

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aurimas Junčis

**Dinaminė interneto sąsaja įmonių klientų-ryšių
valdymo sistemoms patobulinti**

Magistro darbas

Darbo vadovas

prof. dr. L. Nemuraitė

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aurimas Junčis

**Dinaminė interneto sąsaja įmonių klientų-ryšių valdymo
sistemoms patobulinti**

Magistro darbas

Recenzentas

doc. dr. V. Pilkauskas

2008-01-

Vadovas

prof. dr.L. Nemuraitė

2008-01-

Atliko

IFM-2/4 gr. stud.
Aurimas Junčis

2008-01-

Kaunas, 2008

Turinys

1. Įvadas.....	3
2. RIA taikomųjų programų analizė.....	8
2.1. Web 2.0 analizė.....	8
2.2. Raiškių interneto taikomųjų programų analizė.....	10
2.3. Dinamiškų puslapių kūrimo metodų literatūros šaltiniuose analizė.....	12
2.4. CRM sistemų analizė.....	19
2.5. RSS skaitytuvų analizė.....	20
2.6. Architektūros ir galimų įgyvendinimo priemonių variantų analizė.....	23
2.7. Siekiamų CRM sistemos savybių apibrėžimas.....	27
2.8. Kompiuterizuojamos sistemos funkcijos.....	28
2.9. Reikalavimai duomenims.....	30
2.10. Nefunkciniai reikalavimai ir apribojimai.....	30
2.11. Rizikos faktorių analizė.....	31
2.12. Rezultato kokybės kriterijai.....	32
2.13. Analizės išvados.....	33
3. CRM sistemos reikalavimų specifikacija ir analizė.....	34
3.1. Sistemos vartotojų panaudojimo atvejai.....	34
3.2. Sistemos vartotojų panaudojimo atvejų specifikacijos.....	35
3.3. Sistemos vartotojų navigavimo planai.....	57
3.4. Dalykinės srities modelis.....	59
3.5. Reikalavimų analizė.....	60
3.6. Reikalavimų analizės apibendrinimas.....	64
4. CRM sistemos tobulinimo projektas.....	65
4.1. Sistemos įgyvendinimui pasirinkto metodo pagrindimas ir esmės išdėstymas.....	65
4.2. Sistemos architektūra - statinės struktūros modelis.....	66
4.2.1. Loginė visos sistemos architektūra.....	66
4.2.2. Vartotojo paslaugos.....	67
4.2.3. Veiklos paslaugos.....	68
4.2.4. Duomenų paslaugos.....	70
4.2.5. AJAX varikliuko paslaugos.....	70
4.3. Sistemos elgsenos modelis.....	71
4.4. Duomenų bazės schema.....	77
5. Eksperimentinė realizacija.....	80
5.1. Sistemos komponentai.....	80
5.2. Sistemos įdiegimo modelis.....	81
5.3. Reikalavimai sistemos funkcionavimo palaikymui.....	81
5.4. Sistemos veikimo aprašymas.....	82
5.5. Testavimas.....	92
6. CRM sistemos patobulinimo įvertinimas.....	96
6.1. Dinamiškų sistemos savybių tyrimas ir įvertinimas.....	96
6.2. Įvertinimų suvestinė ir išvados.....	105
6.3. RIA taikomųjų programų savybių pritaikymo CRM ir kitoms organizacijų sistemoms rekomendacijos.....	107
7. Išvados.....	108
8. Literatūra.....	109
9. Terminų ir santrumpų žodynas.....	110
Priedai.....	111

SUMMARY

Dynamic Web interface for improvement of enterprise customer relationship management systems

This work is concentrated on creation of dynamic web systems. Web 2.0 brings new requirements to internet based systems (web pages). Users needs to work with large amounts of data (information) using simple web browser are increasing very fast. So there is a problem when we need to represent this data on web page. We need dynamic systems (pages) to do it fast ant easy to operate. Dynamic system (page) is internet based system (web page), which most functions are using Ajax (asynchronous communication with server).

The real experimental system to increase a dynamism of Customer Relationship Management systems by extending them with Rich Internet Applications characteristics is created and analyzed. Recommendations to improve similar existing CRM systems are made.

Tasks of this master work were accomplished:

- Analysed operation of existing CRM system and RIA technologies.
- Improved existing CRM system by creating dynamic interface, which simplified and accelerated information exchange between department and their clients.
- Tested operation of improved system.
- Carried out an experiment, which confirmed efficiency of dynamic.
- Made recommendations to improve similar systems.

1. Įvadas

Efektyvus klientų valdymas – dar sunkiai suprantama sąvoka dideliai daliai Lietuvos verslo atstovų. Ryšių su klientais valdymo sistemų, dar kitaip vadinamų CRM, diegimas kelia įmonės konkurencinį pranašumą rinkoje, leidžia didinti pardavimus ir įmonės pelną. Užsienio įmonių patirtis parodė, kad tokių sistemų diegimas vidutiniškai 43 proc. kelia darbo efektyvumą, 35 proc. didina pardavimų apimtį, 34 proc. mažina kompanijos išlaidas. Atsakingi asmenys pripažįsta, kad tokių sistemų diegimas padėjo įmonėms augti ir užimti pirmaujančias pozicijas rinkoje. Verslo valdymo, klientų valdymo sistemos – tai ne vien programinė įranga. Tai labiau pardavimo filosofija, pelninga šiandienos investicija.

Viena iš karščiausių tendencijų elektroninėje komercijoje ir mažmeninėje prekyboje yra Web 2.0 taikomųjų programų adaptacija ryšių su klientais valdymo procese. Vienos vairių dydžių ir sferų įmonės tai paprastai vadina protinga el. komercija, o kitos savo ruožtu įvardija tai kaip CRM 2.0. Daugelis pasaulyje lyderiaujančių organizacijų diegia Web 2.0 taikomąsias programas savo klientų-ryšių valdymo sistemose tam, kad klientams bedradarbiavimas būtų patogesnis, greitesnis, dinamiškesnis ir jie pasijustų esantys esminiais elementais visame verslo ryšyje. Panaudojamos priemonės sudomina ir įtraukia pirkėją į vykdomą marketingo procesą. Šios CRM sistemos turi pritaikytus individualiam vartotojui, reikšmingus ir automatizuotus ryšius, kurie pakelia pirkėją į aukštesnį ir asmeninį lygį.

Pastaruoju metu Web evoliucijoje vyksta gilūs pasikeitimai. Mes žengiamo į neturintį precedento vartotojų interfeiso inovacijų periodą, kai Web programuotojai galiausiai gali sukurti tokio lygio Web taikomąsias programas, kokios egzistuoja lokaliuose darbastaliuose. Modernios Web 2.0 taikomiosios programos vis labiau populiarėja įmonių tarpe, nors nebūtinai visi supranta, kas tai iš tikrųjų yra, todėl šia tema vyksta daug diskusijų. WEB 2.0, tai frazė kurią pirmą kartą paminėjo O'Reilly Media [1] 2005 metais ir nurodo kaip antrosios kartos, internetu paremtas paslaugas. Visa tai sudaro visuomeninio tinklo svetainės, "wikis", komunikaciniai įrankiai ir dar daugybė kitų priemonių leidžiančių žmonėms bendrauti ir dalintis informacija internete, anksčiau nežinomais būdais.

WEB 2.0 svetainėje būdingos šios technologijos:

- AJAX ir kitos galingos taikomųjų interneto programų technologijos
- CSS (kaskadinės stilių lentelės)
- Galiojantis XHTML ir/arba mikroformatų naudojimas
- Duomenų publikavimas ir kaupimas RSS formatu
- Tvarkingi ir prasmingi URL adresai
- Internetinių dienoraščių publikavimas
- REST, XML, Webservices, APIs

Anksčiau buvo tokie naujienų puslapiai kaip Delfi arba Omni, dabar naujienas gali rašyti kiekvienas tokiuose puslapiuose kaip blogas.lt, blogger.com, livejournal.com, wordpress.com ir nereikia sukurti galvos kaip kuriami puslapiai. Seniau buvo galima tik skaityti, dabar galima ir rašyti. Prisiminus Britanica arba Encarta - tai brangus malonumas, tačiau dabar turime Wikipedia - nemokamą enciklopediją, kurią rašo šimtai, tūkstančiai žmonių, ir vienintelis atlygis, kurį jie gauna, tai skaitytojų informuotumas.

Seniau internetas buvo išrinktųjų teisė, dabar jis visų, kiekvienas gali kurti interneto turinį, naujienas, prisidėti prie enciklopedijų kūrimo. Visas šias galimybes suteikia moderniosios Web 2.0 taikomosios programos.

Todėl šio darbo **tyrimo sritis** yra raiškaus interneto taikomųjų programų (RIA) kūrimo technologijos ir jų taikymas tobulinant organizacijų CRM informacines sistemas. **Tyrimo objektas** – interneto vartotojo interfeiso kūrimo procesas, leidžiantis pagerinti dinamines vartotojo sąveikos su interneto sistema savybes.

Šiame darbe sprendžiama **problema** – esamos CRM sistemos, naudojamos įmonėse, vartotojams yra nepatogios naudoti, užima per didelę laiko dalį komunikuojant tarpusavyje, yra morališkai pasenusios ir nesuteikia pageidaujamo funkcionalumo bei interaktyvumo. Šiame darbe bus siekiama parodyti, kaip šias problemas gali išspręsti AJAX ir PHP integravimas.

Pirmiausiai ir apsibrėšime, ką mes laikysime dinamiška sistema arba dinamišku internetiniu puslapiu (toliau puslapis). Sistema gali būti dinamiška, jei bent keli jos puslapiai yra dinamiški. **Dinamiška sistema (puslapiu)** šiame darbe laikysime sistemą (puslapį), kurios (-io) daugumos funkcijų veikimas didžiąja dalimi paremtas asinchroniniu bendravimu su serveriu arba, kitaip tariant, naudoja Ajax technologijas. Sistemos kiekvienos funkcijos paprastai nereikia realizuoti, pasinaudojus Ajax technologija, nes tai apsunkina patį kūrimo procesą ir sistemos dizaino tvarkymą. Sistemą galima laikyti dinamiška, jei pagrindinės, dažniausiai naudojamos funkcijos yra sukurtos, naudojant Ajax galimybes. Geriausia nustatyti sistemos funkcijas, kurios realizuotas ne dinamiškai, veikia per daug lėtai. Šias funkcijas tikslinga kiek įmanoma padaryti dinamiškesnėmis. Negalima imti tiesiog ir naudoti Ajax technologijų bet kaip. Tam reikia sukurti dinamiškos sistemos interfeiso patobulinimo rekomendacijas ir šablonus, kurie padės įgyvendinti ne vieną funkciją, o iš karto kelias ar net visas, kurias reikia realizuoti, kad sistema taptų dinamiška.

Darbe **siekiama** pritaikyti RIA taikomųjų programų savybes CRM sistemai, suteikiant vartotojų interfeisui didesnę dinamiškumą ir naudojimo patogumą. Tam tikslui bus tiriamos galimybės pritaikyti analizės metu nustatytus RIA taikomųjų programų privalumus. CRM sistemoje bus įdiegtos raiškaus interneto taikomosios programos, kurių pagrindinis uždavinys – parodyti Web 2.0 taikomųjų programų pritaikymo galimybes įmonės vartotojo interfeiso tobulinime.

Taigi **darbo tikslas** – padidinti CRM sistemų dinamiškumą, išplečiant jas raiškių interneto programų savybėmis ir sudarant rekomendacijas esamoms sistemoms patobulinti.

Darbo uždaviniai:

- Išanalizuoti internetinius projektus įgyvendinančios organizacijos automatizuoto pirkėjų aptarnavimo (klientų-ryšių valdymo) sistemos veikimą ir RIA technologijas;
- Patobulinti esamą CRM sistemą sukuriant dinaminę interneto sąsają, kuri supaprastintų ir pagreitintų informacijos mainus tarp skyriaus ir jo klientų,
- Patikrinti jos veikimą bandomojo diegimo metu,
- Atlikti eksperimentą, patvirtinantį dinaminės sąsajos efektyvumą,
- Pateikti rekomendacijas panašioms sistemoms tobulinti.

Šis magistro darbas atitinka šiandienines tendencijas ir gali būti pagrindu programuotojui, kuris nori pradėti kurti dinamiškas sistemas. Taip pat kai kuriems šis darbas bus paskatinimas naudoti Ajax. Kadangi viskas yra dokumentuota – labai nesunku perprasti sistemos veikimą. Detali dokumentacija padės sistemos ateities tobulinimui. Atsirasi naujos technologijos, tobulės esamos. Ateityje dinamiškų sistemų poreikis vis labiau augs, todėl sukurtos dinamiškos sistemos negalima padėti į „lentyną“ ir užmiršti, o palaipsniui reikės tobulinti.

Sistema buvo ištirta ir nustatyti bei įvertinti jos puslapių dinamiškumo rodikliai, kurie parodė, kad sukurta sistema yra **dinamiška**. Taip pat buvo nustatyta, kad funkcijos, realizuotos pasitelkus Ajax veikia daug greičiau nei funkcijos, kurioms nebuvo naudojamos asinchroninės technologijos. Kad visa tai nustatyti buvo atliekami realūs eksperimentai, kurių metu eksperimento dalyvių atliktas devynių tipinių sistemos veikimo scenarijų įvertinimas pagal 6 dinamiškumo kriterijus parodė, kad buvo pasiektas pageidautas sistemos dinamiškumas. Viskas vyko realiomis sąlygomis. Taip pat buvo vertinamas sistemos puslapių programavimo sudėtingumas, kurį įnešė Ajax naudojimas.

Atlikus eksperimentą, buvo sudarytos RIA taikomųjų programų savybių pritaikymo CRM ir kitoms organizacijų sistemoms rekomendacijos. Šias rekomendacijas tikslinga taikyti ir kitoms įmonių taikomosioms programoms, kadangi RIA technologijos ne tik padidina sąveikos dinamiškumą, bet ir sumažina administravimo sąnaudas. Interneto taikomosios programos diegiamos serveryje, todėl administratorius gali jas tvarkyti vienoje vietoje vieną kartą. Ankstesnės interneto technologijos neturėjo pakankamo raiškumo ir dinamiškumo, todėl nebuvo tinkamos visoms organizacijų sistemoms. RIA technologija suteikia galimybes visą įmonės programinę įrangą realizuoti vieninga interneto technologija.

Atliekant Web 2.0 ir RIA taikomųjų programų analizę, buvo pasinaudota [1], [2], [15] ir [16] literatūros šaltiniais, kuriuose buvo rasta daug naudingos informacijos apie šių taikomųjų programų panaudojamumą, teigiamas ir neigiamas savybes bei teikiamą naudą. Problemos sprendimo metodų literatūros šaltiniuose analizės dalyje buvo remiamasi [3], [4], [7] literatūros šaltiniais.

Projektuojant ir kuriant sistemą, buvo remiamasi [7], [11], [12], [13], [14] literatūros šaltiniais, kuriuose buvo pateikti įvairūs Ajax programavimo metodai, būdai bei taikymo pavyzdžiai kitose sistemose

Analizuojant CRM sistemų trūkumus buvo remiamasi tiek informacijos apie CRM sistemas literatūros šaltiniais [5], [6], tiek nagrinėjant ir išbandant realiai pačias lietuviškas sistemas (realiai išbandytos buvo *Smart CRM*, *Goldmine*, *OpenCRM* sistemos). Gauta informacija leido identifikuoti esminius CRM sistemų trūkumus – nei vienoje iš analizuotų CRM sistemų nebuvo panaudotos Web 2.0 taikomosios programos, dėl ko jos yra vartotojams nepatrauklios naudoti. Šių trūkumu buvo stengiamasi išvengti tobulinamoje sistemoje.

Sukurta magistro darbą sudaro **šešios pagrindinės dalys:**

- 1) RIA taikomųjų programų analizė
- 2) CRM sistemos reikalavimų specifikacija ir analizė
- 3) CRM sistemos tobulinimo projektas
- 4) Eksperimentinė realizacija
- 5) CRM sistemos patobulinimo įvertinimas
- 6) Bendros darbo išvados.

Remiantis šiuo darbu paruoštas straipsnis, kuris pateiktas konferencijai IT 2008.

2. RIA taikomųjų programų analizė

Vykdomos **analizės tikslas** – aprašyti tyrimo sritį, objektą bei esamas problemas, kuo detaliau išsinagrinėti kompiuterizuojamai sistemai keliamus reikalavimus, veiklos dalyvius, veiklos procesus, jų tarpusavio ryšius, duomenų srautus, kuriamos sistemos teikiamus privalumus ir trūkumus kitų atžvilgiu. Taip pat atlikti problemos sprendimo metodų, panašių sistemų, įrankių, architektūros bei rizikos faktorių analizę.

Pagal suformuluotus reikalavimus vėliau bus modeliuojami įvairūs sistemos aspektai, projektuojami sistemos komponentai.

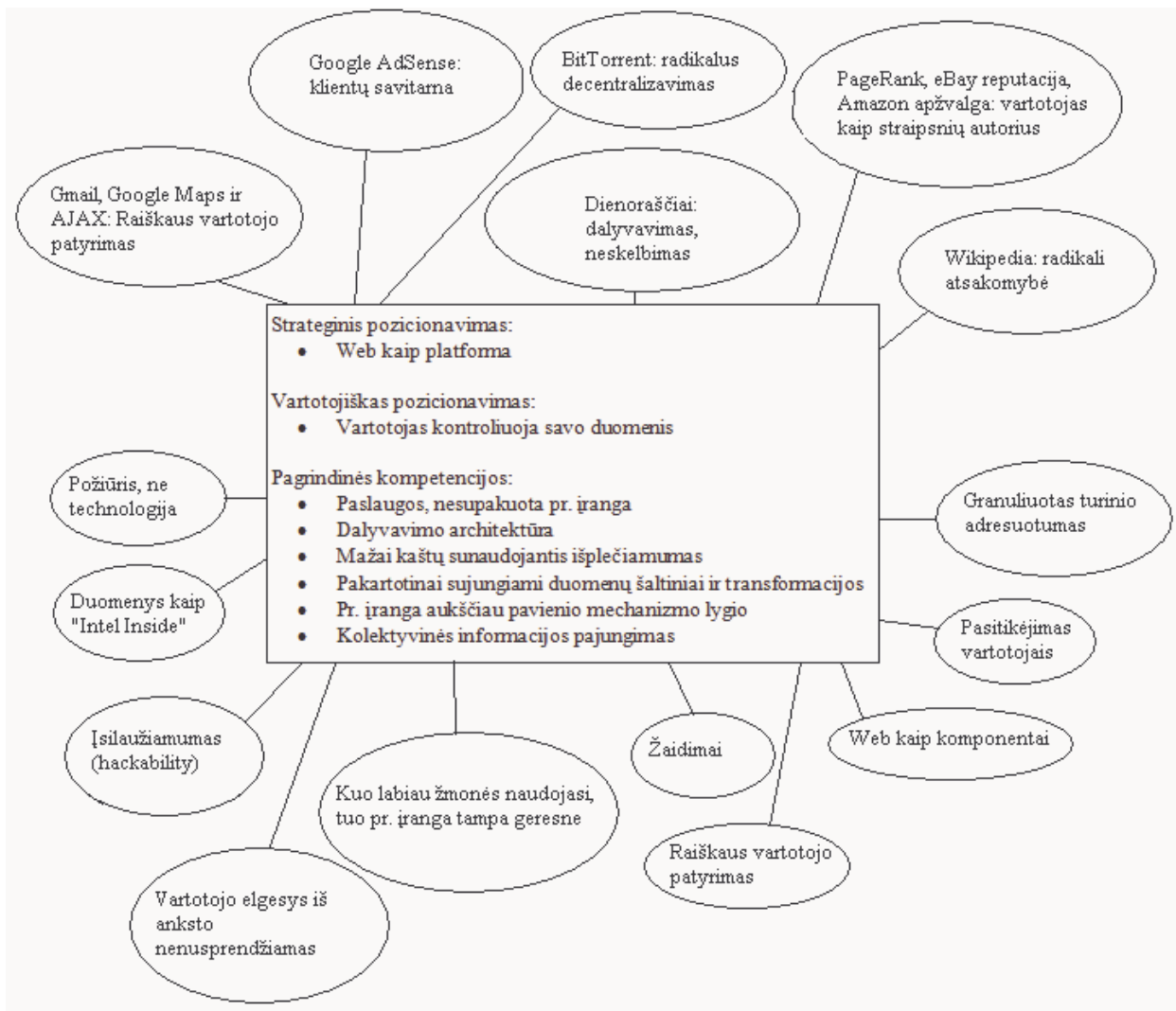
Tyrimo sritis – raiškaus interneto taikomųjų programų (RIA) kūrimo technologijos ir jų taikymas tobulinant organizacijų CRM informacines sistemas.

Darbo objektas - interneto vartotojo interfeiso kūrimo procesas, leidžiantis pagerinti dinamines vartotojo sąveikos su interneto sistema savybes.

Sprendžiama problema – esamos CRM sistemos, naudojamos įmonėse, vartotojams yra nepatogios naudoti, užima per didelę laiko dalį komunikuojant tarpusavyje, yra morališkai pasenusios ir nesuteikia pageidaujamo funkcionalumo bei interaktyvumo. Šiame darbe bus siekiama parodyti, kaip šias problemas gali išspręsti AJAX ir PHP integravimas.

2.1. Web 2.0 analizė

Kaip ir dauguma svarbių konceptų, Web 2.0 neturi griežtos ribos, greičiau turi gravitacinį branduolį. Web 2.0 savybes galima įsivaizduoti, kaip rinkinį principų ir praktinių panaudojimų, kurie kartu susijungia į tam tikrą soliarinę sistemą, kuri vaizduoja kai kuriuos principus, išsidėsčiusius įvairiais atstumais apie šį branduolį, pavaizduotą 2.1 paveiksle.



2.1 pav. Web 2.0 savybės

Timas Orelis savo straipsnyje [1] pabrėžia, kad naujos kartos interneto kompanijomis galima vadinti tik tasias, kurios tenkina šias sąlygas:

- siūlo paslaugas (o ne supakuotą programinę įrangą) su nuo kainos priklausančiu plečiamumu.
- kontroliuoja unikalų, sunkiai atkuriamų duomenų šaltinius, kurie auga priklausomai nuo to, kiek žmonių juos naudoja.
- pasitiki naudotojais kaip pagalbinais programinės įrangos vystytojais.
- išnaudoja kolektyvinį protą.
- didina konkurenciją suteikdami galimybę klientams kurti savo paslaugas.
- kuria programinę įrangą, kuri neapsiriboja vienu prietaisu.
- lengvina naudotojo interfeisus, vystymo modelius ir verslo modelius.

AJAX yra vienas pagrindinių Web 2.0 taikomųjų programų komponentas, tokiuose projektuose kaip *Flickr*, dalis *Yahoo!*, taip pat Google taikomosios programos – *Gmail*, *Orkut*, *Google Maps*, *Google Docs* ir daugelyje kitų. AJAX reiškia Asynchronous Javascript And XML. Iš tiesų, AJAX

vadinama Web 2.0 (naujojo interneto) dalimi. Nors tai ir nėra technologija, tai greičiau tam tikrų technologijų sukomponavimas, tačiau suteikiantis tam tikrų galių.

AJAX leidžia mums atlikti šiuos dalykus:

- elementarus CSS ir XHTML panaudojimas,
- dinaminis duomenų vaizdavimas ir jų ryšiai su DOM,
- manipuliavimas ir keitimasis duomenimis su XMLHttpRequest,
- visa tai apjungia Javascript.

Tradiciniai Web tinklapiai veikia taip: vartotojas savo veiksmais išsiunčia HTTP serveriui užklausa, serveris atlieka veiksmus - paima duomenis, juos apdoroja, praeina pro kelias sistemas, ir tada gražina sugeneruotą Web puslapį vartotojui. Tokia yra Web'o specifika, tačiau nesunku suprasti, jog tokia schema ne visai tinkama programinės įrangos kūrimui. Kol serveris atlieka savo darbus, vartotojas laukia. Ir taip kiekviename žingsnyje, jis turi laukti kol užklausa nukeliauja, serveris atlieka skaičiavimus ir gražina tai ką reikia. Be abejo, šiandien kuriant Web'ą nuo pagrindų, vartotojas neturėtų būt verčiamas laukti. Kodėl viskas turi kraunama iš naujo, jei vartotojo interfeisas užkrautas ir jis realiai nesikeičia? O kodėl išvis vartotojui rodyti, jog siunčiamos užklauskos ar gaunami duomenys?

AJAX schemoje tarp vartotojo ir serverio įterpiamas AJAX varikliukas, kuris ir apsiima išspręsti šias blygybes. AJAX varikliukas paprastai užkraunamas javascript'ų pavidalu, kurie atsakingi ir už vartotojo interfeiso apdorojimą ir komunikavimą su serveriu. Vartotojo bendravimas su serveriu tampa nesinchroniškas - tai yra, nepriklauso nuo pačio ryšio su serveriu. Taigi, vartotojas nebemato tuščio lango, laukdamas kol pasikraus puslapis. Kiekvienas vartotojo, kuris paprastai atlieka HTTP užklausa, veiksmas kreipiasi į AJAX varikliuko funkciją. Veiksmai, kurie nereikalauja susijungimo su serveriu įvykdomi iškart varikliuko lygmenyje. Jei veiksmas reikalauja susijungimo su serveriu, užklauskos atliekamos asinchroniškai, dažniausiai naudojantis XML, ir neįtraukiant į šį tarpą vartotojo.

2.2. Raiškių interneto taikomųjų programų analizė

Terminas raiškaus interneto taikomosios programos (Rich Internet Applications) buvo pristatytas Macromedia šaltinyje [2] 2002 metais tam, kad pažymėti tradicinio darbaltio ir Web taikomųjų programų susivienijimą, stengiantis pasverti abiejų architektūrų privalumus ir išskylančius trūkumus.

Lentelėje (2.1 lentelė) pateiktas darbaltio (įskaitant kliento serverį) ir bendrai priimtų HTML+HTTP Web taikomųjų programų palyginimas, išryškinant potencialą raiškaus interneto taikomųjų programų, kurios paremtos Web skirstymo modeliu, siūlo patobulintus interfeisus ir sumažina komunikacijos išlaidas.

2.1 lentelė Darbastalio ir bendrai priimtų HTML+HTTP Web taikomųjų programų palyginimas

Savybė	C / S, Darbastalis	Web	RIA
Universalaus kliento panaudojimas (naršyklė)	Ne	Taip	Taip
Paprastas kliento įdiegimas	Ne	Taip	Taip
Raiškios sąveikos galimybės	Taip	Ribotos	Taip
Serverio pusės biznio logika	Taip	Taip	Taip
Kliento pusės biznio logika	Taip	Ribota	Taip
Dalinis puslapio atnaujinimas (refresh)	Taip	Ne	Taip
Retos serverio iteracijos (round-trips)	Taip	Ne	Taip
Serveris-klientas komunikavimas	Taip	Ne	Taip
Funkcionavimas atsijungus	Taip	Ne	Taip
Navigavimas klaviatūra	Taip	Taip	Taip

Pavaizduoti RIA privalumai yra tinkami daugeliui užduočių, tokių kaip duomenų rūšiavimas, filtravimas, sudėtingas vaizdų apdorojimas ir t.t. Paprastai klientas užkrauna raiškaus interneto taikomąją programą kartu su tam tikrais pradiniais duomenimis, tuomet ji, komunikuodama su serveriu tik tuomet, kai vartotojui prireikia sekančios informacijos arba kai jis turi išsiųsti duomenis, valdo duomenų interpretavimą ir įvykių apdorojimą. Yra siūloma keletas technologijų RIA kūrimui. Nors ir yra skirtingos daugeliu požiūrių, tačiau jos gali būti klasifikuojamos į keturias plačias kategorijas:

1. Skriptais grindžiamos (Scripting-based), kur kliento pusės logika yra įgyvendinama naudojantis skriptais grindžiamomis kalbomis, tokiomis kaip JavaScript, ir interfeisai yra pagrįsti HTML ir CSS kombinacija;
2. Įskiepiais pagrįstos (Plugin-based), kur pažangus duomenų pateikimas (rendering) ir įvykių apdorojimas yra leidžiamas naršyklės įskiepių interpretuojant XML, bendrų tikslų programas ar media failus (Flash, Flex, Lszlo, Xamlon);
3. Naršykle pagrįstos, kur raiški sąveika yra iš prigimties palaikoma kai kurių naršyklių, kurios interpretuoja deklaratyvias interfeiso apibrėžimo kalbas (XUL);
4. Web pagrįstos darbastalio technologijos, kur taikomosios programos yra parsiončiamos iš Web'io, bet įvykdomos naršyklės išorėje (Java Web Start, Window Smart Client).

