloginiai' kataloginiai  
dom rezervuoti' dom rezervuoti pav_kodas?  
available' available pav_kodas?  
zinate! "statusas pakeistas sekmingai"
```

PakeisitiStatusaIsRezervKlaida

```
state  
state'  
pav_kodas?: PAV_KODAS  
log?: login  
zinate!: string  
  
pre (pav_kodas? kataloginiai) (  
pav_kodas? dom rezervuoti) (  
pav_kodas? available) (  
log? dom visi_vart)  
kataloginiai' kataloginiai  
zinate! "statusas nepakeistas"
```

*RezervKeitimas (PakeisitiStatusaIsRezerv PakeisitiStatusaIsRezervKlaida)
op KeiciaRezervStatusa
RezervKeitimas*

// ivedamas naujas vartotojas - naujas menininkas
// priregistruojant menininka reikia ir kontaktiniu duomenu (pavardes, adreso, telefono)
NaujasMenininkas

```
state  
state'  
new_men?: login  
new_slapt?: slapt  
new_adr?: adresas  
new_tel?: telefonas  
new_pav?: pavarde  
zinate!: string  
  
pre ((new_men? dom menininkai) (  
new_men? dom registruoti))  
menininkai' menininkai (new_men?, new_slapt?)  
tel' tel (new_men?, new_tel?)  
adr' adr (new_men?, new_adr?)  
pav' pav (new_men?, new_pav?)
```

zinate! "menininkas ivestas sekmingai"

// ivyksta klaida

NaujasMenininkasKlaida

state

state'

new_men?: login

zinate!: string

pre ((new_men? login))

zinate! "menininkas neivestas"

MenininkasIvestas (NaujasMenininkas NaujasMenininkasKlaida)

op IvestiMenininka

MenininkasIvestas

// ivedami registruoti vartotojai - potencialus paveikslu pirkejai

// reikalingi ju kontaktiniai duomenys

NaujasRegistruotas

state

state'

new_reg?: login

new_slapt?: slapt

new_adr?: adresas

new_tel?: telefonas

new_pav?: pavarde

zinate!: string

pre ((new_reg? dom registruoti) (

new_reg? dom menininkai))

registruoti' registruoti (new_reg?, new_slapt?)

tel' tel (new_reg?, new_tel?)

adr' adr (new_reg?, new_adr?)

pav' pav (new_reg?, new_pav?)

zinate! "registruotas ivestas sekmingai"

NaujasRegistruotasKlaida

state

state'

new_reg?: login

zinate!: string

pre ((new_reg? login))

zinate! "registruotas neivestas"

```
RegistruotasIvestas (NaujasRegistruotas NaujasRegistruotasKlaida)
  op IvestiRegistruota
    RegistruotasIvestas
```

```
// menininkas istrinamas, jam palikus draugija
```

```
IstrintiMenininka
```

```
state
state'
men?: login
sen_slapt: slapt
sen_tel?: telefonas
sen_adr?: adresas
sen_pav?: pavarde
zinute!: string
```

```
pre ((men? dom menininkai) (
men? dom registruoti))
menininkai' menininkai (men?, sen_slapt)
adr' adr (men?, sen_adr?)
tel' tel (men?, sen_tel?)
pav' pav (men?, sen_pav?)
zinute! "menininkas istrintas sekmingai"
```

```
IstrintiMenininkaKlaida
```

```
state
state'
men?: login
sen_slapt: slapt
zinute!: string
```

```
pre ((men? dom menininkai) (
men? dom registruoti))
zinute! "menininkas istrintas sekmingai"
```

```
MenininkasIstrintas (IstrintiMenininka IstrintiMenininkaKlaida)
  op IstrintiMen
    MenininkasIstrintas
```

```
// registruotas vartotojas gali buti istrintas savo noru arba priverstinai
```

```
IstrintiRegistruota
```

```
state
state'
reg?: login
sen_slapt?: slapt
sen_adr?: adresas
```

```
sen_tel?: telefonas
sen_pav?: pavarde
zinate!: string
```

```
pre ((reg? dom registruoti) (
reg? dom menininkai))
registruoti' registruoti (reg?, sen_slapt?)
tel tel (reg?, sen_tel?)
adr' adr (reg?, sen_adr?)
pav' pav (reg?, sen_pav?)
zinate! "registruotas istrintas sekmingai"
```

IstrintiRegistruotaKlaida

