

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Artūras Maraška

**Paskirstytų duomenų surinkimo ir apdorojimo
technologijos**

Magistro darbas

Recenzentas

Prof. Dr. R. Butleris

Darbo vadovas

Doc. Dr. E. Karčiauskas

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Artūras Maraška

Paskirstytų duomenų surinkimo ir apdorojimo technologijos

Magistro darbas

Recenzentas

Prof. Dr. R. Butleris
2009.05.22

Vadovas

Doc. Dr. E. Karčiauskas
2009.05.22

Atliko

IFM-3/2 gr. stud.
Artūras Maraška
2009.05.22

Kaunas, 2009

TURINYS

SUMMARY	4
SANTRUMPŲ IR TERMINŲ ŽODYNAS	5
1. ĮVADAS	6
2. TECHCNOLOGIJŲ IR PROGRAMŲ SISTEMŲ APŽVALGA	9
2. 1. Tyrimo sritis, objektas ir problema.....	9
2. 2. Objektas – unikali, lanksti žinių bazės sistema	9
2. 2. 1. <i>Sistemos problemos.....</i>	10
2. 2. 2. <i>Sistemos kūrimo tikslai.....</i>	10
2. 2. 3. <i>Duomenų surinkimo sistemų veiklos modeliai.....</i>	11
2.2.3.1. <i>Interaktyvios sistemos.....</i>	12
2.2.3.2. <i>„Siuntos“ („batch“) sistemos.....</i>	12
2.2.3.3. <i>Hibridinės sistemos</i>	12
2. 3. <i>„Wiki“ technologijos.....</i>	13
2. 3. 1. <i>„Wiki“ organizacine prasme</i>	13
2. 4. <i>Duomenų surinkimo ir valdymo metodai</i>	15
2. 4. 1. <i>Pagrindiniai principai</i>	15
2. 5. <i>Duomenų analizės metodai.....</i>	16
2. 5. 1. <i>Aktyvi analizė.....</i>	16
2. 5. 2. <i>Pasyvi analizė.....</i>	17
2. 6. <i>Egzistuojančių sistemų apžvalga.....</i>	17
2. 6. 1. <i>Sistemų vertinimo kriterijai</i>	18
2. 6. 2. <i>„VLE“ Moodle sistema</i>	18
2. 6. 3. <i>Išvados apie „Moodle“</i>	21
2. 6. 4. <i>„Microsoft Sharepoint-Wiki“ sistema.....</i>	21
2. 6. 5. <i>Išvados apie „Shaperiont-WIKI“</i>	23
2. 7. <i>Analizės išvados ir sprendimų pagrindimas</i>	23
3. PROJEKVINĖ DALIS	24
3. 1. <i>Sistemos panaudos atvejai.....</i>	24
3. 1. 1. <i>Sistemos administratoriaus pagrindiniai panaudojimo atvejai</i>	25
3. 1. 2. <i>Sistemos kliento pagrindiniai panaudojimo atvejai.....</i>	26
3. 1. 3. <i>Sistemos pagrindiniai panaudojimo atvejai.....</i>	28
3. 1. 4. <i>Sistemos vartotojo pagrindiniai panaudojimo atvejai.....</i>	29
3. 2. <i>Sistemos komponentų detalizavimas.....</i>	30
3. 3. <i>Sistemos pagrindiniai paketai</i>	32
3.3.1. <i>Paketas „GVS“</i>	32
3.3.1.1. <i>DefaultGVS</i>	33
3.3.1.2. <i>ClientGVS.....</i>	33
3.3.1.3. <i>AdminGVS.....</i>	34
3.3.2. <i>Paketas „Servisai“</i>	35
3.3.3. <i>Paketas „Duomenys“</i>	37

4. TYRIMO DALIS	40
4. 1. Įdiegimas	40
4. 2. Funkcionalumas ir Sudėtingumas.....	41
4. 3. Duomenų valdymas.....	44
4. 4. Išplėtimo galimybės.....	45
4. 5. Išvados.....	46
5. Eksperimentinė dalis.....	47
5. 1. Vartotojo krepšelio dydžio eksperimentas	47
5. 2. „Zend“ bendravimo su duomenų bazė modulio eliminavimas	48
5. 3. Eksperimentų išvados.....	50
Išvados	51
LITERATŪROS SĄRAŠAS	53

SUMMARY

Term WIKI defines ideas and basis for data collection and management, necessary and mandatory methods required to ensure quality of collectable data.

The purpose of this master work is analysis of existing solutions of data collecting and analysis systems in order to create a new unique WIKI community based system for price and other information (related to products on the market) analysis. System target – build up user manageable knowledge base for recognition of changes in the market, fast, easy and cheapest products information withdraw.

It gives an overview of the best data collection approaches structure, techniques and logic, data analysis and collection methods (based on the system target), also proposes created model and defines means and tools, for model creation and realization.

SANTRUMPŲ IR TERMINŲ ŽODYNAS

PHP	- Kliento serverio architektūros programavimo kalba;
Zand Framework	- PHP pagrindu veikiantis programinis paketas;
MySql	- Duomenų bazės valdymo sistema;
Apache	- Tinklo HTTP serveris;
WEB	- žiniatinklio sutrumpinimas (kitaip <i>www</i> – angl. „ <i>World Wide Web</i> “);
WIKI	- Terminas apibrėžiantis duomenų surinkimo ir valdymo logiką ir idėjas;
XML	- Struktūrizuotas duomenų saugojimo ir perdavimo būdas;
HTTP	- hiper teksto kalba, skirta internetinių puslapių atvaizdavimui.

1. ĮVADAS

Duomenų „surinkimas“ – tai procesai, skirti tam tikros informacijos, apie dominančią objektinę sritį, kaupimui. Tokių duomenų „apdorojimo“ sąvoka apima jų filtravimą, priežiūrą, saugojimą, analizę ir pan. Kriterijai pagal kuriuos duomenys yra renkami, taip pat metodika ir filtravimas juos padaro specifiniais arba „paskirstytais“.

Paskirstytų duomenų surinkimo ir apdorojimo sąvoka yra gana plati ir apima ne tik elementariausius Web klausimynus ar testus, bet ir dideles sistemas, skirtas analizuoti ir apdoroti struktūrizuoto (pvz.: XML) duomenų srauto informaciją, prireikus ją konvertuoti, saugoti ir adaptuoti.

Sisteminis duomenų surinkimas leidžia nustatyti ne vien tik dominančios srities (objekto, tikslo ar padėties) esamą būklę, bet taip pat išryškina (aplinkos) ją įtakančius faktorius, yra pagalbinis įrankis stebėjimo procesams, taip pat sukuria tvirtą žinių (statistinę) bazę įvairių aspektų analizei atlikti. Reikia pažymėti, kad analizė ir surinktų duomenų interpretacija visuose nagrinėtose atvejuose pateikiama kaip pagrindinis sistemos veiklos tikslas, todėl nuo duomenų surinkimo yra neatsiejamas. Magistriniame darbe duomenų surinkimo metodika buvo taikyta ir nagrinėta informacijos analizės kontekste.

Informacijos surinkimas internetu, skirtas analizei ar žinių bazės sudarymui, įgauna vis didesnę reikšmę. Pagrindinis tokių sistemų tikslas - pateikti (sugeneruoti) instrumentus ir priemones centralizuotam paskirstytų duomenų surinkimui, apdorojimui ir saugojimui. Tokių įrankių įvairovė, infrastruktūra bei duomenų srauto intensyvumas iškelia papildomus reikalavimus ar sąlygas ir taip dar labiau komplikuoja sprendžiamą uždavinį.

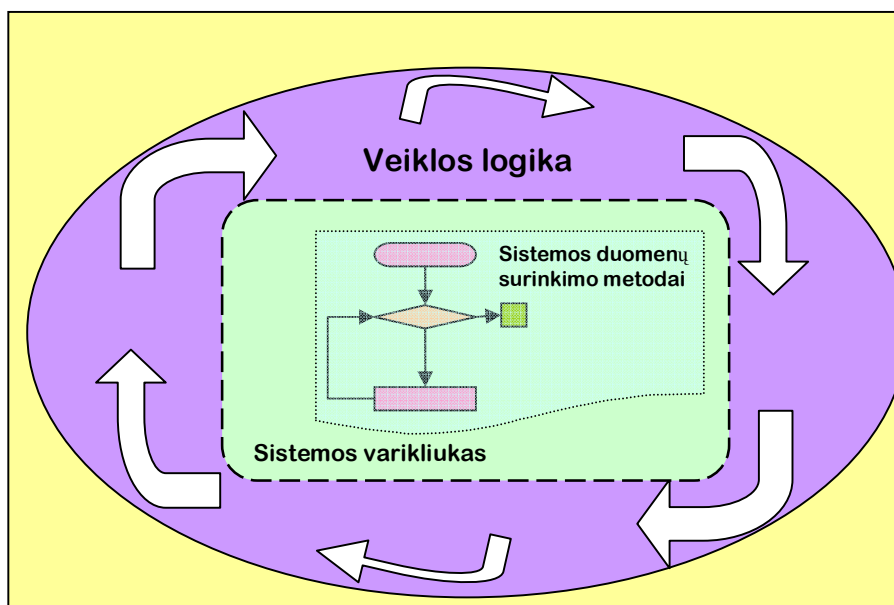
Tipinės uždavinio sprendimo problemos:

- informacijos surinkimo būdai, metodai ir priemonės;
- surenkamos (Web) informacijos kiekis;
- surinktos informacijos tinkamumas, jos įvertinimas nuo bereikšmės iki labai svarbios;
- informacijos įvairumas ir sudėtingumas – nuo paprasto teksto iki struktūrizuotų dokumentų (pvz.: XML) ar daugialypių (angl. *Multimedia*) bylų [2].

Reikia apibrėžti tris pagrindinius analizuojamų sistemų lygius (žr. pav. 1):

1. Sistemos veikimo logika (modelis) – nusako kaip sistema turėtų veikti, kokių taisyklių ir normų laikytis, tačiau neapibrėžia priemonių tikslui pasiekti;

2. Sistemos veikimo schema (varikliukas) – fizinis sistemos schemas įgyvendinimas: platforma, programavimo kalba, duomenų bazė ir .t.t. Reikia pažymėti, kad šios priemonės gali turėti savo specifinę veikimo (logikos) schemą;
3. Sistemos naudojami duomenų surinkimo, valdymo ir analizės metodai (funkcijos) – metodų realizavimas programavimo kalba, sistemos vartotojų tikslams pasiekti;



1 pav. Trys sistemų analizės lygiai

Sistemos įgyvendinimui buvo pasirinkta web aplinka veikianti kliento serverio architektūros pagrindu. Darbo organizavimui, bei duomenų srautų valdymui – „wiki“ idėjos. Duomenų surinkimas pagrįstas bendruomene. Analizės metodai unikalūs kiekvienam vartotojui.

Realiai veikiančios sistemos paprastai skirtos specifiniams tikslams įgyvendinti – tai yra stengiasi atsakyti į konkrečius ir dažniausiai skirtingus klausimus. Analoginės, tai yra sprendžiančios tą patį uždavinį sistemos, yra panašios, tačiau turi ir savitų (unikalių) atributų ar įrankių. Populiariausios bendruomene pagrįstos sistemos (pvz.: MS Sharepoint-Wiki, Wikia, LIKI) veikia „wiki“ principais, varomos taip vadinamais „wiki varikliukais“ (angl. „Wiki engines“).

Lyginant sukurtąją sistemą su kitomis [3] [4] [5] [6], galima išskirti tokius privalumus: sistema neapsiriboja vien tik informacijos surinkimu, saugojimu ar analize. Dėmesys skiriamas vartotojo darbui sistemoje, analizuojant jo pasirinkimus, atliekamas operacijas ir pan., siekiant

personalizuoti (suasmeninti) aplinką – pavyzdžiui, sudaryti vartotojo labiausiai dominančių produktų *top10*. Nors WIKI pagrindu veikiančių sistemų yra daug ir įvairių, analogo Lietuvoje sukurta nebuvo, arba jis oficialiai neskelbiamas.

2. TECHNOLOGIJŲ IR PROGRAMŲ SISTEMŲ APŽVALGA

2.1. Tyrimo sritis, objektas ir problema

Tyrimo sritis – duomenų surinkimo ir analizės metodų (jų algoritmų) taikymas, kuriant bendruomene paremtos, „wiki“ pagrindu veikiančios, žinių bazės sistemą. Pats taikymas turi daugiau organizacinę, o ne teorinę (pvz.: *Data Mining*) prasmę, tai yra siekiama tobulesnio veiklos modelio.

Terminas „wiki“ apibrėžia kur kas platesnę reikšmę nei jo tiesioginis vertimas – „greitai“. Jau dabar žodis asocijuojasi su daug vartotojų turinčia, paprasto naudojimo sistema, kurios publikuojamą informaciją lengvai ir greitai gali pakeisti (ar patobulinti) kiekvienas grupės ar bendruomenės narys. Įvėlus klaidą, pokyčiai gali būti nesunkiai atstatyti į pradinę būseną. Pagrindiniai wiki sistemų principai:

- automatinis sąsajų generavimas; (*angl.* „automated link generation“)
- visų keičiama informacija; (*angl.* „content editable by all“)
- lengvas informacijos (teksto) įkėlimas/įvedimas; (*angl.* „Easy text input“)
- atstatomumas. (*angl.* „back links“)

Vartotojų bendruomenė – tai suinteresuotų žmonių (pirkėjų) grupė, norinti turėti atnaujintą (ir beatsinaujinančią) informaciją apie esamą Lietuvos prekybą tinkluose parduodamų prekių kainų situaciją. Pagrindinis jos keliamas klausimas „kur pigiau?“.

Informacijos kokybė, esanti žinių bazėje, yra kritinis (aktualiausias) rodiklis. Svarbu suprasti ir suvokti kokios informacijos reikia, kaip ją surinkti ir analizuoti. Surinkimo priemonės negali būti suprantamos dviprasmiškai ar paviršutiniškai. Rezultatų pateikimo priemonės (bei patys rezultatai) taip pat. Tokios sistemos aktualumas ir poreikis akivaizdus – visi linkę taupyti ir išleisti mažiau.

2.2. Objektas – unikali, lanksti žinių bazės sistema

Objektas – norima sukurti unikalią žinių bazės sistemą, kuri pasižymėtų išskirtinėmis analitinėmis ir veiklos funkcijomis, būtų nesunkiai palaikoma (*angl.* *maintain*), bei lengvai ir greitai adaptuojama ar keičiama stengiantis prisitaikyti prie besikeičiančius aplinkos (rinkos padėties).

2. 2. 1. Sistemos problemos

Sprendžiamos problemos [3]:

1. Egzistuojančios sistemos komercinio pobūdžio ir dažnai neprieinamos eiliniam vartotojui, arba neteikia norimos (reikiamos) informacijos, ar jai publikuoti patogesnių įrankių ir priemonių.

Sprendimas – sukurta nemokama, specialiai kainų ir informacijai apie prekes rinkimui, analizei ir skelbimui skirta sistema. Pateikia patogius įrankių sprendimus tiek pradedantiems, tiek patyrusiems interneto technologijų vartotojams.

2. Wiki varikliukų pagrindu veikiančios sistemos informacijos atstatymui, saugojimui ar atstatymo funkcijai palaikyti naudoja papildomai kuriamus failus (versijų istoriją). Atsiranda perteklinės informacijos, nes papildomai saugojama ir išvaizdos (HTML kodo fragmentų) konfigūracijos. Taip bereikalingai švaistoma atmintis.

Sprendimas – sukurta tokia sistema, kuri atskiria tarpusavyje nesusijusias informacines dalis: sistemos reprezentacinius elementus (vartotojo sąsają), bei pateikimo turinį (informacijos vienetus). Tokiu būdu duomenų saugojimas ir atvaizdavimas yra dinamiškai generuojamas procesas.

3. Wiki sistemose nerealizuota arba mažai išvystytas personalizuotas informacijos pateikimas, informacijos turinio, o ne išvaizdos perspektyvoje. Atitinkamų teisių vartotojas dažniausiai gauna bendrą, visiem to paties lygios vartotojam prieinamą, informaciją.

Sprendimas – sukurta informacinė žinių bazės sistema, kuri leidžia susikurti, išsaugoti ir keisti asmeninius prekių krepšelius, prireikus – analitinę informaciją ir rezultatus pateikia atsižvelgiant į juos.

2. 2. 2. Sistemos kūrimo tikslai

Kuriant žinių bazės sistemą buvo įgyvendinti šie tikslai:

1) Bendruomenės vartotojų susidomėjimo srities (individualių krepšelių) analizė, pagal kurią generuojama tik jiems skirta informacija (pranešimų, pasiūlymų ar kita forma);

2) Efektyvus ir greitas didesnio informacijos kiekio nuskaitymas ir apdorojimas (XML forma);

3) Sudarytas efektyvus mechanizmas esamos padėties, bei jos pokyčių sekimui ir analizavimui.