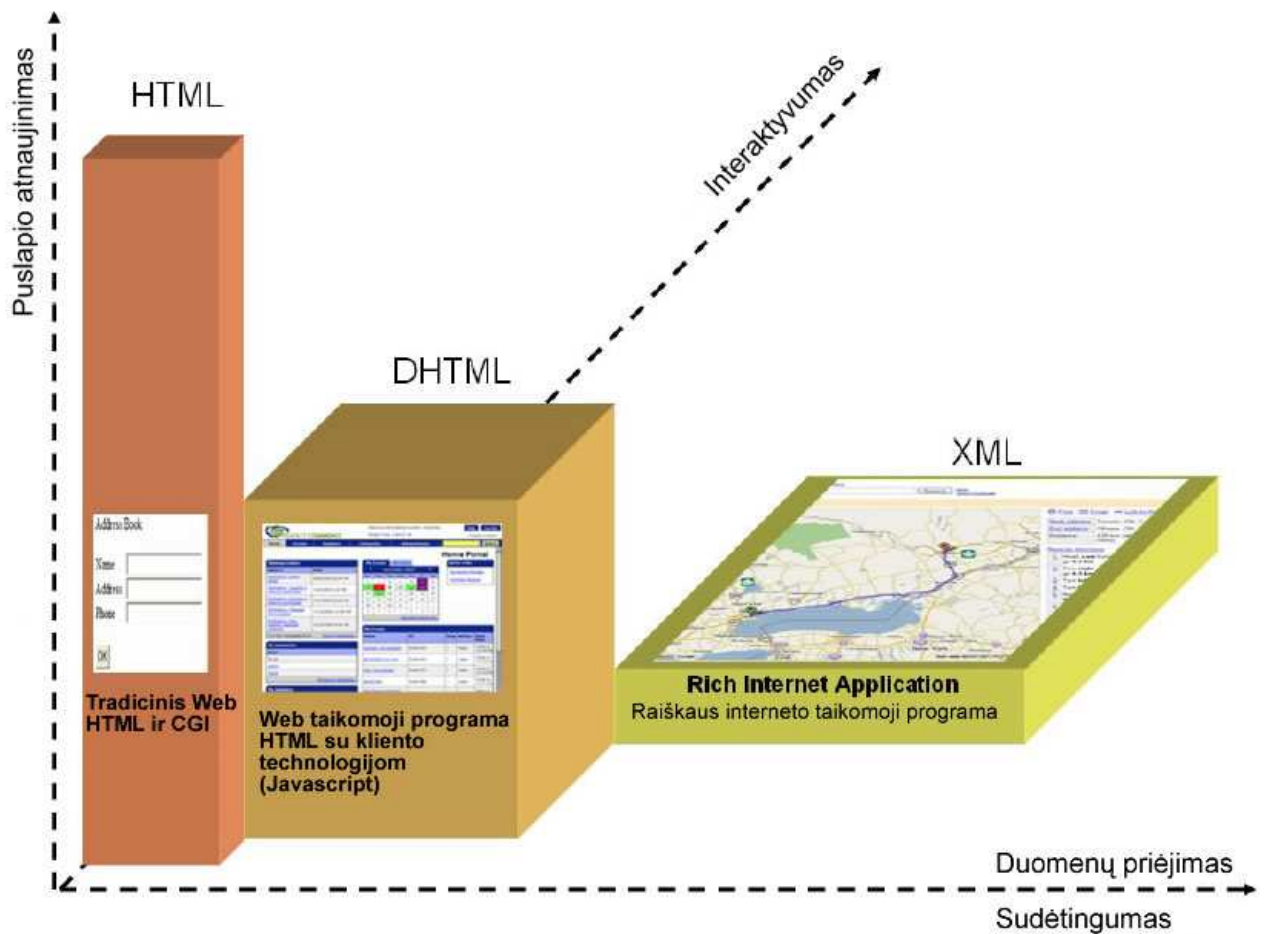
Įprastas savybių rinkinys gali būti aptariamasis charakterizuojant RIA metodą konceptualaus modeliavimo ir kodo generavimo tikslais:

- a) RIA turėtų palaikyti kliento pusės apdorojimą ir sumažinti komunikaciją su serveriu iki minimumo;
- b) Taikomųjų programų duomenys galėtų būti saugomi skirtingo atkaklumo lygiuose tiek kliento, tiek serverio pusėje;

- c) Duomenų apdorojimas (pvz.: sukūrimas, redagavimas, filtravimas) galėtų įvykti tiek kliento, tiek serverio pusėje;
- d) Turėtų būti galimas tiek „online“, tiek „offline“ naudojimas.

Kiti poreikiai liečia rafinuotą interfeiso elgesį, tokį kaip „drag&drop“, animacija, multimedia sinchronizacija ir serverio-kliento komunikacija.

Žemiau grafike (2.2 pav.) pavaizduotas duomenų priėjimo/sudėtingumo kontinuumas Web taikomosiose programose.



2.2 pav. Duomenų priėjimo/sudėtingumo kontinuumas Web taikomosiose programose.

2.3. Dinamiškų puslapių kūrimo metodų literatūros šaltiniuose analizė

Šiame poskyryje nagrinėjami fragmentais, dalykinės srities modeliais grindžiami bei Ajax įgyvendinimo metodai, apžvelgiami jų privalumai ir trūkumai.

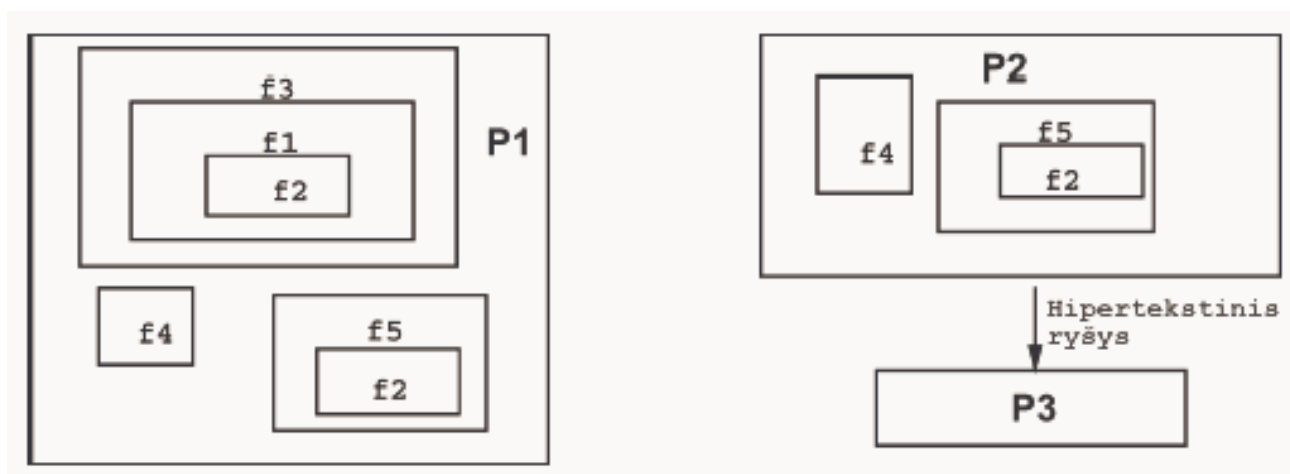
Fragmentais pagrįstas metodas

Šio metodo [3] esmė yra ta, kad sudėtingi Web puslapiai yra konstruojami iš paprastesnių fragmentų (2.3 pav). Fragmentai gali rekursyviai įdėti kitus fragmentus. Ryšiai tarp Web puslapių ir

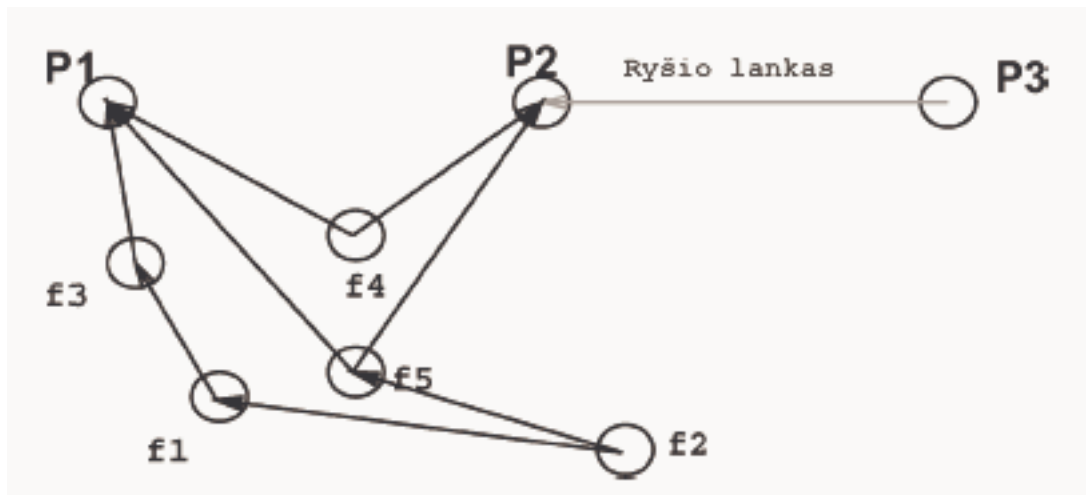
fragmentų yra pavaizduojami objektų priklausomybės grafais. Vartotojai gali atnaujinti Web puslapių turinį modifikuodami šablonus, o sistema automatiškai atnaujina pakeistus puslapius.

Laikoma, kad fragmentas arba puslapis yra atominis, jei jis neturi savyje jokių kitų fragmentų. Priešingai, jei fragmentas turi savyje kitų fragmentų, tai jis yra sudėtinis. Objektas yra arba puslapis, arba fragmentas. Šis metodas efektyvus, kadangi pridėtinės išlaidos sudarant objektą iš paprastesnių fragmentų yra paprastai nedidelės. Naudojant fragmentų metodą yra įmanoma pasiekti žymų sistemos našumo pagerėjimą nekešuojant dinamių puslapių, yra lengviau įterpti informaciją į keletą Web puslapių ir suteikti jiems dinamiškumą. Puslapių rinkiniai turintys panašią informaciją gali būti apdorojami kartu. Dinaminiai Web puslapiai, kurie turi savyje fragmentus yra besąlygiškai atnaujinami bet kuriuo metu, kai fragmentai pasikeičia, taigi nuoseklumas yra pasiekiamas automatiškai. Vienas esminių metodo dalykų yra tai, kad, kuriant Web puslapį, galima atskirti kūrybinį procesą nuo mechaninio.

Kai puslapiai yra konstruojami iš fragmentų, yra svarbu konstruoti f1 fragmentą prieš objekto susidedančio iš f1 sukonstravimą. Tam, kad efektyviu būdu sukonstruot objektus, naudojamas objektų priklausomybės grafas (2.4 pav). Šis grafas gali turėti keletą skirtingų lankų tipų. Įtraukimo lankas pažymi, kad objektas turi savyje fragmentą. Ryšio lankas pažymi, kad objektas turi hipertekstinę nuorodą į kitą objektą.



2.3 pav. Web puslapių konstravimas iš paprastesnių fragmentų



2.4 pav. Objektų priklausomybės grafas

Lankas iš f_4 į P_1 pažymi, kad P_1 susideda iš f_4 . Taigi, kai f_4 pasikeičia, tuomet f_4 turėtų būti atnaujintas prieš P_1 . Lankas iš P_3 į P_2 yra ryšio lankas, kuris pažymi, kad P_2 turi hipertekstinį ryšį su P_3 . Pagrindinė ryšio lankų palaikymo priežastis – išvengti kabančių arba nesuderinamų hipertekstinių ryšių. Šiame paveikslėlyje ryšio lankas iš P_3 į P_2 pažymi, kad publikuojant P_2 prieš P_3 rezultate gausis nutrauktas hipertekstinis ryšys. Taigi, kai abu, P_3 ir P_2 , pasikeičia, publikuojant P_2 versiją prieš P_3 versiją gali būti pateikiama nenuosekli informacija vartotojams, kurie peržiūri atnaujintą P_2 versiją.

Kūrimo laikotarpiu atlikus eilę Web puslapio pakeitimų, jis turi būti publikuojamas. Publikuojant Web puslapį vieninteliu atominiu veiksmu išvengiama nuoseklumo problemų, bet tai gali sukelti publikavimo uždelsimą, ypač jei naujai sukonstruotas puslapis prieš tai dar turi būti taisomas. Augantis (incremental) publikavimas informaciją gali pateikti greičiau, bet taip pat gali sukelti Web puslapių nesuderinamumą. Naudojant fragmentų metodą yra įmanoma pasiekti žymų sistemos našumo ir efektyvumo pagerėjimą.

Dalykinės srities modeliais grindžiami metodai

Dauguma esamų projektavimo modelių yra pagrįsti dalykinės srities modeliavimu ir pagrindinį dėmesį skiria hipermedijos programų projektavimo gyvavimo ciklo architektūros etapui [10]. Visi šie modeliai akcentuoja palapsninį ir interaktyvų projektavimo procesą ir dažniausiai susideda iš keletos ontogonaliųjų modeliavimo lygmenų. Dažniausiai programos projektavimo metu yra naudojami šie modeliavimo lygmenys: konceptualus arba struktūrinis lygmuo (nusakantis informacijos srities struktūra), hiperteksto lygmuo (parodantis kuriamos programos kompoziciją bei navigacijos struktūra), vaizduojamasis arba prezentacinis lygmuo (nusakantis vartotojo sąsają) personalizacijos lygmuo (individualus dizainas), įgyvendinimo arba realizacijos lygmuo. Kiekvienas iš nagrinėjamų modelių skirtingai dengia visus šiuos lygmenis, tačiau dauguma jų akcentuoja ir didžiausią dėmesį kreipia į tris lygmenis – konceptualų,

hiperteksto ir vaizduojamąjį [11]. Konceptualiajame lygmenyje informacijos sritis yra užfiksuojama ir modeliuojama pasinaudojant trimis architektūros technologijomis:

- Esybių sąryšių metodika – informacijos objektai ir duomenų struktūra yra nusakoma esybių bei jų ryšių reikšmėmis.
- Objektiškai orientuota metodika – informacijos objektai yra modeliuojami kaip objektai arba klasės.
- Ontologija pagrįsta metodika – informacijos objektai yra modeliuojami kaip ontologijos klasės.

Įvairių požiūrių ir perspektyvų koncepcija yra naudojama hiperteksto lygmenyje siekiant modeliuoti įvairių tipų programas, nagrinėjant struktūrinio lygmens analizės ir projektavimo rezultatus. Keletas metodų, tokių kaip WSDM, OOHDM ir WebML suteikia galimybę lankstesnėms personalizacijos ypatybėms nagrinėti (turinio, nuorodų, struktūros ir konteksto individualiam pritaikymui).

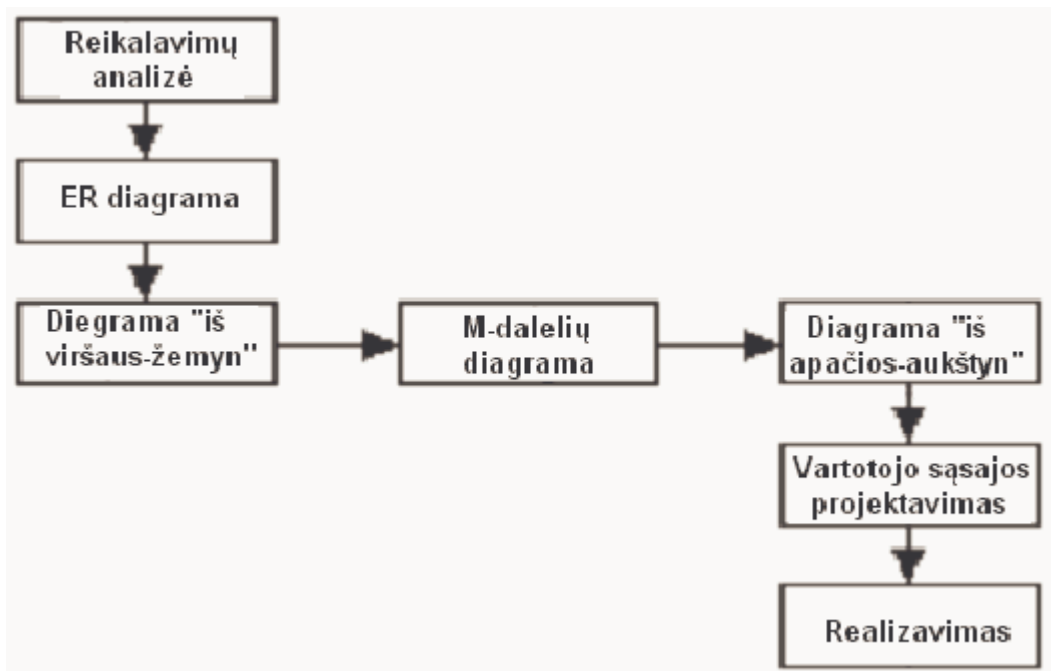
Modeliuojamos taikomosios programos kaip sistemos komponavimas ir navigacijos struktūra yra sudaroma mazgų (puslapių, navigacijos elementų, turinio elementų) ir įvairių ryšių (perspektyvos, struktūrinių, sisteminių bei kt.) tarp jų pagrindu. Navigacijos elementai (mazgų vienetai) yra susieti su koncepciniais elementais (esybėmis arba klasėmis) tam kad būtų galima pateikti informaciją bei duomenis vaizdavimo metu [12].

Nesunku pastebėti kad projektavimo metodai ir modeliai, naudojami modeliuojant internetines programas, turi pakankamai daug panašumų. Pagrindinis jų skirtumas yra nagrinėjamų lygių išsamios analizės apimtis, projektavimo metu suteikiamų galimybių lygmuo skirtinguose veiklos etapuose.

HDM (angl. Hypermedia Design Model) yra vienas iš ilgiausiai naudojamų projektavimo modelių, paremtas esybių ryšių modeliavimu ir padedantis nustatyti didelių ir tik informaciją pateikiančių (bet nepriimančių (tik skaitymui) hipermedijos sistemų sąveikas ir struktūrą. Šis modelis yra tinkamas sritims pasižyminčioms aukštu organizavimo, moduliavimo ir vientisumo lygiu. Naudojant šį modelį pagrindinis dėmesys yra skiriamas hierarchiniam informacijos objektų aprašymui. Šie objektai apibrėžiami kaip esybės sudarytos iš komponentų, savyje talpinančių informacijos vienetų bei navigacines struktūras, neatsižvelgiant nuo jų realizacijos. Navigacinė struktūra yra sudaroma iš perspektyvinių ryšių tarp vienetų, struktūrinių ryšių tarp komponentų, taikomosios programos ryšių tarp esybių, komponentų arba vienetų [9]. Vaizdavimo projektavimas yra sudaromas informacijos vienetų bei karkasų (informacinių vienetų grupių) pagrindu.

RMM (angl. Realationship Management Methodology) yra esybių ryšiais pagrįstas internetinių taikomųjų programų modelis, tinkamas struktūrizuotoms hipermedijos programoms [8]. Pats projektavimo procesas yra suskaidytas į septynis žingsnius (matomus 2.5 pav): esybių bei jų ryšių projektavimas, sluoksnių projektavimas (esybių atributų grupavimas vaizduojant juos mazgais bei vaizdavimo vienetais, vadinamais dalelėmis arba M-dalelėmis (modeliavimo dalelėmis), navigacijos projektavimas (prieigos tipo nustatymas organizuojant nuorodas, meniu, indeksus, supažindinimo kelius), protokolo konvertavimo

projektavimas (verčiant projekto komponentus į fizinius objektus), vartotojo sąsajos projektavimas (vaizdavimo maketavimas), veikiančios sistemos funkcionavimo projektavimas, konstravimas ir testavimas.



2.5 pav. RMM procesų seka

OOHDM (angl. Object Oriented Hypermedia Design Model) yra objektiškai orientuotas projektavimo modelis kuris leidžia specifikuoti hipermedijos programos kaip navigacijos pjūvius konceptualiame modelyje. Projektavimo procesas čia susideda iš keturių lygmenų: konceptualaus modeliavimo, navigacinės sistemos architektūros, abstrakčios sąsajos projektavimas ir realizavimas, prie kurių visai neseniai buvo pridėti reikalavimų rinkimas ir personalizacijos projektavimas. Navigacijos mazgai yra susiejami su konceptualiomis klasėmis, o tik skaitymui skirtų internetinių taikomųjų programų generavimui yra sukurtas specialus CASE įrankis vadinamas OOHDM-Web.

SOHDM (angl. Scenario-based object-oriented Hypermedia Design Methodology) yra kitas objektiškai orientuotas požiūris į internetinių programų modeliavimą. Šis modelis didžiausią dėmesį skiria į procesą orientuotoms hipermedijos sistemoms, kurios padeda palaikyti organizacijoje vykstančius procesus [13]. Scenarijai yra sukuriami nagrinėjant organizacijos veiklą ir tai tampa pagrindu objektiniam ir modeliavimui ir navigacijos projektavimui. Projektavimas susideda iš šešių fazių: aplinkos analizės, objektiškai orientuoto modeliavimo, pjūvių kūrimo, navigacijos sukūrimo, realizacijos architektūros sudarymo, konstravimo.

WDSM (angl. Web Site Design Method) yra į vartotoją nukreiptas projektavimas, kadangi modeliuojant yra panaudojamas vartotojų modelis, sukuriamos vartotojų klasės ir jų ypatybės bei pjūviai. Dizaino procesas susideda iš trijų stadijų: vartotojų modeliavimo, konceptualaus projektavimo, realizacijos architektūros sudarymo, konstravimo.

EORM (Enhanced Object-Relationship Model) – objektiškai orientuotas projektavimo modelis, kurio pagrindinė savybė yra ta, kad čia ryšiai tarp objektų yra apibrėžiami kaip atskiri objektai. Visi šie ryšio bei sąveikos objektai yra išreiškiami kaip atskiros klasės. Metodas yra paremtas trimis karkasais – klasių

karkasu, kompozicijos karkasu, grafiniu vartotojo sąsajos karkasu. Šio metodo savybe taip pat galima pavadinti ankstyvų vartotojo sąsajos projektavimą..

AJAX įgyvendinimo metodai

Siekiant svetainėje išgauti kuo daugiau dinamiškumo yra dažniausiai naudojamas AJAX [4]. Yra keletas skirtingų AJAX kliento pusės varikliuko kūrimo technologijų, iš kurių JavaScript, Java ir Flash yra naudojamos dažniausiai:

1. Java metodas (Asinchroninis Java + XML) – šis metodas paprastai naudoja naršykle pagrįstą lengvo svorio Java varikliuką kliento pusės apdorojimui, tokiam kaip vartotojo interfeiso pateikimas (rendering), dalinio ekrano atnaujinimo atlikimas ir asinchroninis serverio komunikavimas. Šiame metode vartotojo interfeisas gali būti apibrėžtas naudojant XML, tuo tarpu kliento pusės taikomosios programos logika yra suprogramuojama naudojant standartinę Java.
2. JavaScript/DHTML metodas (Asinchroninis JavaScript + XML) - šis metodas paprastai naudoja naršykle pagrįstą lengvo svorio JavaScript biblioteką kliento pusės apdorojimui, tokiam kaip dalinio ekrano atnaujinimo atlikimas ir asinchroninis foninis tinklo komunikavimas. Šiame metode vartotojo interfeisas paprastai yra apibrėžiamas naudojant DHTML, o kliento pusės taikomosios programos logika yra suprogramuojama naudojant JavaScript.
3. Flash metodas (Asinchroninis ActionScript + SWF) - šis metodas kliento pusės apdorojimui paprastai naudoja naršykle pagrįstą Flash varikliuką su ActionScript biblioteka. Šiame metode vartotojo interfeisas paprastai apibrėžiamas naudojant SWF, tuo tarpu kliento pusės taikomosios programos logika yra suprogramuojama naudojant ActionScript.

Kiekvienas iš šių metodų turi savo plusų ir minusų, dėl kurių jie pritakomi skirtingoms taikomosioms programoms. Lentelėje 2.2 pavaizduotos metodų stiprybės ir silpnybės.

2.2 lentelė AJAX įgyvendinimo metodų palyginimas

Metodai	Privalumai	Trūkumai
Java	<ul style="list-style-type: none"> • Didelis našumas ir patikimumas • Lengvas palaikymas dėl objektiškai orientuoto programavimo • Tinkamas Java programuotojams 	<ul style="list-style-type: none"> • Reikalauja Java programavimo žinių • Egzistuojančią CRM sistemą reiktų kurti iš naujo
DHTML/JavaScript	<ul style="list-style-type: none"> • Gerai pritampa prie egzistuojančių HTML Web svetainių/Web taikomųjų 	<ul style="list-style-type: none"> • Šis metodas gali būti problematiškas didelės aprėpties biznio taikomosiuose programose dėl

	programų <ul style="list-style-type: none"> • Tinkamas DHTML Web programuotojams • Nebrangus būdas egzistuojančiai taikomajai programai suteikti daugiau dinamiškumo 	JavaScript programavimo ir palaikymo sunkumų
Flash	<ul style="list-style-type: none"> • Patrauklios animacijos • Tinkamas grafikos projektuotojams 	<ul style="list-style-type: none"> • ActionScript kodas yra sunkiai palaikomas ir nėra suprojektuotas komandiniam darbui • Veikimo ir funkcionalumo apribojimai • Egzistuojančią CRM sistemą reiktų kurti iš naujo

Java metodas pasižymi geru funkcionalumu dėl binarinio vykdymo, multi siūlinio (multi-threaded) skaičiavimo ir Java virtualios mašinos industrinės stiprybės. Taikomosios programos, parašytos Java kalba, yra lengvai kuriamos ir palaikomos dėl objektiškai orientuotos programavimo prigimties. Šis metodas labiausiai tinka programuotojams, turintiems Java programavimo įgūdžių.

DHTML/JavaScript metodas labai gerai tinka egzistuojančiose HTML Web taikomose programose. HTML programuotojui yra lengviau programuoti su JavaScript ir tai yra ne toks brangus būdas egzistuojančiai taikomajai programai suteikti daugiau interaktyvumo. Paprastai šis metodas reikalauja parsisiųsti JavaScript bibliotekų, tam kad kompensuoti skirtingas kliento naršyklės galimybes. Gauta taikomoji programa bus tiek raiški, kiek tai leis naršyklė. Šis metodas gali būti problematiškas didelės aprėpties biznio taikomose programose dėl JavaScript programavimo ir palaikymo sunkumų, bei DHTML funkcionalumo apribojimų.

Flash metodas yra geras taikomosioms programoms, kurioms reikia patrauklaus vaizdo ir pojūčio. Tai natūralus pasirinkimas grafikos dizaineriams. Flash programuotojams suteikia galimybę kurti labai raiškų vartotojo interfeisą. Šiuo metu, Macromedia yra nusitaikiusi į tris Flash panaudojimo atvejus: valdomas pardavimas, valdomos formos ir pradinė vartotojo apdaila arba duomenų vizualizavimo taikomosios programos. Flash taikomosios programos gali pasiekti Java ir .NET objektus, serveryje veikiančius asinchroniškai. Pagrindiniai šio metodo apribojimai yra ActionScript ir Flash varikliuko ribotos galimybės.