```
state
state'
reg?: login
zinate!: string
```

```
pre ((reg? dom registruoti))
zinate! "regstruotas neistrintas"
```

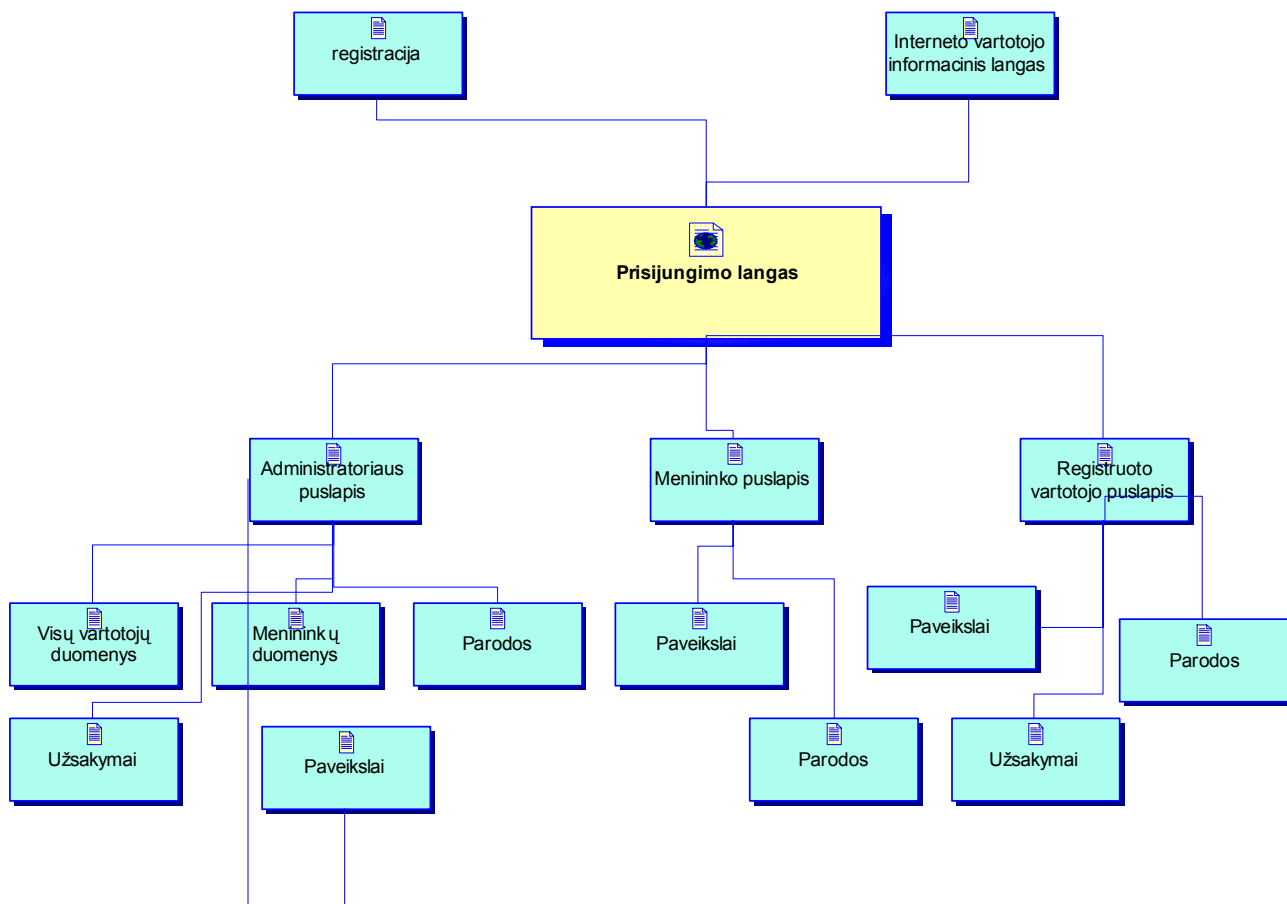
```
RegistruotasIstrintas (IstrintiRegistruota IstrintiRegistruotaKlaida)
op IstrintiReg
RegistruotasIstrintas
```

3.3. Vartotojų sąsajos modelis

Vartotojų sąsajos svarbiausias reikalavimas yra paprasta navigacija po puslapius. Pagrindiniame puslapyje yra prisijungiamo. Jei vartotojas dar neturi virtualių asmens duomenų (prisijungimo vardo ir/ ar slaptažodžio), jis gali prisiregistruoti (paspaudus atitinkamą mygtuką pagrindinėje formoje, vartotojas yra nukreipiamas į registracijos langą, kur sėkmingai suvedęs registracijos formoj reikalaujamus duomenis, gali jungtis prie sistemos). Jei į pagrindinį puslapį užklydęs interneto vartotojas nori įsitikinti, ar jam verta registruotis ir nori peržiūrėti informacinės sistemos duomenis, jis gali papulti į interneto vartotojui tik peržiūrai skirtą puslapį.

Yra trys sėkmingo prisijungimo alternatyvos. Jos yra nusakomos vartotojo teisių. Jei besiregistruojantis yra administratorius, sistema pagal jo prisijungimo vardą ir slaptažodį užkrauna jam administratoriaus aplinką. Atitinkamai skirtinga aplinka yra užkraunama tiek menininkui, tiek registruotam vartotojui. Šiose aplinkose vartotojai gali atlikti jiems pagal teises bei jų prisijungimo vardą ir slaptažodį leidžiamas operacijas. Menininkams leidžiama peržiūrėti esamą menininkų

informaciją(ne asmeninę, o tik susijusią su kūryba) bei atlikti kitas su informacija susijusias operacijas(pvz, paiešką, savų duomenų korekciją), taip pat leidžiama įvesti naujo paveikslų informaciją bei įdėti jo pavyzdį. Registruotas vartotojas gali peržiūrėti menininkų duomenis bei užsisakyti jam patikusį kūrinį.



2 pav. Vartotojų sąsajos modelis

3.4. Klasių aprašymas

Meno draugijos informacinėje sistemoje vienas internetinis puslapis (*Web_Form*) vartotojo sąsajos modelio (2 pav.) atitinka vieną programinę klasę, kurių kiekviena detaliau yra aprašyta 12 lentelėje.

12 lentelė. Sistemos klasių aprašas

<table border="1"> <thead> <tr> <th>Prisijungimas</th> </tr> </thead> <tbody> <tr> <td> #TextBox1 : TextBox #TextBox2 : TextBox #Button1 : Button #Button2 : Button #Label1 : Label #Label2 : Label #Label3 : Label #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Button3 : Button #Label4 : Label </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	Prisijungimas	#TextBox1 : TextBox #TextBox2 : TextBox #Button1 : Button #Button2 : Button #Label1 : Label #Label2 : Label #Label3 : Label #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Button3 : Button #Label4 : Label	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)	<p>Tai prisijungimo klasė, kurios veikimas detaliau aprašomas 6 lentelėje PA „Prisijungti“ bei 5 pav. „Prisijungimo puslapis“.</p>
Prisijungimas				
#TextBox1 : TextBox #TextBox2 : TextBox #Button1 : Button #Button2 : Button #Label1 : Label #Label2 : Label #Label3 : Label #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Button3 : Button #Label4 : Label				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>Registracija</th> </tr> </thead> <tbody> <tr> <td> #TextBox1 : TextBox #TextBox2 : TextBox #TextBox3 : TextBox #TextBox4 : TextBox #Label1 : Label #Label2 : Label #Label3 : Label #Label4 : Label #TextBox5 : TextBox #Label5 : Label #Label6 : Label #Label7 : Label #TextBox6 : TextBox #Button1 : Button #Button2 : Button #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Label_klaida : Label #TextBox7 : TextBox </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	Registracija	#TextBox1 : TextBox #TextBox2 : TextBox #TextBox3 : TextBox #TextBox4 : TextBox #Label1 : Label #Label2 : Label #Label3 : Label #Label4 : Label #TextBox5 : TextBox #Label5 : Label #Label6 : Label #Label7 : Label #TextBox6 : TextBox #Button1 : Button #Button2 : Button #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Label_klaida : Label #TextBox7 : TextBox	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai registracijos klasė, kurios veikimas detaliau aprašomas 5 lentelėje PA „Registruotis“ bei 6 pav. „Registracijos langas“.</p>
Registracija				
#TextBox1 : TextBox #TextBox2 : TextBox #TextBox3 : TextBox #TextBox4 : TextBox #Label1 : Label #Label2 : Label #Label3 : Label #Label4 : Label #TextBox5 : TextBox #Label5 : Label #Label6 : Label #Label7 : Label #TextBox6 : TextBox #Button1 : Button #Button2 : Button #SqlConnection1 : SqlConnection #SqlCommand1 : SqlCommand #Label_klaida : Label #TextBox7 : TextBox				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs)				

<table border="1"> <thead> <tr> <th style="text-align: center;">Internetui</th> </tr> </thead> <tbody> <tr> <td> #ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet91 : DataSet9 #Image1 : Image #Button2 : Button #Button1 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	Internetui	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet91 : DataSet9 #Image1 : Image #Button2 : Button #Button1 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)	<p>Tai demonstracinė versija, kurios veikimas detaliau aprašomas 3 lentelėje PA “Peržiūrėti informaciją apie meno kūrinius, autorius, parodas“ bei 7 pav. „Demonstracinė versija“.</p>
Internetui				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet91 : DataSet9 #Image1 : Image #Button2 : Button #Button1 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th style="text-align: center;">Administratoriaus</th> </tr> </thead> <tbody> <tr> <td> #Button1 : Button #Button3 : Button #Button4 : Button #Button5 : Button #Button6 : Button #Button2 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button5_Click(in sender : object, in e : EventArgs) -Button6_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	Administratoriaus	#Button1 : Button #Button3 : Button #Button4 : Button #Button5 : Button #Button6 : Button #Button2 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button5_Click(in sender : object, in e : EventArgs) -Button6_Click(in sender : object, in e : EventArgs)	<p>Tai administratoriaus aplinkos pirmasis puslapis, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 8 pav. „Administratoriaus aplinka“.</p>
Administratoriaus				
#Button1 : Button #Button3 : Button #Button4 : Button #Button5 : Button #Button6 : Button #Button2 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button5_Click(in sender : object, in e : EventArgs) -Button6_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th style="text-align: center;">AdmMenininkai</th> </tr> </thead> <tbody> <tr> <td> #ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet71 : DataSet7 #Button2 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	AdmMenininkai	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet71 : DataSet7 #Button2 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių administratoriaus aplinkos puslapių, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 10 pav. „Visų menininkų puslapis“.</p>
AdmMenininkai				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet71 : DataSet7 #Button2 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button2_Click(in sender : object, in e : EventArgs)				

<table border="1"> <thead> <tr> <th>AdmParodos</th> </tr> </thead> <tbody> <tr> <td> #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Button1 : Button #ASPxGrid1 : ASPxGrid </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	AdmParodos	#sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Button1 : Button #ASPxGrid1 : ASPxGrid	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių administratoriaus aplinkos puslapių, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 13 pav. „Visų parodų puslapis“.</p>
AdmParodos				
#sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Button1 : Button #ASPxGrid1 : ASPxGrid				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>AdmUzsakymai</th> </tr> </thead> <tbody> <tr> <td> #ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #dataSet41 : DataSet4 #Button1 : Button #sqlDataAdapter1 : SqlDataAdapter </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	AdmUzsakymai	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #dataSet41 : DataSet4 #Button1 : Button #sqlDataAdapter1 : SqlDataAdapter	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių administratoriaus aplinkos puslapių, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 12 pav. „Visų užsakymų puslapis“.</p>