Realizuotais tikslais pasiekta:

- Efektyvus pranešimas apie pasikeitusią situaciją (pvz.: parduotuvių tinkle nauja akcija);
- Vartotojų individualaus krepšelio analizė;
- Struktūrizuotos informacijos nuskaitymas ir atvaizdavimas;
- Įdiegtas patogus ir paprasto naudojimo įrankis prekių kainų analizei atlikti.

2. 2. 3. Duomenų surinkimo sistemų veiklos modeliai

Informacijos sistemos dėl savo tikslų ir veiklos pobūdžio yra labai įvairios ir skirtingos, veikia skirtingose aplinkose. Kai kurios yra pakankamai sudėtingos analizuoti, nes turi didelį vartotojų skaičių, ar įgyvendintos specifinėse (kartais kritinėse) aplinkose.

Pagrindinės dvi tokių sistemų rūšys, skirstomos pagal veikimo aplinką:

- 1) web aplinkos aplikacijos;
- 2) „savarankiškos“ (angl. Stand Alone) aplikacijos.

Web aplikacijos – vis labiau populiarėjančios, dažniausiai šabloninių varikliukų veikimu pagrįstos, tradicinėje kliento serverio aplinkoje veikiančios sistemos.

Savarankiškos sistemos - paprastai senesnės, analogų neturinčios unikalios sistemos (pvz.: SAP), savo veikimo principais bei greičiu vis dar pranokstančios naujais išleistas sistemas. Kitas svarbus faktorius yra šių sistemų vartotojai – kompanijos milžinės, kurioms įsidiesti naują aplinką gali būti didelis finansinis sunkumas.

Iš esmės, pagal savo veikimo pobūdį, duomenų surinkimo sistemas galima suskirstyti į tris pagrindines rūšis:

- 1) Interaktyvios sistemos; (angl. „Interactive system“)
- 2) Siuntos sistemos; (angl. „Batch system“)
- 3) Hibridinės (mišrios sistemos). (angl. „Hybrid system“) [8]

2.2.3.1. Interaktyvios sistemos

Sistemą sudaro vienas ar daugiau centralizuotai veikiančių kompiuterių (serverių), iš kur yra valdomi vartotojų įvedamų duomenų rinkimas ir patvirtinimas. Šios sistemos turi keletą pranašumų lyginant su „siuntos“ sistemomis:

- Skubi duomenų peržiūra (*angl.* „Immediate data verifications“) – kaip ir kokius duomenis įveda vartotojas, koks šių duomenų pagrįstumas ir pan. Vienas svarbiausių aspektų vertinant būsimos sistemos saugumą.
- Klaidų sumažinimas (*angl.* „Error reduction“) – palaikomas glaudus ryšys tarp sistemos ir vartotojo. Įvykus klaidai apie tai iškart gali pranešti ir taip išvengti galimo informacijos praradimo ar sugadinimo.
- Tokias sistemas paprasčiau (lengviau) suplanuoti ir suprojektuoti. Nereikalauja papildomų suderinimų ar tyrimų, pavyzdžiui, kada tinklo apkrova mažiausia „siuntai“ išsiųsti, arba ar tam tikrą valandą nutolęs serveris yra pasiekiamas.

2.2.3.2. „Siuntos“ („batch“) sistemos

Duomenų surinkimo paketais, arba siuntom, sistemos. Paskirstytai ir pasvertai naudoja sistemos serverio resursus veiklai atlikti. Sistemos viduje budėjimo režimu veikia vienas ar keli procesai, kurie esant tam tikrom sąlygom aktyvuojami. Surinkus duomenis visas paketas registruojamas (išsaugomas) duomenų bazėje. Paprastai tokios sistemos naudojamos statistiniams duomenims rinkti ir kaupti, jos dažniausiai būna apibrėžtos iš anksto. Tokių sistemų privalumai:

- Vartotojo klaidų eliminavimas – kadangi procesai visiškai automatizuoti, iš duomenų įvedimo proceso visiškai pašalinamas eilinis vartotojas. Vis dėlto tai neapsaugo nuo procesų klaidų (pvz.: procesų kolizijos) – tokiu atveju klaidos ištaisymas ir reakcija į ją pailgėja.
- Ekonomiškumas – sistema atsakingai naudoja serverio resursus. Aktyvavimo ir veikimo sąlygos paprastai būna iš anksto apibrėžtos – „aktyvuotis esant mažiausiam tinklo aprovimui – 3 nakties“.

2.2.3.3. Hibridinės sistemos

Tokios sistemos - tai dažniausiai pasitaikantis realiai veikiantis modelis, savyje apjungiantis tiek interaktyvias sistemas, tiek „siuntas“. Paprastai jos veikia kaip daugiafunkcinės į vartotoją orientuotos aplinkos, tam tikru metu persijungiančios į „siuntos“ sistemų režimą.

2. 3. „Wiki“ technologijos

Šiandien žodžiu „wiki“ nusakoma atviro kodo programinė įranga (*angl.* „Open Source“), su atitinkama licenzija. Pagrindinė sistemos mintis – kiekvienas, bet kur ir bet kada, gali pridėti naują informaciją, jau esamą keisti ar ištrinti.

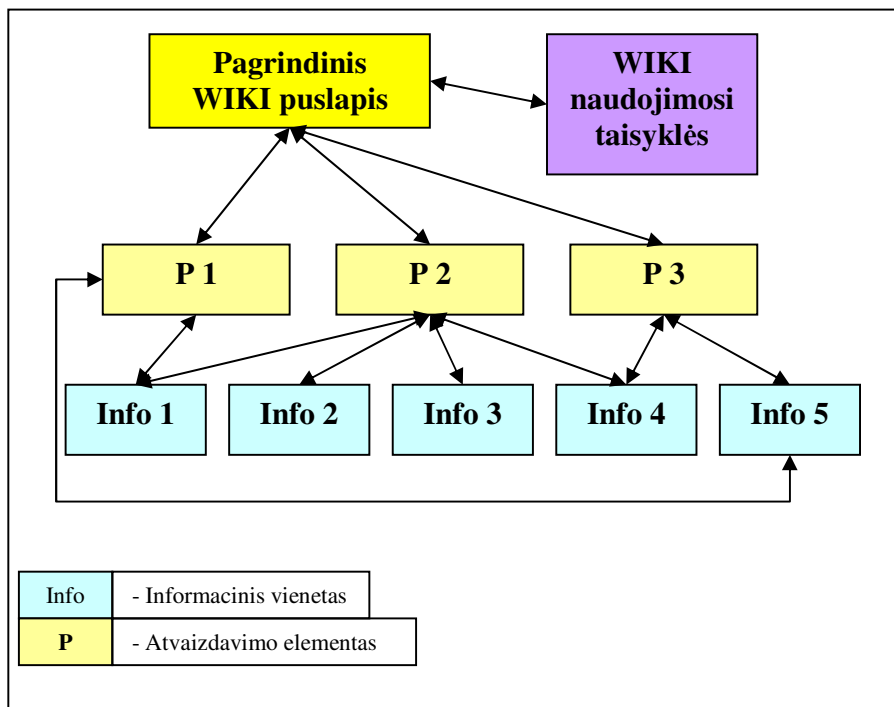
Tai - vartotojui draugiškos aplinkos, pagalbinių įrankių, sistema, informacijos saugojimui kurianti eilę web puslapių, tarpusavyje juos automatiškai susiedama nuorodomis. Viso veikimo pagrindas – XML schemas, atskiriančios turinio elementus nuo formatavimo elementų.

Sistemos sėkmės pagrindas – pasiekiamumas kiekvienam vartotojui. Tai yra patys vartotojai toliau plėtoja sistemą informaciniu požiūriu, siekiant ją išplėsti ir padidinti. [14]

Magistriniam darbui buvo pasirinktos „wiki“ technologijos idėjos kaip veiklos logikos schema – pagrindinė loginė terpė kurioje realizuojami sistemos procesai ir veiksmai.

2. 3. 1. „Wiki“ organizacine prasme

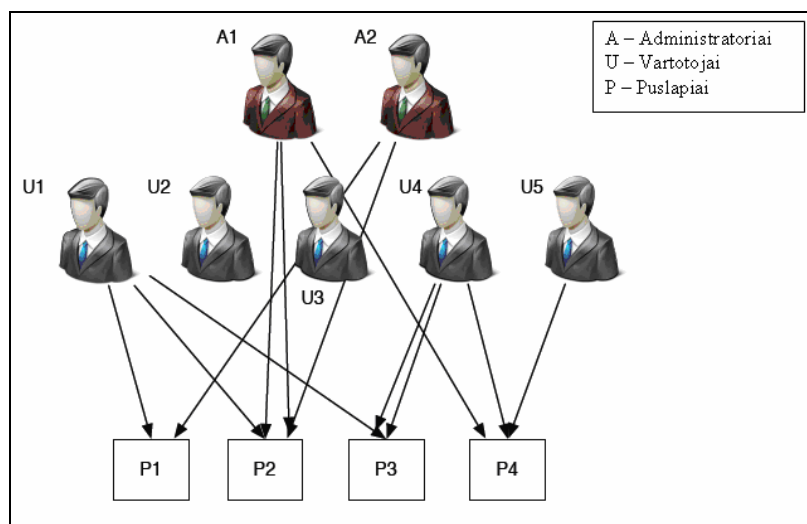
Wiki sistema nėra elektroninė knyga ir neturi tiesios (linijinės) struktūros – tai yra, ji nėra sudaryta puslapis po puslapio. Tai daugiau tarpusavio nuorodų (*angl.* links) struktūrizuotas (ir prasmingas) rinkinys [15] [16] (*žr. pav. 2*).



2pav. Sąlyginis WIKI struktūros grafinis atvaizdavimas

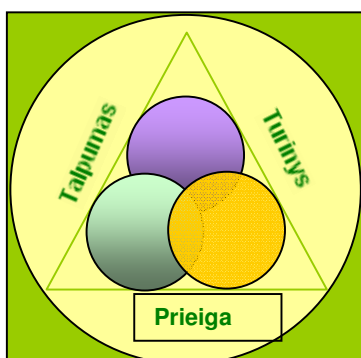
Elementus „P“ ir „info“ kuria vartotojai, tačiau naudojimosi taisykles ir pagrindinį puslapį formuoja administratoriai.

Apskritai WIKI sistemos turi ne mažesnę nei dviejų lygių vartotojų teisių paveldimąjį modelį, tai yra kiekvienas aukštesnis lygis paveldi žemiau esančiojo teisią ir galimybes, bei įgauna naujų (žemesniajam nepasiekiamų funkcijų) [15] (žr. pav. 3).



3 pav. WIKI sistemų vartotojų teisių paveldimasis modelis

Kadangi informacinė sistema periodiškai plečiasi, vienas svarbiausių sistemos palaikymo tikslų – uždavinio „Talpus“, „Turinys“, „Prieiga“ – sprendimas (žr. pav. 4). Daugiau informacijos reiškia daugiau vartotojų (koks maksimalus skaičius?) - reikia iš pakankamai didelio informacijos kiekio išskirti kokybiškiausią, pagal tai stengtis prognozuoti sistemos plėtimą (kiek atminties reikė ateityje?). Tikslui pasiekti naudojami vadinamieji „Data Mining“ algoritmai situacijos prognozavimui, taip pat papildomi analitiniai metodai [14] [15] [16].



4 pav. WIKI sistemų sprendžiamas uždavinys

Magistrinio projekto realizavimui buvo pasirinkta WIKI kaip veiklos loginė schema, bei įgyvendintos visos pagrindinės idėjos.

2. 4. Duomenų surinkimo ir valdymo metodai

Duomenų surinkimas neapsiriboja vien tik apibrėžtos formos informacijos saugojimu į, bei ištraukimu iš duombazės. Čia sukurtai sistemai galima pritaikyti ir bendrąsias informacinių ar žinių bazių sistemų nuostatas ir normas. Stengiamasi išlaikyti sistemos naudingumo ir naudojamumo vertę, tai yra periodiškai atnaujinti sistemą informacine prasme. Geri informacijos surinkimo ir valdymo metodai svarbūs kuriant santykius su naujais ir palaikant arba gilinant juos su senais klientais ar vartotojais.

2. 4. 1. Pagrindiniai principai

Pagrindinė užduotis – apklausų pavidalu (*angl.* questioners) surinkti kiek galima daugiau naudingos informacijos. Pagrindiniai principai, kuriais reikėtų vadovautis norint sukurti web duomenų surinkimo mechanizmą:

1. Suvokiamumas – svarbu pasverti kokio sudėtingumo bus informacijos surinkimo forma, kiek ir kokie klausimai ar laukai bus pateikti užpildymui, koks bus bendras informacinis vaizdas. Ar pateikiama forma nesudaro perdėtai komplikuoto išpūdžio.

2. Trumpumas (*angl.* „keep it short“) – kitas svarbus aspektas yra kokie klausimai bus pateikti. Vietoj to, kad konstruoti formą su informacija, kurią norėtusi turėti, reikėtų nustatyti kokia informacija yra reikalinga vartotojui. Tokiu būdu galima suformuoti pirmuosius informacijos kokybės vertinimo filtrus.

3. Paprastumas (*angl.* „keep it simple“) – svarbu naudoti kiek galima daugiau realių formų atsakymo variantų, tai yra kaip į formas tariamai galėtų atsakyti respondentas. Kitaip tariant, užpildyt labiausiai tikėtinas (būtinąsias) reikšmes (pvz.: lytis, vardas, miestas ir t.t.), pateikiant galimybę jas keisti ir koreguoti.

4. Tvarkingumas (*angl.* „keep it clean“) – reikia numatyti galimybę aptikti ir pašalinti „šiukšles“ iš sistemos. Pavyzdžiui vartotojas pateikdamas asmeninę informaciją, vietoj vardo įveda beprasmi simbolio derinį „alkhdfkhsdf“ ir taip sėkmingai praeina registracijos procedūrą. Nereikalingi ar beverčiai įrašai ne tik užima sistemos atmintį, bet ir neverti visų pasekos pastangų. [10]

„Accela Communication“ išskirti duomenų surinkimo ir valdymo metodus:

- Išankstinis internetinių formų užpildymas. Vartotojui grįžus į svetainę jis tikisi būti atpažintas ir sugražintas į savo personalizuotą aplinką, informacine bei dizaino ar išvaizdos prasme. Ypatingai svarbu, jei atliekami veiksmai yra pasikartojantys.

- Pasyvi registracija. Šis modelis labiau derinamas su „siuntos“ ar hibridinėmis sistemomis turinčiomis išorinius susisiekimo agentus ar klientus. Dalis informacijos yra perimama iš jų, todėl vartotojas gali praleisti tam tikrus informacijos suvedimo niuansus.

- Inkrementinis duomenų registravimas. Tai vartotojo profilio konstravimas stebint jo veiksmus. Tai yra visą laiką įsimenant ką vartotojas darė, kuo labiausiai domėjosi, ko ieškojo, kokias operacijas atliko ir pan. Tokiu būdu per laiką vartotojui pateikiamas visiškai personalizuotas informacinė aplinka.

- Multi-formų registracija. Kaip ir inkrementinė, užtrunka tam tikrą laiką – periodiškai vartotojui pateikiami vienas ar keli klausimai, siekiant gauti atsakomąjį ryšį. Efektyvus būdas rinkti statistikai, nevarginant vartotojų ilgomis ir nuobodžiomis apklausomis. [9]

Išskyrus „pasyvią registraciją“, magistrinio projekto realizavime buvo įgyvendinti visi duomenų surinkimo ir valdymo metodai.

2. 5. Duomenų analizės metodai

Kalbant apie tradicinę kliento serverio architektūrą, yra du pagrindiniai duomenų analizavimo požūriai:

- Aktyvus, arba „staigos reakcijos“;
- Pasyvus, arba statistinis. [11] [12]

Nors ir išskirtos dvi kategorijos, abu požūriai priklauso nuo sistemos įvykių (*angl.* „event driven analysis“), tai yra kažkas turėjo nutikti ir tai turėjo būti užfiksuota. Kaip bus reaguojama į „tai“ (veiksmą ar įvykį sistemoje) iš esmės ir apibrėžia pagrindinį analizavimo požūrių skirtumą.

2. 5. 1. Aktyvi analizė

Paprastai susijus su sistemos saugumo ir informacijos klasifikavimo ar prieigos klausimais. Geriausias pavyzdys – apsauga nuo „burtualios jėgos“ (*angl.* „Brute Force“) įsilaužimų: vartotojui kelis kartus įvedus neteisingą slaptažodį ir prisijungimo vardą, sistema tikslingai daro kelių sekundžių prieigos ribojimą šiam vartotojui.

2. 5. 2. Pasyvi analizė

Atliekama bet kuriuo metu ir nereikalauja žaibiškos reakcijos į išorinius įvykiu. Paprastai susideda iš kelių vartotojų poreikiams pritaikytų funkcijų, naudojančių matematinius bei statistinius metodus.