Išvados: aukščiau išnagrinėti formalizuoti problemos sprendimo metodai. Dauguma dalykinės srities modeliais grindžiamų metodų yra skirti modeliuoti tik skaitymui skirtas web svetaines (statines), o tai yra didelis minusas, kadangi numatoma kurti sistema yra skirta tiek duomenų peržiūrai, tiek dinaminiam jų redagavimui. Visi šie modeliai turi lygmeninę aplinkos, kuriai yra kuriama internetinė taikomoji programa, nagrinėjimo sistemą. Taip pradžioje yra kruopščiai išnagrinėjama dalykinė sritis, vėliau sudaromas svetainės

turinio bei navigacijos, o taipogi realizacijos planas. Tačiau visi šie modeliavimo lygmenys yra silpnai susiję tarpusavyje, nėra konkretaus sektinumo, kada iš vieno lygmens pereinant į kitą būtų jaučiama aiški siekiamų rezultatų taša. Visų šio tipo metodų privalumas yra gera dalykinės srities analizė, tačiau jos rezultatai nėra pilnai išnaudojami, nepakanka tiesiog sudaryti sistemos prototipo klases ar esybes, kurios smarkiai kinta pereinant iš vieno lygmens į kitą. Naudojant fragmentų metodą yra įmanoma pasiekti žymų sistemos našumo ir efektyvumo pagerėjimą.

Pasirinktas JavaScript/DHTML AJAX įgyvendinimo metodas (Asinchroninis JavaScript + XML) dėl to, kad gerai pritampa prie egzistuojančių HTML Web taikomųjų programų (automatizuota pirkėjų aptarnavimo sistema realizuota šiuo pagrindu) bei yra tinkamas DHTML Web programuotojams.

2.4. CRM sistemų analizė

Yra sukurta daug ryšių su klientais valdymo sistemų dar vadinamų CRM (Customer Relationship Management), čia bus bandoma apžvelgti kai kurias jų ir palyginti tarpusavyje.

Plačiausiai naudojamos ir galingiausios komercinės lietuviškos CRM sistemos: *Smart CRM*, *AIVA 9001*, *Goldmine*, *OpenCRM*, *Microsoft CRM*.

Panagrinėkime keletą jų detaliau:

OpenCRM - tai pakankamai galinga ir paprastai naudojama programa. Programa paplitusi pasaulyje. Yra anglų, prancūzų, ispanų, vokiečių ir kitų kalbų versijos. Įgyvendintas klientų filtravimas pagal daug parametrų. Konfigūruojami prioritetai ir būsenos. Yra užduočių priminimas el.paštu, aliarmų nustatymas, duomenų eksportavimas į MS Exel ir pdf formatus, kas man labai patiko ir ruošiuosi įgyvendinti savo sistemoje. Mobilus struktūra: galimas SMS, PALM, POCKET ir kitų technologijų panaudojimas.

Goldmine - šioje sistemoje įdiegtos galimybės valdyti kontaktus-klientus, valdyti darbo laiką – kalendorių, pardavimų automatizavimas, projektų ir pardavimo projektų valdymas. Galimybė naudoti GoldMine standartines ataskaitas, galimybė susikurti savo ataskaitas, galimybė siųsti masinius e-mail, fax ir laiškus naudojant šablonus. Ir tai tik maža sistemos siulomų funkcijų dalis. Tai išties labai funkcionali sistema, iš kurios galima pasisemti idėjų kuriamam darbui. Tačiau dėl sistemos perkrovimo galybe funkcijų, ją yra ganėtinai sunku valdyti. Tačiau ir čia gelbsti detali naudojimosi instrukcija.

Smart CRM – ši sistema taip pat nusileidžia prieš tai nagrinėtoms sistemoms. Yra daug įvairių naudingų modulių. Vieni idomesnių - hualterinė programa "Sąskaitos", paslaugų formavimas ir užsakymas, virtuali piniginė, WAP palaikymas, elektroniniai atvirukai, reklamos internete demonstravimo sistema (banner), kūno masės indekso skaičiuoklė, partnerių sistema, transportų kainų skaičiuoklė ir kt. Tačiau vienas dalykas nuliūdino – labai jau skurdi bazinė sistemos komplektacija.

Išvada: kaip matome šios sistemos yra tikrai labai plačių galimybių, turi daug papildomų modulių, tačiau yra gana brangios ir firmoms – rinkos naujokėms sunkiai įkandamos. Be to, nei vienoje iš analizuotų CRM sistemų nebuvo panaudotos Web 2.0 taikomosios programos, dėl ko jos yra vartotojams nepatrauklios naudoti. Todėl įmonė, kuriai buvo sukurta mini CRM sistema, nutarė suteikti galimybę patobulinti klientų-ryšių valdymo sistemą, sukuriant dinaminę interneto sąsają.

2.5. RSS skaitytuvų analizė

Pradedant kurtis Web 2.0 bendrovėms W3C sukūrė naują programavimo kalbą - XML. Ji greitai išpopuliarėjo, nes buvo labai paprasta. Tuomet atsirado RSS (Really Simple Syndication) - naujienu publikavimo kalba, kurioje labai paprastai buvo pateikiamas naujienos pavadinimas, data ir turinys. Ir pasitvirino sena tiesa - genialūs yra paprasti dalykai - RSS labai išpopuliarėjo. Žmonės pradėjo kurti įvairius naujienu skaitytuvus ir publikuoti savo naujienas RSS formatu - tai buvo paprasta ir paprastai skaitoma. Šiuo metu RSS naujienu feed'ų (naujienu tiekimo kanalų) yra apie viską - patiekalus iš šokolado, Lietuvos žvaigždžių naujienas ar sąmokslų teorijas. Toliau bandoma apžvelgti keletą internete esančių RSS skaitytuvų:

1. *Bloglines* - vienas seniausių RSS skaitytuvų. Jį sukūrė kompanija Ask Jeeves. Veikia jis paprastai ir aiškiai, dėl elementarios sistemos beveik neturi spragų, todėl daug kas pasaulyje jį įvardija kaip "mylimiausią" naujienu skaitytuvą. Registracija nėra trumpa, nes čia prašome įvesti ne tik el. pašta - kitaip nei kai kuriuose kituose skaitytuvuose. Visgi, kad ir kaip seniai ši sistema būtų sukurta, ji vis dar turi galimybių, kurių neturi kitos. Viena jų - galimybė pačiam rašyti blog'ą (dienoraštį) ir turėti savo RSS pašarinę liniją. Tačiau pats RSS skaitymas apkrautas - tam naudojami polangiai (frames), visi įrašai kraunami iš karto o navigacija tarp RSS kanalų sudėtinga.

Išvada: Nors ir pasenusi, ši sistema turi daug mylėtojų (matyt, užsiregistravusių tuomet, kai sistema buvo vienintelė rinkoje), tačiau žvelgiant į bendrą paslaugų asortimentą nėra verta tapti geriausia.

2. *Pluck* - senas ir nelabai besiskiriantis nuo anksčiau minėto. Tačiau visgi, keletas pliusų yra. Visų pirma, mažiau apkrautas ir malonesnis dizainas. Toliau - nėra reklamų. Nors kaip ir anksčiau minėtas, jis naudoja polangius, o RSS skaitymas visiškai keistas - paprastai pašarinės linijos neįsidėsi, jos atidaromos naujame lange.

Išvada: Laikui bėgant RSS skaitytuvai tobulėja - šiame atsiranda žymelių sistema. Bet visgi, sistema dar nepatogi.

3. *Rojo* - pagaliau baigėsi polangiai. Šis skaitytuvas tikrai malonus akiai ir paprastesnis naudoti - dizainas labai paprastas ir neapkrautas, naujienu skaitymas patogus, kaip ir naujų linijų įdėjimas. Yra galimybė rūšiuoti linijas, peržvelgti ta pačia tema rašančias kitas linijas ir kitos malonios galimybės. Didelių trūkumų sistemoje nėra.

Išvada: Sistema tikrai maloni ir viena geriausių tiems, kas naudoja naršyklės, kuriose neveikia kai kurie skriptai - pavyzdžiui netscape. Tačiau visos jos galimybės yra ir AJAX sistemose.

4. *My Yahoo!* - šis RSS skaitytuvas neabejotinai tiks tik tiems vartotojams, kurie nenori judėti kur nors iš Yahoo! puslapio. Jame yra užtektinai papildomų modulių, o pats RSS skaitymas puikiai atlieka savo užduotį. Labai paprastas naujų RSS linijų pridėjimas - nežinia kodėl dauguma puslapių įsideda tik "Add to My Yahoo!" mygtuką. Viena kiek labiau išskirtinė savybė - galimybė keisti išvaizdą. Čia paliekama daug laisvės ir susikurti savo stilių čia išties paprasta.

Išvada: Sistema tiks tiems, kas tingi judėti iš Yahoo, tačiau jei neesate Yahoo fanas tikrai rasite geresnį RSS skaitytuvą.

5. *Google Reader, Google Personalized Homepage* - Google sukūrė du RSS skaitytuvus. Pirmasis - Google Reader - skirtas skaityti tik RSS naujienas. Jame nėra papildomų modulių, tačiau valdymas išties pritrenkiantis, o visas ten esantis AJAX įspūdingas. Yra galimybė pasižymėti patikusius straipsnius, naujų naujienų linijų paieška ir t.t. Deja, sistema yra kiek kvailoka, mėgsta nulūžti skaitydama sudėtingesnius RSS kanalus, o naujų linijų įkėlimas neretai užstringa. Vienas naujų technologijų specialistas Google Reader pavadino „kreivu“. Nepripratusiems dirbti su AJAX technologijomis šis skaitytuvas pasirodys per sudėtingas. Antras - Google Sukonfigūruojamas Pagrindinis Puslapis. Jis RSS naujienas skaito ne taip įmantriai, tačiau veikia paprasčiau. Galima lengvai keisti skirtingų naujienų kanalų vietą ekrane, yra tikrai daug papildomų modulių, o vieną kitą galima nesunkiai susikurti pačiam. Mėgstantiems Google paiešką tai tikrai vienas geresnių sprendimų.

Išvada: Jei vartotojas mėgsta Google ir yra įpratęs jį turėti kaip pagrindinį savo naršyklės puslapį, tai gali šį skaitytuvą naudoti.

6. *Eskobo* – platus AJAX panaudojimas. Galimybė keisti išvaizdą, labai patogi web paieška, paprasta pridėti naujus RSS kanalus, RSS naujienos peržiūra jos dar nepakrovus, naujienų skaitymas tiesiai eskobo puslapyje - visa tai leidžia labai paprastai daryti AJAX. Tai jau naujesnės kartos RSS skaitytuvas, kuris tikrai patiks mėgstanitiems funkcionalumą ir paprastumą. Didelių trūkumų nelabai yra, tačiau palyginus su kitais AJAX skaitytuvais šiame praktiškai nėra papildomų modulių, kurie labai praverčia.

Išvada: Tai tik vienas iš AJAX skaitytuvų, tačiau jis vertas dėmesio.

7. *Goowy* - labai išskirtinis projektas. Registracija ilga, tačiau tai ką vartotojas gauna po jos yra verta keliolikos minučių. Visų pirma - 2GB dydžio el. paštas. O jeigu vartotojas naudojasi Yahoo! ar Gmail Goowy, net perkels visus kontaktus. Toliau - kalendorius, adresų knygutė, orai, žaidimai, dienoraštis ir t.t. Deja, visa tai kraunasi ypatingai ilgai. Šis projektas turi ir dar

vieną ypatingą savybę - darbalaukio modulį. Į savo kompiuterį galima parsisiųsti specialią programą, kuri perkels Goowy ant vartotojo ekrano.

Išvada: Jei vartotojas nori kažko ypatingo ir galingo ir turite greitą internetą - goowy pats tas.

RSS skaitytuvų stiprybių ir silpnybių palyginimui pateikta 2.3 lentelė.

2.3 lentelė Panašių sistemų stiprybių ir silpnybių palyginimas

Sistema	Plusai	Minusai
<i>Bloglines</i>	<ul style="list-style-type: none"> • Paprasta ir užtikrintai veikianti sistema. • Galimybė rašyti savo dienoraštį. • Galimybė keistis pašarinėmis linijomis su kitais nariais. 	<ul style="list-style-type: none"> • Varginanti polangių sistema. • Nekeičiamas dizainas. • Yra keletas reklamų.
<i>Pluck</i>	<ul style="list-style-type: none"> • Malonus dizainas. • Paprasta registracija. • Žymelių (bookmarks) sistema. 	<ul style="list-style-type: none"> • Kvailokas naujienų skaitymas. • Sunku įsidėti naują pašarinę liniją.
<i>Rojo</i>	<ul style="list-style-type: none"> • Gražus ir paprastas dizainas. • Greita registracija. • Susijusių straipsnių peržiūra. • Naujienų išskleidimas nenaudojant polangių. 	<ul style="list-style-type: none"> • Didelių minusų nėra, tačiau AJAX sistemos (jos paminėtos vėliau) yra tobulesnės.
<i>My Yahoo!</i>	<ul style="list-style-type: none"> • Paprasta įkelti naujas linijas. • Galimybė keisti išvaizdą. • Papildomi moduliai. 	<ul style="list-style-type: none"> • DAUG reklamų. • RSS naujienos skaitomos originaliame puslapyje.
<i>Google Reader</i>	<ul style="list-style-type: none"> • Galimybė žymėti patikusius straipsnius. • Labai įmantri sistema. 	<ul style="list-style-type: none"> • Per sudėtingas paprastiems vartotojams. • Dažnai lūžta.
<i>Google Personalized Homepage</i>	<ul style="list-style-type: none"> • Konfigūruojama skydelių pozicija ekrane. • Daug papildomų modulių. 	<ul style="list-style-type: none"> • Ilgokai kraunasi.
<i>Eskobo</i>	<ul style="list-style-type: none"> • Konfigūruojama išvaizda. • Konfigūruojama skydelių pozicija. • Patogus RSS naujienų skatymas, paieška internete ir naujienų pridėjimas. 	<ul style="list-style-type: none"> • Keletas problemų su RSS atkodavimu.

Goowy	<ul style="list-style-type: none"> • Krūvos ypatingų savybių. • 2GB pašto dėžutė. • Darbalaukio modulis. 	<ul style="list-style-type: none"> • Labai ilgai kraunasi.
-------	---	---

2.6. Architektūros ir galimų įgyvendinimo priemonių variantų analizė

Pagal savo kilmę Web buvo sukurtas HTML dokumentų skaitymui. Klasikinės Web taikomosios programos modelis yra perėmęs “spausk, lauk ir atnaujink” vartotojo interakcijos paradigmą ir sinchroninės “užklausa/atsakymas” komunikacijos mechanizmą (2.6 pav):

1. “Spausk, lauk ir atnaujink” paradigma: naršyklė atsako į vartotojo veiksmą atmesdama esamą HTML puslapį ir išsiųsdama HTTP užklausą atgal Web serveriui. Serveris užklausą apdoroja ir gražina naršyklei naują HTML puslapio turinį. Naršyklė atnaujina ekraną ir parodo naują HTML puslapį.
2. Sinchroninės „užklausa/atsakymas“ komunikacijos modelis: naršyklė visada inicijuoja užklausas, tuo tarpu Web serveris paprasčiausiai atsako į šias naršyklės užklausas. Web serveris niekada neinicijuoja užklausų – komunikacija visuomet inicijuojama viena kryptimi. „Užklausa/atsakymas“ yra sinchroninis ciklas, kurio metu vartotojas neprivalo laukti.

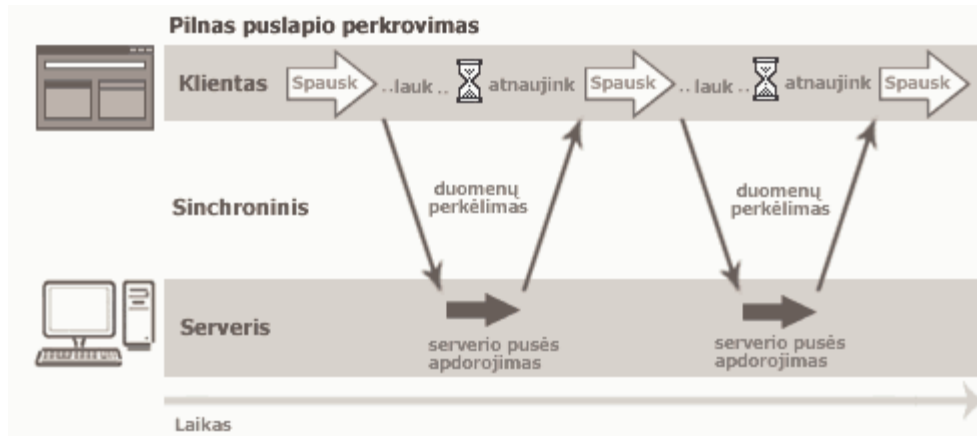
Tačiau šios dvi pagrindinės klasikinio Web modelio elgsenos yra netinkamos programinės įrangos taikomosios programoms. Šiame kontekste klasikinis Web taikomųjų programų modelis sukelia daug problemų:

- lėtas vykdymas dėl “spausk, lauk ir atnaujink”;
- operacijos konteksto praradimas puslapio atnaujinimo metu;
- pernelyg didelė serverio apkrova ir pralaidumo suvartojimas dėl bereikalingų puslapio atnaujinimų;
- trūksta dviejų krypčių, realaus laiko komunikacijos sugebėjimų serverio inicijuotiems atnaujinimams atlikti.

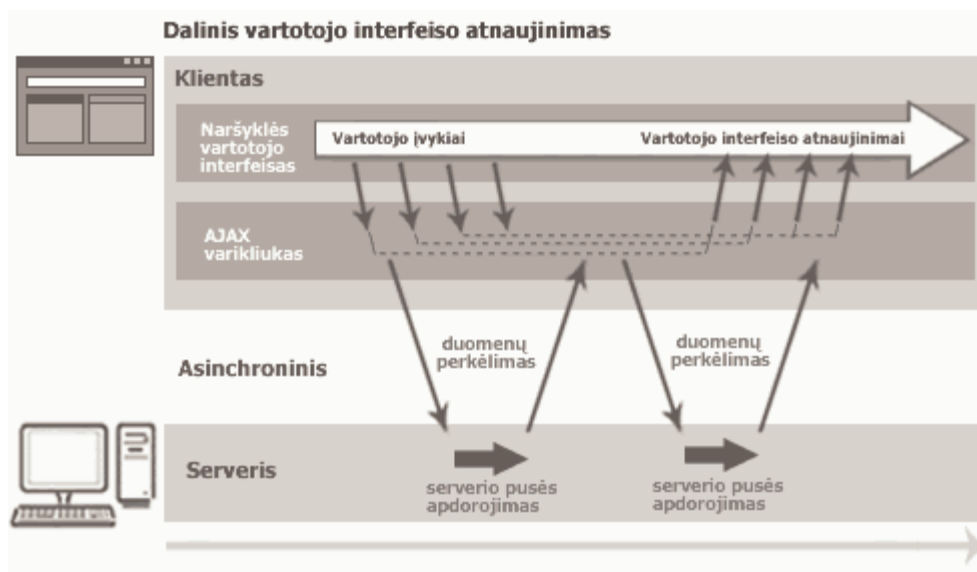
Viso to rezultate turime lėtas, nepatikimas, mažo produktyvumo ir neefektyvias Web taikomasias programas. Tam, kad pasiekti geresnes charakteristikas, daugiau interaktyvumo ir efektyvesnes Web taikomasias programas, naudojamas AJAX taikomųjų programų modelis (2.7 pav):

1. “Dalinis ekrano atnaujinimas” pakeičia “spausk, lauk ir atnaujink” modelį: vykstant vartotojo interakcijoms AJAX pagrįstose taikomosiose programose, atnaujinami tik naują informaciją turintys vartotojo interfeiso elementai. Likę interfeiso elementai išlieka nepakitę. Šis “dalinio ekrano atnaujinimo” sąveikos modelis ne tik įgalina nenutrūkstamo veikimo kontekstą, bet taip pat padaro įmanoma netiesinę darbų seką.

2. Asinchroninė komunikacija pakeičia sinchroninį “užklausa/atsakymas” modelį: AJAX pagrįstose taikomose programose “užklausa/atsakymas” gali būti asinchroninis, atskiriant vartotojo ir serverio interakcijas. To pasekoje, vartotojas gali toliau naudoti taikomąją programą be pertraukimų, kol kliento programa užklausia informacijos iš serverio. Kai iš serverio gaunama nauja informacija, atnaujinamos tik su šia informacija susijusios vartotojo interfeiso dalys.



2.6 pav. Klasikinis Web taikomųjų programų modelis



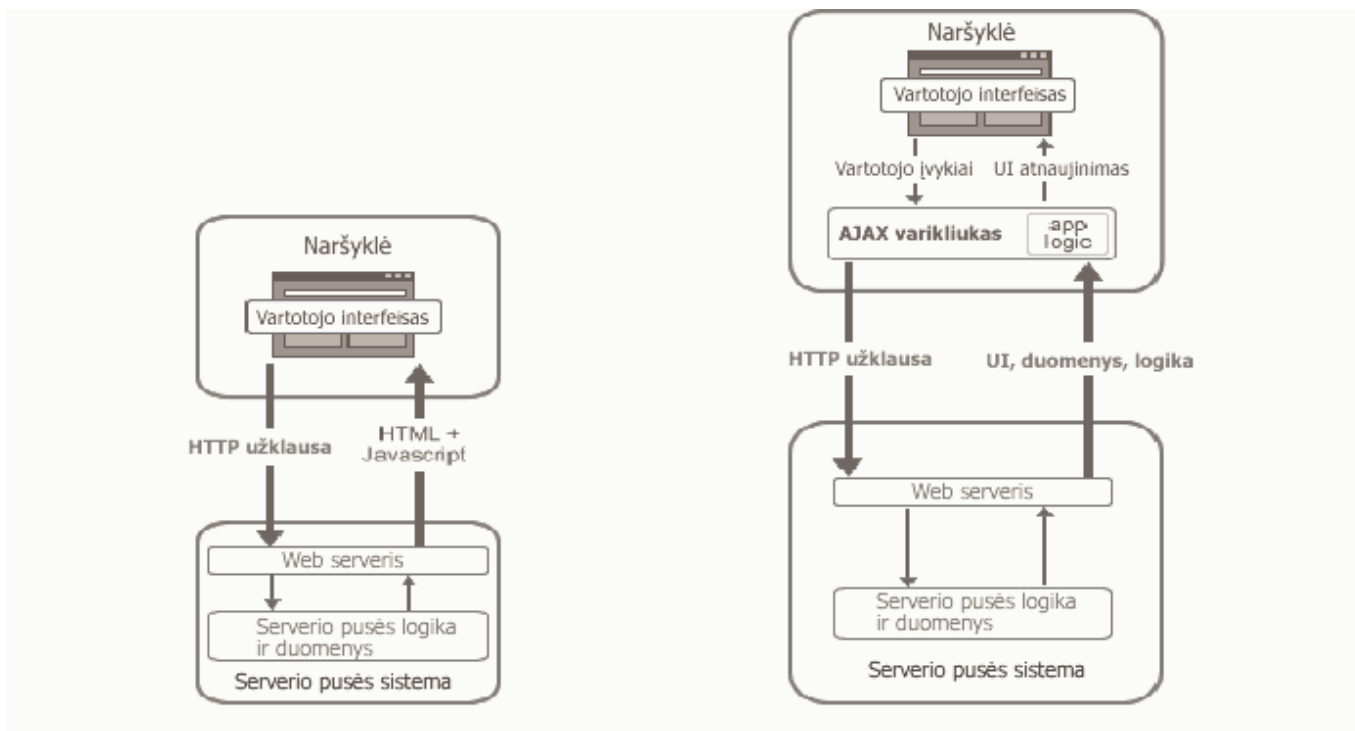
2.7 pav. AJAX pagrįstas Web taikomųjų programų veikimo modelis

Programinės įrangos architektūros požiūriu, AJAX skiriasi nuo klasikinės Web taikomųjų programų architektūros, atsirandus kliento pusės varikliukui, šiais atžvilgiais:

1. Kliento pusės varikliukas naudojamas kaip tarpininkas tarp vartotojo interfeiso ir serverio;
2. Vartotojo veikla priveda prie programų kreipimosi į kliento pusės varikliuką, vietoj puslapio užklauskos tiesiai iš serverio;
3. XML duomenų perdavimas tarp serverio ir kliento pusės varikliuko.

Kliento pusės varikliukas yra AJAX modelio pagrindas. Be šio varikliuko, kiekvienas vartotojo įvykis turi grįžti atgal į serverį apdorojimui. Vartotojo interakcija yra tvirtai surišta su serverio komunikacijomis - kliento pusės varikliukas panaikina šią priklausomybę. Šis varikliukas, dirbdamas Web naršyklėje, suteikia naršyklei papildomų sugebėjimų, kad atlikti „dalinį ekrano atnaujinimą“ vietoje „pilno puslapio atnaujinimo“.

Klasikinio Web ir AJAX architektūrų palyginimui pateiktas 2.8 paveikslas.



2.8 pav. Klasikinio Web ir AJAX architektūrų palyginimas

Galimų AJAX įgyvendinimo priemonių variantų analizei [5] pasirinkti šiuo metu populiariausi AJAX karkasai ir įrankiai (2.5 lentelė):

- Prototype with Script.aculo.us
- Dojo 0.4
- Direct Web Remoting (DWR) 2.0
- Yahoo UI Library (YUI) 0.12
- Google Web Toolkit (GWT) 1.2.22

Šie įrankiai bus vertinami pagal pagrindines tris kategorijas:

1. Įdiegimo ir administravimo procedūros;
2. Kūrimo aspektai: API gausumas, praplečiamumas, daugialypis serverio kalbų palaikymas
3. Palaikymo ir kokybės aspektai: karkaso užbaigtumas, stabilumas, dokumentacijos buvimas.

Palyginimų matricos legenda 2.4 lentelė:

2.4 lentelė Palyginimų matricos legenda

Legenda	
5 – Puiku. Dauguma įgyvendinimo aspektų yra nuostabūs.	2 – Žemiau vidutiniško. Karkasas tik vos palaiko savybę, bet išlaiko normą.
4 – Labai gerai. Kai kurie įgyvendinimo aspektų yra nuostabūs.	1 – Silpnai. Karkasas labai menkai įgyvendina savybę.
3 – Vidutiniškai. Karkasas palaiko savybę.	N/A. Karkasas savybės nepalaiko.

2.5 lentelė Palyginimų matricos lentelė

	Prototype with Scrip.aculo.us	Dojo	DWR	YUI	GWT
Diegimas					
Pakavimas ir išdėstymas (Packaging and Distribution)	4	4	3	5	5
Naršyklės išdėstymas (Browser Deployment)	4	4	4	4	4
Kūrimas					
Integracija su kūrimo karkasais (Integration with Development Frameworks)	5	4	N/A	3	1
Web Widgets būvimas (Availability of Web Widgets)	5	3	2	5	4
XML/DOM paslaugos (Utilities)	4	3	2	4	4
Cross-Browser suderinamumas	4	3	3	4	4
Praplečiamumas (Extensibility)	5	3	4	4	4
Malonus JavaScript palaikymo nuvertėjimas (Graceful Degradation for JavaScript Support)	3	3	3	3	N/A
Palaikymas					
Dokumentacija	4	3	3	5	5
Pavyzdžiai	4	2	3	5	5
Trečios šalies dokumentacija	4	4	2	2	2

Iš visų išnagrinėtų karkasų labiausiai išbaigtas pasirodė *YUI library*. Jis siūlo subalansuotą palaikymą, visų dažnai pasitaikančių užduočių, reikalingų AJAX taikomosioms programoms, pilną

dokumentacijos rinkinį, puikius pavyzdžius, gerai pripažintą diegimą ir naujų versijų paprogrames, profesionalų kokybės palaikymą.

Kai kuriais atvejais *Prototype with Script.aculo.us* yra išties puikus lyginant su kitais karkasais. Pavyzdžiui, jo JavaScript kalbos praplečiamumas neturi sau lygių. Tai plačiai naudojamas karkasas, turintis gerą pripažinimą Web bendruomenėje.

Dojo turi geriausią serverio pusės kalbų (kitų nei Java) palaikymą. Jis taip pat labai susikcentravęs ties prieinamumo savybėmis. Šio karkaso dokumentacija, lyginant su kitais įrankiais, yra labai kukli.

