

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Simonas Šerlinskas

# **WSDL servisų sujungimas ir kodo generavimas**

Magistro darbas

Darbo vadovas

prof. dr. Rimantas Butleris

Konsultantas

dokt. Edvinas Šinkevičius

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Simonas Šerlinskas

**WSDL servisų sujungimas ir kodo generavimas**

Magistro darbas

Recenzentas

doc. dr. D. Makackas

2012-05-28

Vadovas

prof. dr. Rimantas Butleris

Konsultantas

dokt. Edvinas Šinkevičius

2012-05-28

Atliko

IFME-0/4 gr. stud.

Simonas Šerlinskas

2012-05-28

Kaunas, 2012

# Turinys

<b>TURINYS</b> .....	<b>3</b>
<b>LENTELIŲ SĄRAŠAS</b> .....	<b>4</b>
<b>PAVEIKSLĖLIŲ SĄRAŠAS</b> .....	<b>5</b>
<b>WSDL SERVICES MAPPING AND CODE GENERATION</b> .....	<b>7</b>
<b>SUMMARY</b> .....	<b>7</b>
<b>1. ĮVADAS</b> .....	<b>8</b>
<b>2. WSDL SERVISŲ SUJUNGIMAS IR KODO GENERAVIMAS ANALIZĖ</b> .....	<b>11</b>
2.1. Tyrimo objektas, sritis ir problema.....	11
2.2. Tyrimo objekto analizė .....	12
2.3. Vartotojų analizė.....	17
2.3.1. Programuotojai .....	17
2.3.2. Įmonių darbuotojai (neskaitant programuotojų).....	18
2.3.3. Mokymo įstaigos .....	18
2.3.4. Vartotojų tikslai ir savybės .....	18
2.4. Esamų sprendimų analizė .....	18
2.5. Siekiamas sprendimas.....	20
2.6. Darbo tikslas ir uždaviniai .....	20
2.7. Tyrimo planas .....	21
2.8. Rizikos faktorių analizė .....	21
2.9. Rezultato kokybės kriterijai .....	22
2.10. Analizės išvados.....	22
<b>3. WSDL SERVISŲ SUJUNGIMAS IR KODO GENERAVIMO SISTEMOS REIKALAVIMAI</b> .....	<b>23</b>
3.1. Reikalavimų specifikacija.....	23
3.1.1. Funkciniai reikalavimai: .....	23
3.2. Dalykinės srities modelis .....	24
<b>4. WSDL SERVISŲ SUJUNGIMAS IR KODO GENERAVIMO SISTEMOS PROJEKTAS</b> .....	<b>24</b>
4.1. Sistemos architektūros projektas .....	24
4.1.1. Sistemos loginė architektūra.....	24
4.1.2. Vartotojo paslaugos (sąsaja, sąsajos navigacijos planas) .....	33
4.1.3. Veiklos paslaugos .....	37
4.1.4. Duomenų paslaugos.....	37
4.2. Sistemos elgsenos modelis.....	38
4.3. Duomenų bazė .....	39
<b>5. WSDL SERVISŲ SUJUNGIMAS IR KODO GENERAVIMO SISTEMOS REALIZACIJA</b> .....	<b>42</b>
5.1. Sistemos integravimas .....	42

5.2.	Sistemos veikimo aprašymas .....	42
5.3.	Testavimo modelis ir duomenys, kontrolinis pavyzdys.....	46
<b>6.</b>	<b>IŠVADOS.....</b>	<b>56</b>
<b>7.</b>	<b>LITERATŪROS SĄRAŠAS .....</b>	<b>57</b>
<b>8.</b>	<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS.....</b>	<b>59</b>
<b>9.</b>	<b>PRIEDAI .....</b>	<b>60</b>
9.1.	Funkciniai reikalavimai .....	60
9.2.	Nefunkciniai reikalavimai.....	70
9.3.	Diegimo aktas .....	74
9.4.	Darbuotojų apklausa .....	75
9.5.	Darbuotojų apklausos rezultatai.....	75

## **Lentelių sąrašas**

9.1 lentelė.	Vartotojo prisijungimas PA .....	60
9.2 lentelė.	Vartotojo atsijungimas PA.....	60
9.3 lentelė.	Registruoti grupes PA .....	61
9.4 lentelė.	Registruoti vartotojus PA.....	61
9.5 lentelė.	Nustatyti teises PA.....	62
9.6 lentelė.	Administruoti projektus PA .....	62
9.7 lentelė.	Sukurti projektą PA .....	63
9.8 lentelė.	Redaguoti projektą PA.....	64
9.9 lentelė.	Trinti projektą PA .....	64
9.10 lentelė.	Peržiūrėti projekto informacija PA .....	65
9.11 lentelė.	Peržiūrėti WSDL servisų informaciją PA.....	65
9.12 lentelė.	Įtraukti servisą PA .....	66
9.13 lentelė.	Ištrinti servisą PA.....	66
9.14 lentelė.	Peržiūrėti serviso informacija ir kintamuosius PA .....	67
9.15 lentelė.	Priskirti funkcijos kintamąjį į duomenų bazės lentelę PA .....	67
9.16 lentelė.	Valdyti duomenų bazę PA .....	68
9.17 lentelė.	Sinchronizuoti duomenų bazę PA.....	69
9.18 lentelė.	Eksportuoti į programinį kodą PA .....	69
9.19 lentelė.	Paprastas naudojimas .....	70
9.20 lentelė.	Tvarkinga vartotojo sąsaja .....	70
9.21 lentelė.	Patraukli sistema .....	71

9.22 lentelė. Sistemos efektyvumas.....	71
9.23 lentelė. Lengvas sistemos tobulinimas .....	71
9.24 lentelė. Sistema turi būti stabili .....	72
9.25 lentelė. Sistema turi būti lengvai testuojama .....	72
9.26 lentelė. Sistema turi būti perkeliama į kitą serverį .....	72

## **Paveikslėlių sąrašas**

2.1 pav. Serviso pritaikymas daugiaplatformai .....	13
2.2 pav. WSDL aprašo sandara .....	14
3.1 pav. Panaudojimo atvejai.....	23
3.2 pav. Dalykinės srities diagrama.....	24
4.1 pav. Projekto sukūrimas .....	25
4.2 pav. Projekto ištrynimasis .....	26
4.3 pav. Pridėti servisą prie projekto .....	27
4.4 pav. Analizuojamas rezultatas .....	28
4.5 pav. Sinchronizuoti duomenų bazę.....	29
4.6 pav. Prijungti serviso funkciją į duomenų bazę.....	30
4.7 pav. Eksportuojamas programinis kodas .....	31
4.8 pav. Įtraukti servisų klasių generavimas.....	32
4.9 pav. Pridedamos sukurtos užklauskos .....	33
4.10 pav. Visi projektai .....	34
4.11 pav. Projekto redagavimas.....	35
4.12 pav. Pridėti servisą.....	36
4.13 pav. Prijungti servisą .....	36
4.14 pav. Veiklos paslaugos .....	37
4.15 pav. Sistemos elgsenos modelis .....	38
4.16 pav. Duomenų bazės ER diagrama.....	39
4.17 pav. Duomenų bazės schema.....	40
5.1 pav. Sistemos integravimo schema.....	42
5.2 pav. Vartotojų prisijungimo forma .....	43
5.3 pav. Projektų sąrašas .....	43

5.4 pav. Projekto redagavimas.....	44
5.5 pav. Funkcijos prijungimo tempimo vieta.....	44
5.6 pav. Funkcijos prijungimo parametrų forma .....	45
5.7 pav. Sugeneruoto programinio kodo archyvo turinys .....	46
4.8 pav. Serviso rezultato duomenų vaizdas.....	46
5.9 pav. Testuojamo serviso funkcijų sąrašas.....	47
5.10 pav. Serviso aprašo iškarpa .....	47
5.11 pav. Testuojamo serviso funkcijų sąrašas.....	47
5.12 pav. Serviso aprašo iškarpa .....	47
5.13 pav. Sukurtos lentelės servisų testavimui.....	48
5.14 pav. Prijungtos funkcijos prie lentelių.....	49
5.15 pav. Gauti testiniai duomenys .....	55

# **WSDL services mapping and code generation**

## **Summary**

These days are a variety of online services that growing constantly by changing their methods of communication and quality. This work discusses one of the most widely developed WSDL services technology and internet data protocol SOAP. WSDL description, which is an XML-based service is a structured language. There is no set standard SOAP protocol to call at web browser, in order to form a query and get the results. Ordinary users are difficult or impossible to use this service independently.

To solve the problem was chosen develop the new system, because there is no such a decision that has been made or is currently under construction. The system will have a graphical user interface that will be generated by the WSDL description. The main requirement is that a developed system can generate the programing code by combining several existing e-service SOAP protocols.

The study area of work - Information systems development methodology, information systems design: methods, tools, architecture, electronic services.

The object of research - several similar bundling of electronic services into one data flow. Combine multiple WSDL services, and one generate programming code.

Developing the system intended goal was to give the user the ability to combine multiple WSDL services into one. It is more common to output data flows, which was modeling by graphic user interface in system.

The entire system developed without third parties programming code. The system is fully object-oriented, so it can be easily expanded. As each key is braked down into modules that can be overwritten and extended functionality. The system can be installed on most web hosting provider's servers. For this purpose was designed installation module, which adds a one click installation future.

# 1. Įvadas

Internetu teikiamų paslaugų įvairovė didėja nuolat, kinta jų perdavimo būdai, kokybė. Šiame darbe aptariama viena plačiausiai išplėtota *WSDL* servisų technologija ir galimi klientai informacijai gauti bei siųsti. Naudojantis elektroninėmis paslaugomis lengviau perduoti duomenis tarp taikomųjų programų. Tai populiari buhalterinėse, bankinėse, elektroninės komercijos, orų programose. Kaip bebūtų, šia paslauga pasinaudoti praktiškai yra sudėtinga. Kol kas nėra nustatyto standarto, *SOAP* protokolą iškviešti per interneto naršyklę, o norint suformuoti užklausą ir gauti rezultatus, reikalinga taikomoji programa. Paprastiems vartotojams yra sudėtinga arba neįmanoma naudotis tokiomis paslaugomis savarankiškai. *WSDL* aprašas, pagal kurį veikia servisas, yra paremtas *XML* struktūrizuota kalba, ir tokioje aplinkoje nesinaudojant trečiųjų šalių įrankiais, sudėtinga suprasti kaip naudotis paslauga. Taip pat ir vartotojams, kurie specializuojasi programavimo srityje, reikia nemažai laiko resursų, o kai kada iškyla problemų gauti programinį kodą, kuris priimtų duomenis naudojant *REST* arba *SOAP* protokolą teisingai. Šių problemų sprendimui yra sukurta įvairių įrankių, tokių kaip *XMLspy*, *StylusStudio* ir kitų (jų apžvalga pateikta analogų analizėje). Naudojant bet kurį iš analizuotų įrankių, generuoti galima tik iš vieno *WSDL* ir su ribotomis galimybėmis.

Problemos sprendimui pasirinkta realizuoti naują sistemą, kadangi tokio sprendimo kol kas nėra, arba jis šiuo metu kuriamas. Sistema turės grafinę vartotojo sąsają, kuri bus generuojama pagal gaunamą *WSDL* aprašą [21]. Pagrindinis reikalavimas, kad naujai kuriama sistema galėtų sugeneruoti programinį kodą, sujungiant keletą elektroninių paslaugų servisų, veikiančių *SOAP* protokolu. Taip pat vartotojui būtų pateikta grafinė vartotojo sąsaja, su atitinkamais duomenų sujungimo pasirinkimais ir integracijos pritaikymu į duomenų bazę, gaunamiems duomenims išsaugoti.

Šio tyrimo sritis – informacinių sistemų kūrimo metodologijos, informacinių sistemų projektavimas: metodai, įrankiai, architektūros, elektroninės paslaugos.

Atliekamo tyrimo objektas - kelių panašių elektroninių paslaugų sujungimas į vieną paslaugą: sujungti kelis *WSDL* servisas ir išvesti vieną klasę.

Realizuojant sistemą, siekiama tikslo – suteikti vartotojui galimybę sujungti keletą *WSDL* servisų į vieną. Tai yra iš kelių duomenų srautų išvesti bendrą, nustatytos struktūros duomenų srautą. Tam keliami uždaviniai:

1. Iš pateiktos *WSDL* schemos sugeneruoti klases ir metodus;
2. Klasėms sugeneruoti grafinę aplinką su galimybe ją lengvai modifikuoti;



3. Reikalinga duomenų bazė pakeitimams saugoti: konkrečioms laukų reikšmėms, keisti laukų pavadinimus, pervadinti metodus;
4. Galimybė prie projekto prijungti dar viena *WSDL* ir grafiškai modifikuoti, prisitaikant konkrečiai dalykinei sričiai;
5. Galimybė sukurtą projektą, kuriame bus sugeneruoti programiniai kodai iš *WSDL*, pritaikyti kitoms informacinėms sistemoms;
6. Programuoti naudojant plačiai naudojamą [10] atvirojo kodo *PHP* programavimo kalbą ir *MySQL* duomenų bazę.

Atliekant *WSDL* servisų sujungimo ir duomenų gavimo analizę, buvo išanalizuoti literatūros šaltiniai [1], [4], [20], [21], [22]. Juose nuodugniai apžvelgiama *WSDL* technologija, jos pritaikymo ir panaudojimo galimybės. Viską įvertinus, sistemai realizuoti buvo pasirinkta atvirojo kodo *PHP* programavimo kalba, o elektroninių paslaugų duomenų perdavimui pasirinktas *SOAP*, kadangi šiuo metu tai vienas populiariausių ir saugiausių perdavimo būdų [11]. Analizuoti *WSDL* servisų aprašai [21] buvo pasitelktos *XSLT* transformacijos [16], [19]. Grafinis servisų vaizdavimas funkcijų ir kintamųjų gyliu, programinio kodo generavimas ir kitos pagrindinės funkcijos buvo sukurtos naujai, nesiremiant jokia kita programine įranga.

Sistema, pagal atliktą analizę ir projektavimą, realizuota sėkmingai. Vartotojui sistema pateikiama intuityvia ir tuo pačiu funkcionalia grafine sąsaja. Naudojantis ja, galima pasinaudoti visomis teikiamomis duomenų paslaugomis. Reikia atkreipti dėmesį, kad sistema sukurta palaikyti tik *SOAP* tipo protokolą dėl jo populiarumo [11], *REST* tipu veikiantys protokolai nepalaikomi. Visas sistemos programinis kodas parašytas nenaudojant kitų, jau sukurtų sistemų. Sistema yra objektiškai orientuota, todėl gali būti lengvai plečiama, nes kiekviena pagrindinė dalis yra išskaidyta į modulius, kurie gali būti perrašomi (*angl. extended*) arba praplečiamas jų funkcionalumas (*angl. inheritance*). Sistema gali būti įdiegta daugelyje interneto tiekėjų siūlomose serverių prieglobose (*angl. hosting*). Šiai paskirčiai sukurta instaliacijos modulis, kuris vienu mygtuko paspaudimu sukonfigūruoja sistemą ir sukuria duomenų bazės lenteles, bei užpildo jas pradiniais reikalingais duomenimis. Plačiau apie tai rašoma Sistemos projekto skyriuje [24].

Realizavus ir ištestavus sistemą, ji buvo pristatyta 2012 m. balandžio 20 d. tarptautinėje elektroninės komercijos internetinės rinkodaros ir IT srityje specializuojančioje kompanijoje UAB „Net Frequency“. Po pristatymo nuspręsta, jog sistema yra perspektyvi, ir buvo įdiegta į įmonės vidiniame tinkle esantį serverį. Tai patvirtina įdiegimo aktas, kuris pateiktas prieduose. Taip pat planuojama investuoti į sistemą ir tęsti tobulinimo darbus. Įmonės programuotojams sistema paspartina darbą su kai kuriais *WSDL* servisais.

**Darbo struktūra.** Darbas pradedamas santrauka, kuri parašyta anglų kalba. Ji skirta pristatyti idėją, jos naujumą, privalumus bei trūkumus, ir atpažinti darbą tiems, kuriems jis gali būti aktualus. Įvadas – jame trumpai apžvelgiamas visas darbas. Problemos ir sprendimai aprašomi analitinėje dalyje, o sistemos prototipas – darbo projektavimo ir realizacijos skyriuose. Darbas apibendrinamas išvadamis, o pabaigoje pateikiamas naudotos literatūros sąrašas ir priedai.

Antrajame skyriuje pateikiami atliktos tyrimo analizės rezultatai. Apžvelgiama probleminė sritis, siūlomi galimi sprendimai. Taip pat pateikiamas WSDL serviso paaiškinimas ir veikimo principas. Pristatoma visų egzistuojančių interneto servisų protokolų analizė. Apžvelgiami skirtingi vartotojų tipai, taip lyginant šiuo metu rinkoje esančius panašiausius sprendimus, nors nė vienas jų negali išspręsti iškeltos problemos.

Trečiajame skyriuje nustatomi kuriamos sistemos funkciniai ir nefunkciniai reikalavimai. Taip pat pateikiama dalykinės sistemos srities diagrama.

Sistemos projektas pateikiamas ketvirtajame šio darbo skyriuje, kuriame apžvelgiama būsima sistemos grafinė vartotojo sąsaja, klasės ir funkcijos. Sistemos architektūra detalizuojama panaudojimo atvejų diagramomis. Taip pat sekų diagrama pavaizduojamas sistemos elgsenos modelis, kuriame matomi funkciniai sąryšiai tarp kontrolių ir paslaugų. Duomenų bazė pateikiama tiek esybių ryšių diagrama, tiek paprasta schema.

Penktajame šio darbo skyriuje kalbama apie sistemos realizavimą. Čia vaizduojamos realiai sukurtos sistemos funkcijos. Pateikiama detali vartotojo iliustruota instrukcija, kaip naudotis sistema. Taip pat pateikiamas sistemos testavimo modelis, su konkrečiais duomenimis, paaiškinimais ir rezultatais.

Šeštajame skyriuje pateikiamos darbo išvados.

## 2. WSDL servisų sujungimas ir kodo generavimas analizė

Išanalizuoti literatūros šaltinius, kuriuose rašoma apie *WSDL* pritaikymą ir plėtojimą. Išanalizuoti praktinį *WSDL* panaudojimą. Apžvelgti galimybes ir panaudojimo atvejus įvairiose programavimo kalbose.

### 2.1. Tyrimo objektas, sritis ir problema

**Tyrimo sritis** — informacinių sistemų kūrimo metodologijos, informacinių sistemų projektavimas: metodai, įrankiai, architektūros, elektroninės paslaugos.

**Tyrimo objektas** — kelių panašių elektroninių paslaugų apjungimas į vieną paslaugą. Sujungti kelis *WSDL* servigus ir išvesti vieną klasę.

**Problema** — su lyg kiekviena diena rinkoje atsiranda vis daugiau elektroninių paslaugų. Pavyzdžiui, orų prognozės, žodynai, elektroninė prekyba, tikslųjų mokslų skaičiavimai, ir kitos paslaugos, kurioms reikalingas didelių duomenų srautų perdavimas. Kiekviena įmonė stengiasi savo veiklos paslaugas pateikti elektroninėje erdvėje. Daugumai tokių paslaugų realizuoti yra naudojamos *WSDL* [4] (*angl. web service definition language*) technologijos. WEB serviso klientą galima pritaikyti bet kurioje platformoje: telefone, *Windows*, *Linux* ar *MAC* operacinėje sistemoje ir daugelyje kitų kuriame veikia *HTTP* protokolas. Naudojant *WSDL* teikiamos įvairios paslaugos perduodant duomenis. *WSDL* servigai pritaikyti daugeliui programavimo kalbų ir jų yra įvairių, iš kurių dažniausiai naudojamos *SOAP* [1] ir *XML*. Problema iškyła, kai visa tai reikia realizuoti praktiškai, kadangi tam reikia aukšto lygio programavimo žinių. Be to, tai užima daug laiko. Realizavimo procesui palengvinti yra sukurta įrankių, tokių kaip *IBM websphere* [2], *ALTOVA*[5], *Stylus Studio*®[8], bet šitie įrankiai turi didelių trūkumų. Visų pirma, jų kaina yra didelė, o ir dirbti galima tik su vienu *WSDL* servisu. Augantys poreikiai ir didėjančios galimybės elektroninėje erdvėje, priverčia ieškoti sprendimų, tokių kaip vartotojo galimybė dirbti su keletą *WSDL* servisų. Reiktų atsižvelgti į tai, jog sistema galėtų naudotis vartotojas, neturintis arba turintis tik pradines programavimo žinias.