Realizuotoje sistemoje naudojami tiek aktyvios analizės (apsaugai nuo „brutualios jėgos“ įsilaužimų), tiek pasyvios analizės metodai.

2. 6. Egzistuojančių sistemų apžvalga

Pagal paskirtį WIKI sistemos skirstomos į:

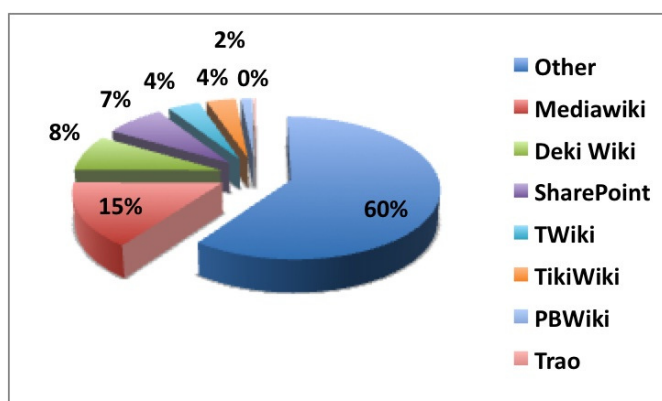
1. Atviro kodo paprastąsias (angl. „open source“);
2. Atviro kodo virtualios mokymo aplinkos (angl. VLE – Virtual Learning Enviroment).

Keli tokių sistemų pavyzdžiai (žr. pav. 5):

<p>„Paprastosios“:</p> <ul style="list-style-type: none">• Mediawiki• MoinMoin• Kwiki• TikiWiki• PHPWiki	<p>„VLE“:</p> <ul style="list-style-type: none">• Moodle• Bodington• Claroline• ATutor• Sakai
--	---

5 pav. Egzistuojančių sistemų pavyzdžiai

Aukščiau pateikti tik keli jau egzistuojančių sistemų pavyzdžiai. Šiuo metu priskaičiuota daugiau kaip 50 oficialiai skelbiamų WIKI sistemų pavadinimų. Visos jos atviro kodo, o tai reiškia, kad gali būti adaptuojamos kiekvieno vartotojo poreikiams. Remiantis 2008 metais „Open Source Marketing“ atliktais tyrimais, didesnioji naudojamų WIKI sistemų jų rinkoje nežinomos arba oficialiai neskelbiamos [17] [23] (žr. pav. 6):



6 pav. Atviro kodo sistemų pasiskirstymas

Taip pat egzistuoja mažiau paplitę komerciniai produktai, tokie kaip, pavyzdžiui, „Microsoft Sharepoint-Wiki“. [17]

2. 6. 1. Sistemų vertinimo kriterijai

Apžvalgai pasirinktos dvi sistemos – viena atviro kodo „VLE“ nemokama („Moodle“) sistema, kita komercinė uždaro kodo („Microsoft Sharepoint-Wiki“) sistema. Buvo taikomi tokie pagrindiniai kriterijai analizuojant ir vertinant sistemas:

1. Įdiegimas – kokios sistemos įdiegimo sąlygos;
2. Funkcionalumas – kokį funkcionalumą siūlo (pobūdis);
3. Sudėtingumas – kiek sudėtinga yra perprasti sistemą;
4. Duomenų valdymas – kaip sistema organizuota informacijos saugojimo, surinkimo ir valdymo prasme;
5. Išplėtimo galimybės – kiek sudėtinga papildyti sistemą nauju funkcionalumu.

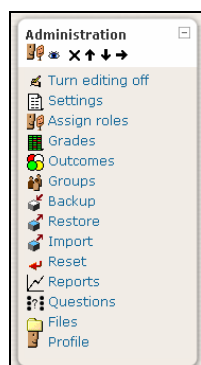
2. 6. 2. „VLE“ Moodle sistema

Moodle yra programinės įrangos paketas, skirtas internetinių mokymo kursų svetainių kūrimui. Tai socialiniu konstravimu paremta aplinka, palaikanti savotišką mokymo (*angl.* education) veiklos organizavimo šabloną. Pati sistema turi kur kas platesnį panaudojimo spektrą nei elementarus informacijos skelbimas ar redagavimas – tai struktūrizuotas procesų organizavimas, turintis sudėtingesnę vartotojų, teisių, bei informacijos prieigos realizavimo mechanizmą. [18]

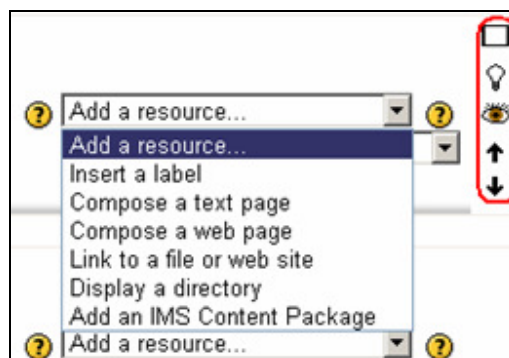
1. Įdiegimas. Sistemos įdiegimui reikalinga papildoma programinė įranga – funkcionuojantis web serveris palaikanti php programavimo kalbą, bei duomenų bazė. Sistemos

kūrėjai rekomenduoja naudoti taip pat nemokamus, išvardintus funkcionalumus palaikančius, produktus: APACHE, MYSQL, bei PHP. [18] [20]

2. Funkcionalumas. Pateikiama visa eilė gana gerai organizuotų įrankių skelbiamos informacijos atvaizdavimui, bei ir aplinko personalizavimui (žr. pav. 7, 8).



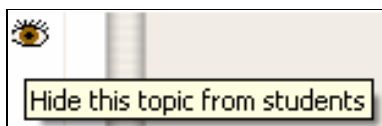
7 pav. Personalizavimo įrankiai



8 pav. Informacijos skelbimo įrankiai

Dešiniajame kampe esanti meniu lengvai leidžia suteikti publikuojamai informacijai papildomus atributus (pvz.: palėpti, rodyt tik antraštę ir pan.).

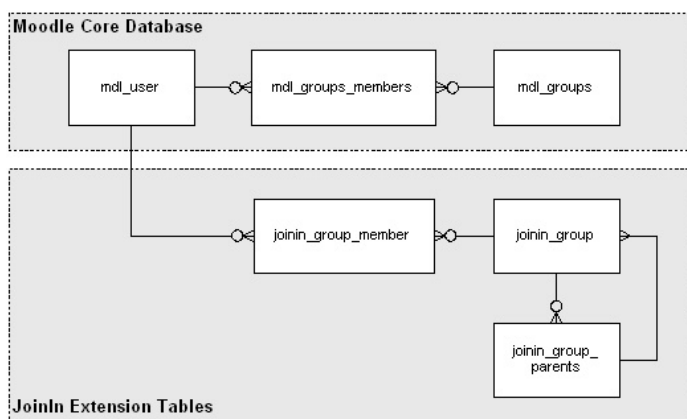
3. Sudėtingumas. Aplinka draugiška vartotojui – atliekant bet kokią veiksmą gali rasti paaiškinimą koks bus rezultatas (žr. pav. 9).



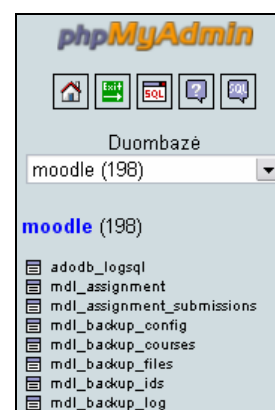
9 pav. Funkcijos rezultato paaiškinimas

Dominuoja „look & feel“ principai – daugelis funkcijų nuspėjama, sistema elgiasi taip kaip tikimasi ir nepateikia netikėto rezultato. Yra papildoma galimybė uždrausti arba aktyvuoti vartotojų registravimo procesą.

Galima nesunkiai keisti atvaizdavimo elementų išdėstymo vietą, pakeisti spalvas ir pan. Tačiau sudėtinga yra pakeisti pačius elementus. Viena to priežasčių - sudėtinga duomenų bazės struktūra (198-uonios lentelės) (žr. pav. 10, 11).[22]

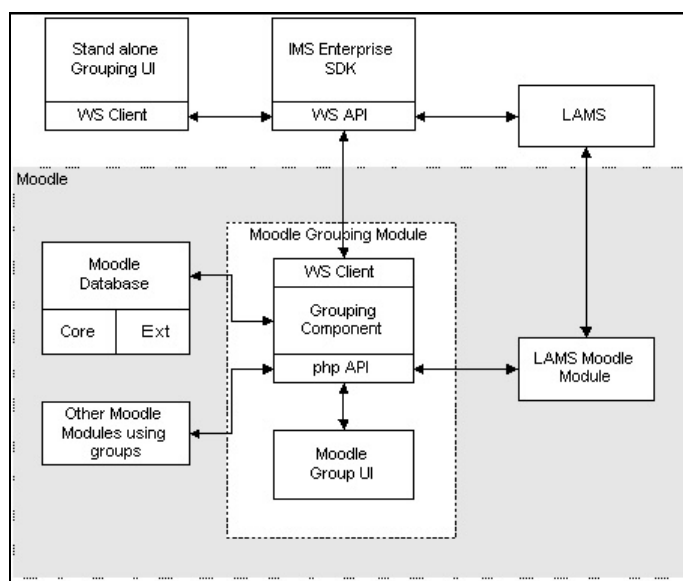


10 pav. Principinė MOODLE duomenų bazės struktūra



11 pav. Moodle duomenų bazės lentelių vaizdas

4. Duomenų valdymas – duomenys saugomi struktūrizuotai – išskaidomi. Kiekvieną kartą vaizdas generuojamas iš sudėtinių elementų, remianti duomenų bazės įrašais. Vaizdiniai bei informaciniai elementai saugomi atskirai (žr. pav. 12). Procesas gana sudėtingas. [22]



12 pav. MOODLE architektūros diagrama

5. Išplėtimo galimybės – kadangi sistema atviro kodo, nesudėtingai galima keisti, papildyti ar pridėti naują funkcionalumą. Yra sukurta atskira sistema klaidoms registruoti, analizuoti ir priimamiems sprendimams sekti. Egzistuoja atskira dokumentacija, kaip atskiri patarimai vartotojui, kaip ir kokių taisyklių reikėtų laikytis plėtojant „Moodle“ sistemą. [21]

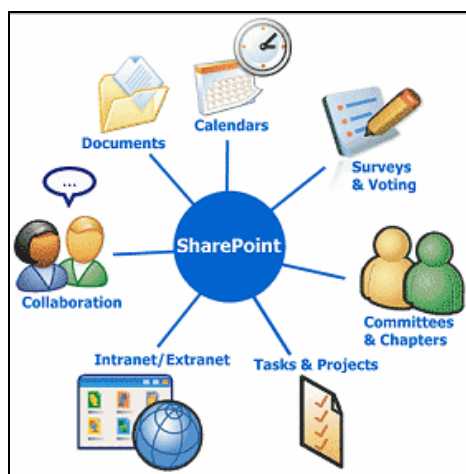
2. 6. 3. Išvados apie „Moodle“

Sistema gali būti įdiegta kaip atskiras programinis paketas, tačiau reikalauja išankstinių funkcionalumui palaikyti sąlygų – veikiančio WEB serverio palaikančio PHP kalbą, bei surišto su duomenų baze. Realizuota draugiška vartotojui aplinka - informacijos rinkimui, formatavimui, skelbimui, bei aplinkos personalizavimui pateikiami patogūs vartotojo sąsajos įrankiai.

Informacijos surinkimo, saugojimo bei atvaizdavimo procesai gana sudėtingi ir reikalauja nemažai laiko nuodugnesniam sistemos tyrimui ir perpratimui. Sistema yra tikslinga skirta virtualaus mokymo procesams realizuoti.

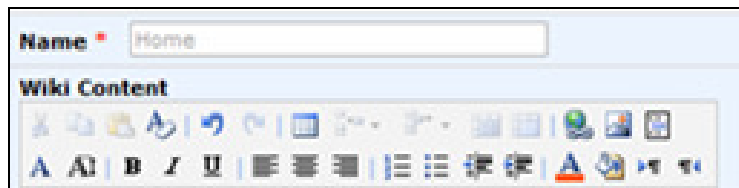
2. 6. 4. „Microsoft Sharepoint-Wiki“ sistema

Reikia pažymėt, kad sistema yra galingas informacijos valdymo ir skelbimo internete įrankis, o WIKI yra tik viena iš papildomų nebūtinų funkcijų (žr. pav. 13). Pagrindinis šios sistemos skirtumas nuo tradicinių WIKI – tai komercinis uždaro kodo produktas. [5] [6] [19]



13 pav. Microsoft Sharepoint veiklos

1. Įdiegimas - Pats įrankis privalo būti diegiamas papildomai, kaip „Sharepoint“ plėtinys, tačiau pati „Sharepoint“ sistema reikalauja eilės kitų „Microsoft“ produktų sėkmingam funkcionavimui (pvz.: „Windows Server 2003“, „ASP.Net“).
2. Funkcionalumas. Įrankis skirtas paprastam informacijos skelbimui (žr. pav. 14).



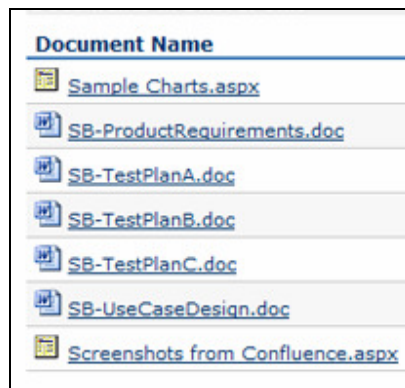
14 pav. Teksto redagavimo įrankiai

Nors vartotojui pateikiama įrankių panelė teksto formatavimui, kai kurios funkcijos labai nepatogios. Pavyzdžiui, paveikslėlių įkėlimas: iš pradžių paveikslėlį reikia įsikelti į serverį (*angl.* „upload“), tada gauti jo nuorodą ir galų gale iškvietus paveikslėlio įkėlimo dialogą, jį nurodyti. Viso proceso metu vartotojui pateikiami 3 skirtingi dialogų langai, be to vartotojas pats turi įvykdyti paveikslėlio įsikėlimą – nėra galimybės visko atlikti vieno dialogo metu.

Nėra galimybės vartotojams patiems registruotis sistemoje – jie turi būti žinomi iš anksto.

3. Sudėtingumas – teksto redaktorius paremtas „WYSIWYG“ principu (*angl.* „What You See Is What You Get“). Redagavimo metu įrašytas ar suformatuotas tekstas vienodai atvaizduojamas jį publikuojant.

4. Duomenų valdymas – kiekvienas atskiras puslapis saugomas kaip atskiras failas. Kiek puslapių, tiek failų sistemoje (*žr. pav. 15*). Kuriant didesnę informacinę sistemą, priežiūra tampa problematika.



15 pav. Sukurtų puslapių sąrašas

5. Išplėtimo galimybės – sistema uždaro kodo, todėl nėra jokių galimybių WIKI funkcionalumui plėsti. Įmanoma redaguoti pavienių failų HTML kodą, tačiau pats įrankis yra nepatogus ir nenaudingas – jame sunku orientuosis, sudėtinga navigacija.

2. 6. 5. Išvados apie „Sharepoint-WIKI“

„Microsoft Sharepoint“ yra galinga dokumentų valdymo įrankis, puikiai suderintas su kitais „Microsoft“ produktais (pvz.: „Microsoft Office“). Tačiau papildoma WIKI funkcija lyginant su atviro kodo sistemomis smarkiai nusileidžia. Įdiegiama atskirai, tačiau tik kito, didesnio paketo („Sharepoint“) kontekste. Informacijos redagavimo ir rinkimo įrankiai nepatogūs ir galėtų būti tobulinami. Nėra galimybės pridėti naujo funkcionalumo ar personalizuoti aplinkos (keisti išvaizdos). Kiekvieną kartą įdedant naują skelbiamą informacinį vieneta (išsaugojus jį sistemoje), kuriamas naujas HTML dokumentas – didelėje žinių bazėje tokių puslapių priežiūra tampa sudėtinga.

2. 7. Analizės išvados ir sprendimų pagrindimas

Pasirinktas interaktyvus duomenų surinkimo sistemos modelis – sistemai nėra tikslinga naudoti automatinius informacijos surinkimo procesus. Atsiradus būtinybei sistema nesunkiai gali būti transformuota į hibridinės sistemos modelį.

Sukurtos sistemos realizavimui pasirinkta WIKI veiklos logika. Kaip rodo kaikurie tyrimai ([15] [24]), nemokamos WIKI sistemos auga sparčiau vartotojų bei informacijos kiekio atžvilgiu. Nemokama sistema tikimasi pritraukti vartotojų, kurie sukurs sistemos duomenų bazę.

Funkcinės dalies įgyvendinimui pasirinktas „Zend Framework“ šablonų varikliukas. Jis veikia PHP pagrindu pagal MVC (angl. Model View Controller) schemą. Produktas nemokamas ir lengvai įdiegiamas.