AdmUzsakymai				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #dataSet41 : DataSet4 #Button1 : Button #sqlDataAdapter1 : SqlDataAdapter				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>AdmPaveikslai</th> </tr> </thead> <tbody> <tr> <td> #Button1 : Button #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #dataSet201 : DataSet20 #ASPxGrid1 : ASPxGrid </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	AdmPaveikslai	#Button1 : Button #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #dataSet201 : DataSet20 #ASPxGrid1 : ASPxGrid	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių administratoriaus aplinkos puslapių, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 11 pav. „Visų paveikslų puslapis“.</p>
AdmPaveikslai				
#Button1 : Button #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #dataSet201 : DataSet20 #ASPxGrid1 : ASPxGrid				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				

<table border="1"> <thead> <tr> <th>AdmVartotojai</th> </tr> </thead> <tbody> <tr> <td> #Button3 : Button #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection2 : SqlConnection #dataSet21 : DataSet2 #ASPxGrid1 : ASPxGrid </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -ASPxGrid1_UpdateCommand(in source : object, in e : RowCommandEventArgs) -Button3_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	AdmVartotojai	#Button3 : Button #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection2 : SqlConnection #dataSet21 : DataSet2 #ASPxGrid1 : ASPxGrid	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -ASPxGrid1_UpdateCommand(in source : object, in e : RowCommandEventArgs) -Button3_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių administratoriaus aplinkos puslapių, kurio veikimas detaliau aprašomas 10 lentelėje PA „Tvarkyti vartotojų informaciją“ bei 9 pav. „Visų vartotojų puslapis“.</p>
AdmVartotojai				
#Button3 : Button #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection2 : SqlConnection #dataSet21 : DataSet2 #ASPxGrid1 : ASPxGrid				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -ASPxGrid1_UpdateCommand(in source : object, in e : RowCommandEventArgs) -Button3_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>MenInterface</th> </tr> </thead> <tbody> <tr> <td> #Button3 : Button #Button4 : Button #Button1 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	MenInterface	#Button3 : Button #Button4 : Button #Button1 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)	<p>Tai pirmasis menininko aplinkos puslapis, kurio veikimas detaliau aprašomas 14 pav. „Menininko aplinka“.</p>
MenInterface				
#Button3 : Button #Button4 : Button #Button1 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button4_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>MenParodos</th> </tr> </thead> <tbody> <tr> <td> #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet61 : DataSet6 #Button1 : Button #ASPxGrid1 : ASPxGrid </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	MenParodos	#sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet61 : DataSet6 #Button1 : Button #ASPxGrid1 : ASPxGrid	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių menininko aplinkos puslapių, kurio veikimas detaliau aprašomas 9 lentelėje PA „Ivesti ir koreguoti informaciją apie parodas bei 16 pav. „Menininko parodų puslapis“.</p>
MenParodos				
#sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet61 : DataSet6 #Button1 : Button #ASPxGrid1 : ASPxGrid				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				

<table border="1"> <thead> <tr> <th>Menininko</th> </tr> </thead> <tbody> <tr> <td> #ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Image1 : Image #Button1 : Button #Button2 : Button #Button3 : Button #filUpload : HtmlInputFile </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	Menininko	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Image1 : Image #Button1 : Button #Button2 : Button #Button3 : Button #filUpload : HtmlInputFile	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių menininko aplinkos puslapių, kurio veikimas detaliau aprašomas 8 lentelėje PA „Įvesti ir koreguoti informaciją apie meno kūrinius“ bei 15 pav. „Menininko paveikslo įkėlimo puslapis“.</p>
Menininko				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet51 : DataSet5 #Image1 : Image #Button1 : Button #Button2 : Button #Button3 : Button #filUpload : HtmlInputFile				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>RegInterface</th> </tr> </thead> <tbody> <tr> <td> #Button1 : Button #Button2 : Button #Button3 : Button #Button4 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button4_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	RegInterface	#Button1 : Button #Button2 : Button #Button3 : Button #Button4 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button4_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)	<p>Tai pirmasis registruoto vartotojo puslapis, kurio veikimas detaliau aprašomas 17 pav. „Registruoto vartotojo puslapis“.</p>
RegInterface				
#Button1 : Button #Button2 : Button #Button3 : Button #Button4 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button4_Click(in sender : object, in e : EventArgs) -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>RegParodos</th> </tr> </thead> <tbody> <tr> <td> #ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet101 : DataSet10 #Button1 : Button </td> </tr> <tr> <td> -Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) </td> </tr> </tbody> </table>	RegParodos	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet101 : DataSet10 #Button1 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių registruoto vartotojo puslapių, kurio veikimas detaliau aprašomas 3 lentelėje PA „Peržiūrėti informaciją apie meno kūrinius, autorius, parodas“ bei 20 pav. „Registruoto vartotojo parodų puslapis“.</p>
RegParodos				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet101 : DataSet10 #Button1 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				

<table border="1"> <thead> <tr> <th>RegUzsakymai</th> </tr> </thead> <tbody> <tr> <td>#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #sqlCommand1 : SqlCommand #Button1 : Button</td> </tr> <tr> <td>-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)</td> </tr> </tbody> </table>	RegUzsakymai	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #sqlCommand1 : SqlCommand #Button1 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių registruoto vartotojo puslapių, kurio veikimas detaliau aprašomas 19 pav. „Registruoto vartotojo užsakymų puslapis“.</p>
RegUzsakymai				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet211 : DataSet21 #sqlCommand1 : SqlCommand #Button1 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs)				
<table border="1"> <thead> <tr> <th>Registruotas</th> </tr> </thead> <tbody> <tr> <td>#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet31 : DataSet3 #Button2 : Button #Button3 : Button #sqlCommand1 : SqlCommand #Label_klaida : Label #Image1 : Image #sqlCommand2 : SqlCommand #Button1 : Button</td> </tr> <tr> <td>-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)</td> </tr> </tbody> </table>	Registruotas	#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet31 : DataSet3 #Button2 : Button #Button3 : Button #sqlCommand1 : SqlCommand #Label_klaida : Label #Image1 : Image #sqlCommand2 : SqlCommand #Button1 : Button	-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)	<p>Tai vienas iš vidinių registruoto vartotojo puslapių, kurio veikimas detaliau aprašomas 7 lentelėje PA „Užsisakyti meno kūrinis“ bei 18 pav. „Registruoto vartotojo paveikslų peržiūros puslapis“.</p>
Registruotas				
#ASPxGrid1 : ASPxGrid #sqlSelectCommand1 : SqlCommand #sqlInsertCommand1 : SqlCommand #sqlUpdateCommand1 : SqlCommand #sqlDeleteCommand1 : SqlCommand #sqlConnection1 : SqlConnection #sqlDataAdapter1 : SqlDataAdapter #dataSet31 : DataSet3 #Button2 : Button #Button3 : Button #sqlCommand1 : SqlCommand #Label_klaida : Label #Image1 : Image #sqlCommand2 : SqlCommand #Button1 : Button				
-Page_Load(in sender : object, in e : EventArgs) #OnInit(in e : EventArgs) -InitializeComponent() -Button1_Click(in sender : object, in e : EventArgs) -Button3_Click(in sender : object, in e : EventArgs) -Button2_Click(in sender : object, in e : EventArgs)				

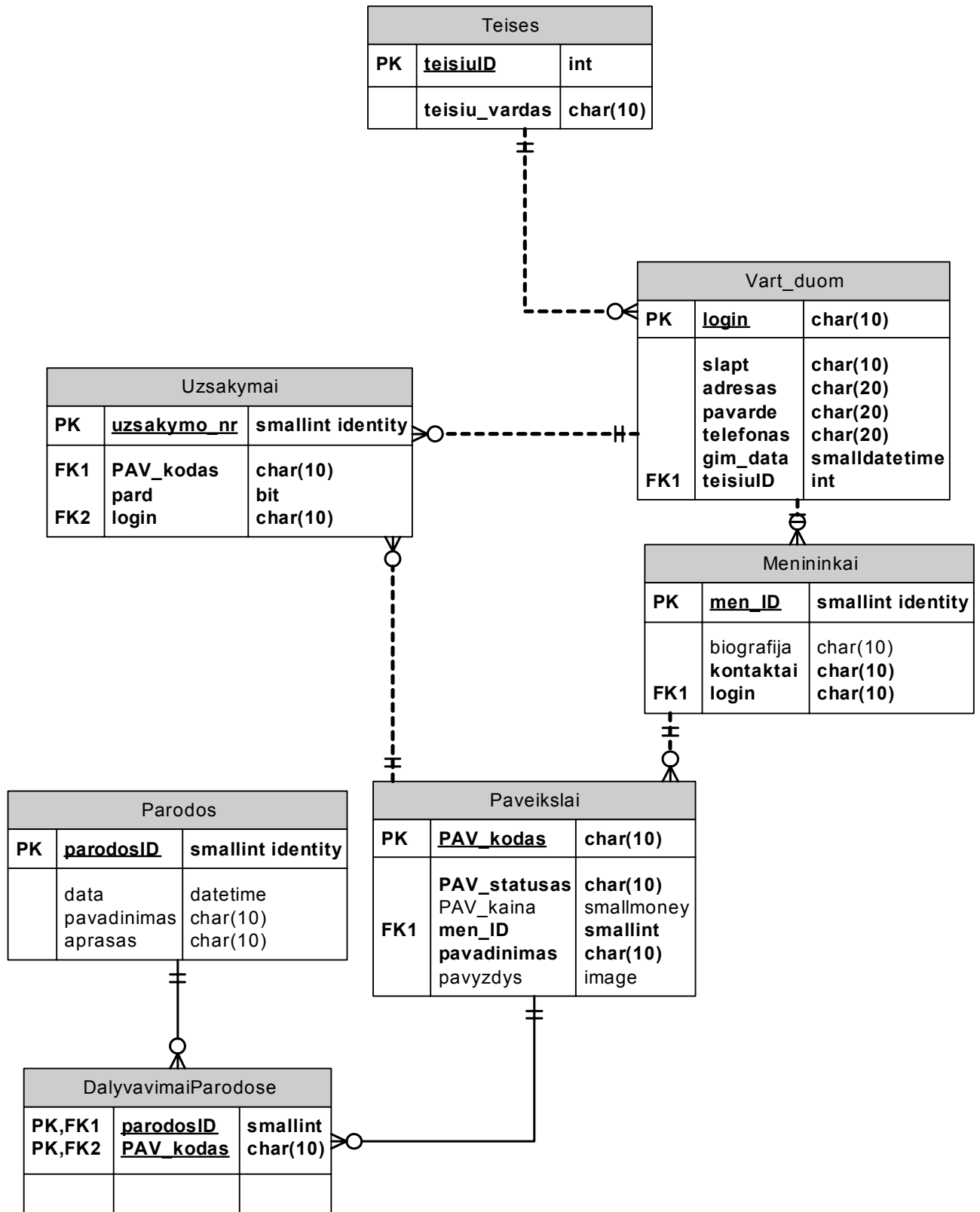
3.5. Duomenų bazės schema