**Sprendimas** — problemą išspręstų įrankis, kuris padėtų sujungti keletą *WSDL* servisų. Jis turėtų duomenų srautus nukreipti į norimą struktūrą, pagal kurią sugeneruojamas programinės klasės kodas. Generuojant kodą, turėtume galimybę pasirinkti kaip įvesti duomenis. Kitaip sakant, pateikti atskirą konfigūracijos failą ar tiesiogiai išsaugoti duomenis sugeneruotame kode. Įrankio,

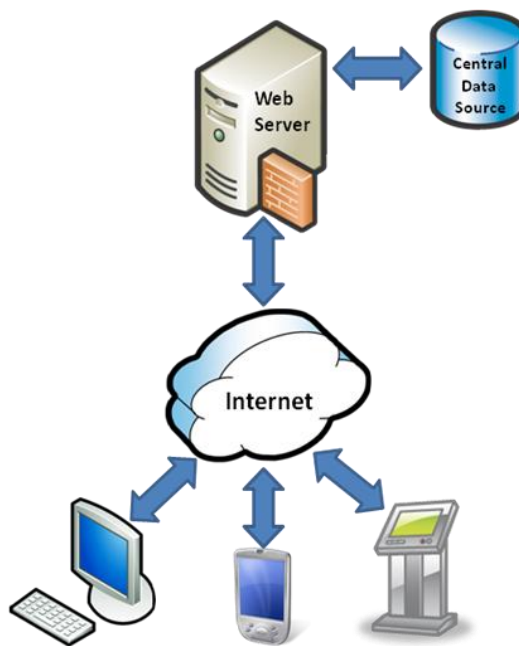
kuris išspręstų nagrinėjamą problemą, kol kas dar nėra suprogramuoto nei viena programavimo kalba, arba jis šiuo metu dar tik kuriamas. Įrankis atpažins *WSDL* servisas, veikiančius *SOAP* protokolo pagrindu. Pasirinktas *SOAP* protokolas yra bene plačiausiai naudojamas duomenų srautams perduoti ir greičiausiai veikiantis iš visų protokolų [9]. *SOAP* protokolas gali būti naudojamas apsaugotu šifruotu protokolu „SSH“.

## 2.2. Tyrimo objekto analizė

**WSDL servisas** — patogus, nes veikia su visomis platformomis, kurios palaiko naudojamą *HTTP* protokolą. *WSDL* yra universalus, tiek kalbant apie serverį, tiek apie klientą, ir gali būti suprogramuotas daugeliu programavimo kalbų (*JAVA*, *C#*, *PHP*, *RUBY*, *PYTHON*). Serverio dalies programinis kodas nebūtinai turi būti parašytas ta pačia kalba kaip ir kliento dalies.

**WSDL naudojimas** — duomenų paslaugos tiekėjas turi savo informacinę sistemą, kurioje įveda arba atnaujina duomenis duomenų bazėje. Serveryje veikia *WSDL* servisas, kuris atsako į klientų užklausas. Jos gali būti pateikiamos įvairiais protokolais, iš kurių šiai dienai populiariausi yra *SOAP*, *XML*, *JSON*. Užklauso tipas priklauso nuo serverio konfigūracijos. Serveris gali palaikyti ir keletą tipų. Tokiu būdu galima perduoti įvairius duomenis (orų prognozes, prekių informaciją, kelių eismo informaciją, matematinius ir fizikinius skaičiavimus ir t.t.). Duomenys gali būti perduoti objektu arba *XML* failu.

Toliau pateikiamas grafinis pavyzdys [13] (žiūrėti Paveikslukas 7.1) kaip veikia *WSDL* serveris. Duomenis saugo duomenų bazės serveris (*angl. Database server*). Duomenų srautus, priimamus ir siunčiamus duomenis, valdo *WEB* serviso serveris (*angl. Application & Web Service Server*). Prie šio serverio galima jungtis bet kuriuo elektroniniu prietaisu, kuris palaiko *HTTP* protokolą. Optimalesniam veikimui rekomenduojama jungtis naudojant *SOAP* arba *REST* tipo klientu, žinoma, jei juos palaiko *WEB* serviso serveris.

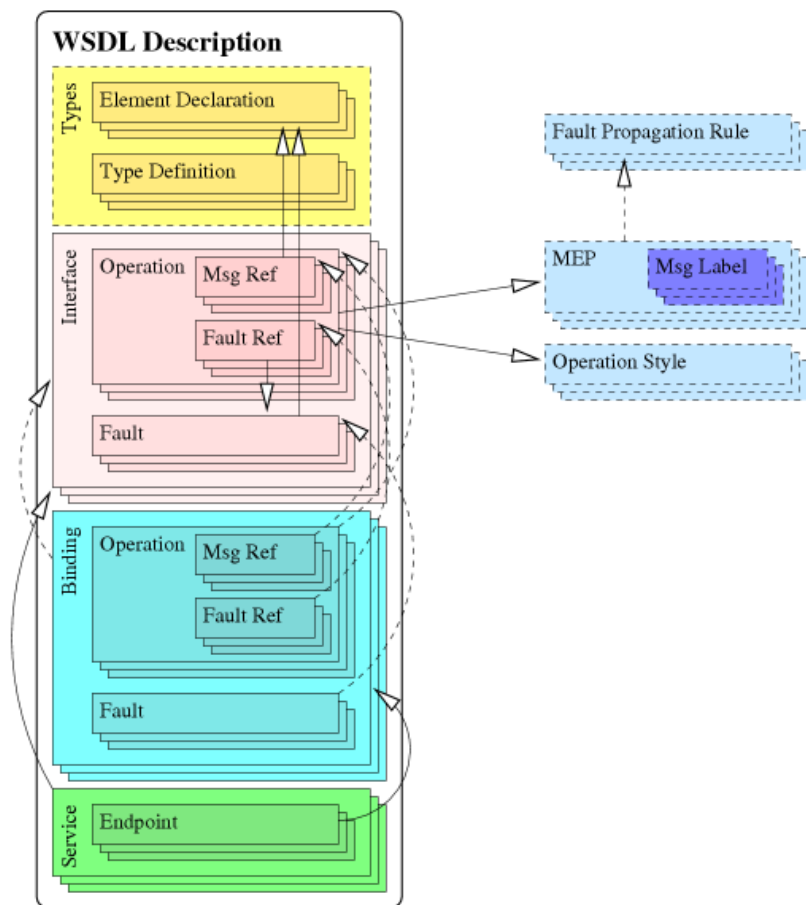


2.1 pav. Serviso pritaikymas daugiaplatformai

Šaltinis: <http://msdn.microsoft.com/en-us/sync/bb887608>

WSDL serveris [12] susideda iš atskirų dalių (žiūrėti Paveikslukas 2.2).

1. Servisas (*angl. Service*) — paslaugos sąsajos perduodami duomenys, kurių funkcionalumas nieko nesiskiria nuo paskutinių užklausų (*angl. Endpoint*). Duomenys apie servisą perduodami sąsajai (*angl. Interface*), kur jie apdorojami ir siunčiami toliau. Paskutinis taškas (*angl. Endpoint*) nusako informaciją apie servisą, kurią sudaro trys grupės:
  1. Pavadinimas (*angl. Name*) — serviso pavadinimas;
  2. Aprašymas (*angl. Binding*) — serviso detalesnis aprašymas;
  3. Adresas (*angl. Address*) — serviso adresas.
2. Aprašymas (*angl. Binding*) — aprašo informacijos tipą, pačią informaciją ir galimas klaidas. Visa informacija perduodama į sąsają.
3. Serviso sąsaja (*angl. Interface*) — saugomos funkcijos, kurios grąžina duomenis klientui. Klientas kreipiasi pagal aprašytą funkciją operacijų bloke (*angl. Operation*). Rezultatai grąžinami XML fragmentu, kaip žinutė. Jei užklausoje yra neteisingų parametru, grąžinama klaida, taip pat žinutės pavidalu.
4. Tipai (*angl. Types*) — tiek žinutės, tiek turinys gali būti aprašomas įvairiais tipais, kurie aprašomi tipų objekte. Taip pat saugomi parametru aprašai (*angl. Element Declaration*).



2.2 pav. WSDL aprašo sandara

Šaltinis: <http://www.wsmo.org/TR/d11/v0.2/20070421/>

**XMLRPC** — nuotolinių užklausų procedūrų protokolas, dar kitaip RPC, kuris koduoja užklausas XML formatu ir siunčia naudojantis HTTP protokolu. Gali perduoti tiek paprastus duomenis, tiek objektus kaip užklausas ar atsaką į užklausą.

Privalumai:

1. Nesukelia problemų skirstant duomenis per ugniasienes;
2. Dėl daugybės aprašytų bibliotekų, labai paprasta naudotis dauguma objektinių programavimo kalbų;
3. Kreipimuisi naudoja vieną metodą ir turi tik vieną kodavimą, kuriuo galime siųsti tiek objektus, tiek paprastus duomenis;
4. Turi aprašytą klaidų sąrašą. Įvykus klaidai grąžinamas pranešimas klientui;
5. Gerai dokumentuota;

6. Specifikacija nesikeičia jau ilgą laikotarpį, kas reiškia, jog protokolas veikia užtikrintai ir tobulinimai nereikalingi.

Trūkumai:

1. Objektų kodavimas XML formatu labai sulėtina darbą;
2. Nesupranta lizdinio tipo (*angl. Nested set*) sąrašo;
3. Lūžta siunčiant didelius failus.

**REST** — programinės įrangos architektūra, kurios pagrindu veikia *RESTful* web servisas. *RESTful* palaiko *HTTP*, *XML*, *JSON*, *YAML* duomenų perdavimus. Taip pat, galima naudotis daugelio programavimo kalbų karkasais. PHP(Symfony, Zend Framework, CakePHP, CodeIgniter), Python(Django-piston), .NET(Open-Rasta), JAVA(Restlet, JBoss, Restfulie), Ruby(Ruby on Rails, Sinatra, Restfulie). Veikia lėtai dėl atsako vėlinimo.

Privalumai:

1. Nesukelia problemų skirstant duomenis per ugniasienes;
2. Naudingas siųsti failams, siunčiant didelį failą nenaudoja daug resursų;
3. Labai paprastas naudojimas paprastiems veiksams, tarkim pateikus užklausą gauti sveiką skaičių vieną arba nulį, kurie atitiktų teigiama arba neigiamą rezultatą.

Trūkumai:

1. Neturi aprašytų jokių formatų, standartų, kuriais yra apsieičiami duomenys, gali būti *XML*, *JSON 1.0* ar *2.0* ir *XML*;
2. Nėra aprašytos galimos klaidos sujungimo metu. Įvykus klaidai klientas negauna jokio pranešimo;
3. Nėra galimybės sukurti objektų;
4. Norint pritaikyti naudoti interneto servisą reikalingas labai ilgas programinis kodas;
5. Lėtas.

**SOAP** — plačiausiai naudojamas protokolas prisijungti prie WSDL serviso. Dažniausiai paslaugų tiekėjai pateikia kodo pavyzdžius, kaip galima jungtis, naudojantis šiuo klientu, prie jų serverių (*GOOGLE*, *AMAZON*, *YAHOO*, *ASK*, *EBAY*). *SOAP* veikia *REST* architektūros pagrindu, kuris gali grąžinti *XML* failą įvykdžius užklausą, bet galima gauti ir objektą, kas pravartu manipuliuojant duomenimis, kai atsakas yra sudarytas iš didelių duomenų kiekio. Didesnis

privalumas dirbant su objektais, nes *XML* faile, pasiekti tam tikrą vietą, reikia nuskaityti visą failą į atmintį, o objekte tai atlikti galima iš karto.

Privalumai:

1. Nesukelia problemų paskirstant duomenis per ugniasienes;
2. Plačiai naudojamas duomenų valdymo tipo sistemose;
3. Palaiko kompleksinius objektus. Taip atsisakoma didelio kiekio programinio kodo, tai palengvina naudojimą;
4. Turi tokią savybę kaip žinutės (*angl. SOAP Envelopes*), kurios dėka galima XML failą siųsti kaip SOAP žinutę;
5. Galimos klaidos yra aprašytos, joms įvykus yra grąžinama klientui;
6. Labai gerai dokumentuota;
7. Sukurta daug darbo įrankių įvairiomis programavimo kalbomis.

Trūkumai:

1. Suprogramuoti kliento dalį reikalingos stiprios objekcinio programavimo žinios;
2. Daug susijusių ir neprivalomų standartų, kartais yra klaidinantys ir pilnai neišbaigti.

**JAVA RMI** — JAVA programavimo sąsaja, kurios pagalba galima atlikti objektiškai orientuotus duomenų sujungimo veiksmus nuotoliniu būdu. Veikia tik su JAVA programavimo kalba.

Privalumai :

1. Gerai dokumentuota;
2. Lengva manipuluoti gautais duomenimis iš serverio.

Trūkumai:

1. Veikia tik su JAVA;
2. Per didelis duomenų kiekis gali pristabdyti serviso veikimą;
3. Karkaso failai turi būti kaskart kompiliuojami paleidžiant servisą;
4. Tinklo sujungimus stabdo ugniasienė. Kaskart reikia sukurti ugniasienės išimtį, kad galėtų susijungti su serveriu.



**CORBA** — bendro objekto užklauso brokerio architektūra (*angl. Common Object Request Broker Architecture*). Tai yra standartas aprašytas *OMG (Object management group)*, kuris įgalina programos komponentus, parašytus skirtingomis kalbomis, veikti keliuose kompiuteriuose vienu metu.

**CURL** — kliento tipas veikiantis *http* protokolu. Duomenis gaunami išsiuntus užklausa serveriui per *POST* arba *GET* parametrus ir gaunamas rezultatas *XML*. Palyginus su *SOAP* veikia lėtai.

**XSLT** — dar kitaip vadinama *XSL (angl. eXtensible Stylesheet Language transformations)* kalba, aprašanti *XML* dokumento transformaciją į *HTML* dokumentą arba į kitokios struktūros *XML* dokumentą. Ši kalba neretai naudojama tinklapių turiniui tiesiogiai iš *XML* duomenų vaizdavimui generuoti, tačiau tai tik vienas iš daugelio jos panaudojimo pavyzdžių. *XSLT* naudoja *XPath* elementus struktūrizuoti *XML* failą. Šią kalbą patogiu naudoti generuojant programinį kodą kuriamam *WEB* servisų sujungimo įrankiui.

## 2.3. Vartotojų analizė

### 2.3.1. Programuotojai

**Jauni programuotojai (kuklios programavimo žinios)** — didžiausia vartotojų publika, tai eiliniai įmonės darbuotojai, neturintys objektinio programavimo žinių. Tokie vartotojai kuria blogus ar kitokias sistemas pagal parengtus karkasus, kurie nereikalauja ypatingo modifikavimo. Retai naudoja tinklo paslaugas teikiamas per *web* servisus. Servisu nesinaudojama dėl šių technologijų nežinojimo ir nesugebėjimo pritaikyti. Pritaikyti *WSDL* praktikoje sugebėtų tik naudojant grafinę sąsają, kai kodo redagavimas nereikalingas.

**Vidutiniai vartotojai (programavimo žiniomis)** — paprastai tik paviršutiniškai susipažinę su viena ar dauguma programavimo kalbų, suprantantys objektinę logiką. Tokie vartotojai dažniausiai naudoja *WSDL* įrankius, kurie padeda integruoti duomenų srautus į jų kuriamas arba prižiūrimas sistemas.

**Patyrę programuotojai** — tai siauresnė žmonių grupė, lyginant su kitais programuotojais. Turi profesionalias žinias, kuria ir plėtoja *WSDL* technologiją. Neturėtų susidurti su problemomis duomenų srautą integruojant iš *WSDL* rankiniu būdu. Šios srities programuotojams vargu ar reikalingas įrankis dirbti su *WEB* servais.

### 2.3.2. Įmonių darbuotojai (neskaitant programuotojų)

Tai vartotojai, kurie yra panašūs į jau minėtą jaunų programuotojų grupę. Kai kurie gali turėti programavimo žinių. Kita vertus, tai nieko nekeičia, nes jie nėra pakankamai kompetetingi koreguoti programinį kodą. Tam, kad jie galėtų naudotis įvairiomis tinklo paslaugomis, jiems reikalingas įrankis, padėsiantis sugeneruoti programinį kodą. Kyla klausimas, ar duomenų bazės lentelės ir jų prasmė bus aiški, nes programai būtina nustatyti duomenų srautų kelius, ir nurodyti, kur jie turi būti nukreipti. Tam turėtų būti pateikta dokumentacija apie esamą duomenų bazę įmonėje.

### 2.3.3. Mokymo įstaigos

**Mokytojai, dėstytojai** — tokio tipo vartotojai turbūt sėkmingai naudoja įvairias tinklo paslaugas. Dauguma moka programuoti kuria nors programavimo kalba. Esant būtinybei yra nesunku persiorientuoti prie kitos programavimo kalbos, pakoreguoti reikiamus kodo fragmentus ar pridėti naujas funkcijas. Tokiems vartotojams įrankis yra reikalingas iš dalies. Jis nebūtinai turi apjungti keletą paslaugų, kadangi tokio tipo vartotojai yra pakankamai kvalifikuoti tai padaryti patys.

**Moksleiviai, studentai** — dauguma tikriausiai net nežino, kas tai yra WEB servisai. Tinklo paslaugų vartotojų ratas yra siauras. Greičiausiai jie jau žino kaip elgtis su SOAP ar kitais klientais ir gauti, siųsti duomenis. Be abejo, įrankis, kuris apjungtų keletą servisų ir pateiktų sugeneruotą programinį kodą, daugumai šio tipo vartotojų palengvintų pradinį darbą ir atvertų galimybę naudotis paslaugomis.

### 2.3.4. Vartotojų tikslai ir savybės

Kadangi *WSDL* pateikia įvairias paslaugas, tai bet kuris vartotojas gali tuo pasinaudoti. Žinoma, tai nėra taip paprasta, ir tam naudojami įvairūs įrankiai, kurių naudojimui reikalingos žinios: nuo elementarių iki aukšto lygio žinių. *WSDL* gali būti naudojamas įvairiose sistemose, tačiau ne visi administruojantys vartotojai turi geras programavimo žinias servisui konfigūruoti.

## 2.4. Esamų sprendimų analizė

**XMLSpy [5]** — pateikia labai platų funkcijų pasirinkimą. Palaikomi formatai *XML* ir *DTD* (angl. *Document Type Definition*). Galima apjungti keletą *WSDL* tarpusavyje ir sugeneruoti vieną XML failą, bet negalima gauti programinio kodo, kuris duomenis sukeltų į duomenų bazę. Palaiko SOAP klientą ir daugelį kitų standartų ir protokolų, tokių kaip *Microsoft Office 200*, *OOXML*(*Office*

*Open XML*), *XBRL*. Duomenis galima išsaugoti *Microsoft Word*, *Excel* formatuose. *XMLSpy* palaiko *XBRL*, atvirojo standarto verslo ir finansinės atskaitomybės tinklo paslaugą, su pilnu *XBRL* redaktoriumi. *XMLSpy* automatiškai redaguoja duomenis tikrinant *XSLT* transformaciją. Taip gaunama sintaksė, kurią galima eksportuoti į pasirinktą programavimo kalbą. Problema yra ta, kad paketas mokamas, o programinis kodas pateikiamas tik duomenų gavimui, bet ne jų eksportavimui į pasirinktą duomenų bazę.

**Stylus Studio® 2011 XML Enterprise Suite [8]** — pažangi XML kūrimo aplinka (*XML IDE*), kuri gali vieną bendrą failą sudaryti iš kelių XML failų pagal pasirinktus parametrus, tai yra duomenų srautai suskirstomi norima eile ir pozicija. Galima projektuoti *Java* ir *.NET* kalbos komponentų duomenų integravimo programas. *Stylus Studio* pateikia galingas naujas funkcijas verslo inovacijos tinklo paslaugų valdymui. Tinkamas įrankis dirbti su WSDL servisais, tačiau jis taip pat mokamas.

**IBM websphere [6]** — programinės įrangos produktų šeima, skirta aplikacijų infrastruktūrai, verslo sistemų integravimui, portalams, elektroninei komercijai, mobiliajai prieigai ir prieigai balsu.

Websphere programų paketą sudaro trys pagrindinės dalys:

**IBM WebSphere Application Server** — *Websphere* šeimos pagrindas, pateikia *J2EE v1.4* ir *WSDL* aplinką elektroninių verslo aplikacijų diegimui, integravimui ir valdymui. Naujausių *WSDL* standartų palaikymas, įvairios diegimo galimybės – nuo vieno serverio iki grupės serverių (*angl. cluster*), užtikrinančių didelį aplikacijų pasiekiamumą ir galingumą. Pateikia daug galimybių valdomų naudojant naršyklę, taip pat automatinį aplikacijų atnaujinimą pasikeitus artefaktams, platų konfigūracijų skirtingose platformose palaikymą. Našus aplikacijų kūrimas ir diegimas naudojant *Service Data Objects (SDO)*, *Java Server Faces*, *JMS* palaikymą su didelio pasiekiamumo valdykle (*angl. High Availability Manager*) ir unifikuoto serverių grupavimo sistemą (*angl. Unified Clustering Framework*).

**IBM WebSphere Portal** — siūlo vieningą personalizuotą aplinką darbui su įvairiomis aplikacijomis, darbo sekomis, turinio valdymu, bendravimu, taip vartotojui suteikiant vientisumo pojūtį. Daug bendradarbiavimo galimybių: elektroninės žinutės, atskiros darbo vietos darbuotojų komandoms, asmenų paieška ir elektroniniai susitikimai. Tai lankstus, atviras ir išplečiamas būdas kurti sėkmingus verslas-darbuotojui (*B2E*), verslas-verslui (*B2B*) ir verslas-klientams (*B2C*)

portalus. *WebSphere Portal* įgalina greitai integruoti serverio aplikacijas dėl jau paruoštų šablonų, šablonų kūrimo įrankių ir atvirų standartų, tokių kaip *JSR 168*.