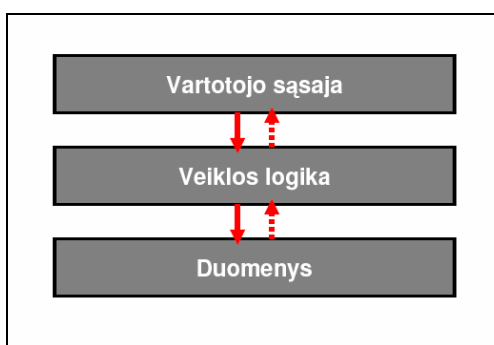
Duomenų surinkimo formoms sukurti buvo taikomi visi privalomieji ir rekomendaciniai principai: suvokiamumas, trumpumas, paprastumas, tvarkingumas. Stengtasi išlaikyti „look & feel“ principus – įdėti papildomi paaiškinimai, vartotoja sąsaja realizuota per grafinį meniu, sistema neatlieka netikėtų ar nelauktų funkcijų.

Surenkamų duomenų kokybei bei prieigai užtikrinti turi būti taikomi aktyvios ir pasyvios analizė metodai. Sukurtoje sistemoje taip pat įgyvendinti kai kurie rekomendaciniai vartotojų išlaikymo ir sudominimo (pritraukimo) metodai (pvz.: aplinkos personalizavimas, pagal dažniausiai atliekamas funkcijas ir pasirinkimus).

3. PROJEKVINĖ DALIS

Sistema skirta viešos informacinės bazės, apie rinkoje esančius produktus, sudarymui. Pagrindinė idėja – suteikti visiems laivai prieinamą (web terpėje) įrankį, leidžianti lengvai ir greitai susirasti kur dominantis produktas yra pigiausias ar kaip kito jo kainą per kažkokį periodą. Kiekvienas informaciją gali keisti, redaguoti, ar įkelti. Taip pat suteikiama galimybė susidaryti asmeninius prekių krepšelius, ar išsirinkti dominančias (konkrečias) parduotuves ir gauti pagal šios pasirinkimus orientuotus paieškos rezultatus.

Sistemos loginė architektūra kuriama vadovaujantis MVC šablonu (žr. pav. 16):

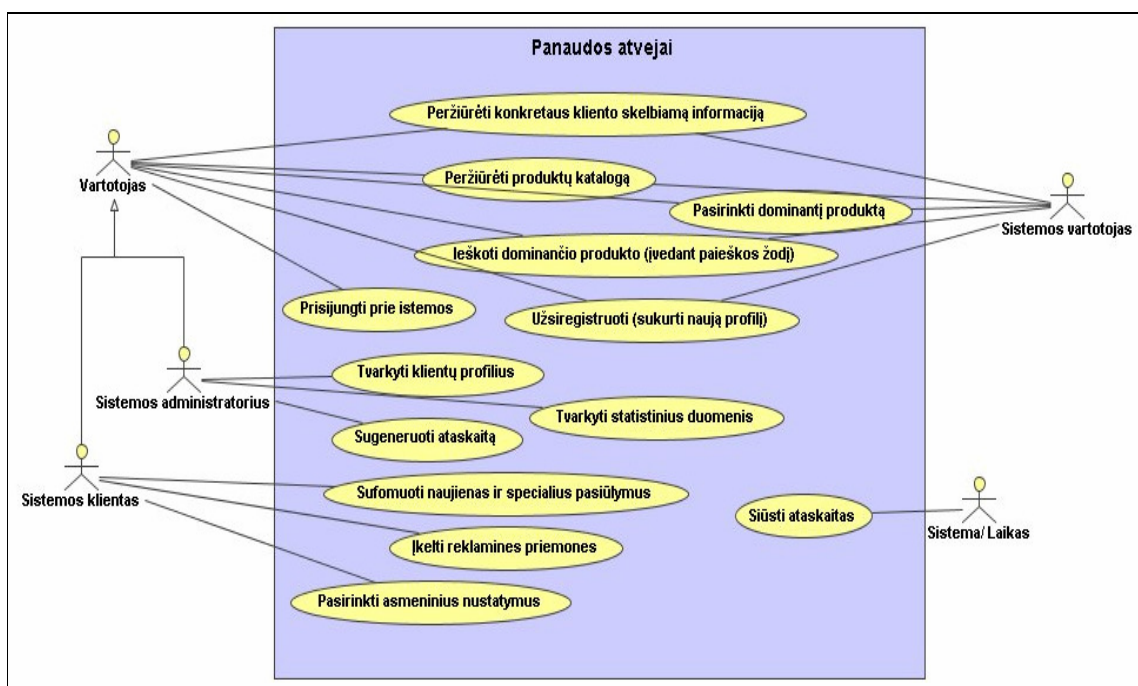


16 pav. Veiklos logikos šablono MVC principas

Kodo komponentų realizavimui pasirinktas Zend Framework šablonų varikliukas, veikiantis PHP kalbos pagrindu.

3. 1. Sistemos panaudos atvejai

Sistemos panaudos atvejai pateikti žemiau esančiame paveikslėlyje (žr. pav. 17)



17 pav. Sistemos panaudos atvejai

3. 1. 1. Sistemos administratoriaus pagrindiniai panaudojimo atvejai

1. lentelė. Klientų profilių tvarkymas

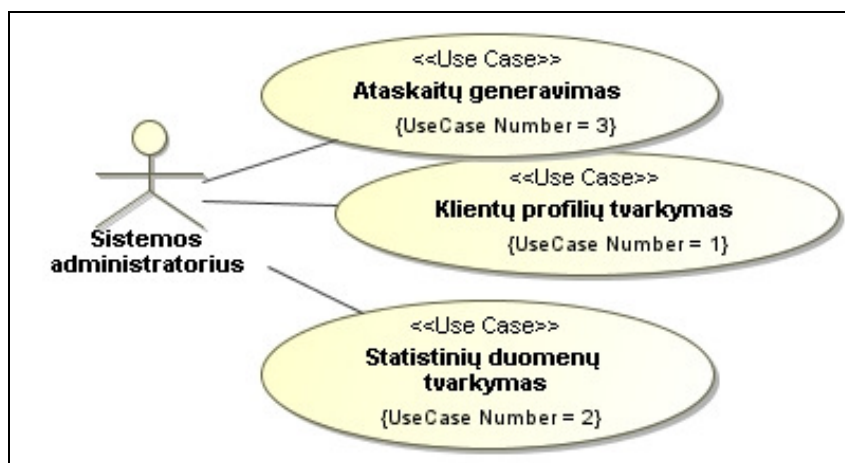
1. PANAUDOJIMO ATVEJIS: Klientų profilių tvarkymas	
Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas kurio metu sistemos administratorius teisia kliento profiliams.
Prieš sąlyga:	Klientas yra registruotas sistemoje
Sužadinimo sąlyga:	Užpildyta nauja registracijos forma
Po-sąlyga:	Naujai registruotas profilis tampa aktyvus

2. lentelė. Statistinių duomenų tvarkymas

2. PANAUDOJIMO ATVEJIS: Statistinių duomenų tvarkymas	
Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu administratorius, esant būtinybei, koreguoja statistinius apsilankymų duomenis
Prieš sąlyga:	Įvyko statistinių duomenų rinkimo klaida
Sužadinimo sąlyga:	Gautas pranešimas apie klaidą
Po-sąlyga:	Sistemoje pataisyti statistiniai duomenys
	Užregistruojama statistinių duomenų korekcija

3. PANAUDOJIMO ATVEJIS: Ataskaitų generavimas	
Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Šio proceso metu administratorius sudaro tam tikro laikotarpio vartotojų apsilankymų ataskaitas kiekvienam klientui. Sugeneravus ataskaita pasirenkamas jos išsiuntimo laikas ir data
Prieš sąlyga:	Praėjo tam tikras laiko tarpas Apsilankė bent vienas vartotojas
Sužadinimo sąlyga:	Aktyvuojama duomenų išrinkimo funkcija
Po-sąlyga:	Sistemoje sukuriama nauja, tam tikro kliento, tam tikro laikotarpio ataskaita

Sistemos administratoriaus diagrama pateikta žemiau (žr. pav. 18)



18. pav. Sistemos administratoriaus diagrama

3. 1. 2. Sistemos kliento pagrindiniai panaudojimo atvejai

4. PANAUDOJIMO ATVEJIS: Registracija (profilio sukūrimas)	
Vartotojas/Aktorius:	Sistemos klientas
Aprašas:	Proceso metu sukuriamas naujas kliento profilis
Prieš sąlyga:	Klientas neregistruotas sistemoje
Sužadinimo sąlyga:	Atsirado naujas asmuo (bendrovė) norintis reklamuoti savo produktus
Po-sąlyga:	Sistemoje sukuriamas naujas kliento profilis

5 lentelė. Asmeninio profilio nustatymų pasirinkimas

5. PANAUDOJIMO ATVEJIS: Asmeninio profilio nustatymų pasirinkimas	
Vartotojas/Aktorius:	Sistemos klientas
Aprašas:	Proceso metu klientas pasirenka savo profilio elementu išdėstymą, nuorodų pavadinimą, spalvas ir t.t.
Prieš sąlyga:	Sukurtas kliento profilis Kliento profilis patvirtintas administratoriaus Aktyvuota nustatymų pasirinkimo funkcija
Sužadinimo sąlyga:	Pasirinkta nustatymų pasirinkimų funkcija
Po-sąlyga:	Kliento profilio nustatymai išsaugoti DB

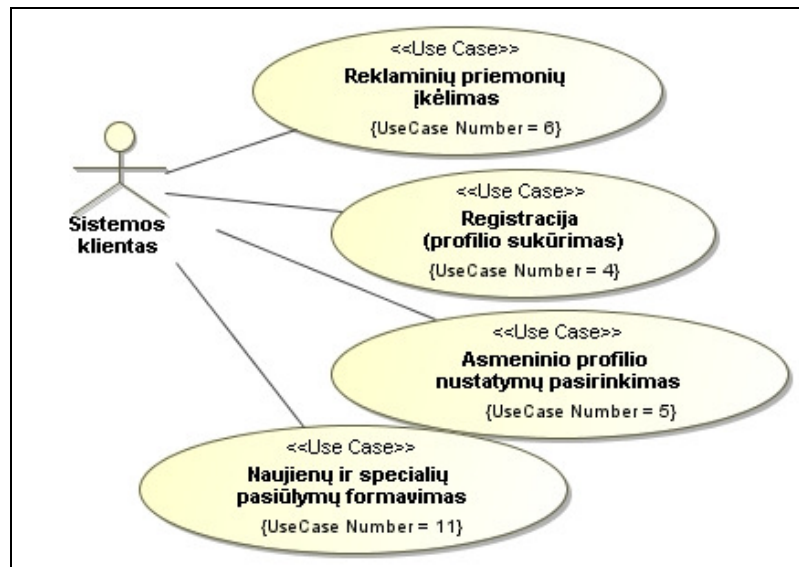
6 lentelė. Reklaminių priemonių įkėlimas

6. PANAUDOJIMO ATVEJIS: Reklaminių priemonių įkėlimas	
Vartotojas/Aktorius:	Sistemos klientas
Aprašas:	Proceso metu klientas įkelia reklaminių produktų pavadinimus, aprašymus, nuorodas, bei ikonas („baner“)
Prieš sąlyga:	Kliento profilio nustatymai pasirinkti
Sužadinimo sąlyga:	Aktyvuota informacijos įkėlimo funkcija
Po-sąlyga:	Sistemoje išsaugoma kliento produktų informacija

7 lentelė. Naujienų ir specialių pasiūlymų formavimas

11. PANAUDOJIMO ATVEJIS: Naujienų ir specialių pasiūlymų formavimas	
Vartotojas/Aktorius:	Sistemos klientas
Aprašas:	Proceso metu klientas suformuoja specialių pasiūlymų aprašus
Prieš sąlyga:	Baigtas reklaminių priemonių įkėlimo procesas
Sužadinimo sąlyga:	Pasirinkta naujienų formavimo funkcija
Po-sąlyga:	Sistemoje išsaugomi specialūs pasiūlymai galiojantys tam tikriems produktams

Sistemos kliento diagrama pateikta žemiau esančiame paveiksle (žr. pav. 19):



19 pav. Sistemos kliento diagrama

3. 1. 3. Sistemos pagrindiniai panaudojimo atvejai

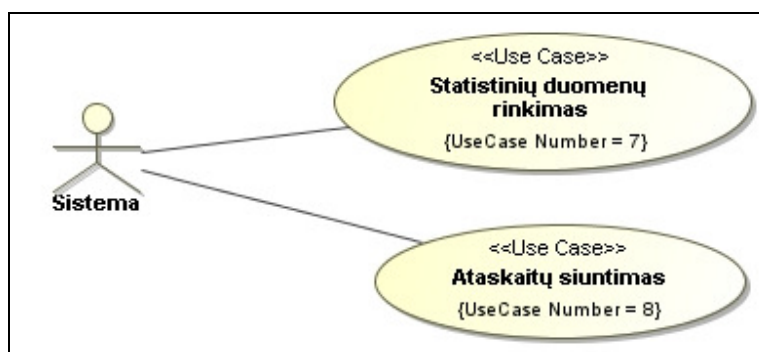
8 lentelė. Statistinių duomenų rinkimas

7. PANAUDOJIMO ATVEJIS: Statistinių duomenų rinkimas	
Vartotojas/Aktorius:	Sistema
Aprašas:	Sistema skaičiuoja kiek kartų buvo paspausta tam tikros produkto nuorodos
Prieš sąlyga:	Baigtas reklaminių priemonių įkėlimo procesas
Sužadinimo sąlyga:	Vartotojas paspaudė ant produkto nuorodos
Po-sąlyga:	Produkto aplankymas užregistruotas sistemoje

9 lentelė. Ataskaitų siuntimas

8. PANAUDOJIMO ATVEJIS: Ataskaitų siuntimas	
Vartotojas/Aktorius:	Sistema
Aprašas:	Proceso metu sistema išsiuntinėja administratoriaus suformuotas apsilankymų ataskaitas klientams
Prieš sąlyga:	Administratorius suformavo ataskaitas Ataskaitos patalpintos „išsiųsti“ direktorijoje
Sužadinimo sąlyga:	Atėjo pasirinkto išsiuntimo laikas
Po-sąlyga:	Išsiųstos ataskaitos patalpinamos „išsiųsta“ direktorijoje

Sistemos diagrama pateikta žemiau (žr. pav. 20):



20 pav. Sistemos diagrama

3. 1. 4. Sistemos vartotojo pagrindiniai panaudojimo atvejai

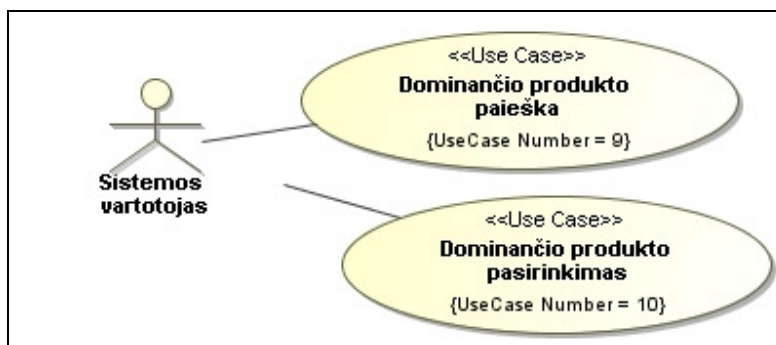
10 lentelė. Dominančio produkto paieška

9. PANAUDOJIMO ATVEJIS: Dominančio produkto paieška	
Vartotojas/Aktorius:	Sistemos vartotojas
Aprašas:	Vartotojas sukuria norimo produkto paieška sistemoje
Prieš sąlyga:	Suformuoti paieškos kriterijai
Sužadinimo sąlyga:	Iškviesta paieškos funkcija
Po-sąlyga:	Paieškos rezultatai pateikiami vartotojui

11 lentelė. Dominančio produkto pasirinkimas

10. PANAUDOJIMO ATVEJIS: Dominančio produkto pasirinkimas	
Vartotojas/Aktorius:	Sistemos vartotojas
Aprašas:	Vartotojas pasirenka prieinamiausią produktą
Prieš sąlyga:	Vartotojas atliko paiešką, arba išsirinktas produktas iš produktų katalogo
Sužadinimo sąlyga:	Vartotojas paspaudė ant produkto nuorodos
Po-sąlyga:	Vartotojas įkelia prekę į savo krepšelį

Sistemos vartotojo diagrama pateikta žemiau (žr. pav. 21):

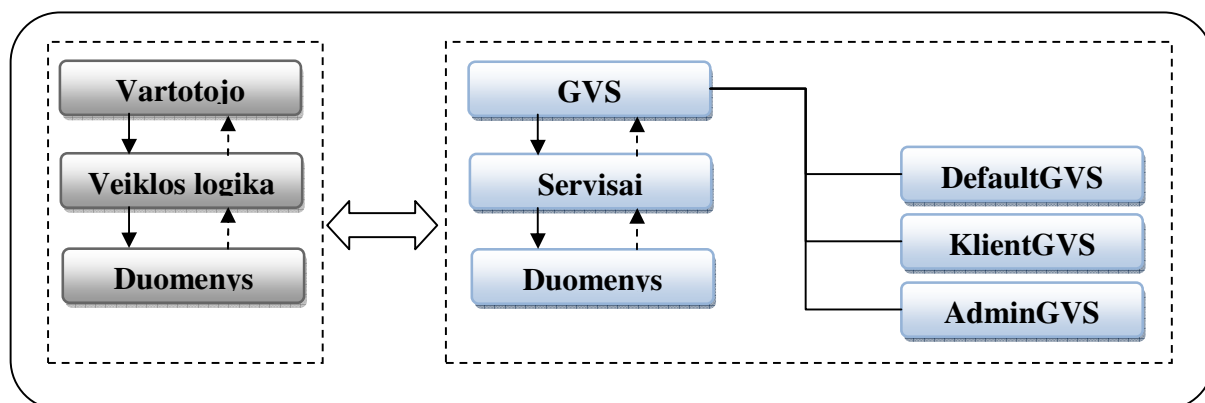


21. pav. Sistemos vartotojo diagrama

3. 2. Sistemos komponentų detalizavimas

Sistemos komponentai pagal savo paskirtį bei funkcionalumą sugrupuoti į paketus, vadovaujantis MVC (*model-view-controller*) šablonu. Apskritai visas sistemos funkcionalumas skaidomas į tris lygius: vartotojo, kliento ir administratoriaus. Visi funkcionalumai pasiekiami vartotojui, pasiekiami ir sistemos klientui, bet ne atvirkščiai. Atitinkamai visi funkcionalumai pasiekiami klientui pasiekiami ir administratoriui (bet taip pat ne atvirkščiai).