Meno draugijos informacinės sistemos duomenų bazė projektuojama su MS Visio. Apgalvojus vartotojų reikalavimus sistemai duomenų bazė sudaroma iš 7 lentelių: Vart_duom – tai lentelė, kurioje yra visų sistemos vartotojų visi asmens duomenys, tokie kaip pavardė, adresas, telefonas... Taip pat virtualūs asmens duomenys, tokie kaip prisijungimo vardas(*login*) ir slaptažodis(*slapt*), kuris užtikrina vartotojo saugų darbą sistemoje su tiek viešais, tiek jam asmeniškai skirtais duomenimis (pavyzdžiui, savais užsakymais). Identifikacinis lentelės laukas yra *login*. Jis yra unikalus – duomenų bazėje negali būti dar vieno tokio paties prisijungimo vardo.

Lentelėje “Teisės” teisės yra sunumeruojamos bei įvardinama, koks teisių identifikatorius kokio tipo vartotojui priskiriamas. Pagal lauko „teisiuID“ numerį sistemoje vartotojas bus atpažįstamas ir, kaip minėta 7 lentelėje (PA „Prisijungti“), jam bus užkraunamas atitinkamas interfeisas.

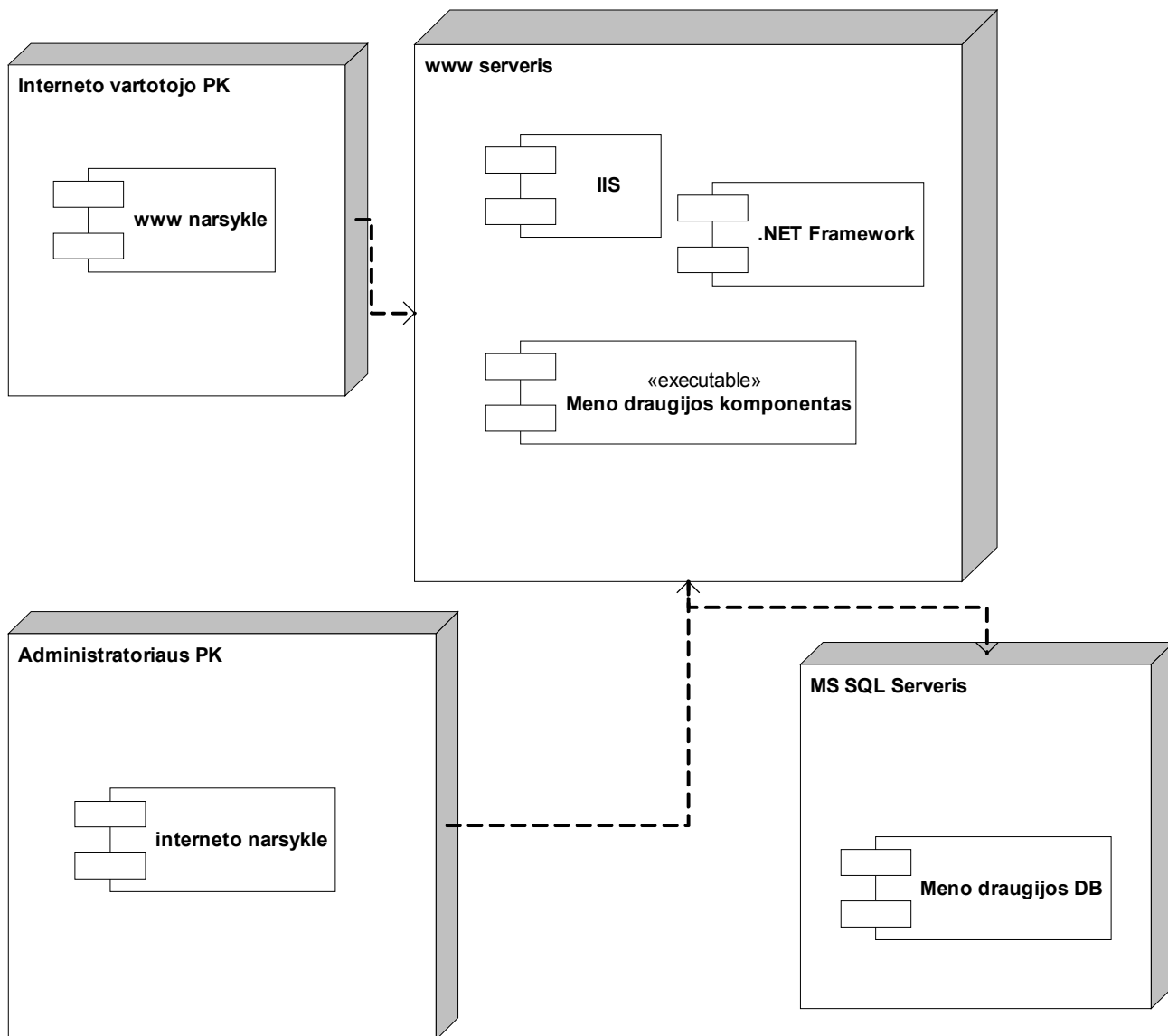
Ypatingi meno draugijos nariai yra menininkai. Papildomai jų asmens duomenims bei informacijai yra kuriamos lentelės „Menininkai“, „Paveikslai“, „Parodos“, „DalyvavimasParodose“. Į lentelę „Menininkai“ yra talpinami papildomi menininko asmens duomenys, kurie gali sudominti interneto ar registruotą vartotoją: biografija, kontaktai. Lentelėje „Paveikslai“ talpinami visi paveikslų duomenys, „Parodos“ – duomenys apie įvykusias ar vyksiančias parodas, „DalyvavimaiParodose“ – talpina duomenis apie konkretaus paveikslo dalyvavimą parodoje.

Atskira lentelė yra kuriama ir individualiems vartotojų užsakymams – komercinei sistemos daliai.



3 pav. Duomenų bazės struktūra MS Visio aplinkoje

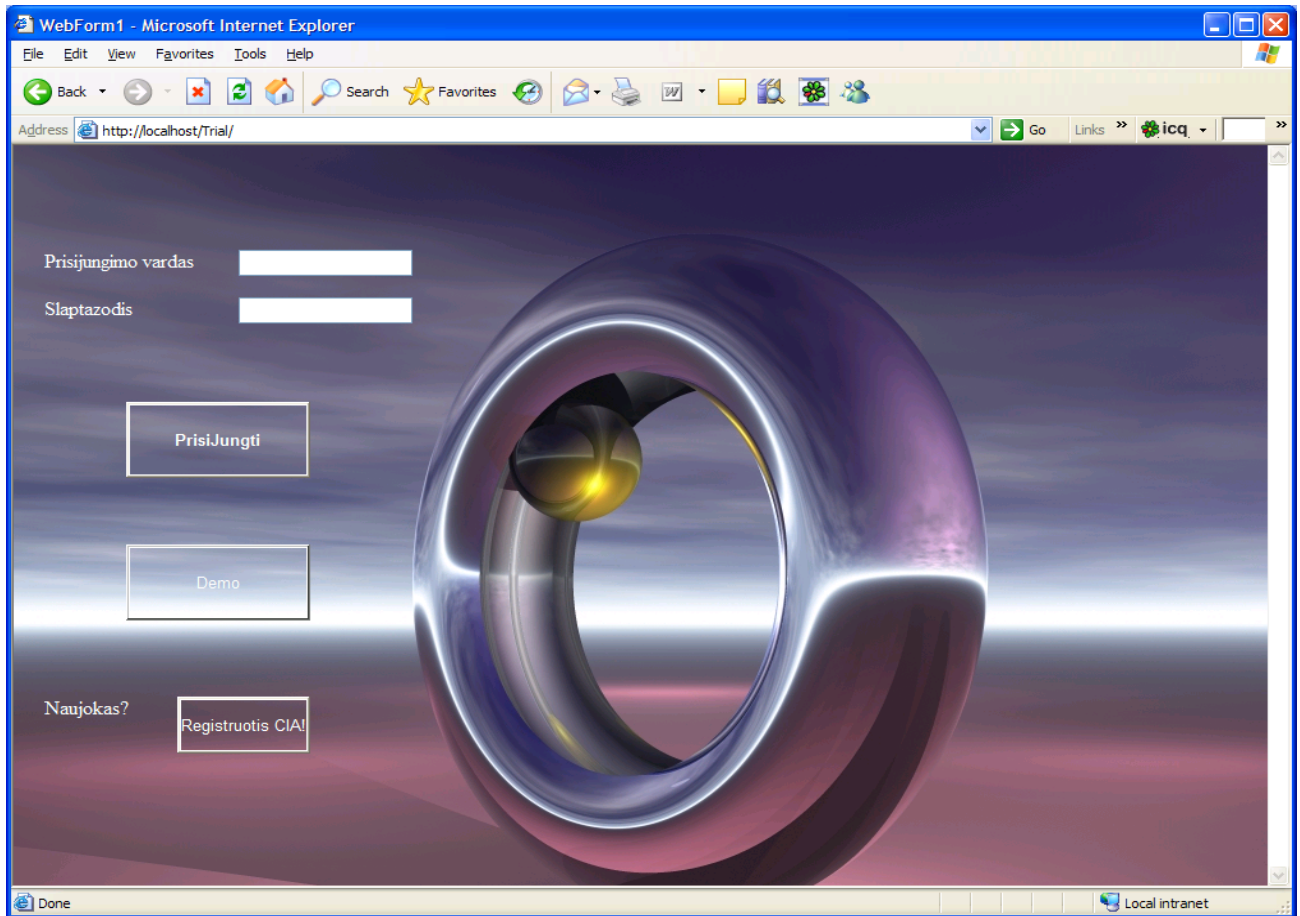
3.6. Programinēs ģrangos diegimo architektūra



4 pav. Programinēs ģrangos diegimo architektūra

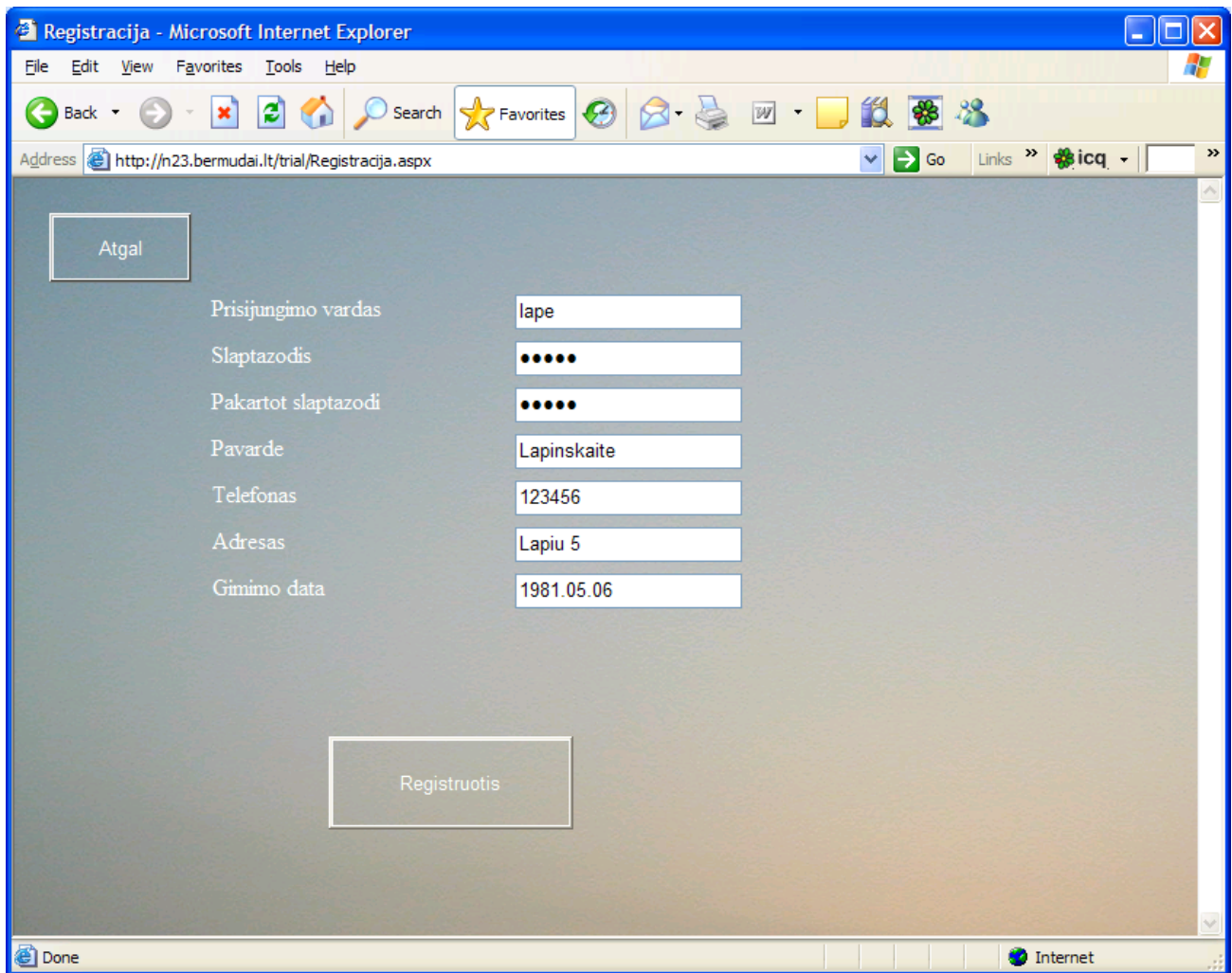
3.7. Sistemos realizacija ir veikimo analizė

Užklydęs į sistemą internete, vartotojas pirmiausia pamato pagrindinį prisijungimo puslapį, kuris atrodo štai taip:



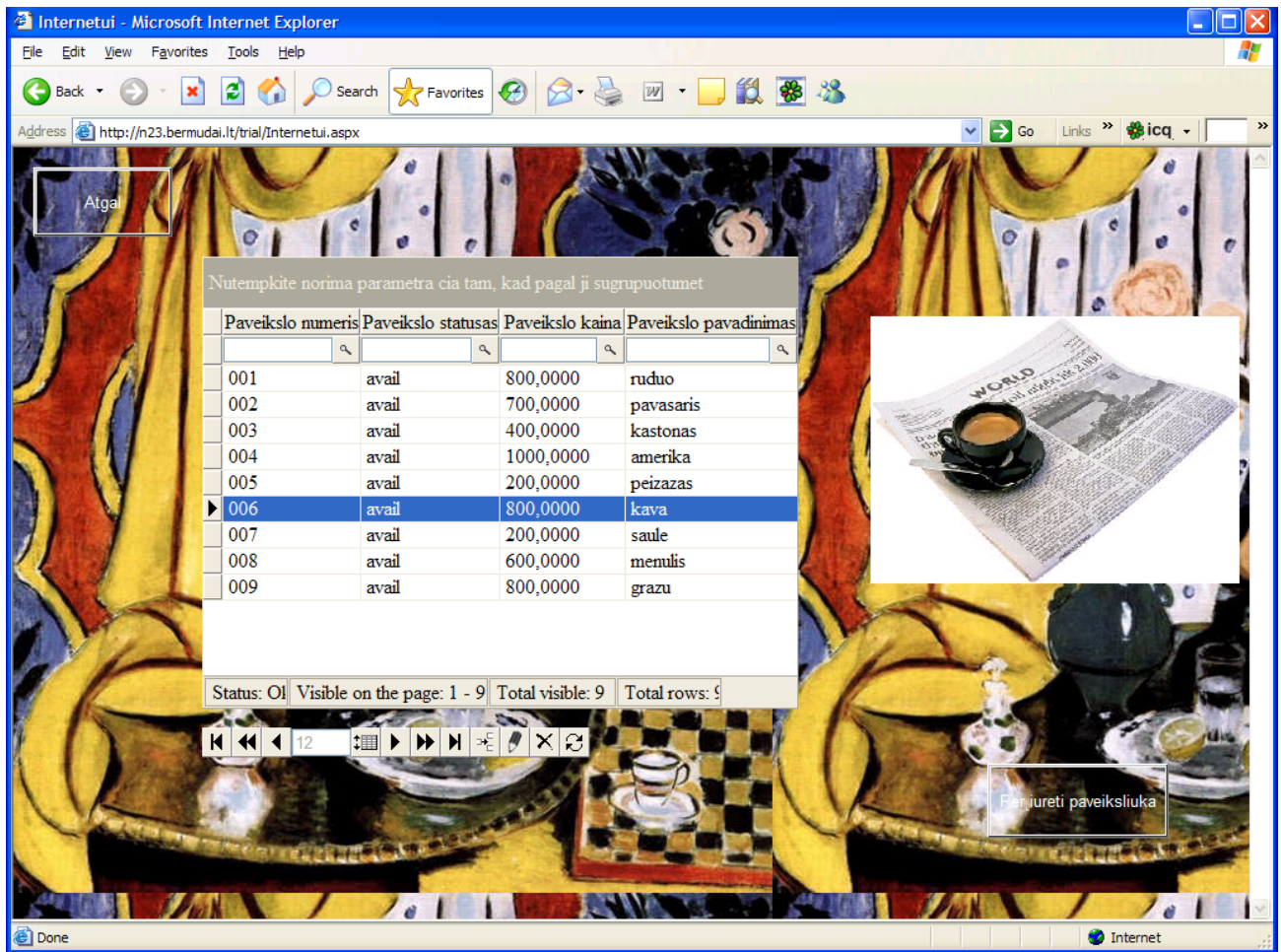
5 pav. Prisijungimo puslapis

Vartotojas šiame puslapyje gali rinktis iš trijų alternatyvų. Pirmoji – spausti mygtuką “Registruotis CIA” ir suvesti sistemos reikalaujamus duomenis. Antroji yra pasirinkti mygtuką “Demo” ir peržiūrėti neregistruotam vartotojui skirtą demonstracinę versiją. O trečioji prisijungti, naudojant mygtuką “PrisiJungti”, jei jau turimi virtualūs asmens duomenys(prisijungimo vardas ir slaptažodis), kurie į SQL duomenų bazę įtraukiami registruojantis. Prsijungiant turi būti suvestas tikslus prisijungimo vardas ir slaptažodis. Jei šie duomenys netikslūs, sistema neprijungia vartotojo ir perspėja apie įvykusią klaidą paragindama jungtis iš naujo.



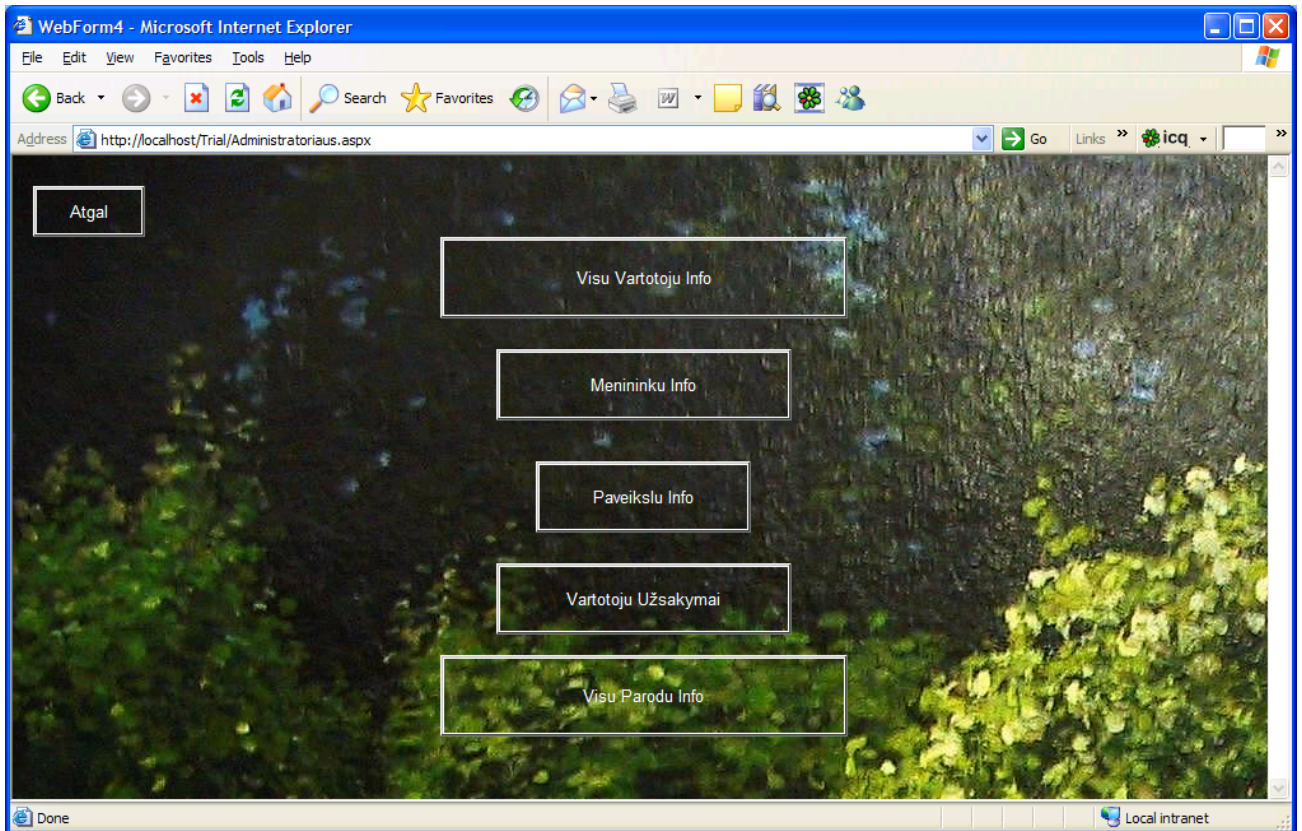
6 pav. Registracijos langas

Šitaip atrodo registracijos langas, į kurį papuolama iš pagrindinio puslapio. Norint gauti galimybę prisijungti sistemoje, reikia suvesti visus sistemos prašomus duomenis. Jei nors vienas laukas neužpildytas ir slaptažodžių laukai nesutampa, vartotojui pranešama, kad įvyko klaida ir jis paprašomas pataisyti įvestus duomenis.



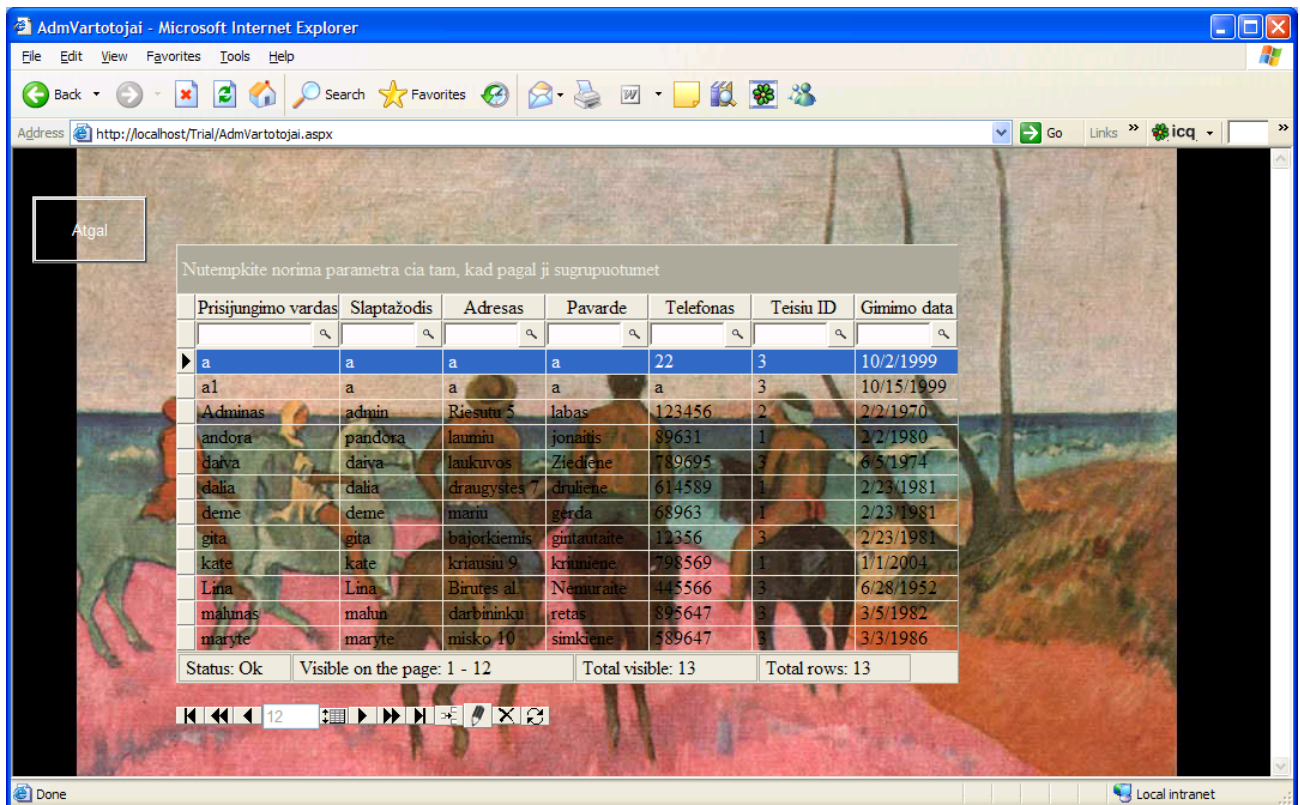
7 pav. Demonstracinė versija

Į šį puslapį patenka vartotojas, pasirinkęs prisijungimo lange “Demo” mygtuką. Čia jis gali peržiūrėti sistemoje esančią informaciją apie paveikslius, o pažymėjęs eilutę, su juo dominančiu paveikslu, ir peržiūrėti paveiksliuką. Dešinėje tam yra skirtas “Peržiūrėti paveiksliukus” mygtukas.



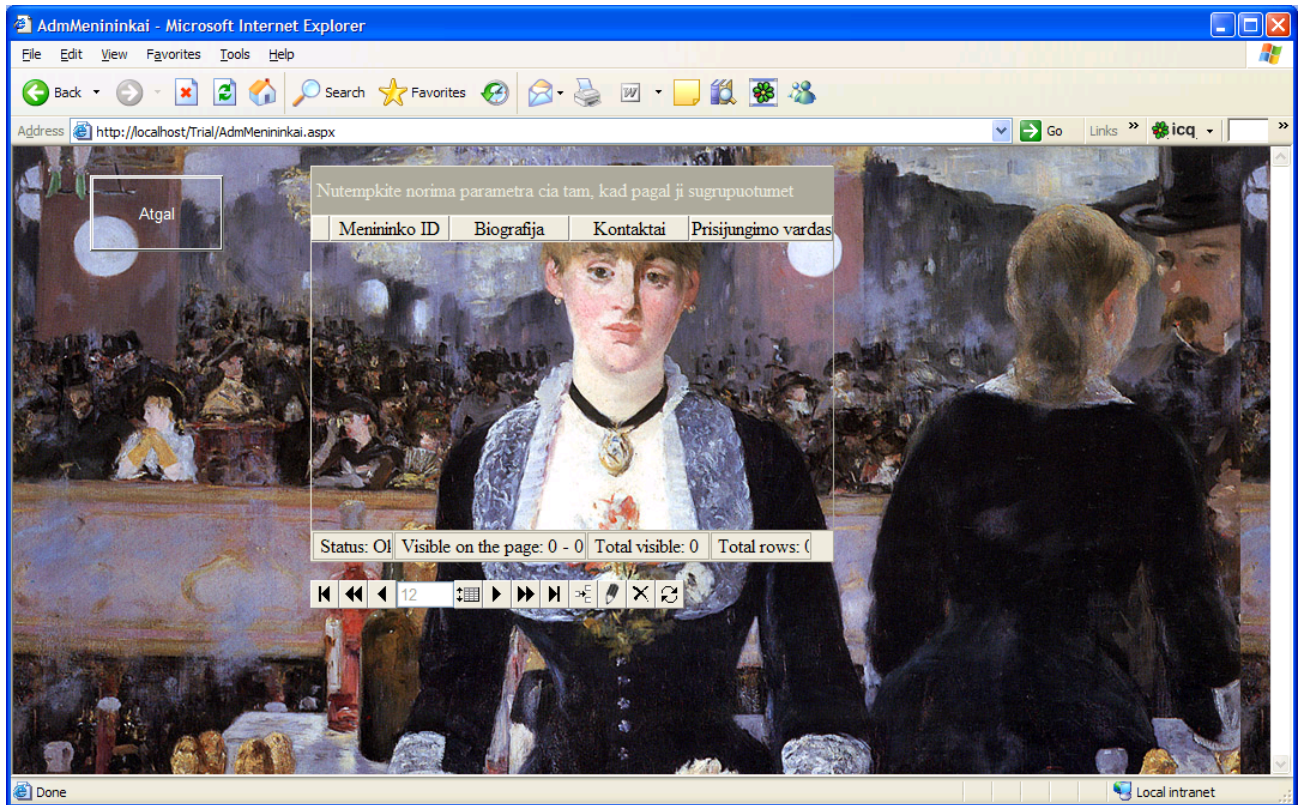
8 pav. Administratoriaus aplinka

Tokį puslapį mato prie sistemos prisijungęs administratorius. Jis gali pasirinkti, kokius duomenis nori peržiūrėti. Kiekvieno pobūdžio duomenis atspindi užrašai ant mygtukų. Jei administratorius nori administruoti visų sistemos vartotojų duomenis, jis renkasi pirmąjį mygtuką, jei menininkų – antrąjį, jei paveikslų informaciją – trečiąjį, jei vartotojų užsakymus ketvirtąjį ir jei parodų, tuomet paskutinį penktąjį mygtuką.

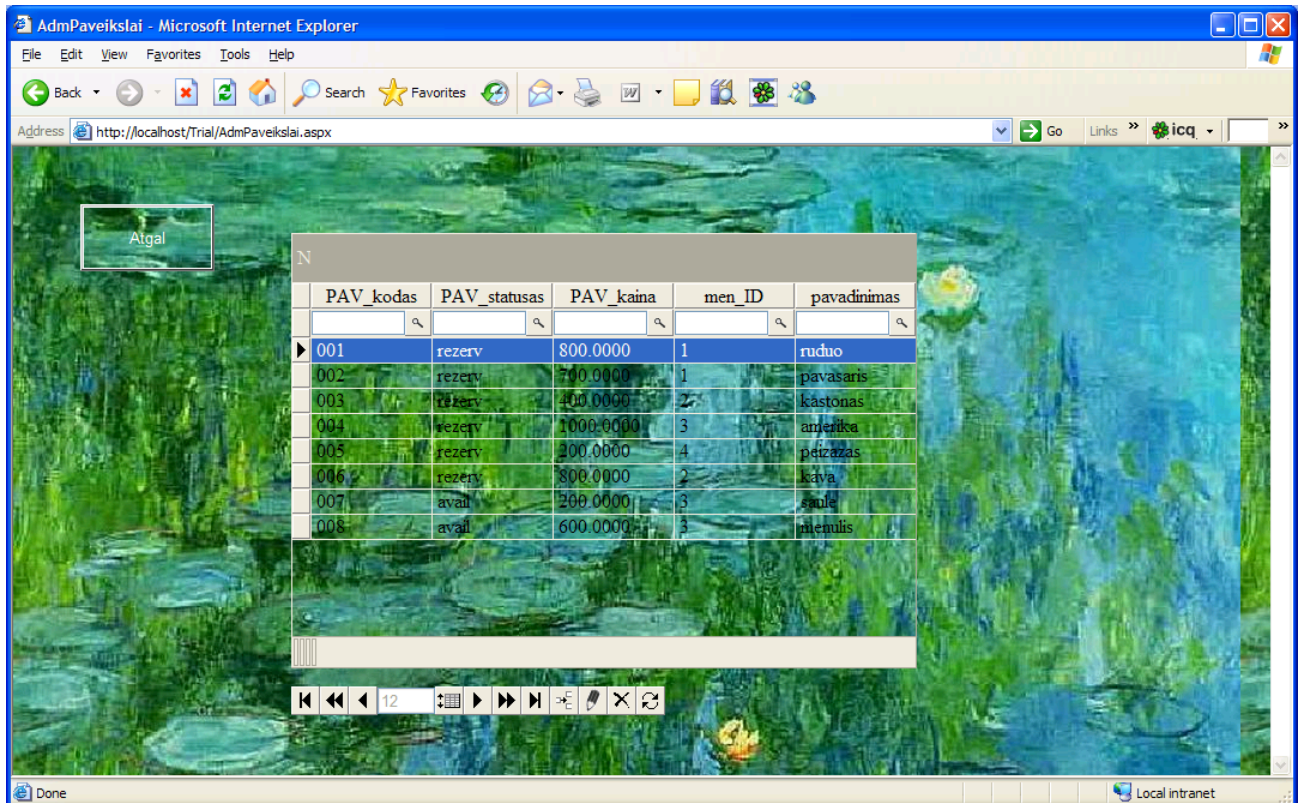


9 pav. Visų vartotojų puslapis

Šitame paveiksle pavaizduotas administratoriaus aplinkos visų vartotojų puslapis. Per šią sąsają administratorius vykdo visų vartotojų duomenų koregavimą SQL duomenų bazėje. Čia galimos visos SQL komandos: galima įterpti naują vartotoją, keisti, peržiūrėti ar trinti jau esamų vartotojų duomenis. Taip pat administratorius gali atlikti paiešką virš norimo stulpelio įrašęs ieškomą žodį bei paspaudęs ikoniukę šalia. Toliau vaizduojami administratoriaus langai (7 pav., 8 pav., 9 pav., 10 pav.) savo funkcionalumu nėra nesiskiria nuo šio puslapio. Visuose juose esančius duomenis administratorius gali keisti per šią sąsają.



10 pav. Visų menininkų puslapis



11 pav. Visų paveikslų puslapis

AdmUzsakymai - Microsoft Internet Explorer

Address: http://localhost/Trial/AdmUzsakymai.aspx

Atgal

Nutempkite norima parametra cia tam, kad pagal ji sugrupuotumet

Uzsakymo numeris	Paveikslslo numeris	Statusas	Uzsakiusiojo vardas
10	003	False	maryte
11	004	False	maryte
12	002	False	maryte
13	006	False	maryte
14	006	False	maryte
15	002	False	maryte
16	003	False	vilkas
17	007	False	vilkas

Status: Ol Visible on the page: 1 - 8 Total visible: 8 Total rows: 8

Done Local intranet

12 pav. Visų užsakymų puslapis

AdmParodos - Microsoft Internet Explorer

Address: http://localhost/Trial/AdmParodos.aspx

Atgal

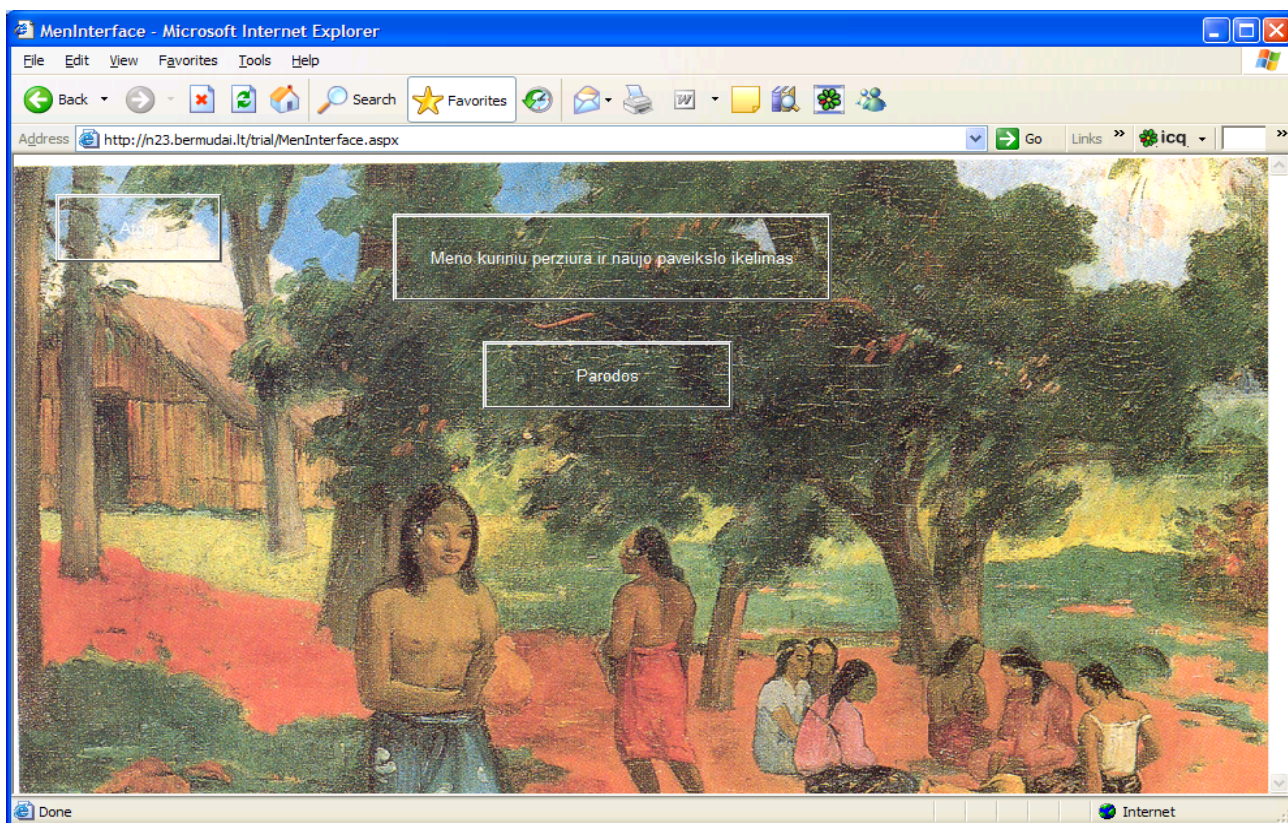
Nutempkite norima parametra cia tam, kad pagal ji sugrupuotumet

Parodos Numeris	Parodos data	Parodos pavadinimas	Aprašas
1	10/23/2000	modernu	ispudinga
2	11/30/2000	peizazai	nebloga

Status: Ol Visible on the page: 1 - 2 Total visible: 2 Total rows: 2

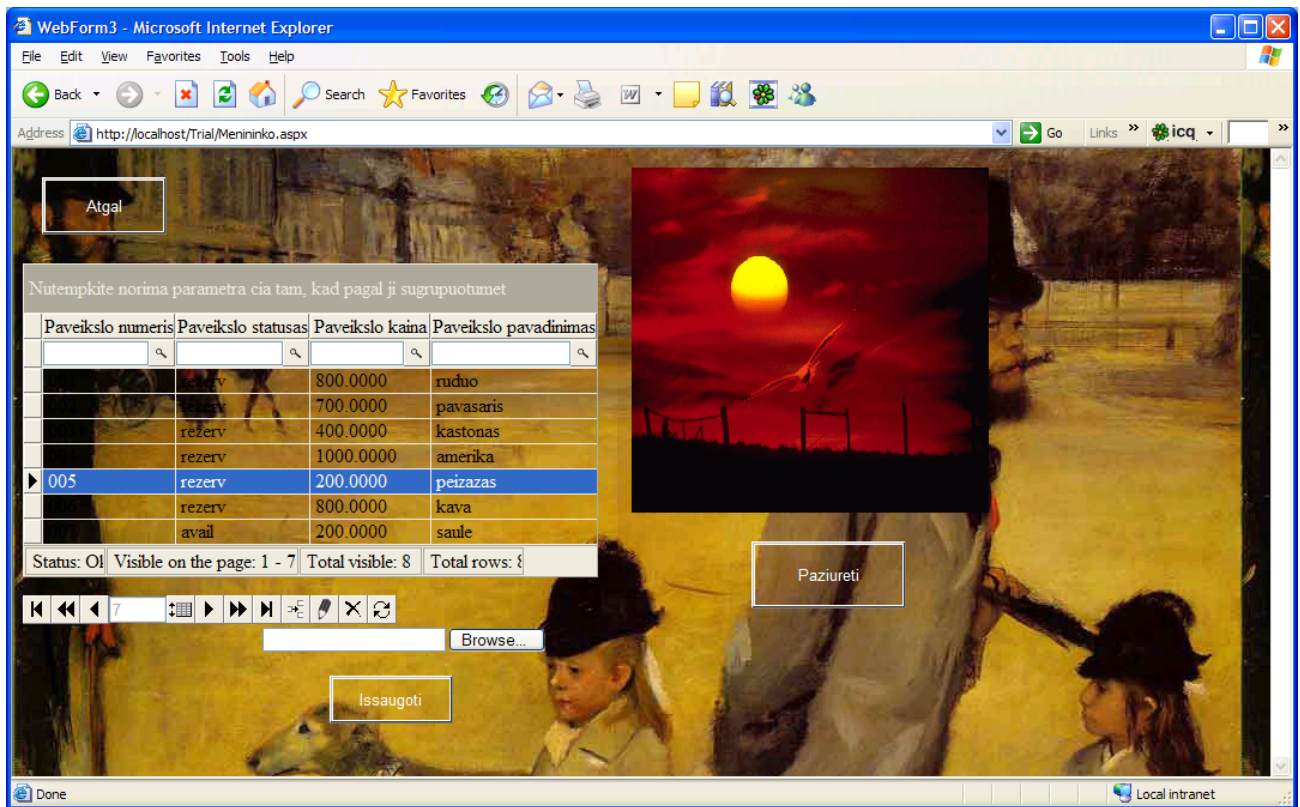
Done Local intranet

13 pav. Visų parodų puslapis

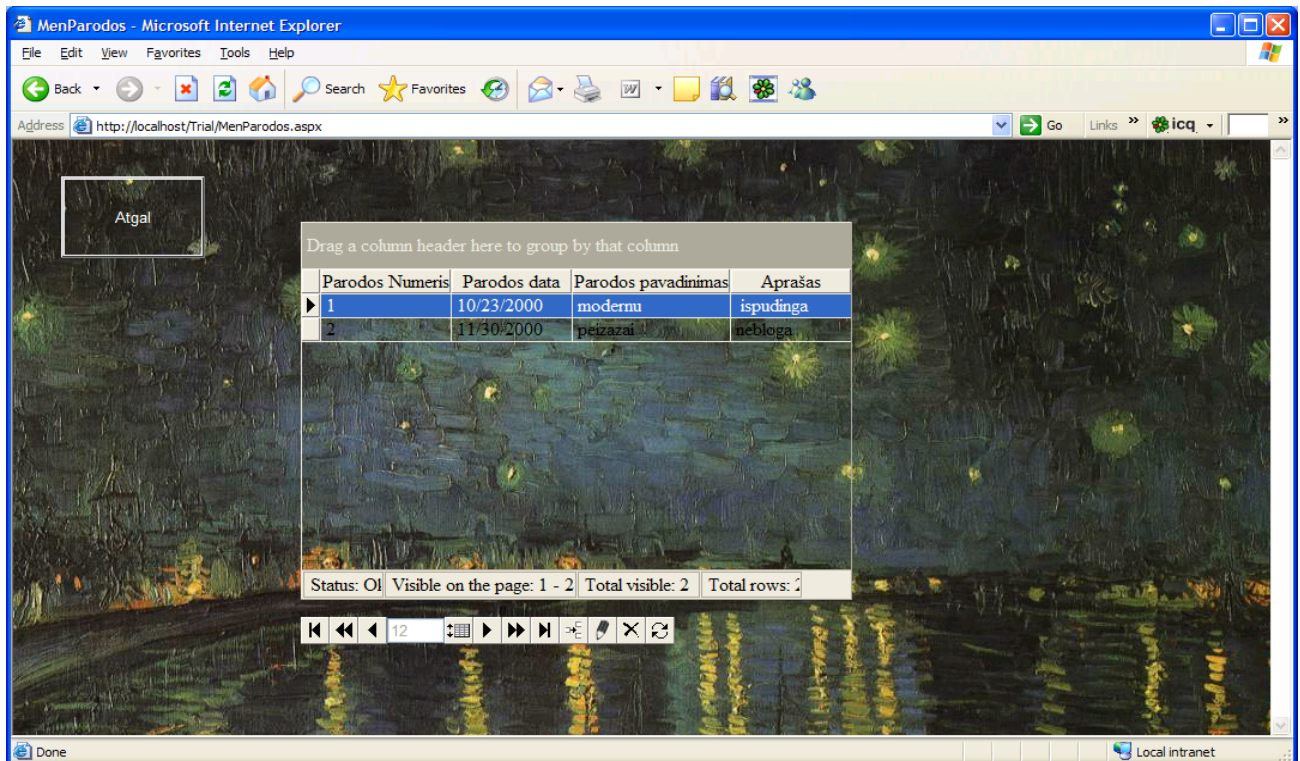


14 pav. Menininko aplinka

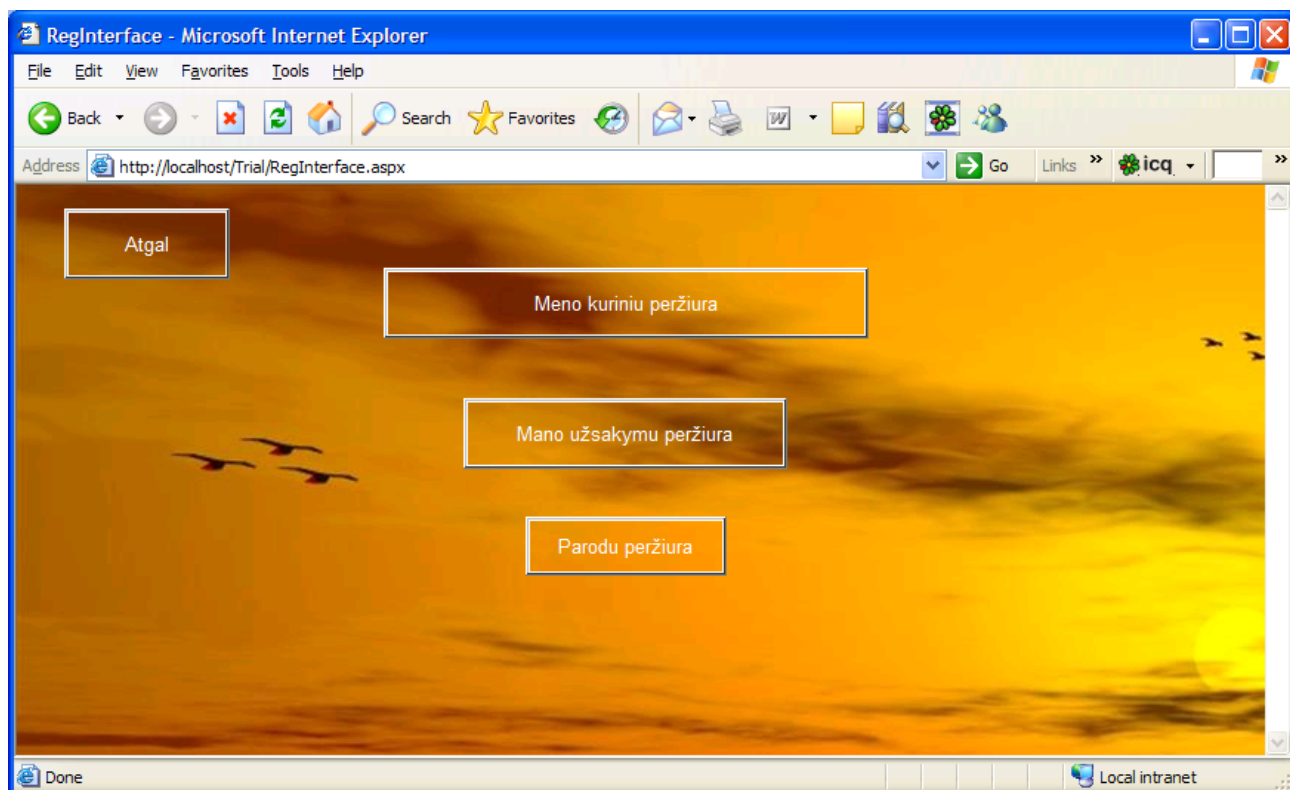
Į šį puslapį patenka prisiregistravęs menininkas. Jis turi dvi alternatyvas: gali peržiūrėti duomenų bazėje esančius paveikslus su pilnais aprašais bei peržiūrėti jų pavyzdžius. Ten pat gali įkelti naują paveikslą(12 pav.). Arba gali pasirinkti mygtuką “Parodos” bei patekti į parodų puslapį(13 pav.). Šiame puslapyje menininkas nieko negali keisti. Parodas skelbia ir aprašinėja meno draugijos administracija.



15 pav. Menininko paveikslo įkėlimo puslapis



16 pav. Menininko parodų puslapis



17 pav. Registruoto vartotojo puslapis

Taip atrodo registruoto vartotojo puslapis, kuriame jis gali pasirinkti iš trijų mygtukų: “Meno kūrinį peržiūra”, “Mano užsakymų peržiūra” ir “Parodų peržiūra”.

Algal

Nutempkite norima parametra cia tam, kad pagal ji sugrupuotumet

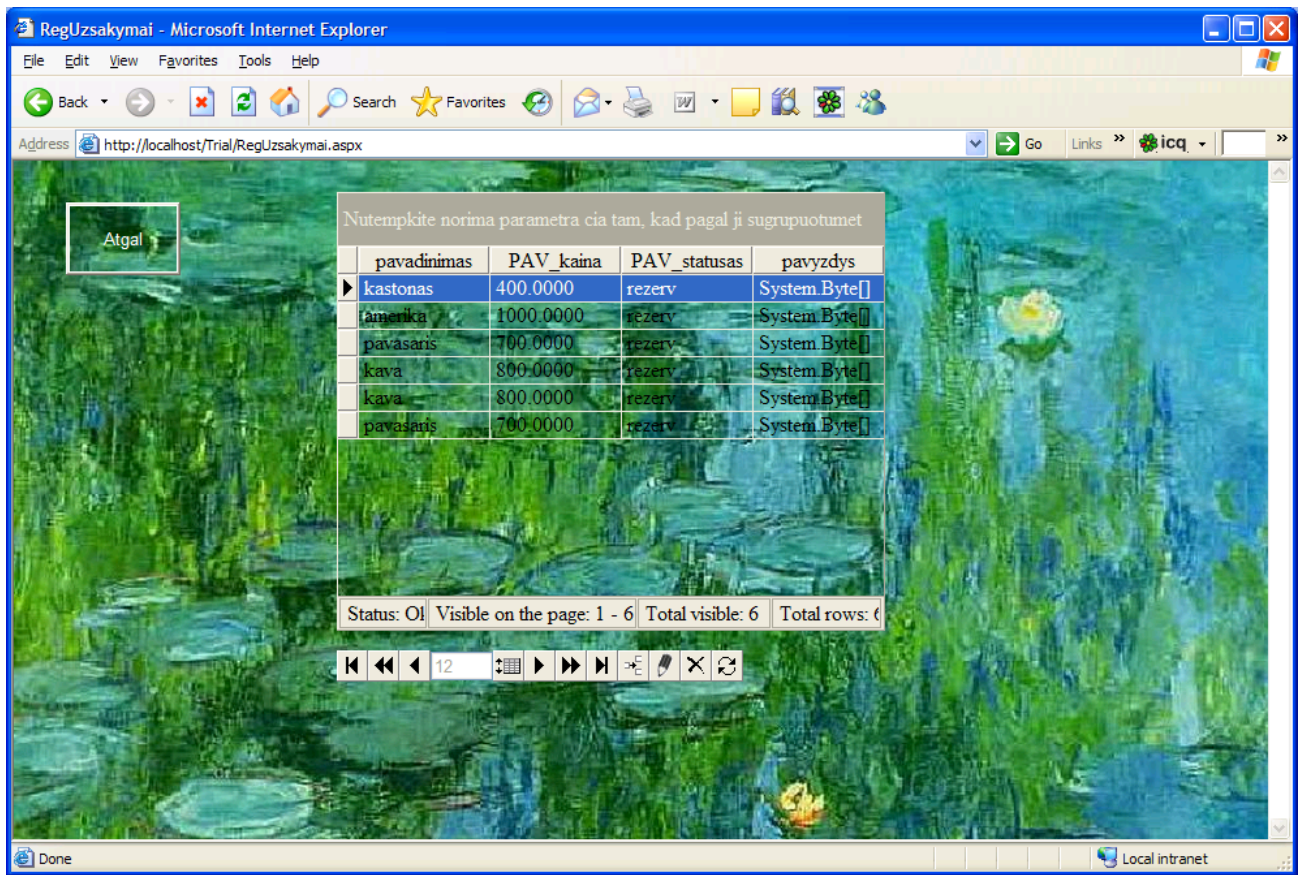
Paveikslo numeris	Paveikslo statusas	Paveikslo kaina	Paveikslo pavadinimas	Paveikslo pavyzdys
001	rezerv	8000000	duo	System.Byte[]
002	rezerv	7000000	duo	System.Byte[]
003	rezerv	4000000	duo	System.Byte[]
004	rezerv	10000000	amerika	System.Byte[]
005	rezerv	2000000	perlas	System.Byte[]
006	rezerv	8000000	kava	System.Byte[]
007	avali	2000000		System.Byte[]
008	avali	6000000		System.Byte[]

Status: Ok Visible on the page: 1 - 8 Total visible: 8 Total rows: 8

Užsakyti Peržiureti

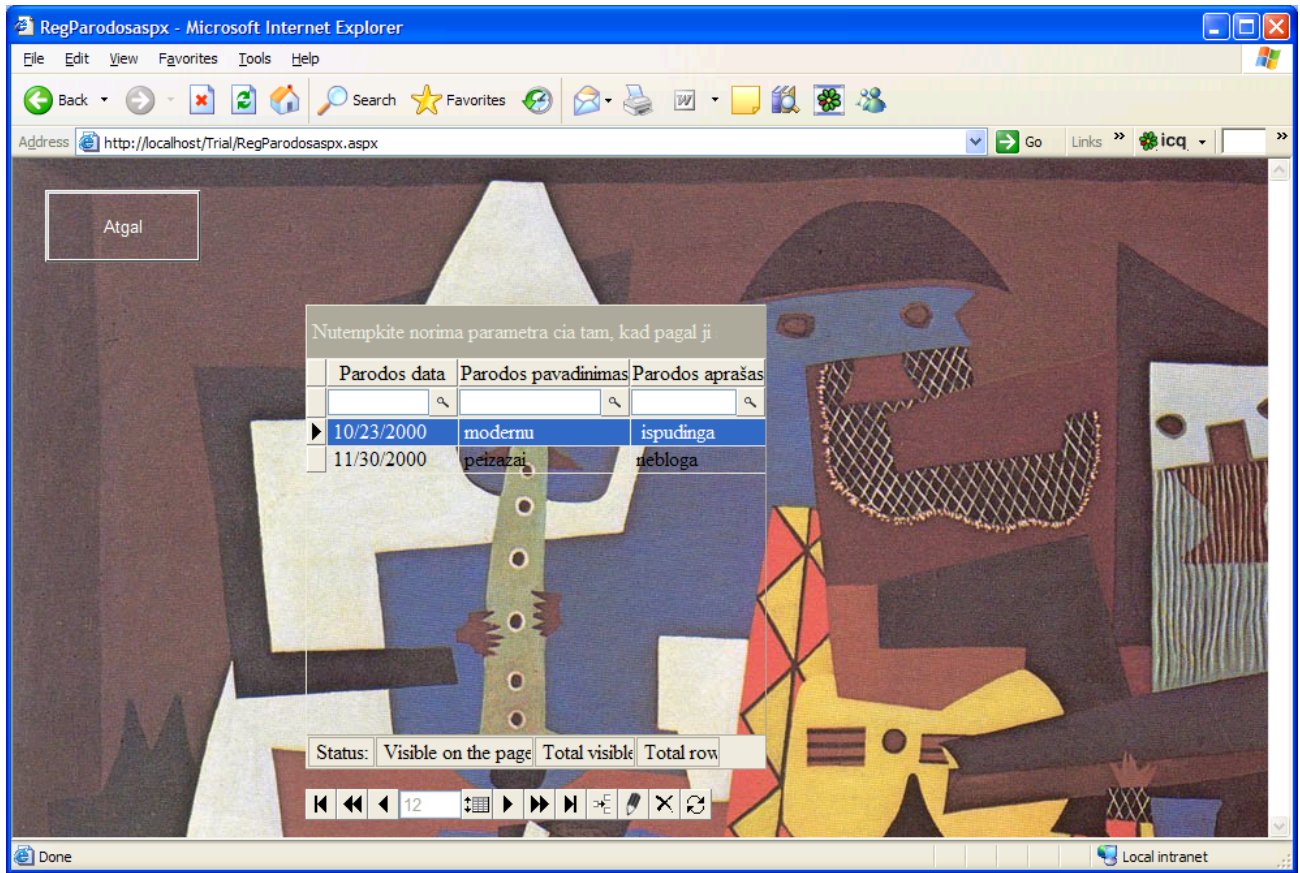
18 pav. Registruoto vartotojo paveikslų peržiūros puslapis

Šiame puslapyje registruotas vartotojas gali pilnai peržiūrėti visą informaciją apie paveikslus. Jis jos negali pakeisti, tik gali užsisakyti patikusį ir dar nerezervuotą paveikslą. Užsakyto paveikslo statusas pasikeičia į rezervuoto ir sistema nebeleidžia kitam vartotojui pakartotinai užsisakyti to paties paveikslo. Bandant tai padaryti sistema perspėja, kad paveikslas užsakytas.



19 pav. Registruoto vartotojo užsakymų puslapis

Taip atrodo užsakymų puslapis. Šiame puslapyje vartotojas gali peržiūrėti savo užsakymus. O kitame paveiksle (17 pav.) pavaizduota, kaip atrodo registruoto vartotojo parodų peržiūros puslapis. Šiame puslapyje vartotojas gali peržiūrėti vykusias ar vykiančias parodas.



20 pav. Registruoto vartotojo parodų puslapis

4. VARTOTOJO DOKUMENTACIJA

Vartotojo dokumentacija susideda is trijų poskyrių. Kiekvieno jų paskirtis yra padėti vartotojui tik trim skirtingais aspektais. Pirmasis poskyris “Sistemos funkcinis aprašymas” yra skirtas bet kuriam vartotojui, kuriam įdomu, ar sistema atitinka jo poreikius. Jame yra glaustai aprašomos sistemos galimybės ir paskirtis.

“Sistemos vadovas” yra išsamesnis. Jame detaliau aprašomos sistemos vykdomos funkcijos bei kokiomis priemonėmis tas vykdymas yra užtikrinamas. Taip pat aptariamos dažniausios klaidos sistemoje bei kaip jos apdorojamos.

“Sistemos instaliavimo dokumente” aptariama, kokios techninės ir programinės priemonės yra reikalingos, kad vartotojas galėtų ne tik naudotis sistema, bet ir išnaudoti visus jos teikiamus privalumus.