**Rational Application Developer for Websphere** — anksčiau vadintas *Websphere Studio* yra labai galingas programų kūrimo įrankis, su kuriuo galima greitai projektuoti, programuoti, analizuoti, testuoti, konfigūruoti ir diegti interneto, *WSDL*, *Java*, *Java 2 Enterprise Edition (J2EE)* portalų aplikacijas. Optimizuotas *IBM Websphere* programinei įrangai, bet kartu palaikantis ir kitų kūrėjų sukurtas programų vykdymo aplinkas, tokias kaip *BEA* ir *Tomcat*. *IBM Rational Application Developer for WebSphere Software* sukurtas naudojant *Eclipse* atviro kodo platformą, todėl programų kūrėjai gali jį pritaikyti savo poreikiams ir išplėsti programavimo aplinką taip padidindami darbo našumą. Naudojant kartu su *IBM* programinės įrangos kūrimo platforma, programinės įrangos kūrėjai gali tiesiogiai iš *Application Developer* naudoti reikalavimų ir pakeitimų valdymo funkcijas.

## 2.5. Siekiamas sprendimas

Eilinis vartotojas negali naudotis *WSDL* paslauga pagal savo poreikius, nes norint atlikti kliento programinio kodo redagavimą, reikalingos programuotojo paslaugos, o kartais jos būna naudingos tik iš dalies. Tai priklauso nuo vartotojo poreikių. Pagal mūsų siekiamą sprendimą, kai norime apjungti keletą *WSDL* servisų, eiliniam vartotojui tai nėra lengvai prieinama. Būtent todėl kuriamas naujas įrankis su grafiniu valdymu internetiniame puslapyje (*angl. web application*). Naudojantis tokiu įrankiu, visas informacijos valdymo procesas automatizuojamas. Tokiu atveju vartotojui nebereikės programuotojo paslaugų, jam teliks įkelti sugeneruotą klasę į serverį su veikiančią *PHP* varikliu ir *MySQL* duomenų baze.

## 2.6. Darbo tikslas ir uždaviniai

**Tikslas** — suteikti vartotojui galimybę sujungti keletą *WSDL* servisų į vieną. Tai yra iš kelių duomenų srautų išvesti bendrą nustatytos struktūros duomenų srautą.

### Uždaviniai :

1. Iš pateiktos *WSDL* schemos sugeneruoti klases ir metodus;
2. Klasėms sugeneruoti grafinę aplinką, su galimybe ją lengvai modifikuoti;
3. Reikalinga duomenų bazė pakeitimams saugoti: konkrečioms laukų reikšmėms, keisti laukų pavadinimus, pervadinti metodus;

4. Galimybė prie projekto prijungti dar viena *WSDL* ir grafiškai modifikuoti, prisitaikant konkrečiai dalykinei sričiai;
5. Galimybė sukurtą projektą, kur bus sugeneruoti programiniai kodai iš *WSDL*, pritaikyti kitoms informacinėms sistemoms;
6. Programuoti naudojant plačiai naudojamą [10] atvirojo kodo *PHP* programavimo kalbą ir *MySQL* duomenų bazę.

Sugeneruoti klases iš *WSDL*, sugeneruoti grafinę aplinką sukurtoms klasėms, suteikti galimybę lengvai manipuluoti sugeneruotais metodais, suteikti galimybę prijungti nuo platformos nepriklausomą DBVS. Sukurtas įrankis leistų lengvai, su menkomis programavimo žiniomis, prie jau esamos informacinės sistemos prijungti išorines paslaugas.

## 2.7. Tyrimo planas

Tyrimas sudarytas iš keturių etapų: analizės, projektavimo, eksperimento ir magistro tezių rašymo.

**Analizė** — analizuojami ir lyginami *WSDL* tipai, jų panaudojimo galimybės. Taip pat analizuojama kokią programavimo kalbą ir duomenų bazę pasirinkti realizuojant sistemą.

**Projektavimo etapas** — atlikus analizę sudarytas būsimos sistemos struktūros ir duomenų bazės modelis. Pagal sudarytą projektą, naudojant pasirinktą programavimo kalbą ir duomenų bazę, kuriamas įrankis.

**Eksperimentas** — eksperimento metu bandomas naujai sukurtas įrankis su įvairiausiais *WSDL* servisais. Stebima, ar teisingai apdorojami duomenys. Taip pat testuojama, ar gautą programinį kodą galima pritaikyti prie kitos sistemos.

**Magistro tezių rašymas** — rašomos magistro tezės, taip pat įvertinama, ar visi išskelti uždaviniai įvykdyti, ar buvo išspręsta aptikta problema.

## 2.8. Rizikos faktorių analizė

Kaip ir kiekvienoje taikomojoje programoje taip ir šioje gali pasitaikyti netikslumų arba projektas gali būti nesėkmingas nors yra deramai suprojektuotas. Elektroninių paslaugų servisai gali

būti sukurti nesilaikant griežtai nustatytų W3C konsorciumo [1] standartų. Tokiu atveju naudojant tokį servisą jis gali neturėti WSDL aprašo arba būti nekorektiškas, tada sistema negalės pasinaudoti jo teikiamomis duomenų paslaugomis ir sugeneruoti grafinės sąsajos.

Kuriant taikomąją programą gali pasitaikyti įrankių stoka, programavimo kalbos trukumų, trūkstamo funkcionalumo. Realizavus nepilnai algoritmus gali būti nekorektiškai formuojama grafinė vartotojo sąsaja.

Interneto ryšys taip pat gali įtakoti sistemos veikimą, kadangi jei vieną dieną servisas yra pasiekiamas ir pagal taikomosios programos projektavimo sąsaja sumodeliuojamas serviso prijungimas, kitą dieną generuojant programinį kodą jei servisas bus neprieinamas, tikslas nebus pasiektas.

## **2.9. Rezultato kokybės kriterijai**

Pagrindinis kriterijus kokybei garantuoti yra teisingas programinio kodo generavimas iš sumodeliuotų servisų sujungimo. Taip pat standartus atitinkančių elektroninių paslaugų servisų pridėjimas ir visų galimų funkcijų aptikimas analizuojant serviso aprašą.

## **2.10. Analizės išvados**

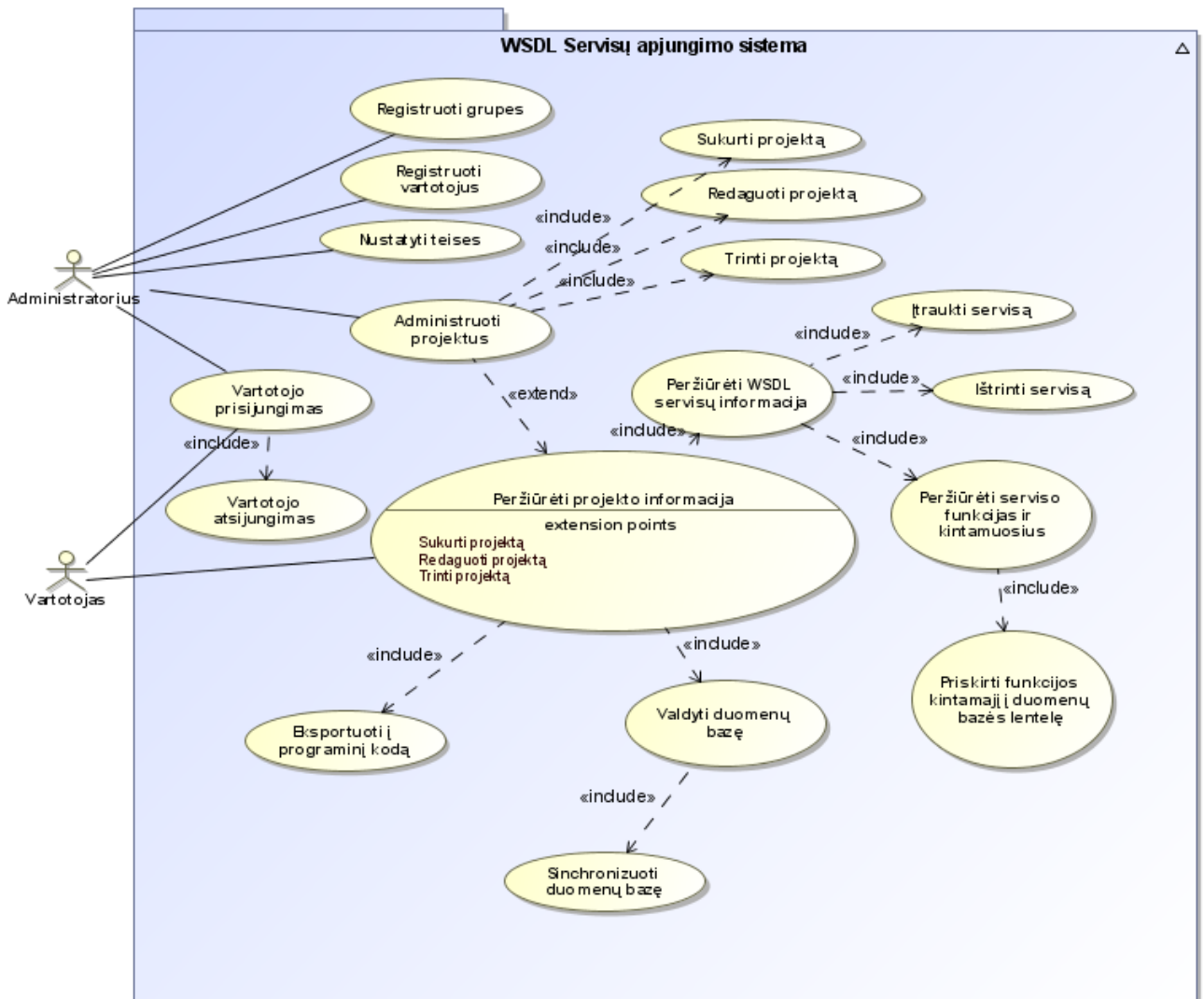
Tinklo paslaugų vartotojų aibė yra plati: nuo paprastų darbuotojų, iki profesionalių programuotojų. Visi jie įvairiai naudoja WEB servigus. Galima teigti, jog bet kuriam vartotojų tipui būtų reikalingas *WSDL* sujungimo įrankis. Analoginių sistemų siekiamam sprendimui pagal norimas funkcijas nėra, arba tai gali būti kuriama. Analogai egzistuoja, tačiau tik iš dalies suteikia galimybę dirbti su *WSDL* kaip norima, tai yra generuoja iš vieno *WSDL*, o kiti gali sujungti iš kelių, bet negali sugeneruoti kodo, kuris importuotų duomenis į pasirinktą duomenų bazę. Atlikus analogų ir tyrimo objekto analizę nuspręsta sukurti naują įrankį, kuris bus pasiekiamas kaip internetinė aplikacija. Įrankis jungsis prie *WSDL* plačiausiai naudojamu *SOAP* klientu, naudos *XSLT* transformaciją generuoti programiniam kodui iš suformuoto *XML* failo, kurį vartotojas galės naudoti savo sistemoje.

### 3. WSDL servisų sujungimas ir kodo generavimo sistemos reikalavimai

#### 3.1. Reikalavimų specifikacija

Šiame skyriuje pateikiami sistemos funkciniai ir nefunkciniai reikalavimai, kurių bus laikomasi kuriant sistemą.

##### 3.1.1. Funkciniai reikalavimai:

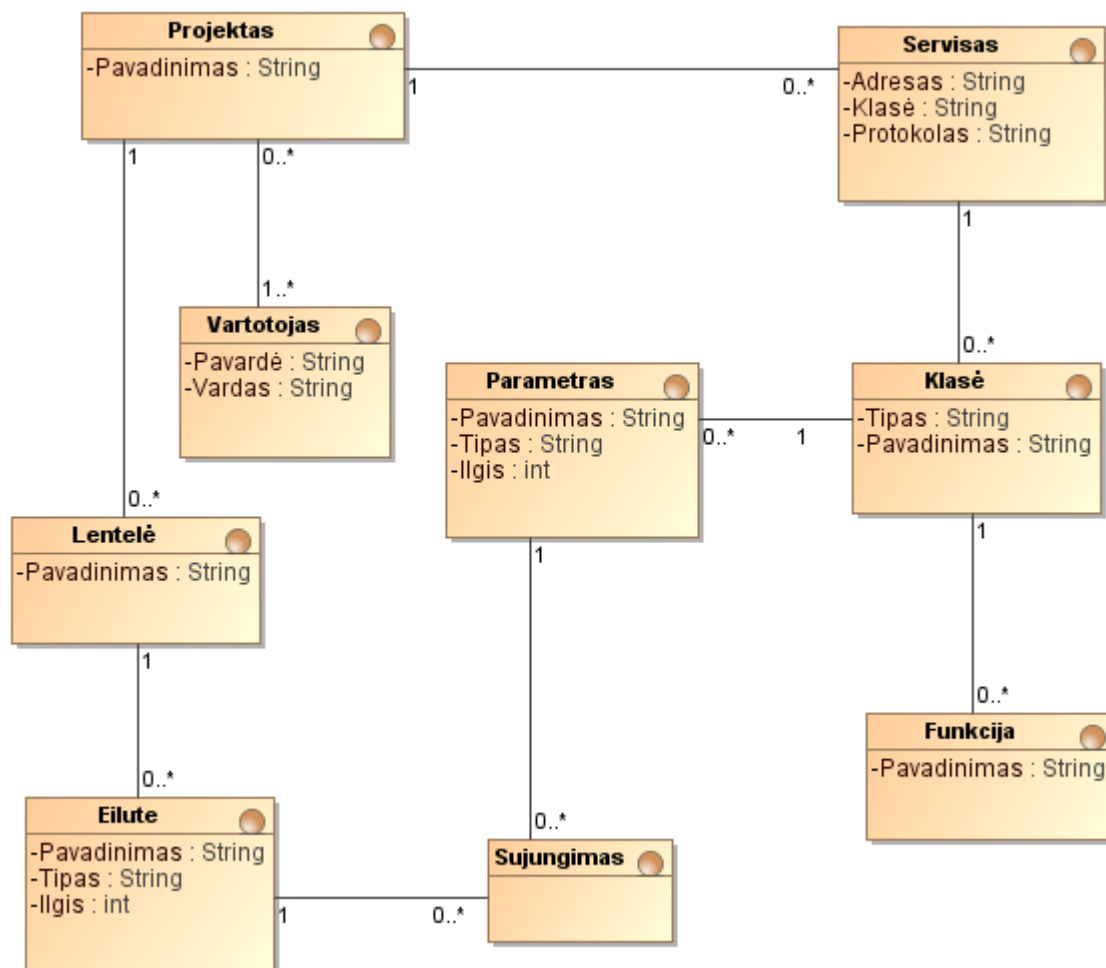


3.1 pav. Panaudojimo atvejai

Detaliau panaudojimo atvejai aprašomi funkcinųjų reikalavimų lentelėse kurios pateiktos prieduose.

### 3.2. Dalykinės srities modelis

Žemiau pateiktas (žiūrėti 3.2 pav.) dalykinės srities modelis, kurį sudaro metaduomenys „Projektas“, „Servisas“, „Vartotojas“, „Parametras“, „Klasė“, „Funkcija“, „Prijungimas“, „Eilutė“, „Lentelė“.



3.2 pav. Dalykinės srities diagrama

## 4. WSDL servišų sujungimas ir kodo generavimo sistemos projektas

### 4.1. Sistemos architektūros projektas

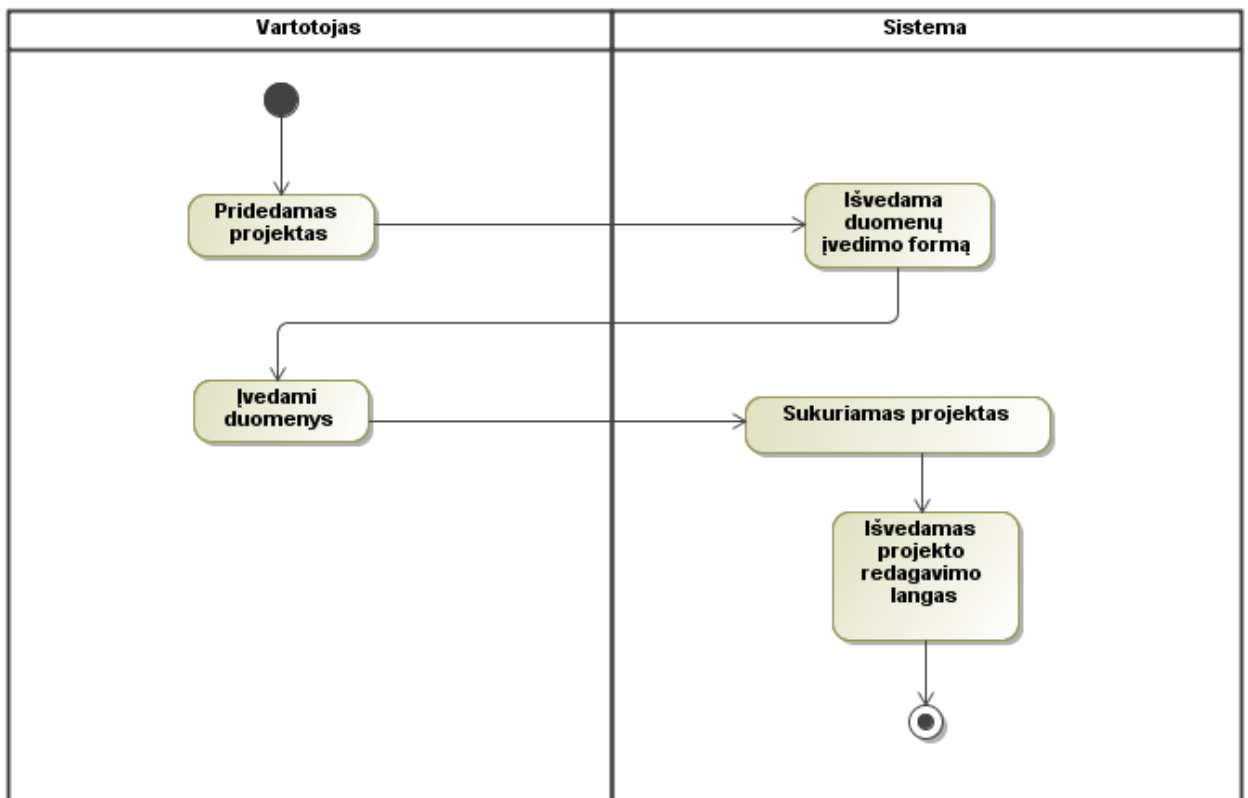
#### 4.1.1. Sistemos loginė architektūra

Šiame skyriuje pateikiamas sistemos pagrindinių funkcijų architektūros projektas, realizuotas panaudojimo atvejų diagramose. Pagrindinis funkcionalumas vaizduojamas pav. „Pridėti servisą“ ir „Eksportuoti programinį kodą“. Sistema, kaip ir minėta ankščiau, gali atlikti veiksmus su keleta elektroninių paslaugų servišų. Kiekvienas servišas išsaugomas kaip atskiras objektas, o atitinkami



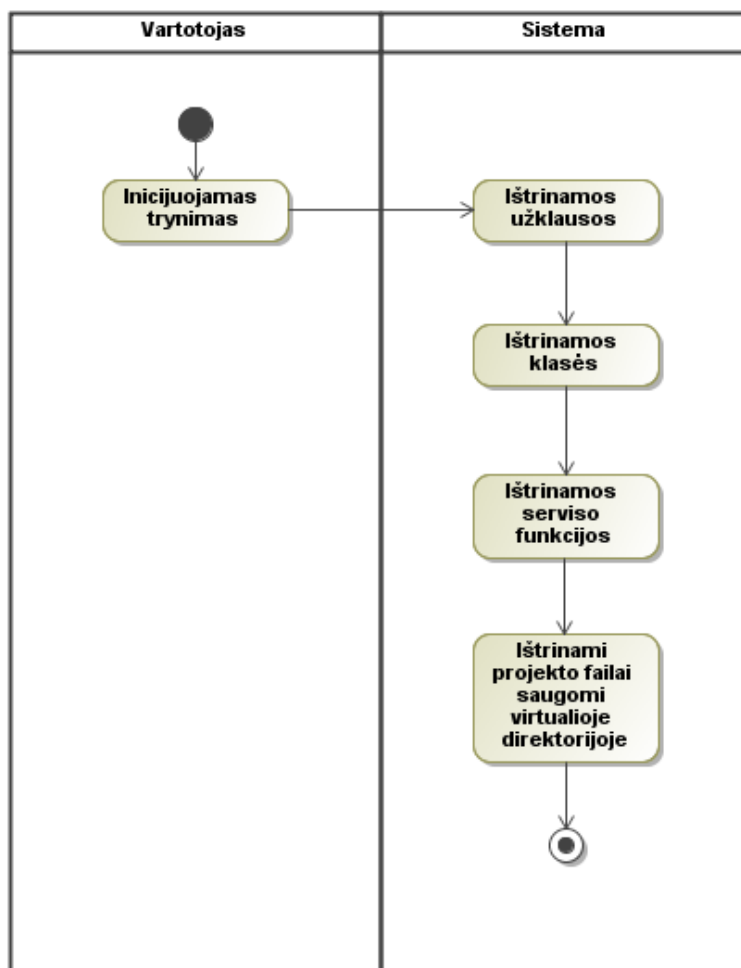
serviso komponentai, tokie kaip užklausų funkcijos, aprašo klasės, gražinami parametrai, užklausos parametrai susiejami ryšiu. Tarpusavyje servisi yra nepriklausomi, vienintelė juos siejanti vieta yra duomenų bazės lentelės. Tarp serviso ir lentelės galima sukurti ryšį, kuriame apibrėžiami užklausos parametrai ir kokie rezultatai bus naudojami.

