

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

Aivaras Juzvikas

**ELEKTRONINIO VERSLO CENTRO TINKLO
PASLAUGŲ SISTEMA**

Magistro darbas

**Vadovė
doc. Lina Nemuraitė**

KAUNAS, 2006

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

**ELEKTRONINIO VERSLO CENTRO TINKLO
PASLAUGŲ SISTEMA**

Magistro darbas

Vadovė

doc. L. Nemuraitė

Recenzentas

doc. V. Pilkauskas

Atliko

IFM 0/4 gr. stud.

A. Juzvikas

KAUNAS, 2006

WEB SERVICES FOR E-COMMERCE HUB SUMMARY

There are more and more incompatible systems with the growing number of services provided by computers. The user has to choose between expensive huge systems, offered by large companies like Microsoft or Oracle and losing competition. While these huge systems have a lot of possibilities, small and middle sized companies can only use a small portion of options offered there. With the increasing pressure from competitors, this forces users to choose other technologies like Web Services, in order to survive rivalry.

WSDL (*Web Services Description Language*) was invented to describe Web Services in the most simple way there is.

Web Service is a collection of protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.

In this work the methods are suggested for creation of stand-alone E-Commerce Hub and combining other systems into it. By the help of Web Services, users of Internet Stores, and other systems based on product and service data, are able to share their information with others through the E-Commerce Hub. It is easily automated and requires no maintenance.

TURINYS

ĮVADAS.....	7
1. TINKLO PASLAUGŲ ANALIZĖ	8
1.1. Verslo sistemos efektyvumas	8
1.2. Tinklo paslaugų problematika.....	8
1.3. Tinklo paslaugų architektūra.....	9
1.3.1. Terminai	9
1.3.2. Tinklo paslaugų istorija.....	9
1.3.3. Tinklo paslaugų pagrindas	10
1.3.4. Tinklo paslaugos architektūra	12
1.4. Tinklo paslaugų aprašymo kalba WSDL	13
1.4.1. WSDL sintaksė.....	14
1.5. Interneto paslaugų kūrimas	16
1.6. Elektroninio verslo vartotojai.....	16
1.6.1. Tinklo paslaugų vartotojai.....	16
1.6.2. Elektroninio verslo vartotojai.....	17
1.6.3. Vartotojų tikslai ir problemos	17
1.7. Pasirinktų technologijų apžvalga	17
1.7.1. Duomenų bazių valdymo sistemų (<i>DBVS</i>) analizė	18
1.8. Sistemų naudojančių tinklo paslaugas analizė	19
1.9. Elektroninio verslo sistemos apibrėžimas	21
1.10. Darbo tikslas ir siejami privalumai	22
1.11. Kompiuterizuojamos sistemos funkcijos	22
1.12. Reikalavimai duomenims.....	23
1.13. Nefunkciniai reikalavimai ir apribojimai	23
1.13.1. Reikalavimai standartams	23
1.13.2. Reikalavimai veikimui	24
1.13.3. Reikalavimai sąveikai ir suderinamumui su kitomis sistemomis.....	24
1.14. Rizikos faktorių analizė.....	24
1.15. Rezultato kokybės kriterijai	24
1.16. Analizės išvados.....	25
2. ELEKTRONINĖS VERSLO APLINKOS PROJEKTAVIMAS.....	26
2.1. Elektroninio verslo tinklo paslaugų sistemos modelis	26
2.1.1. Elektroninio verslo aplinkos aktoriai	26
2.2. Reikalavimų specifikacija	26
2.3. Nefunkciniai reikalavimai sistemai.....	31
2.3.1. Paslaugų centro dalykinės srities modelis.....	32
2.4. Sistemos architektūra – statinės struktūros modelis.....	35
2.4.1. Loginė visos sistemos architektūra	35
2.5. Duomenų bazės schema	36
2.6. Realizacijos modelis.....	40
3. ELEKTRONINĖS VERSLO APLINKOS REALIZAVIMAS.....	41
3.1. Elektroninio verslo aplinkos sąsaja	41
3.1.1. Administratoriaus sąsaja	41
3.1.2. Pirkėjo sąsaja.....	42
3.1.3. Tinklo paslaugos sąsaja.....	43
3.2. Sukurtos sistemos kokybė	46
3.2.1. Sistemos kokybės užtikrinimas	46
3.2.2. Elektroninio verslo aplinkos kokybės įvertinimas	47

3.3.	Kokybės charakteristikos	47
3.3.1.	Realaus laiko informacijos apsikeitimas	47
3.1.1.	Integravimo paprastumas su kitomis sistemomis.....	47
3.2.1.	Funkcinių galimybių išplečiamumas.....	48
3.3.1.	Instaliavimas ir valdymas	48
3.4.1.	Daugiaplatformiškumas	48
3.5.1.	Kokybės įvertinimo apibendrinimas	48
IŠVADOS.....		50
LITERATŪRA.....		51
TERMINAI IR ŽODYNAS		53

Lentelių sąrašas

1.1 lentelė Duomenų bazių valdymo sistemų kokybinės charakteristikos.....	19
3.1. lentelė Realaus laiko informacijos gavimo tinklo paslaugų pagalba įvertinimas	47
3.2. lentelė Integravimo paprastumas su kitomis sistemomis įvertinimas	47
3.3. lentelė Funkcinių galimybių išplečiamumo įvertinimas	48
3.4. lentelė Instaliavimo ir valdymo paprastumo įvertinimas	48
3.5. lentelė Daugiaplatformiškumo įvertinimas	48
3.6. lentelė Kokybės įvertinimo apibendrinimas.....	49

Paveikslėlių sąrašas

1.1 pav. Tinklo paslauga programoje	9
1.2 pav. Tinklo paslaugų technologijų stekas [15].....	10
1.3 pav. Tinklo paslaugos architektūra.....	12
1.4 pav. Alexa tinklo informacijos paslauga	20
1.5 pav. Sistemos kontekstinė diagrama	22
1.6 pav. Sistemos administratoriaus ir vartotojo panaudojimo atvejai	23
2.1 pav. Pirkėjo informacijos paieškos panaudos atvejis	26
2.2 pav. Tinklo paslaugos vartotojo panaudos atvejis.....	27
2.3 pav. Sistemos informacijos administravimo panaudos atvejis.....	27
2.4 pav. Sistemos administravimo panaudos atvejis	28
2.5 pav. Dalykinės srities diagrama	32
2.6 pav. Klasių diagrama.....	33
2.7 pav. Paieškos sekų diagrama: informacijos ieškoma centre	34
2.8 pav. Paieškos sekų diagrama: informacijos ieškoma kitoje sistemoje.....	34
2.9 pav. Sistemos architektūra.....	35
2.10 pav. Duomenų bazės schema	36
2.11 pav. Sistemos diegimo diagrama.....	40
3.1 pav. Administratoriaus sąsaja: prekių sąrašas	41
3.2 pav. Administratoriaus sąsaja: prekės redagavimas	42
3.3 pav. Administratoriaus sąsaja: kategorijų medis.....	42
3.4 pav. Pirkėjo sąsaja: prekės informacija	43

ĮVADAS

Plečiantis kompiuterių bei interneto teikiamoms galimybėms atsiranda vis daugiau ir daugiau įvairių tarpusavyje nesuderinamų sistemų. Taigi tenka rinktis didžiųjų firmų siūlomas sistemas, kurios nėra pigios. Tai galingos, daug galimybių turinčios sistemos, tačiau smulkioms ir vidutinėms įmonėms šios sistemos yra per brangios ir per daug sudėtingos, nes tokių įmonių veikloje gali būti panaudotos tik kai kurios jų funkcijos. Didėjančios konkurencijos sąlygomis tai lemia kitų sprendimų pasirinkimą, norint maksimizuoti pelningumą ir išlikti konkurencinėje kovoje.

Siekiant išspręsti šias problemas ir suderinti įvairias sistemas, buvo sukurta nauja programavimo kalba – WSDL (*Web Services Description Language*) tinklo paslaugoms (*Web services*) aprašyti.

Tinklo paslaugų dizainas suteikia galimybe naudotis skirtingomis programomis (skirtingose sistemose) ir komunikuoti internetu nepriklausomai nuo to, kokios programavimo kalbos buvo pasirinktos jas realizuojant. Tinklo paslauga - tai savakrankiška aplikacija, kuri nustatoma universalaus resursų identifikatoriaus URI, kuris remiasi XML sintakse. Tinklo paslauga komunikuoja su kitomis programinės įrangos aplikacijomis naudodama tinklo komunikavimo protokolus ir siuntinėja XML pranešimus. Tinklo paslaugų struktūra suteikia galimybę susikoncentruoti į pačias verslo aplikacijas ir jų apjungimą, o ne į jų sąsajos suderinamumą.

Tinklo paslaugų aprašymui naudojamoje programavimo kalboje WSDL metodai, duomenų struktūros, protokolu asignavimai bei žinutės aprašomos aukštesniame lygyje ir nuo kažkokios vienos sistemos nepriklauso, todėl jų naudojimas įgalina ne tik sujungti tarpusavyje nesuderinamas sistemas, bet ir leidžia žymiai sumažinti verslo sistemų realizavimo kaštus greitai besikeičiančioje rinkoje.

1. TINKLO PASLAUGŲ ANALIZĖ

1.1. Verslo sistemos efektyvumas

Darbo tikslas – realizuoti internetinio verslo centro aplinką, kurioje tinklo paslaugų principu grindžiami duomenų derinimo ir perdavimo sprendimai. Tai įgalintų lengvai automatizuoti duomenų keitimasi ir padarytų tradicinį komunikavimą tarp elektroninio verslo objektų paprastesniu.

Norint tai įgyvendinti bus sprenždiami tokie uždaviniai:

- išanalizuoti tinklo paslaugų architektūrą bei ja paremtas sistemas;
- išsiaiškinti tinklo paslaugų bei jomis paremtų sistemų trūkumus bei esamas savybes;
- sukurti komunikavimo tarp skirtingų programų skirtingose sistemose sprendimą ir išdėstyti reikalavimus realizuojamam elektroninio verslo centrui;
- apibrėžti savo kuriamos verslo aplinkos tikslus ir siekiamus kriterijus;
- atlikti elektroninio verslo centro kokybinį įvertinimą.

1.2. Tinklo paslaugų problematika

Interneto paslaugų srityje dažnai tenka naudoti kelias sistemas, kurios sukelia tam tikrų problemų. Aktualiausia iš jų yra ta, kad dažniausiai sistemos yra nesuderinamos, todėl tenka dubliuoti informaciją, operacijas informacijai apdoroti ar net pačias sistemas. Visos šios problemos didina kaštus bei daro darbą su sistema, jos tobulinimą, priežiūrą ar prieinamumą sudėtingesniais. Taip pat gan svarbu yra tai, kad kuriant sistemas, operacijos atliekamos su duomenimis kartojasi ir sistemų kūrėjai yra priversti jas perrašyti, kaskart kurdami ta patį produktą naujoje platformoje. Tai tampa ypač problematiška, kuomet sistemai reikalingi papildymai, pakeitimai ar klaidų ištaisymai. Žymiai patogiau būtų turėti galimybę ištaisyti klaidas ar sukurti naujas galimybes sistemai vienoje vietoje – aukštesniame lygmenyje. Tokiu atveju visos naudojamos sistemos turėtų būti įtakotos šio vienoje vietoje esančio pakeitimo, kuris savo ruožtu būtų smulkesnis.

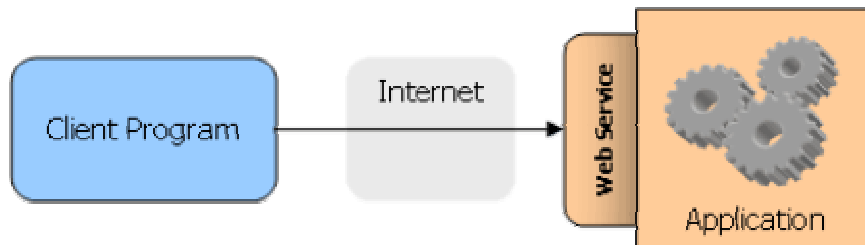
Šių problemų sprendimo būdas – tinklo paslaugos. Naudojant tinklo paslaugas sistemos veiksmai ir struktūra aprašomas aukštesniame lygyje (pasitelkiant WSDL), kuris yra pasiekiamas iš visu sistemų, taip sujungiant nesuderinamų sistemų duomenis ir operacijas į vieną bendrą aplinką.

Pasirinktas tyrimo objektas – tinklo paslaugos elektroninio verslo srityje, taikant WSDL.

1.3. Tinklo paslaugų architektūra

1.3.1. Terminai

Tinklo paslauga yra nepriklausoma aplikacija, pasiekama pasitelkiant tinklo (ar interneto) komunikavimo protokolus; ji naudoja XML sintaksę [14] žinutėms apsikeisti [15].



1.1 pav. Tinklo paslauga programoje

Tinklo paslauga - tai programinė aplikacija, kuri yra patalpinama URI ir ten lengvai randama. Tinklo paslaugų elementai aprašomi pasitelkus XML sintaksę. jos su kitomis programomis komunikuoja be tarpininkų [16].

XML struktūra buvo išrasta duomenų aprašymo standartizavimui. Ji užtikrina programų bendradarbiavimą tarpusavyje, sujungiant duomenis ir aplikacijų bendradarbiavimą vienu standartu. Dėl to, įvairios sistemos laisvai tarpusavyje komunikuoja.

Interneto paslaugų teikiamas privalumas yra tas, kad jos gali pasinaudoti kitų taikomųjų uždavinių siūlomomis paslaugomis nepriklausomai nuo to, kokioje platformoje jos veikia ar kokia specifinė techninė įranga naudojama jų susisiekimui.

1.3.2. Tinklo paslaugų istorija

Tinklo paslaugos – tai egzistuojančių tinklo protokolų evoliucijos pasekmė. 1994-ais metais Microsoft, NeXT, Apple ir IBM pradėjo kurti paskirstytuosius objektus (*Distributed Objects*). Visos firmos savo technologijas pavadino skirtingai:

- Microsoft – Component Object Model;
- NeXT – Portable Distributed Objects;
- Apple – OpenDoc;
- IBM – System Object Model.

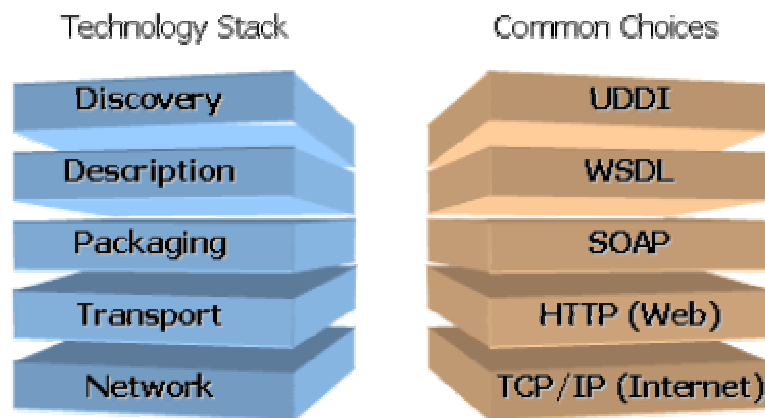
Vėliau jos įkūrė OMG ir įtvirtino standartą CORBA. Pastangos buvo dedamos tam, kad sukūrus technologiją, kuri įgalintų taikomasias aplikacijas naudoti bendrą komunikavimo standartą ne tik duomenims perduoti, bet ir vykdyti viena kitos funkcijas. Dabar naujai sukurta architektūra žinoma tinklo paslaugų vardu.