4.1 Sistemos funkcinis aprašymas

Sistema yra skirta meno draugijai, kuri yra tiek informatyvi (daug informacijos apie menininkus, jų kūrybą bei parodas), tiek ir komercinė (patikusį darbą galima užsisakyti čia pat internete). Sistemoje galima registruotis, prisijungti, peržiūrėti informaciją, atlikti paiešką, koreguoti pagal savo teises atitinkamus duomenis sistemos duomenų bazėje.

4.2 Sistemos vadovas

Sistemos duomenų bazė yra peržiūrima *DevExpress ASPxGrid* pagalba. Jame galima ne tik peržiūrėti, bet ir koreguoti duomenis: galima įterpti naują eilutę, ištrinti įrašą, pakeisti ar papildyti. Taip pat pagal bet kokį duomenų stulpelyje esantį reikšminį žodį atliekama paieška. Priėjimo prie duomenų saugumas yra sprendžiamas prisijungiant, kuomet pagal atitinkamą prisijungimo vardą ir slaptažodį yra užkraunami skirtingi vartotojo aplinkos langai, o programiniame kode sesijos būdu kiekviename puslapyje yra tikrinami tie patys virtualūs duomenys ir jiems atitikus užkraunami kai kurie individualūs duomenys, kuriuos vartotojas pagal savo teises gali taisyti arba ne. Visi duomenys yra laikomi MS SQL Server 2000 duomenų bazėje. Be duomenų koregavimo, menininkai čia pat internete gali įkelti savo paveikslų pavyzdį bei aprašą, o registruoti vartotojai suformuoti užsakymą. Tikslus kiekvienam vartotojui prieinamų sistemos funkcijų aprašas yra detaliame panaudojimo atveju (3.1 skyrelis) bei sistemos realizavimo (3.7 skyrelis) aprašuose bei pavyzdžiuose.

Galimos klaidos:

- Prisijungiant įvedamas neegzistuojantis prisijungimo vardas ir/arba slaptažodis. Tuomet sistema perspėja apie tai klaidos pranešimu.
- Registruojantis įvedami ne visi duomenys. Sistema praneša apie klaidą.
- Registruojantis į slaptažodžių laukus įvedami slaptažodžiai nesutampa. Sistema praneša apie klaidą.
- Vartotojas per ilgai būna sistemoje neaktyvus ar bando tiesiogiai pasiekti jam pagal teises nepriklausantį puslapį. Sistema nukreipia jį į klaidos puslapį, kuriame paragina prisijungti iš naujo.
- Vartotojas bando užsisakyti jau rezervuotą paveikslą. Sistema perspėja, kad negali to padaryti dėl netinkamo užsakymui paveikslų statuso.

4.3 Sistemos instaliavimo dokumentas

Sistema galima naudotis tik tuo atveju, jei kompiuteryje yra instaliuotas IIS ir .NET Framework. Taip pat yra tam tikrų reikalavimų techninei įrangai:

Reikalavimai techninei įrangai:

Procesorius: 450-megahercų (MHz) Pentium II-class processor, 600-MHz Pentium III-class processor rekomenduojamas

Taikomosios programos gali būti įdiegtos šiose sistemose:

13 lentelė. Operacinės sistemos

Windows Server 2003
Windows XP Professional
Windows XP Home Edition
Windows 2000 (Service Pack 2 rekomenduotinas)
Windows Millennium Edition (Windows Me)
Windows 98
Microsoft Windows NT® 4.0 (Service Pack 6a reikalingas)

Atmintis:

14 lentelė. Reikalinga atmintis

Windows Server 2003	160 MB RAM
Windows XP Professional	160 MB RAM
Windows XP Home Edition	96 MB RAM
Windows 2000 Professional	96 MB RAM
Windows 2000 Server	192 MB RAM

Kietasis diskas: 900 MB tuščios vietos sistemai, 3.3 gigabaitų (GB) tuščios vietos instaliavimui.

Papildomi 1.9 GB tuščios vietos reikalingi neprivalomai MSDN bibliotekai.

Diskas (drive): CD-ROM ar DVD-ROM.

Displėjus: Aukštos VGA (1024 x 768) ar dar didesnės rezoliucijos displėjus su 256 spalvomis
Pelė ar kitas nurodantis įrenginys.

Internetas, kadangi IS yra orientuota į bet kurį žmogų, kurio buvimo vieta nėra apibrėžta.

5. PRODUKTO KOKYBĖS ĮVERTINIMAS

Sistemos kokybę tyrė 3 ekspertai. Jie turėjo galimybę prisijungti prie sistemos kelių dienų bėgyje, peržiūrėti kūrinius, pateikti užsakymus ir pan. Buvo vertinami nefunkciniai ir registruotam vartotojui aktualūs funkciniai reikalavimai 5 balų sistemoje:

- 5 – puikiai tenkina
- 4 – tenkina
- 3- neutrali nuomonė
- 2- silpnai
- 1- netenkina

15 lentelė. PĮ kokybės įvertinimas

Reikalavimas	Ekspertas	Balas
Panaudojamumas	1	5
	2	4
	3	5
	Vidurkis	4,7
Estetinė išvaizda	1	5
	2	5
	3	5
	Vidurkis	5
Vaizdų rodymo greitis	1	4
	2	5
	3	4
	Vidurkis	4,3
Naudojimo lengvumas	1	4
	2	3
	3	5
	Vidurkis	4
Patogumas	1	5
	2	5
	3	5
	Vidurkis	5
Užsakymų sudarymas	1	5
	2	5
	3	5
	Vidurkis	5
Prisijungimas	1	5
	2	5
	3	5
	Vidurkis	5

Vertinime dalyvavo ekspertai:

Pavardė, vardas	Organizacija, pareigos
Nemuraitė Lina	KTU, doc.
Čeponienė Lina	KTU, doktorantė
Balandytė Milda	KTU, doktorantė

Apibendrinant rezultatus, galima padaryti išvadą, kad vartotojų tenkinimo vidurkis yra labai aukštas, jokių priekaištų nebuvo estetinei išvaizdai, užsakymų sudarymui ir prisijungimui.

6. IŠVADOS

1. Norint sukurti Meno draugijos informacinę sistemą internete, buvo atlikta panašių sistemų analizė ir nustatytos funkcijos, kurias turėtų leisti vykdyti ši sistema. Tai informacijos apie meno kūrinius ir menininkus teikimas, meno kūrinių užsisakymas, taip pat įvairios informacijos tvarkymo funkcijos.

2. Skirtingi vartotojai (menininkai, administratorius, registruoti ir neregistruoti vartotojai) sistemoje gali atlikti skirtingas funkcijas, kurių naudojimas apribotas slaptažodžiais.

3. Sistemai realizuoti pasirinktos pažangios NET technologijos: Visual C#, ASP.NET, ADO.NET, MS SQL Server 2000, kurios leidžia paprastai ir greitai kurti interneto sistemas, naudojančias ir apdorojančias duomenų bazėse saugomus duomenis.

4. Tiksliems vartotojo reikalavimams nustatyti reikalavimai buvo įvardinti natūralia kalba, pavaizduoti panaudojimo atvejų diagrama, aprašyti, o kai kurios kritinės sistemos dalys aprašytos Z specifikavimo kalba.

5. Suformuluota projekto užduotis pilnai įvykdyta. Darbe pateiktos diagramos, vaizduojančios projekto elementus: vartotojo sąsaja, duomenų bazė, programinių komponentų išdėstymą. Sukurta duomenų bazė ir programinė įranga, atliktas testavimas. Pateikti sistemos vaizdai, iliustruojantys vartotojo darbą su sistema.

6. Šios sistemos privalumas tame, kad ji skirta tam tikros visuomenės dalies poreikiams menui bent dalinai patenkinti, suteikti galimybę didesniai jos narių skaičiui patekti į meno pasaulį, kadangi didelis gyvenimo tempas dažnai neleidžia tų poreikių patenkinti kitais būdais. Be to, sistema teikia ne tik informaciją, bet ir paslaugas, kontaktinę informaciją. Tokia sistema galėtų išaugti į Meno draugijos komunikacinį portalą.

7. Naudojama tokia sistema turėtų būti plečiama, atsižvelgiant į naujai atsiradusius poreikius. Todėl buvo siekiama sukurti tokią įrangą, kuri būtų nesunkiai modifikuojama ir greitai nepasentų, tam pasirinktos pažangios technologijos, sukurta nesudėtinga ir patogi sąsaja.

7. LITERATŪRA

1. Bagdonaitė B., Nemuraitė L. Panaudojimo atvejų atvaizdavimas Z specifikacija// Informacinės technologijos: tarpuniversitetinės konferencijos pranešimų medžiaga [Kaunas, 2005 m. balandžio 26]. Kaunas, 2005, p. 209-213.
2. Jacobson I. Use Cases – Yesterday, Today and Tomorrow. Rational Software, 2002. p.15.
3. Microsoft Corporation. Visual Studio Home. – [žiūrėta 2005-01-15]. Prieiga per internetą: <http://msdn.microsoft.com/vstudio/>
4. Sekliuckis V., Gudas S., Garšva G. Informacijos sistemos ir duomenų bazė: vadovėlis. K.:Technologija, 2003. 338p.
5. Spivey J. The Z Notation: Reference Manual. Oxford: 1998. p.168

8. PRIEDAI

9.1 PRIEDAS. Straipsnis

PANAUDOJIMO ATVEJŲ ATVAIZDAVIMAS Z SPECIFIKACIJA

Brigita Bagdonaitė

KTU Informacijos sistemų katedra, Studentų 50-308 tel. +370-37-300397

Panaudojimo atvejai plačiai taikomi sistemos bei vartotojo funkciniam reikalavimams apibrėžti. Tačiau jie neapsaugo nuo klaidų bei skirtingo interpretavimo. Dviprasmybių išvengti padeda formalūs matematiniai metodai. Turint neformalų aprašą ir naudojantis keletu paprastų taisyklių, nesunku šį aprašą formalizuoti ir išvengti galimų nesupratimų tolimesniame sistemos kūrimo procese. Šiame darbe parodyta, kaip panaudojimo atvejus galima atvaizduoti Z specifikacija.

IVADAS

Nuo tada, kai 1986 metais Jacobson pasiūlė panaudojimo atvejus (use cases), funkciniam sistemos bei vartotojo reikalavimams pavaizduoti ir išsiaiškinti, panaudojimo atvejų koncepcija buvo tobulinama, ir tai, kas anuomet prasidėjo kaip teorija, 1992 metais jau turėjo nemažą praktinį pagrindą [2]. Panaudojimo atvejų populiarumą tiek tada, tiek šiandien, lėmė jų paprastumas, aiškumas bei tinkamumas visiems funkciniam vartotojo ir sistemos reikalavimams apibrėžti.

PANAUDOJIMO ATVEJAI