Funkcijų, kaip vartotojų valdymo, prisijungimų ir registracijų apžvalgos nėra, nes tai nesusiję su aktuali tema, kuri analizuojama šiame darbe. Iš esmės pradinis sistemos procesas prasideda nuo projekto sukūrimo, kuris pavaizduotas 4.1 pav. „Projekto sukūrimas“. Proceso eiga prasideda inicijavus „Pridedamas projektas“ veiksmą. Vartotojas paspaudžia mygtuką „Sukurti projektą“, o sistema į tai atsako išvesdama duomenų įvedimo formą. Tinkamai užpildyti duomenys yra siunčiami į serverį, kur jie apdorojami ir išsaugomi. Tuomet vartotojui pateikiama jau darbinė aplinka, kurioje galima atlikti servisų įtraukimo ir apjungimo darbus.



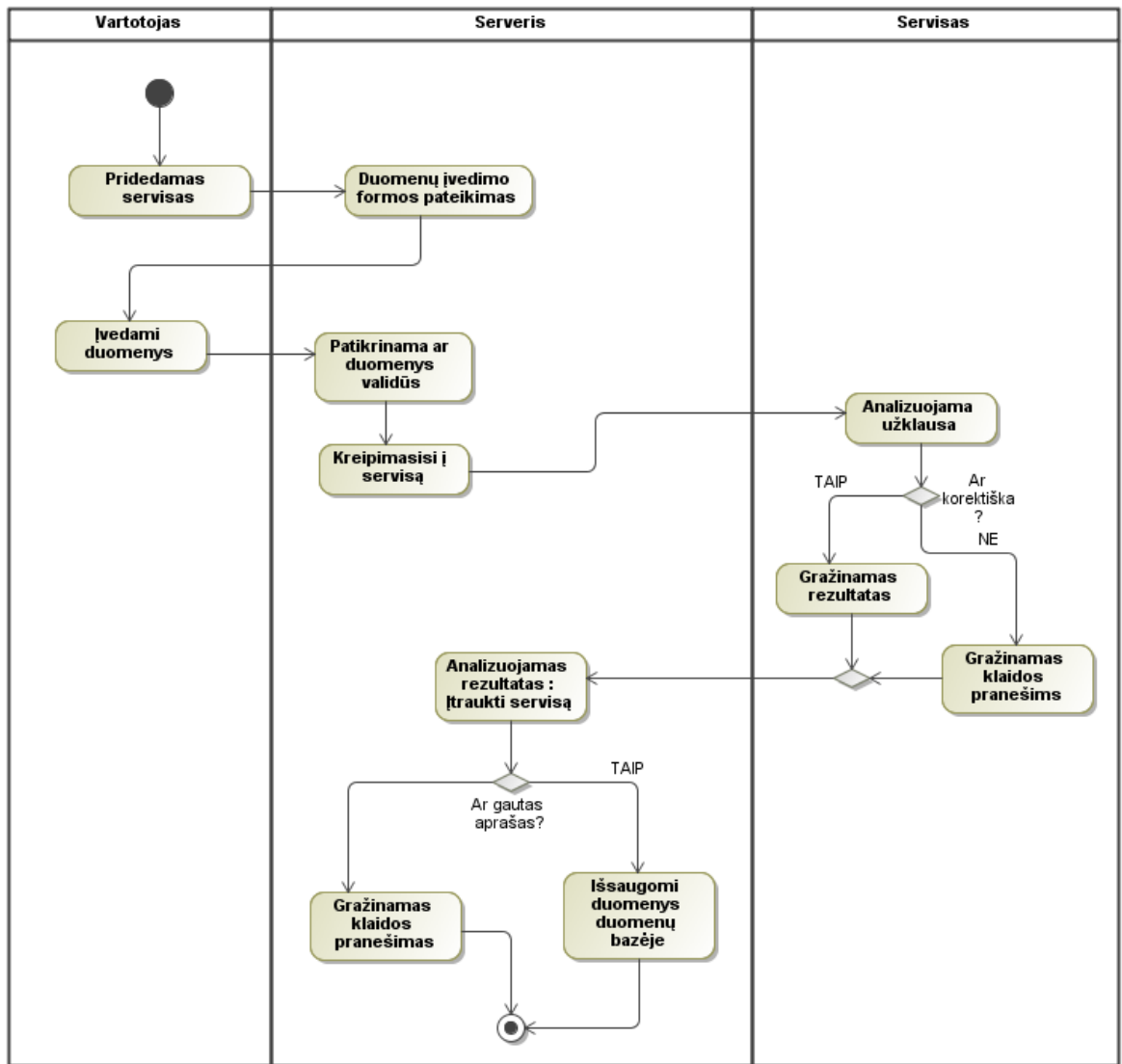
4.1 pav. Projekto sukūrimas

Taip kaip sukuriamas projektas, taip jis gali būti ir ištrinamas, kurio procesas pavaizduotas 4.2 pav. „Projekto ištrynimasis“. Inicijavimus projekto trynimą sistema surenka visus duomenų bazės įrašus apie projektą, tai yra servisi jų funkcijos, parametrų klasės, virtuali duomenų bazė.



4.2 pav. Projekto ištrynimas

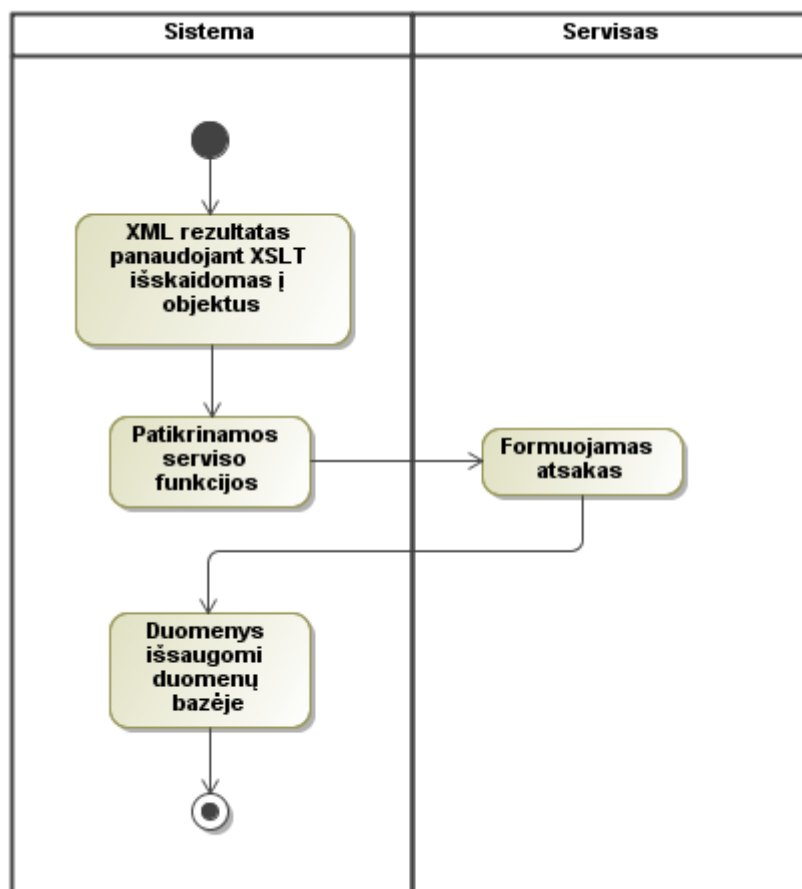
Kaip minėta projekto sukūrimo aprašyme, atlikęs visus veiksmus, vartotojas turi galimybę atlikti kitus veiksmus. Vienas iš jų pateikiamas 4.3 pav. „Įtraukti servisą“. Tai interneto serviso įtraukimas į projektą. Prie projekto servisų galima prijungti neribotą skaičių. Inicijavus serviso pridėjimą, vartotojui pateikiama duomenų įvedimo forma, kur prašoma įvesti serviso adresą ir klasės pavadinimą (klasės pavadinimas naudojamas generuojant programinį kodą). Jei duomenys korektiški, sistema kreipiasi į serverį pagal pateiktą serviso adresą. Interneto servisas, tai nepriklausomai nuo kuriamos sistemos veikianti tarnyba, kuri gali keisti savo veikimą iš anksto nieko neperspėdama. Pagal nutylėjimą kreipiantis serviso adresu grąžinamas aprašas, kuris analizuojamas 4.4 pav. „Analizuojamas rezultatas“. Jei aprašas grąžinamas korektiškas, o analizuojant pavyksta aptikti visus reikalingus kriterijus serviso kodo generavimui, tada duomenys išsaugomi į duomenų bazę, kitu atveju parodoma įvykusi klaida.



4.3 pav. Pridėti servisą prie projekto

Sistemai gavus rezultatą iš interneto serviso yra papildomai detalizuotas procesas 4.4 pav. „Analizuojamas rezultatas“. Iš pateiktos diagramos galima matyti, jog gautas rezultatas analizuojamas pasinaudojant „XSLT“ transformacijos pagalba. Transformacija suteikia galimybę iš skirtingų struktūros tipų, naudojantis sukurta stilistika, gauti fiksuotos struktūros rezultatą. Pagal tai galima atskirti skirtingas dalis, tokias, kaip serviso funkcijos, užklausių ir rezultatų parametrai iš serviso aprašo, net jeigu interneto serviso gauto aprašo „XML“ žymės nenumatyta tvarka.

Išanalizavus aprašą, sistema patikrina kiekvieną funkciją, ar ji yra teisinga, tai yra atsako į užklausimą. Užklausimas atliekamas nepaisant ar funkcija turi parametru ar ne. Iš interneto serviso tikimasi negauti pranešimo, jog tokia funkcija neegzistuoja. Atlikus visus patikrinimus, duomenys išsaugomi duomenų bazėje.

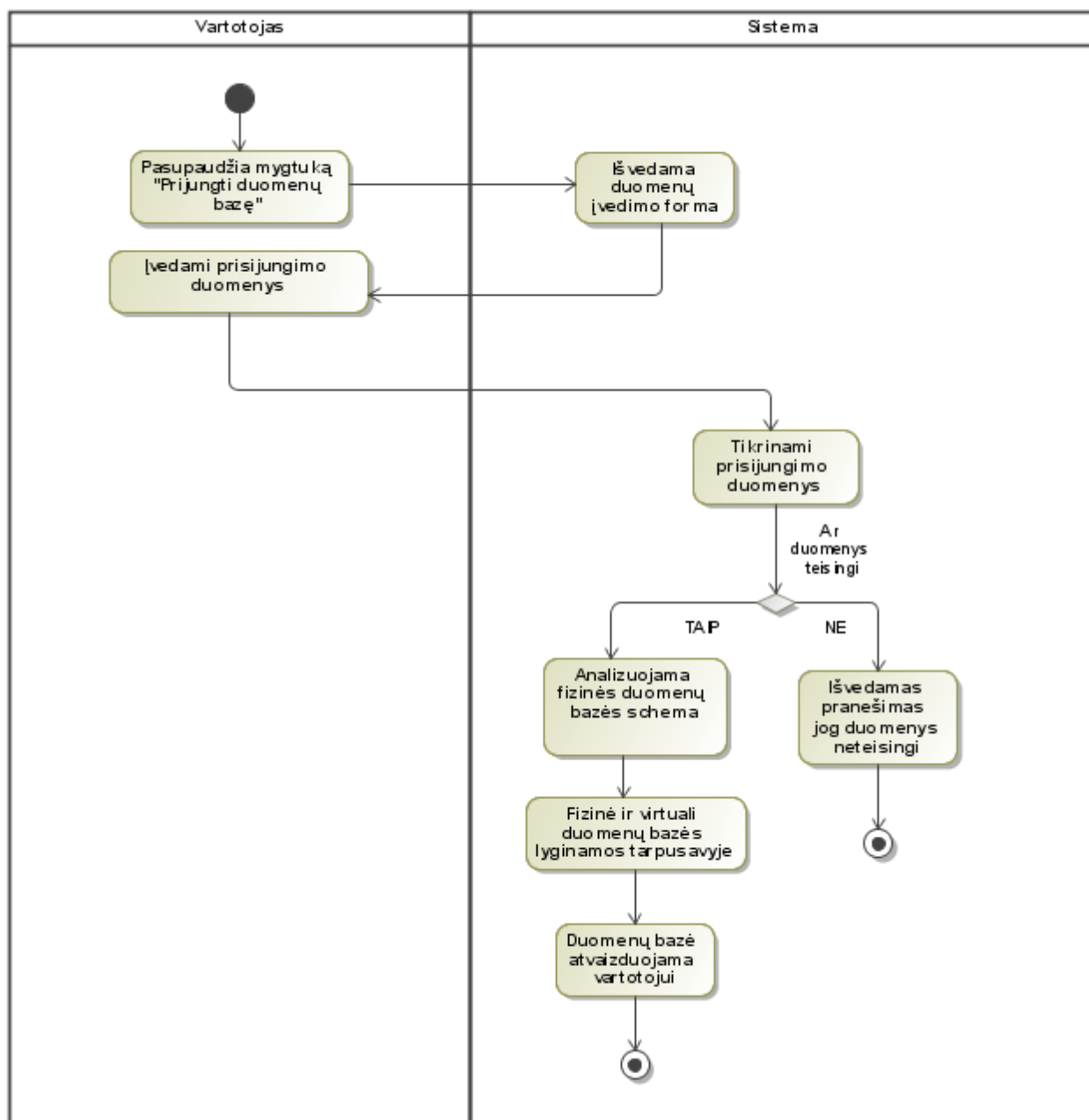


4.4 pav. Analizuojamas rezultatas

Sistema turi galimybę sinchronizuoti virtualios duomenų bazės schemą su fizine duomenų baze. Kol kas palaikoma tik „MySQL“ tipo duomenų bazė. Ši funkcija yra skirta tam, jog vartotojas galėtų prisijungti prie savo kitos turimos elektroninės sistemos duomenų bazės ir į ją prijungti interneto servisą. Duomenų bazės sinchronizavimo procesas pavaizduotas 4.5 pav. „Sinchronizuoti duomenų bazę“. Inicijavus procesą, vartotojui pateikiama duomenų įvedimo forma, kur prašoma įvesti duomenų bazės prisijungimo duomenis (adresą, kur įdiegta duomenų bazė, vartotojo vardą, vartotojo slaptažodį ir duomenų bazės pavadinimą). Jeigu sistemai, su pateiktais duomenimis, pavyksta sukurti ryšį su duomenų baze, tada procesas tęsiamas, priešingu atveju, vartotojui pranešama, kad prisijungimo duomenys yra neteisingi. Sinchronizavimo algoritmas vykdomas gaunant fizinės duomenų bazės schemą (lentelių sąrašą su kiekvieniui lentelei priklausančiais laukais, jų ilgiais ir tipais). Gauta schema lyginama su virtualia, ir jei virtualioje schemeje neegzistuoja lentelė arba laukas, jie yra sukuriami. Baigus lyginimą, pradedamas atvirkštinis procesas – kai lyginama virtuali schema su fizine duomenų baze, ir esant trūkstamų įrašų fizinėje duomenų bazėje, jie yra sukuriami. Kiekvienai lentelei papildomai yra sukuriamas laukas,

pavadinimu „wsdl\_info“ su tekstinės informacijos paskirties tipu ir 255 simboliais. Šis laukas skirtas atliktiems serviso įrašams saugoti, trumpiau tariant, išsaugoti atliktų operacijų istoriją. Saugoma informacija susideda iš Serviso klasės ir vykdytos funkcijos pavadinimo, kurias kviečiant, į duomenų bazės eilutę buvo įterptas įrašas.

Inicijavus sinchronizaciją ir po to atlikus schemos modifikacijos veiksmus, lygiagrečiai atliekami tokie patys veiksmai ir fizinėje duomenų bazėje. Operacijos gali būti tokios kaip naujos lentelės sukūrimas, lauko redagavimas ar trynimo operacijos.

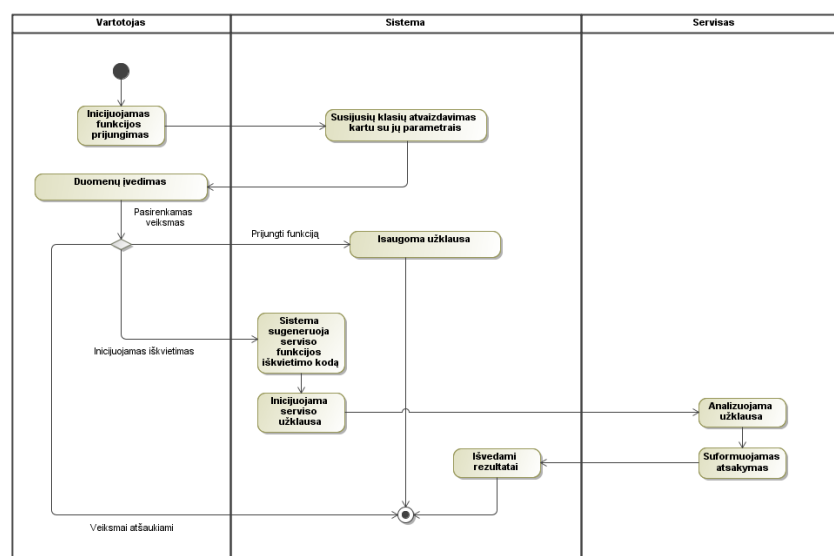


4.5 pav. Sinchronizuoti duomenų bazę

Jeigu į projektą įtraukus interneto servisą, aptinkamos funkcijos, sistema suteikia galimybę prijungti jas prie duomenų bazės lentelių. Šio veiksmo procesas pavaizduotas 4.6 pav. „Prijungti serviso funkciją į duomenų bazę“. Sistema jau žino serviso sandarą iš atliktos analizės, kuri apžvelgta 4.3 pav., taigi inicijavus funkcijos prijungimą, sistema pateikia funkcijos užklausos parametrus (jei jų yra) ir parametrus, kurie bus gražinami įvykdžius serviso funkciją. Parametrai nebūtinai gražinami vieno lygmens sąrašu, bet atitinkamai pagal serviso paskirtį: gali grįžti ir masyvų ar objektų rinkinys. Sistema supranta tipus, ir priklausomai nuo cikliško pasikartojimo, atvaizduoja parametrus medžio šakų principu. Jei parametras priklauso kuriai nors šakai, vadinasi, jis gali būti įrašomas tiek kartų, kiek bus duomenų. Prie kiekvieno parametro galima pasirinkti lentelės lauką, kuriame saugoma informacija. Sistema turi apribojimą: viena užklausa gali būti jungiama tik su viena lentele, priešingu atveju, kodo generavimas neįmanomas.

Vartotojui, suvedus užklausos duomenis, yra leidžiama patikrinti gražinamus rezultatus. Inicijavus šį veiksmą, servisas suformuoja klasės programinį kodą ir funkcijos užklausą, detaliau tai pavaizduota 4.8 pav., kur apžvelgiamas programinio kodo generavimas. Nusiuntus suformuotą užklausą, gaunamas rezultatas su realiais duomenimis, kokius gautume įvykdę užklausą. Tokiu būdu galima pasitikrinti, kokie duomenys yra gražinami, jei iš kintamojo ar parametro pavadinimo nėra aišku.

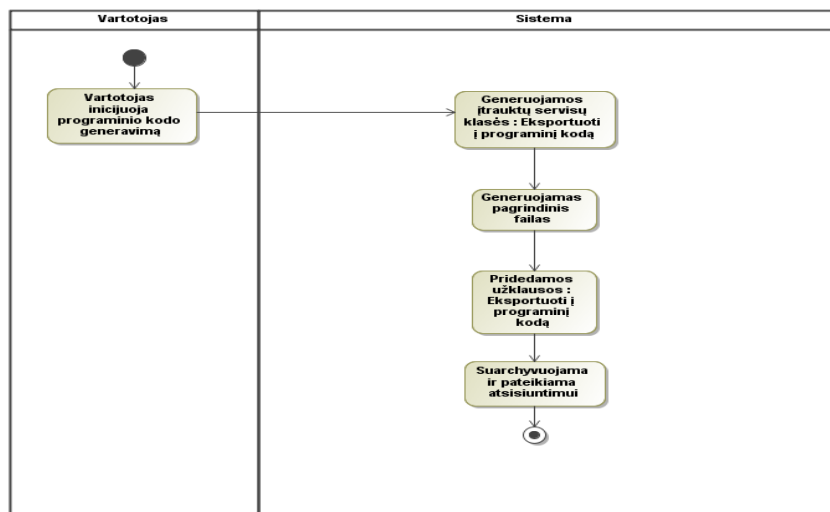
Sistema, vartotojui pasirinkus išsaugoti duomenis, atrenka, kurios užklausos yra kuriai lentelei ir automatiškai juos užkrauna projekto lange po duomenų bazės lentele.