1.3.3. Tinklo paslaugų pagrindas

Tinklo paslaugos remiasi šiomis pamatinėmis technologijomis [15]:

- Atradimas (*Discovery*);
- Aprašymas (*Description*);
- Pakavimas (*Packaging*);
- Transportavimas (*Transport*);
- Tinklas (*Network*).

Aukščiau esančių technologijų pamatas yra giliau steke veikiančios technologijos. T.y. tinklo paslaugos transportavimas įgalinamas tinklo ir t.t.



1.2 pav. Tinklo paslaugų technologijų stekas [15]

- *Tinklas* - Kompiuterius reikia sujungti internetu arba kitokiu tinklu, paremtu tokiu pačiu protokolu kaip ir internetas, kad būtų įmanoma komunikuoti siunčiant užklausas ir gaunant atsakymus. Dažniausiai naudojamas protokolas [15], kuriuo remiasi toks tinklas yra TCP/IP.
- *Transportavimas* - Pranešimus ir atsakymus reikia kažkoku būdu transportuoti, kad interneto paslauga galėtų juos pasiekti. Čia yra galimybė pasitelkti vieną iš daugelio esamų informacijos perdavimo protokolų:
 - *HTTP* – tradicinis pasirinkimas;
 - *FTP* – patogi alternatyva dideliems pranešimams;
 - *SMTP* – esant tam tikrai serverio konfigūracijai itin patogu;
 - Ir kiti..
- *Pakavimas* - Pakavimas naudojamas interneto paslaugų komunikavime, t.y. užklausoms ir gražinamiems atsiliepimams supakuoti į transportavimui patogų formatą. To reikia tam, kad nebūtų tinklo paslaugos priklausomybės nuo instaliavimui pasirinktos platformos ar

realizavimui naudojamos technologijos. Yra du pagrindiniai pakavimo standartai tai SOAP ir REST.

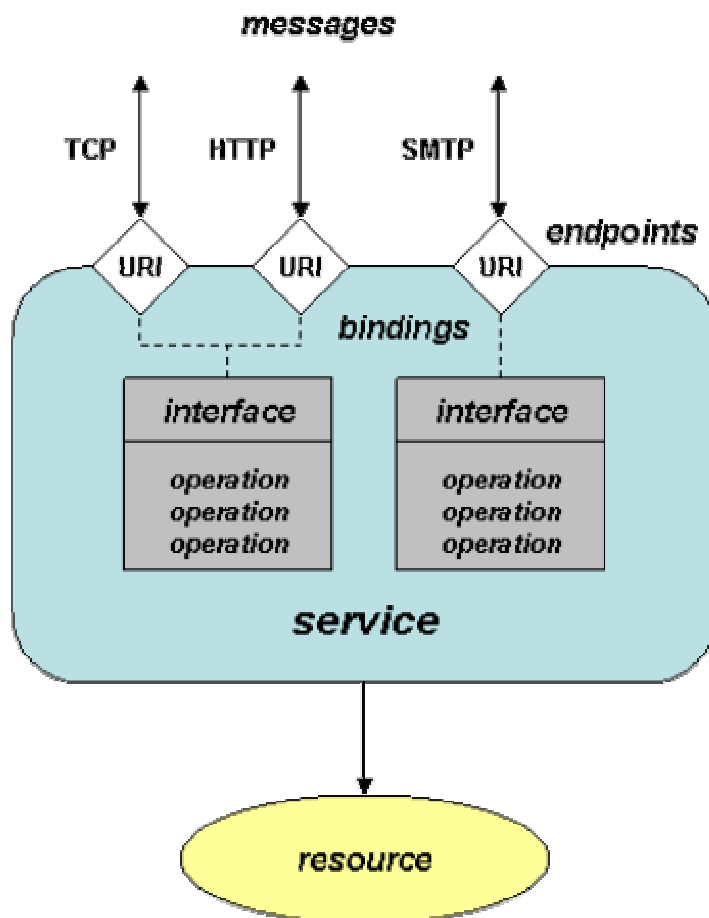
- SOAP [15] – XML pagrindu veikianti technologija [18], ji pakuoja informaciją ir užklausas tinklo paslaugų iškvietimų principu;
 - REST – HTTP POST, GET funkcijomis pagrįsta pakavimo technologija.
- *Aprašymas* - Dažniausiai tinklo paslaugos aprašymui naudojama WSDL kalba. Ja aprašomi tinklo paslaugos teikiami metodai ir sąsaja. Tik turint tinklo paslaugos aprašą, aplikacija turi galimybę pasinaudoti šia paslauga. Kadangi WSDL yra sukurta XML kalbos pagrindu, pranešimai turi būti aprašomi XML kalbos naudojama struktūra.
 - *Atradimas* – vyksta UDDI, globaliame tinklo paslaugų registre [15]. Dažniausiai UDDI registrą aptarnauja paskirstytas tarnybinių stočių tinas, dėl to jį lengva plėsti ar taisyti nesutrikdant jo darbo. Atradimo sluoksnis leidžia nesudėtingai atrasti tinklo paslaugas naudojimui.

Šių penkių skirtingų technologijų stiprus bendradarbiavimas ir yra tinklo paslaugų pagrindas.

Norint sujungti WSDL su PHP yra du labiausiai paplitę protokolų standartai [11]:

- **REST** – Priešingai nei SOAP nereikalauja papildomų įrankių instaliavimo duomenų siuntimui ir gavimui. Taip pat vienas iš REST privalumų yra tai, kad be plačiai paplitusių XML ir HTTP protokolų nereikia jokių papildomų technologijų. Komunikavimui naudojamos paprastos HTTP užklaustos: GET, POST, PUT ir DELETE, o duomenys aprašomi pasitelkus XML struktūrą. HTTP privalumas yra tas, kad tai yra pakankamai senas ir labai paplitęs protokolas, kuris yra prieinamas beveik visom, su internetu dirbančiom technologijom. Bet dėl savo senumo, HTTP protokolas nepateikia visų funkcionalumo savybių interneto paslaugų teikimo programavimui. Todėl, būtinybė rinktis papildomas technologijas, niekais paverčia visus aukščiau išvardintus REST privalumus;
- **SOAP** – Žinomiausia ir labiausiai paplitęs protokolas. Šis protokolas perduoda duomenis, juos enkapsuliuojes. Nors SOAP naudoja savo vidinius komunikavimo metodus, kurie yra pakankamai sudėtingi, tačiau jie suteikia šiam protokolui ypatingai didelį lankstumą realizavime. Taigi dėl plačiai paplitusio naudojimo tai yra priimtinausias sprendimas.

1.3.4. Tinklo paslaugos architektūra



1.3 pav. Tinklo paslaugos architektūra

Nors pati paslaugų architektūra yra neobjektinė, bet ji remiasi objektinio projektavimo principu ir tinklo paslaugų stilium veikiantys programinio kodo komponentai gali būti panaudojami pakartotinai (*Reused*). Paslaugų architektūra yra tarpusavyje susijusių objektų evoliucijos rezultatas. Rezultate paslaugų architektūroje atsirado funkcionalumo sąsaja (*Adapter*), kuri reikalinga tam, kad standartizuotu komunikavimą tarp skirtingose operacinėse sistemose naudojamu skirtingų programinės įrangos priemonių. Priklausomybės nuo platformos nebuvimas suteikia teigiamų ir neigiamų savybių tinklo paslaugoms. Nors priklausomybės trūkumas leidžia tinklo paslaugas vienodai aprašyti visose operacinėse sistemose, taip suteikiant joms galimybę komunikuoti nors ir visiškai skirtingose platformose, tačiau yra ir keli šios architektūros trūkumai:

- Norint pilnai aprašyti pranešimus sistemoje reikia papildomai naudoti išplečiamų schemų technologiją. Taip yra dėl to, kad turint visišką nepriklausomybę nuo

sistemos, pranešimuose elgesio aprašymas yra labai ribotas. Taigi šių schemų naudojimas apriboja pranešimų laisvę, jų struktūrą;

- Nors sąsajos visose sistemose yra vienodos ir iš visur pasiekiamos (pagrindinis tinklo paslaugų privalumas), tačiau ši nepriklausomybė pačias sąsajas padaro labai ribotas, jomis aprašoma tik bendra struktūra. Detalesnis aprašymas yra neįmanomas dėl didelio skirtumo tarp įvairių operacinių sistemų ir programavimo technologijų.

1.4. Tinklo paslaugų aprašymo kalba WSDL

Tinklo paslaugų technologiją siūloma rintis tom įmonėms, kurios siekia sumažinti laiko ir pinigų sąnaudas į interneto portalų, svetainių ir programinės įrangos kūrimą [8]. WSDL yra minėtų paslaugų aprašymo kalba, kuri tinklo paslaugas aprašo nuo platformos nepriklausoma, universalia XML sintakse. WSDL skirta aprašyti keturis esminius duomenų segmentus:

- Surišamoji pasirinkto transportavimo protokolo informacija;
- Visų žinutėse aprašomų pranešimų duomenų tipų informacija;
- Norimos paslaugos išieškojimui skirta adreso informacija;
- Visus tinklo paslaugos teikiamus metodus aprašančios sąsajos informacija.

Žinutės ir metodai aprašomi nepriklausomai nuo interneto ar kito naudojamo tinklo protokolų ar programinės įrangos realizacijoje pasirinktos žinučių struktūros, t.y. abstrakčiu stiliumi. Tik po to priskiriamas konkretus protokolas duomenų perdavimui, kuris yra patogus tinklo paslaugos gavėjui [2]. Taigi WSDL tai XML sintaksės pagrindu veikianti programavimo kalba, kuri nusako kaip tinklo paslaugos (metodų ar tiesiog duomenų pavidalo informacija) yra aprašoma ir kaip ji perduodama informacijos gavėjui iš informacijos teikėjo. Patogu yra tai, kad informacija yra aprašoma nepriklausomai nuo informacijos gavėjo ir tiekėjo naudojamų standartų ir komunikavimui naudojamų protokolų.

Kadangi šioje programavimo kalboje žinučių aprašymas nepriklauso nuo realaus jų eiliškumo, jas galima atskirti nuo duomenų struktūros nusakymo ar tinklo naudojamų asignavimų. WSDL apibrėžtos tinklo paslaugos tai paprasčiausiai jungtys, nurodančios informacijos teikėją ir gavėją. Tai patogiu, nes aprašus (*Definitions*) galima panaudoti pakartotinai. Šis pakartotinis ryšys (*Binding*) nusako informacijos gavėjo ar teikėjo naudojama duomenų struktūra ar komunikavimo protokolą.

Taigi naudodamas WSDL, klientas gali susirasti ne tik reikiamą interneto paslaugą bet ir išsikviesti bet kurį jos siūlomą metodą. Ši procesą galima lengvai automatizuoti naudojant WSDL įrankius, suteikiant galimybę aplikacijoms nesudėtingai integruoti naujas tinklo paslaugas beveik ar visai nekoreguojant jų programų išeities tekstų. WSDL yra interneto paslaugų kertinis akmuo,

tai suteikia įprastinę kalbą paslaugų aprašymui ir platformą, automatiniam šių paslaugų integravimui bet kokiose sistemose, naudojančiose įvairiausias, kitaip nesuderinamas, technologijas.

1.4.1. WSDL sintaksė

WSDL sintaksė – tai paprasčiausiai apibrėžimų aibė susidedanti iš šešių skirtingų elementų naudojamų apibrėžimui nusakyti:

- Prievadas (*Port*) apibūdina vieną iš dviejų: informacijos gavėją arba teikėją, nepriklausomai nuo kito ir priskiria jam unikalų adresą. Patys prievadai taip pat turi unikalius vardus, kurie leidžia identifikuoti juos programiniame kode. Taigi prievadas nusako ne pačią priskyrimo informaciją, o tik informaciją apie vieną priskyrimo adresą.
- Prievado tipas (*Port type*) turi unikalų vardą, kuris leidžia jį atskirti nuo kitų prievado tipų programiniame kode. Prievado tipas apjungia abstrakčias žinutes ir abstrakčias operacijas. WSDL kalboje yra šie informacijos perdavimo abstraktūs atvejai, kuriuos palaiko informacijos priėmėjas: žinutė-pranešimas, kreipimasis-atsakymas, užklausa-atsakymas ir vienakryptis. Šie abstraktūs informacijos perdavimo atvejai yra operacijos WSDL sintaksėje. Kreipimasis-atsakymas ir užklausa-atsakymas abstrakčius atvejus atstoja dvi vienakryptės žinutės. Vartotojas gali rinktis tokią žinučių kombinaciją, kuri jam yra patogesnė.
- Žinutės (*Messages*) turi unikalius vardus, kurie leidžia jas identifikuoti programiniame kode. Žinutę gali sudaryti viena ar daugiau loginės skiltys. Kiekviena iš skilčių taip pat turi unikalų identifikatorių-vardą, kuris leidžia atskirti skiltis žinutės viduje. Visos skiltys, žinučių tipo identifikatoriaus pagalba, yra sujungtos su tipais.
- Paslaugos (*Services*) turi unikalius vardus (tam WSDL faile), kurie leidžia identifikuoti jas besikreipiant programiniame kode. Paslaugų funkcija yra sujungti prievadus į atskirą aibę. Dažniausiai į vieną aibę yra sujungiami prievadai, turintys kažką bendro tarpusavyje. Paslaugos prievadai turi šias savybes:
 - Alternatyvūs prievadai. Tai tokie prievadai, kurie priklauso vienai paslaugai, naudoja bendrą prievado tipą. Bet jos gali turėti skirtingus adresus ar priskyrimus. Programuotojas gali rinktis iš alternatyvių prievadų savo nuožiūra, kadangi rezultatas bus tas pats. Taigi alternatyvūs prievadai suteikia pasirinkimo laisvę, kas yra paprasčiau programuotojui.

- Tarpusavyje prievadai nekomunikuoja.
- Programuotojas naudodamas paslaugos prievadą, gali nurodyti ar toliau veiksmų eigoje jis šią paslaugą utilizuos. Tai atliekama naudojant prievado tipus. Sukūrus ryšius tarp prievado tipo metodų ir pačių metodų pasireiškia šios savybės privalumas.
- Priskyrimas (*Binding*) turi unikalų identifikatorių-vardą, kuris yra reikalingas norint išrinti reikiama priskyrimą iš visų WSDL faile aprašomų priskyrimų. Priskyrimas nusako tik vieną protokolą (nenusako adreso informacijos), t.y. žinutės struktūrą ir prievado tipo aprašomų žinučių ir operacijų smulkmenas. Tuo tarpu prievado tipas gali vieną ar daugiau priskyrimų. Informacija pačiam priskyrimui ar jo operacijai, t.y. įvedimo, išvedimo ar klaidos pranešimo buferiai aprašomi priskyrimo elementų pagalba. Operacijos yra atrenkamos pagal jų atributų vardus, nes pačių operacijų vardai nėra unikalūs-identifikatoriai. Priskyrimo, jo informaciją skirtą operacijai nusako operacijos elementas, kurio vardas yra toks pats kaip ir priskyrimo prievado tipe.
- Tipas (*Type*). XSD yra naudojama kaip tipų sistema, tam, kad užtikrinti komunikavimo lankstumą ir priklausomybės nuo operacinės sistemos ar naudojamų technologijų nebuvimą. XSD galima naudoti dviem būdais: elementų ir atributų forma. Naudojant elementų formą, laukui galinčiam įgyti tam tikrą reikšmę ar kintamajam aprašyti naudojamas abstraktus `xsd:anyType` tipas. Šią formą siūloma rinktis, kuomet priskyrimas yra vienas ir nėra plačios tipų sistemos ar kuomet paslaugoje yra daug priskyrimų vienai žinutei. Taigi tipas duoda aprašus duomenų tipam, kurie yra būtini keičiantis žinutėmis. WSDL taip pat suteikia galimybę pasitelkus išplėstinius elementus papildyti tipų sistemą. Norint nustatyti pasirinktą tipų aprašymų sistemą ir kad būtų išskirta vieta tipo aprašui, XML faile įterpiamas išplėstinis elementas (po tipo elementų).