DWR yra kūrybingas Java kalba pagrįstas karkasas, turintis stiprų integracijos su serverio resursais palaikymą. Tačiau jam trūksta tipinių AJAX savybių, tokių kaip interaktyvūs “widgets” ir dinaminiai komponentai rasti *YUI* ar *Prototype with Script.aculo.us* karkasuose.

Galiausiai *Google Web Toolkit*. Google sukūrė, paruošė dokumentaciją ir gerai atlieka produkto palaikymą. Jo pagrindinis apribojimas – palaiko tik Java kalbą. Taigi, kol Java programuotojai galės su juo labai lengvai dirbti, tai ne Java programuotojams teks arba šią kalbą išmokti, arba rasti kitą sprendimą.

2.7. Siekiamų CRM sistemos savybių apibrėžimas

2.2. poskyryje buvo išskirtos RIA taikomosioms programoms būdingos savybės: universalus kliento panaudojimas (naršyklė), paprastas kliento įdiegimas, raiškios sąveikos galimybės, serverio pusės biznio logika, kliento pusės biznio logika, dalinis puslapio atnaujinimas, retos serverio iteracijos (round-trips), serveris-klientas komunikavimas, funkcionavimas atsijungus, navigavimas klaviatūra.

CRM sistemų analizė parodė, kad esamos CRM sistemose dinaminės savybės nėra realizuotos, o jos galėtų žymiai pagerinti CRM sistemų veikimą.

Šiame darbe siekiama patobulinti klientų-ryšių valdymo sistemą, pakeičiant esamą sistemos interneto sąsają dinamine. Ši sąsaja turi supaprastinti informacijos mainus tarp įmonės ir jos klientų, suteikti patogesnę ir efektyvesnę komunikaciją, paspartinti projektų vykdymo procesą, pateikti vartotojams interaktyvias priemones dinaminiam turinio valdymui bei suteikti papildomo funkcionalumo jau egzistuojančiai CRM sistemai.

Potencialūs sistemos vartotojai yra tie, kuriems neužtenka internete atlikti tik pirkėjo ir vadybininko rolę. Tai vartotojai, kurie nori ne tik atlikti privalomas funkcijas, bet ir sąveikauti su kitais, patys aktyviai dalyvauti vykdomame marketingo procese. Pirkėjų ir įmonės darbuotojų poreikiai nuolat auga, jie tampa vis išrankesniais, todėl sistemos prototipas bus paremtas raiškaus interneto taikomosiomis programomis, kurios sistemai sukurs vartotojui patrauklų interaktyvų interfeisą, kuris suteikia Web puslapiams platesnes grafines galimybes. O pagerinta taikomųjų programų sąsaja leis kurti ir naudoti papildomus modulius. Kuriamos sistemos vartotojas – tai išrankus, besitikintis iš CRM

sistemos daugiau dinamikos, operatyvumo ir interaktyvumo vartotojas. Kompiuterizuojamoje sistemoje bus du vartotojų tipai - įmonės klientai ir administratoriai (įmonės projektų vadovai), kurių funkcijos bus apibrėžiamos tolesniame analizės etape.

Darbe **siekama** pritaikyti RIA taikomųjų programų savybes CRM sistemai, suteikiant vartotojų interfeisui didesnę dinamiškumą ir naudojimo patogumą. Tam tikslui bus tiriamos galimybės pritaikyti analizės metu nustatytus RIA taikomųjų programų privalumus. CRM sistemoje bus įdiegtos raiškaus interneto taikomosios programos, kurių pagrindinis uždavinys – parodyti Web 2.0 taikomųjų programų pritaikymo galimybes įmonės vartotojo interfeiso tobulinime.

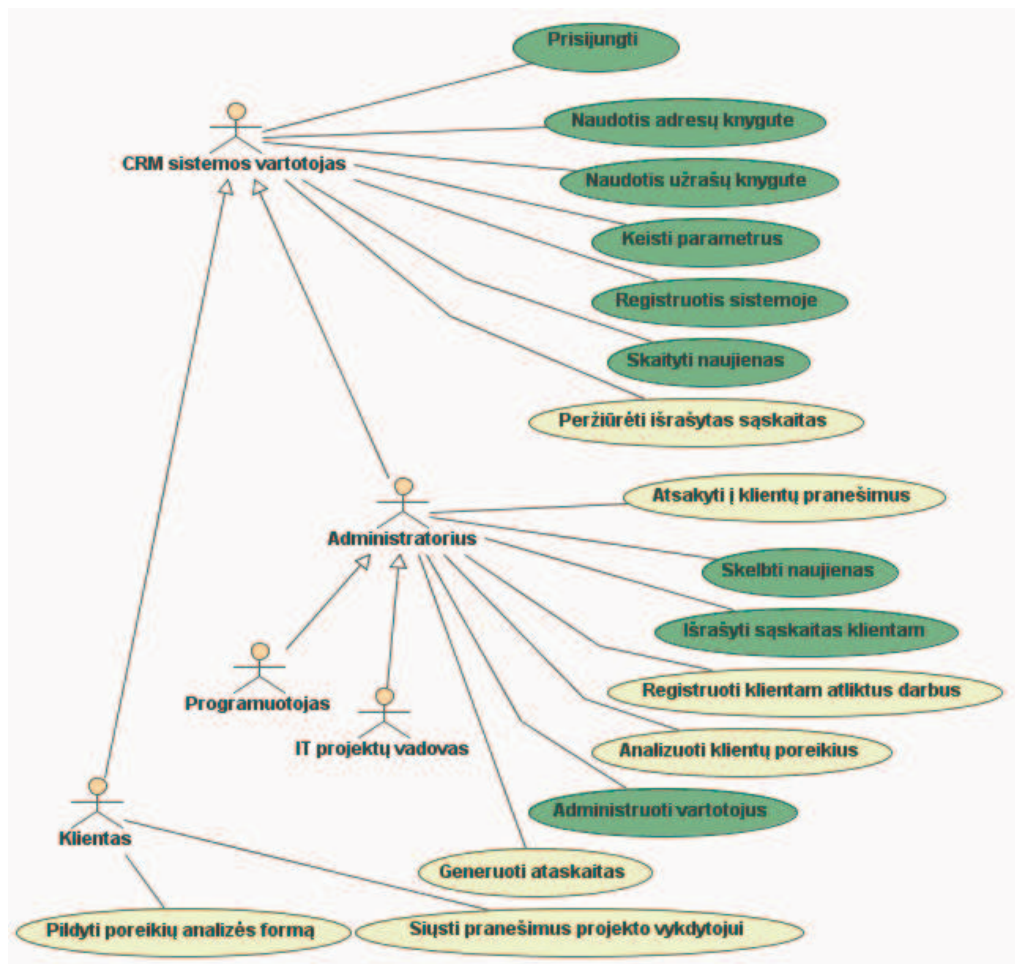
Taigi **darbo tikslas** – padidinti CRM sistemų dinamiškumą, išplečiant jas raiškių interneto programų savybėmis ir sudarant rekomendacijas esamoms sistemoms patobulinti.

2.8. Kompiuterizuojamos sistemos funkcijos

Kompiuterizuojamoje sistemoje bus du vartotojų tipai, kurie galės atlikti tokias funkcijas:

- a) Įmonės klientas – registruoti sistemos klientai galės siųsti projekto vykdymui bei palaikymui reikalingą informaciją bei pastabas atsakingiems už projektą įmonės darbuotojams (šiuo atveju programuotojui, projekto vadovui ar web dizaineriui). Taip pat galės stebėti projekto vykdymą, operatyviai gaus specialius pasiūlymus bei naujienas. Prisijungus prie sistemos bus galima parsisiųsti išrašytas sąskaitas už atliktą darbą. Galės naudotis sistemoje įdiegta užrašų bei adresų knygelės funkcija.
- b) Administratorius – šiuo atveju administratoriaus teises turės projektų vadovas ir programuotojas. Gavęs administratoriaus teises, darbuotojas galės redaguoti savo prisijungimo informaciją, komunikuoti su užsakovu (sukurti, naikinti pranešimus klientui), peržiūrėti vartotojų prisijungimų statistiką, patvirtinti naujus vartotojus, šalinti esamus, priskirti vartotojams teises, sudaryti darbų atlikimo ataskaitą, bei išrašyti sąskaitą klientui. Papildomai galės naudotis sistemoje įdiegta užrašų bei adresų knygelės funkcija.

Žemiau pateiktas (2.9 pav) kompiuterizuojamos sistemos panaudojimo atvejų modelis. Žalia spalva pažymėti panaudojimo atvejai rodo tobulinamas sistemos funkcijas.



2.9 pav. Kompiuterizuojamos sistemos panaudojimo atvejų modelis

Esamos CRM sistemos trūkumai ir sprendimai, pašalinantys nustatytus trūkumus, pateikti 2.6 lentelėje.

2.6 lentelė Esamos sistemos trūkumai ir jų šalinimas

Trūkumas	Sprendimas
Nepatogus formos duomenų validavimas	Vartotojo prisijungimo prie sistemos duomenų tikrinimas ir formų duomenų validavimas realiu laiku
Neefektyvus įrašų paieškos vykdymo procesas	Automatinis galimų rezultatų variantų pasiūlymas paieškos metu
Nėra galimybės nustatyti kontakto buvimo vietas	Momentinis kontakto buvimo vietos nustatymas ir atvaizdavimas integruotoje "Google Maps" sistemoje pagal įvestą vietovę
Nevisuomet pastebimos paskutinės naujienos	Naujienų sąrašo atnaujinimas realiu laiku.
Nėra RSS naujienų skaitytuvo	Integruotas skaitytuvas RSS naujienų skaitymui.
Naviguojant per meniu reikia laukti kol persikrauna visa puslapio informacija	Dalinis informacijos atnaujinimas, neperkraunant viso puslapio navigavimo metu.
Nepatogu komunikuoti su klientais ir sekti sąskaitų apmokėjimą	Momentinis pranešimų pateikimas vartotojui, sistemoje įvykus naujiems svarbiems įvykiams ar pasikeitimams.

nepataisomos žalos sistemai. Pvz., registruotas klientas negali peržiūrėti, keisti ar pasisavinti kitų klientų informacijos, valdyti kai kurias kitas tik administratoriams galiojančias sritis, kur jam nėra suteiktos valdymo teisės.

- Sistema privalo užtikrinti vartotojų prisijungimo slaptažodžių slaptumą ir saugumą. Tiek registruotas klientas, tiek administratorius negali matyti nei savo, nei kitų vartotojų slaptažodžių, administratorius gali tik pakeisti savo, kito administratoriaus bei registruoto kliento slaptažodį, o registruotas klientas – tik savo slaptažodį. Slaptažodžiai privalo būti saugomi duomenų bazėje, užkoduoti pasirinktu kodavimo algoritmu, siekiant užtikrinti maksimalią apsaugą.
- Sistema privalo būti suderinama su populiariausiomis interneto naršyklėmis, tokiomis kaip Internet Explorer, Mozilla, Firefox, Opera, Netscape. Tą pasiekti galima nuodugniai testuojant sukurtą sistemą ir taikant atitinkamas priemones nesuderinamumų pašalinimui.
- Sistema turi būti pasiekama iš bet kurios vietos, kur yra interneto ryšys. Sistemoje turėtų būti naudojamas “plono kliento” modelis, tam, kad klientui nereikėtų papildomos programinės įrangos bei didelių kompiuterinės įrangos resursų – pagrindinė sistemos logika atliekama serverio pusėje.
- Administratorius turi turėti galimybę matyti vartotojų prisijungimų prie sistemos statistiką, vartotojų valdomą informaciją ir esant reikalui ją keisti, taip pat valdyti ją pats.
- Sistema turi būti sukurta naudojant šias programines priemones: PHP, MYSQL, JavaScript, XHTML, CSS.
- Sistemos vykdymui ir palaikymui reikalingas Apache web-serveris.
- Duomenų saugojimui turi būti sukurtas atskiras MYSQL vartotojas, kuriam turi būti sukurtas aukšto sudėtingumo lygio slaptažodis bei atskira MYSQL duomenų bazė.
- Sistemoje turi būti apsauga nuo atsitiktinio klaidingo duomenų įvedimo.
- Sistemos vartotojo interfeisui turi būti naudojamas stilių šablonas CSS, kuriame skoningai parinktas 3-4 spalvų rinkinys.
- Informacine sistema gali naudotis keletas vienu metu prisijungusių vartotojų.
- Iš kliento pusės pasinaudoti kuriama IS reikalinga interneto naršyklė palaikanti nurodytas specifikacijas.

2.11. Rizikos faktorių analizė

Šiame poskyryje įvertinamos galimos sistemos kūrimo rizikos ir numatytomas galimų problemų sprendimo planas.

Galimos sistemos kūrimo rizikos

Galimos sistemos kūrimo rizikos įvertintos 2.7 lentelėje.

2.7 lentelė Galimos sistemos kūrimo rizikos

Rizikos faktorius	Tikimybinis įvertinimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	8
Realizavimo pakitimai dėl patirties trūkumo	5
Vykdytojo liga	2
Pašaliniai faktoriai	3
Užsibrėžtų tikslų nepasiekimas	4
Technologijų amžius (naujumas)	5

Atsitiktinumų (rizikų) planas

Rizikos faktoriai ir numatomi planai problemoms spręsti pavaizduoti 2.8 lentelėje.

2.8 lentelė Problemos sprendimo planai

Rizikos faktorius	Problemos sprendimas
Reikalavimų specifikacijos pasikeitimai realizavimo fazėje	Kad keisti reikalavimus tokioje fazėje jie turi būti tikrai labai svarbūs
Realizavimo pakitimai dėl patirties trūkumo	Mokymasis skaitant techninę literatūrą ir analizuojant panašių projektų kūrimą
Vykdytojo liga	Rizika pašalinama paliekant daugiau laiko numatytiems darbams atlikti. Tai įvertinta projekto plane, vykdytojo tvarkaraštyje.
Pašaliniai faktoriai	Darbų apimties mažinimas, trukmės ilginimas.
Užsibrėžtų tikslų nepasiekimas	Aiškų tikslų apibrėžimas, paaiškinimas. Jei tikslai per daug ambicingi, pasverti juos iš naujo.
Technologijų amžius (naujumas)	Rizika gali būti sumažinama naudojant esamas (current) technologijas vietoj naujų, dar nepripažintų ar nepatvirtintų. Ankstyvo prototipo sukūrimas taip pat gali sumažinti šią riziką.

2.12. Rezultato kokybės kriterijai

Sukurta sistema turi atitikti šiuos kokybės kriterijus:

- Sistema turi turėti griežtą duomenų kontrolę.
- Sistemos formos validuojamos dinamiškai, t.y. pranešimas apie klaidingai vedamus duomenis pasirodo iškart, o ne po formos duomenų išsiuntimo.
- Interaktyvi puslapio navigacija, t.y. pereinant iš vieno meniu punkto į kitą puslapis neperkraunamas iš naujo.

- Informacijos mainai tarp sistemos vartotojų vyksta realiu laiku.
- Patogesnė ir efektyvesnė komunikacija.
- Spartus projektų vykdymo procesas.
- Sistema turi interaktyvias priemones dinaminiam turinio valdymui
- Sistema apsaugota nuo nesankcionuoto vartojimo.
- Pasiektas optimalus populiariausių interneto naršyklių suderinamumas.
- Svetainės interfeisas atitinka W3C konsorciumo standartus.

2.13. Analizės išvados

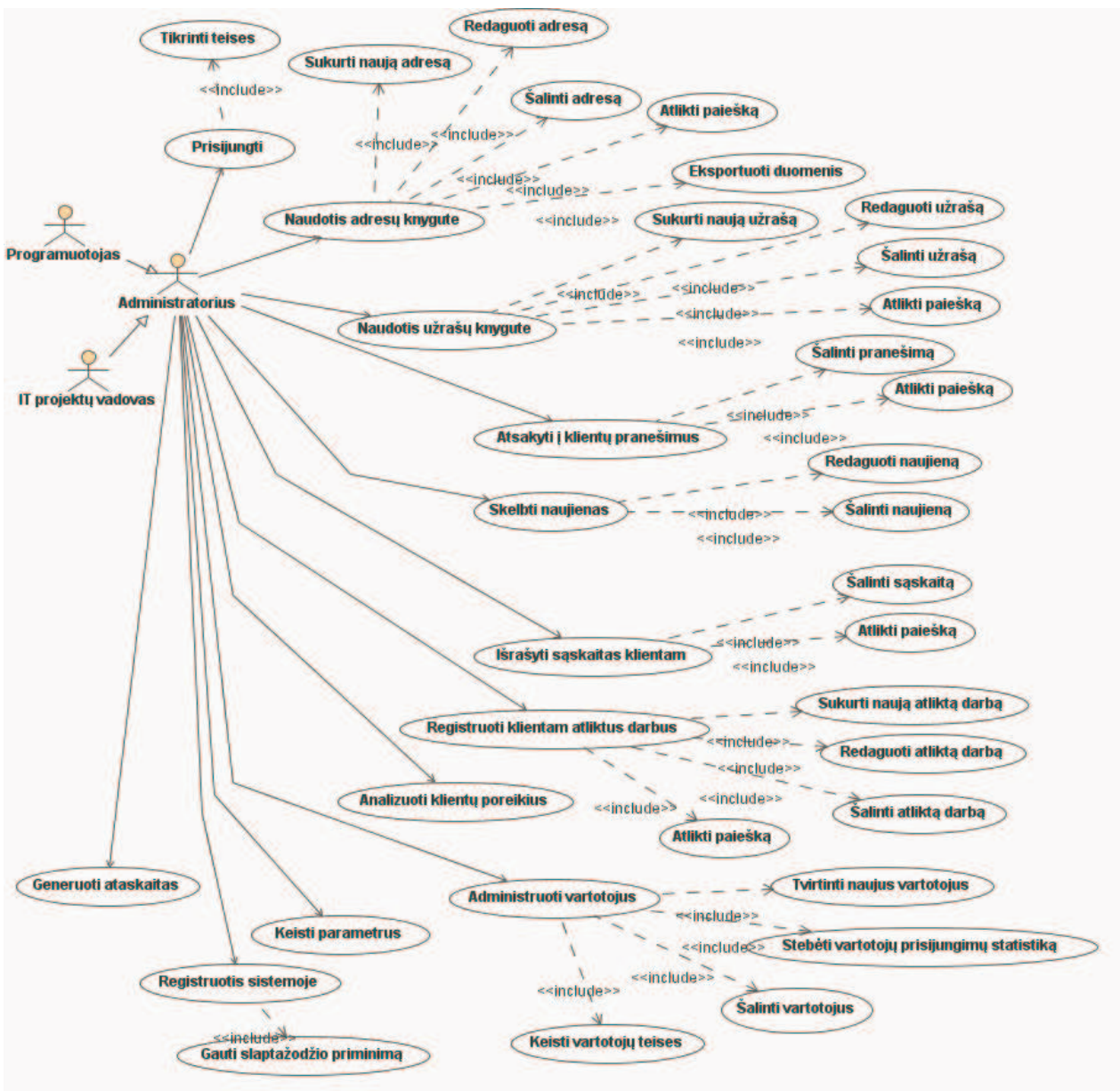
1. RIA taikomųjų programų analizės metu nustatyta, **kad** RIA technologijas tikslinga pritaikyti įmonių informacinėms sistemoms ir ypač CRM, kadangi RIA leidžia pagerinti dinamines vartotojo sąveikos su interneto sistema savybes.
2. Panašių sistemų ir jų realizavimo metodų analizė parodė jų trūkumus, kurių galima išvengti, naudojant RIA technologijas, suteikiančias vartotojui daugiau dinamiškumo ir galimybes, panašias į darbastalio programų.
3. CRM sistemų analizė parodė, kokias funkcijas tikslinga realizuoti naudojant RIA metodus. Tai vartotojo prisijungimo prie sistemos duomenų korektiškumo ir teisingumo tikrinimą, ir formų duomenų validaciją realiu laiku; automatinį galimų rezultatų varintų pasiūlymą paieškos metu; momentinį kontakto buvimo vietos nustatymą ir atvaizdavimą integruotoje “Google Maps” sistemoje pagal įvestą vietovę; naujienų sąrašo atnaujinimą realiu laiku; RSS naujienų skaitytuvą; dalinį puslapių atnaujinimą naudojant administravimo meniu; momentinius pranešimus vartotojui apie gautas žinutes bei administratoriui – apie vartotojų neapmokėtas sąskaitas.
4. Remiantis išsamia problemos sprendimo metodų literatūros šaltiniuose analize, nuspręsta, jog dinaminio vartotojo interfeiso tobulinimui tikslinga pasirinkti JavaScript/DHTML AJAX įgyvendinimo metodą (Asinchroninis JavaScript + XML) dėl to, kad gerai pritampa prie egzistuojančių HTML Web taikomųjų programų bei yra tinkamas DHTML Web programuotojams.
5. Analizės rezultate suformuluoti funkciniai tikslai ir nefunkciniai reikalavimai, apibrėžtos pradinės kompiuterizuojamos sistemos funkcijos, sudarytas duomenų konceptų modelis, įvertinti rizikos faktoriais bei nustatyti rezultato kokybės kriterijai.

3. CRM sistemos reikalavimų specifikacija ir analizė

Šiame skyriuje apibrėžti sistemos vartotojų panaudojimo atvejai, kiekvienam panaudojimo atvejui aprašytos specifikacijos ir sudarytos veiklos diagramos. Panaudojimo atvejams, kuriems buvo atlikti patobulinimai, aprašytos papildomos specifikacijų lentelės

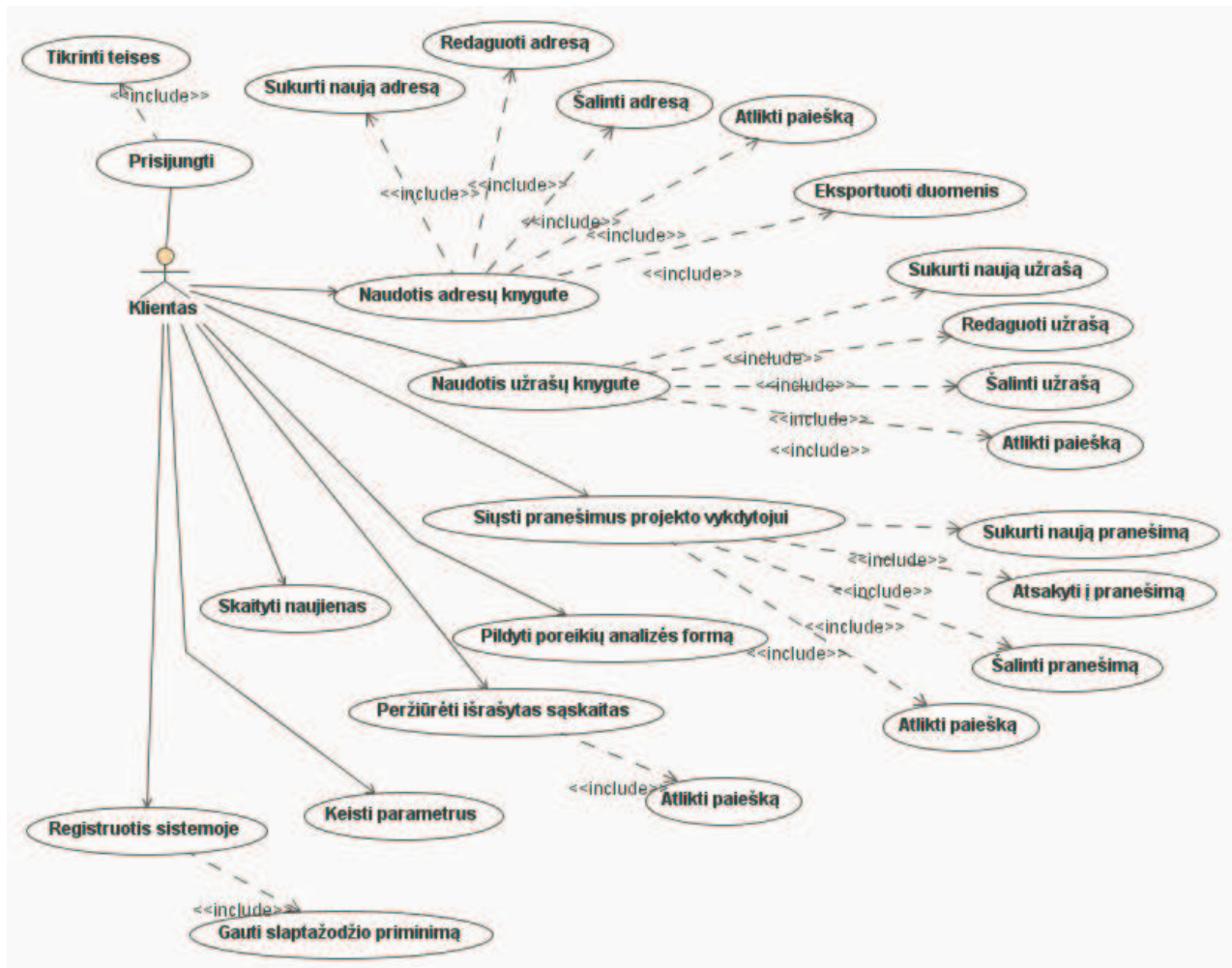
3.1. Sistemos vartotojų panaudojimo atvejai

Žemiau pavaizduotas (3.1 pav.) administratoriaus detalus panaudojimo atvejų modelis. Administratorius apima programuotojas ir projekto vadovas, t.y. abu vartotojai turi pas pačias teises. Administratorius – tai registruoto vartotojo apibendrinimas su papildomomis galimybėmis, nes jis gali atlikti lygiai tuos pačius veiksmus, kaip ir registruotas vartotojas, bet jo teisės yra praplėstos, jis gali atlikti papildomų veiksmų. Taigi, administratorius – tai vartotojas, turintis aukštesnio lygio teises.



3.1 pav. Administratoriaus panaudojimo atvejų modelis

Žemiau pavaizduotas (3.2 pav.) kliento detalus panaudojimo atvejų modelis. Visi įmonės klientai turi apribotas teises, lyginant su administratoriaus teisėmis.