Sistemos paketų hierarchija pateikta žemiau (žr. pav. 22):

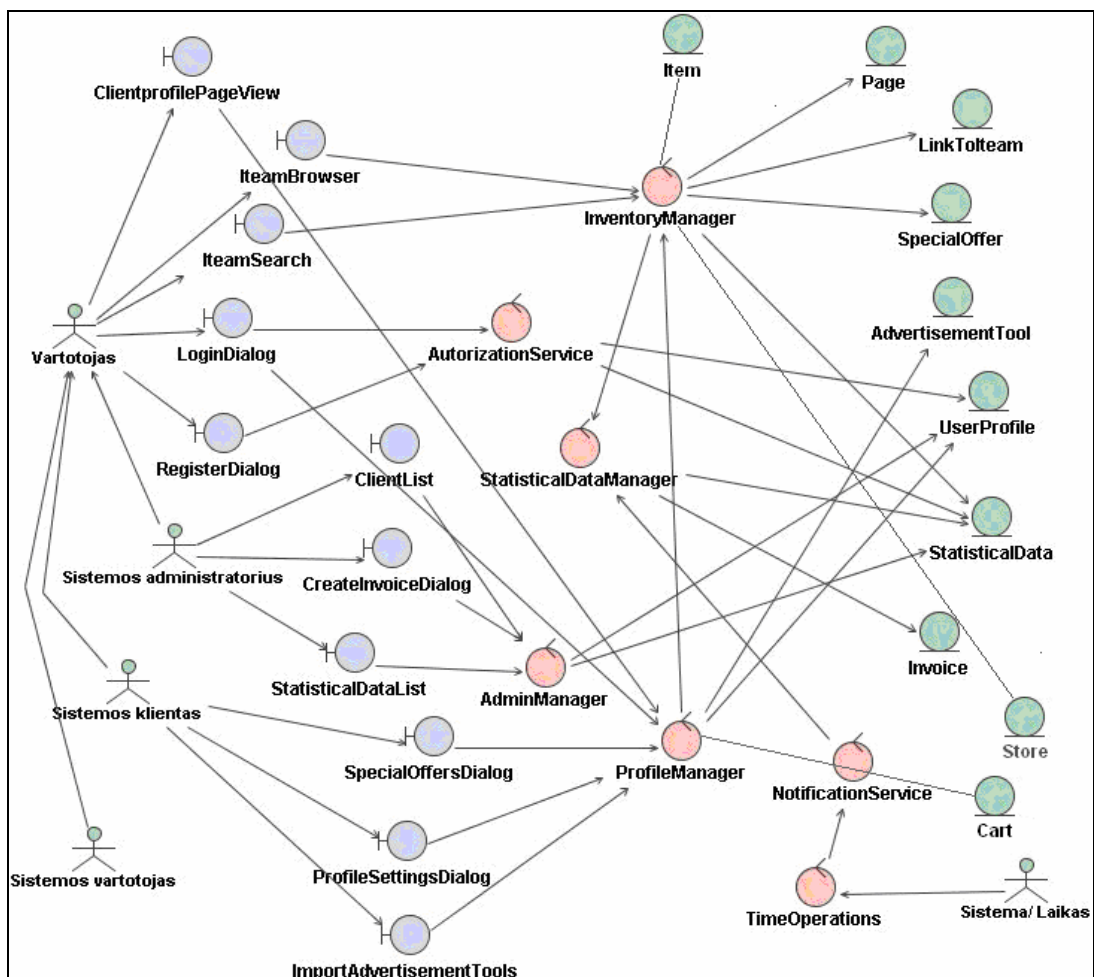


22 pav. Sistemos paketų hierarchija

Sistemos veiklos logikos diagrama, bendrojo panaudojimo kontekste, atvaizduojanti klientu funkcionalumo prieinamumą, serviso paketų bendradarbiavimą bei duomenų sluoksnio valdymą.

Aprašyta sistemos kūrimo strategija paremta komponentiniu kūrimu leidžia prireikus lengvai papildyti sistemą naujais komponentais, pakartotinai juos panaudoti kitose sistemose ar panašiuose projektuose. Apibrėžta duomenų sąsaja naudojant XML formatą leidžia atsiradus poreikiui su minimaliomis pastangomis prijungti naujus duomenų importavimo/eksportavimo šaltinius.

Sistemoje pasirinktas šabloninis grafinės sąsajos ir kitų tekstinių elementų „neįtraukimas“ į programinės įrangos kodą panaudojant šablonus, o jų užkrovimas iš XML failo leidžia ateityje pritaikyti sistemą kitoms kalboms.



23 pav. Sistemos Robustness diagrama

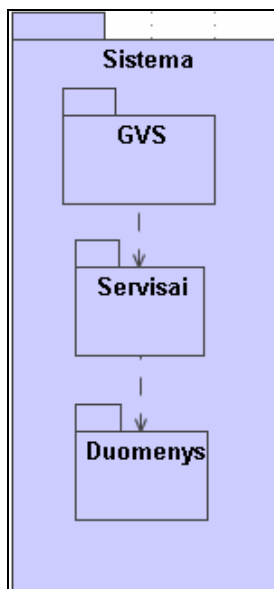
Sistemos robustness diagrama (žr. pav. 23) atvaizduoja bendrą sistemos darbą ir įvairių klasių sąveikavimą tarpusavyje. Kiekviena klasių grupė atsakinga už savo rolės veiksmus (elementų atvaizdavimą, DB įrašymą ir t.t.). Norint suprasti kaip atliekamas koks nors vienas konkretus darbas reiktų nagrinėti tik atskirą sąveikaujančių elementų grupę.

Grafinės vartotojų sąajos (GVS) paketo klasės atsakingos už teisingą informacijos atvaizdavimą bei užklausų pateikimą. Paketo „Servisai“ klasės atsakingos už pateiktų užklausų vykdymą, kreipimąsi į DB (įrašyti, trinti ir t.t.) bei rezultatų gražinimą. „Duomenų“ paketo elementai tarnauja kaip operantai – jie naudojami pateikiant užklausa, atliekant veiksmus bei rezultatų gražinimui.

3. 3. Sistemos pagrindiniai paketai

Aukščiausiam lygyje atskirti trys pagrindiniai loginiai elementai – grafinės vartotojo sąsaja, servisai ir veiklos logika, bei duomenys (paketas atitinkantis duomenų bazės lenteles).

Pagrindinių paketų diagrama pateikta paveikslėlyje apačioje (žr. pav. 24):

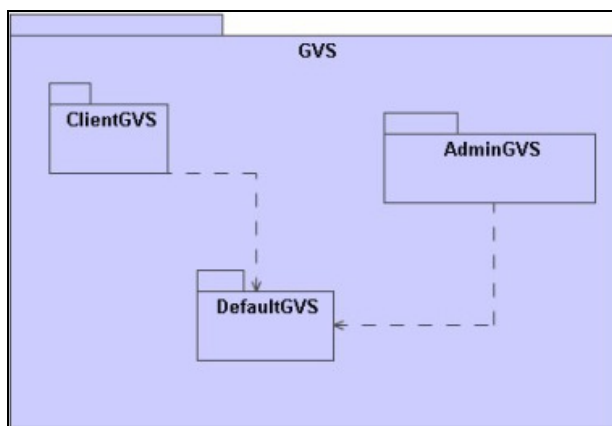


24 pav. Sistemos vaizdas, suskaidytas į paketus

3.3.1. Paketas „GVS“

Pakete saugomi visi bendriniai (default) vaizdai, “layout” šablonai, spalvos, meniu punktai ir t.t. Atitinkamas servisas generuoja vaizdą vartotojui pagal pasirinktą funkcionalumą. Kiekvienam tipui vartotojų priskiriamas skirtingas paketas.

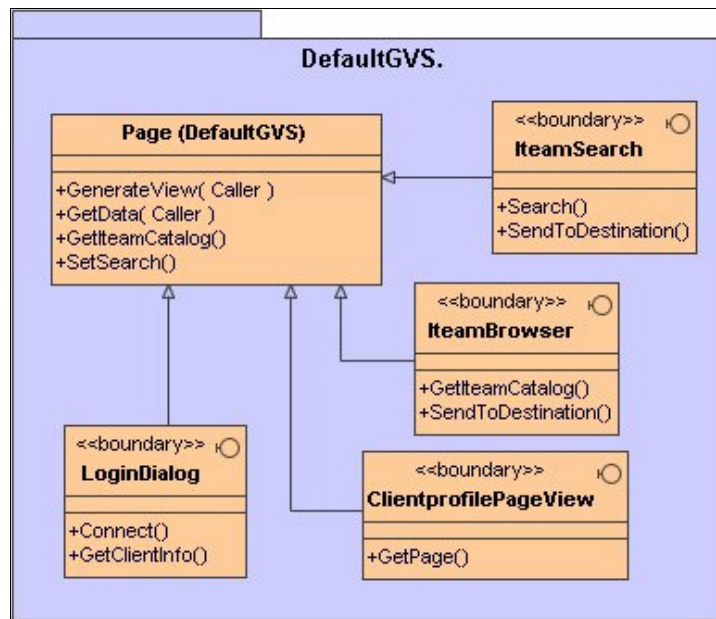
Paketo GVS diagrama pateikta paveikslėlyje apačioje (žr. pav. 25):



25 pav. paketo „GVS“ sudėtis

3.3.1.1. DefaultGVS

Paketas pasiekiamas visų tipų vartotojams. Jame saugomas bendras išdėstymas (default layout) bei komponentai atvaizduojami visuose generuojamuose puslapiuose (žr. pav. 26).

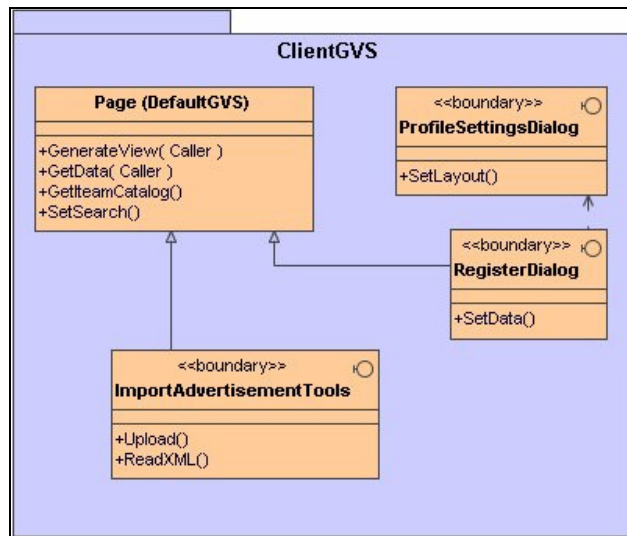


26 pav. Paketo *DefaultGVS* klasės

- **LoginDialog** – Prisijungimo dialogas;
- **ClientProfilePageView** – reprezentacinis kliento profilio puslapis;
- **IteamSearchDialog** – Interfeisas kuriame galima atlikti norimo produkto paiešką;
- **IteamBrowser** – Interfeisas kuriame paiešką galima atlikti pagal produktų kategorijas;
- **Page (DefaultGVS)** – puslapių bendrų metodų ir išdėstymo komponentų viršklasė;

3.3.1.2. ClientGVS

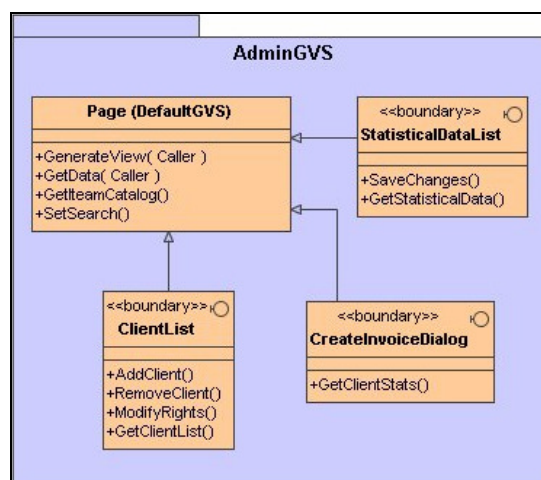
Paketas atspindintis pagrindines sistemos kliento funkcijas – registruotis, sukelti duomenis, nustatyti asmeninius skelbiamo puslapio parametrus ir išvaizdą. Nors funkcionalumas “užsiregistruoti” pasiekiamas visiem, laikoma kad lankytojas pradėjęs registracijos procesas virsta klientu (žr. pav. 27).



27 pav. Paketo *ClientGVS* klasės

- **Page (DefaultGVS)** – bendrinė puslapių struktūra, pasiekiami iš **DefaultGVS**
- **RegisterDialog** – registracijos interfeisas. Pats procesas susideda iš trijų etapų todėl nuo šio interfeiso neatsiejami:
 - **ProfileSettingsDialog** – asmeninių nustatymų interfeisas: pasirenkamas vaizdo ir išdėstymo šablonas, spalvos ir t.t.
 - **ImportAdvertisementTools** – pageidaujamos reklaminių produktų informacijos bei priemonių įkėlimo interfeisas.