WSDL sintaksės elementų reikšmės būtų tokios:

- Prievadas – interneto adresas ir priskyrimas, nurodantis informacijos gavėją arba informacijos teikėją;
- Prievado tipas – tai grupė abstrakčių metodų, kurių kiekvienas užklausia išvedimo, įvedimo arba klaidos žinutę;
- Žinutė – tai loginių dalių aibė, kurios kiekviena loginė dalis veikia nepriklausomai nuo kitos ir dirba su tam tikra operacija ar duomenimis;

- Paslauga – tai informacijos gavėjų ir informacijos teikėjų, tarpininkaujančių tarpusavyje, aibė;
- Priskyrimas – tai vieno prievado tipo nusakomų žinučių ir metodų duomenų struktūros ir komunikavimo protokolo nustatymai;
- Tipas – žinutėms nusakyti reikalingi duomenų tipai;

1.5. Interneto paslaugų kūrimas

Interneto paslaugų kūrimo funkciją galima išskaidyti į tokius procesus:

- Analizę;
- Projektavimą;
- Realizavimą;
- Testavimą;
- Palaikymą.

Analizės etape pirmiausiai išnagrinėjama rinka ir joje randama niša tam tikros srities sistemai. Jei analizės metu išaiškėja, kad kurti sistemą būtų naudinga ir organizacijai, ir klientams, pradedamas projektavimo etapas. Šiame etape nusprendžiame ar verta sistemoje diegti tinklo paslaugas. Tinkamiausi atvejai taikyti šią technologiją yra kuomet tarp kelių firmų naudojama viena duomenų bazė ar kelių firmų duomenų bazės apjungiamos pasirinktame projekte. Jei projektas yra ne trumpalaikis ir planuojama išplėsti galimybes, pvz. prekių asortimentą elektroninėje parduotuvėje, nedvejodami renkamės aukštesniame lygmenyje aprašytas sistemos struktūras bei operacijas. Toliau projektuojamos vartotojo ir administratoriaus funkcijos, sukuriamas dizainas. Realizacijos etape įgyvendinami projektavimo etape užsibrėžti tikslai ir funkcijos. Sukūrus sistemą į ją vedami duomenys ir lygiagrečiai vykdomas testavimas. Vėliau vyksta duomenų atnaujinimas ir atliekami reikalingi sistemos tobulinimai.

1.6. Elektroninio verslo vartotojai

1.6.1. Tinklo paslaugų vartotojai

- Sistemų kūrėjai, kuriems tenka realizuoti panašius duomenis, besikartojančiomis operacijomis apdorojančią programinę įrangą;
- Kelių skirtingų sistemų vartotojai, kurie nori lengvai išgauti, sujungti, bei atlikti kitas operacijas su duomenimis;

1.6.2. Elektroninio verslo vartotojai

- Klientai, ieškantys alternatyvių būdų rasti norimą paslaugą ar prekę;
- Vartotojai norintys išgauti informaciją pasitelkiant minimalius laiko bei piniginius kaštus;

1.6.3. Vartotojų tikslai ir problemos

Bendros vartotojų problemos – skirtingų sistemų nesuderinamumas, nepasiekiamumas, egzistuojančių panašių sistemų dideli tobulinimo, palaikymo kaštai.

Visų jų siekiamas tikslas yra mažesniais kaštais įgalinti klientus prisijungti prie jų sistemos įvairiausiai protokolais. Lengvai ir pigiai keisti tokios universalios sistemos funkcijas, operacijas bei duomenų struktūras, ją tobulinti.

Pagrindines problemas ir tikslus galima būtų sugrupuoti ir išreikšti šiais klausimais:

- Kokios verslo paslaugos yra siūlomos pasirinktoje interneto svetainėje?
- Kaip galima naudotis sistemos siūlomomis verslo paslaugomis?
- Kokios informacijos reikia sistemoje siūlomai verslo paslaugai iš vartotojo, kuomet jis naudojami viena paslauga?
- Kaip vartotojas pateiks reikalingą informaciją?
- Kokių formatu ir forma verslo paslauga atsakys vartotojui?

Tinklo paslaugų aprašymo kalba WSDL [3] pateikia mechanizma visų šių darbų atlikimui.

1.7. Pasirinktų technologijų apžvalga

Projektavimo darbams atlikti buvo naudojama viena iš lyderiaujančių priemonių šioje srityje – MagicDraw UML. MagicDraw UML patogu tuom, kad jame yra galimybė atlikti tiesioginę bei atvirkštinę programinio teksto inžineriją XML, WSDL ir kitoms programavimo kalboms. Taigi naudojant MagicDraw UML yra galimybė iš UML diagramos generuoti programinį tekstą, jį keisti, redaguoti ir iš jo vėl generuoti UML diagramas. Kadangi šis paketas naudoja naujausią UML 2.0 standartą, šiuos veiksmus atliekant informacija nėra prarandama.

Programinei verslo centro realizacijai pasirinkta PHP ir MySQL dėl kelių priežasčių:

- Palaikoma daugelyje operacinių sistemų (Unix, Linux, Windows);
- Lengva keisti nustatymus duomenų bazėje, skaičiavimo algoritmuose – PHP;
- Lengvai keičiama ir pritaikoma interneto bei tinklo paslaugų atžvilgiu, kadangi PHP buvo kuriamas specialiai pasiekti per internetą.

Vienas pagrindinių kriterijų renkant šias technologijas buvo tai, kad jos yra nemokamos ir nuolat atnaujinamos.

Alternatyvos yra mokamos technologijos, pvz. Microsoft priemonės: ASP su MSOL duomenų bazių valdymo sistema arba kitos technologijos. Tačiau, su šiomis technologijomis darbo patirties nėra, be to, kai kurios jų priklauso nuo operacinės sistemos. Svarbu kuo platesnis ir universalesnis taikymas, ir paprastas priėjimas, tinklo paslaugų bei HTTP palaikymas.

1.7.1. Duomenų bazių valdymo sistemų (DBVS) analizė

Duomenų bazių valdymo sistemų (DBVS) yra daug skirtingų ir įvairių, kurių beveik visų pagrindinių programinės įrangos firmų. Pagrindiniai skirtumai tarp duomenų bazių valdymo sistemų yra jų kaina ir kokybinės bei kiekybinės charakteristikos.

Kiekybinės charakteristikos:

- leistinas apdorojamų duomenų kiekis;
- maksimalus lentelių skaičius duomenų bazėje;
- duomenų bazės lentelės apimtis.

Vienas iš pagrindinių sistemų kiekybinių kriterijų yra sistemų apimtis. Pagal savo apimtį duomenų bazių valdymo sistemas galima suskirstyti į dideles, vidutines ir mažas.

Kokybinės DBVS charakteristikos:

- sudėtingumo laipsnis;
- taikymo sritis;
- operacinės sistemos;
- darbo patogumas.

Renkantis duomenų bazių valdymo sistema pagrindiniai kriterijai buvo šie:

- mokama ar nemokama;
- palaikomų OS skaičius;
- suderinamumas su interneto protokolais ir PHP.

Pagrindinių duomenų bazės valdymo sistemų kokybinės charakteristikos pateiktos 1.1 lentelėje [12] [13].

1.1 lentelė Duomenų bazių valdymo sistemų kokybinės charakteristikos

DBVS	Lentelių kiekis	Lentelės dydis	Lentelės plotis	Laukų kiekis	Lauko plotis	Įsigijimo kaina	WEB suderinamumas	OS palaikymas
Oracle	---	---	priklauso nuo laukų pločių	254	2 GB	didelė	Pilnas	ribotas
Sybase	$2 \cdot 10^9$	priklauso nuo OA dydžio	priklauso nuo OA dydžio	250	1962 baitai	--	yra	ribotas
Informix	$477 \cdot 10^6$	64 TB	32767 baitai	2767	32767 baitai	---	yra	ribotas
DB2	priklauso nuo OA dydžio	64 GB	priklauso nuo OA dydžio	255	4005 baitai	---	yra	ribotas
SQL Server	$2 \cdot 10^9$	2 TB	2048 baitai	250	255 baitai	Didelė	Pilnas	Windows
FoxPro	---	2 GB	65500 baitų	255	254 baitai	maža	nėra	Windows
Access	---	1 GB	---	255	255 baitai	maža	nėra	Windows
MySQL	naudojama apie $6 \cdot 10^4$	8TB, Priklauso nuo OS Iki 64TB	priklauso nuo laukų pločių	3398	2 GB	nemokama	Pilnas	Labai plačiai

Kadangi MySQL dar nėra įterptinių funkcijų (*Stored procedures*), ji yra priskiriama prie vidutinių duomenų bazių valdymo sistemų.

1.8. Sistemų naudojančių tinklo paslaugas analizė

SOAP ir REST - dviem labiausiai paplitusiais standartais galima išgauti duomenis iš šių interneto paslaugų teikėjų pasitelkus tinklo paslaugas:

www.amazon.com – knygų bei kitų siūlomų produktų duomenų bazė;

Siūlomų paslaugų principai:

Elektroninės komercijos paslauga (*Amazon E-Commerce Service*) – vartotojai turi galimybę gauti Amazon svetainėje siūlomų prekių sąrašus, prekių kiekius, jų kainas. Taip pat nuorodas į prekes Amazon svetainėje ir gauti užmokestį už iš jų puslapių atėjusius apsiperkančius klientus.

www.alexa.com - tinklo informacijos paslauga (*Alexa Web Information Service*) – vartotojai gali gauti informaciją apie pasirinktos svetainės lankomumą:

- Puslapio lankomumo eilė, t.y. palyginimas su kitais puslapiais;
- Puslapio lankytojų kiekiai paskutinį pusmetį;
- Puslapio subdomenų lankomumas, t.y. informacija apie tai, kur lankytojai daugiausiai lankosi šiame puslapyje;
- Vidutinis lankytojo peržiūrėtų puslapių kiekis pasirinktoje svetainėje;

- Panašių svetainių sąrašas;
- Svetainės kontaktinė informacija;
- Vartotojų atsiliepimai apie šią svetainę.

Delfi [Go to site now](#)
delfi.lt



Naujienų serveris.

Traffic Rank for delfi.lt: 3,963



1.4 pav. Alexa tinklo informacijos paslauga

www.google.com – Paieškos rezultatai gaunami iš daugiau nei 8 milijardų suindeksuotų puslapių.

Siūlomos kelios paslaugos, kurių veikimo principai yra tokie [7]:

Paieškos užklausos (*search requests*) – paieška pagal raktažodį ar sakinį. Google paslaugai paduodamas žodis ar žodžių seka ir gaunamas rezultatas – puslapių aibė susijusi su pasirinktu raktažodžiu.

Atminties užklausos (*cache requests*) – paieška pagal interneto adresą (URL). Google paslaugai paduodamas internetinis svetainės adresas ir gaunamas rezultatas yra pasirinktos svetainės turinys toks, koks jis buvo šio puslapio paskutinio indeksavimo metu.

Rašybos užklausa (*spelling request*) – rašybos klaidų paieška užduotoje žodžių sekoje. Google paslaugai duodamas sakiny susidedantis iš ne daugiau nei dešimties žodžių ir gaunamas rezultatas – siūlomas sakiny, kuriame rašybos klaidos ištaisytos.

Atgalinių nuorodų užklausa (*backlinks request*) – paieška pagal interneto adresą. Gaunamas rezultatas – puslapių, kurie turi nuorodas į užduotą adresą, aibė.

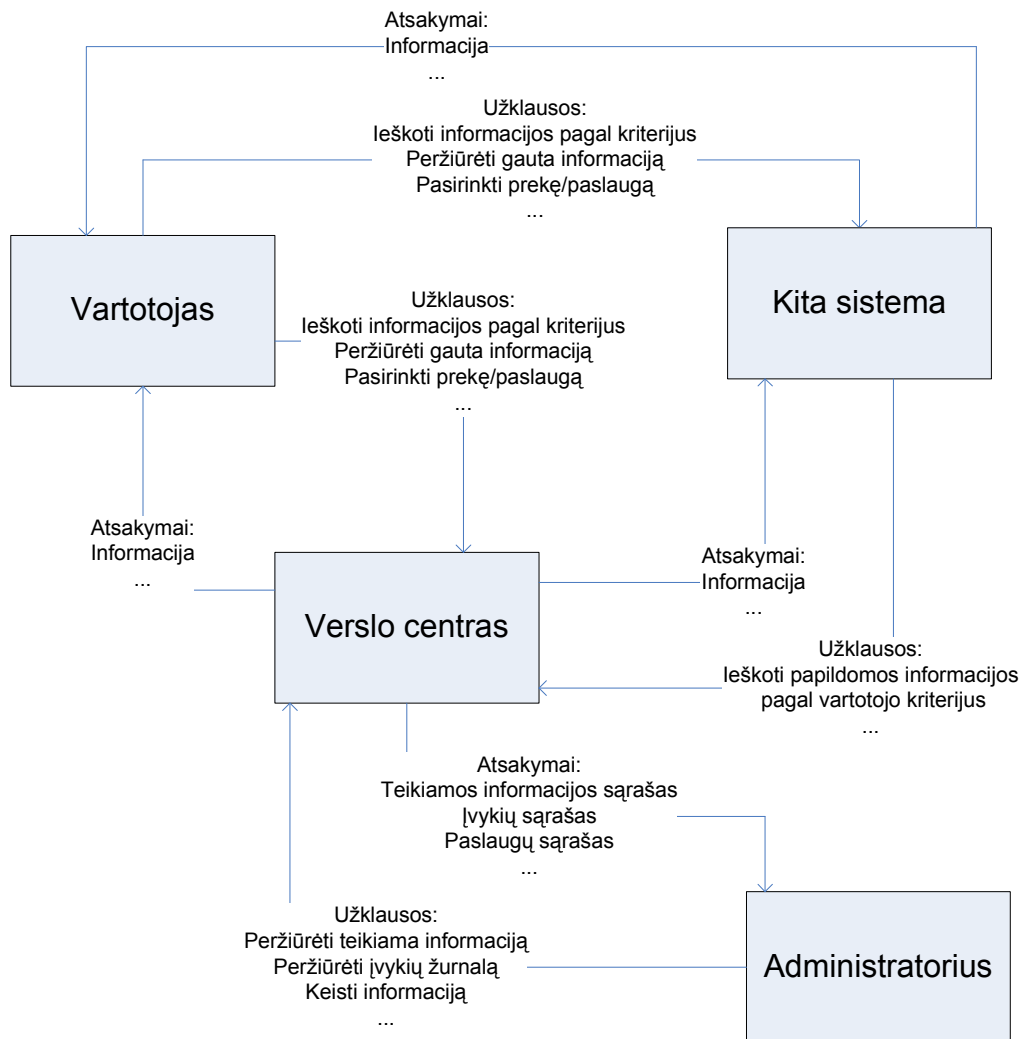
Puslapių sklaidos užklausa (*saturation request*) – paieška pagal interneto adresą. Gaunamas rezultatas – visų vidinių puslapių aibė užduotoje svetainėje.