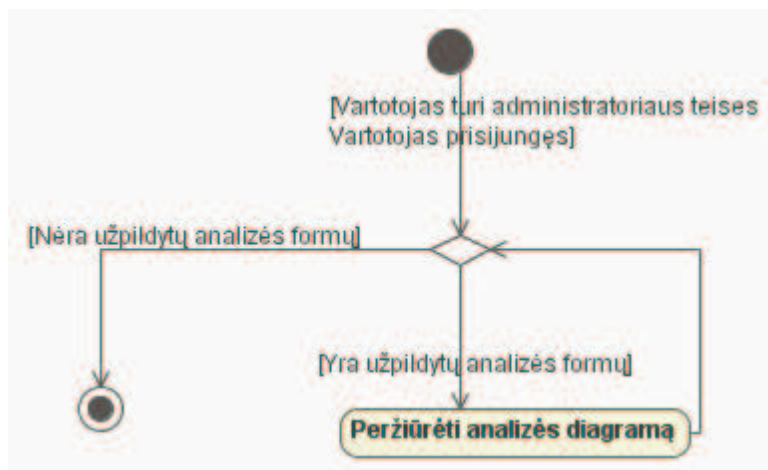


3.2 pav. Kliento panaudojimo atvejų modelis

3.2. Sistemos vartotojų panaudojimo atvejų specifikacijos

Aprašant reikalavimų specifikaciją kiekvienam panaudojimo atvejui nubraižytos veiklos diagramos bei detaliai aprašytos specifikacijų lentelės. Panaudojimo atvejams, kuriems buvo atlikti patobulinimai, aprašytos papildomos specifikacijų lentelės, kurios nusako siekiamą dinamiškumą

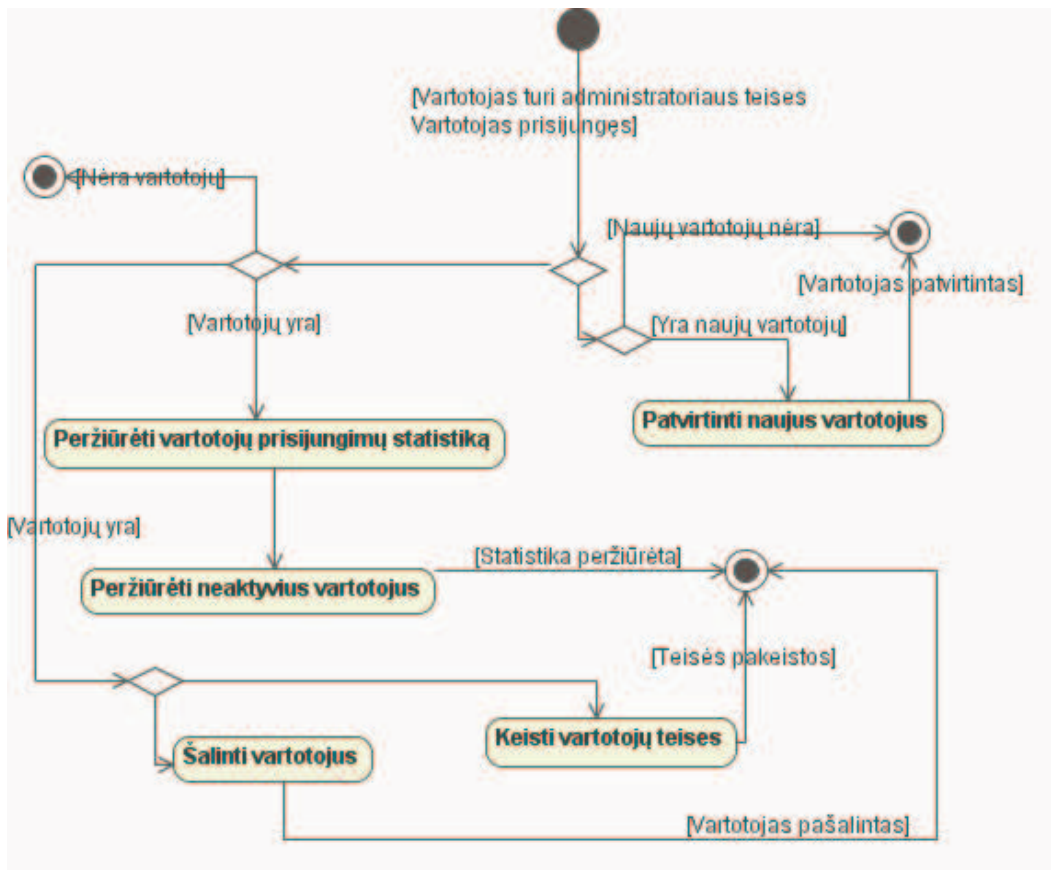
Administratoriaus panaudojimo atvejų specifikacijos pavaizduotos 3.3 - 3.8 paveiksliukuose.



3.3 pav. Administratoriaus panaudojimo atvejo „Analizuoti klientų poreikius“ veiklos diagrama

3.1 lentelė PA „Analizuoti klientų poreikius“ specifikacija

Panaudojimo atvejis	Analizuoti klientų poreikius
Numeris	PA1
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Administratorius nori peržiūrėti klientų poreikius
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises Bent vienas klientas yra užpildęs analizės formą
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka analizės peržiūros meniu punktą.	1.1. Vartotojui parodomas klientų poreikių diagrama grafiniu pavidalu.
Po sąlyga	Peržiūrėta analizės diagrama.
Alternatyvos (nesėkmės atvejai)	1.1.a. Jei nei vienas klientas nėra užpildęs poreikių analizės formos, tai diagrama nėra atvaizduojama.
Vykdyto variantai	1. Vartotojas pasirenka meniu punktą „Poreikių analizės diagrama“.
Veiklos taisyklės	-
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-



3.4 pav. Administratoriaus panaudojimo atvejo „Administruoti vartotojus“ veiklos diagrama

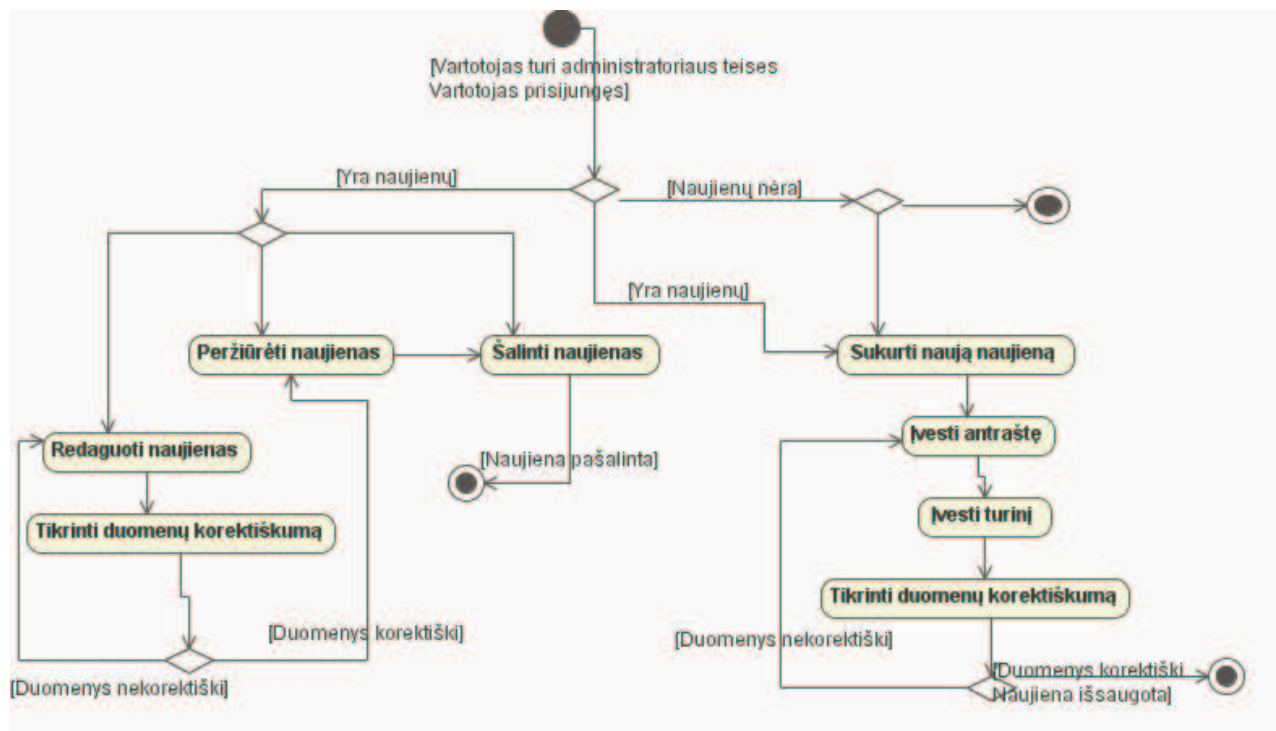
3.2 lentelė PA „Administruoti vartotojus“ specifikacija

Panaudojimo atvejis	Administruoti vartotojus
Numeris	PA2
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadavimo sąlyga:	Administratorius nori valdyti klientų duomenis
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka prisijungimų prie sistemos statistikos, vartotojų patvirtinimo, šalinimo, teisių keitimo punktus. 2. Vartotojas pasirenka klientą, kurį nori patvirtinti. 3. Vartotojas pasirenka šalinamą klientą. 4. Vartotojas keičia sistemos vartotojo teises.	1.1.Vartotojui atidaromi atitinkami langai, perkraunant puslapius. 2.1.Sistema patvirtina pasirinktą klientą. 3.1.Sistema pašalina pasirinktą klientą. 4.1.Sistema pakeičia vartotojo teises.
Po sąlyga	Atliktas vartotojų administravimas
Alternatyvos (nesėkmės atvejai)	3.1.a. Jei vartotojas turi administratoriaus teises, tai jį pašalinti galima tik suteikus paprasto vart. teises.

Vykdymo variantai	1. Vartotojas pasirenka meniu punktą „Prisijungimų statistika“. 2. Vartotojas pasirenka norimą patvirtinti vartotoją. 3. Vartotojas pasirenka norimą šalinti vartotoją. 4. Vartotojas pasirenka vartotoją, kuriam keičia teises.
Veiklos taisyklės	-
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-

3.3 lentelė PA „Administruoti vartotojus“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka prisijungimų prie sistemos statistikos, vartotojų patvirtinimo, šalinimo, teisių keitimo punktus. 2. Vartotojas pasirenka klientą, kurį nori patvirtinti. 3. Vartotojas pasirenka šalinamą klientą. 4. Vartotojas keičia sistemos vartotojo teises.	1.1. Vartotojui akimirksniu parodomi pareikalauti duomenys neperkraunant puslapio. 2.1. Sistema patvirtina pasirinktą klientą (atliekamas tik dalinis puslapio atnaujinimas). 3.1. Sistema pašalina pasirinktą klientą (atliekamas tik dalinis puslapio atnaujinimas). 4.1. Sistema pakeičia vartotojo teises (atliekamas tik dalinis puslapio atnaujinimas).



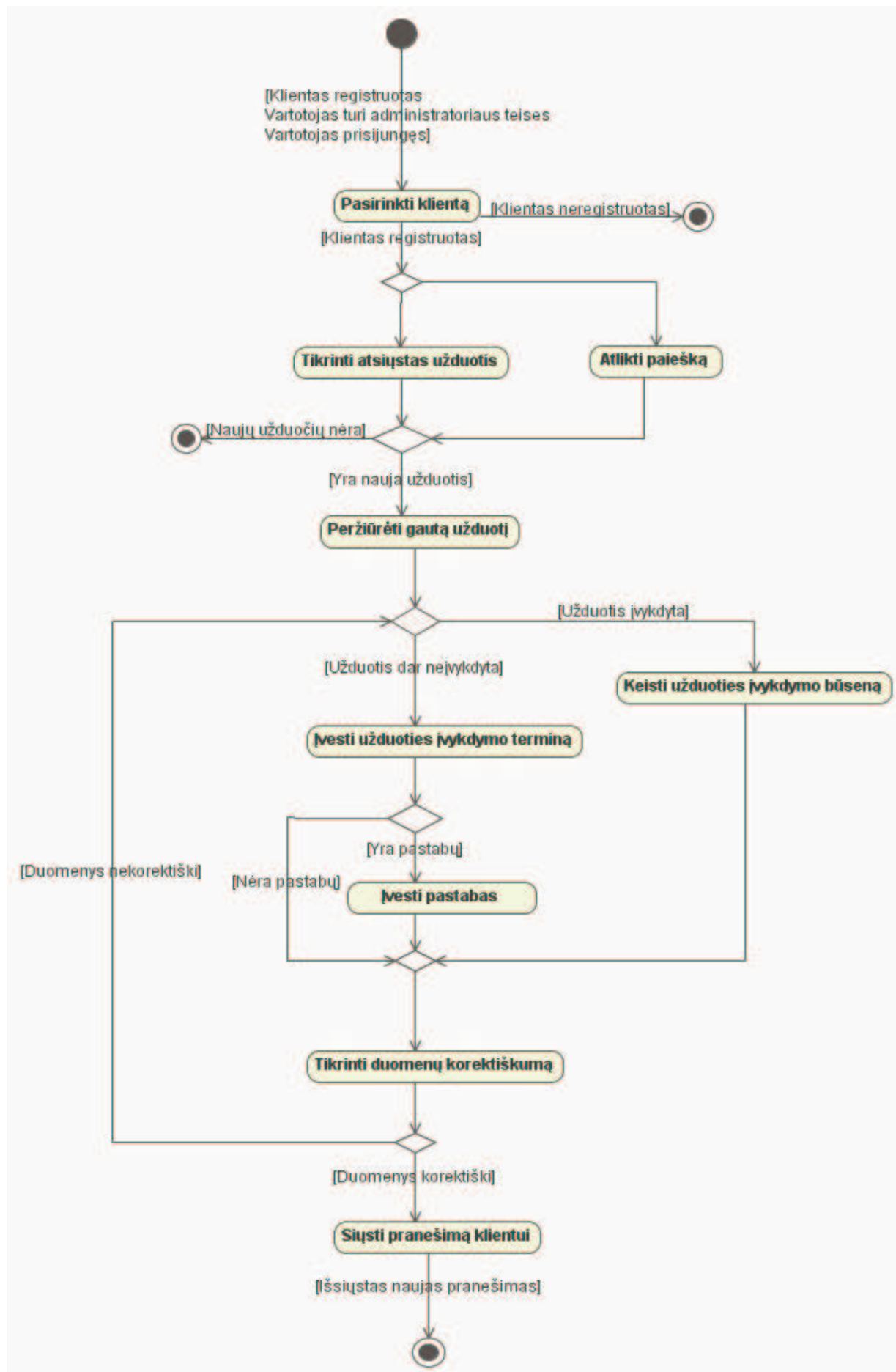
3.5 pav. Administratoriaus panaudojimo atvejo „Skelbti naujienas“ veiklos diagrama

3.4 lentelė PA „Skelbti naujienas“ specifikacija

Panaudojimo atvejis	Skelbti naujienas
Numeris	PA3
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Administratorius nori paskelbti naujieną
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka naujienų peržiūros punktą. 2. Vartotojas pasirenka naujo įrašo punktą. 3. Vartotojas įveda naują naujieną. 4. Vartotojas redaguoja naujieną. 5. Vartotojas šalina naujieną.	1.1.Sistema parodo paskelbtas naujienas. 2.1.Sistema išveda formą naujos naujienos skelbimui. 3.1.Sistema išsaugo informaciją. 4.1.Sistema išsaugo pakeitimus. 5.1.Sistema pašalina įrašą.
Po sąlyga	Atlikti naujienų skelbimo darbai
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyto variantai	1. Vartotojas pasirenka meniu punktą „Naujienos“. 2. Vartotojas pasirenka naujo įrašo punktą 3. Vartotojas įveda naują naujieną 4. Vartotojas redaguoja naujieną 5. Vartotojas šalina naujieną
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Redaguoti naujieną“ ir „Šalinti naujieną“
Pastabos	-
Neišspręstos problemos	-

3.5 lentelė PA „Skelbti naujienas“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka naujienų peržiūros punktą. 2. Vartotojas pasirenka naujo įrašo punktą. 3. Vartotojas įveda naują naujieną. 4. Vartotojas redaguoja naujieną. 5. Vartotojas šalina naujieną.	1.1.Sistema parodo paskelbtas naujienas. Naujienų sąrašas keičiasi realiu laiku. Nereikia papildomai perkrauti puslapio tam, kad patikrinti, ar buvo patalpinta nauja naujiena. 2.1.Sistema išveda formą naujos naujienos skelbimui. 3.1.Sistema išsaugo informaciją. Naujienų sąrašas visiems vartotojams atsinaujina realiu laiku. 4.1.Sistema išsaugo pakeitimus. Naujienų sąrašas visiems vartotojams atsinaujina realiu laiku. 5.1.Sistema pašalina įrašą. Naujienų sąrašas visiems vartotojams atsinaujina realiu laiku.



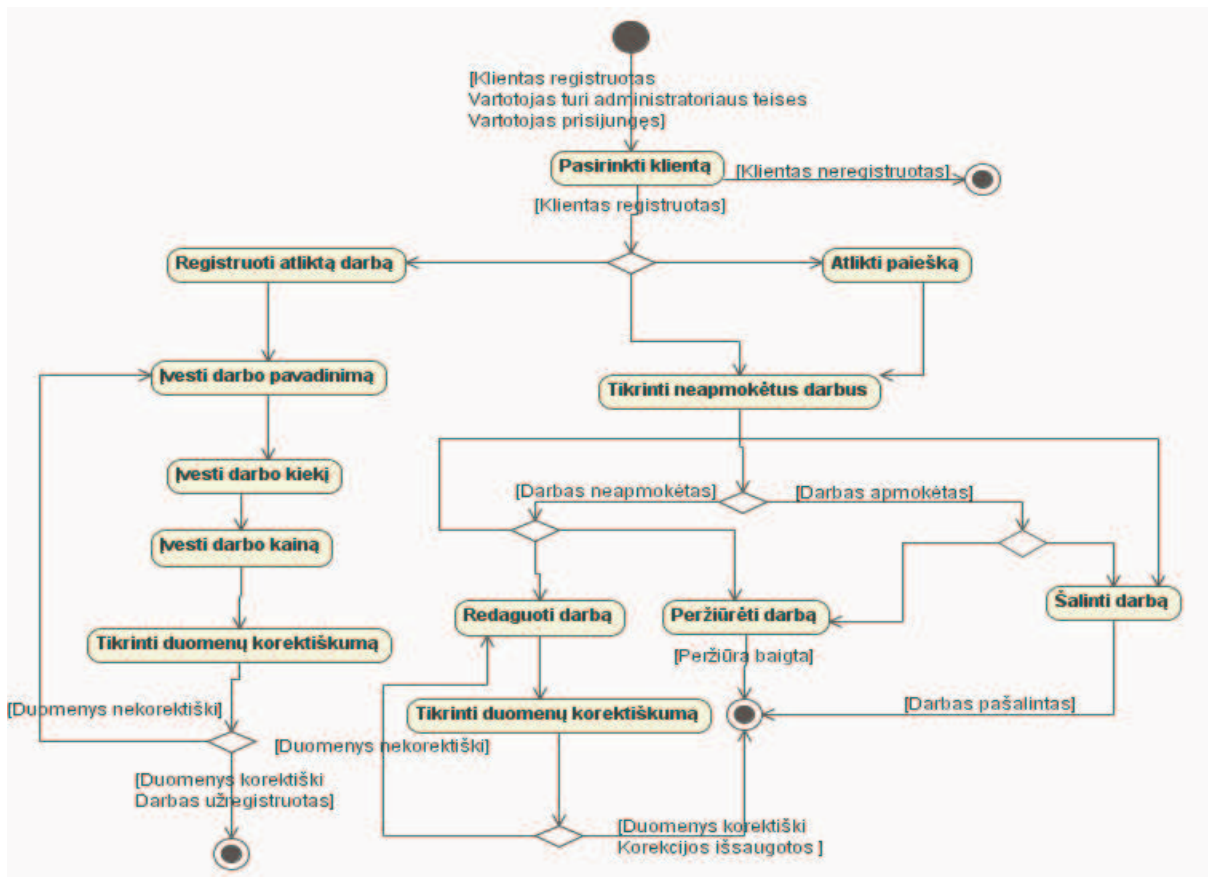
3.6 pav. Administratoriaus panaudojimo atvejo „Atsakyti į klientų pranešimus“ veiklos diagrama

3.6 lentelė PA „Atsakyti į klientų pranešimus“ specifikacija

Panaudojimo atvejis	Atsakyti į klientų pranešimus
Numeris	PA4
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Administratorius nori atsakyti į pranešimą
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises Klientas registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kliento atsiųstų pranešimų punktą. 2. Vartotojas pasirenka pranešimo paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas atsako į pranešimą. 5. Vartotojas šalina pranešimą.	1.1.Sistema parodo atsiųstus pranešimus. 2.1.Sistema išveda paieškos formą. 3.1.Sistema išveda rastus įrašus. 4.1.Sistema išsaugo pakeitimus. 5.1.Sistema išsaugo šalinimo žymę.
Po sąlyga	Atsakyta į pranešimą
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyimo variantai	1. Vartotojas pasirenka klientą 2. Vartotojas pasirenka paieškos punktą 3. Vartotojas įveda ieškomą žodį 4. Vartotojas atsako klientui 5. Vartotojas pašalina pranešimą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Atlikti paiešką“ ir „Šalinti pranešimą“
Pastabos	-
Neišspręstos problemos	-

3.7 lentelė PA „Atsakyti į klientų pranešimus“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kliento atsiųstų pranešimų punktą. 2. Vartotojas pasirenka pranešimo paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas atsako į pranešimą. 5. Vartotojas šalina pranešimą.	1.1.Sistema parodo atsiųstus pranešimus. Jei vartotojas gauna naują pranešimą, akimirksniu pasirodo informacinis langas, kuriame matomas siuntėjas ir pranešimo antraštė. Nereikia papildomai perkrauti puslapio tam, kad patikrinti, ar buvo atsiųstas naujas pranešimas. 2.1.Sistema išveda paieškos formą. 3.1. Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus. 4.1.Sistema išsaugo pakeitimus. Gavėjo darbiname lange akimirksniu pasirodo informacinis langas, kuriame matomas siuntėjas ir pranešimo antraštė. 5.1.Sistema išsaugo šalinimo žymę.

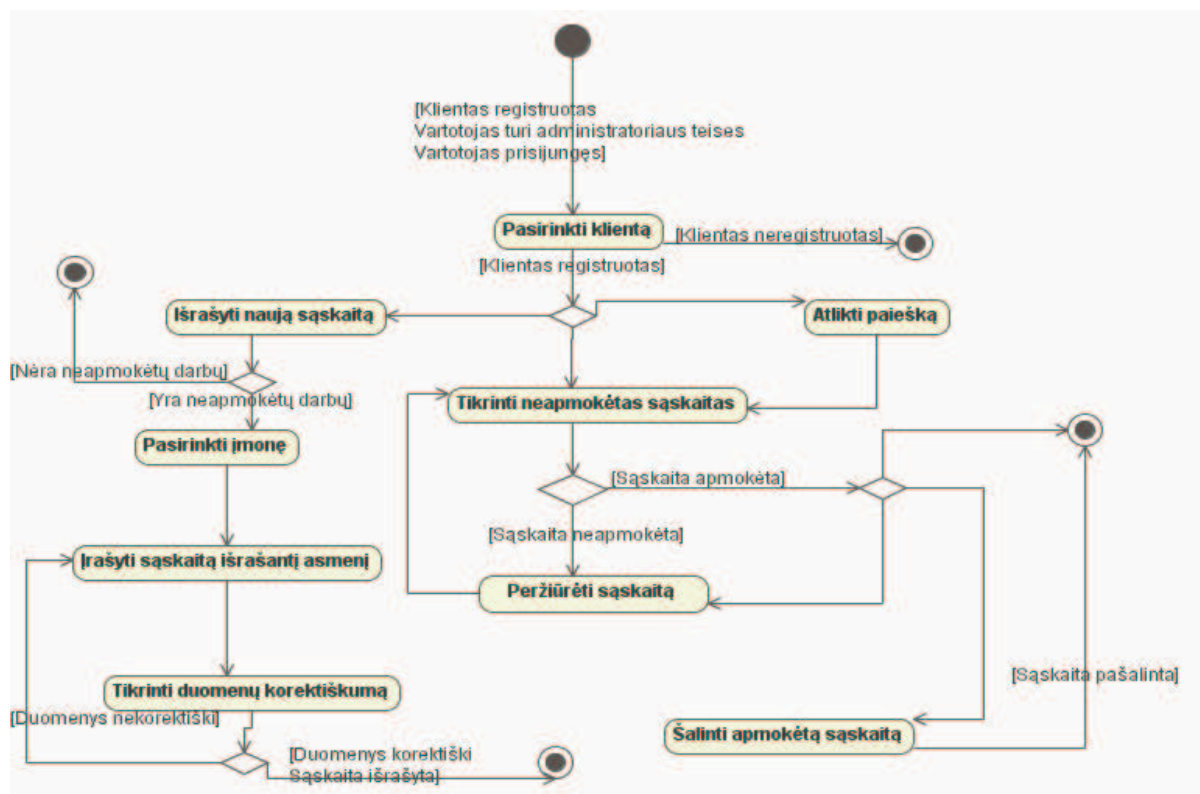


3.7 pav. Administratoriaus panaudojimo atvejo „Registruoti klientam atliktus darbus“ veiklos diagrama

3.8 lentelė PA „Registruoti klientam atliktus darbus“ specifikacija

Panaudojimo atvejis	Registruoti klientam atliktus darbus
Numeris	PA5
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadavimo sąlyga:	Administratorius nori užregistruoti atliktą darbą.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises Klientas registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka klientui atliktų darbų punktą. 2. Vartotojas pasirenka darbo paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas pasirenka naujo darbo registravimo punktą. 5. Vartotojas įveda naują darbą. 6. Vartotojas redaguoja darbą. 7. Vartotojas šalina darbą.	1.1. Sistema parodo atliktus darbus. 2.1. Sistema išveda paieškos formą. 3.1. Sistema išveda rastus įrašus. 4.1. Sistema išveda naujo darbo registravimo formą 5.1. Sistema išsaugo įrašą. 6.1. Sistema išsaugo pakeitimus. 7.1. Sistema pašalina įrašą.
Po sąlyga	Atlikti darbai užregistruoti
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.

Vykymo variantai	<ol style="list-style-type: none"> 1. Vartotojas pasirenka klientą 2. Vartotojas pasirenka paieškos punktą 3. Vartotojas įveda ieškomą žodį 4. Vartotojas pasirenka naujo įrašo punktą 5. Vartotojas įveda naują darbą 6. Vartotojas redaguoja darbą. 7. Vartotojas šalina darbą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Redaguoti atliktą darbą“, „Šalinti atliktą darbą“, „Atlikti paiešką“ ir „Sukurti naują atliktą darbą“
Pastabos	-
Neišspręstos problemos	-



3.8 pav. Administratoriaus panaudojimo atvejo „Įrašyti sąskaitas klientam“ veiklos diagrama

3.9 lentelė PA „Įrašyti sąskaitas klientam“ specifikacija

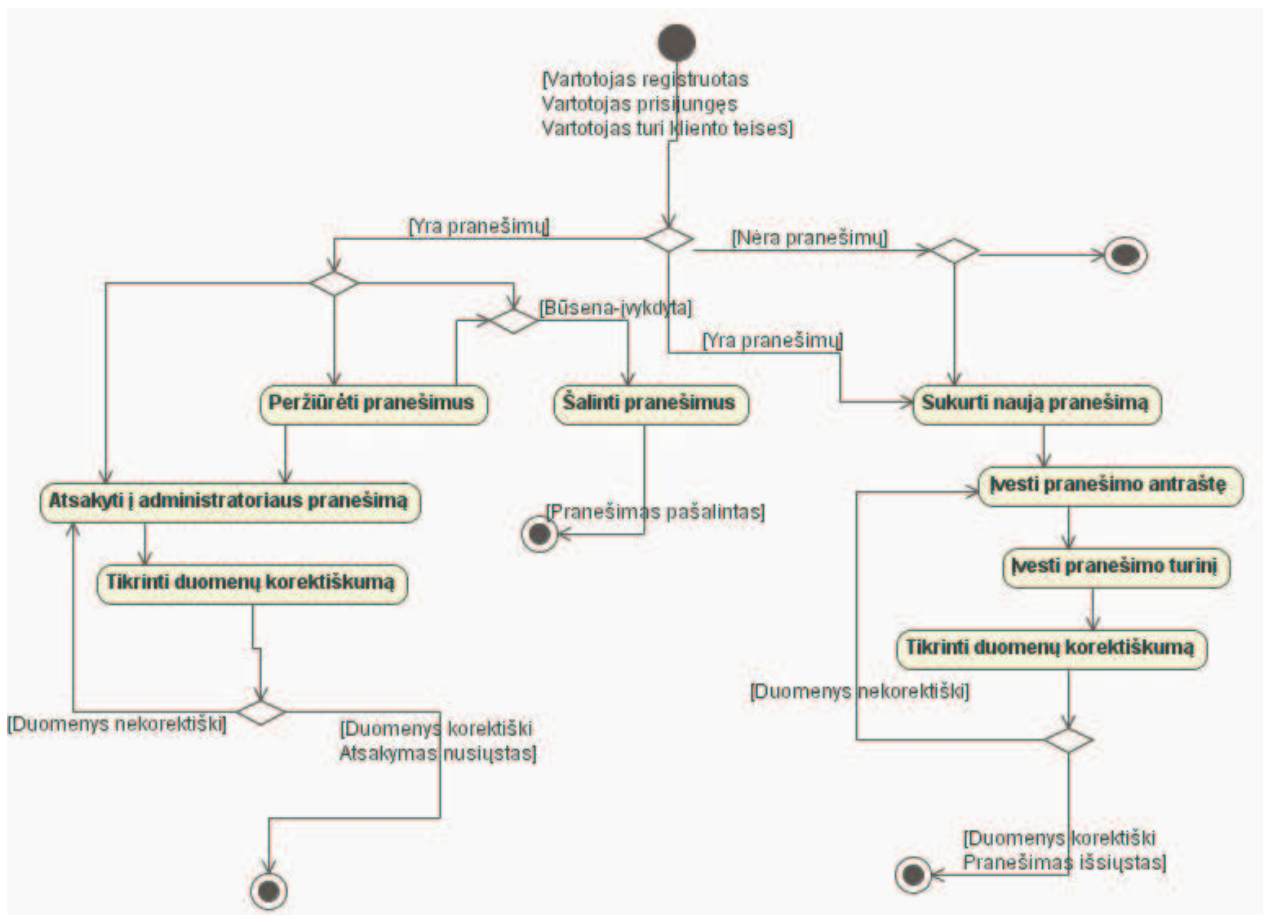
Panaudojimo atvejis	Įrašyti sąskaitas klientam
Numeris	PA6
Aktorius	Administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Administratorius nori įrašyti sąskaitą.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi administratoriaus teises Klientas registruotas

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka klientui išrašytų sąskaitų punktą. 2. Vartotojas pasirenka sąskaitos paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas pasirenka naujos sąskaitos išrašymo punktą. 5. Vartotojas pasirenka klientą ir patvirtina išrašymą. 6. Vartotojas patvirtina sąskaitos apmokėjimą. 7. Vartotojas šalina sąskaitą.	1.1. Sistema parodo išrašytas sąskaitas. 2.1. Sistema išveda paieškos formą. 3.1. Sistema išveda rastus įrašus. 4.1. Sistema išveda naujos sąskaitos išrašymo formą. 5.1. Sistema išsaugo įrašą. 6.1. Sistema išsaugo žymę apie apmokėjimą. 7.1. Sistema pašalina įrašą.
Po sąlyga	Sąskaita išrašyta
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas. Sąskaitos negalima pašalinti kol nėra apmokėjimo žymės.
Vykdyimo variantai	1. Vartotojas pasirenka klientą 2. Vartotojas pasirenka paieškos punktą 3. Vartotojas įveda ieškomą žodį 4. Vartotojas pasirenka naujo įrašo punktą 5. Vartotojas pasirenka klientą ir patvirtina išrašymą 6. Vartotojas patvirtina apmokėjimą. 7. Vartotojas šalina sąskaitą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Šalinti sąskaitą“, „Atlikti paiešką“
Pastabos	-
Neišspręstos problemos	-

3.10 lentelė PA „Išrašyti sąskaitas klientam“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka klientui išrašytų sąskaitų punktą. 2. Vartotojas pasirenka sąskaitos paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas pasirenka naujos sąskaitos išrašymo punktą. 5. Vartotojas pasirenka klientą ir patvirtina išrašymą. 6. Vartotojas patvirtina sąskaitos apmokėjimą. 7. Vartotojas šalina sąskaitą.	1.1. Sistema parodo išrašytas sąskaitas. Jei yra neapmokėtų sąskaitų, pasirodo informacinis langas, kuriame matomi neapmokėję sąskaitų klientai. Patvirtinus sąskaitos apmokėjimą, sąrašas pasikeičia realiu laiku. 2.1. Sistema išveda paieškos formą. 3.1. Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus. 4.1. Sistema išveda naujos sąskaitos išrašymo formą. 5.1. Sistema išsaugo įrašą. 6.1. Sistema išsaugo žymę apie apmokėjimą. Informaciniame lange matomų klientų neapmokėjusių sąskaitų sąrašas pasikeičia realiu laiku. 7.1. Sistema pašalina įrašą.