3.3.1.3. AdminGVS



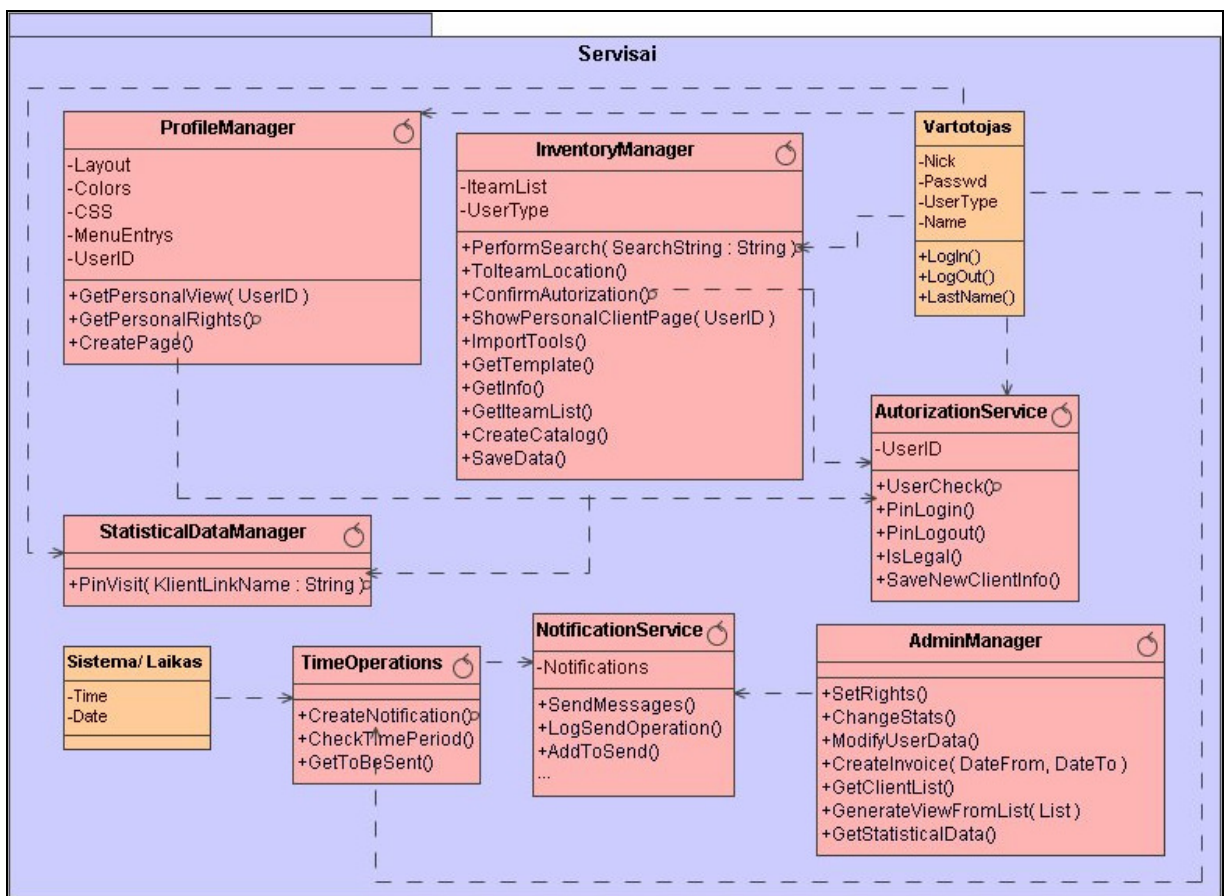
28 pav. Paketo *AdminGVS* klasės

- **Page (DefaultGVS)** – bendrinė puslapių struktūra, pasiekama iš **DefaultGVS**
- **ClientList** – interfeisas leidžiantis administratoriui peržvelgti klientų sąrašą, suteikti/apriboti teisas ir pan.;
- **CreateInvoiceDialog** – interfeisas leidžiantis administratoriui sukurti klientų ataskaitas;
- **StatisticalDataList** – interfeisas leidžiantis administratoriui peržvelgti statistinius duomenis;

3.3.2. Paketas „Servisai“

Pakete saugomas visas pagrindinis funkcionalumas. Čia generuojami asmeninių nustatymų (profilų) puslapiai, saugomi registracijos duomenys, atliekama autorizacija, bei įrašinėjami duomenys į DB. Kadangi visi procesai glaudžiai sąveikauja tarpusavyje nėra prasmės dar labiau juos skaldyt. Pvz.: **ProfileManager** kreipiasi į **AutorizationService** vienokio ar kitokio veiksmo legalumui nustatyti (ar turi vartotojas teisas konkrečiam veiksmui ar ne).

Schemoje pavaizduoti **Vartotojo** bei **Sistemas/Laiko** klasės tik simbolinės bendram kontekstui išryškinti (žr. pav. 29).



29 pav. Paketo „Servisai“ struktūra

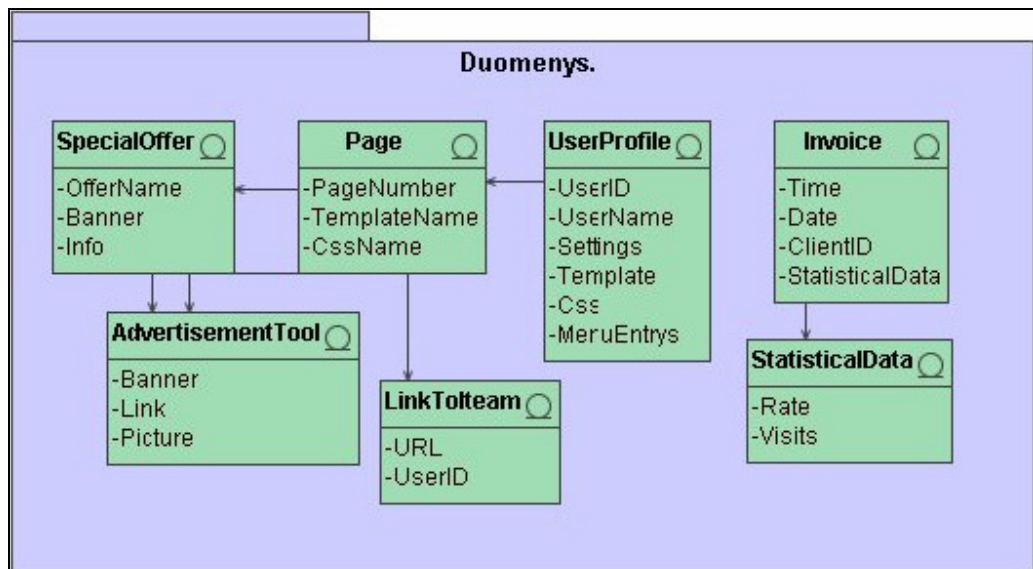
- **ProfileManager** – klasė atsakinga už individualių parametrų atvaizdavimą generuojamame (kliento skelbiamame) puslapyje, teisių nuskaitymą ir nustatymą. Viskas kas siejama su neinformacine dalimi yra šios klasės žinioje;
- **InventoryManager** - valdo visa informacinę dalį (nuotraukas, aprašus, nuorodas, banner'ius). Užpildo asmeninius puslapius informacija.
- **StatisticalDataManager** – įvykiais paremta klasė skaičiuojanti kiekvieno kliento visų produkto apšankimus.
- **TimeOperations** – nuo laiko priklausančios funkcijos, kuriančios automatiškai tam tikro periodo ataskaitas, tikrinančios išsiuntimui skirtų ataskaitų buvimą (nebuvimą) bei periodą (ar laikas siųsti).
- **NotificationService** - servisas paverčiantis ataskaitą į reikiamą formatą ir pridedantis ją prie “siųsti” ataskaitų.

- **AuthorizationService** – atlieka prisijungimo duomenų teisingumo patikrinimą, taip pat atliekamų veiksmų legalumą.
- **AdminManager** – administratoriaus funkcijų servisas (suteikti teisas, ištrinti/pridėti vartotoją, sugeneruot ataskaitą ir t.t.)

3.3.3. Paketas „Duomenys“

Arčiausiai duomenų bazės esantis servisas. Jame saugomi duomenų modeliai atitinkantys DB esančias struktūras, t.y. jų atributai atitinkamai užpildomi DB laukais (žr. pav. 30).

Atitinkamai saugant duomenis duomenų bazėje, šios klasės bus transformuojamos į DB lentelių laukus.



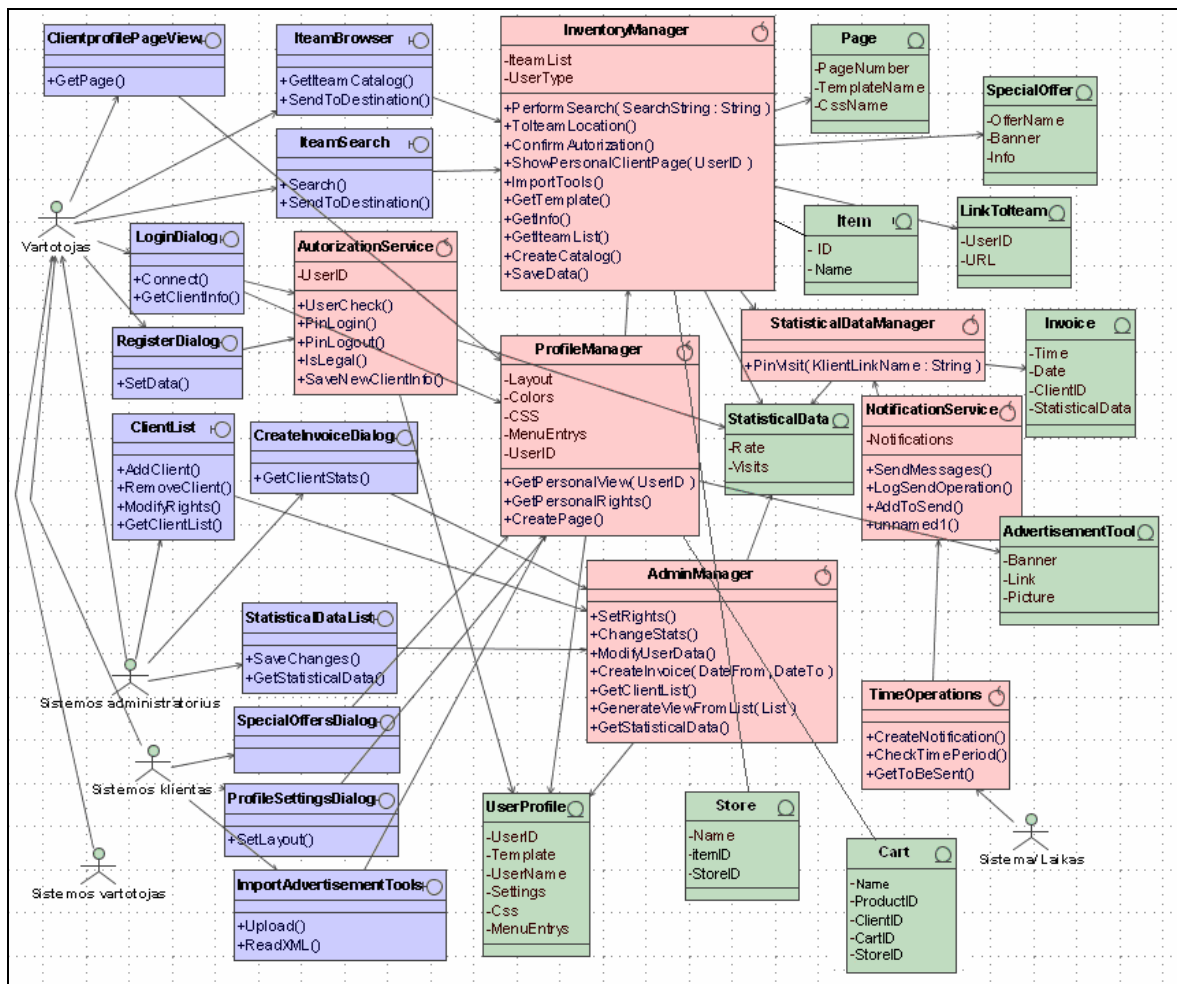
30 pav. Paketo „Duomenys“ struktūra

- **SpecialOffer** – specialių pasiūlymų klasė;
- **Page** – klasė sauganti puslapių skaičių, naudojamo šablono pavadinimą ir pan. Dalyvauja generuojant galutinį vaizdą vartotojui;
- **UserProfile** – informacijai apie vartotoją saugoti skirta klasė;
- **Invoice** – ataskaitų klasė (sukūrimo data, periodas, apsilankymai);
- **AdvertisementTool** – reklaminių priemonių ir kitų pagalbinių priemonių saugojimui skirta klasė (nuotraukos, aprašai, baner'iai ir);
- **LinkToTeam** – informacija apie galutinę reklamuojamo produkto vietą (kokiam klientui priklauso, koks galutinio tikslo adresas);

- **StatisticalData** – statistinių duomenų klasė.

Bendrojo atvejo sąveikos diagrama

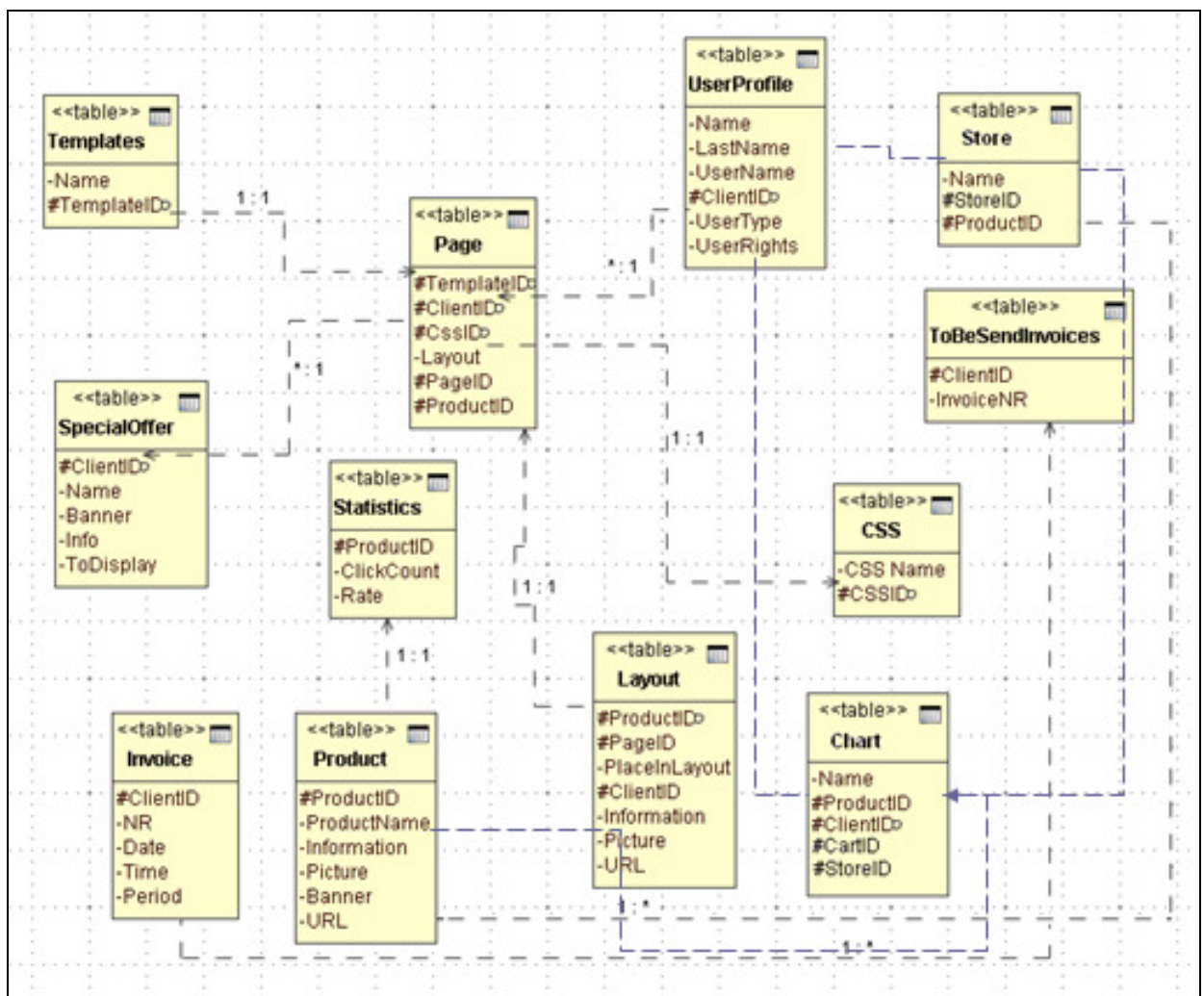
Išskleista bendrojo atvejo sąveikos diagrama atrodo taip (žr. pav. 31):



31 pav. Bendrojo atvejo išskleista robustness diagrama

Duomenų bazės vaizdas

Duomenų bazės vaizdas pateiktas žemiau (žr. pav. 32):

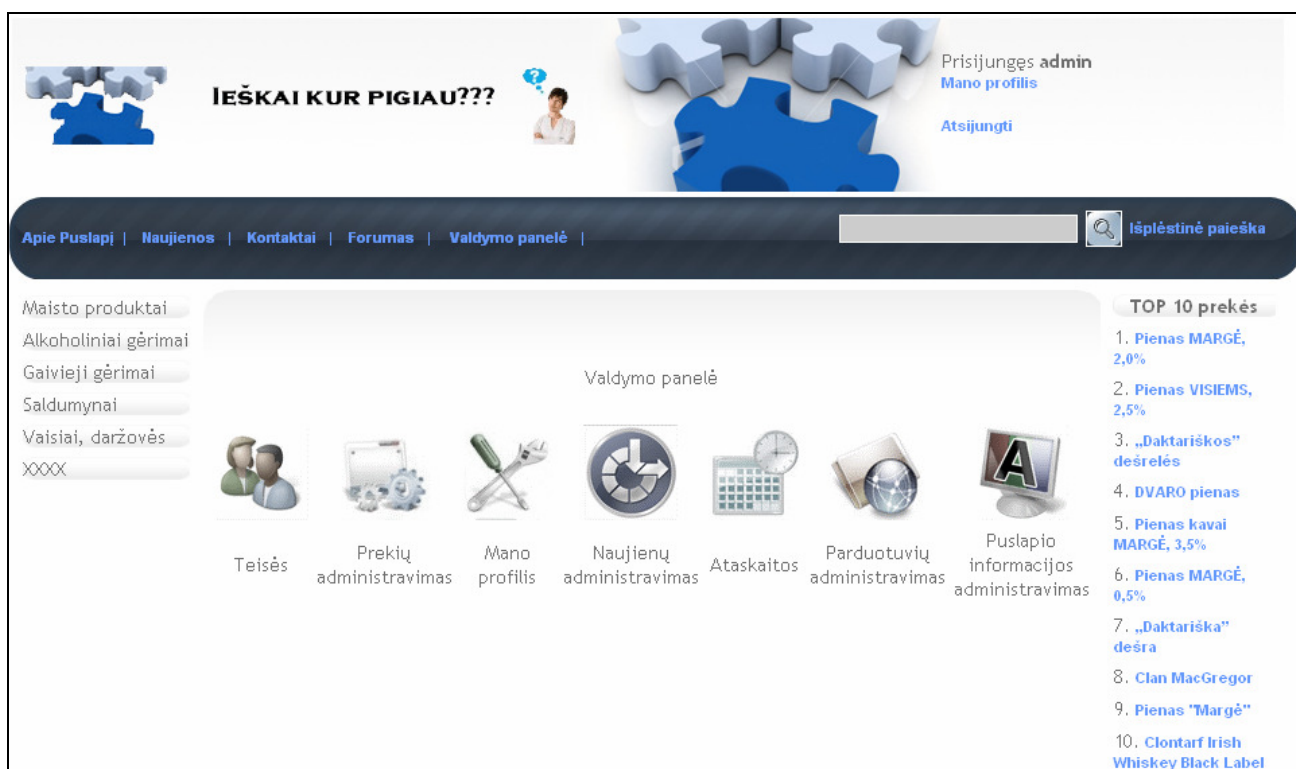


32 pav. Duomenų bazės modelis

4. TYRIMO DALIS

Programinė įrangos kokybės įvertinimui buvo pasirinkti tie patys kriterijai vertinant analogines WIKI sistemas.