Failų tipų paieška (*File Type request*) – paieška pagal raktažodžius norimo tipo failuose. Google paslaugai paduodama aibė failų plaplėtimų, kuriuose atliekama paieška pagal pasirinktą raktažodį ar sakinį. Rezultatai gražinami nuorodų į failus formatu.

1.9. Elektroninio verslo sistemos apibrėžimas

1.5 paveiksle pavaizduotas verslo centras – tai interneto prekybos sistema, kur prekių ar paslaugų teikėjai gali patalpinti savo prekes ar paslaugas, o vartotojai rasti prekes ar paslaugas pagal savo poreikius.

Vartotojams atliekant paiešką teikėjų tinklapiuose, teikėjai gali integruoti verslo centro siūlomą paiešką, kuri realiu laiku gražins papildomos informacijos sąrašą ieškotojams iš verslo centro, jeigu ten jos yra pagal užduotus kriterijus.



1.5 pav. Sistemos kontekstinė diagrama

1.10. Darbo tikslas ir siekiami privalumai

Šio darbo tikslas – išanalizuoti tinklo paslaugų architektūros principus, sukurti jų taikymo metodiką elektroninės komercijos aplinkoje bei pritaikyti realioje sistemoje. Sukurta sistema turėtų būti lanksti naudojimo atžvilgiu – vartotojas turi neįstoti persijungimo prie skirtingų interneto paslaugų. Sistema turi būti lengvai modifikuojama, leisti įdiegti papildomas paslaugas pasitelkiant minimalias pastangas ar leisti pašalinti esamas nesutrikdant bendro verslo sistemos funkcionalumo.

Darbo privalumai – pritaikius šią metodiką ženkliai sumažės skirtingų sistemų integravimo kaštai bei laiko sąnaudos.

1.11. Kompiuterizuojamos sistemos funkcijos

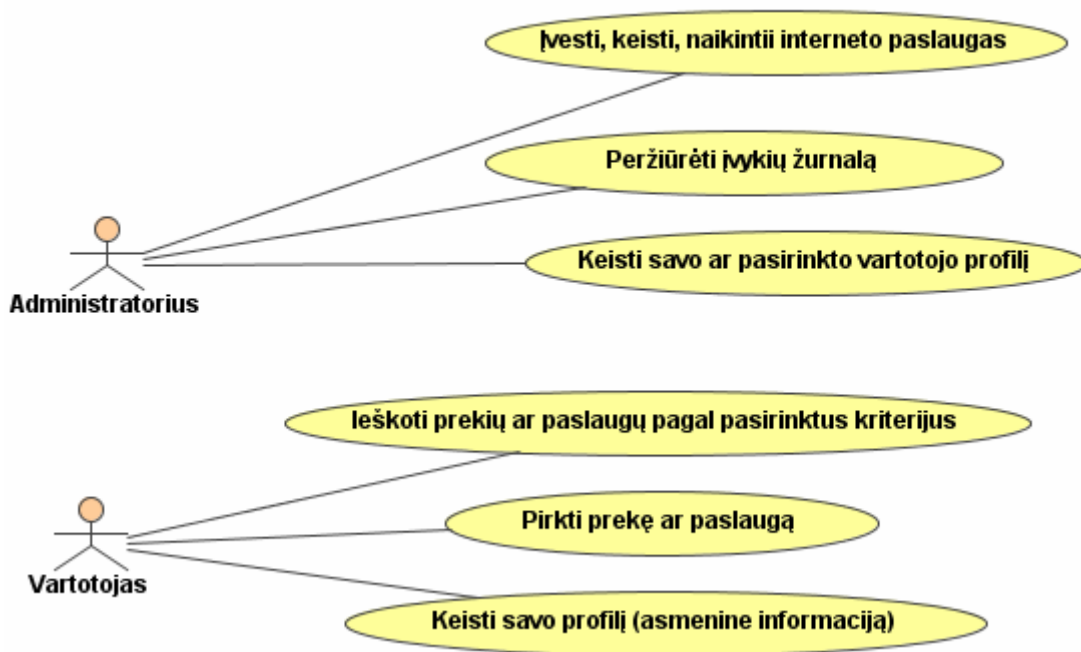
Sistemoje galima išskirti 2 tipų aktorius:

- vartotojas;

- sistemos administratorius;

Administratorius – už sistemoje esantį turinį, t.y. teikiamas paslaugas ar siūlomų prekių sąrašą bei sistemos vartotojus atsakingas asmuo.

Vartotojas – sistemos klientas, ieškantis paslaugų ar prekių.



1.6 pav. Sistemos administratoriaus ir vartotojo panaudojimo atvejai

1.12. Reikalavimai duomenims

Verslo centre esantys loginiai duomenys yra realizuoti, kaip klasės PHP kalboje, saugomos duomenų bazės valdymo sistemoje (pasirinkta MySQL) ir atliekančios verslo centre naudojamų esybių vaidmenį. PHP klases paprasčiausia susieti su reliacine duomenų baze. Tokiu būdu PHP klasės-konteineriai fizinius duomenis (reliacinės duomenų bazės lentelių įrašus) sinchronizuoja su loginiais duomenimis, naudojamais programų sistemoje.

Kad nekiltų problemų suderinant sistemų, tinklo paslaugos turi būti aprašytos WSDL kalba.

1.13. Nefunkciniai reikalavimai ir apribojimai

1.13.1. Reikalavimai standartams

1. Naršyklei pateikiamas turinys turi atitikti W3C rekomendacijas XHTML [9] bei CSS [10] struktūroms.
2. Interneto paslaugos turi būti aprašomos WSDL [2] kalba.
3. WSDL sujungimui su PHP platforma turi būti taikomas SOAP protokolas.

1.13.2. Reikalavimai veikimui

Naujos paslaugos bei prekės turi būti lengvai įdiegiamos, nepakenkiant sistemos veikimui. Taip pat senų paslaugų ir prekių šalinimas neturėtų trukdyti sistemos darbui.

1.13.3. Reikalavimai sąveikai ir suderinamumui su kitomis sistemomis

Kadangi vidutinis ir smulkus verslas neturi finansinių išteklių brangioms didžiųjų programinės įrangos tiekėjų elektroninio verslo valdymo sistemoms, kuriama sistema turi būti itin lanksti ir lengvai integruojama su kitomis elektroninio verslo valdymo sistemoms. Taip pat turi naudoti atvirojo programinio kodo (*Open Source*) technologijas, kurios yra nuolat kuriamos ir atnaujinamos. Sistemų sujungimui naudojamos interneto paslaugos turi būti pakankamai universalios, kad būtų galima tarpusavyje sujungti net ir visiškai kitokios struktūros sistemas ar duomenų bazes.

1.14. Rizikos faktorių analizė

WSDL – pakankamai nauja programavimo kalba, kuria didžiosios firmos (Google, Amazon) teikia paslaugas dar tik bandomojoje stadijoje. Todėl naudojant šių firmų tinklo paslaugas yra rizika, kad jos bus panaikintos, pakeistas veikimo principas, ar pakeista licenzijavimo tvarka, t.y. įvestas naudojimo mokestis.

Norint pilnai išnaudoti šios technologijos galimybes, ją reikia taikyti ypač dideliuose projektuose, apimančiuose kelias sritis, tokio dydžio sistemą sukurti užimtų daug laiko, taigi gali būti, kad bus nesuspėta parodyti visų tinklo paslaugų technologijos privalumų.

1.15. Rezultato kokybės kriterijai

Svarbiausi sistemos kokybės kriterijai:

- Sistemos veiksmingumas:
 - Ar sukurta sistema veikia tinkamai?
 - Ar išsprendžia išsikeltas problemas?
 - Ar atlieka nustatytus funkcinius reikalavimus?
- Universalumas:
 - Ar sistemą galima visiškai ar bent iš dalies pritaikyti įvairių portalų kūrimui?
 - Ar sistema visiškai atstoja alternatyvius produktus?
- Išplečiamumas:
 - Ar sistema lengvai plečiama modifikuojant jau sukurtas interneto paslaugas?

- Ar lengvai prijungiamos papildomos interneto paslaugos?
- Ar sistema leidžia teikti interneto paslaugas?
- Patikimumas:
 - Ar panaikinti arba sumažinti rizikos faktoriai?
 - Koks klaidų skaičius?
- Naudojimosi paprastumas ir lengvumas:
 - Ar sistemą sudėtinga pritaikyti specifinėms vartotojo reikmėms?

1.16. Analizės išvados

- Skersai išilgai išanalizuotos tinklo palaugos: nuo jų apibrėžimo, istorijos iki struktūros ir siūlomų galimybių. Pateikta technologijų apžvalga, kurios yra tinklo paslaugų pagrindas: atradimas, pakavimas, aprašymas, tinklas ir transportavimas;
- Apžvelgtos technologijos, kurios reikalingos norint įgyvendinti tinklo paslaugų veikimą: WSDL, SOAP, REST, HTTP ir TCP/IP;
- Išsiaiškinta, kad pagrindinė tinklo paslaugų aprašymo kalba yra WSDL. Ji aprašo tinklo paslaugas ir jomis susieja komunikavimą tarp informacijos teikėjo ir informacijos gavėjo. Pagrindiniai WSDL programavimo kalbos elementai: tipas, žinutė, prievadas, priskyrimas, paslauga;
- Atlikta kelių populiariausių sistemų teikiančių tinklo paslaugas apžvalga, tai Amazon, Alexa ir Google;
- Apibrėžta kuriamo verslo centro aplinka, nustatyti pagrindiniai funkciniai ir nefunkciniai reikalavimai sistemai, bei jos duomenims;
- Apžvelgti pagrindiniai rizikos faktoriai naudojant tinklo paslaugas. Pagrindinis faktorius – tai tinklo paslaugų naujumas;
- Sistemai projektuoti bus naudojama MagicDraw UML programinė įranga, nes ši įranga turi UML įskiepius WSDL modeliams projektuoti ir galimybę generuoti WSDL programinio kodo karkasą. Realizavimui pasirinkta PHP programavimo platforma, dėl patogumo dirbant su internetinėmis technologijomis (tam ji ir skirta);
- Atlikta duomenų bazių valdymo sistemų (DBVS) analizė ir pasirinkta MySQL programinė įranga dėl plataus palaikymo PHP programavimo kalboje, multiplatformiškumo ir patrauklios kainos (nemokama).

2. ELEKTRONINĖS VERSLO APLINKOS PROJEKTAVIMAS

2.1. Elektroninio verslo tinklo paslaugų sistemos modelis

2.1.1. Elektroninio verslo aplinkos aktoriai

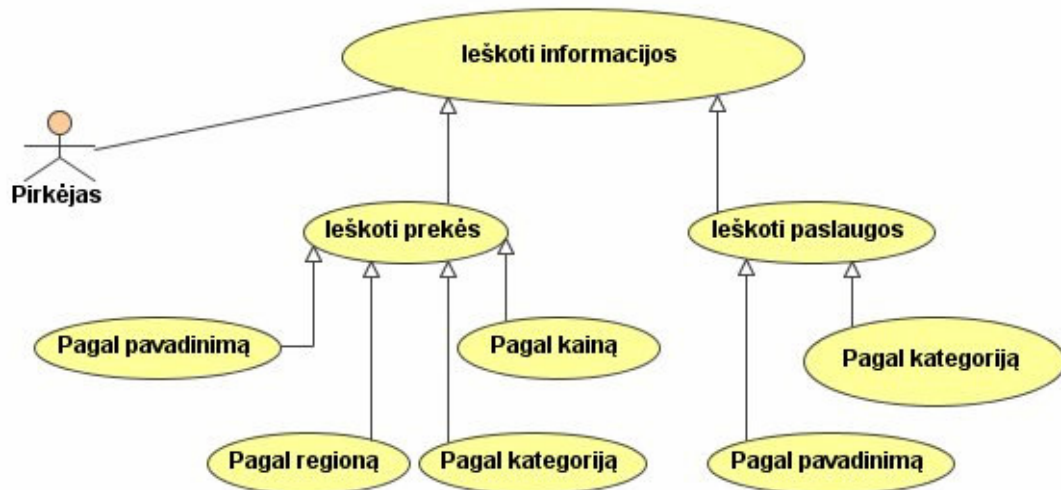
Administratorius: autorizuotas asmuo, turintis jam suteiktą prisijungimo vardą, slaptažodį ir administratoriaus teises sistemoje. gali keisti vartotojų informaciją, prekių, paslaugų duomenis ir sistemos konfigūraciją.

Vartotojas: komunikuoja su sistema tinklo paslaugų pagalba. Tai autorizuotas asmuo, turintis jam suteiktą prisijungimo vardą, slaptažodį. Gali patalpinti bei ištrinti informaciją sistemoje apie savo prekes ar paslaugas, atlikti paiešką.

Pirkėjas: eilinis svetainės lankytojas. Gali peržiūrėti paslaugų ir prekių sarašą, atlikti informacijos paiešką.

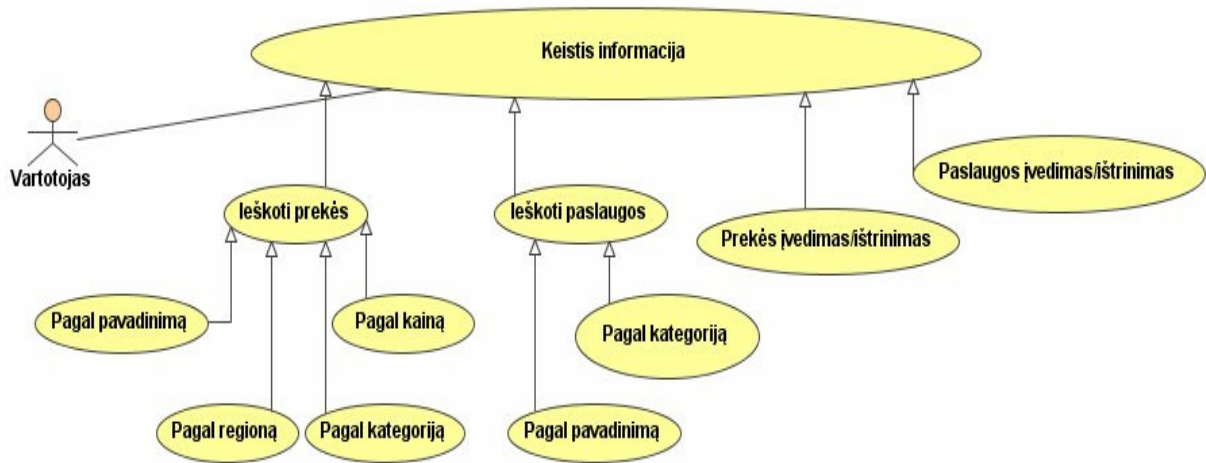
2.2. Reikalavimų specifikacija

Pirkėjas elektroninio verslo sistemoje yra prekių ir paslaugų ieškotojas. Jis gali atlikti paiešką pagal raktažodį (pavadinimą), kainą ar kategoriją, kuriai priklauso jo ieškoma prekė ar paslauga. Pirkėjo panaudos atvejai pateikti 2.1 paveiksle.