Kliento panaudojimo atvejų specifikacijos pavaizduotos 3.9 - 3.12 paveikslukuose ir detaliau panagrinėtos lentelėse.



3.9 pav. Kliento panaudojimo atvejo „Siųsti pranešimus projekto vykdytojui“ veiklos diagrama

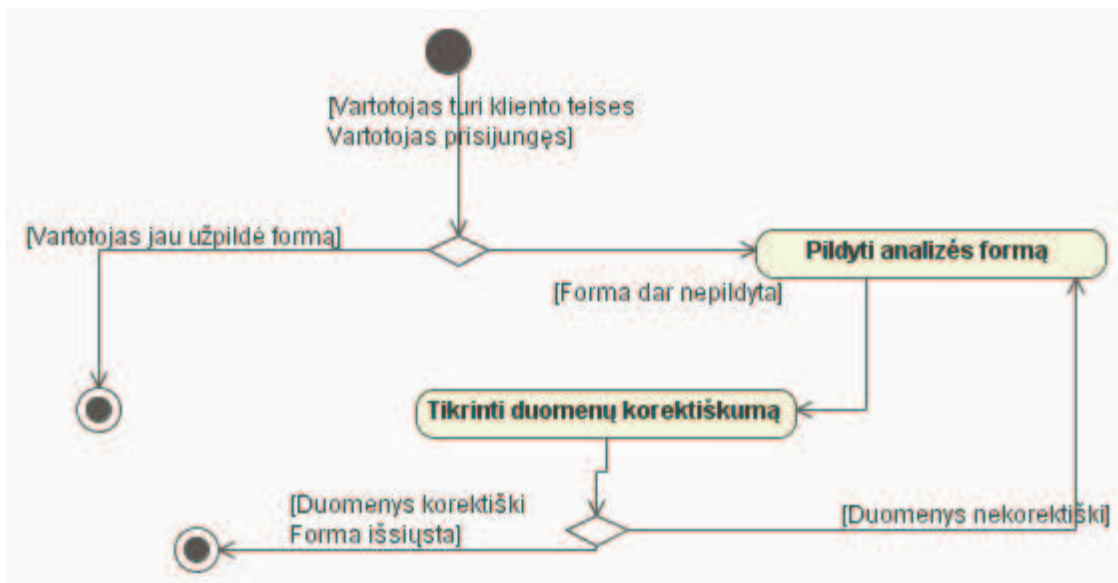
3.11 lentelė PA „Siųsti pranešimus projekto vykdytojui“ specifikacija

Panaudojimo atvejis	Siųsti pranešimus projekto vykdytojui
Numeris	PA7
Aktorius	Klientas
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadavimo sąlyga:	Klientas nori siųsti pranešimą.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi kliento teises Klientas registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka pranešimo paieškos punktą. 2.Vartotojas įveda ieškomą žodį. 3.Vartotojas pasirenka naujo pranešimo punktą. 4.Vartotojas suveda naujo pranešimo duomenis. 5.Vartotojas atsako į pranešimą.	1.1.Sistema išveda paieškos formą. 2.1.Sistema išveda rastus įrašus. 3.1.Sistema išveda naujo pranešimo siuntimo formą. 4.1.Sistema išsaugo įrašą. 5.1.Sistema išsaugo atsaką. 6.1.Sistema išsaugo pranešimo šalinimo žymę.

6. Vartotojas šalina pranešimą.	
Po sąlyga	Pranešimas išsiųstas
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyto variantai	1. Vartotojas pasirenka paieškos punktą 2. Vartotojas įveda ieškomą žodį 3. Vartotojas pasirenka naujo įrašo punktą 4. Vartotojas suveda naujo pranešimo duomenis 5. Vartotojas atsako į pranešimą 6. Vartotojas šalina pranešimą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Šalinti pranešimą“, „Atlikti paiešką“, „Atsakyti į pranešimą“, „Sukurti naują pranešimą“
Pastabos	-
Neišspręstos problemos	-

3.12 lentelė PA „ Siųsti pranešimus projekto vykdytojui“ patobulinimo specifikacija

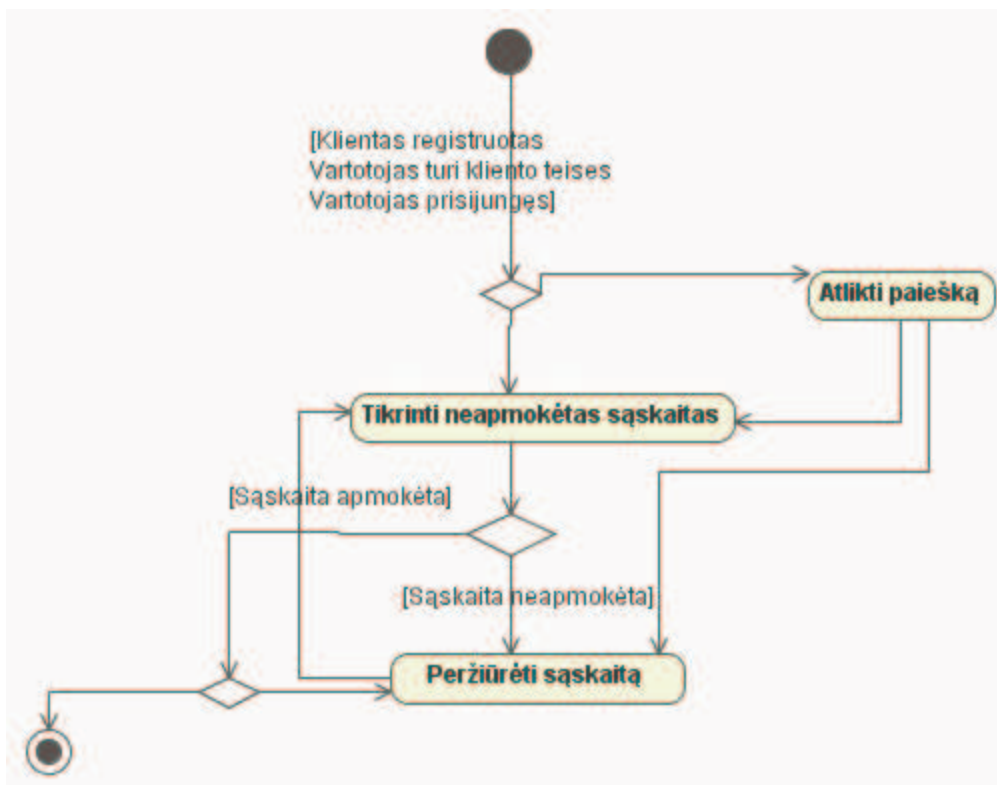
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka gautų pranešimų punktą. 2. Vartotojas pasirenka pranešimo paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas pasirenka naujo pranešimo punktą. 5. Vartotojas suveda naujo pranešimo duomenis. 6. Vartotojas atsako į pranešimą. 7. Vartotojas šalina pranešimą.	1.1. Sistema parodo atsiųstus pranešimus. Jei vartotojas gauna naują pranešimą, akimirksniu pasirodo informacinis langas, kuriame matomas siuntėjas ir pranešimo antraštė. Nereikia papildomai perkrauti puslapio tam, kad patikrinti, ar buvo atsiųstas naujas pranešimas. 2.1. Sistema išveda paieškos formą. 3.1. Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus. 4.1. Sistema išveda naujo pranešimo siuntimo formą. 5.1. Sistema išsaugo įrašą. Gavėjo darbiname lange akimirksniu pasirodo informacinis langas, kuriame matomas siuntėjas ir pranešimo antraštė. 6.1. Sistema išsaugo atsaką. Gavėjo darbiname lange akimirksniu pasirodo informacinis langas, kuriame matomas siuntėjas ir pranešimo antraštė. 7.1. Sistema išsaugo pranešimo šalinimo žymę.



3.10 pav. Kliento panaudojimo atvejo „Pildyti poreikių analizės formą“ veiklos diagrama

3.13 lentelė PA „Pildyti poreikių analizės formą“ specifikacija

Panaudojimo atvejis	Pildyti poreikių analizės formą
Numeris	PA8
Aktorius	Klientas
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Klientas nori pildyti projekto poreikių analizės formą.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi kliento teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka meniu punktą „Pildyti analizės formą“. 2.Vartotojas užpildo formą.	1.1.Sistema išveda projekto analizės formą. 2.1.Sistema išsaugo formos duomenis ir išsiunčia el. paštu.
Po sąlyga	Analizės forma užpildyta ir išsiųsta
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyimo variantai	1.Vartotojas pasirenka meniu punktą „Pildyti analizės formą“. 2.Vartotojas užpildo formą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-



3.11 pav. Kliento panaudojimo atvejo „Peržiūrėti išrašytas sąskaitas“ veiklos diagrama

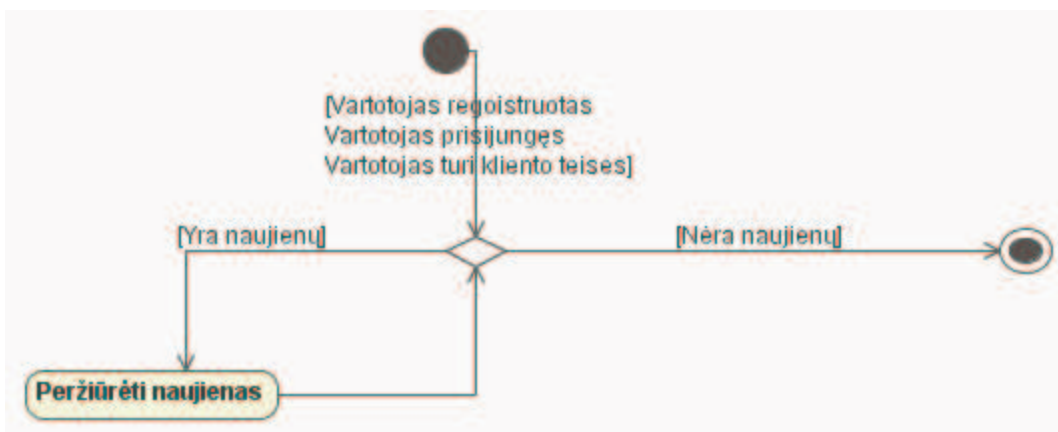
3.14 lentelė PA „Peržiūrėti išrašytas sąskaitas“ specifikacija

Panaudojimo atvejis	Peržiūrėti išrašytas sąskaitas
Numeris	PA9
Aktorius	Klientas
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Klientas nori peržiūrėti išrašytas sąskaitas
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi kliento teises Klientas registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka sąskaitos paieškos punktą. 2. Vartotojas įveda ieškomą žodį. 3. Vartotojas pasirenka sąskaitos peržiūros punktą.	1.1. Sistema išveda paieškos formą. 2.1. Sistema išveda rastus įrašus. 3.1. Sistema išveda sąskaitos duomenis.
Po sąlyga	Klientas peržiūrėjo išrašytas sąskaitas
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdymo variantai	1. Vartotojas pasirenka paieškos punktą 2. Vartotojas įveda ieškomą žodį 3. Vartotojas pasirenka sąskaitos peržiūros punktą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai)	Turi būti palaikoma sesija.

reikalavimai	
Ryšiai su kitais PA	Apima PA „Atlikti paiešką“
Pastabos	-
Neišspręstos problemos	-

3.15 lentelė PA „Peržiūrėti išrašytas sąskaitas“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka išrašytų sąskaitų punktą 2. Vartotojas pasirenka sąskaitos paieškos punktą. 3. Vartotojas įveda ieškomą žodį. 4. Vartotojas pasirenka sąskaitos peržiūros punktą.	1.1. Sistema parodo išrašytas sąskaitas. Jei yra neapmokėtų sąskaitų, pasirodo informacinis langas, kuriame matomos neapmokėtos sąskaitos. Administratoriui patvirtinus sąskaitos apmokėjimą, sąrašas pasikeičia realiu laiku. 2.1. Sistema išveda paieškos formą. 3.1. Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus. 4.1. Sistema išveda sąskaitos duomenis.



3.12 pav. Kliento panaudojimo atvejo „Peržiūrėti naujienas“ veiklos diagrama

3.16 lentelė PA „Peržiūrėti naujienas“ specifikacija

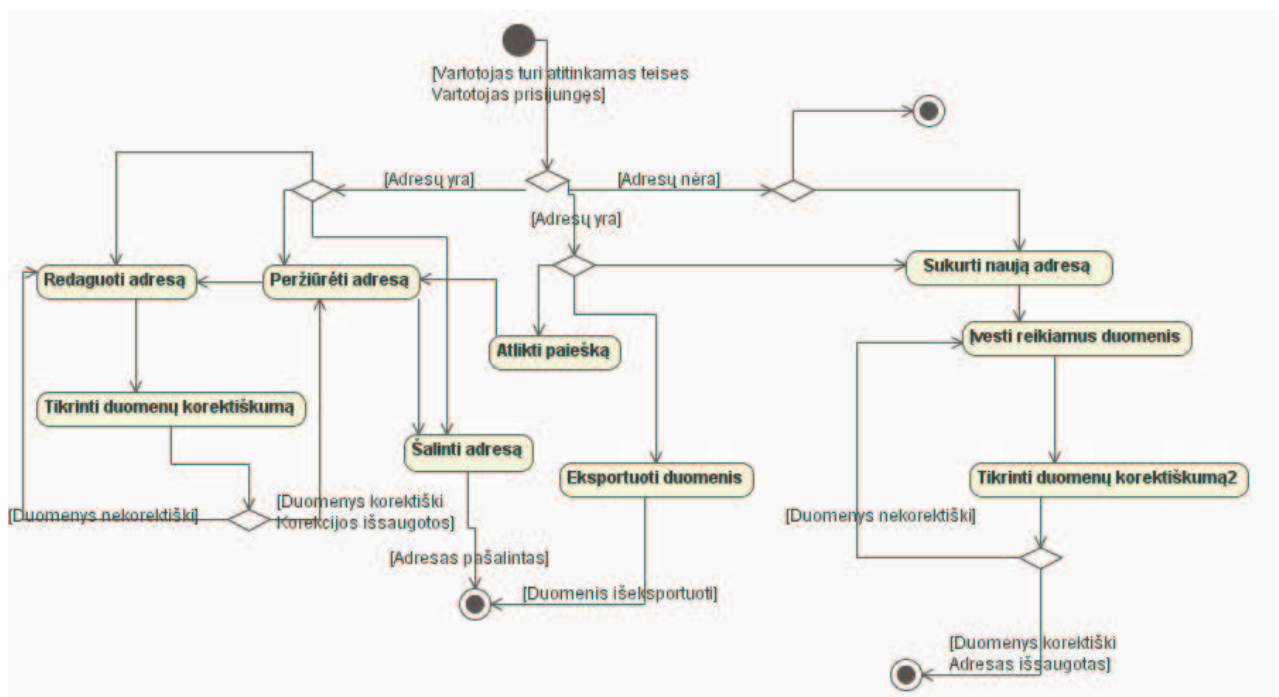
Panaudojimo atvejis	Peržiūrėti naujienas
Numeris	PA10
Aktorius	Klientas
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Klientas nori peržiūrėti naujienas
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi kliento teises Klientas registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka naujienos peržiūros punktą.	1.1. Sistema išveda naujienos turinį.
Po sąlyga	Klientas peržiūrėjo naujienas
Alternatyvos (nesėkmės)	-

atvejai)	
Vykdyto variantai	1. Vartotojas pasirenka naujienos peržiūros punktą.
Veiklos taisyklės	-
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-

3.17 lentelė PA „Peržiūrėti naujienas“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka naujienos peržiūros punktą.	1.1. Sistema išveda naujienos turinį. Naujienų sąrašas keičiasi realiu laiku. Nereikia papildomai perkrauti puslapio tam, kad patikrinti, ar buvo patalpinta nauja naujiena.

Bendros visiems sistemos vartotojams panaudojimo atvejų specifikacijos pavaizduotos 3.13 – 3.17 paveikslėliuose ir detaliau panagrinėtos lentelėse.



3.13 pav. Bendro vartotojų panaudojimo atvejo „Naudotis adresų knygele“ veiklos diagrama

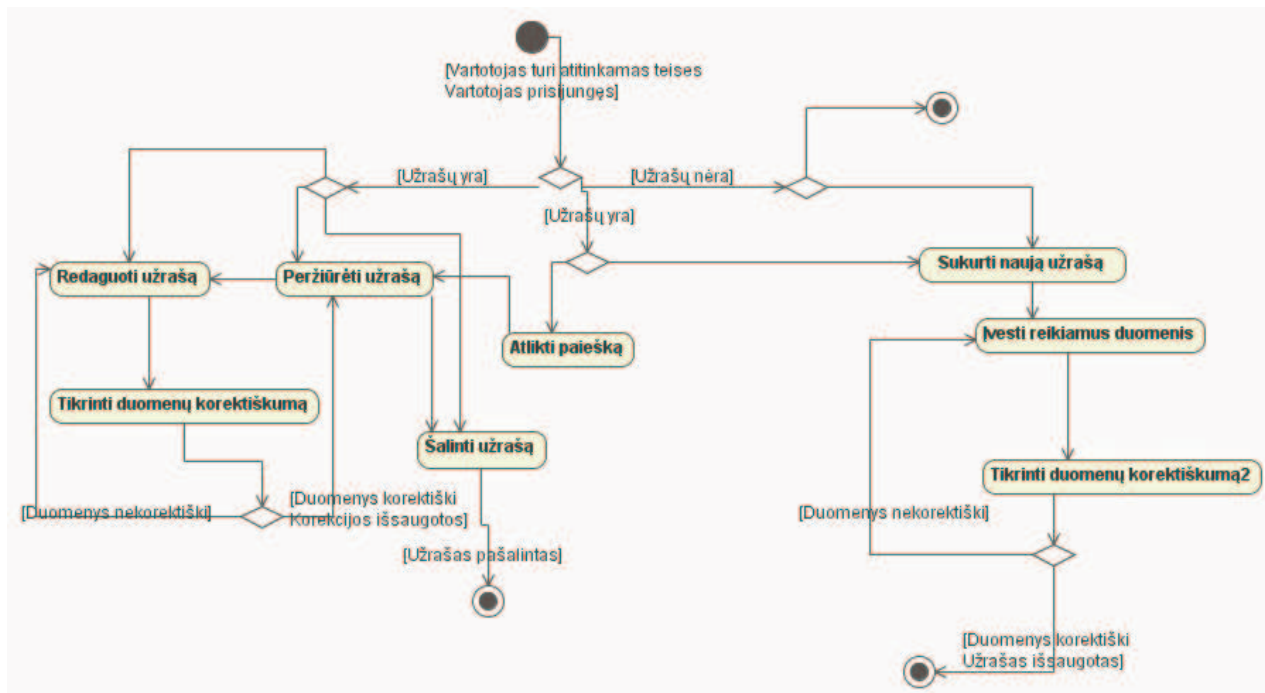
3.18 lentelė PA „Naudotis adresų knygele“ specifikacija

Panaudojimo atvejis	Naudotis adresų knygele
Numeris	PA11
Aktorius	Klientas ir administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Vartotojas nori naudotis adresų knygele.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos

	Vartotojas turi atitinkamas teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka adreso paieškos punktą. 2.Vartotojas įveda ieškomą žodį. 3.Vartotojas pasirenka naujo adreso punktą. 4.Vartotojas suveda naujo adreso duomenis. 5.Vartotojas redaguoja adresą. 6.Vartotojas šalina pranešimą. 7.Vartotojas eksportuoja adresus.	1.1.Sistema išveda paieškos formą. 2.1.Sistema išveda rastus įrašus. 3.1.Sistema išveda naujo adreso sukūrimo formą. 4.1.Sistema išsaugo įrašą. 5.1.Sistema išsaugo pakeitimus. 6.1.Sistema pašalina adresą. 7.1.Sistema išeksportuoja duomenis.
Po sąlyga	Vartotojas pasinaudojo adresų knygele.
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyto variantai	1. Vartotojas pasirenka paieškos punktą 2. Vartotojas įveda ieškomą žodį 3. Vartotojas pasirenka naujo įrašo punktą 4. Vartotojas suveda naujo adreso duomenis 5. Vartotojas redaguoja adresą. 6. Vartotojas šalina adresą. 7. Vartotojas eksportuoja adresus.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Šalinti adresą“, „Atlikti paiešką“, „Redaguoti adresą“, „Sukurti naują adresą“ ir „Eksportuoti duomenis“
Pastabos	-
Neišspręstos problemos	-

3.19 lentelė PA „Naudotis adresų knygele“ patobulinimo specifikacija

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka adreso paieškos punktą. 2.Vartotojas įveda ieškomą žodį. 3.Vartotojas pasirenka naujo adreso punktą. 4.Vartotojas suveda naujo adreso duomenis. 5.Vartotojas redaguoja adresą. 6.Vartotojas šalina pranešimą. 7.Vartotojas eksportuoja adresus.	1.1.Sistema išveda paieškos formą. 2.1.Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus. 3.1.Sistema išveda naujo adreso sukūrimo formą. 4.1.Sistema automatiškai realiu laiku tikrina įvedamų duomenų teisingumą dar prieš išsiunčiant formos duomenis. Sistema išsaugo įrašą. Pagal įvestus duomenis matoma kontakto buvimo vieta integruotoje “Google Maps” taikomojoje programoje. 5.1.Sistema automatiškai realiu laiku tikrina įvedamų duomenų teisingumą dar prieš išsiunčiant formos duomenis. Sistema išsaugo pakeitimus. 6.1.Sistema pašalina adresą. 7.1.Sistema išeksportuoja duomenis.



3.14 pav. Bendro vartotojų panaudojimo atvejo „Naudotis užrašų knygele“ veiklos diagrama

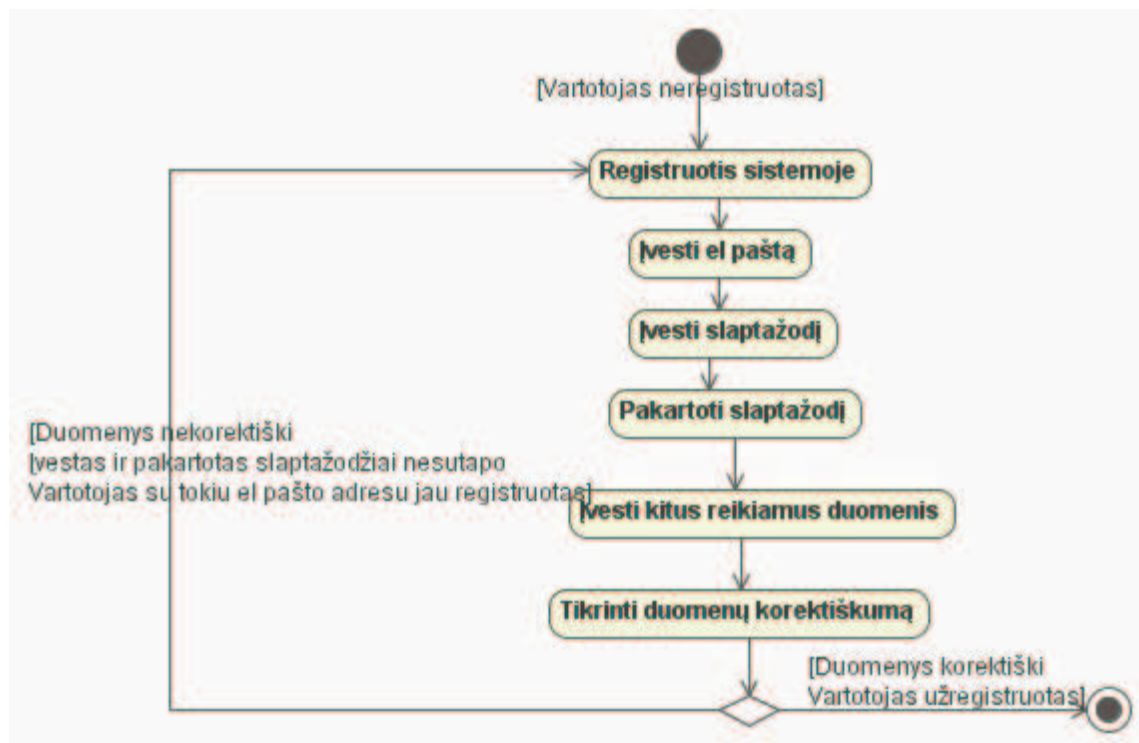
3.20 lentelė PA „Naudotis užrašų knygele“ specifikacija

Panaudojimo atvejis	Naudotis užrašų knygele
Numeris	PA12
Aktorius	Klientas ir administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Vartotojas nori naudotis adresų knygele.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi atitinkamas teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka užrašo paieškos punktą. 2.Vartotojas įveda ieškomą žodį. 3.Vartotojas pasirenka naujo užrašo punktą. 4.Vartotojas suveda naujo užrašo duomenis. 5.Vartotojas redaguoja užrašą. 6.Vartotojas šalina užrašą.	1.1.Sistema išveda paieškos formą. 2.1.Sistema išveda rastus įrašus. 3.1.Sistema išveda naujo užrašo sukūrimo formą. 4.1.Sistema išsaugo įrašą. 5.1.Sistema išsaugo pakeitimus. 6.1.Sistema pašalina užrašą.
Po sąlyga	Vartotojas pasinaudojo užrašų knygele.
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyimo variantai	1. Vartotojas pasirenka paieškos punktą 2. Vartotojas įveda ieškomą žodį 3. Vartotojas pasirenka naujo įrašo punktą 4. Vartotojas suveda naujo užrašo duomenis 5. Vartotojas redaguoja užrašą.