4.6 pav. Prijungti serviso funkciją į duomenų bazę

Sistemos paskirtis yra sugeneruoti kodą, kuris atliks atitinkamas operacijas su servisų funkcijomis. Programinio kodo generavimo procesas pavaizduotas 4.7 pav. „Eksportuojamas programinis kodas“. Procesas atrodo paprastas, bet dar jo dvi dalys yra detalizuotos žemiau (žiūrėti 4.8 pav. ir 4.9 pav). Reikia atkreipti dėmesį į tai, jog kodo generatorius priklauso nuo ankščiau aprašytų procesų. Kuriant kodą yra galimybė gauti tik klases, jeigu nebus sukurta jokių užklausų arba serviso, tada generatorius nieko ir nesugeneruos. Kai procesas inicijuojamas, sistema sugeneruoja servisų klases, tai atitinkamas klasių rinkinys, kurio galima gauti duomenis iš interneto serviso. Sugeneravus ir išsaugojus programinio kodo failus virtualioje direktorijoje (iš kurios vėliau vartotojas gali vykdyti užklausas ir gauti realius rezultatus tiesiai redaguojamame projekte) yra pradedamas generuoti pagrindinis projekto failas (pagal nutylėjimą failas pavadinamas „*index.php*“). Į šį failą yra įtraukiami visi servिसai ir duomenų bazės klasė, kuri turi funkciją pavadinimu „*query()*“. Ši funkcija yra iškelta atskirai, numatant ateityje galimas sistemos modifikacijas ir pritaikymus kitoms duomenų bazėms, ar kitokio pobūdžio duomenų įrašymo poreikiams. Šitaip palikta funkcija lengva modifikuoti nesugadinant sugeneruoto programinio kodo veikimo. Kitame etape sistema analizuoja sukurtas užklausas, pagal kurias kuriami kreipimaisi į interneto servिसus, o gautas rezultatas suskaidomas iki smulkesnių elementų. Pagal pasirinktus parametrus, prijungiant funkciją į duomenų bazę (žiūrėti 4.6 pav.), atitinkami kintamieji įtraukiami į „SQL“ formuojamą užklausą. Atlikus visų servisų analizę, sistema sukuria „ZIP“ tipo archyvą ir suteikia galimybę vartotojui ją atsisiųsti.

Sistemoje nėra numatyta autentifikacijos galimybė, kadangi kiekvienas privatus servisis interpretuoja apsaugas skirtingai, ir tam nėra nustatytų griežtų standartų



4.7 pav. Eksportuojamas programinis kodas

Ankščiau paaiškintoje schemoje (žiūrėti 4.7 pav.) procesas „Generuojamos įtrauktų servisų klasės“ yra papildomai detalizuojamas, jis vaizduojamas 4.8 pav. „Įtrauktų servisų klasių generavimas“. Čia nurodoma, kokiais žingsniais sistema veikia generuodama interneto servisų programini kliento kodą. Pirmame etape sugeneruojamas masyvas, kuriame saugomi turimų parametrų klasių pavadinimai (*angl. Class mapping*). Jis yra reikalingas iškviečiant „SoapClient“ objektą. Toliau generuojamas klasių programinis kodas su atitinkamais parametrais. Atlikus visus kodo generavimus, gautas rezultatas išsaugomas virtualiame direktorijos, failo pavadinimui naudojamas serviso klasės pavadinimas. Projekte šiam klasės pavadinimui taikomas unikalumo reikalavimas, taigi problemų dėl dubliavimosi nekyla.

Svarbu pažymėti, kad sistema laikosi principo – programinį kodą saugoti kiekvienam servisui atskirame faile, nors viename faile išsaugoma daugiau nei viena klasė, tačiau tai nelemia klaidingo programos veikimo. Be abejo, išlieka tikimybė, jog serviso parametrų klasės tarp skirtingų servisų turės vienodus pavadinimus. Tokiu atveju, sistemos veikimas gali sutrikti, jei naudojamos atitinkama klase, kuri dubliuojasi

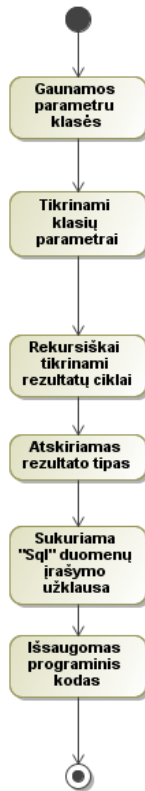


4.8 pav. Įtrauktų servisų klasių generavimas

Dar vienas detalizuojamas procesas, iš ankščiau aprašyto (žiūrėti 4.7 pav.), yra pridedamos užklausos. Tai detalizuojama 4.9 pav. „Priedamos sukurtos užklausos“. Šioje schemoje apžvelgiamas sukurtų užklausų (žiūrėti 4.6 pav.) programinio kodo generavimas. Sistema jau yra



atlikusi išplėstinę kiekvieno serviso struktūros analizę. Pagal atitinkamai sukurtas užklausas patikrinami parametrai: ar jie vis dar egzistuoja, ir ar yra tinkami naudoti, tai yra ar bus korektiškas rezultatas juos iškviečiant. Visas šis procesas yra atliekamas detaliai, nepaisant analizuojamo resurso gylio. Jeigu aptinkamas amžinas ciklas, sistema nutraukia analizę.



4.9 pav. Pridedamos sukurtos užklaupos

#### 4.1.2. Vartotojo paslaugos (sąsaja, sąsajos navigacijos planas)

Šiame skyriuje apžvelgiamas vartotojo sąsajos modelis. Tokios bazinės funkcijos kaip vartotojų prisijungimai ir panašūs, nesusiję su darbo tema, nėra vaizduojami. Kuriant juos, galima realizuoti savo nuožiūra, atitinkami kriterijai šioms sritims jau aprašyti (žiūrėti 3.1.2 punkte „Nefunkciniai reikalavimai“). Pagrindiniai veiksmai prasideda nuo projekto sukūrimo, tada jis nukreipiamas į visų projektų sąsają, kurios prototipas pavaizduotas 4.10 pav. „Visi projektai“. Šiame lange vartotojui pateikiamas sukurtų sistemoje projektų sąrašas. Jeigu vartotojas turi atitinkamas teises, jis gali matyti tik savo arba visų vartotojų projektus.

## Projektai

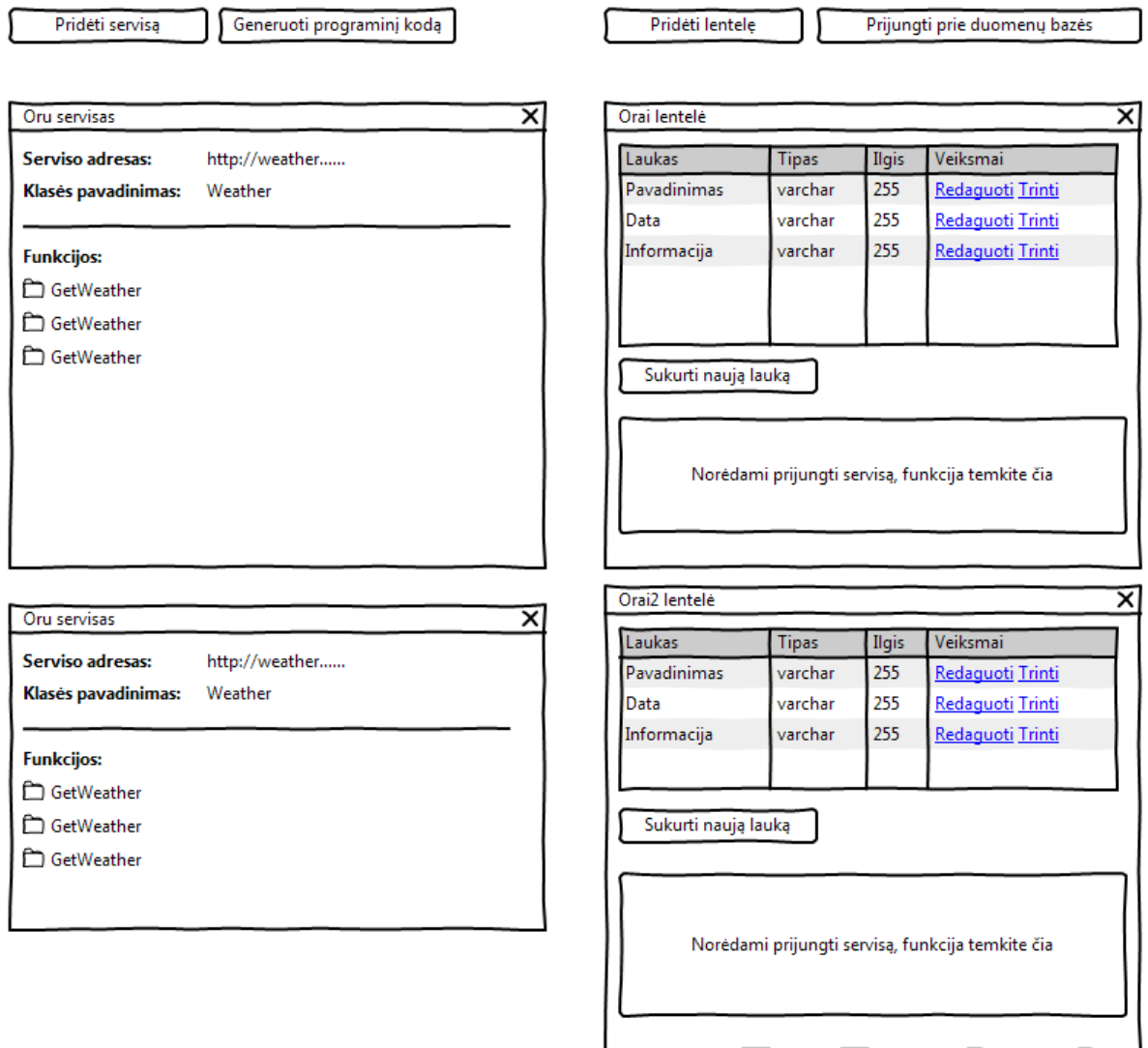
ID	Pavadinimas	Projektą sukūrė	Veiksmai
1	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>
2	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>
3	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>
4	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>
5	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>
6	Orai	Vardenis pavardenis	<a href="#">Redaguoti</a> <a href="#">Trinti</a>

4.10 pav. Visi projektai

Vartotojui, pasirinkus redaguoti projektą, yra atidaromas pagrindinis sistemos darbo langas, kuriame atliekami visi veiksmai. Jame komponentai suskirstomi į dvi dalis: duomenų bazę (dešinėje pusėje) ir prijungtus servisus (dešinėje pusėje), taip kaip pavaizduota 4.11 pav. „Projekto redagavimas“. Viršutinėje servisų pusės dalyje pateikiami pagrindiniai valdymo mygtukai, kuriais galima atlikti pagrindinius veiksmus: „Pridėti servisą“ – rezultatas vaizduojamas žemiau (*žiūrėti 4.12 pav.*) ir „Generuoti programinį kodą“, kuris taip pat aptartas (*žiūrėti 4.13 pav.*). Įtraukti servais vaizduojami sąrašo principu. Ties viršutine dalimi išvedamas klasės pavadinimas, o kiek dešiniau – serviso pašalinimo mygtukas (pašalinus servisą, pašalinama ir prijungimų informacija duomenų bazės lentelių informacijoje). Serviso elementuose pateikiama informacija, kuri buvo suvesta įvedant naują servisą, o žemiau rašto klasės, kurias sistema aptiko serviso aprašo analizėje.

Dešinėje pusėje pavaizduota duomenų bazės projektavimo aplinka. Čia kiekvienas objektas atitinka duomenų bazės lentelę. Mygtukai, esantys aukščiau, atlieka funkcijas, tokias kaip „Pridėti lentelę“ ir „Prijungti prie duomenų bazės“. Pasirinkus vieną iš šių veiksmų, inicijuojamas veiksmas, kuris yra analogiškas aprašytam žemiau „Pridėti servisą“ lange (*žiūrėti 4.12 pav.*). Erdvė pavadinta „Norėdami prijungti servisą, funkcija temkite čia“ skirta priimti funkcijos prijungimo inicijavimą. Šis veiksmas atliekas iš kurio nors įtraukto serviso funkcijų sąrašo, tempiant funkciją į šia vietą. Tai atlikus, iškviečiamas langas „Prijungti servisą“ (*žiūrėti 4.13 pav.*)

## Projekto redagavimas



4.11 pav. Projekto redagavimas

Žemiau pavaizduota programos dalis yra modulinis iššokantis informacinio tipo langas (*angl.* „*popup*“), kuriame gali būti pateikiama atitinkama forma. Pavaizduotas vienas iš galimų variantų, bet yra ir kitų veiksmų, kuriems gali būti panaudojamas toks vaizdavimo tipas. Šis tipas pasirinktas siekiant sukurti vartotojui intuityvią programos naudojimo aplinką.

Pridėti prie projekto serviso

Serviso adresas:

Klasės pavadinimas

Pridėti

4.12 pav. Pridėti servisą

Viena iš pagrindinių šio darbo funkcijų, yra prijungti atitinkamą serviso funkciją prie duomenų bazės atitinkamų laukų. Šios funkcijos inicijuotas veiksmas pavaizduotas žemiau (žiūrėti 4.13 pav. „Prijungti servisą“). Programos lange pavaizduoti įvedimo ir išvedimo parametrai. Rezultatams gauti užklauskos parametrai nėra būtini, jei jų nėra, sistema tiesiog nerodys parametru. Kiek žemiau yra pavaizduotas galimas funkcijos gražinamų rezultatų pavyzdys. Ties kiekvienu kintamuoju, pateikiamas pasirinkimo tipo laukas, kuriame galima pasirinkti prijungiamos lentelės lauko pavadinimą.

Vartotojui yra suteikiama papildoma galimybė iškviešti serviso funkciją, kurią dabar bandoma prijungti, tai minima sistemos loginėje architektūroje (žiūrėti 4.6 pav.).

Prijungti servisą prie lentelės

Užklauskos parametras 1

Užklauskos parametras 2

.....

Išvesti serviso funkcija

Rezultatai:

☞ Kintamųjų grupė (masyvas)

☞ Kintamas

☞ Kintamas

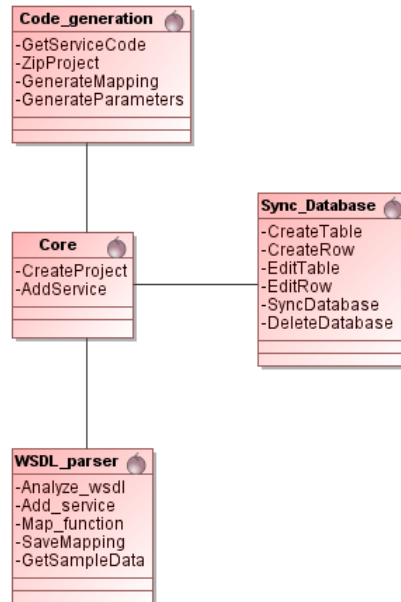
☞ Kintamas

Prijungti servisą

4.13 pav. Prijungti servisą

### 4.1.3. Veiklos paslaugos

Veiklos paslaugų schema pateikiama žemiau (žiūrėti 4.14 pav.). Čia pavaizduoti pagrindiniai kontrolieriai ir jų funkcijos.



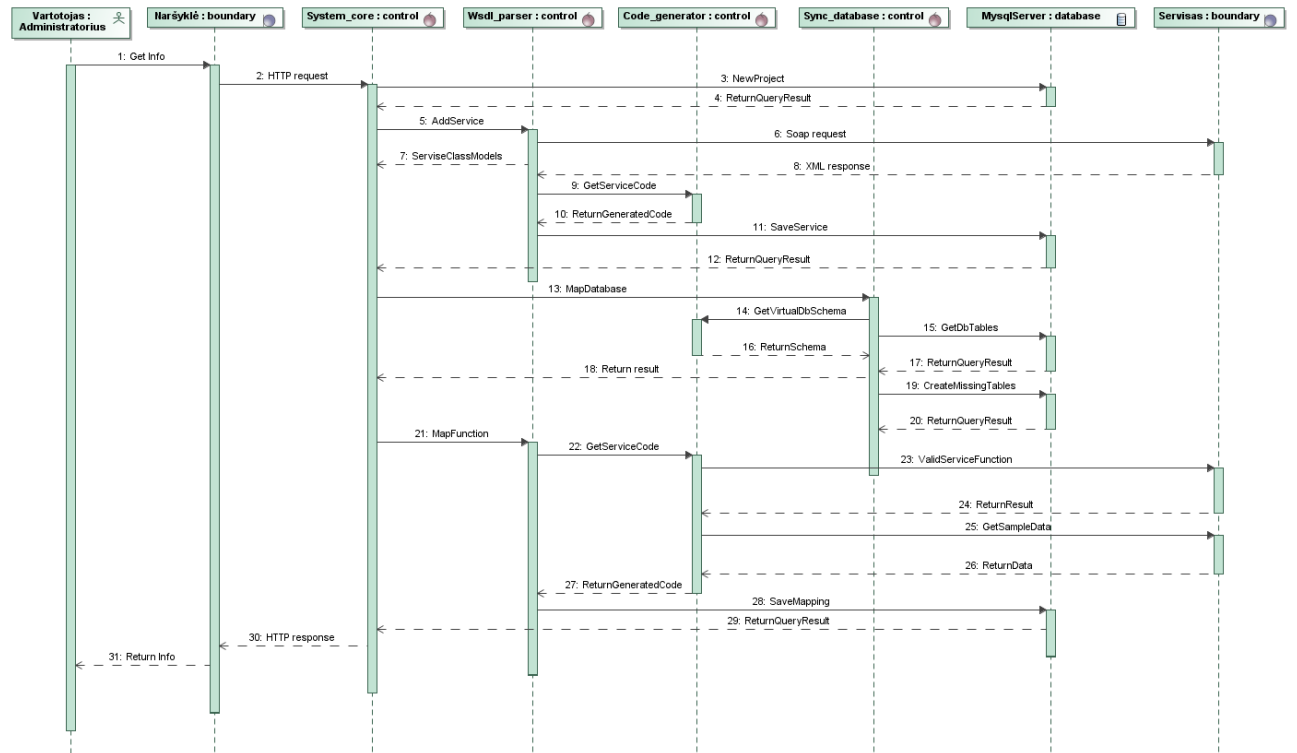
4.14 pav. Veiklos paslaugos

### 4.1.4. Duomenų paslaugos

Sistema suformavus programinį kodą gali gauti įvairius duomenis iš elektroninių paslaugų servisų. Kodas sugeneruojamas pasinaudojant wsdl aprašu. Duomenys gaunami kreipiantis pagal atitinkamas servisų funkcijas, kurioms gali būti papildomai nurodomi užklausų parametrai (jeigu jie reikalingi).

## 4.2. Sistemos elgsenos modelis

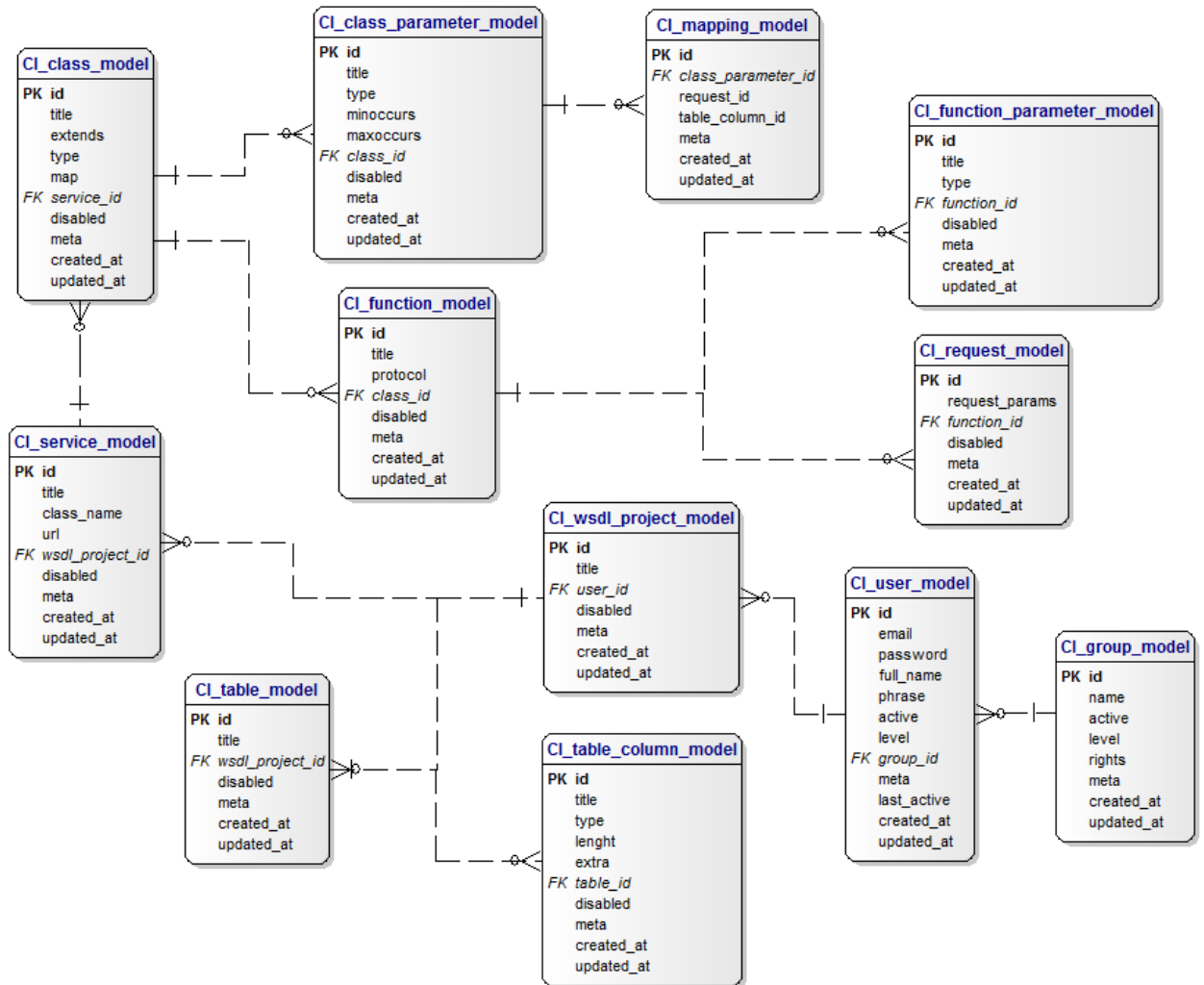
Šiame skyriuje aprašomas sistemos elgsenos modelis, kuris pavaizduotas sekų diagrama 4.15 pav.