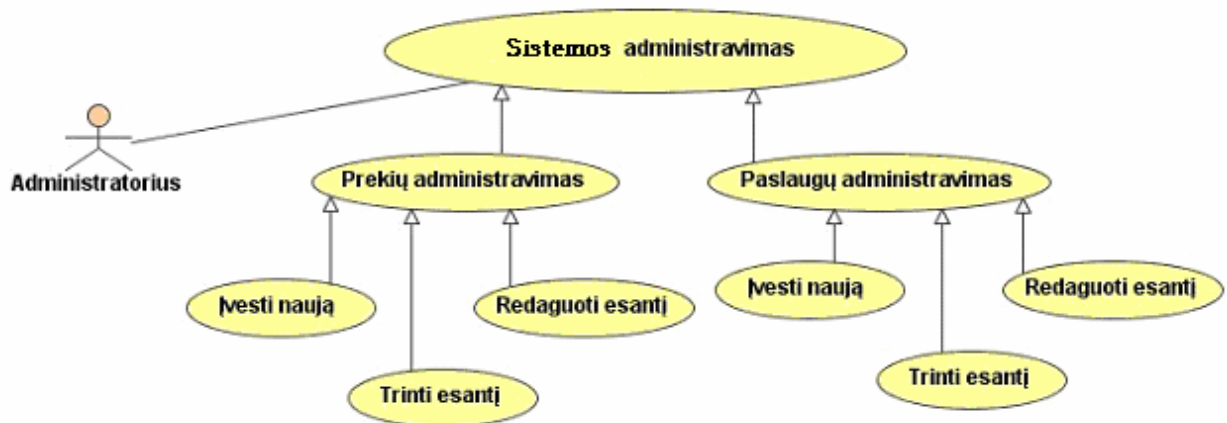
2.1 pav. Pirkėjo informacijos paieškos panaudos atvejais

Vartotojas elektroninio verslo sistemoje yra prekių ar paslaugų ieškotojas arba savininkas. Jis savo veiksmus atlieka per tinklo paslaugas. Jis gali atlikti paiešką pagal raktažodį (pavadinimą), kainą ar kategoriją, kuriai priklauso jo ieškoma prekė ar paslauga. Jis taip pat gali patalpinti savo informaciją (apie paslaugą ar prekę) sistemoje bei ją ištrinti. Vartotojo panaudos atvejai pateikti 2.2 paveiksle.

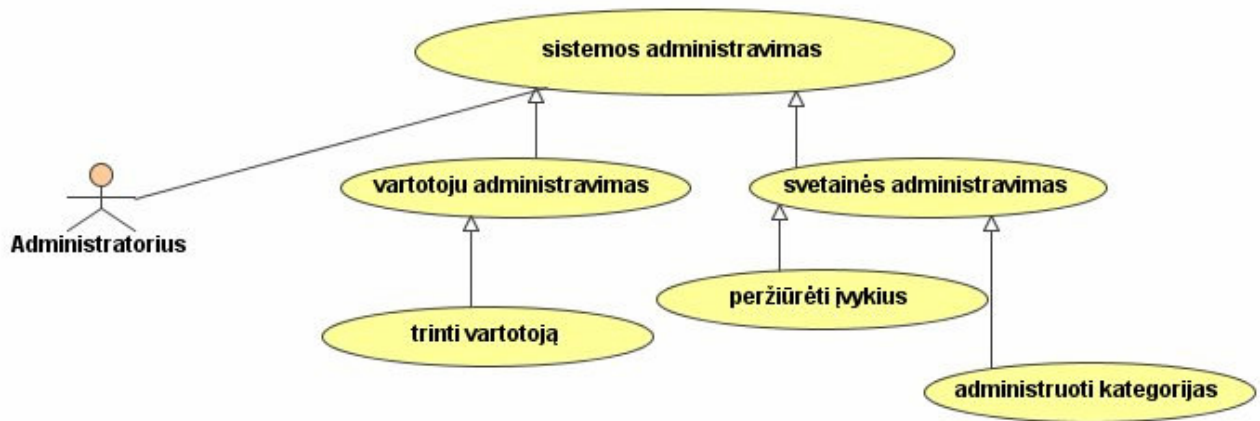


2.2 pav. Tinklo paslaugos vartotojo panaudos atvejis

Administratorius elektroninio verslo sistemoje yra prekių ir paslaugų informacijos redaguotojas, sistemos prižiūrėtojas. Administratorius vienintelis iš visu vartotojų dirba su kategorijų medžiu. Jis gali patalpinti įvesti naują paslaugą ar prekę į sistemą ar ištrinti sistemoje esančia informaciją. Administratorius taip pat gali peržiūrėti visu sistemos įvykių žurnalą, kuriame registruojama tokia informacija, kaip naujų įrašų įvedimas, senų ištrinimas. Administratoriaus panaudos atvejai pateikti 2.3 ir 2.4 paveiksluose.



2.3 pav. Sistemos informacijos administravimo panaudos atvejis



2.4 pav. Sistemos administravimo panaudos atvejais

Detalios panaudojimo atvejų specifikacijos administratoriaus veiksmams

Panaudojimo atvejis	<i>Prekių/paslaugų įvedimas</i>
Aktorius	Administratorius
Sistema	Administratoriaus panelė
Prieš sąlyga	Administratorius turi būti registruotas sistemoje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius įveda prekės/paslaugos duomenis	1.1. Sistema pateikia įvedimo formą
2. Vartotojo patvirtinimas	2.1. Sistema praneša, kad duomenys priimti
Po sąlyga	Sistemos DB atsiranda nauja prekė/paslauga
Alternatyvos	Administratorius suveda nekorektiškus duomenis. Sistema prašo juos pataisyti arba nutraukti įvedimo procesą.
Veiklos taisyklės	Nauja prekė/paslauga bus įvesta jei bus suvesti visi reikalingi duomenys ir jie bus korektiški.

Panaudojimo atvejis	<i>Prekių/paslaugų trynimas</i>
Aktorius	Administratorius
Sistema	Administratoriaus panelė
Prieš sąlyga	Administratorius turi būti registruotas sistemoje
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius pasirenka prekę/paslaugą	1.1. Sistema pateikia sąrašą
2. Administratoriaus patvirtinimas	2.1. Sistema praneša, kad duomenys ištrinti
Po sąlyga	Sistemos DB dingsta įrašas
Alternatyvos	Administratorius nepatvirtina veiksmo.
Veiklos taisyklės	Prekė/paslauga bus ištrinta, jei administratorius patvirtins jos trynimą

Panaudojimo atvejis	<i>Vartotojo trynimas</i>
Aktorius	Administratorius
Sistema	Administratoriaus panelė
Prieš sąlyga	Nėra
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius pasirenka vartotoją	1.1. Sistema pateikia vartotojų sąrašą
2. Administratoriaus patvirtinimas	2.1. Sistema praneša, kad duomenys ištrinti
Po sąlyga	Sistemos DB dingsta vartotojas
Alternatyvos	Administratorius nepatvirtina veiksmo.
Veiklos taisyklės	Administratorius bus ištrintas, jei administratorius patvirtins jo trynimą

Panaudojimo atvejis	<i>Kategorijos įvedimas</i>
Aktorius	Administratorius
Sistema	Administratoriaus panelė
Prieš sąlyga	Nėra
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius suveda savo prisijungimo vardą ir slaptažodį.	1.1. Sistema tikrina duomenis. Jei neteisingi duomenys, sistema praneša.
2. Administratorius pasirenka kategoriją	2.1. Sistema pateikia kategorijų sąrašą
3. Administratorius įveda naują kategoriją	3.1. Sistema validuoja kategoriją
4. Administratorius patvirtinimas	4.1. Sistema praneša, kad duomenys priimti
Po sąlyga	Kategorijų lentelėje atsiranda nauja kategorija
Alternatyvos	Administratorius suveda nekorektiškus duomenis. Sistema prašo juos pataisyti arba nutraukti įvedimo procesą.
Veiklos taisyklės	Kategorija bus įvesta, jei visa reikalinga informacija bus suvesta ir patvirtinta administratoriaus.

Panaudojimo atvejis	<i>Kategorijos trynimas</i>
Aktorius	Administratorius
Sistema	Administratoriaus panelė
Prieš sąlyga	Nėra
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius suveda savo prisijungimo vardą ir slaptažodį.	1.1. Sistema tikrina duomenis. Jei neteisingi duomenys, sistema praneša apie klaidą.
2. Administratorius pasirenka kategoriją	2.1. Sistema pateikia kategorijų sąrašą
3. Administratorius patvirtinimas	3.1. Sistema praneša, kad kategorija ištrinta
Po sąlyga	Kategorijų lentelėje dingsta įrašas
Alternatyvos	Administratorius nepatvirtina veiksmo
Veiklos taisyklės	Kategorija bus ištrinta, jei administratorius patvirtins jos trynimą

Detalios panaudojimo atvejų specifikacijos tinklo paslaugų vartotojo veiksmams

Panaudojimo atvejis	<i>Prekių/paslaugų įvedimas</i>
Aktorius	Vartotojas
Sistema	Tinklo paslaugos
Prieš sąlyga	Vartotojas turi būti registruotas sistemoje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pateikia prekės/paslaugos duomenis, vartotojo vardą ir slaptažodį	1.1. Sistema praneša, kad duomenys priimti
Po sąlyga	Sistemoje atsiranda nauja prekė/paslauga
Alternatyvos	Vartotojas suveda nekorektiškus duomenis. Sistema gražina klaidą.
Veiklos taisyklės	Nauja prekė/paslauga bus įvesta jei bus suvesti visi reikalingi duomenys ir jie bus korektiški.

Panaudojimo atvejis	<i>Prekių/paslaugų trynimasis</i>
Aktorius	Vartotojas
Sistema	Tinklo paslaugos
Prieš sąlyga	Vartotojas turi būti registruotas sistemoje
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pateikia prekės/paslaugos identifikacinį numerį, vartotojo vardą ir slaptažodį	1.1. Sistema praneša, kad duomenys priimti
Po sąlyga	Sistemos DB dingsta įrašas
Alternatyvos	Vartotojas suveda nekorektiškus duomenis. Sistema gražina klaidą.
Veiklos taisyklės	Prekė/paslauga bus ištrinta jei bus suvesti visi reikalingi duomenys ir jie bus korektiški.

Panaudojimo atvejis	<i>Prekių/paslaugų paieška</i>
Aktorius	Vartotojas
Sistema	Tinklo paslaugos
Prieš sąlyga	Vartotojas turi būti registruotas sistemoje.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pateikia paieškos duomenis, paieškos kriterijų, vartotojo vardą ir slaptažodį	1.1. Sistema gražina duomenų sąrašą
Po sąlyga	Nėra
Alternatyvos	Vartotojas suveda nekorektiškus duomenis. Sistema gražina klaidą.
Veiklos taisyklės	Rezultatas bus gautas jei bus suvesti visi reikalingi duomenys ir jie bus korektiški.

Detalios panaudojimo atvejų specifikacijos pirkėjo veiksmams

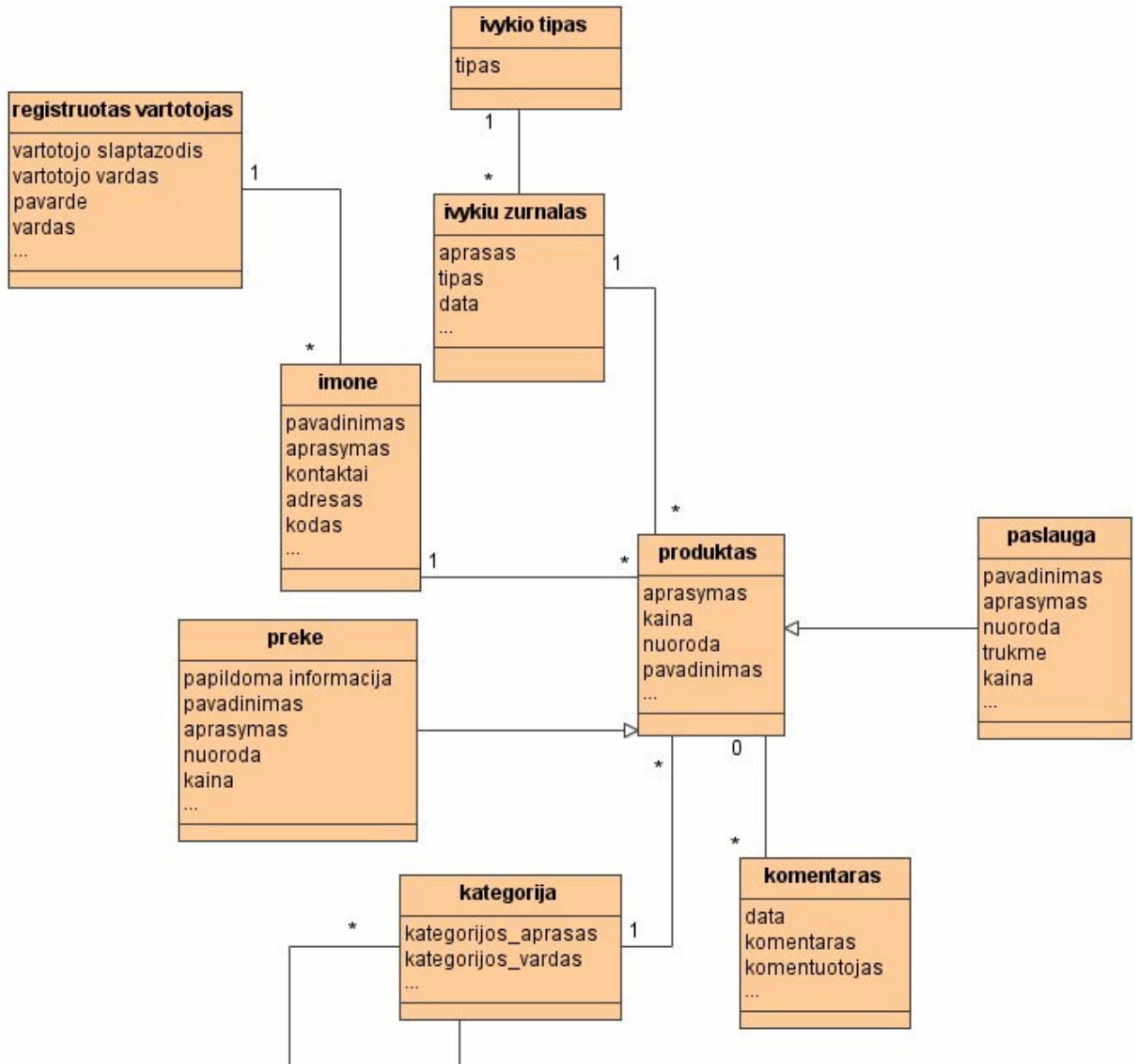
Panaudojimo atvejis	<i>Prekių/paslaugų paieška</i>
Aktorius	Pirkėjas
Sistema	Prekių/paslaugų direktorija
Prieš sąlyga	Nėra.
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pirkėjas įveda paieškos duomenis	1.1. Sistema gražina duomenų sąrašą
Po sąlyga	Nėra
Alternatyvos	Nėra
Veiklos taisyklės	Rezultatas bus gautas jei bus įvestas paieškos kriterijus.

2.3.Nefunkciniai reikalavimai sistemai

- Sistema privalo užtikrinti vartotojų prisijungimo slaptažodžių slaptumą ir saugumą;
- Sistema privalo būti suderinama su populiariausiomis interneto naršyklėmis;
- Sistema turi būti pasiekiamą iš bet kurios vietos, kur yra interneto ryšys;
- Sistema turi būti suderinama su bet kokia platforma, turinčia šiuolaikišką interneto naršyklę.