	6. Vartotojas šalina užrašą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	Apima PA „Šalinti užrašą“, „Atlikti paiešką“, „Redaguoti užrašą“, „Sukurti naują užrašą“
Pastabos	-
Neišspręstos problemos	-

3.21 lentelė PA „Naudotis adresu knygele“ patobulinimo specifikacija

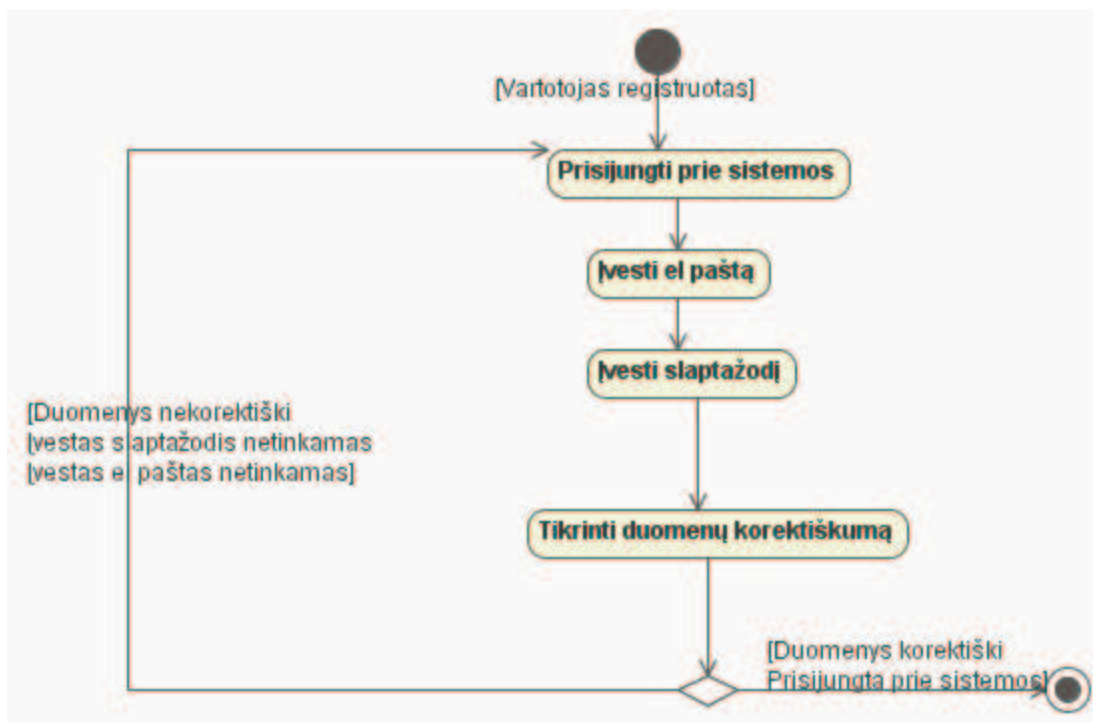
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka užrašo paieškos punktą. 2.Vartotojas įveda ieškomą žodį. 3.Vartotojas pasirenka naujo užrašo punktą. 4.Vartotojas suveda naujo užrašo duomenis. 5.Vartotojas redaguoja užrašą. 6.Vartotojas šalina užrašą.	1.1.Sistema išveda paieškos formą. 2.1.Pagal vedama paieškos žodį sistema automatiškai pasiūlo galimus variantus dar prieš išsiunčiant formos duomenis. Po duomenų išsiuntimo sistema išveda rastus įrašus.. 3.1.Sistema išveda naujo užrašo sukūrimo formą. 4.1.Sistema automatiškai realiu laiku tikrina įvedamų duomenų teisingumą dar prieš išsiunčiant formos duomenis. Sistema išsaugo įrašą. 5.1.Sistema automatiškai realiu laiku tikrina įvedamų duomenų teisingumą dar prieš išsiunčiant formos duomenis. Sistema išsaugo įrašą. 6.1.Sistema pašalina užrašą.



3.15 pav. Bendro vartotojų panaudojimo atveju „Registruotis sistemoje“ veiklos diagrama

Panaudojimo atvejis	Registruotis sistemoje
Numeris	PA13
Aktorius	Klientas ir administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadinimo sąlyga:	Vartotojas nori registruotis sistemoje.
Prieš sąlyga	Vartotojas turi būti neregistruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka registracijos punktą. 2.Vartotojas suveda registracijos duomenis. 3.Vartotojas pasirenka slaptažodžio priminimo punktą. 4.Vartotojas įveda prisijungimo vardą.	1.1.Sistema išveda registracijos formą. 2.1.Sistema išsaugo duomenis ir išsiunčia pranešimą el. paštu. 3.1.Sistema išveda slaptažodžio priminimo formą. 4.1.Sistema išsiunčia prisijungimo duomenis el. paštu.
Po sąlyga	Vartotojas prisiregistravo sistemoje.
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyimo variantai	1.Vartotojas pasirenka registracijos punktą. 2.Vartotojas suveda registracijos duomenis. 3.Vartotojas pasirenka slaptažodžio priminimo punktą. 4.Vartotojas įveda prisijungimo vardą.
Veiklos taisyklės	Suvedama informacija turi būti korektiška. Pakartotas slaptažodis turi sutapti. Vartotojas tuo pačiu el. pašto adresu negali antrą kart registruotis.
Specialūs (nefunkciniai) reikalavimai	-
Ryšiai su kitais PA	Apima PA „Priminti slaptažodį“
Pastabos	-
Neišspręstos problemos	-

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas pasirenka registracijos punktą. 2.Vartotojas suveda registracijos duomenis. 3.Vartotojas pasirenka slaptažodžio priminimo punktą. 4.Vartotojas įveda prisijungimo vardą.	1.1.Sistema išveda registracijos formą. 2.1.Sistema automatiškai realiu laiku tikrina įvedamų duomenų teisingumą dar prieš išsiunčiant formos duomenis. Sistema išsaugo duomenis ir išsiunčia pranešimą el. paštu. 3.1.Sistema išveda slaptažodžio priminimo formą. 4.1.Sistema automatiškai realiu laiku tikrina, ar įvedamas prisijungimo vardas egzistuoja dar prieš išsiunčiant formos duomenis. Sistema išsiunčia prisijungimo duomenis el. paštu.

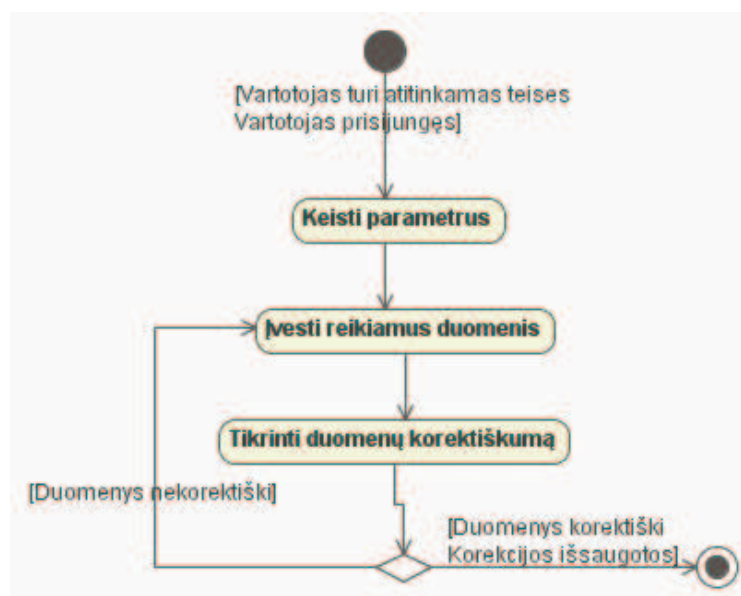


3.16 pav. Bendro vartotojų panaudojimo atvejo „Prisijungti prie sistemos“ veiklos diagrama

3.24 lentelė PA „Prisijungti prie sistemos“ specifikacija

Panaudojimo atvejis	Prisijungti prie sistemos
Numeris	PA14
Aktorius	Klientas ir administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadavimo sąlyga:	Vartotojas nori prisijungti prie sistemos.
Prieš sąlyga	Vartotojas turi būti registruotas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas įveda prisijungimo informaciją.	1.1. Sistema patikrina įvestus duomenis, jeigu jie teisingi, atidaro darbo sesiją, suteikia atitinkamas vartotojo teises.
Po sąlyga	Vartotojas prisijungė prie sistemos.
Alternatyvos (nesėkmės atvejai)	1.1.a. Blogas prisijungimo vardas. Sistema apie tai išveda pranešimą, grįžta į 1 žingsnį 1.1.b. Blogas prisijungimo slaptažodis. Sistema apie tai išveda pranešimą, grįžta į 1 žingsnį
Vykdomo variantai	1. Vartotojas įveda prisijungimo duomenis.
Veiklos taisyklės	Suvedama informacija turi būti korektiška.
Specialūs (nefunkciniai) reikalavimai	Vartotojo prisijungimo duomenys privalo būti perduodami internetu saugiai, turi būti sukuriama ir tikrinama sesija bei jos duomenys
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-

Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas įveda prisijungimo informaciją.	1.1.Sistema automatiškai realiu laiku tikrina, ar įvedamas prisijungimo vardas egzistuoja, ir ar tinka įvedamas slaptažodis dar prieš išsiunčiant formos duomenis. Sistema patikrina įvestus duomenis, jeigu jie teisingi, atidaro darbo sesiją, suteikia atitinkamas vartotojo teises.

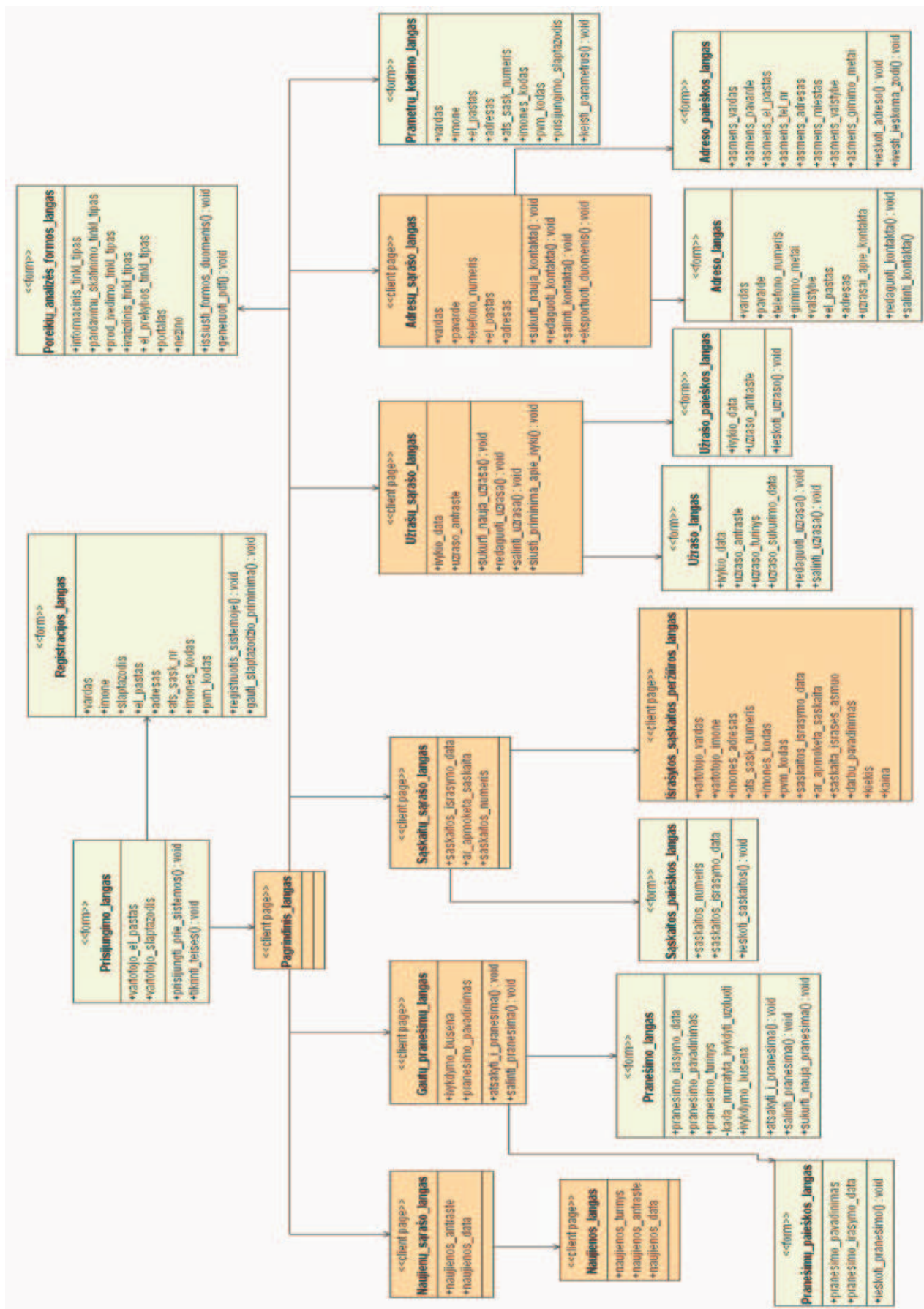


3.17 pav. Bendro vartotojų panaudojimo atvejo „Keisti parametrus“ veiklos diagrama

3.26 lentelė PA „Keisti parametrus“ specifikacija

Panaudojimo atvejis	Keisti parametrus
Numeris	PA15
Aktorius	Klientas ir administratorius
Sistema	Automatizuoto pirkėjų aptarnavimo IS internete
Sužadavimo sąlyga:	Vartotojas nori keisti parametrus.
Prieš sąlyga	Vartotojas turi būti prisijungęs prie sistemos Vartotojas turi atitinkamas teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1.Vartotojas redaguoja informaciją apie save.	1.1.Sistema išsaugo pakeitimus.
Po sąlyga	Vartotojas pakeičią parametrus.
Alternatyvos (nesėkmės atvejai)	Suvedant netinkamus duomenis išvedamas klaidos pranešimas.
Vykdyimo variantai	1.Vartotojas redaguoja informaciją apie save.
Veiklos taisyklės	Suvedama informacija turi būti korektiška. Pakartotas slaptažodis turi sutapti.
Specialūs (nefunkciniai) reikalavimai	Turi būti palaikoma sesija.
Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-

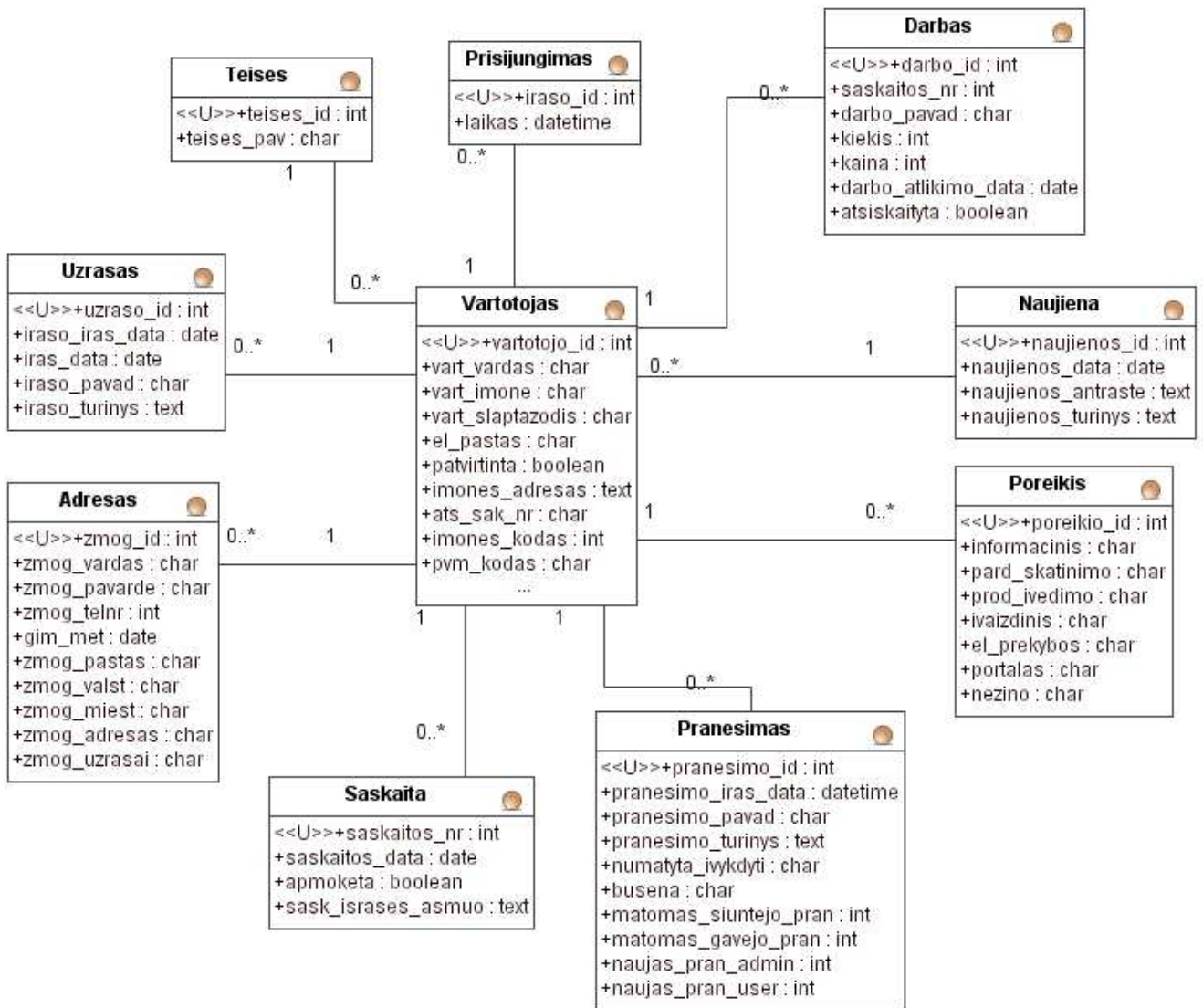
Kliento navigavimo planas pavaizduotas 3.19 paveikslėlyje.



3.19 pav. Kliento navigavimo planas

3.4. Dalykinės srities modelis

Dalykinės srities klasių diagrama (3.20 pav.) buvo sudaryta naudojantis analizės klasių diagrama. Dalykinės srities esybių klasių diagrama – koncepcinis duomenų modelis; tai nėra galutinis duomenų modelio projektas – vėlesnėse projektavimo iteracijose jis bus patikslintas ir sudaryta duomenų bazės schema. Duomenų klasių paskirtis aprašyta 3.27 lentelėje.



3.20 pav. Dalykinės srities klasių diagrama

3.27 lentelė. Duomenų klasių paskirtis

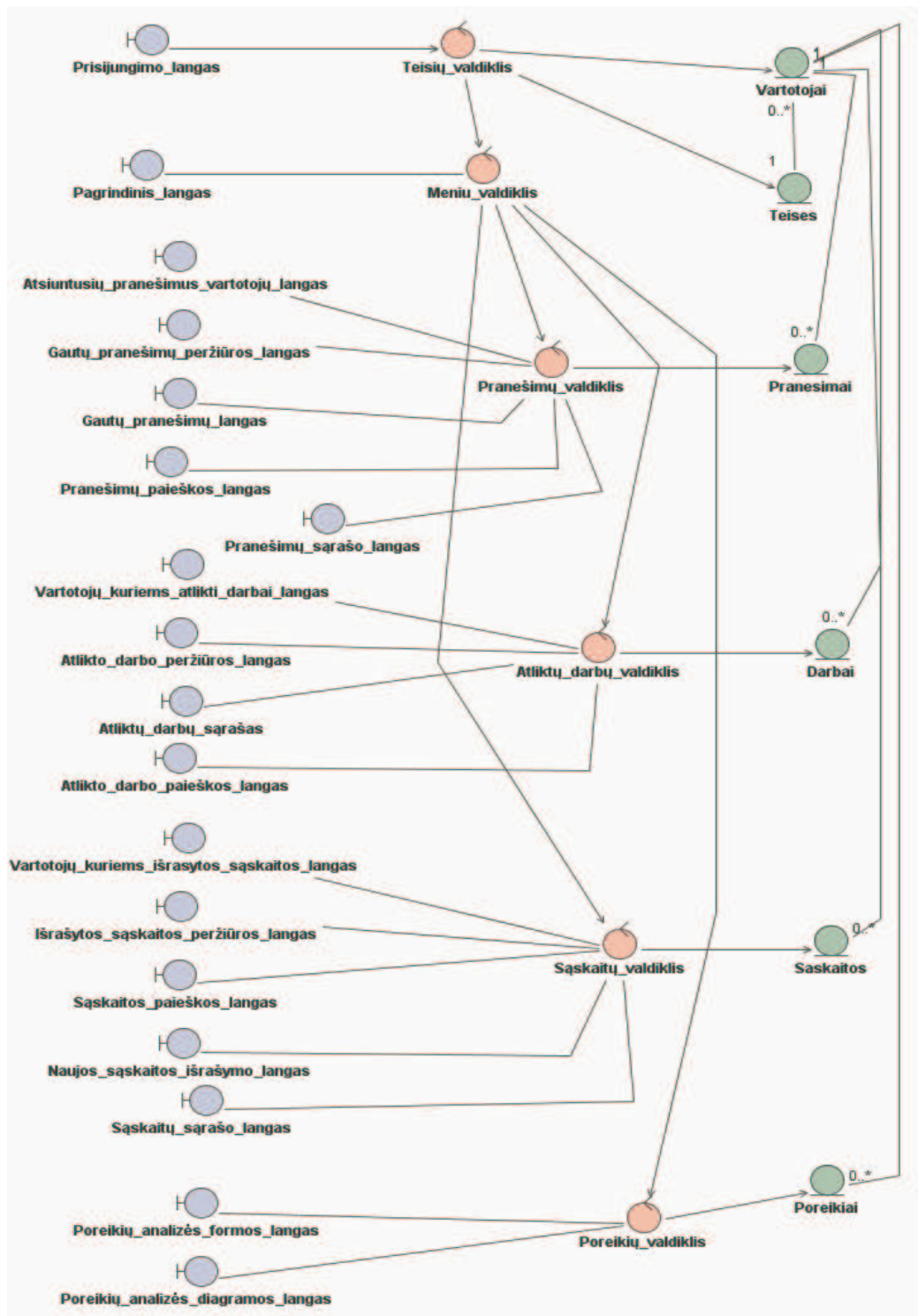
Klasė	Paskirtis
Vartotojas	Klasė atspindi sistemos vartotojus bei saugo informaciją apie juos.
Teises	Tai teisių klasė, kur saugomi duomenys apie egzistuojančias teises, kurios gali būti priskirtos skirtingiems sistemos vartotojams.
Prisijungimas	Statistikos klasė, kur saugoma informacija apie vartotojų naudojimosi sistema statistiką.
Poreikis	Poreikių klasė, saugomi vartotojų užpildytos poreikių analizės formos duomenys.

Naujiena	Naujienu klasė, šioje klasėje saugomos administratoriaus sukurtos naujienos.
Darbas	Darbu klasė, kurioje saugoma informacija apie klientams atliktus darbus.
Pranešimas	Pranešimų klasė, kurioje saugomi tarp klientų ir administratorių parašyti pranešimai.
Saskaita	Sąskaitų klasė, kurioje saugomos klientams išrašytos sąskaitos.
Adresas	Adresų klasė, saugomi sistemos vartotojų įvesti adresai.
Užrašas	Užrašų klasė, saugomi sistemos vartotojų įvesti užrašai.

3.5. Reikalavimų analizė

Analizės (*robustness*) diagrama vaizduoja kaip realizuojami panaudojimo atvejai. Analizės diagramos sudarytos apjungiant logiškai susijusius panaudojimo atvejus. Gautos trys diagramos – pagrindinių sistemos funkcijų, papildomų sistemos resursų ir klientų administravimo.

Pagrindinių sistemos funkcijų analizės diagrama pavaizduota 3.21 paveikslėlyje.



3.21 pav. Pagrindinių sistemos funkcijų analizės diagrama

Sistema funkcionuoja tokia tvarka: pirmiausia vartotojas suveda savo prisijungimo duomenis „Prisijungimo lange“, kuriuos „Teisių valdiklis“ tikrina su duomenimis saugomais duomenų bazėje (kreipiamasi į esybes „Teisės“ ir „Vartotojas“). Jei vartotojo duomenys teisingi, kreipiamasi į „Menu valdiklį“, kuris suformuoja „Pagrindinį langą“.

Vartotojui pasirinkus pranešimų punktą pagrindiniame lange „Menu valdiklis“ kreipiasi į „Pranešimų valdiklį“, kuris naudodamasis esybe „Pranesimas“ suformuoja ir pateikia „Pranešimų sąrašą“ langą. Visus kitus veiksmus, susijusius su pranešimais, valdo „Pranešimų valdiklis“.

Vartotojui pasirinkus darbų punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Atliktų darbų valdiklį*“, kuris naudodamasis esybe „*Darbas*“ suformuoja ir pateikia „*Atliktų darbų sąrašo*“ langą. Visus kitus veiksmus, susijusius su darbų administravimu, valdo „*Atliktų darbų valdiklis*“.

Vartotojui pasirinkus sąskaitų punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Sąskaitų valdiklį*“, kuris naudodamasis esybe „*Saskaita*“ suformuoja ir pateikia „*Sąskaitų sąrašo*“ langą. Visus kitus veiksmus, susijusius su sąskaitų administravimu, valdo „*Sąskaitų valdiklis*“.

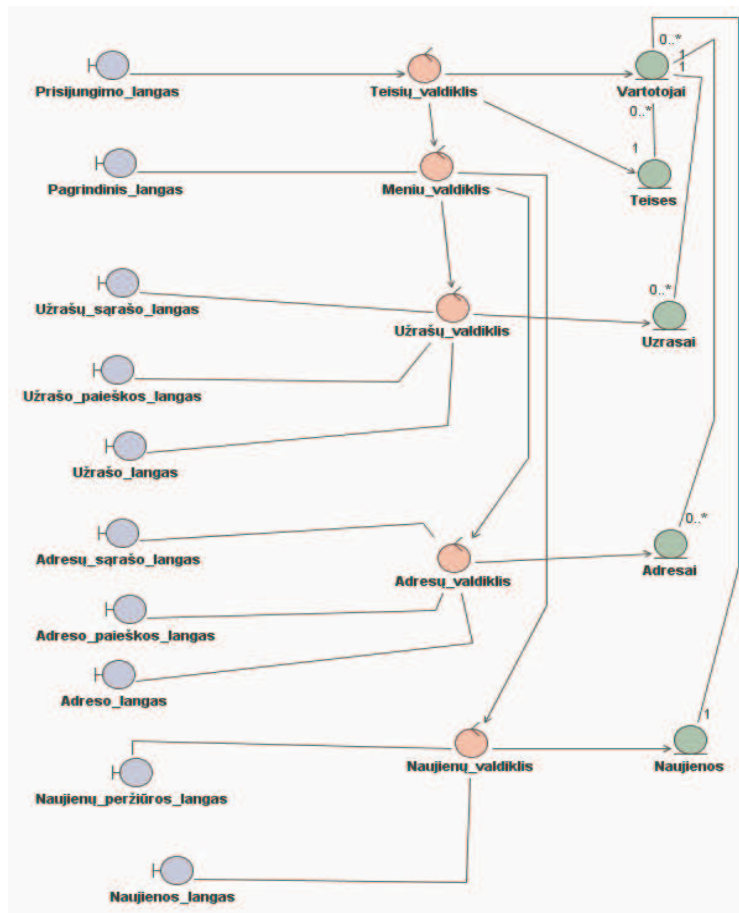
Vartotojui pasirinkus poreikių analizės punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Poreikių valdiklį*“, kuris naudodamasis esybe „*Poreikis*“ suformuoja ir pateikia: klientui - „*Poreikių analizės formos*“ langą, administratoriui – „*Poreikių analizės diagramos*“ langą.