4.15 pav. Sistemos elgsenos modelis

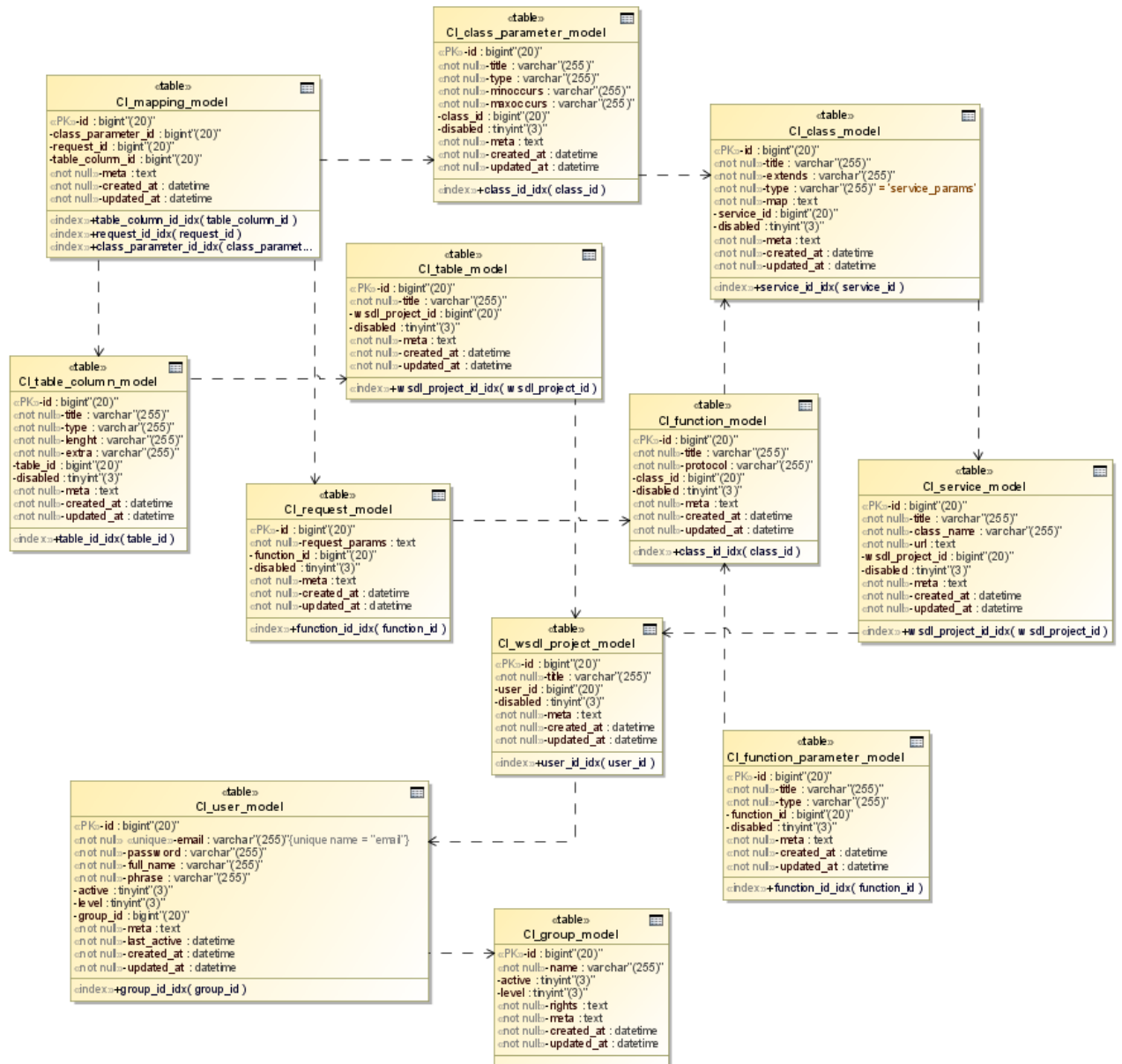
### 4.3. Duomenų bazė

4.16 pav. pateikia sistemos duomenų bazės esybių ryšių modelis.



4.16 pav. Duomenų bazės ER diagrama

Duomenų bazės lentelių schema pavaizduota 4.17 pav.



4.17 pav. Duomenų bazės schema

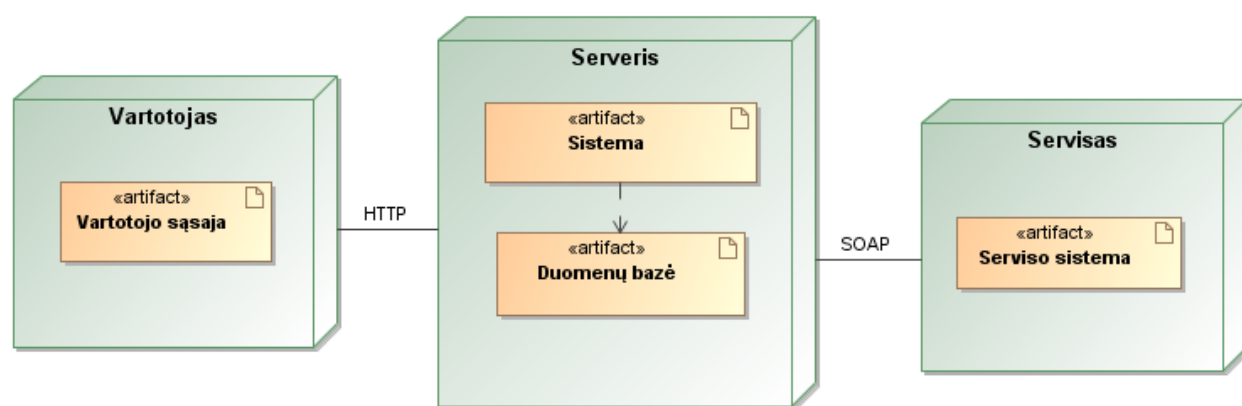




## 5. WSDL servisų sujungimas ir kodo generavimo sistemos realizacija

### 5.1. Sistemos integravimas

Realizacijos integravimo schema pavaizduota 5.1 pav. Sistemai realizuoti reikalingas serveris, kuriame būtų įdiegtas interneto servisas, galintis interpretuoti PHP programavimo kalbą (gali būti vienas iš šių: „Apache“, „IIS“, „Ngnix“, „LightHTTP“). Vartotojo sąsaja pasiekama per naršyklę, kuri per HTTP protokolą bendrauja su serveriu. Serveryje, be interneto serviso, dar turi būti įdiegtas duomenų bazės serveris. Sistema palaiko tik „MySQL“ tipo duomenų bazės serverį. Serveriui bendrauti su elektroninių paslaugų servais reikalingas „PHP“ papildinys „php\_soap“, kuris įdiegiamas papildomai. Taip serveris naudodamas „SOAP“ protokolą gali siųsti ir gauti duomenis su elektroninių paslaugų serveriu.



5.1 pav. Sistemos integravimo schema

Vartotojas, naudodamasis intuityvia grafine sąsaja, gali iškviešti atitinkamas funkcijas. Kadangi sistema pasiekama per HTTP protokolą, paslaugas galima pasiekti per įvairius įrenginius: tiek mobilius telefonus, tiek planšetinius ar nešiojamus kompiuterius.

### 5.2. Sistemos veikimo aprašymas

Darbas su sistema pradedamas nuo vartotojo prisijungimo. Sistema gali būti įdiegta į serverį ir prieinama tiek viešai, tiek vietiniame potinklyje. Tai priklauso nuo „DNS“ serviso konfigūracijos. Tokiu atveju, reikalinga vartotojų identifikavimo sistema, kurios pavyzdys pateikiamas 5.2 pav.

5.2 pav. Vartotojų prisijungimo forma

Prisijungus, vartotojui pateikiamas projektų sąrašas (žiūrėti 5.3 pav.). Čia galima sukurti naują projektą arba redaguoti jau sukurtus projektus.

ID	Pavadinimas	Sukūrė	Veiksmai
1	Oras	[simonas.serlinskas@gmail.com] Simonas S	[edit] [delete]
15	Valiutos	[simonas.serlinskas@gmail.com] Simonas S	[edit] [delete]
16	PasaulioOrai	[edvinas.sinkevicius@ktu.lt] Edvinas Šinkevičius	[edit] [delete]

5.3 pav. Projektų sąrašas

Pasirinkus kurti naują projektą, pateikiama duomenų įvedimo forma, kurioje užklauiamas projekto pavadinimas. Pavadinimui taikoma duomenų įvedimo apsauga, kuri tikrina ar neegzistuoja kuriamas projektas tokiu pačiu pavadinimu. Nusprendus redaguoti arba sukūrus projektą, pateikiama pagrindinė darbo aplinka, kuri pavaizduota 5.4 pav.

**Projekto redagavimas**

Čia galite pridėti naują servisą ir atlikti su juo veiksmus.

[IVESTI NAUJĄ SERVISĄ](#)
[GENERUOTI PROGRAMŲ KODĄ](#)

[SUKURTI LENTELĘ](#)
[PERŽIŪRĖTI DUOMENIS](#)
[ATSIJUNGTI NUO DUOMENŲ BAZĖS](#)

Prijungta DB: saimaz\_wsdl1

**Weather2** [Redaguoti](#) [Trinti](#)

Serviso aprašo adresas: <http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl>  
 Klasės pavadinimas: Weather2

Funkcijos:

- [GetWeatherInformation](#)
- [GetCityForecastByZIP](#)
- [GetCityWeatherByZIP](#)

**Currency** [Redaguoti](#) [Trinti](#)

Serviso aprašo adresas: <http://www.websvcx.net/currencyconverter.asmx?wsdl>  
 Klasės pavadinimas: Currency

Funkcijos:

- [ConversionRate](#)

**Weather** [Redaguoti](#) [Trinti](#)

Serviso aprašo adresas: <http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl>  
 Klasės pavadinimas: Weather

Funkcijos:

- [GetWeatherInformation](#)
- [GetCityForecastByZIP](#)
- [GetCityWeatherByZIP](#)

**Oras** [Pridėti eilutę](#) [Redaguoti lentelę](#) [Trinti](#)

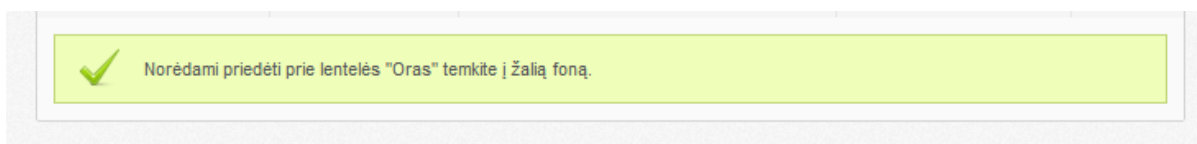
Pavadinimas	Tipas	Ilgis	Veiksmai
id	int	10	<a href="#">Pridėti</a> <a href="#">Trinti</a>
wsdl_info	text		
Miestas	varchar	255	<a href="#">Pridėti</a> <a href="#">Trinti</a>
Data	varchar	255	<a href="#">Pridėti</a> <a href="#">Trinti</a>
Temperatura	varchar	255	<a href="#">Pridėti</a> <a href="#">Trinti</a>
success	varchar	255	<a href="#">Pridėti</a> <a href="#">Trinti</a>

**Prijungti servisai**

Lent. eilutė	Klasė	Funkcija	Kintamasis	Veiksmai
<b>Užklausa:</b>				
success	Weather2	GetCityForecastByZIP	Success	<a href="#">Trinti</a>
Miestas	Weather2	GetCityForecastByZIP	City	<a href="#">Trinti</a>
Data	Weather2	GetCityForecastByZIP	Date	<a href="#">Trinti</a>
Temperatura	Weather2	GetCityForecastByZIP	DaytimeHigh	<a href="#">Trinti</a>
<b>Užklausa:</b>				
id	Weather2	GetWeatherInformation	WeatherID	<a href="#">Trinti</a>
success	Weather2	GetWeatherInformation	Description	<a href="#">Trinti</a>

5.4 pav. Projekto redagavimas

Projekto redagavimo lange vartotojui suteikiama galimybė atlikti norimus duomenų prijungimus. Servisų lentelėse esančias funkcijas galima tempti į lentelę. Pradėjus tempti, prie kiekvienos lentelės atsiranda žalia zona (žiūrėti 5.5 pav.), nutempus virs šios zonos ir paleidus – išskviečiamas valdymo langas (žiūrėti 5.6 pav.).



5.5 pav. Funkcijos prijungimo tempimo vieta

Parametrų pasirinkimo lange galima pasirinkti tiek užklauso parametrus (jei jų reikalauja funkcija), tiek rezultatų paskirstymą į duomenų bazės lenteles. Atitinkamai kiekvienam grįžtamam funkcijos parametru, galima pasirinkti lentelės lauką. Suvedus duomenis ir paspaudus išsaugoti – lentelių vaizdas atnaujinamas. Funkcijų prijungimai matomi lentelių apačioje, punkte „Prijungti servisai“, kiekviena užklausa atitinka atskirą funkciją.

Funkcijos prijungimas
✕

**Funkcijos užklauso parametrai**

ZIP:

Kreiptis į serviso funkciją

**Funkcijos gražinami rezultatai**

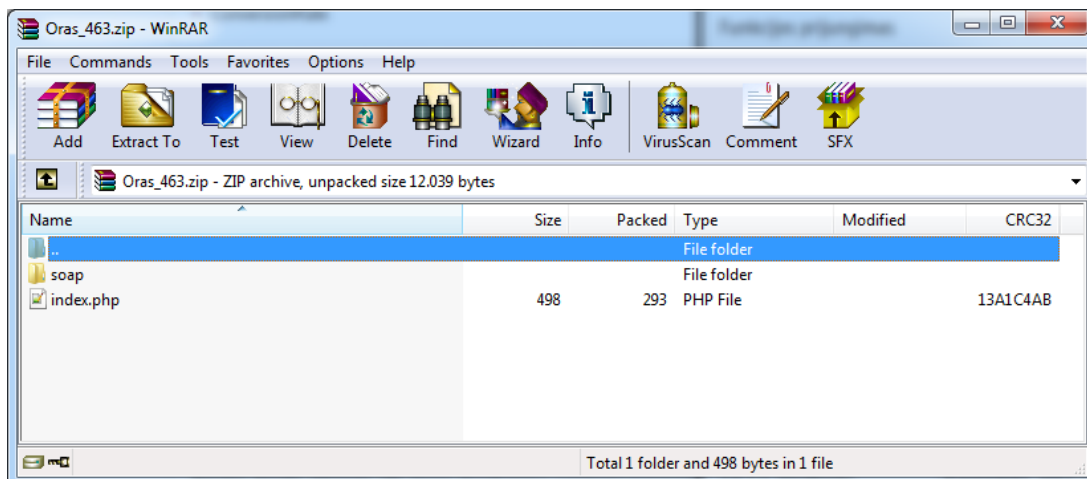
1.  
 GetCityForecastByZIPResult:
 

1.1. Success:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.2. ResponseText:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.3. State:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.4. City:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.5. WeatherStationCity:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6. ForecastResult:	
1.6.1. Forecast:	
1.6.1.1. Date:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.2. WeatherID:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.3. Description:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.4. Temperatures:	
1.6.1.4.1. MorningLow:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.4.2. DaytimeHigh:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.5. ProbabilityOfPrecipitation:	
1.6.1.5.1. Nighttime:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>
1.6.1.5.2. Daytime:	<input style="width: 100%; height: 15px;" type="text" value="Pasirinkite arba palikite tuščia"/>

SAUGOTI
ATŠAUKTI

5.6 pav. Funkcijos prijungimo parametru forma

Atlikus visus norimus prijungimo veiksmus, sistema suteikia galimybę sugeneruoti programinį kodą, kuris vykdys vartotojo suprojektuotus veiksmus. Tai atliekama paspaudus mygtuką „Generuoti programinį kodą“. Įvykdžius užklausą, pateikiamas „ZIP“ tipo archyvas atsisiuntimui (žiūrėti 5.7 pav.). Archyve pateikiamas kodas, kuris suskirstytas į atitinkamas dalis. Tai „index.php“ failas skirtas duomenų bazės klasei ir servisų funkcijų iškvietimui (toms, kurioms sukurtos užklausos). Taip pat archyve yra „soap“ aplankas, kuriame saugomas kiekvieno serviso klasės failas.



5.7 pav. Sugeneruoto programinio kodo archyvo turinys

Vartotojams, kurie nežino, kaip tinkamai elgtis su programiniu kodu, sistema suteikia galimybę peržiūrėti serviso gautus duomenis. Paspaudus mygtuką „Peržiūrėti duomenis“, pateikiamas virtualus lentelės duomenų bazės vaizdas (žiūrėti 5.8 pav.), kuriame galima iškviešti sukurtas užklausas.

id	wsdl_info	Miestas	Data	Temperatura	success
1	Weather2: GetWeatherInformation				Thunder Storms
2	Weather2: GetWeatherInformation				Partly Cloudy
3	Weather2: GetWeatherInformation				Mostly Cloudy
4	Weather2: GetWeatherInformation				Sunny
5	Weather2: GetWeatherInformation				Rain
6	Weather2: GetWeatherInformation				Showers
7	Weather2: GetWeatherInformation				Haze
9	Weather2: GetWeatherInformation				Partly Sunny
10	Weather2: GetWeatherInformation				Mostly Sunny

5.8 pav. Serviso rezultato duomenų vaizdas

### 5.3. Testavimo modelis ir duomenys, kontrolinis pavyzdys

Eksperimento tikslas yra ištestuoti elektroninių paslaugų servisų apjungimą ir sugeneruoto programinio kodo teisingumą. Pagal pateiktą naudojimosi instrukciją (žiūrėti 5.2 punktą) yra pridamas interneto servisas. Šiuo atveju orų duomenis teikiantis servisas. Serviso aprašas pasiekiamas adresu „http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl“. Įvedus servisą projekte sistema pateikia funkcijų sąrašą (žiūrėti 5.9 pav.)

Serviso aprašo adresas: <http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl>

Klasės pavadinimas: Weather2

Funkcijos:

- GetWeatherInformation
- GetCityForecastByZIP
- GetCityWeatherByZIP

5.9 pav. Testuojamo serviso funkcijų sąrašas

Pagal serviso aprašą, kurį galima pasiekti tuo pačiu adresu, pastebime, kad atrastos funkcijos teisingos. Toliau atitinkamai yra pridėdamos ir antras elektroninių paslaugų servisas, kurio paskirtis panaši. Serviso adresas: <http://www.deeptraining.com/webservices/weather.asmx?wsdl>

```
▼<wsdl:portType name="WeatherSoap">
  ▼<wsdl:operation name="GetWeatherInformation">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Gets Information for each WeatherID</wsdl:documentation>
    <wsdl:input message="tns:GetWeatherInformationSoapIn"/>
    <wsdl:output message="tns:GetWeatherInformationSoapOut"/>
  </wsdl:operation>
  ▼<wsdl:operation name="GetCityForecastByZIP">
    ▼<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      Allows you to get your City Forecast Over the Next 7 Days, which is updated hourly. U.S. Only
    </wsdl:documentation>
    <wsdl:input message="tns:GetCityForecastByZIPSoapIn"/>
    <wsdl:output message="tns:GetCityForecastByZIPSoapOut"/>
  </wsdl:operation>
  ▼<wsdl:operation name="GetCityWeatherByZIP">
    ▼<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      Allows you to get your City's Weather, which is updated hourly. U.S. Only
    </wsdl:documentation>
    <wsdl:input message="tns:GetCityWeatherByZIPSoapIn"/>
    <wsdl:output message="tns:GetCityWeatherByZIPSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

5.10 pav. Serviso aprašo iškarpa

Pridėjus servisą matomos funkcijos (žiūrėti 5.11 pav.).

Serviso aprašo adresas: <http://www.deeptraining.com/webservices/weather.asmx?wsdl>

Klasės pavadinimas: Weather3

Funkcijos:

- GetWeather



5.11 pav. Testuojamo serviso funkcijų sąrašas

Patikrinant serviso aprašą (žiūrėti 5.12 pav.), matome jog sistemos atrastos serviso funkcijos yra teisingos.







```
▼<wsdl:portType name="WeatherSoap">
  ▼<wsdl:operation name="GetWeather">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Returns the weather for a given city</wsdl:documentation>
    <wsdl:input message="tns:GetWeatherSoapIn"/>
    <wsdl:output message="tns:GetWeatherSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

5.12 pav. Serviso aprašo iškarpa

Toliau ištestuojame duomenų bazės sinchronizavimą. Susikuriame dvi lenteles su keletą laukų (žiūrėti 5.13 pav.).