2.3.1. Paslaugų centro dalykinės srities modelis

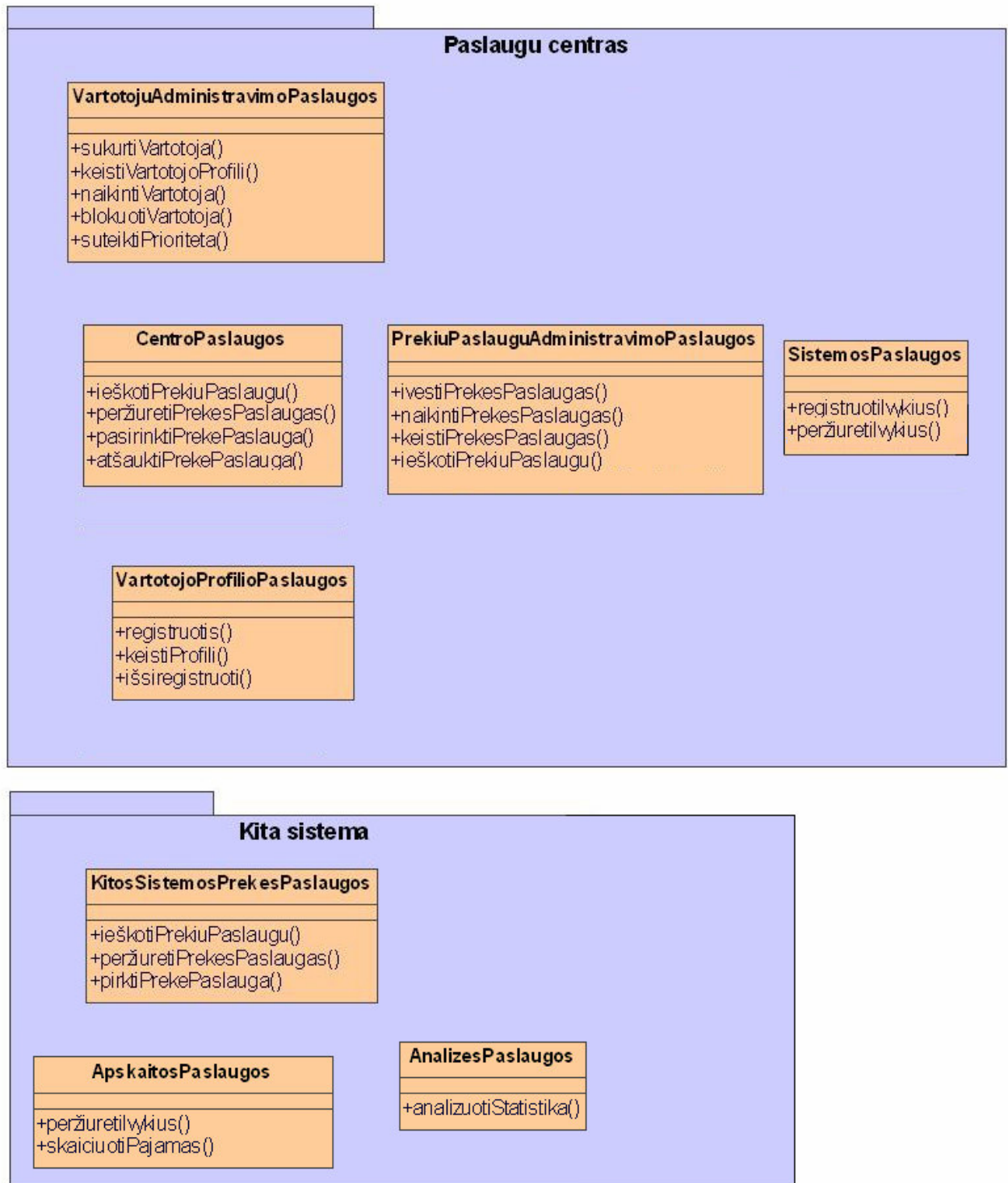
Dalykinės srities diagrama ir jos aprašymas



2.5 pav. Dalykinės srities diagrama

- Registruotas vartotojas – atspindi sistemoje registruotus vartotojus bei informaciją apie juos.
- Prekė – atsakinga už prekes;
- Paslauga – apjungia metodus darbui su paslaugom;
- Produktas – susideda iš prekės arba paslaugos;
- Imonė – naudojama produkto, saugo detalią informaciją apie tiekėją, t.y. jo adresą, kontaktinę informaciją, aprašą ir kt;
- Kategorija – kategorijų medis;

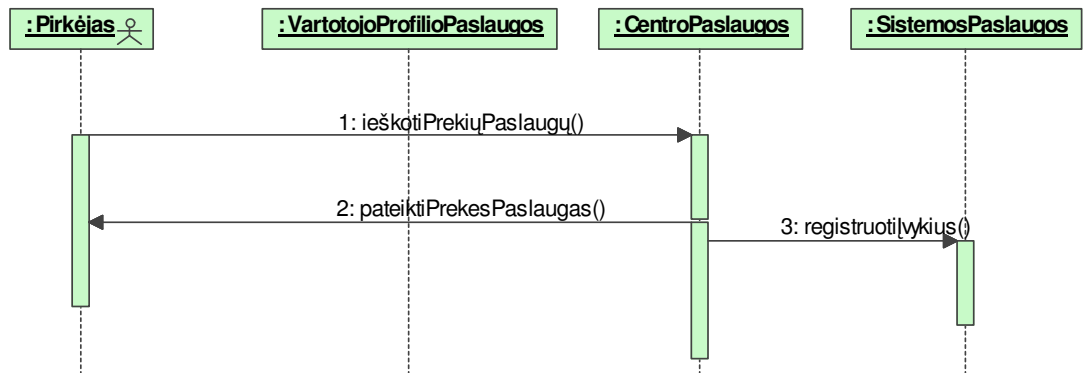
- Komentaras – komentarai.
- Įvykių žurnalas – informacija apie sistemos veiklą: įdėtos naujos, ištrintos senos prekės, paslaugos ir kt;
- Įvykio tipas – įvykio tipas nusako įvyki: veikla susijusi su vartotoju, prekėm, paslaugom.



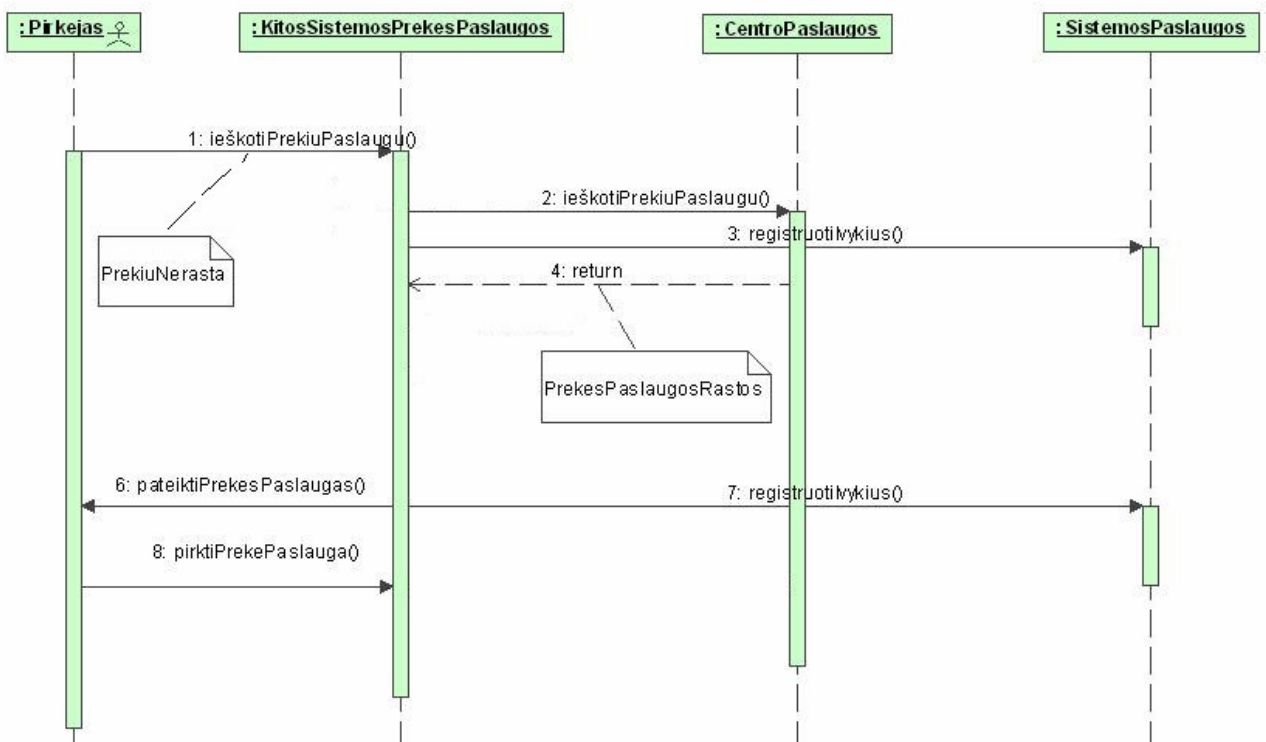
2.6 pav. Klasių diagrama

2.6 paveiksle pateikiama apibendrinta klasių diagrama.

2.7 paveikle parodyta paieškos sekų diagrama, kuomet paslaugos ar prekės yra ieškomos paslaugų centre.



2.7 pav. Paieškos sekų diagrama: informacijos ieškoma centre



2.8 pav. Paieškos sekų diagrama: informacijos ieškoma kitoje sistemoje

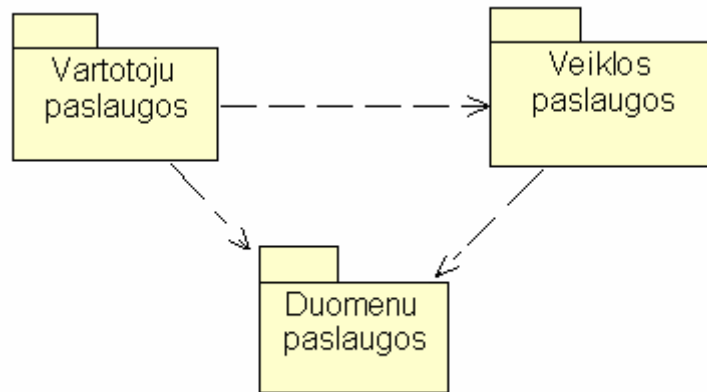
2.8 paveikle pateikta paieškos sekų diagrama, kuomet informacija pas prekių ar paslaugų teikėja nėra randama, tuomet automatiškai paieška išplečiama ir radus informacijos atitinkančios paieškos kriterijus, paslaugų centras ją pateikia.

2.4.Sistemos architektūra – statinės struktūros modelis

2.4.1. Loginė visos sistemos architektūra

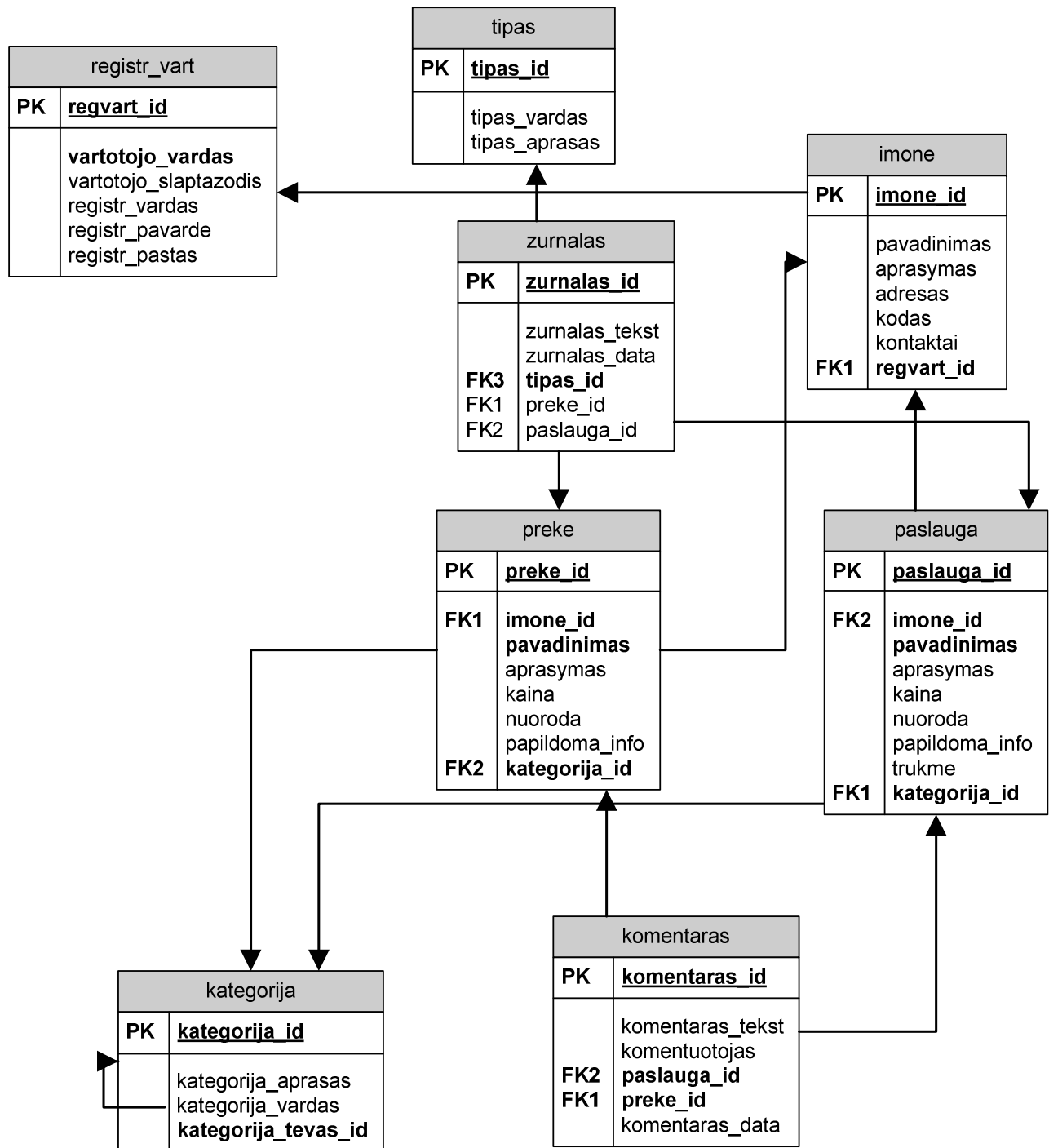
Loginę visos sistemos architektūrą sudaro šie komponentai:

- Vartotojų paslaugos;
- Veiklos paslaugos;
- Duomenų paslaugos.



2.9 pav. Sistemos architektūra

2.5. Duomenų bazės schema



2.10 pav. Duomenų bazės schema

paslauga:

- **paslauga_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **imone_id** – įmonės, kuri siūlo šią prekę, lauko lentelėje unikalus identifikatorius;

- **aprasymas** – paslaugos bei jos savybių aprašas;
- **pavadinimas** – teikiamos paslaugos pavadinimas realiame pasaulyje;
- **kaina** – siūlomos paslaugos kaina realiame pasaulyje;
- **nuoroda** – nuoroda į informaciją apie šią paslaugą, įmonės svetainėje;
- **papildoma_info** – papildoma informacija apie siūlomą paslaugą;
- **trukme** – informacija apie teikiamos paslaugos trukmę;
- **kategorija_id** – kategorijos įrašo identifikatorius, nurodantis, kuriai kategorijai ši paslauga priskiriama.