Kaip būtų galima apibrėžti panaudojimo atvejus? Ar yra tam tikras panaudojimo atvejų skaičius, kuris garantuotų tobulą veiklos aprašą? Paprasčiausias panaudojimo atvejų apibūdinimas – tai yra tiesiog tam tikrų veiksmų seka, kurią vykdo aktorius kartu su sistema [2], [3], [6]. Nuo to laiko šio apibrėžimo supratimas šiek tiek pasikeitė, tačiau ne jo esmė. O dėl panaudojimo atvejų skaičiaus Jacobson turi savo nuomonę. Nors jis mano, kad didelės organizacijos veiklai aprašyti turėtų užtekti 20 panaudojimo atvejų, tačiau su šiuo skaičiumi elgiasi atsargiai. Jis pats sako, kad konkretaus skaičiaus paskelbimas gali klaidinti analitikus, būtų imta dirbtinai mažinti ar didinti panaudojimo atvejų skaičių konkrečiu atveju vien tam, kad „pritempti“ iki tobulo 20. Todėl Jacobson siūlo remtis ne skaičiumi, o kokybe ir konkretumu. Svarbiausia yra tiksliai apibrėžti, ką konkretus aktorius duoda sistemai ir ką iš jos gauna.

Šiandien panaudojimo atvejai tapo universalios modeliavimo kalbos UML (Unified Modeling Language) dalimi. Kadangi UML yra tiksliai apibrėžta, panaudojimo atvejų reikšmė bei su tuo susijusios sąvokos taip pat yra aiškiai apibrėžtos. Ir nors ankstesnis apibrėžimas, kad „panaudojimo atvejis yra tam tikra sistemos vykdoma veiksmų seka, duodanti vertingą rezultatą vienam ar daugiau sistemos aktorių“ nepaseno, yra ir tikslesnis apibrėžimas [6]: panaudojimo atvejis yra klasifikatorius, kuris apibrėžia tam tikrą elgseną, kurią gali atlikti subjektas (sistema, jos dalis, klasė ir pan.), sąveikaudamas su vienu ar daugiau aktorių; panaudojimo atvejis gali turėti praplėtimo taškus bei „extend“ ir „include“ ryšius su kitais panaudojimo atvejais.

Panaudojimo atvejus galima skirstyti į:

- Konkretius (concrete) panaudojimo atvejus;
- Apibendrinančius (generalization) panaudojimo atvejus, kurie palaiko pakartotinį panaudojimo atvejų panaudojimą;
- Išplėtimo (extension) panaudojimo atvejus, kurie prideda funkcionalumo esamam panaudojimo atvejui nepakeisdamas pradinio jo varianto;

- Priklausomus (inclusion) panaudojimo atvejus, kurie prideda funkcionalumo kitiems panaudojimo atvejams bei juos pakeičia.

Tam, kad sukurtume panaudojimo atvejį, reikia apibrėžti sistemą, išskirti aktorių, kuris yra tam tikros rūšies klasifikatorius, vaizduojantis esybes, esančias subjekto (sistemos) išorėje bei turintis binarines asociacijas su panaudojimo atvejais, subsistemomis, komponentais ar klasėmis, bei aprašyti panaudojimo atvejus, kurių apibrėžimas įvardintas aukščiau. Naujausia UML 2.0 versija įgalina aprašyti panaudojimo atvejus būsenų mašina bei tiesiogiai jį susieti su subjektu (sistema, kuriai priklauso panaudojimo atvejis), o tai suteikia daugiau galimybių modeliuoti siekiamą ar esamą sistemos funkcionalumą.

Tačiau panaudojimo atvejai neapsaugo nuo galimų klaidų, kurios gali kilti dėl dviprasmiškai suprastų reikalavimų, galimų skirtingų interpretacijų „savime suprantamų“ faktų praleidimo. Tokiais atvejais reikia formalaus metodo, kuris leistų panaikinti neapibrėžtumus ir išgauti visišką reikalavimų aiškumą [1], [4]. Gali būti, kad vėliau specifikacija bus keičiama, nes sistemos kūrimo procese gali išryškėti nauji faktai ar reikalavimai, tačiau ji padeda suprasti pirminį sistemos ir reikalavimų jai vaizdą, o tai labai svarbu.

Z NOTACIJA

Formali Z specifikacija, naudinga kompiuterinėms sistemoms, yra pagrįsta Zermelo-Fraenkel aibių teorija bei predikatų logika [5]. Apskritai formali specifikacija apibūdina, iš ko informacinė sistema privalo susidėti, tačiau nenurodo, koku būdu tai turi būti pasiekta. Ji nurodo, ką sistema darys, bet nenusako kaip. Toks specifikacijos abstraktumas naudingas, kadangi leidžia suvokti, ko norima iš sistemos, be to, gauname pirminį sistemos vaizdą, kol sistema dar nėra pradėta kurti. Tam nereikia klaidžioti po programos kodą ar stengtis išgauti tikslus reikalavimus iš beletristinio teksto.

Z specifikacija yra smulkinama į mažesnes dalis – schemas. Prie kiekvienos schemas gali būti neformalus komentaras, kuris ją paaiškina ir padeda greičiau susiorientuoti visoje sistemos specifikacijoje. Z schemas yra naudojamos tiek statiniams, tiek dinaminiam aspektams apibūdinti. Statiniai aspektai apima būsenas, į kurias sistema gali pereiti, bei ryšius, kurie išlieka pereinant iš vienos būsenos į kitą, o dinaminiai apsketai apima visas galimas operacijas, ryšius tarp įvedamų ir išvedamų duomenų bei būsenos pasikeitimus[5].

Z daugiausiai naudojama modelių (Model Oriented) specifikacijose. Jos taip vadinamos todėl, kad leidžia sukonstruoti norimos sistemos matematinį modelį. Iš tiesų bet kuri formali specifikacija yra matematinis sistemos modelis. Modeliams orientuotose kalbose pateikiama sistemos būsena (state) kartu su visomis operacijomis, kurios priklauso šiai būsenai. Operacija parodo būsenos reikšmę kartu su jai priklausančių parametrų reikšmėmis bei atvaizduoja, kaip sistema pereina iš vienos būsenos į kitą ir kaip keičiasi jos parametrai.

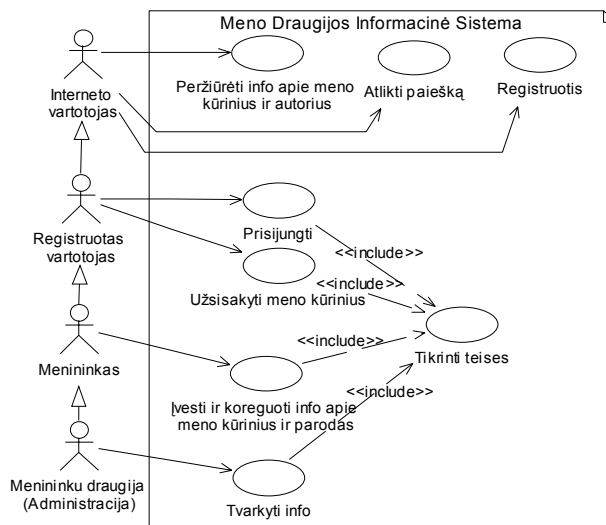
FORMALIŲ SPECIFIKACIJŲ TAIKYMAS PRAKTIKOJE

Norint neformalų aprašą paversti formaliu, reikia laikytis tam tikrų paprastų taisyklių, kurios garantuotų darbo nuoseklumą ir norimą rezultatą. Pirmiausia reikia sukurti neformalų aprašą, išskirti panaudojimo atvejus, sukurti atitinkamas diagramas bei aprašyti išskirtuosius panaudojimo atvejus. Toliau pereinama prie formalizavimo. Formali sistemos aplinka apibrėžiama, formalizuojant panaudojimo atvejus tam tikra tvarka.

Panaudojimo atvejų išskyrimas ir vaizdavimas

Dabar panagrinękime konkretų pavyzdį – vieną iš panaudojimo atvejų, aprašančių konkrečią Meno draugijos informacinę sistemą, sukurtą internete. Šis pavyzdys galėtų tikti daugeliui kitų sistemų, kuriose reikalinga naujų sistemos vartotojų registracija. Kad vartotojas taptų registruotu vartotoju, interneto sistemoje jam reikia pateikti savo asmens duomenis bei unikalų prisijungimo

vardą. Pirmiausia sudaroma pilna panaudojimo atvejų diagrama, kuri šiuo atveju parinkta labai paprasta:



1 pav. Meno draugijos informacinės sistemos panaudojimo atvejų diagrama

Čia apibrėžiame informacinės sistemos sritį – Meno Draugiją, išskiriame aktorių – registruotą vartotoją bei jo atliekamą veiksmą sistemoje – registraciją.

Panaudojimo atvejų aprašymas

Toliau pateikiamas pasirinkto panaudojimo atvejo neformalus aprašymas natūralia kalba, nusakant, kokių reakcijų tikimasi iš sistemos, kai naujas vartotojas vykdo registraciją. Įvardinamas panaudojimo atvejis, aktorius, sistema, aprašoma, kokia yra norinčio užsiregistruoti vartotojo būsena prieš jam atliekant sistemoje bet kuri veiksmą ir kaip atrodo jo veiksmų dar nepaveikta sistema („Prieš sąlyga“). Vėliau aprašoma, kokie veiksmai bus vykdomi sistemoje tam, kad pasiekti norimą rezultatą – užsiregistruoti – ir kaip atrodo sėkmingai įvykdyta operacija („Po sąlyga“). Taip pat reikia įvardinti sąlygas, kurių būtina laikytis, kad registravimasis būtų sėkmingas bei kas nutiktų, jei šios sąlygos nebūtų įvykdytos.

1 lentelė Panaudojimo atvejis „Registruotis“

Panaudojimo atvejis	Registruotis
Numeris	PA3
Aktorius	Registruotas vartotojas
Sistema	Meno draugijos informacinė sistema
Prieš sąlyga	Vartotojas dar nėra tapęs Registruotu vartotoju, todėl jo asmens duomenų Menininkų draugijos informacinės sistemos duomenų bazėje nėra, tačiau jis gali užsiregistruoti ir tapti Registruotu vartotoju.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Įvesti pasirinktą vartotojo vardą.	
2. Įvesti ir pakartoti pasirinktą slaptažodį.	

3. Įvesti savo pavardę.	
4. Įvesti savo namų adresą.	
5. Įvesti savo telefono numerį.	1-5. Jei vartotojo, turinčio tokį prisijungimo vardą ir slaptažodį nėra, toks vartotojas yra įtraukiamas į Menininkų draugijos informacinės sistemos duomenų bazę kaip naujas Registruotas vartotojas.
6. Išsaugoti įvestus duomenis.	6.1. Sistema praneša apie duomenų išsaugojimą.
Po sąlyga	Duomenų bazėje išsaugoti naujo Registruoto vartotojo duomenys.
Alternatyvos (nesėkmės atvejai)	1-5a. Jei sistema randa tokį prisijungimo vardą jau esamoje duomenų bazėje, vartotojas neįtraukiamas kaip naujas Registruotas vartotojas ir sistema praneša apie nutrauktą darbą.
Veiklos taisyklės	Naujas Registruotas vartotojas privalo įvesti unikalų prisijungimo vardą.

Transformacija į Z

Dabar panagrinękime kaip tokį neformalų aprašą galėtume paversti formaliu. Pirmiausia reikia apibrėžti būseną. Akivaizdu, kad atlikus schemoje esančias operacijas, sistemos būseną pasikeis – ji pasipildys naujais duomenimis (nauju vartotoju) ir tai mes galėsime vertinti, kaip sėkmingai įvykdytą sistemos darbą. Tuo atveju, jei įvyktų klaida ar dėl kitokių priežasčių sistemos darbas nutrūktų, mes tai traktuosime, kaip nesėkmingą sistemos darbą ir norėsime, kad sistemoje niekas nepasikeistų.