Orai			
Pavadinimas	Tipas	Ilgis	Veiksmai
wSDL_info	text		
Rezultatas	varchar	255	 




  

Orai2			
Pavadinimas	Tipas	Ilgis	Veiksmai
wSDL_info	text		
ID	int	10	 
Aprašymas	varchar	255	 
Pav	varchar	255	 










5.13 pav. Sukurtos lentelės servisų testavimui

Toliau prijungiame po funkciją iš abiejų servisų (žiūrėti 5.14 pav.).



Orai				
Pavadinimas	Tipas	Ilgis	Veiksmai	
wSDL_info	text			
Rezultatas	varchar	255	 	
Prijungti servaisai				
Lent. eilute	Klasė	Funkcija	Kintamasis	Veiksmai
Užklausa:				
Rezultatas	jop	GetWeather	GetWeatherResult	

Orai2				
Pavadinimas	Tipas	Ilgis	Veiksmai	
wSDL_info	text			
ID	int	10	 	
Aprašymas	varchar	255	 	
Pav	varchar	255	 	
Prijungti servaisai				
Lent. eilute	Klasė	Funkcija	Kintamasis	Veiksmai
Užklausa:				
ID	Weather2	GetWeatherInformation	WeatherID	
Aprašymas	Weather2	GetWeatherInformation	Description	
Pav	Weather2	GetWeatherInformation	PictureURL	

5.14 pav. Prijungtos funkcijos prie lentelių

Pasirinkus kodo generavimą gaunamas archyvas su kodo turiniu:

soap/Orai.php failo turinys:

```
<?php
if (!class_exists('GetWeatherInformation')) {
    class GetWeatherInformation {
    }
}

if (!class_exists('GetWeatherInformationResponse')) {
    class GetWeatherInformationResponse {
        public $GetWeatherInformationResult;
    }
}

if (!class_exists('ArrayOfWeatherDescription')) {
    class ArrayOfWeatherDescription {
        public $WeatherDescription;
    }
}
```

```

    }}

    if (!class_exists('WeatherDescription')) {
        class WeatherDescription {
            public $WeatherID;
            public $Description;
            public $PictureURL;
        }
    }

    if (!class_exists('GetCityForecastByZIP')) {
        class GetCityForecastByZIP {
            public $ZIP;
        }
    }

    if (!class_exists('GetCityForecastByZIPResponse')) {
        class GetCityForecastByZIPResponse {
            public $GetCityForecastByZIPResult;
        }
    }

    if (!class_exists('ForecastReturn')) {
        class ForecastReturn {
            public $Success;
            public $ResponseText;
            public $State;
            public $City;
            public $WeatherStationCity;
            public $ForecastResult;
        }
    }

    if (!class_exists('ArrayOfForecast')) {
        class ArrayOfForecast {
            public $Forecast;
        }
    }

    if (!class_exists('Forecast')) {
        class Forecast {
            public $Date;
            public $WeatherID;
            public $Description;
            public $Temperatures;
            public $ProbabilityOfPrecipitation;
        }
    }

```

```

    }}

if (!class_exists('temp')) {
    class temp {
        public $MorningLow;
        public $DaytimeHigh;
    }
}

if (!class_exists('POP')) {
    class POP {
        public $Nighttime;
        public $Daytime;
    }
}

if (!class_exists('GetCityWeatherByZIP')) {
    class GetCityWeatherByZIP {
        public $ZIP;
    }
}

if (!class_exists('GetCityWeatherByZIPResponse')) {
    class GetCityWeatherByZIPResponse {
        public $GetCityWeatherByZIPResult;
    }
}

if (!class_exists('WeatherReturn')) {
    class WeatherReturn {
        public $Success;
        public $ResponseText;
        public $State;
        public $City;
        public $WeatherStationCity;
        public $WeatherID;
        public $Description;
        public $Temperature;
        public $RelativeHumidity;
        public $Wind;
        public $Pressure;
        public $Visibility;
        public $WindChill;
        public $Remarks;
    }
}

```

```

class w1 extends DB {

public $s;

    function __construct() {
        parent::__construct();
        $map = array('GetWeatherInformation' =>
'GetWeatherInformation', 'GetWeatherInformationResponse' =>
'GetWeatherInformationResponse', 'ArrayOfWeatherDescription' =>
'ArrayOfWeatherDescription', 'WeatherDescription' =>
'WeatherDescription', 'GetCityForecastByZIP' =>
'GetCityForecastByZIP', 'GetCityForecastByZIPResponse' =>
'GetCityForecastByZIPResponse', 'ForecastReturn' =>
'ForecastReturn', 'ArrayOfForecast' => 'ArrayOfForecast', 'Forecast' =>
'Forecast', 'temp' => 'temp', 'POP' => 'POP', 'GetCityWeatherByZIP' =>
'GetCityWeatherByZIP', 'GetCityWeatherByZIPResponse' =>
'GetCityWeatherByZIPResponse', 'WeatherReturn' => 'WeatherReturn',);
        $this->s = new SoapClient(
            'http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl',
            array('classmap' => $map,
                'trace' => 1,
                'features' => SOAP_SINGLE_ELEMENT_ARRAYS
            ));
    }

    function GetWeatherInformation() {
        $result = $this->s->GetWeatherInformation();
        foreach($result->GetWeatherInformationResult-
>WeatherDescription as $v_460) {
            $v_ID = $v_460->WeatherID;
            $v_Aprasymas = $v_460->Description;
            $v_Pav = $v_460->PictureURL;
            $q = $this->query("INSERT INTO `Orai2`
(wsdل_info, ID, Aprasymas, Pav) VALUES ('w1::GetWeatherInformation', '$v_ID',
'$v_Aprasymas', '$v_Pav')");
        }
    }
}

```

soap/Orai2.php failo turinys:

```
<?php
    if (!class_exists('GetWeather')) {
        class GetWeather {
            public $CityName;
            public $CountryName;
        }

        if (!class_exists('GetWeatherResponse')) {
            class GetWeatherResponse {
                public $GetWeatherResult;
            }
        }
    }

    class w2 extends DB {

        public $s;

        function __construct() {
            parent::__construct();
            $map = array('GetWeather' =>
'GetWeather', 'GetWeatherResponse' => 'GetWeatherResponse',);
            $this->s = new SoapClient(

'http://www.deeptraining.com/webservices/weather.asmx?wsdl',
            array('classmap' => $map,
                'trace' => 1,
                'features' => SOAP_SINGLE_ELEMENT_ARRAYS
            ));
        }

        function GetWeather() {

            $p = new GetWeather();
            $p->CityName = 'Vilnius';
            $p->CountryName = 'Lithuania';
            $result = $this->s->GetWeather($p);
            $v_Rezultatas = $result->GetWeatherResult;
            $q = $this->query("INSERT INTO `Orai` (wsdl_info,
Rezultatas) VALUES ('w2::GetWeather', '$v_Rezultatas')");

        }
    }
}
```

```
}
```

### Ir index.php failo turinys:

```
<?php
$files_location = isset($defined) ? $defined : 'soap/*.php';

class DB {
    public $db_execute;

    function __construct() {
        $this->db_execute = new mysqli('','','','');
    }

    function query($sql) {
        return $this->db_execute->query($sql);
    }
}

foreach (glob($files_location) as $filename) {
    include $filename;
}

$s = new w1();
$s->GetWeatherInformation();

$s = new w2();
$s->GetWeather();
```

Programinis kodas sugeneruotas validus ir tinkamai kreipiasi į servisus. Gauti testiniai duomenys iš sujungtų servisų pateikti pav 5.15

■ Orai	
wsd_info	Rezultatas
Weather:GetWeather	Cloudy
w2:GetWeather	Sunny

■ Orai2			
wsd_info	ID	Aprasymas	Pav
w1:GetWeatherInformation	1	Thunder Storms	<a href="http://ws.cdnyne.com/WeatherWS/Images/thunderstorms.gif">http://ws.cdnyne.com/WeatherWS/Images/thunderstorms.gif</a>
w1:GetWeatherInformation	2	Partly Cloudy	<a href="http://ws.cdnyne.com/WeatherWS/Images/partlycloudy.gif">http://ws.cdnyne.com/WeatherWS/Images/partlycloudy.gif</a>
w1:GetWeatherInformation	3	Mostly Cloudy	<a href="http://ws.cdnyne.com/WeatherWS/Images/mostlycloudy.gif">http://ws.cdnyne.com/WeatherWS/Images/mostlycloudy.gif</a>
w1:GetWeatherInformation	4	Sunny	<a href="http://ws.cdnyne.com/WeatherWS/Images/sunny.gif">http://ws.cdnyne.com/WeatherWS/Images/sunny.gif</a>
w1:GetWeatherInformation	5	Rain	<a href="http://ws.cdnyne.com/WeatherWS/Images/rain.gif">http://ws.cdnyne.com/WeatherWS/Images/rain.gif</a>
w1:GetWeatherInformation	6	Showers	<a href="http://ws.cdnyne.com/WeatherWS/Images/showers.gif">http://ws.cdnyne.com/WeatherWS/Images/showers.gif</a>

### 5.15 pav. Gauti testiniai duomenys

## 6. Išvados

1. Atlikta literatūros šaltinių, WSDL serviso, interneto protokolų analizė parodė, kad būtų patogu automatiškai generuoti programinį kodą iš kelių WSDL servisų.
2. Atlikta esamų sprendimų analizė parodė jog nėra įrankio kuris leistų dirbti su keliais servisais vienu metu rezultate gaunant programinį kodą.
3. Įvertinus iškeltus darbo uždavinius ir tikslus nuspręsta sukurti sistemą, kuri leistų pasitelkiant grafinę vartotojo sąsaja sujungti WSDL servisus. Atliktus projektavimo veiksmus būtų sugeneruotas programinis kodas.
4. Projektuojant sistemą buvo sukurti algoritmai:
  - 4.1. WSDL analizės;
  - 4.2. Duomenų bazės sinchronizacijos;
  - 4.3. Programinio kodo generavimo;
  - 4.4. Serviso duomenų peržiūros;
5. Realizuotos sistemos prototipo išbandymas su įvairiomis tinklo paslaugomis parodė, jog tyrimo iškelti tikslai pasiekti. Sistema tinkamai atpažįsta SOAP protokolu veikiančius WSDL servisus. Suteikia intuityvią aplinką atlikti norimus prijungimo veiksmus su fizine arba virtualia duomenų baze.
6. Sukurtam sprendimui testuoti buvo pasirinkta įmonė UAB „Net Frequency“, dėl labai aukšto darbuotojų IT kvalifikacijos lygio, kurie praktiškai gali panaudoti visas sistemos funkcijas.
7. Sukurtas sprendimas pristatytas ir įdiegtas įmonėje UAB „Net Frequency“, kur įvertintas kaip perspektyvus produktas. Įmonės darbuotojai įrankį naudoja atitinkamų projektų WSDL servisų testavimui.
8. Remiantis sistemos eksperimentinio tyrimo vartotoju apklausos rezultatais galima teigti jog vartotojai yra patenkinti sukurta sistema. Vartotojai išvelgė sistemos trūkumų: galimybė dirbti tik su atvirais servisais arba tik SOAP tipo protokolu, bet taip pat ir labai naudingų privalumų: intuityvi vartotojo sąsaja, duomenų peržiūra, duomenų bazės sinchronizavimas, programinio kodo generavimas.



## 7. Literatūros sąrašas

1. *SOAP* Version 1.2 Part 1: Messaging Framework (Second Edition), prieiga per internetą : <http://www.w3.org/TR/SOAP12-part1/>
2. WebSphere DataPower XML Accelerator XA35, prieiga per internetą: <http://www-01.ibm.com/software/integration/datapower/xa35/highlights.html>
3. Collibra – put your business into context, prieiga per internetą: <http://www.collibra.com/solutions>
4. Web Services Essentials (O'Reilly XML) ISBN-10: 0596002246 ISBN-13: 978-8173663390
5. XMLSpy - XML Editor for Modeling, Editing, Transforming, & Debugging XML Technologies prieiga per internetą: <http://www.altova.com/xmlspy.html>
6. WebSphere software prieiga per internetą: <http://www-01.ibm.com/software/websphere/>
7. Programming Web Services with *SOAP* O'Reilly Media; 1 edition (December 15, 2001) ISBN-10: 0596000952
8. Stylus Studio: Powerful XML Integrated Development Environment (XML IDE) [žiūrėta 2010 m. lapkričio 15 d.] prieiga per internetą: [http://www.stylusstudio.com/xml\\_product\\_index.html](http://www.stylusstudio.com/xml_product_index.html)
9. REST vs. SOAP – The Right Webservice prieiga per internetą: <http://www.taranfx.com/rest-vs-soap-using-http-choosing-the-right-webservice-protocol>
10. PHP- Most Popular Open Source Programming Language 2011m prieiga per internetą: <http://www.rishabhsoft.com/blog/php-most-popular-web-development-language>
11. XMLRPC vs REST vs SOAP vs CIM vs RMI vs Message Bus vs ... Lots of RPC Options prieiga per internetą: <http://michaeldehaan.net/2008/07/17/xmlrpc-vs-rest-vs-soap-vs-all-your-rpc-options/>
12. Programming Web Services with XML-RPC (O'Reilly Internet Series) ISBN-10: 0596001193 ISBN-13: 978-0596001193
13. Introduction to PHP Data Services, prieiga per internetą : <http://wso2.org/library/articles/introduction-php-data-services>
14. CompuCal Mobile Calibration prieiga per internetą : <http://www.compucalsolutions.com/website/ProductMobileApplication.aspx>
15. WSMO data in the WSDL model prieiga per internetą : <http://www.wsmo.org/TR/d11/v0.2/20070421/>
16. XSL Transformations (XSLT) prieiga per internetą: <http://www.w3.org/TR/xslt>

17. Michael Kassoff, Daishi Kato, and Waqar Mohsin, *Stanford University* „Creating GUIs for Web Services“ prieiga per internetą: <http://logic.stanford.edu/~mkassoff/papers/wsgui.pdf>
18. SLT 2.0 and XPath 2.0 Programmer's Reference, 2011m. ISBN:978-1-4571-0497-8 ISBN 10:1-4571-0497-0.
19. Beginning XSLT and XPath: Transforming XML Documents and Data 2009m. ISBN-10: 0470477253 ISBN-13: 978-0470477250
20. Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More 2005m. ISBN-10: 0131488740 ISBN-13: 978-0131488748
21. Semantic annotations for WSDL and XML schema 2007m. World Wide Web Consortium, prieiga per internetą: <http://travesia.mcu.es/portalnb/jspui/handle/10421/2422>
22. David Martin. Semantic Web Services 2010m, Knowledge Media Institute, Open University.
23. UDDI - A Foundation for Web Services, prieiga per internetą: <http://www.idealliance.org/papers/xml2001/papers/html/03-02-03.html#d28e60352>
24. Andrea D'Ambrogio. A Model-driven WSDL Extension for Describing the QoS of Web Services 2010. University of Roma Tor Vergata
25. Alonso G., Casati F., Kuno H., Machiraju V. Web Services: concepts, architectures and applications 2004m. Berlin university.

## 8. Terminų ir santrumpų žodynas

1. **DB** (angl. Database) – duomenų bazė.
2. **XML** (angl. Extensible Markup Language) – W3C konsorciumo standartizuota duomenų struktūrų aprašomoji kalba.
3. **WSDL** (angl. Web service definition language) – Elektroninių paslaugų servisų aprašomoji kalba.
4. **SOAP** (angl. Simple Object Access Protocol) – Interneto protokolas, naudojamas prisijungti prie WSDL serviso.
5. **XSLT** (angl. *eXtensible StyleSheet Language Transformation*) – XML failų transformavimo kalba.
6. **GUI** (angl. *Graphic user interface*) – Grafinė vartotojo sąsaja.

## 9. Priedai

### 9.1. Funkciniai reikalavimai

9.1 lentelė. Vartotojo prisijungimas PA

<b>1. PA:</b> „Vartotojo prisijungimas“	
<b>Tikslas:</b> Atliekamas autorizuotas prisijungimas prie sistemos	
<b>Aprašymas:</b> Sistemos vartotojas prisijungia prie sistemos įveddamas savo prisijungimo vardą ir slaptažodį. Tik prisijungęs prie sistemos vartotojas pasiekia visą funkcionalumą.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti sukurtas Sistemoje (vartotojo įrašas duomenų bazėje). Vartotojus gali kurti tik <i>Administratorius</i> .
<b>Aktorius:</b>	Vartotojas, Administratorius
<b>Susiję panaudojimo atvejai:</b>	
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paleidęs sistemą patenka į prisijungimo langą, kur įveda savo prisijungimo duomenis	
2. Spaudžia mygtuką prisijungti	2.1.Sistema užfiksuoja prisijungimą įvykių žurnale (laiką, IP adresą, vartotojo identifikacijos numerį).
	2.2.Sistema patikrina ar toks vartotojas egzistuoja
	2.3.Sistema atidaro pagrindinį langą
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas nerastas duomenų bazėje	A1.1. Sistema praneša vartotojui jog toks vartotojas neegzistuoja
A2. Vartotojo slaptažodis neteisingas	A2.1. Sistema praneša vartotojui jog jo prisijungimo duomenys neteisingi

9.2 lentelė. Vartotojo atsijungimas PA

<b>2. PA:</b> „Vartotojo atsijungimas“	
<b>Tikslas:</b> Nutraukiama autorizuota prieiga prie sistemos	
<b>Aprašymas:</b> vartotojas atsijungia nuo sistemos.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti prisijungęs prie sistemos
<b>Aktorius:</b>	Vartotojas, Administratorius
<b>Susiję panaudojimo atvejai:</b>	1
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia „Atsijungti“	1.1.Sistema užfiksuoja atsijungimą įvykių

mygtuką	<p>žurnale (laiką, IP adresą, vartotojo identifikacijos numerį).</p> <p>1.2.Sistema pašalina duomenis iš naršyklės sausainėlių apie vartotoją.</p> <p>1.3.Sistema sunaikina sesijos duomenis.</p> <p>1.4.Vartotojui išvedamas pradinis prisijungimo langas</p>
---------	--

9.3 lentelė. Registruoti grupes PA

<b>3. PA: „Registruoti grupes“</b>	
<b>Tikslas:</b> Registruoti sistemoje vartotojų grupes, kurios supaprastina vartotojų valdymą	
<b>Aprašymas:</b> Administratorius turi teisę sukurti ir redaguoti vartotojų grupes, kurioms gali priskirti įvairias teises. Šitokiu būdu supaprastinimas vartotojų valdymas, kai galima priskirti keletą vartotojų vienai grupei ir jiems visiems galioja grupės teisės. Administratorius gali ištrinti grupę, tuo pačiu ištrinami ir visi vartotojai priklausantys šiai grupei.	
<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	1
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas atidaro meniu punktą „Vartotojų grupės“.	1.1.Sistema atidaro visų grupių sąrašą.
2. Vartotojas paspaudžia nuorodą „Pridėti grupę“	2.1.Sistema atidaro naujos grupės įvedimo formą.
3. Vartotojas įveda duomenis ir paspaudžia saugoti.	3.1.Sistema išsaugo įvestus duomenis į duomenų bazę. 3.2.Vartotojui atidaromas visų grupių sąrašas.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojo įvesti duomenys nėra korektiški.	A1.1. Sistema išveda pranešimą apie nekorektiškus duomenis.

9.4 lentelė. Registruoti vartotojus PA

<b>4. PA: „Registruoti vartotojus“</b>	
<b>Tikslas:</b> Registruoti sistemoje vartotojus	
<b>Aprašymas:</b> Administratorius turi teisę sukurti ir redaguoti vartotojus, kurie galės prisijungti prie duomenų bazės ir atlikti įvairius veiksmus.	

<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	1
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas atidaro meniu punkta “Vartotojai”.	
2. Vartotojas paspaudžia nuorodą „Pridėti vartotoją“	2.1.Sistema atidaro naujos vartotojo įvedimo formą.
3. Vartotojas įveda duomenis ir paspaudžia saugoti.	3.1.Sistema išsaugo įvestus duomenis į duomenų bazę. 3.2.Vartotojui atidaromas visų vartotojų sąrašas.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojo įvesti duomenys nėra korektiški.	A1.1. Sistema išveda pranešimą apie nekorektiškus duomenis.

9.5 lentelė. Nustatyti teises PA

<b>5. PA: „Nustatyti teises“</b>	
<b>Tikslas:</b> Nustatyti sistemoje vartotojų grupių teises	
<b>Aprašymas:</b> Administratorius turi teisę nustatyti teises vartotojų grupėms. Šitokios teisės galios visiems vartotojams esantiems toje grupėje.	
<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	1
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia meniu punktą “Vartotojų grupės”.	1.1.Vartotojui pateikiamas vartotojų grupių sąrašas.
2. Vartotojas paspaudžia mygtuką „Redaguoti grupę“	2.1.Sistema atidaro grupės redagavimo formą.
3. Vartotojas nurodo pasirinktas teises ir paspaudžia “Saugoti” mygtuką.	2.1.Sistema išsaugo įvestus duomenis į duomenų bazę. 3.2.Vartotojui atidaromas visų vartotojų grupių sąrašas.