Esybė skirta apibūdinti vienai iš įmonės siūlomų paslaugų.

preke:

- **preke_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **imone_id** – įmonės, kuri siūlo šią prekę, lauko lentelėje unikalus identifikatorius;
- **aprasymas** – prekės bei jos savybių aprašas;
- **pavadinimas** – prekės pavadinimas realiame pasaulyje;
- **kaina** – prekės kaina realiame pasaulyje;
- **nuoroda** – nuoroda į prekę, įmonės svetainėje;
- **papildoma_info** – papildoma informacija apie prekę;
- **kategorija_id** – kategorijos įrašo identifikatorius, nurodantis, kuriai kategorijai ši prekė priskiriama.

Esybė skirta apibūdinti vienai iš įmonės teikiamų prekių.

tipas:

- **tipas_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **tipas_vardas** – žurnalo įvykio tipo vardas;
- **tipas_aprasas** – žurnalo įvykio tipo aprašas.

Esybė aprašo tipą, kuriam priklauso įvykių žurnalo įrašai.

zurnalas:

- **zurnalas_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **tipas_id** – tipo įrašo identifikatorius skirtas nurodyti, kokiam tipui šis įrašas priskiriamas;
- **preke_id** – prekės įrašo identifikatorius skirtas nurodyti, apie kurią prekę yra šis įrašas;
- **paslauga_id** – paslaugos įrašo identifikatorius skirtas nurodyti, apie kurią paslaugą yra šis įrašas;
- **zurnalas_tekst** – informacija apie sveiksmą tekstiniame formate;
- **zurnalas_data** – įrašo įterpimo data ir laikas.

Esybė aprašo įvykių žurnalą, saugo informaciją, apie kurią prekę ar paslaugą šis įrašas yra, bei kuomet šis įvykis įvyko.

komentaras:

- **komentaras_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **komentuotojas** – svetainės lankytojo pasirašytas vardas;
- **preke_id** – prekės įrašo identifikatorius skirtas nurodyti, kuriai prekei šis komentaras priklauso;
- **paslauga_id** – paslaugos įrašo identifikatorius skirtas nurodyti, kuriai paslaugai šis komentaras priklauso;
- **komentaras_tekst** – komentaras tekstiniame formate;
- **komentaras_data** – įrašo įterpimo data ir laikas.

Esybė aprašo komentarą, saugo informaciją, kuriai prekei ar paslaugai komentaras priklauso, taip pat pasirašyta komentuotojo slapyvardį bei kuomet šis komentaras buvo parašytas.

registr_vart:

- **regvart_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **vartotojo_vardas** – unikalus įrašas, vartotojo vardas prisijungiant prie sistemos;
- **vartotojo_slaptazodis** – vartotojo slaptažodis sistemoje;

- **registr_vardas** – vartotojo vardas realiame pasaulyje;
- **registr_pavarde** – vartotojo pavardė realiame pasaulyje;
- **registr_pastas** – vartotojo elektroninio pašto adresas.

Esybė skirta nusakyti registruotą sistemos vartotoją.

katgorija:

- **katgorija_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **katgorija_vardas** – katgorijos pavadinimas realiame pasaulyje;
- **katgorija_tervas_id** – katgorijos įrašo identifikatorius skirtas susieti katgorijos įrašą su jo tėvo katgorijos įrašu;
- **katgorija_aprasas** – katgorijos įrašo tekstinis aprašymas;

Esybė sudaro katgorijų medį, kurio kiekvienam lapui gali priklausyti paslaugos ar prekės.

imone:

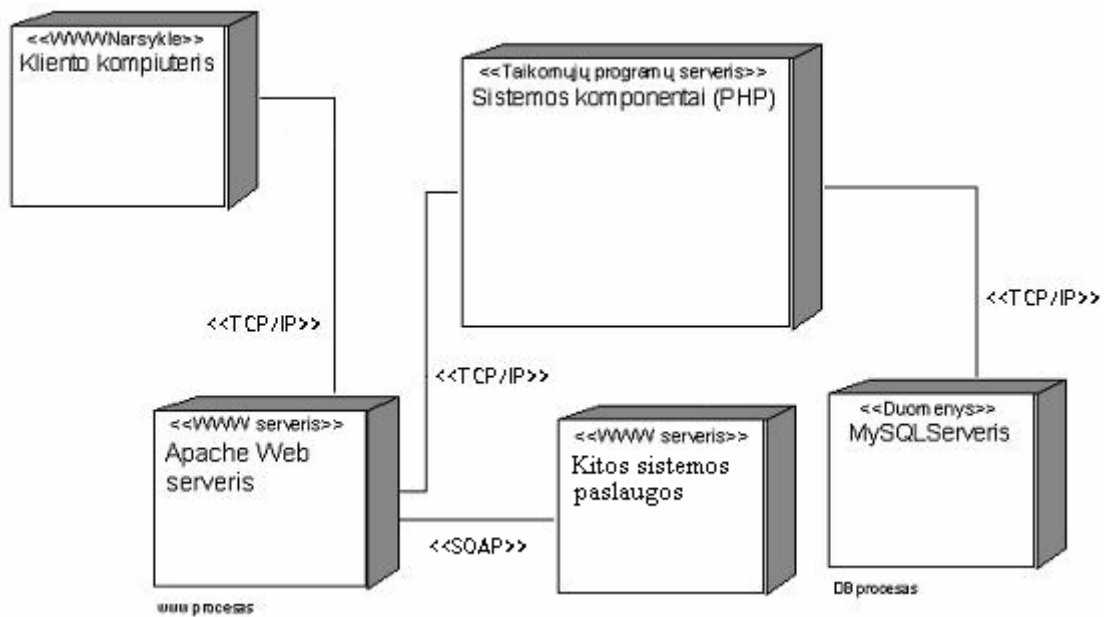
- **imone_id** - pirminis raktas, kodinis pavadinimas kreipiantis į lentelę;
- **pavadinimas** – įmonės pavadinimas realiame pasaulyje;
- **regvart_id** – registruoto vartotojo įrašo identifikatorius skirtas nurodyti, kas įvedė šią įmonę;
- **adresas** – įmonės adresas realiame pasaulyje;
- **kodas** – įmonės kodas;
- **kontaktai** – įmonės kontaktinė informacija.

Esybė aprašo įmonę, saugo jos kontaktinę ir kt. informaciją.

2.6. Realizacijos modelis

Įdiegimo diagrama sudaryta iš šių komponentų:

- Kliento kompiuteris;
- Apache interneto tarnybinė stotis;
- Sistemos komponentai;
- Duomenų bazės tarnybinė stotis;
- Tinklo paslaugų tarnybinė stotis.



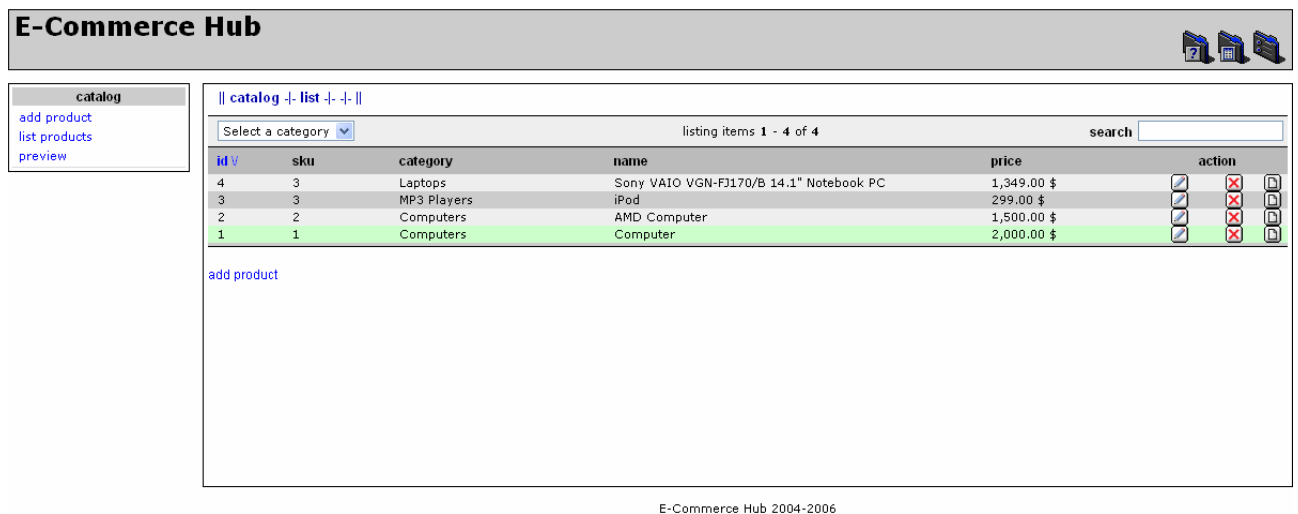
2.11 pav. Sistemos diegimo diagrama

3. ELEKTRONINĖS VERSLO APLINKOS REALIZAVIMAS













3.1. Elektroninio verslo aplinkos sąsaja

Vienas vizualus šablonas (*Template*) buvo naudojamas vartotojo sąsajai realizuoti tam, kad sistema atitiktų viena pagrindinių reikalavimų – nesudėtingas sistemos naudojimas. Visi vartotojo sąsajos langai naudoja bendrą struktūrą ir dizainą, dėl to vartotojas greitai apsipranta su sistema ir jau pirmomis darbo dienomis jo daromų klaidų skaičius sumažėja iki minimumo.

3.1.1. Administratoriaus sąsaja



The screenshot displays the 'E-Commerce Hub' administrator interface. On the left is a sidebar menu with 'catalog' selected, containing links for 'add product', 'list products', and 'preview'. The main content area shows a table of products with columns for 'id', 'sku', 'category', 'name', 'price', and 'action'. The table lists four items, with the first one highlighted in green. Below the table is an 'add product' link. The footer of the interface reads 'E-Commerce Hub 2004-2006'.

id	sku	category	name	price	action
4	3	Laptops	Sony VAIO VGN-FJ170/B 14.1" Notebook PC	1,349.00 \$	  
3	3	MP3 Players	iPod	299.00 \$	  
2	2	Computers	AMD Computer	1,500.00 \$	  
1	1	Computers	Computer	2,000.00 \$	  

3.1 pav. Administratoriaus sąsaja: prekių sąrašas

Administratorius gali atlikti 2.2 skyriuje aprašytus veiksmus: redaguoti, trinti, sukurti naują ar paprasčiausiai peržiūrėti prekių ir paslaugų sąrašus - 3.1 paveikslas. Prekės redagavimo sąsaja parodyta 3.2 paveiksle, o kategorijų medis su galimybe redaguoti, trinti ar įterpti naują kategoriją – 3.3 paveiksle.

E-Commerce Hub

catalog

add product
list products
preview

|| catalog - edit - - ||

product sku: 3

product name: Sony VAIO VGN-F3170/B 14.1" Notebook PC

price: 1349.00 \$

color: black

size: 10.0 x 13.4 x 1.3

category: Laptops

description: Intel Pentium M Processor 740, 512 MB RAM, 100 GB Hard Drive, DVD+R Dobule Layer/DVD+/-R Drive

image file: Browse... B000BK1IBM.01-A1ND6S7YGOPBD6_SCLZZZZZZZ.jpg

thumbnail generation: yes no Scale: 50 %

thumbnail file: Browse... not_available.png

use tag: For Sale Sold Special

save

E-Commerce Hub 2004-2006

3.2 pav. Administratoriaus sąsaja: prekės redagavimas

E-Commerce Hub

config

categories
config

|| config - categories - - ||

id	category	action		
1	Root Category	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Electronics	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Computers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Laptops	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Sound Systems	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	MP3 Players	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	TVs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

E-Commerce Hub 2004-2006

3.3 pav. Administratoriaus sąsaja: kategorijų medis

3.1.2. Pirkėjo sąsaja

Pirkėjo sąsaja susideda iš vertikalios kategorijų medžio, horizontalios kelio nuo pradinio svetainės puslapio iki kategorijos, kurioje pirkėjas susirado prekę. Pagal 2.2 skyrių yra galimybė atlikti prekių ar paslaugų paiešką. Pirkėjo sąsaja pavaizduota 3.4 paveiksle.

Navigation Home / Electronics / Computers / Laptops

Search

Electronics
Computers
Laptops
Sound Systems
MP3 Players
TVs



Category: Laptops
Item: Sony VAIO VGN-FJ170/B 14.1" Notebook PC
Color: black
Size: 10.0 x 13.4 x 1.3
Price: 1,349.00 \$
Intel Pentium M Processor 740, 512 MB RAM, 100 GB Hard Drive, DVD+R/Dobule Layer/DVD+-RW Drive

Guest

Comment

[back](#)

Top Products
[Sony VAIO VGN-FJ170/B 14.1" Notebook PC \(2\)](#)
[iPod \(1\)](#)

3.4 pav. Pirkėjo sąsaja: prekės informacija

3.1.3. Tinklo paslaugos sąsaja

Tinklo paslaugos pagalba atliekama prekės paieška pateikiama WSDL programinio kodo pavidale:

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:si="http://soapinterop.org/xsd" xmlns:tns="http://localhost/mag/test"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://localhost/mag/test">
  <types>
    <xsd:schema targetNamespace="http://localhost/mag/test">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      <xsd:complexType name="StrArr">
        <xsd:complexContent>
          <xsd:restriction base="SOAP-ENC:Array">
            <xsd:attribute ref="SOAP-ENC:arrayType"
wsdl:arrayType="xsd:string[]" />
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>
  </types>
</definitions>
```

```

<xsd:complexType name="Preke">
  <xsd:all>
    <xsd:elekodt name="preke_id" type="xsd:string" />
    <xsd:elekodt name="kaina" type="xsd:string" />
    <xsd:elekodt name="aprasymas" type="xsd:string" />
    <xsd:elekodt name="imone_id" type="xsd:string" />
    <xsd:elekodt name="nuoroda" type="xsd:string" />
    <xsd:elekodt name="kategorija" type="xsd:int" />
    <xsd:elekodt name="pavadinimas" type="xsd:string" />
    <xsd:elekodt name="papildoma_info" type="xsd:string" />
    <xsd:elekodt name="imones_pavadinimas" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="PrekesArr">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType"
wsdl:arrayType="tns:Preke[]" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PrekeType">
  <xsd:all>
    <xsd:elekodt name="kategorija" type="xsd:int" />
    <xsd:elekodt name="pavadinimas" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="TypesArr">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType"
wsdl:arrayType="tns:PrekeType[]" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>
</types>
<message name="ar_egzist_prekeRequest">
  <part name="pav" type="xsd:string" />
  <part name="kategorija" type="xsd:string" />
</message>
<message name="ar_egzist_prekeResponse">
  <part name="return" type="tns:StrArr" />
</message>
<message name="gauti_prekeRequest">
  <part name="pav" type="xsd:string" />
  <part name="kategorija" type="xsd:string" />
  <part name="kain" type="xsd:string" />
  <part name="kod" type="xsd:string" />
  <part name="nuor" type="xsd:string" />
</message>
<message name="gauti_prekeResponse">
  <part name="return" type="tns:PrekesArr" />
</message>
<message name="get_AllTypesRequest" />
<message name="get_AllTypesResponse">
  <part name="return" type="tns:TypesArr" />
</message>
<message name="prisijungtiRequest">
  <part name="userid" type="xsd:string" />

```