```

NaujasRegistruotas
state
state'
new_reg?: login
new_slapt?: slapt
new_adr?: adresas
new_tel?: telefonas
new_pav?: pavarde
zinute!: string

pre ((new_reg? dom registruoti) (
new_reg? dom menininkai))
registruoti' registruoti (new_reg?, new_slapt?)
tel' tel (new_reg?, new_tel?)
adr' adr (new_reg?, new_adr?)
pav' pav (new_reg?, new_pav?)
zinute! "registruotas vartotojas ivestas sėkmingai"

```

2 pav. Schema "NaujasRegistruotas"

Taigi „state“ yra pradinė būseną schemoje NaujasRegistruotas, o „state“ yra pasikeitusi būseną, kurios atitinka panaudojimo atvejų skirsnius „Prieš sąlyga“ ir „Po sąlyga“. Kaip jau buvo minėta, kad operacija būtų įvykdyta sėkmingai ir sistema pereitų į pasikeitusią būseną „Po sąlyga“, vartotojas turi įvesti unikalų prisijungimo vardą bei asmens duomenis. „?“ ženklų Z specifikacijoje pažymime įvedamus duomenis, kurie priklauso šalia išvardintoms aibėms: „login“ – unikalų prisijungimo vardų aibė, „slapt“ – slaptažodžių aibė, „adresas“ – adresų aibė ir t.t. Įvedus visus reikiamus duomenis, sistema praneša apie sėkmingą operacijos įvykdymą žinute. Kad operacija būtų įvykdyta sėkmingai, yra tam tikros sąlygos, kurios Z specifikacijoje pateikiamos kaip predikatai. Kad besiregistruojantis nepasirinktų jau sistemoje esančio prisijungimo vardo, tokio vardo negali būti nei tarp prisiregistravusių registruotų vartotojų, nei tarp prisiregistravusių menininkų ((new_reg? dom registruoti) (new_reg? dom menininkai)). Jei ši sąlyga tenkinama, tuomet aib „registruoti“ (turima omeny registruot vartotoj aib) pasipildo nauju registruotu nariu: nauju prisijungimo vardu, nauju, slaptažodžiu bei visa kita asmens informacija, ir sistema praneša apie sėkmingai užbaigtą darbą. Panaudojimo atvejo „Registruotis“ aprašyta nesėkmingai, transformuota Z, atrodyt taip:

NaujasRegistruotasKlaida


```

state
state'
new_reg?: login
zinute!: string

pre ((new_reg? login)
zinute! "registruotas neivestas")

```

3 pav. Schema "NaujasRegistruotasKlaida"

Bandant įvesti prisijungimo vardą, sistema aptinka, kad toks prisijungimo vardas jau yra aibėje „login“, tuomet apie klaidą ji praneša žinute ir nutraukia savo darbą. Sistemoje niekas nepasikeičia. Taigi operacija galinti įvesti naują registruotą vartotoją gali būti arba sėkminga, arba ne:

```

RegistruotasIvestas (NaujasRegistruotas NaujasRegistruotasKlaida)
op IvestiRegistruota
RegistruotasIvestas

```

4 pav. Operacija "RegistruotasIvestas"

IŠVADOS

Panaudojimo atvejai yra taikomi jau daugiau nei 15 metų, tačiau neprarado savo aktualumo. Straipsnyje pateiktas paprastas pavyzdys, tačiau naudojantis šiomis nesudėtingomis taisyklėmis, galima formaliai specifikuoti įvairaus sudėtingumo bei tikslumo panaudojimo atvejus. Pradėti nuo neformalaus aprašo yra labiau priimtina: reikalavimus įvardinti natūralia kalba yra paprasčiau, kaip ir mąstyti objektais. Specifikavimas Z, kaip formalus matematinis metodas, įgalina išlaikyti panaudojimo atvejų tikslumą ir konkretumą, padeda išvengti klaidingų interpretacijų bei nesusipratimų.

Z specifikacija buvo panaudota, specifikuojant ir vėliau realizuojant Meno draugijos informacinę sistemą, kurios vienas panaudojimo atvejų šiame straipsnyje pateiktas kaip pavyzdys. Nors specifikavimas pareikalavo daug kruopštaus darbo, šios sąnaudos pasiteisino vėlesniuose kūrimo etapuose: realizuojant sistemą pagal tikslias specifikacijas, klaidų skaičius buvo labai nedidelis ir nereikėjo pakartotinių iteracijų. Suprantama, didelei sistemai panaudoti Z specifikacijas būtų sunku, tačiau kritiniams sistemos elementams tai tikrai yra tikslinga.

LITERATŪROS ŠARAŠAS

- [1] **R. Clark, A. Moreira.** Constructing Formal Specifications from Informal Requirements. *Proc. Software Tecnology and Engineering Practice, IEEE Computer Society*, 1997.
- [2] **I. Jacobson.** Object-Oriented Software Engineering – a Use Case Driven Approach. *Addison-Wesley, Reading Massachusetts*. 1992.
- [3] **I. Jacobson.** Use Cases – Yesterday, Today and Tomorrow. *Rational Software*, 2002.
- [4] **A. Moreira, R. Clark.** Adding Rigour to Object-Oriented Analysis. *Software Engineering Journal*, 1996, 11(5), 270-280.
- [5] **J.M.Spivey.** The Z Notation: Reference Manual. *Oriel College, Oxford, OX1 4EW, England*, 1998.
- [6] Unified Modeling Language Superstructure Specification. *Version 2.0, OMG document ptc/03-08-02. Available at <http://www.omg.org>*, 2003.

Use cases are widely used for the representation of system and user requirements, but they may vary due to different interpretations. Ambiguity can be avoided by using formal mathematical methods alongside. If you have informal requirements and follow certain rules, it is easy to formalize them. The specification of use cases with Z is demonstrated.

9.2 PRIEDAS. DB schemos specifikacijos MS Visio ataskaita

Table Report

Drawing1

Database summary

Target DBMS: Microsoft SQL Server
Number of tables: 7
Number of views: 2
Number of columns: 39
Number of indexes: 0
Number of foreign keys: 7
Last build date: Not built

Extended attributes:

Filegroup PRIMARY

Tables	Col umns	Ind exes	Foreign keys	Notes
Uzsakymai	5	0	2	
Vart_duom	8	0	1	
Teises	2	0	0	
Paveikslai	6	0	1	
Parodos	4	0	0	
Menininkai	5	0	1	
DalyvavimaiParodose	2	0	2	

Views	Columns
kas	2
Kuriniai	5

DalyvavimaiParodose

Conceptual name: DalyvavimaiParodose
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 2
Number of indexes: 0
Number of foreign keys: 2
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
parodosID (FK)	smallint	Not allowed	
PAV_kodas (FK)	char(10)	Not allowed	

Foreign keys	Child	Parent
Parodos_DalyvavimaiParodose_FK1	parodosID	Parodos.parodosID
Paveikslai_DalyvavimaiParodose_FK1	PAV_kodas	Paveikslai.PAV_kodas

Column details

1. parodosID (FK)

Conceptual name: parodosID
Physical data type: smallint
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. PAV_kodas (FK)

Conceptual name: PAV_kodas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

Foreign key details (child)**Parodos_DalyvavimaiParodose_FK1**

Definition:	Child parodosID	Parent Parodos.parodosID
Relationship type:	Identifying	
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Verb phrase:	hasParodos_DalyvavimaiParodose_FK1	
Inverse phrase:	is of	
Ref. Integrity on update:	No action	
Ref. Integrity on delete:	No action	

Paveikslai_DalyvavimaiParodose_FK1

Definition:	Child PAV_kodas	Parent Paveikslai.PAV_kodas
Relationship type:	Identifying	
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Verb phrase:	hasPaveikslai_DalyvavimaiParodose_FK1	
Inverse phrase:	is of	
Ref. Integrity on update:	No action	
Ref. Integrity on delete:	No action	

kas

Conceptual name: kas
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 2
Number of indexes: 0
Number of foreign keys: 0
Codes: 0
Type: View

Columns	Data type	Allow NULLs	Value/Range
teisiu_vardas	char(10)	Not allowed	
pavarde	char(20)	Not allowed	

Column details

1. teisiu_vardas

Conceptual name: teisiu_vardas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

2. pavarde

Conceptual name: pavarde
Physical data type: char(20)
Portable data type: C-Fixed Length(20)
Allow NULLs: Not allowed

View definition

```
CREATE VIEW dbo.kas
```

```
AS
```

```
SELECT  dbo.Teises.teisiu_vardas, dbo.Vart_duom.pavarde
```

```
FROM    dbo.Teises INNER JOIN
```

```
        dbo.Vart_duom ON dbo.Teises.teisiuID = dbo.Vart_duom.teisiuID
```


Kuriniai

Conceptual name: Kuriniai
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 5
Number of indexes: 0
Number of foreign keys: 0
Codes: 0
Type: View

Columns	Data	Allow	Value/Range
	type	NULLs	
parodosID	smallint	Not allowed	
PAV_kodas	char(10)	Not allowed	
pavadinimas	char(10)	Not allowed	
PAV_statusas	char(10)	Not allowed	
PAV_kaina	smallmoney	Allowed	

Column details

1. parodosID

Conceptual name: parodosID
Physical data type: smallint
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. PAV_kodas

Conceptual name: PAV_kodas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

3. pavadinimas

Conceptual name: pavadinimas

Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

4. PAV statusas

Conceptual name: PAV_statusas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

5. PAV kaina

Conceptual name: PAV_kaina
Physical data type: smallmoney
Portable data type: N-Money(10,4)
Allow NULLs: Allowed

View definition

```
CREATE VIEW dbo.Kuriniai
AS
SELECT  dbo.DalyvavimaiParodose.parodosID, dbo.DalyvavimaiParodose.PAV_kodas,
        dbo.Paveikslai.pavadinimas, dbo.Paveikslai.PAV_statusas,
        dbo.Paveikslai.PAV_kaina
FROM    dbo.Paveikslai INNER JOIN
        dbo.DalyvavimaiParodose ON dbo.Paveikslai.PAV_kodas = dbo.DalyvavimaiParodose.PAV_kodas
INNER JOIN
        dbo.Parodos ON dbo.DalyvavimaiParodose.parodosID = dbo.Parodos.parodosID
```

Menininkai

Conceptual name: Menininkai
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 5
Number of indexes: 0
Number of foreign keys: 1
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
men_ID	smallint identity	Not allowed	
biografija	char(10)	Allowed	
kontaktai	char(10)	Not allowed	
vart_ID (FK)	smallint	Not allowed	
login (FK)	char(10)	Allowed	

Foreign keys	Child	Parent
Vart_duom_Menininkai_FK1	vart_ID login	Vart_duom.vart_ID Vart_duom.login
Menininkai_Paveikslai_FK1	Paveikslai.men_ID	men_ID

Column details

1. men_ID

Conceptual name: men_ID
Physical data type: smallint identity
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. biografija

Conceptual name: biografija
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Allowed

3. kontaktai

Conceptual name: kontaktai
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

4. vart_ID (FK)

Conceptual name: vart_ID
Physical data type: smallint
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

5. login (FK)

Conceptual name: login
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Allowed

Foreign key details (child)

Vart duom Menininkai FK1

Definition:	Child	Parent
	vart_ID	Vart_duom.vart_ID
	login	Vart_duom.login
Relationship type:	Non-Identifying	
Cardinality:	Zero-or-One -to- Zero-or-More	
Allow NULLs:	Allowed	
Verb phrase:	hasVart_duom_Menininkai_FK1	
Inverse phrase:	is of	
Ref. Integrity on update:	No action	
Ref. Integrity on delete:	No action	

Parodos

Conceptual name: Parodos
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 4
Number of indexes: 0
Number of foreign keys: 0
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
parodosID	smallint identity	Not allowed	
data	datetime	Allowed	
pavadinimas	char(10)	Allowed	
aprasas	char(10)	Allowed	

Foreign keys	Child	Parent
Parodos_DalyvavimaiParodose_FK1	DalyvavimaiParodose.parodosID	parodosID

Column details

1. parodosID

Conceptual name: parodosID
Physical data type: smallint identity
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. data

Conceptual name: data
Physical data type: datetime
Portable data type: T-Date & Time
Allow NULLs: Allowed

3. pavadinimas

Conceptual name: pavadinimas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Allowed

4. aprasas

Conceptual name: aprasas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Allowed

Paveikslai

Conceptual name: Paveikslai
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 6
Number of indexes: 0
Number of foreign keys: 1
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
PAV_kodas	char(10)	Not allowed	
PAV_statusas	char(10)	Not allowed	
PAV_kaina	smallmoney	Allowed	
men_ID (FK)	smallint	Not allowed	
pavadinimas	char(10)	Not allowed	
pavyzdys	image	Allowed	

Foreign keys	Child	Parent
Menininkai_Paveikslai_FK1	men_ID	Menininkai.men_ID
Paveikslai_DalyvavimaiParodose_FK1	DalyvavimaiParodose.PAV_kodas	PAV_kodas
Paveikslai_Uzsakymai_FK1	Uzsakymai.PAV_kodas	PAV_kodas

Column details

1. PAV_kodas

Conceptual name: PAV_kodas
Physical data type: char(10)

Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

2. PAV statusas

Conceptual name: PAV_statusas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

3. PAV kaina

Conceptual name: PAV_kaina
Physical data type: smallmoney
Portable data type: N-Money(10,4)
Allow NULLs: Allowed

4. men ID (FK)

Conceptual name: men_ID
Physical data type: smallint
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

5. pavadinimas

Conceptual name: pavadinimas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

6. pavyzdys

Conceptual name: pavyzdys
Physical data type: image
Portable data type: R-Picture
Allow NULLs: Allowed

Foreign key details (child)

Menininkai Paveikslai FK1

Definition:	Child	Parent
	men_ID	Menininkai.men_ID

Relationship type: Non-Identifying
Cardinality: One -to- Zero-or-More

Allow NULLs:	Not allowed
Verb phrase:	hasMenininkai_Paveikslai_FK1
Inverse phrase:	is of
Ref. Integrity on update:	No action
Ref. Integrity on delete:	No action

Teises

Conceptual name: Teises
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 2
Number of indexes: 0
Number of foreign keys: 0
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
teisiuID	int	Not allowed	
teisiu_vardas	char(10)	Not allowed	

Foreign keys	Child	Parent
Teises_Vart_duom_FK1	Vart_duom.teisiuID	teisiuID

Column details

1. teisiuID

Conceptual name: teisiuID
Physical data type: int
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. teisiu_vardas

Conceptual name: teisiu_vardas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

Uzsakymai

Conceptual name: Uzsakymai
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 5
Number of indexes: 0
Number of foreign keys: 2
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
uzsakymo_nr	smallint identity	Not allowed	
PAV_kodas (FK)	char(10)	Not allowed	
pard	bit	Not allowed	
vart_ID (FK)	smallint	Allowed	
login (FK)	char(10)	Allowed	

Foreign keys	Child	Parent
Paveikslai_Uzsakymai_FK1	PAV_kodas	Paveikslai.PAV_kodas
Vart_duom_Uzsakymai_FK1	vart_ID login	Vart_duom.vart_ID Vart_duom.login

Column details

1. uzsakymo_nr

Conceptual name: uzsakymo_nr
Physical data type: smallint identity
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. PAV_kodas (FK)

Conceptual name: PAV_kodas
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

3. pard

Conceptual name: pard
Physical data type: bit
Portable data type: L-True or False
Allow NULLs: Not allowed

4. vart_ID (FK)

Conceptual name: vart_ID
Physical data type: smallint
Portable data type: N-Signed Integer
Allow NULLs: Allowed

5. login (FK)

Conceptual name: login
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Allowed

Foreign key details (child)

Paveikslai Uzsakymai FK1

Definition:	Child	Parent
	PAV_kodas	Paveikslai.PAV_kodas
Relationship type:	Non-Identifying	
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Verb phrase:	hasPaveikslai_Uzsakymai_FK1	
Inverse phrase:	is of	
Ref. Integrity on update:	No action	
Ref. Integrity on delete:	No action	

Vart duom Uzsakymai FK1

Definition:	Child	Parent
	vart_ID	Vart_duom.vart_ID

login

Vart_duom.login

Relationship type:

Non-Identifying

Cardinality:

Zero-or-One -to- Zero-or-More

Allow NULLs:

Allowed

Verb phrase:

hasVart_duom_Uzsakymai_FK1

Inverse phrase:

is of

Ref. Integrity on update:

No action

Ref. Integrity on delete:

No action

Vart_duom

Conceptual name: Vart_duom
Owner: dbo
Target DB name: st04dbtbagbr
Number of columns: 8
Number of indexes: 0
Number of foreign keys: 1
Codes: 0
Type: Table

Extended attributes:

OnFileGroup PRIMARY
Clustered PK Yes

Columns	Data type	Allow NULLs	Value/Range
vart_ID	smallint identity	Not allowed	
login	char(10)	Not allowed	
slapt	char(10)	Not allowed	
adresas	char(20)	Not allowed	
pavarde	char(20)	Not allowed	
telefonas	char(20)	Not allowed	
teisiuID (FK)	int	Not allowed	
gim_data	smalldateti me	Not allowed	

Foreign keys	Child	Parent
Teises_Vart_duom_FK1	teisiuID	Teises.teisiuID
Vart_duom_Menininkai_FK1	Menininkai.vart_ID Menininkai.login	vart_ID login
Vart_duom_Uzsakymai_FK1	Uzsakymai.vart_ID	vart_ID

Column details**1. vart_ID**

Conceptual name: vart_ID
Physical data type: smallint identity
Portable data type: N-Signed Integer
Allow NULLs: Not allowed

2. login

Conceptual name: login
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

3. slapt

Conceptual name: slapt
Physical data type: char(10)
Portable data type: C-Fixed Length(10)
Allow NULLs: Not allowed

4. adresas

Conceptual name: adresas
Physical data type: char(20)
Portable data type: C-Fixed Length(20)
Allow NULLs: Not allowed

5. pavarde

Conceptual name: pavarde
Physical data type: char(20)
Portable data type: C-Fixed Length(20)
Allow NULLs: Not allowed

6. telefonas

Conceptual name: telefonas
Physical data type: char(20)
Portable data type: C-Fixed Length(20)
Allow NULLs: Not allowed

7. teisiuID (FK)

Conceptual name: teisiuID
Physical data type: int

Portable data type: N-Signed Integer
Allow NULLs: Not allowed

8. gim_data

Conceptual name: gim_data
Physical data type: smalldatetime
Portable data type: T-Date & Time
Allow NULLs: Not allowed

Foreign key details (child)

Teises Vart duom FK1

Definition:	Child	Parent
	teisiuID	Teises.teisiuID
Relationship type:	Non-Identifying	
Cardinality:	One -to- Zero-or-More	
Allow NULLs:	Not allowed	
Verb phrase:	hasTeises_Vart_duom_FK1	
Inverse phrase:	is of	
Ref. Integrity on update:	No action	
Ref. Integrity on delete:	No action	