1. Įdiegimas – kokios sistemos įdiegimo sąlygos;
2. Funkcionalumas – kokį funkcionalumą siūlo (pobūdis);
3. Sudėtingumas (vartotojo aplinka (žr. pav. 33)) – kiek sudėtinga yra perprasti sistemą;
4. Duomenų valdymas – kaip sistema organizuota informacijos saugojimo, surinkimo ir valdymo prasme;
5. Išplėtimo galimybės – kiek sudėtinga papildyti sistemą nauju funkcionalumu (kokia programavimo kalba).



33 pav. Sukurtos sistemos pagrindinis puslapis

4. 1. Įdiegimas

Sistemos funkcionavimui reikalingas web aplikacijas palaikantis serveris su veikiančia duomenų baze. Taip pat reikia, kad serveris palaikytų PHP kalbą. Rekomenduojama – nemokama programinė įranga „WAMP Server“, savyje apjungianti tris suderintus komponentus: „Apache server“, PHP bei „MySQL“. Darbo realizavime buvo naudojama „WAMP Server 2.0“.

Papildoma sąlyga – duomenų bazėje turi būti sukurta atskira duomenų bazė su specialia lentelių struktūra. Kai kurie duomenys turi būti įrašyti iš anksto.

Kita papildoma sąlyga – išankstinis katalogų medis, bei atitinkamai sukeltos bibliotekos, tiek procesų, tiek „Zend Framework“ funkcijoms palaikyti.

Galimi patobulinimai

Įdiegimo dialogo sukūrimas. Dialogo metu iš vartotojo būtų gaunamas duomenų bazės pavadinimas ir prieigos detalės, o sistemos būtina lentelių struktūra, bei kaikurie privalomieji duomenys būtų įdiegiami automatiškai. Taip pat šio proceso metu būtų sukurtas procesams reikalingas katalogų medis, bei įkeltos visos procesų, bei „Zend Framework“ funkcionalumo palaikymui reikalingos, bibliotekos. Vartotojui tereikėtų paspausti keletą mygtukų.

4. 2. Funkcionalumas ir Sudėtingumas

Sistema nutaikyta į tikslingos apie produktų ir paslaugų informacijos (kainų) rinkimą. Sistemos klientai – interneto vartotojai, norintys surinki ar gauti informaciją apie pigiausias siūlomas, tos pačios rūšies, prekes. Jų tikslas – „rasti kur pigiau“. Kadangi nebuvo galimybės nuspėti būsimų vartotojų amžiaus, ar techninio IT žinių lygio, kuriant sistemos funkcijas buvo laikomasi „look & feel“ principo – kad dauguma funkcijų būtų lengvai nuspėjamos, laukiamas rezultatas nebūtų netikėtas.

Visą funkcionalumą reikia nagrinėti dviem požiūriais:

- Vartotojiškos funkcijos;
- Administratoriškos funkcijos.

Vartotojiškos funkcijos apima informacijos paieškos, redagavimo, šalinimo ar įkėlimo procesus, taip pat krepšelių administravimą. Visi procesai pasiekiami per patogiu, kategorizuotus vaizdinius valdymo meniu (34 pav., 35 pav.):



34 pav. Valdymo meniu

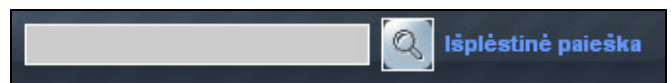


35 pav. Valdymo kategorijos „Prekių administravimas“ submeniu

Norimo produkto paieška atlieka tradiciškai per paieškos lauką, arba per pateikiamų produktų medį (žr.36 pav., 37 pav.):

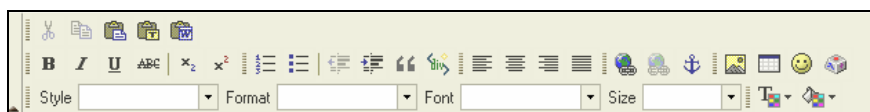


36 pav. Produktų medis



37 pav. Paieškos laukas

Tekstinės informacijos redagavimui, sistemos reikmėms buvo adaptuotas „FCKeditor“ – dar vienas nemokamas, FLASH pagrindu veikiantis, produktas (žr. 38 pav.).



38 pav. „FCKeditor“ teksto redaktorius

Taip pat yra galimybė struktūrizuotų duomenų įkėlimui – tereikia nurodyti pagal tam tikras taisykles sudaryta XML failą, ir jame esantys duomenys nuskaitomi automatiškai (žr. pav. 39).

39 pav. ML įkėlimo funkcija

Administratoriškos funkcijos – susijusios su vartotojų profilių tvarkymu, teisių priskyrimu, sistemos reprezentacinių skilčių (pagrindinio ar kontaktų puslapio) tvarkymu, ataskaitų generavimu ir pan. (žr. pav. 40). Procesai pasiekiami per valdymo panelės meniu grafiniu elementus.

Pienas "Margė" 3,2%	2.6	3.05	17.31%
Pienas, 2,5% riebi	2.11	2.16	2.37%
Pienas Dvaro	1.99	2	0.5%
Pienas ŽVAIGŽDUTĖ, 2,0%	1.89	1.88	-0.53%
Pienas VALIO ZERO LACTOSE 1,5%	4.68	4.86	3.85%

40 pav. Kainų kitimo ataskaitos vaizdas

Galimi patobulinimai

1. Sistemos duomenų bazė automatinio suarchyvavimo funkcijos administratoriui sukūrimas. Funkcija turėtų priminti periodiškai kurti duomenų bazės kopijas atsitiktinio

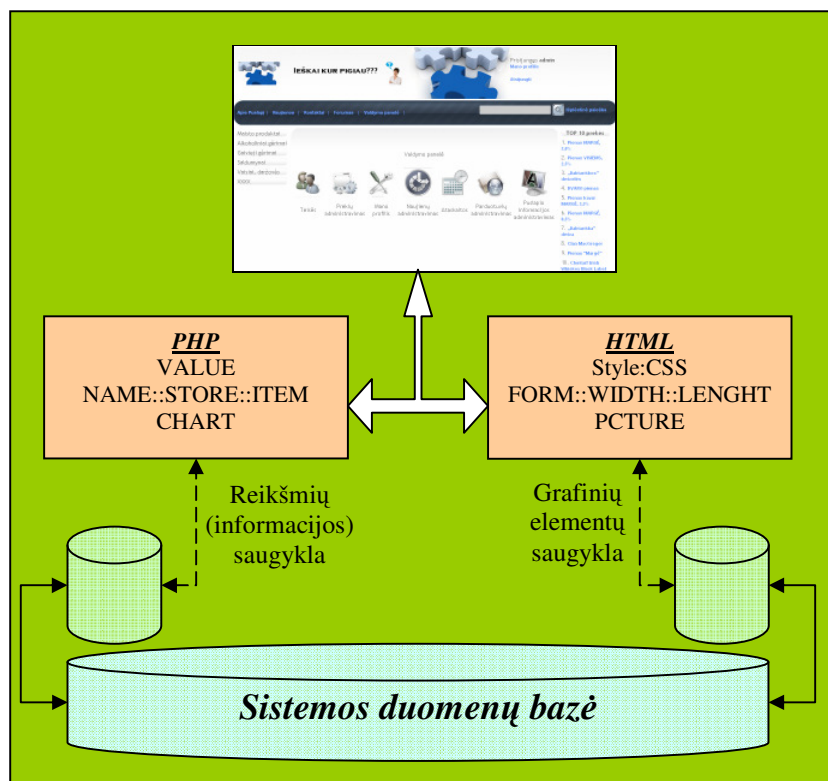
informacijos praradimo išvengimui. Prireikus informacija galėtų būti nesunkiai atstatoma. Nauja problema – reikalinga papildoma atminties talpa.

2. Automatinis, neteisingai įvestų kainų atpažinimo procesas. Galima būtų apskaičiuoti konkrečios prekės, ar tos prekių grupės statistinį vidurkį ir dispersiją (standartinį nuokrypį) bendrai, ar konkrečiame parduotuvių tinkle. Tokiu būdu patikrinus galima būtų eliminuoti dalį neteisingų, į kainų išsibarstymo statistines (dispersijos) ribas nepatenkančių, kainų.
3. Registravimo funkcijos aktyvavimo/deaktyvavimo galimybė sukūrimas. Sprendžiant WIKI sistemų uždavinį svarbu žinoti ribines sistemos galimybes, pavyzdžiui.: kiek vartotojų gali vienu metu būti prisijungę prie sistemos. Pasiiekus ribinę reikšmę, registracija turėtų būti sustabdyta automatiškai, ar administratoriaus pagalba.
4. Automatinis XML failų formatų nustatymas pagal vokus (*angl.* Envelope). Vartotojas nurodo pagal kokias taisykles sudarytas XML ir sistema automatiškai prisiderina prie pokyčių.

4. 3. Duomenų valdymas

Duomenų valdymas vyksta struktūrizuotai, tačiau skirtingai nei lyginant su „Moodle“ sistema, duomenų bazė nėra perkrauta. Grafinių elementų struktūra nesudėtinga ir gali būti lengvai keičiama, formos, spalvų ar kitais atžvilgiais – išvaizdos priemonės visiškai atskirto nuo funkcinio, „*Zend Framework*“ dėka.

Lyginant su „Microsoft Sharepoint-Wiki“ kiekvienos skelbiamos informacijos vienetui nekuriamas naujas HTML puslapis. Vietoj to informacija skaidoma saugojimui, o atvaizduojant kiekviena kartą generuojama iš naujo (*žr. pav. 41*).

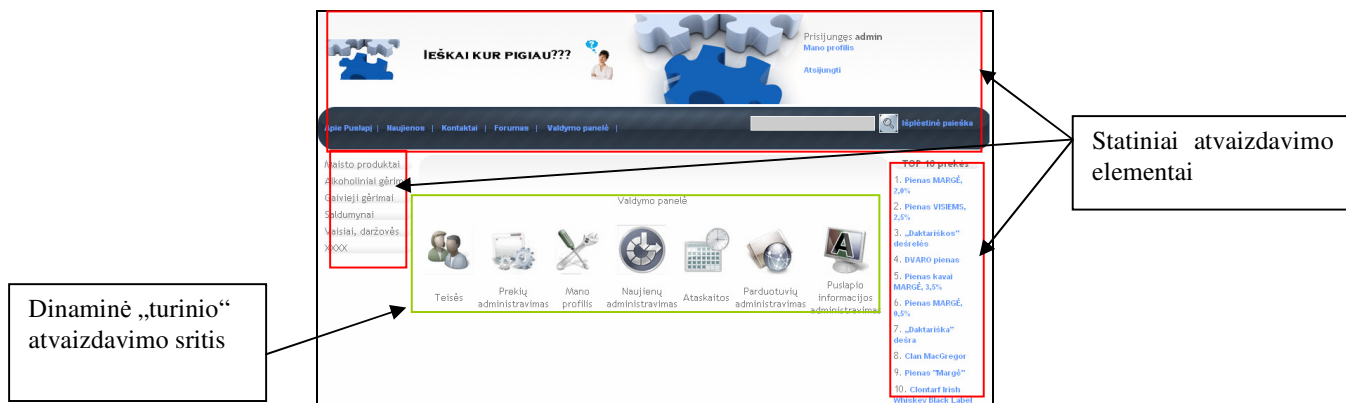


41 pav. Duomenų saugojimo schema

4. 4. Išplėtimo galimybės

Sistema realizuota taip, kad funkciniai paketai, ar jų metodai, visiškai nepriklauso vienas nuo kito. Naujai diegiamas funkcionalumas neturėtų įtakos jau egzistuojančiam sistemos darbui (veikimo prasme).

Atvaizdavimo (grafiniai) elementai kuriami kiekvienam funkcinio paketo metodui atskirai – sistema organizuota taip (žr. pav. 42):



42 pav. Grafiniai elementai

Kiekvieno metodo skaičiavimų rezultatas, bei jam priklausantis atvaizdavimo elementas pateikiami turinio srityje. Tai pagrindiniai „Zend Framework“ veikimo principai.

Nors taisyklės ir nėra visiškai paprastos ir reikalauja šiek tiek laiko jas perprasti, vis vien jos yra paprastesnės nei lyginant su kitomis analoginėmis sistemomis.

4. 5. Išvados

Sistema gali būti naudojama kaip atskiras programinis paketas. Sukūrus įdiegimo meniu palengvėtų administratoriaus darbas – visa sistema būtų įdiegta kelių mygtukų paspaudimu.

Funkcijos nesunkiai perprantamos – teksto redaktorius veikia WYSIWYG (*angl.* „What You See Is What You Get“) principu, tai yra rezultate gaunamas redagavimo metu kuriamas vaizdas. Kitas funkcionalumas įgyvendintas per grafinę vartotojo sąsają („look & feel“ principas).

Informacijos kokybės užtikrinimui siūlomas matematinių metodų (vidurkio ir dispersijos) pritaikymas; atstatomumui – periodinis duomenų bazės kopijų kūrimas.

Kadangi planuojama, kad sistema bus pasiekiamą visiems norintiems, svarbu įgyvendinti sistemos registracijos skelbimo pradžios ir pabaigos galimybes, kad netyčia nebūtų peržengtos palaikančios techninės įrangos galimybės. Šis kriterijus turėtų būti kintamas dydis priklausantis tik nuo sistemos veikimo aplinkos (techninės įrangos).

Kita aktuali problema – struktūrizuotas didelio informacijos kiekio automatinis nuskaitymas ir duomenų bazės užpildymas. Planuojama darbų kryptis - apibendrinto XML failų nuskaitymo mechanizmo sukūrimas, kai dokumento pradžioje nurodomos taisyklės, kaip dokumentas buvo sukurtas.

5. Eksperimentinė dalis

5.1. Vartotojo krepšelio dydžio eksperimentas

Vartotojas savo aplinkos profilyje gali susikurti neribotą skaičių krepšelių ir paieškos (analizės) rezultatus gauti pagal vieną iš jų.

Eksperimento tikslas – ištirti ar krepšelio dydis gali peržengti sistemos galimybes, tai yra rasti ribą, kada sistema nustoja funkcionuoti ir lūžta (priklausomai nuo užklausos dydžio).

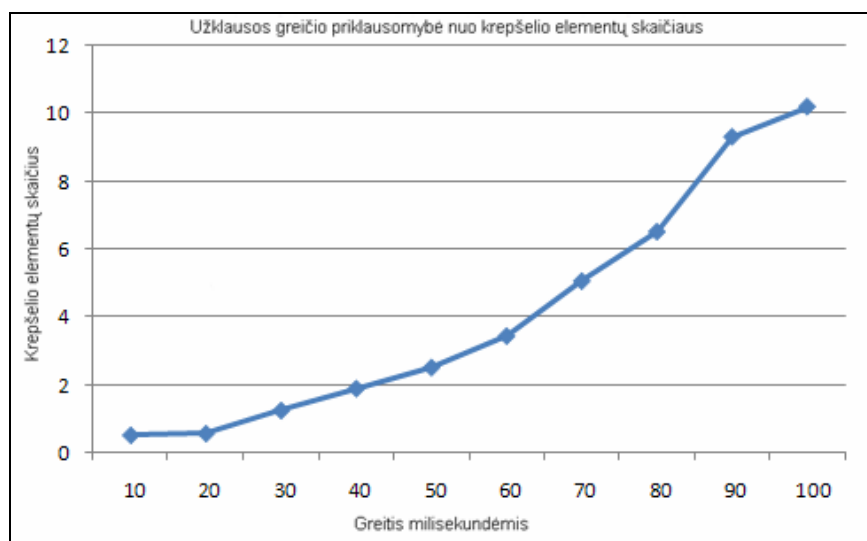
Laukiamas eksperimento rezultatas – krepšelio kritinio elementų kiekio nustatymas. Eksperimentas atliekamas naudojant papildomą, specialiai sukurtą, sistemos modulį – atliekant operaciją jis fiksuoja operacijos pradžios laiką, o pateikus rezultatus – pabaigos laiką.