```

    <part name="pswd" type="xsd:string" />
</message>
<message name="prisijungtiResponse">
    <part name="return" type="xsd:string" />
</message>
<portType name="aPortType">
    <operation name="ar_egzist_preke">
        <input message="tns:ar_egzist_prekeRequest" />
        <output message="tns:ar_egzist_prekeResponse" />
    </operation>
    <operation name="gauti_preke">
        <input message="tns:gauti_prekeRequest" />
        <output message="tns:gauti_prekeResponse" />
    </operation>
    <operation name="get_AllTypes">
        <input message="tns:get_AllTypesRequest" />
        <output message="tns:get_AllTypesResponse" />
    </operation>
    <operation name="prisijungti">
        <input message="tns:prisijungtiRequest" />
        <output message="tns:prisijungtiResponse" />
    </operation>
</portType>
<binding name="aBinding" type="tns:aPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"
/>
    <operation name="ar_egzist_preke">
        <soap:operation
soapAction="http://localhost/mag/test/index.php?a=ar_egzist_preke" style="rpc"
/>
        <input>
            <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output>
            <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </output>
    </operation>
    <operation name="gauti_preke">
        <soap:operation
soapAction="http://localhost/mag/test/index.php?a=gauti_preke" style="rpc" />
        <input>
            <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output>
            <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </output>
    </operation>
    <operation name="get_AllTypes">
        <soap:operation
soapAction="http://localhost/mag/test/index.php?a=get_AllTypes" style="rpc" />
        <input>
            <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output>

```

```

        <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
</operation>
<operation name="prisijungti">
    <soap:operation
soapAction="http://localhost/mag/test/index.php?a=prisijungti" style="rpc" />
    <input>
        <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
        <soap:body use="encoded" namespace="http://localhost/mag/test"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
</operation>
</binding>
<service name="a">
    <port name="aPort" binding="tns:aBinding">
        <soap:address location="http://localhost/mag/test/index.php" />
    </port>
</service>
</definitions>

```

3.2. Sukurtos sistemos kokybė

Eksperimentinį kokybės įvertinimą atliko trys IT specialistai:

- Paulinas Dzenkauskas – Ekspertas 1;
- Aurimas Kučinovas – Ekspertas 2;
- Birutė Kudirkaitė – Ekspertas 3.

3.2.1. Sistemos kokybės užtikrinimas

Sukurtos sistemos kokybė yra tuo aukštesnė, kuo labiau sistema atitinka jai iškeltus reikalavimus analizės ir projektavimo etapuose.

Yra daug sistemos kokybės užtikrinimo būdų. Vienas jų yra kiekybinis kokybės užtikrinimas. Šio metodo esmė yra ta, kad iš pradžių nustatoma maksimali riba galimam klaidų kiekiui sistemoje, o projekto įgyvendinimo metu stebima kuriamos sistemos kokybė ir neatitikus išsikeltų reikalavimų atliekami papildomi darbai, kad sistema jiems atitiktų. Tačiau dėl patirties trūkumo realizuojant tinklo paslaugas, šio metodo teko atsisakyti.

Kitas sistemos kokybės užtikrinimo būdas yra toks: sistemos analizės pabaigoje sukuriama kokybės užtikrinimui skirtos procedūros ir projekto realizacijos metu atliekant atitikimo kokybės užtikrinimui testą klaidos buvo randamos. Šis metodas vadinamas procedūriniu.

Didžioji dalis programinio kodo klaidų buvo randamos programavimo pradžioje, nes programuojant, nuolat buvo atliekama sistemos struktūros pakeitimai radus neatitikimų. O likusi nedidelė klaidų grupė buvo eliminuota jau sukūrus ir ištestavus verslo centrą. Atlikus visus

pakeitimus buvo panaikintos visos atrastos klaidos ir galima teigti, kad likęs neatrastų klaidų kiekis yra minimalus ir sistema atitinka išsikeltus kokybės kriterijus.

3.2.2. Elektroninio verslo aplinkos kokybės įvertinimas

Kokybės įvertinimo atlikimo procedūra tokia: sukurtas elektroninio verslo centras bus palyginamas su kitomis rinkoje įsitvirtinusiomis sistemomis, teikiančiomis tinklo paslaugas. Kadangi tos sistemos buvo kuriamos pagal kitokius reikalavimus, šis kokybės įvertinimas nėra tikslus.

Kokybės įvertinimas buvo atliktas 5 balų skalėje, kur kuo didesnis skaičius, tuo didesnis kokybės atitikimas.

3.3. Kokybės charakteristikos

3.3.1. Realus laiko informacijos apsikeitimas

3.1.lentelė Realus laiko informacijos gavimo tinklo paslaugų pagalba įvertinimas

Vertintojas	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Ekspertas 1	3	3	4	4
Ekspertas 2	3	3	4	4
Ekspertas 3	3	3	4	4
Vidurkis	3	3	4	4

3.1.1. Integravimo paprastumas su kitomis sistemomis

3.2.lentelė Integravimo paprastumas su kitomis sistemomis įvertinimas

Vertintojas	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Ekspertas 1	2	4	4	4
Ekspertas 2	2	4	4	4
Ekspertas 3	2	4	4	4
Vidurkis	2	4	4	4

Šio projekto metu sukurtas verslo centras su kitomis sistemomis komunikuoja tinklo paslaugų technologijos, kuri užtikrina didelį lankstumą, pagalba.

Amazon gražinamas prekių sąrašas, esant mažai paklausai pasirinktam raktažodžiui, ne visuomet atitinka norimą rezultatą: gali būti gautas produktas, kuris jau yra parduotas ar produktas neatitinkantis užduoto raktažodžio.

3.2.1. Funkcinių galimybių išplečiamumas

3.3.lentelė Funkcinių galimybių išplečiamumo įvertinimas

Vertintojas	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Ekspertas 1	2	2	3	5
Ekspertas 2	2	2	3	5
Ekspertas 3	2	2	3	5
Vidurkis	2	2	3	5

Alternatyvios sistemos yra dar bandomojoje stadijoje, dėl to jų plečiamumo kokybė gali kisti. Amazon ir Alexa sistemos riboja užklausų kiekį per parą, tai gali sukelti tam tikru nepatogumu.

Tinklo paslaugų technologijos architektūra leidžia sistemos funkcionalumą plėsti ne tik PHP programavimo kalbos pagalba, bet ir remtis kitokiomis programinės įrangos kūrimo priemonėmis.

3.3.1. Instaliavimas ir valdymas

3.4.lentelė Instaliavimo ir valdymo paprastumo įvertinimas

Vertintojas	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Ekspertas 1	5	5	3	5
Ekspertas 2	5	5	3	5
Ekspertas 3	5	5	3	5
Vidurkis	5	5	3	5

3.4.1. Daugiaplatformiškumas

3.5.lentelė Daugiaplatformiškumo įvertinimas

Vertintojas	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Ekspertas 1	5	5	5	5
Ekspertas 2	5	5	5	5
Ekspertas 3	5	5	5	5
Vidurkis	5	5	5	5

3.5.1. Kokybės įvertinimo apibendrinimas

Verslo centro kokybės įvertinimo metu buvo nustatyta, kad centras atitinka jam iškeltus kokybės reikalavimui ir dėl savo paprastumo pagal kai kuriuos kokybės kriterijus pasirodė šiek

ties geriau už kitas sistemas. Galime teigti, kad dėl sėkmingo verslo centro kokybės užtikrinimo projekto realizavimo metu, sistema atitinka šiuolaikinius kokybės kriterijus.

3.6 lentelėje pateikiamas projekto metu sukurtos sistemos ir kitų trijų sistemų įvertinimas pagal aukščiau išvardintus kriterijus.

3.6.lentelė Kokybės įvertinimo apibendrinimas

Kriterijus	Amazon E-Commerce	Alexa Web Info.	Google API	Sukurta sistema
Realaus laiko informacija, gaunama iš kitų sistemų	3	3	4	4
Integravimo paprastumas su kitomis sistemomis	2	4	4	4
Funkcinių galimybių išplečiamumas	2	2	3	5
Instaliavimo ir valdymo paprastumas	5	5	3	5
Daugiaplatformiškumas	5	5	5	5

IŠVADOS

- Ankstesnės skirtingų sistemų komunikavimo technologijos tampa neefektyvios dėl informacijos perdavimo ir metodų iškvietimo bendro standarto nebuvimo. Išspręsti šiomis problemomis prireikė sukurti bendrą skirtingų sistemų komunikavimo standartą;
- Turint tikslą sukurti lengvai automatizuojamą informacijos apsikeitimo sistemą paprasčiausias ir pigiausias variantas yra pasitelkti tinklo paslaugų technologiją, kuri suteikia galimybę skirtingoms sistemoms (skirtingose platformose ar operacinėse sistemose) komunikuoti vienu globaliu standartiniu būdu. Šis priklausomybės nebuvimas ir yra pagrindinis tinklo paslaugų privalumas;
- Išanalizavimo tinklo paslaugas ir jų aprašymo kalbą WSDL, galime padaryti išvadą, kad tai rimta priemonė sistemų komunimavimui standartizuoti ir realizuoti. Rezultate gaunama programinė įranga kuri yra pigesnė, lengviau plečiama ir taisoma bei nenusileidžia funkcinėmis galimybėmis ir neturi priklausomybės nuo pasirinktos platformos ar realizacijai naudojamos programavimo kalbos;
- Sukurtas elektroninio verslo tinklo paslaugų sistemos modelis, kuris leidžia automatizuoti informacijos apsikeitimą su kitomis sistemomis dirbančiomis su panašia informacija. Tai ženkliai sumažina su centru bendraujančių sistemų priežiūros darbą bei darbui su duomenimis skirtų priemonių tobulinimo kaštus;
- Sistema leidžia interneto parduotuvių ir kitiems prekių savininkams publikuoti savo turimą informaciją sukurtame elektroninio verslo centre bei gauti papildomos informacijos savo svetainių inventoriui užpildyti. Visai tai atliekama tinklo paslaugų pagalba, dėl to yra lengvai automatizuojama ir nepriklauso nuo to, kokias technologijas naudoja vartotojai;
- Elektroninio verslo sistemos modelį galima naudoti kaip šabloną realizuojant kitų sričių sistemas, kuriose realizuojamas informacijos apsikeitimas tarp skirtingų įmonių ir gali būti duomenų apsikeitimas skirtingose platformose;
- Elektroninio verslo kokybės tyrimo rezultatai parodo, kad sukurtas sistemos variantas atitinka analizės ir projekto etapuose išsikeltus techninius reikalavimus.

LITERATŪRA

1. Using WSDL in SOAP applications, Uche Ogbuji. [žiūrėta 2004-12-17]. Prieiga per internetą: <http://www-106.ibm.com/developerworks/webservices/library/ws-soap/?dwzone=ws>
2. Web Services Description Language (WSDL) 1.1, W3C. [žiūrėta 2004-11-21]. Prieiga per internetą: <http://www.w3.org/TR/wsdl>
3. Top Ten FAQs for Web Services, Ethan Cerami. [žiūrėta 2004-11-27]. Prieiga per internetą: <http://www.oreillynet.com/lpt/a/webservices/2002/02/12/webservicefaqs.html>
4. Web Service Description Usage Scenarios, Waqar Sadiq, Sandeep Kumar. [žiūrėta 2004-12-07]. Prieiga per internetą: <http://www.w3.org/TR/2002/WD-ws-desc-usecases-20020604>
5. Understanding WSDL, Aaron Skonnard. [žiūrėta 2004-12-07]. Prieiga per internetą: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/understandWSDL.asp>
6. An Introduction to the Web Services Architecture and Its Specifications, Luis Felipe Cabrera, Christopher Kurt, Don Box. [žiūrėta 2004-12-25]. Prieiga per internetą: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwsdl/html/boxwsdl.asp>
7. Google Web APIs Reference. [žiūrėta 2004-12-27]. Prieiga per internetą: <http://www.google.com/apis/reference.html>
8. WSDL Essentials. [žiūrėta 2005-01-05]. Prieiga per internetą: <http://www.developer.com/services/article.php/1602051>
9. XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition) [žiūrėta 2004-12-28]. Prieiga per internetą: <http://www.w3.org/TR/xhtml1/>
10. Cascading Style Sheets [žiūrėta 2004-12-28]. Prieiga per internetą: <http://www.w3.org/Style/CSS/>
11. SOAP v1.1 Specification [žiūrėta 2005-05-12]. Prieiga per internetą: <http://www.w3.org/tr/SOAP>
12. Informatika: Populiariųjų duomenų bazių valdymo sistemų palyginimas [žiūrėta 2005-06-12]. Prieiga per internetą: <http://distance.ktu.lt/kursai/informatika1/8/teorija13.html>

13. MySQL Manual: 1.2.4 How Big MySQL Tables Can Be [žiūrēta 2005-06-12]. Prieiga per internetą: http://dev.mysql.com/doc/mysql/en/Table_size.html
14. Extensible Markup Language (XML). [žiūrēta 2005-06-12]. Prieiga per internetą: <http://www.w3.org/XML/>
15. Web Services Demystified, K. Yank. [žiūrēta 2005-06-13]. Prieiga per internetą: <http://www.sitepoint.com/article/web-services-demystified>
16. BOOTH, D.; HAAS, H.; NEWCOMER, E. Web Services Architecture, D. Booth, E. Newcomer, H. Haas. [žiūrēta 2005-06-16]. Prieiga per internetą: <http://www.w3.org/tr/2004/NOTE-ws-arch-20040211/>
17. About UDDI. [žiūrēta 2005-06-16]. Prieiga per internetą: <http://www.uddi.org/>
18. Simple Object Access Protocol (SOAP). [žiūrēta 2005-06-16]. Prieiga per internetą: <http://www.w3.org/TR/soap/>

TERMINAI IR ŽODYNAS

Reused

URI (*Unified Resource Identifier*)

HTTP (*Hypertext Transport Protocol*)

XML (*Extensible Markup Language*)

SOAP (*Simple Object Access Protocol*)

UML (*Unified Modelling Language*)

MySQL

MagicDraw UML

TCP/IP

PHP

REST (*Representational State Transfer*)

OMG (*Object Management Group*)

WSDL (*Web Service Description Language*)

CORBA (*Common Object Request Broker
Architecture*)

COM (*Component Object Model*)

SOM (*System Object Model*)

UDDI (*Universal Description, Discovery, and
Integration*)

Pakartotinis programinio kodo panaudojimas.

Bendras resursų identifikatorius.

Aktyviojo teksto perdavimo protokolas.

Išplėsta žymėjimo kalba.

Paprasto objektų pasiekiamumo protokolas.

Apjungta modeliavimo kalba.

Duomenų bazių valdymo sistema

Projektavimo priemonė.

Standartinis interneto duomenų perdavimo
protokolas.

Serverio pusės programavimo kalba.

Tinklo programų kūrimo architektūrinis stilius

Objektų valdymo grupė

Tinklo paslaugų aprašymo kalba.

Bendra objektų kreipinių valdytojo architektūra.

Komponentinis objektų modelis.

Sistemos objektų modelis.

Universalus aprašymas, atradimas ir
integravimas.