9.6 lentelė. Administruoti projektus PA

<b>6. PA: „Administruoti projektus“</b>	
<b>Tikslas:</b> Atlikti projektuose leistinus veiksmus	

<b>Aprašymas:</b> Vartotojas gali atlikti tokius veiksmus kaip sukurti projektą, trinti projektą ar redaguoti jo informaciją.	
<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	1,2,7,8,9
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia meniu punktą „WSDL projektai“	1.1.Sistema vartotojui pateikia projektų sąrašą
2. Vartotojas paspaudžia trinti projektą	2.1.Sistema ištrina iš duomenų bazės duomenis apie projektą ir visus įrašus susijusius su projekto įrašu.
3. Vartotojas paspaudžia redaguoti projektą	3.1.Sistema pateikia projekto redagavimo formą.
4. Vartotojas suveda norimus parametrus	4.1.Sistema išsaugo duomenis duomenų bazėje.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	

9.7 lentelė. Sukurti projektą PA

<b>7. PA:</b> „Sukurti projektą“	
<b>Tikslas:</b> Sukurti naują WSDL projektą	
<b>Aprašymas:</b> Administratorius gali sukurti naują WSDL projektą. Nurodyti kokiam vartotojui priklausys projektas.	
<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	1
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia meniu punktą „WSDL projektai“	1.1.Sistema vartotojui pateikia projektų sąrašą
2. Vartotojas paspaudžia sukurti projektą	2.1.Sistema pateikia vartotojui naujo projekto įvedimo formą
3. Vartotojas įveda duomenis ir paspaudžia „Saugoti“	3.1.Sistema išsaugo duomenis duomenų bazėje. 3.2.Sistema atidaro visų projektų sąrašą.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojo įvesti duomenys nėra korektiški.	A1.1. Sistema išveda pranešimą apie nekorektiškus duomenis.

9.8 lentelė. Redaguoti projektą PA

<b>8. PA:</b> „Redaguoti projektą“	
<b>Tikslas:</b> Sukurti WSDL projekto informacija	
<b>Aprašymas:</b> Administratorius gali redaguoti sukurto projekto informaciją (pavadinimą, projekto vartotoją).	
<b>Prieš sąlygą:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	6,7,9
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia meniu punktą „WSDL projektai“	1.1.Sistema vartotojui pateikia projektų sąrašą
2. Vartotojas paspaudžia mygtuką „Redaguoti projektą“	2.1.Sistema pateikia vartotojui projekto redagavimo formą
3. Vartotojas įveda duomenis ir paspaudžia „Saugoti“	3.1.Sistema išsaugo duomenis duomenų bazėje. 3.2.Sistema atidaro visų grupių sąrašą.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojo įvesti duomenys nėra korektiški.	A1.1. Sistema išveda pranešimą apie nekorektiškus duomenis.

9.9 lentelė. Trinti projektą PA

<b>9. PA:</b> „Trinti projektą“	
<b>Tikslas:</b> Trinti WSDL projektą	
<b>Aprašymas:</b> Administratorius gali ištrinti projektą, taip ištrinama visa informacija susijusi su projektu.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti administratoriaus teises.
<b>Aktorius:</b>	Administratorius
<b>Susiję panaudojimo atvejai:</b>	6,7,8
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia meniu punktą „WSDL projektai“	1.1.Sistema vartotojui pateikia projektų sąrašą
2. Vartotojas paspaudžia mygtuką „Trinti projektą“	2.1.Sistema ištrina visus susijusius įrašus su projektu ir projektą. 2.2.Sistema atidaro visų projektų sąrašą.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	



9.10 lentelė. Peržiūrėti projekto informacija PA

<b>10. PA: „Peržiūrėti projekto informacija“</b>	
<b>Tikslas:</b> Peržiūrėti informacija apie projektą	
<b>Aprašymas:</b> Vartotojas gali peržiūrėti informacija apie savo projektus. Gali redaguoti servisų aprašus, duomenų bazę ir ryšius tarp servisų (visi jie detalizuoti žemiau parašytuose PA).	
<b>Prieš sąlyga:</b>	Vartotojas turi būti prisijungęs. Vartotojas turi turėti teisę peržiūrėti projektus Peržiūrimas projektas turi būti priskirtas vartotojui.
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia meniu punktą „WSDL projektai“	1.1.Sistema vartotojui pateikia projektų sąrašą
2. Vartotojas paspaudžia projekto peržiūros mygtuką.	2.1.Sistema pateikia projekto suvestinės vaizdą. Jame atvaizduojami pridėti WSDL servisi su savo funkcijų sąrašais, sukurtos duomenų bazės lentelės.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	

9.11 lentelė. Peržiūrėti WSDL servisų informaciją PA

<b>11. PA: „Peržiūrėti WSDL servisų informaciją“</b>	
<b>Tikslas:</b> Peržiūrėti informacija apie WSDL servisą	
<b>Aprašymas:</b> Vartotojas gali peržiūrėti informacija įvestus WSDL servisius. Servisius mato projekto redagavimo lange. Juos galima išskleisti, tada žemiau pateikiamas medžio pavidalų klasių ir funkcijų sąrašas, juos galima pervadinti arba atlikti apjungimo su duomenų baze galutinio kodo generavimo reikmėms.	
<b>Prieš sąlyga:</b>	Vartotojas turi būti atsidaręs projekto redagavimo langą.
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia serviso išskleidimo mygtuką	1.1.Sistema kreipiasi į serveriu su “POST” tipo užklausa gauti informacija apie pasirinktą projekte įvestą servisą. 1.2.Informacija gaunama ir atvaizduojama medžio pavidalu.
<b>Po sąlyga:</b>	

<b>Alternatyvūs scenarijai:</b>	
---------------------------------	--

9.12 lentelė. Įtraukti servisą PA

<b>12. PA: „Įtraukti servisą“</b>	
<b>Tikslas:</b> Projekte įtraukti naują WSDL servisą	
<b>Aprašymas:</b> Vartotojas gali redaguojamame projekte įtraukti naują WSDL servisą. Tai gali padaryti nurodydamas WSDL serviso aprašo internetinį adresą.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti atsidaręs projekto redagavimo langą.
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia mygtuką pridėti servisą.	1.1.Sistema pateikia iššokančiame ( <i>angl. popup</i> ) lange įvedimo formą.
2. Vartotojas įveda WSDL serviso aprašo internetinį adresą ir paspaudžia mygtuką pridėti.	2.1.Sistema išsaugo serviso įrašą servisų lentelėje su ryšiu į projektą. 2.2.Sistema inicijuoja “curl” biblioteką ir kreipiasi į pateiktą adresą gauti WSDL aprašą. 2.3.Gautas aprašas išsaugomas duomenų bazėje. 2.4.Aprašas užkraunamas į atmintį pasinaudojant “DomDocument” klase ir paverčiamas į objektą. 2.5.Sistema pasinaudodama XSLT transformacijos (“DomDocument” klasėje nurodant kaip stiliaus failą) atskiriamos klasės ir jų funkcijos su kintamaisiais. Gauta informacija išsaugoma duomenų bazėje.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojo įvesti duomenys nėra korektiški.	A1.1. Sistema išveda pranešimą apie nekorektiškus duomenis.

9.13 lentelė. Ištrinti servisą PA

<b>13. PA: „Ištrinti servisą“</b>	
<b>Tikslas:</b> Ištrinti projekte esantį WSDL servisą	
<b>Aprašymas:</b> Vartotojas gali ištrinti nereikalingą WSDL servisą, kuris buvo ankščiau pridėtas projekte.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti atsidaręs projekto

	redagavimo langą.
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia serviso ištrynimo mygtuką	1.1.Sistema išveda papildomo patvirtinimo formą išokančiame lange ( <i>angl. popup</i> ), kurioje pasiteirauja ar tikrai norite ištrinti servisą?
2. Vartotojas patvirtina ištrynimą	2.1.Sistema iš duomenų bazės ištrina WSDL serviso įrašą ir visa susijusią informaciją su juo.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas nepatvirtina ištrynimo	A1.1. Sistema uždaro formą, kitų veiksmų neatlieka.

9.14 lentelė. Peržiūrėti serviso informacija ir kintamuosius PA

<b>14. PA: „Peržiūrėti serviso informacija ir kintamuosius“</b>	
<b>Tikslas:</b> Peržiūrėti serviso apraše esančias klases, funkcijas ir kintamuosius.	
<b>Aprašymas:</b> Vartotojas gali peržiūrėti visas serviso apraše gautas funkcijas.	
<b>Prieš sąlygą:</b>	Vartotojas įveda serviso aprašo adresą ir paspaudžia „Pridėti“
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojui paspaudžia serviso ištrynimo mygtuką	1.1.Sistema išveda papildomo patvirtinimo formą iššokančiame lange ( <i>angl. popup</i> ), kurioje pasiteirauja ar tikrai norite ištrinti servisą?
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas paspaudžia atšaukti	A1.1. Sistema uždaro formą, kitų veiksmų neatlieka.

9.15 lentelė. Priskirti funkcijos kintamąjį į duomenų bazės lentelę PA

<b>15. PA: „Priskirti funkcijos kintamąjį į duomenų bazės lentelę“</b>	
<b>Tikslas:</b> Pasirinkti serviso funkcijos rezultatų įrašymui atitinkamą duomenų bazės lentelę	
<b>Aprašymas:</b> Vartotojas pasirinkęs funkciją ir nutempęs ją į atitinkamą duomenų bazės lentelę, iškviečia dialogą, kuriame pasirenka gražinamų kintamųjų atitinkamas duomenų bazės lenteles eilutes.	

<b>Prieš sąlygą:</b>	Vartotojas nutempia funkciją į duomenų bazės lentelę
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas nutempia funkciją į lentelę	1.1.Sistema išveda rezultatų priskyrimo formą
2. Vartotojas įveda (jeigu reikalinga) užklausos duomenis ir pasirenka lentelės eilutes atitinkančius rezultatus.	2.1.Sistema išsaugo pasirinkimus
<b>Po sąlyga:</b>	
	Sistema perkrauna lentelių rodymą, atnaujindama rezultatus.
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas nepatvirtina prijungimo	A1.1. Sistema uždaro formą, kitų veiksmų neatlieka.

9.16 lentelė. Valdyti duomenų bazę PA

<b>16. PA: „Valdyti duomenų bazę“</b>	
<b>Tikslas:</b> Projekte vartotojas gali sukurti virtualią duomenų bazę, lenteles ir jų laukus, taip pat juos redaguoti	
<b>Aprašymas:</b> Vartotojas pagal poreikius galu susikurti duomenų bazę, lenteles ir laukus. Veiksmai atliekami naudojantis intuityvia grafine vartotojo sąsaja.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti sukūręs projektą
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6, 17
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia sukurti lentelę	1.1.Sistema išveda lentelės įvedimo formą
2. Vartotojas įveda lentelės pavadinimą	2.1.Sistema išsaugo virtualią lentelę. Lentelėje įtraukia „wsdl_info” laukelį skirta saugoti prijungiamų servisų informacijai.
<b>Po sąlyga:</b>	
	Sistema perkrauna lentelių rodymą, atnaujindama rezultatus.
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas paspaudžia atšaukti	A1.1. Sistema uždaro formą, kitų veiksmų neatlieka.

<b>17. PA: „Sinchronizuoti duomenų bazę“</b>	
<b>Tikslas:</b> Projekte vartotojas gali prijungti virtualią sukurtą duomenų bazę prie fizinės MySQL duomenų bazės.	
<b>Aprašymas:</b> Jei vartotojas nusprendžia jog norės matyti rezultatus projekte, tai yra ne tik sugeneruoti programinį kodą, yra galimybė prijungti sukurtą duomenų bazę prie fizinės MySQL tipo duomenų bazės. Suvedus teisingus prisijungimus sistema sulygina virtualią duomenų bazę su fizine ir atitinkamai sukuria trūkstamus laukus abiejose duomenų bazėse.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti sukūręs projektą
<b>Aktorius:</b>	Vartotojas
<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia mygtuką prijungti duomenų bazę	1.1.Sistema išveda prisijungimo duomenų įvedimo formą
2. Vartotojas įveda duomenis	2.1.Sistema prisijungia prie fizinės duomenų bazės ir atsisiunčia jos schema. 2.2.Pagal schema virtualioje duomenų bazėje sukuriamos trūkstamos lentelės ir laukai. 2.3.Jei yra sukurtų virtualių lentelių ar laukų kurių nėra fizinėje duomenų bazėje jie atitinkamai sukuriami.
<b>Po sąlyga:</b>	
	Sistema perkrauna lentelių rodymą, atnaujindama rezultatus.
<b>Alternatyvūs scenarijai:</b>	
A1. Vartotojas paspaudžia atšaukti	A1.1. Sistema uždaro formą, kitų veiksmų neatlieka.
A2.Vartotojo prisijungimo duomenys nekorektiški	A2.1. Sistema išveda pranešimą jog duomenys neteisingi.

<b>18. PA: „Eksportuoti į programinį kodą“</b>	
<b>Tikslas:</b> Projekte įtrauktus servisus ir prijungtus duomenis vartotojas gali eksportuoti į PHP kalbos programinį kodą.	
<b>Aprašymas:</b> Vartotojas atlikęs visus norimus prijungimo veiksmus gali eksportuoti projektą į PHP kalbos programinį kodą. Vartotojui pateikiamas atsisiuntimui ZIP archyvas su viduje esančiais visais reikalingais failais, kuriuos paleidus HTTP serveryje, bus importuojami iš WSDL servisų visi duomenys pagal suprojektuota struktūrą.	
<b>Prieš sąlygą:</b>	Vartotojas turi būti sukūręs projektą
<b>Aktorius:</b>	Vartotojas

<b>Susiję panaudojimo atvejai:</b>	6
<b>Pagrindinis įvykių srautas:</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas paspaudžia mygtuką generuoti programinį kodą	1.1.Sistema pateikia ZIP archyvą su projekto failais atsisiuntimui.
<b>Po sąlyga:</b>	
<b>Alternatyvūs scenarijai:</b>	

## 9.2. Nefunkciniai reikalavimai

9.19 lentelė. Paprastas naudojimas

<b>Paprastas panaudojimas</b>	
<b>Numeris:</b>	1
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Vartotojas neturi sugaišti daug laiko mokytis dirbti su sistema.
<b>Tinkamumo kriterijus:</b>	80% atvejų pirmą kartą naudojantis sistema rezultatas turi būti sėkmingas.
<b>Užsakovo patenkinimas:</b>	5
<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.20 lentelė. Tvarkinga vartotojo sąsaja

<b>Tvarkinga vartotojo sąsaja</b>	
<b>Numeris:</b>	2
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistemos grafinė sąsaja turi būti tvarkinga.
<b>Tinkamumo kriterijus:</b>	Grafiniai elementai išdėstyti tvarkingai, laikomasi vienodų tarpų tarp elementų, ne daugiau, kaip 5 skirtingų šriftų dydžių.
<b>Užsakovo patenkinimas:</b>	4
<b>Užsakovo nepatenkinimas:</b>	3
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.21 lentelė. Patraukli sistema

Patraukli sistema	
<b>Numeris:</b>	3
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistemos grafinė sąsaja turi būti patrauklaus dizaino.
<b>Tinkamumo kriterijus:</b>	Spalvos tarpusavyje turi derintis. Turi būti naudojamos švelnios spalvos kurios nevargintų vartotojo akių.
<b>Užsakovo patenkinimas:</b>	4
<b>Užsakovo nepatenkinimas:</b>	3
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.22 lentelė. Sistemos efektyvumas

Sistemos efektyvumas	
<b>Numeris:</b>	4
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistemoje turi būti panaudotos AJAX technologijos.
<b>Tinkamumo kriterijus:</b>	Paprastiems veiksmai kuriuose nereikia perkrauti viso turinio turi būti panaudota AJAX technologija, kuri atnaujina atitinkamą bloką.
<b>Užsakovo patenkinimas:</b>	5
<b>Užsakovo nepatenkinimas:</b>	2
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.23 lentelė. Lengvas sistemos tobulinimas

Lengvas sistemos tobulinimas	
<b>Numeris:</b>	5
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistema turi būti sukurta remiantis kažkuriuo plačiai naudojamu PHP kalbos karkasu.
<b>Tinkamumo kriterijus:</b>	Sistemos bazinei dalei sukurti buvo panaudotas vienas iš populiarių PHP karkasų tokių kaip „Symfony“, „CodeIgniter“ ar „Zend Framework“

<b>Užsakovo patenkinimas:</b>	5
<b>Užsakovo nepatenkinimas:</b>	5
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.24 lentelė. Sistema turi būti stabili

<b>Sistema turi būti stabili</b>	
<b>Numeris:</b>	6
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistema nepriklausomai nuo įvedamų duomenų kiekio turi veikti stabiliai.
<b>Tinkamumo kriterijus:</b>	Sistemos puslapio perkrovimas turi užtrukti ne ilgiau kaip 2s.
<b>Užsakovo patenkinimas:</b>	3
<b>Užsakovo nepatenkinimas:</b>	2
<b>Priklausomybės:</b>	Interneto greitis kompiuteryje turi būti didesnis nei 10 Mbit/s
<b>Konfliktai:</b>	nėra

9.25 lentelė. Sistema turi būti lengvai testuojama

<b>Sistema turi būti lengvai testuojama</b>	
<b>Numeris:</b>	7
<b>Panaudojimo atvejai:</b>	1-18
<b>Pagrindimas:</b>	Sistema turi turėti galimybę kaupti istoriją apie atliktus veiksmus programinio kodo failuose.
<b>Tinkamumo kriterijus:</b>	Sistema atskirame aplanke kaupia atliktus klasių ir funkcijų veiksmus failuose pagal datą.
<b>Užsakovo patenkinimas:</b>	4
<b>Užsakovo nepatenkinimas:</b>	1
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

9.26 lentelė. Sistema turi būti perkeliama į kitą serverį

<b>Sistema turi būti perkeliama į kitą serverį</b>	
<b>Numeris:</b>	8
<b>Panaudojimo atvejai:</b>	1-18



<b>Pagrindimas:</b>	Sistema turi būti lengvai įdiegiama naujame serveryje.
<b>Tinkamumo kriterijus:</b>	Sistema yra realizuotas sistemos įdiegimas su grafine vartotojo sąsaja.
<b>Užsakovo patenkinimas:</b>	3
<b>Užsakovo nepatenkinimas:</b>	1
<b>Priklausomybės:</b>	nėra
<b>Konfliktai:</b>	nėra

### 9.3. Diegimo aktas



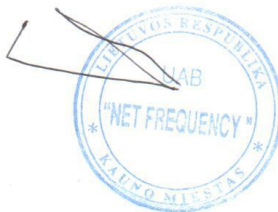
**UAB Net Frequency**

#### PATVIRTINIMAS DĖL GENERAVIMO SISTEMOS ĮDIEGIMO

2012 m. balandžio 20 d.  
Kaunas

Šiuo raštu patvirtinu, kad 2012 m. balandžio 20 d. UAB „Net Frequency“ buvo įdiegta įmonės darbuotojo Simono Šerlinsko sukurta WSDL programinio kodo generavimo sistema.

Direktorius



Darius Lebedzinskas

---

Įmonės kodas 135867375  
PVM kodas LT358673716

M. Daukšos g. 34a-5,  
LT-44283 Kaunas

Įregistruota 2002m. kovo 4d.  
Rejestro tvarkytojas Ūkio ministerija

## 9.4. Darbuotojų apklausa

Vartotojams buvo pateikta apklausos forma naudojantis internetine „Google Docs“ teikiama forma.

### WSDL programinio kodo generavimo sistemos apklausa

#### Kaip vertinate serviso pridėjimą į projektą?

Ar visi parametrai reikalingi užklausos formoje? Ar sistema tinkamai pateikia pridedamo serviso funkcijas?

#### Kaip vertinate serviso prijungimą prie duomenų bazės?

Ar prijungimo forma teisingai pateikia užklausos parametrus? Ar teisingi rezultatų parametrai? Ar pakankamai aišku kaip jais naudotis?

#### Sistemos intuityvumas

Ar sistema paprasta ir patogus naudotis?

- Labai paprasta
- Paprasta
- Sudėtinga
- Labai sudėtinga

#### Kaip vertinate rezultatų peržiūrą?

- Gerai
- Labai naudinga
- Nereikalinga
- Neveikia

#### Gal turite pastabų?

## 9.5. Darbuotojų apklausos rezultatai

Kaip vertinate serviso pridėjimą į projektą?	Sistemos intuityvumas	Kaip vertinate serviso prijungimą prie duomenų bazės?	Kaip vertinate rezultatų peržiūrą?	Gal turite pastabų?
Klasės pavadinimas galėtų būti sugeneruojamas iš bendros informacijos. Sistema tinkamai pateikia visas serviso funkcijas.	Labai paprasta	Labai patiko "drag and drop" principas. Tikrai labai patogiu naudotis, viskas aiškiai matoma ir suprantama. Suklydus galima atstatyti į pradinę būseną.	Gerai	Sistema galėtų palaikyti servisu su autentifikacija.
Gerai. Galėtų būti protokolo pasirinkimas.	Paprasta	Labai patogus parametru pasirinkimas.	Labai naudinga	Sistema galėtų palaikyti daugiau protokolų, ne tik SOAP
Lėtai veikia.	Paprasta	Negali pridėti vienos funkcijos į dvi lenteles iškart, tai reikia atlikti atskirai. Puikus sprendimas prijungti funkcijas tempiant.	Gerai	Sistema gali dirbti tik su atvirais servisiais. Būtų didelis plusas jei dirbtų ir su privačiais.