Eksperimento metodika: testiniam vartotojui sukuriama krepšelis į kurį įtraukiamas skirtingas kiekis prekių, kuris vis didinamas po 10 vnt. Kiekvieną kartą rezultatų paieška atliekama tris kartus siekiant įsitikinti, kad užklausos laikas nebuvo atsitiktinis.

Rezultatai pateikti 12 lentelėje bei 43 paveiksle:

12 lentelė. Eksperimento rezultatai

Krepšelio elementų kiekis	10	20	30	40	50	60	70	80	90	100
Užklausos laikas (milisekundėmis)	0.51	0.56	1.24	1.69	2.51	3.44	5.06	6.52	9.32	10.21



43 pav. Eksperimento rezultatai

Eksperimento Išvados

Sistemos veikla nesutriko. Aiškiai pastebima užklauso atsako laiko augimo tendencija. Tyrimo rezultatai priklauso nuo techninės įrangos bei internetinio ryšio stiprumo ir stabilumo, tačiau bendros laiko augimo tendencijos turėtų išlikti tos pačios. Didėjant duomenų bazei užklauso atsako greitis turėtų taip pat didėti.

Rekomenduojamas krepšelio dydis - ne daugiau 40 elementų. Iki šios ribos rezultatai kinta nežymiai.

5. 2. „Zend“ bendravimo su duomenų bazė modulio eliminavimas

Visa sistemos funkcinė bazė realizuota „Zend Framework“ paketo pagalba. Kadangi jo veiklos metodika veikia pagal tam tikrą šabloną, kyla klausimas ar darbas su duomenų baze neužtrunka ilgiau nei tą patį darbą atliekant tiesioginio kodo pagalba.

Eksperimento tikslas – ištirti ar „Zend Framework“ veikimo logika neprailgina darbo su duomenų baze atliekant elementariausias funkcijas.

Eksperimento metodika – atskirai nuo šabloninio varikliuko buvo sukurtas papildomas išorinis modulis skirtas darbui su duomenų baze. Atliekant rezultatų pateikimą pagal pasirinktą krepšeljį, fiksuojamas laikas nuo operacijos pradžios iki pabaigos.

Rezultatai pateikiami 13 ir 14 lentelėse:

13 lentelė. Eksperimento su „Zend“ moduliu atliktų užklauso atsako rezultatai

Krepšelio elementų kiekis	10	20	30	40	50	60	70	80	90	100
Užklauso laikas (milisekundėmis)	0.51	0.56	1.24	1.69	2.51	3.44	5.06	6.52	9.32	10.21

14 lentelė. Eksperimento su atskiru moduliu atliktų užklauso atsako rezultatai

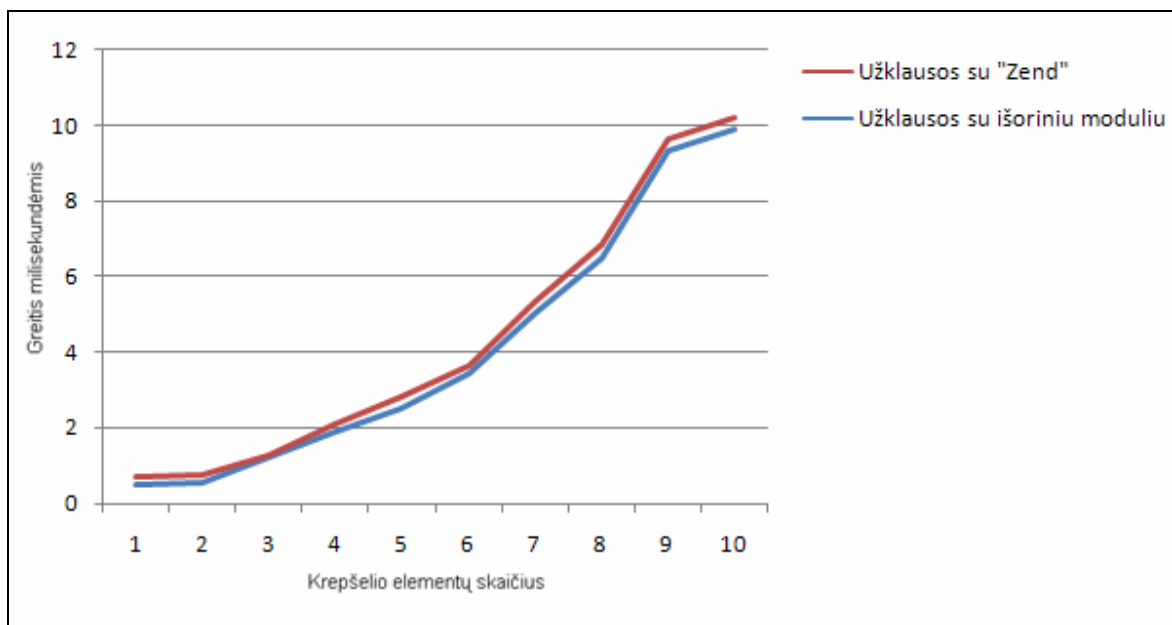
Krepšelio elementų kiekis	10	20	30	40	50	60	70	80	90	100
Užklauso laikas (milisekundėmis)	0.49	0.53	1.24	1.66	2.41	3.41	5.08	6.22	9.22	9.91

Užklausių atsako laiko palyginimas su išoriniu moduliu ir be jo:

15 lentelė. Eksperimento su atskiru moduliu atliktų užklausių atsako rezultatai

Krepšelio elementų kiekis	10	20	30	40	50	60	70	80	90	100
Užklauso laikas (milisekundėmis) su išoriniu moduliu	0.49	0.53	1.24	1.66	2.41	3.41	5.08	6.22	9.22	9.91
Užklauso laikas (milisekundėmis) su „Zend Framework“	0.51	0.56	1.24	1.69	2.51	3.44	5.06	6.52	9.32	10.21

Palyginimo rezultatų grafikas:



44 pav. Eksperimento rezultatai: užklausių atsako laikas su „Zend“ ir su išoriniu moduliu

Eksperimento išvados

Iš pateikto grafiko matyti, kad išorinis modulis, bendradarbiaudamas tiesiogiai su duomenų baze, veiksmus atlieka šiek tiek greičiau nei „Zend Framework“. Kadangi skaičiavimai atlikti milisekundėmis, skirtumas yra nereikšmingas. Keičiantis duomenų bazės dydžiui jis gali keistis.

5. 3. Eksperimentų išvados

Aiškiai pastebima užklauso atsako laiko augimo tendencija. Tyrimo rezultatai priklauso nuo techninės įrangos bei internetinio ryšio stiprumo ir stabilumo, tačiau bendros laiko augimo tendencijos turėtų išlikti tos pačios. Didėjant duomenų bazei užklauso atsako greitis turėtų taip pat didėti.

Rezultatų dėka nustatytas rekomenduojamas vartotojo krepšelio elementų skaičius – 40. Su didesniu elementų skaičiumi užklauso atsako laikas ženkliai pasikeičia. Taip pat nustatyta, kad nėra prasmės keisti darbo su duomenų baze modelio – išorinio modulio ir „*Zend Framework*“ formuojamų užklauso laikas beveik nesiskiria, tačiau šis skirtumas gali keistis duomenų bazei plečiantis.

Išvados

1. Apibrėžtos ir išanalizuotos duomenų surinkimo sistemos, jų veikimo modeliai ir pagrindiniai principai. Sistemos realizavimui pasirinktas interaktyvus duomenų surinkimo sistemos modelis su WIKI veiklos logikos realizavimo schema. Kaip rodo kai kurie tyrimai ([15] [24]), nemokamos WIKI sistemos sparčiau auga vartotojų bei informacijos kiekio atžvilgiu. Funkcinės dalies įgyvendinimui pasirinktas „Zend Framework“ šablonų varikliukas kaip vienas geriausių.
2. Atlikta duomenų surinkimo ir valdymo metodikos analizė. Praktiškai pritaikyti „Išankstinio internetinių formų užpildymo“, „Inkrementinio duomenų registravimo“, bei „Multi-formų registracijos“ metodai.
3. Surenkamų duomenų kokybei bei prieigai užtikrinti taikomi aktyvios ir pasyvios analizė metodai. Sukurtoje sistemoje taip pat įgyvendinti kai kurie rekomendaciniai vartotojų išlaikymo ir sudominimo (pritraukimo) metodai (pvz.: aplinkos personalizavimas, pagal dažniausiai atliekamas funkcijas ir pasirinkimus).
4. Sistemos realizacija rodo, kad pasirinktas architektūrinis modelis užtikrina grįžtamąjį ryšį, kuris veikia savarankiškai, tačiau priklauso nuo vartotojo priimtų sprendimų, atliktų veiksmų. Sistemą galima laikyti intelektualia ir adaptyvia, nes ji padeda vartotojams priimti tam tikrus sprendimus, bei prisitaiko prie nuolat kintančios situacijos sistemoje.
5. Norint įsitikinti sistemoje pritaikytų metodų efektyvumu ir naudingumu, reikėtų sistemą stebėti realioje (veikimo) aplinkoje tam tikrą laiką: sekti kaip plečiasi sistemos duomenų bazė, ar yra daug netikslingos informacijos, ar nėra (kritinių) vartotojų nusiskundimų sistemos atžvilgiu. Sistemos tyrimo metu, pagal nustatytus kriterijus, išskirtos galimos tobulinimo sritys išvardintiems sekimo veiksams palengvinti.
6. Atlikus analizę, daroma prielaida, kad nors ir be siūlomų funkcinių patobulinimų, sistema turėtų sėkmingai veikti: nesunkiai ir greitai pateiktų ieškomų prekių rezultatus, rezultatus filtruotų pagal pasirinktus krepšelius, automatiškai aptiktų neteisingai ar klaidingai įvestas informacines reikšmes.

7. Sistemos eksperimentinių tyrimų metu rezultatai leido apibrėžti vartotojų krepšelio dydį, taip pat parodė, kad realizacine prasme „Zend Framework“ šablono darbo su duomenų baze rezultatai mažai skiriasi nuo tiesioginio modulio darbo.

LITERATŪROS SĄRAŠAS

- [1] CARMAN, Bill. "OVERVIEW OF DATA COLLECTION TECHNIQUES" [interaktyvus]. 2009 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.idrc.ca/en/ev-56606-201-1-DO_TOPIC.html>
- [2] FERNADEZ, Mary; SUCIU, Dan. „WORKSHOP DATA PROCESSING ON THE WEB“ [interaktyvus] 2000 [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://dimacs.rutgers.edu/Workshops/Processing/abstracts.html>>
- [3] BARTLEY, Curtis. BROAN, Jim. "MS WIKI" [interaktyvus]. 2005 lapkritis [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://c2.com/cgi/wiki?MsWiki>>
- [4] CARDARELLI, Mauro. „WIKI WHILE YOU WORK“ [interaktyvus]. 2008 [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://technet.microsoft.com/en-us/magazine/cc162514.aspx>>
- [5] KWizCom org. „SHAREPOINT WIKI PLUS“ [interaktyvus]. 2005 [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://www.kwizcom.com/ProductPage.asp?ProductID=524&ProductSubNodeID=525>>
- [6] THAKE, Jeremy . „SHAREPOINT DEV“ [interaktyvus]. 2009 kovas [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://www.sharepointdevwiki.com>>
- [7] METABALL org. „WIKI PRONCIPLES“ [interaktyvus]. 2009 kovas [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://c2.com/cgi/wiki?WikiPrinciples>>
- [8] DATA ID org. „DATA COLLECTION SYSTEMS“ [interaktyvus]. 2006 kovas [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://www.dataid.com/dcsystems/collect.htm>>
- [9] LYNCH, John. Accela Communications, Inc. "ONLINE DATA COLLECTION AND MANAGEMENT. TECHNIQUES FOR BUILDING INTELIGENT WEB" [interaktyvus]. 2008 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.accelacomm.com/PDFs/WP_ODC_P1.pdf>
- [10]] LYNCH, John. Accela Communications, Inc „DATA COLLECTION. OPTIMIZING AND MANAGING DATA CAPTURE OVER THE WEB“ [interaktyvus]. 2008 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.accelacomm.com/PDFs/WP_Online_Data_Collection.pdf>
- [11] HANSEN, Todd. OTERO, Jose. MCGREGOR, Tony. BRAUN, Hans-Werner. „ACTIVE MEASUREMENT DATA ANALYSIS TECHNIQUES“ [interaktyvus]. 2000 kovas [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.nlanr.net/Papers/AMPcase_studies.pdf>

- [12] MUKHERJEE, Pia. "DATA ANALYSIS TECHNIQUES" [interaktyvus]. 2007 [žiūrėta 2009-05-20]. Prieiga per internetą: http://astronomy.sussex.ac.uk/~pm52/data_analysis_techs/coursedoc.pdf
- [13] FOUNTAIN, Renee. „WIKI PEDAGOGY“ [interaktyvus]. 2004 2007 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://jedsberg.googlepages.com/WikiPedagogy.pdf>
- [14] BHOLE, Abhijit. FORTUNA, Blaž. GROBELNIK, Marko. MLADENIC, Dunja. „MINING WIKIPEDIA AND RELATING NAMED ENTITIES OVER TIME“ [interaktyvus]. 2008 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.cse.iitb.ac.in/~abhijit.bhole/SiKDD2007ExtractingWikipedia.pdf>
- [15] TARABORELLI, Dario. „MEASURING WIKI VIABILITY“ [interaktyvus]. 2008 vasaris [žiūrėta 2009-05-20]. Prieiga per internetą: http://cress.soc.surrey.ac.uk/CressPresentations/cress_sem_slides-Dario.pdf
- [16] FAULKNER, George. SUAREZ, Luis. „HOW TO STRUCTURE A WIKI“ [interaktyvus]. 2007 sausis [žiūrėta 2009-05-20]. Prieiga per internetą: http://www.ibm.com/podcasts/shortcuts/SC_02022007_tr24.pdf
- [17] METCALFE, Randy. „EVALUATING OPEN SOURCE WIKIS AND VLES“ [interaktyvus]. 2007 kovas [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.oss-watch.ac.uk/talks/2007-03-20-leeds/index.pdf>
- [18] MOODLE org. „ABOUT MOODLE“ [interaktyvus]. 2009 [žiūrėta 2009-05-20]. Prieiga per internetą: http://docs.moodle.org/en/About_Moodle
- [19] MICROSOFT TECH-NET co. „HARDWARE AND SOFTWARE REQUIREMENTS (WINDOWS SHAREPOINT SERVICES 2.0)“ [interaktyvus]. 2009 [žiūrėta 2009-05-20]. Prieiga per internetą: <http://technet.microsoft.com/en-us/library/cc288713.aspx>
- [20] MOODLE org. „INSTALLING MOODLE“ [interaktyvus]. 2009 [žiūrėta 2009-05-20]. Prieiga per internetą: http://docs.moodle.org/en/Installing_Moodle
- [21] MOODLE org. „DEVELOPMENT: DEVELOPER DOCUMENTATION“ [interaktyvus]. 2009 [žiūrėta 2009-05-20]. Prieiga per internetą: http://docs.moodle.org/en/Development#How_Moodle_development_works
- [22] R. THIRUMALAI SELVI. Dr.N. V. BALASUBRAMANIAN. GEORGE T. MANOHAR. „FRAMEWORK AND ARCHITECTURAL STYLE METRICS FOR COMPONENT BASED SOFTWARE ENGINEERING“. [interaktyvus]. 2008 balandis [žiūrėta 2009-05-20]. Prieiga per internetą: <http://www.wseas.us/e-library/conferences/2008/hangzhou/acacos/11-586-119.pdf>

- [23] BOLLOG, Ian. „WIKI OPEN SOURCE BAROMETER” [interaktyvus]. 2008 lapkritis [žiūrėta 2009-05-20]. Prieiga per internetą: <<http://alfresco.com/wiki-open-source-barometer/>>
- [24] ROTH, Camille. TARABORELLI, Dario. GILBERT, Nigel “MEASURING WIKI VIABILITY” [interaktyvus]. 2008 [žiūrėta 2009-05-20]. Prieiga per internetą:<<http://nitens.org/docs/wikidyn.pdf>>