Papildomų sistemos resursų analizės diagrama pavaizduota 3.22 paveikslėlyje.

Prisijungimo prie sistemos seka tokia pati kaip aukščiau aprašytoje diagramoje. Vartotojui pasirinkus užrašų punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Užrašų valdiklį*“, kuris naudodamasis esybe „*Uzrasas*“ suformuoja ir pateikia „*Užrašų sąrašo*“ langą. Visus kitus veiksmus, susijusius su užrašais, valdo „*Užrašų valdiklis*“.

Vartotojui pasirinkus adresų punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Adresų valdiklį*“, kuris naudodamasis esybe „*Adresas*“ suformuoja ir pateikia „*Adresų sąrašo*“ langą. Visus kitus veiksmus, susijusius su adresais, valdo „*Adresų valdiklis*“.

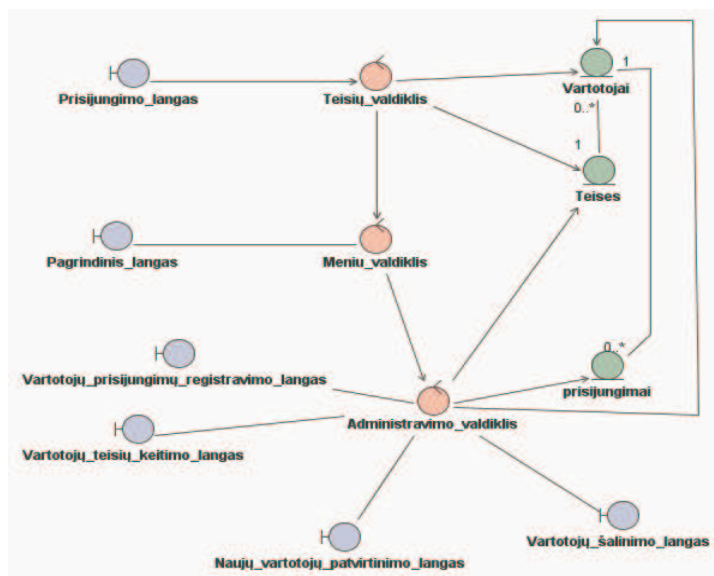
Vartotojui pasirinkus naujienų punktą pagrindiniame lange „*Meniu valdiklis*“ kreipiasi į „*Naujienų valdiklį*“, kuris naudodamasis esybe „*Naujiena*“ suformuoja ir pateikia „*Naujienų*“ langą.



3.22 pav. Papildomų sistemos resursų analizės diagrama

Klientų administravimo analizės diagrama pavaizduota žemiau 3.23 paveikslėlyje.

Vartotojui pasirinkus administravimo punktą pagrindiniame lange „Meniu valdiklis“ kreipiasi į „Administravimo valdiklį“, kuris naudodamasis esybėmis „Vartotojas“ ir „Prisijungimas“ suformuoja ir pateikia „Vartotojų prisijungimų registravimo“ langą. Vartotojui pasirinkus kitus administravimo punktus „Administravimo valdiklio“ pagalba atidaromi atitinkami langai.



3.23 pav. Klientų administravimo analizės diagrama

3.6. Reikalavimų analizės apibendrinimas

Reikalavimų analizės metu kiekvienam panaudojimo atvejui buvo sudaryta analizės diagramos vaizduojančios, kaip realizuojami panaudojimo atvejai. Taip pat buvo sudaryta dalykinės srities klasių diagrama, sistemos vartotojų navigavimo planai ir kiekvienam panaudojimo atvejui nubraižytos veiklos diagramos bei detalios aprašytos specifikacijų lentelės. Reikalavimų analizė buvo atlikta tam, kad pereiti prie sistemos projekto modelių sudarymo.

4. CRM sistemos tobulinimo projektas

Šiame skyriuje yra pagrindžiamas sistemos įgyvendinimui pasirinktas metodas, išdėstoma metodo pagrindinė esmė. Taip pat pavaizduojama sistemos architektūra, sudaromas sistemos elgsenos modelis bei suprojektuojama duomenų bazės schema.

4.1. Sistemos įgyvendinimui pasirinkto metodo pagrindimas ir esmės išdėstymas

Analizės etape pasirinktas JavaScript/DHTML įgyvendinimo metodas (Asinchroninis JavaScript + XML) dėl to, kad gerai pritampa prie egzistuojančių HTML Web taikomųjų programų (automatizuota pirkėjų aptarnavimo sistema realizuota šiuo pagrindu) bei yra tinkamas DHTML Web programuotojams. Šis metodas paprastai naudoja naršykle pagrįstą lengvo svorio JavaScript biblioteką kliento pusės apdorojimui, tokiam kaip dalinio ekrano atnaujinimo atlikimas ir asinchroninis foninis tinklo komunikavimas. Šiame metode vartotojo interfeisas paprastai yra apibrėžiamas naudojant DHTML, o kliento pusės taikomosios programos logika yra suprogramuojama naudojant JavaScript.

DHTML/JavaScript metodas labai gerai tinka egzistuojančiose HTML Web taikomose programose. HTML programuotojui yra lengviau programuoti su JavaScript ir tai yra ne toks brangus būdas egzistuojančiai taikomajai programai suteikti daugiau interaktyvumo. Paprastai šis metodas reikalauja parsisiųsti JavaScript biblioteką, tam kad kompensuoti skirtingas kliento naršyklės galimybes. Gauta taikomoji programa bus tiek raiški, kiek tai leis naršyklė. Šis metodas gali būti problematiškas didelės aprėpties biznio taikomose programose dėl JavaScript programavimo ir palaikymo sunkumų, bei DHTML funkcionalumo apribojimų.

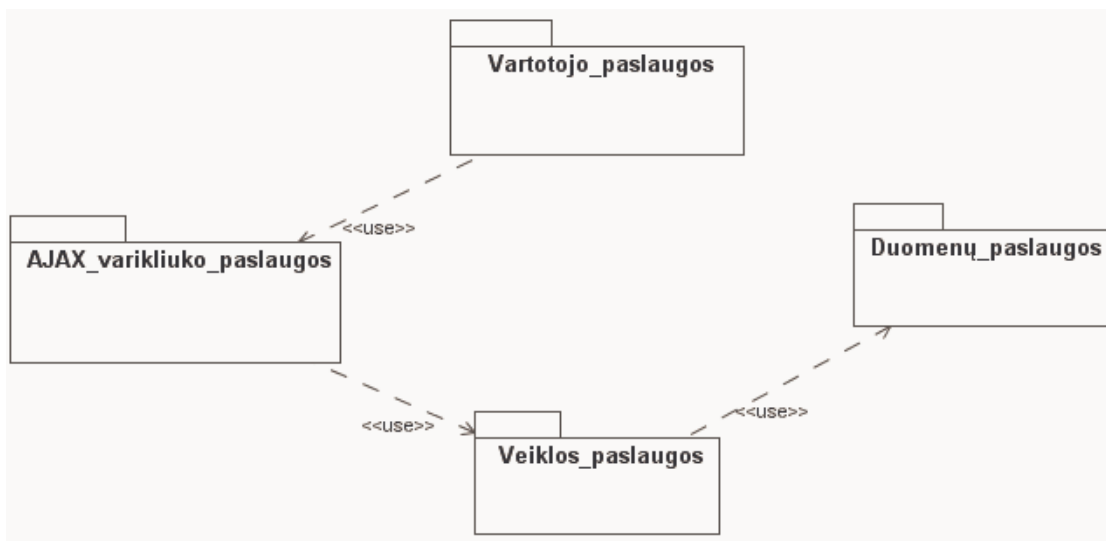
Kadangi Ajax apjungia aibę technologijų – reikia surasti priemonę, kuri padėtų suvaldyti tas technologijas. Be to pats vienas Ajax negalės sukurti viso puslapio, reikės papildomai naudoti ir kitas technologijas, kurios galėtų paprastai ir lengvai atvaizduoti visą reikiamą informaciją. Ajax tik priemonė padaryti puslapius labiau dinamiškus. Taigi CRM sistemos interfeiso patobulinimui bus panaudota konkrečiai PHP ir AJAX technologijų derinys. PHP - serverio pusės programavimo technologija, kuri realizuoja programos logiką bei HTML kodo generavimą realiu laiku. Tai plačiai naudojama atviro kodo bendros paskirties interpretuojama programavimo kalba, ypač tinkanti Web programų kūrimui. Išskirtinis PHP bruožas yra tai, kad ji gali būti įterpta į HTML kodą. Veikia bet kokiose operacinėse sistemose, kur veikia Apache Web serveris. PHP yra puikiai pritaikytas Web puslapių vartotojo sąsajos kūrimui, nėra labai sudėtinga sintaksė, didelis funkcionalumas, be to, nemokama licencija. PHP turi daug modulių ir standartinių funkcijų, skirtų darbui su duomenų bazėmis (mySQL, Oracle, MSSQL ir kt.), interneto protokolais (SMTP, IMAP, POP3 ir kt.) ir teksto apdorojimui.

4.2. Sistemos architektūra - statinės struktūros modelis

Šiame poskyryje pateikiama loginė sistemos architektūra, sudaromi vartotojo, veiklos, duomenų ir Ajax varikliuko paslaugų modeliai

4.2.1. Loginė visos sistemos architektūra

Sistemos architektūros kūrimui panaudotas tipinis trijų lygių architektūros modelis, kuris papildomas AJAX varikliuko paslaugomis (4.1 pav). Remiantis šiuo modeliu kuriamą sistemą sudaro vartotojo, veiklos bei duomenų paslaugos, kurios sugrupuojamos į atskirus paketus. Tarp vartotojo ir veiklos paslaugų įsiterpia AJAX varikliuko paslaugas.



4.1 pav. Loginis sistemos architektūros modelis

Tobulinama CRM sistema yra sukurta pagal MVC architektūros modelį. Iš tiesų tai yra struktūrinis šablonas (design pattern), pagal kurio architektūrą yra kuriamos paprastos programos, o ne web aplikacijos.

MVC architektūros naudojimas web aplikacijose atsirado seniai tačiau pati “banga” prasidėjo tada, kai O’Reilly apibrėžė terminą Web 2.0. Tada, paplitus AJAX technologijų grupei, pradėjus taikyti folksonomiją, į interneto puslapį pradėta žiūrėti kaip į aplikaciją. Žinoma, tokioje situacijoje norint suprogramuoti interneto svetainę kaip aplikaciją reikia taikyti paprastų aplikacijų kūrimo “filosofiją”.

MVC veikimas yra paremtas separation of concerns (SoC) principu. Tai reiškia, kad programa yra išskaidoma į tokias dalis, kad jų funkcionalumas/poreikiai nepersidenginėtų. Šis principas yra taikomas daug kur. Vienas iš pavyzdžių būtų [x]HTML ir CSS. [x]HTML kalbos paskirtis yra nurodyti dokumento struktūrą, kai tuo tarpu CSS nurodo, kaip pats dokumentas turėtų atrodyti.

Objektiškai orientuotame programavime šis principas atsispindi išskaidant užduočių atlikimą į objektus ir jų metodus.

Pačiame MVC šis principas pasireiškia logikos, prezentacijos ir turinio atskyrimu. MVC reiškia Model-View-Controller. Trys dalys atsakingos už savo “veiklos sritį”.

Model - tam tikros srities duomenų valdymo/pateikimo vienetas. Jeigu skamba per sudėtingai, štai paprastesnis apibūdinimas: modelis gali būti atsakingas už mokesčių apskaičiavimą tam tikram pristatymui iš internetinės svetainės, privalo patikrinti ar šiandien yra tam tikro vartotojo gimtadienis, išsiųsti mėnesines naujienas el. paštu prenumeratoriams ir pan.

View - suteikia programos modeliui išvaizdą, tinkamą vartotojui suprasti (formas, sąrašai). Web aplikacijos tai ne kas kita kaip sugeneruotas HTML/CSS kodas.

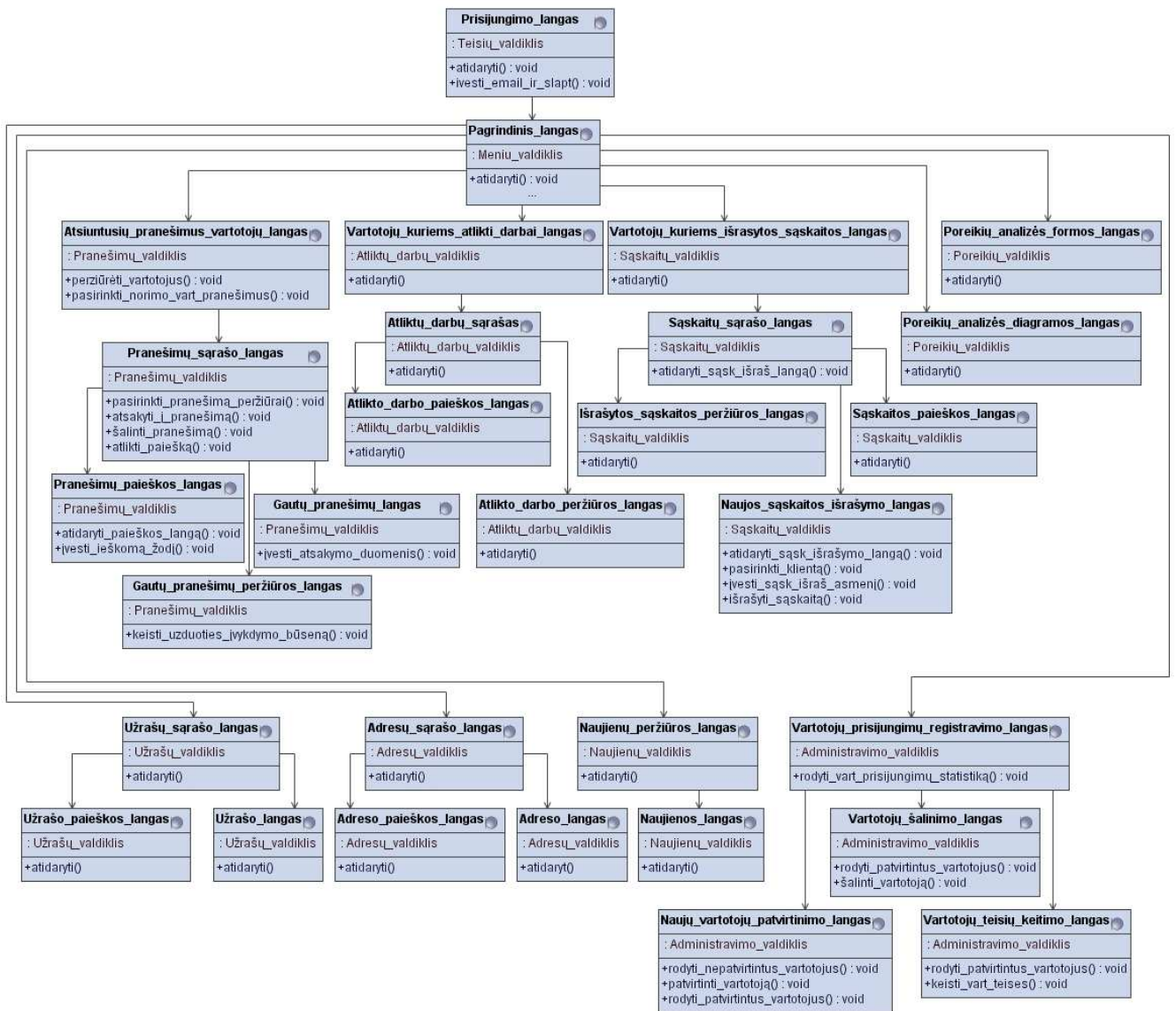
Controller - atsako už modelio valdymą bei “vaizdų” pateikimus pagal atitinkamus veiksmus (vartotojo interakciją): mygtuko paspaudimas, teksto įvedimas ir pan.

4.2.2. Vartotojo paslaugos

Vartotojo paslaugų modelis pateiktas 4.2 paveikslėlyje.

Administratorius, atidaręs prisijungimo langą ir teisingai įvedęs prisijungimo vardą bei slaptažodį, patenka į pagrindinį langą, kuriame pateikiamas meniu. Pasirinkdamas meniu punktus jis patenka į atitinkamus langus: atsiuntusių pranešimus vartotojų sąrašo langą, vartotojų, kuriems atlikti darbai, sąrašo langą, vartotojų, kuriems išrašytos sąskaitos langą, poreikių analizės langą, užrašų sąrašą, adresų sąrašą, naujienų sąrašą, klientų administravimo langą.

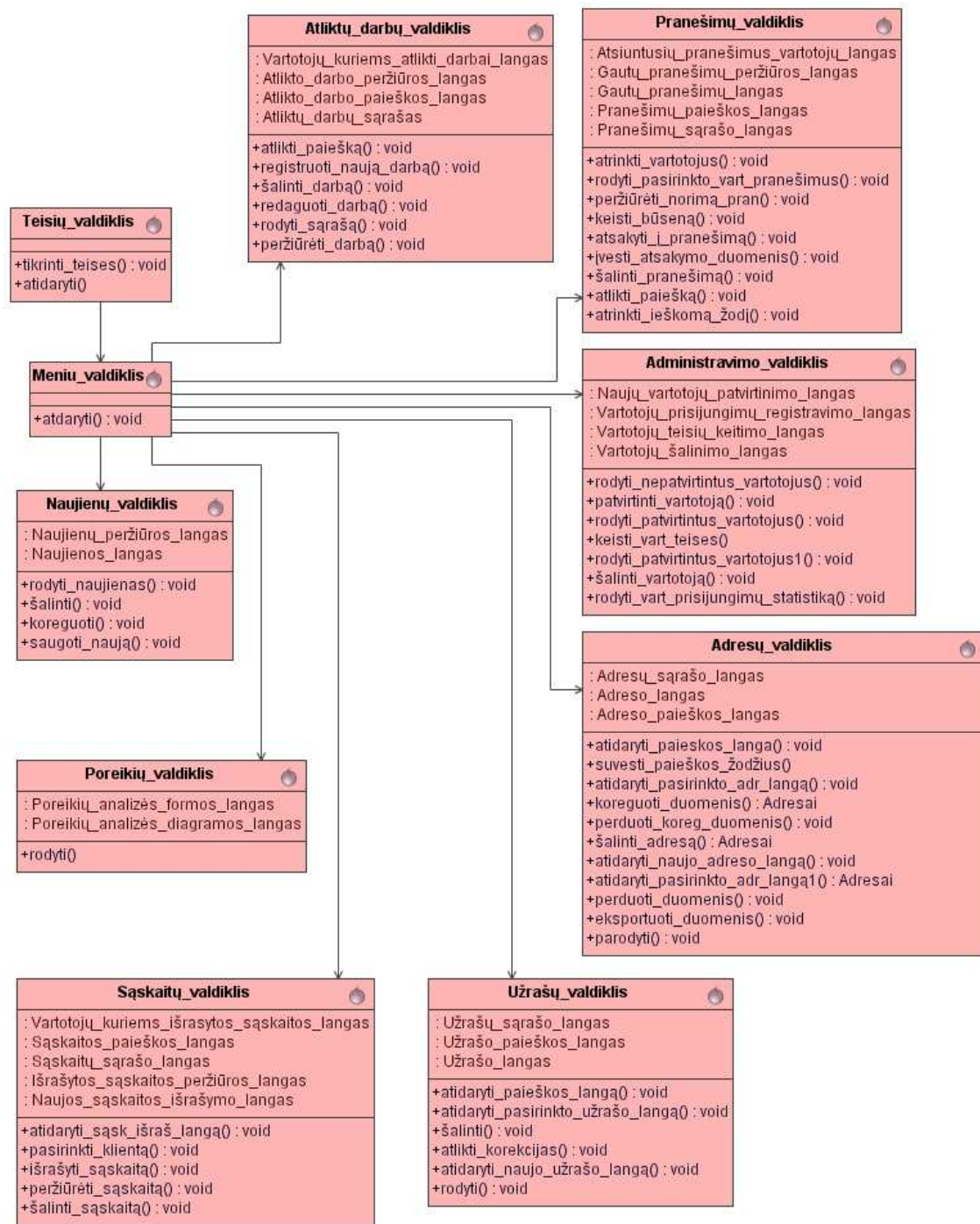
Iš pagrindinių meniu punktų vartotojas patenka į naujų įrašų sukūrimo, redagavimo, peržiūros bei paieškos langus, kurie pavaizduoti 4.2 paveikslėlyje.



4.2 pav. Vartotojo paslaugų modelis

4.2.3. Veiklos paslaugos

Valdymo klasių modelyje (4.3 pav.) pavaizduotos veiklos logiką realizuojančios klasės ir jų metodai. Metodų parametrų sąrašas nenurodytas. Kiekvienos klasės paskirtis aprašyta 4.1 lentelėje.



4.3 pav. Valdymo klasių modelis

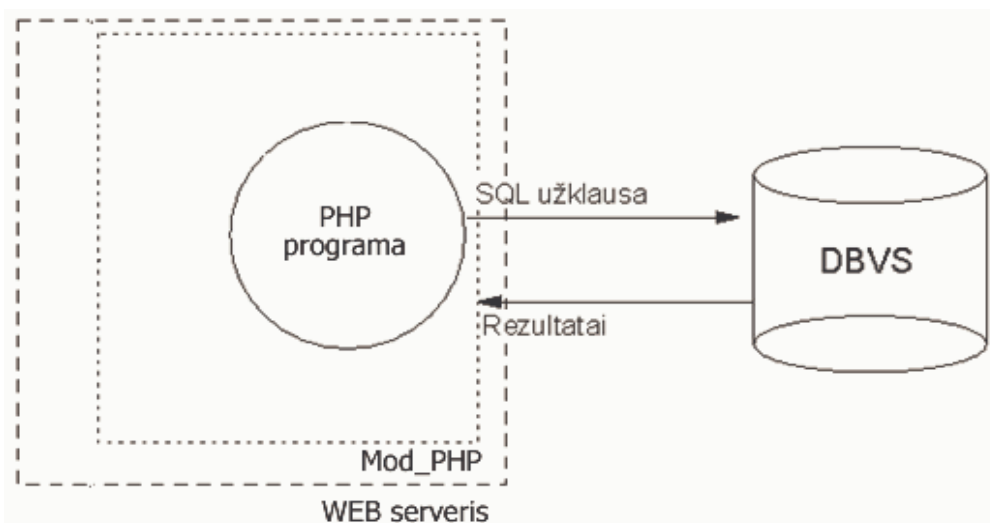
4.1 lentelė. Valdymo klasių paskirtis

Klasės pavadinimas	Paskirtis
Atliktu_darbu_valdiklis	Suformuoti darbų sąrašą, atidaryti naujo darbo sukūrimo, koregavimo, paieškos langus, išsaugoti, šalinti darbą.
Naujienu_valdiklis	Suformuoti naujienu sąrašą, atidaryti naujos naujienos sukūrimo, koregavimo langus, išsaugoti, šalinti naujieną.
Teisiu_valdiklis	Valdyti vartotojų prisijungimą.
Menu_valdiklis	Valdyti pagrindiniame lange pateikiamą meniu. Atverti langą pagal pasirinktą meniu punktą.

Pranešimų_valdiklis	Valdyti pranešimų sąrašo, naujo pranešimo sukūrimo, atsakymo į pranešimą langus, išsaugoti, šalinti pranešimo duomenis.
Sąskaitų_valdiklis	Atidaryti išrašytų sąskaitų sąrašo, naujos sąskaitos išrašymo langus bei valdyti sąskaitos duomenų saugojimą ir šalinimą.
Poreikių_valdklis	Suformuoti poreikių analizės grafiką, poreikių analizės forma bei išsaugoti formos duomenis.
Administravimo_valdiklis	Atidaryti vartotojų patvirtinimo, šalinimo, teisių keitimo, statistikos peržiūros langus. Patvirtinti, pašalinti vartotoją bei keisti teises.
Adresų_valdiklis	Suformuoti adresų sąrašą, atidaryti naujo adreso sukūrimo, koregavimo, paieškos langus, išsaugoti, šalinti adresą.
Užrašų_valdiklis	Suformuoti užrašų sąrašą, atidaryti naujo užrašo sukūrimo, koregavimo, paieškos langus, išsaugoti, šalinti užrašą.

4.2.4. Duomenų paslaugos

Duomenų saugojimui ir nuskaitymui yra naudojamos specialios PHP funkcijos (apdorojimui naudojamas skriptų kalbų interpretatorius Mod_PHP), kurios sukuria ryšį su DBVS, parenka naudojamą duomenų bazę ir siunčia SQL užklausas. Užklausos rezultatai grąžinami PHP programai, kuri apdoroja rezultatus, sugeneruoja HTML kodą, atspausdina ekrane ir uždaro ryšį su DBVS.



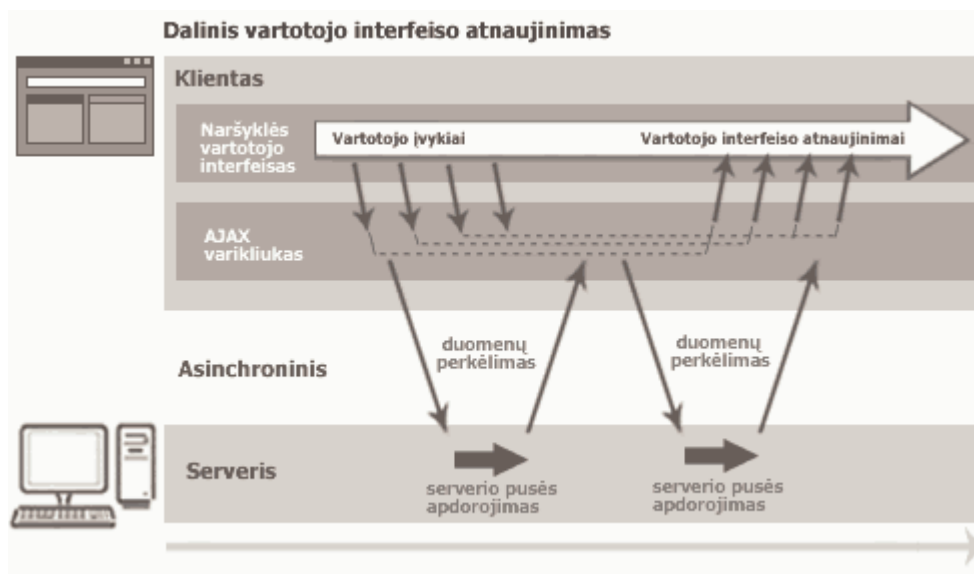
4.4 pav. Duomenų paslaugų modelis

4.2.5. AJAX varikliuko paslaugos

Kliento pusės varikliukas yra AJAX modelio pagrindas. Be šio varikliuko, kiekvienas vartotojo įvykis turi grįžti atgal į serverį apdorojimui. Vartotojo interakcija yra tvirtai surišta su serverio komunikacijomis - kliento pusės varikliukas panaikina šitą priklausomybę. Šis varikliukas, dirbdamas Web naršyklėje, suteikia naršyklei papildomų sugebėjimų, kad atlikti „dalinį ekrano atnaujinimą“ vietoje „pilno puslapio atnaujinimo“ bei vykdyti asinchroninę komunikaciją su serveriu.

“Dalinis ekrano atnaujinimas” - vykstant vartotojo interakcijoms AJAX pagrįstose taikomiosiose programose, atnaujinami tik naują informaciją turintys vartotojo interfeiso elementai. Likę interfeiso elementai išlieka nepakitę. Šis “dalinio ekrano atnaujinimo” sąveikos modelis ne tik įgalina nenutrūkstamo veikimo kontekstą, bet taip pat padaro įmanoma netiesinę darbų seką.

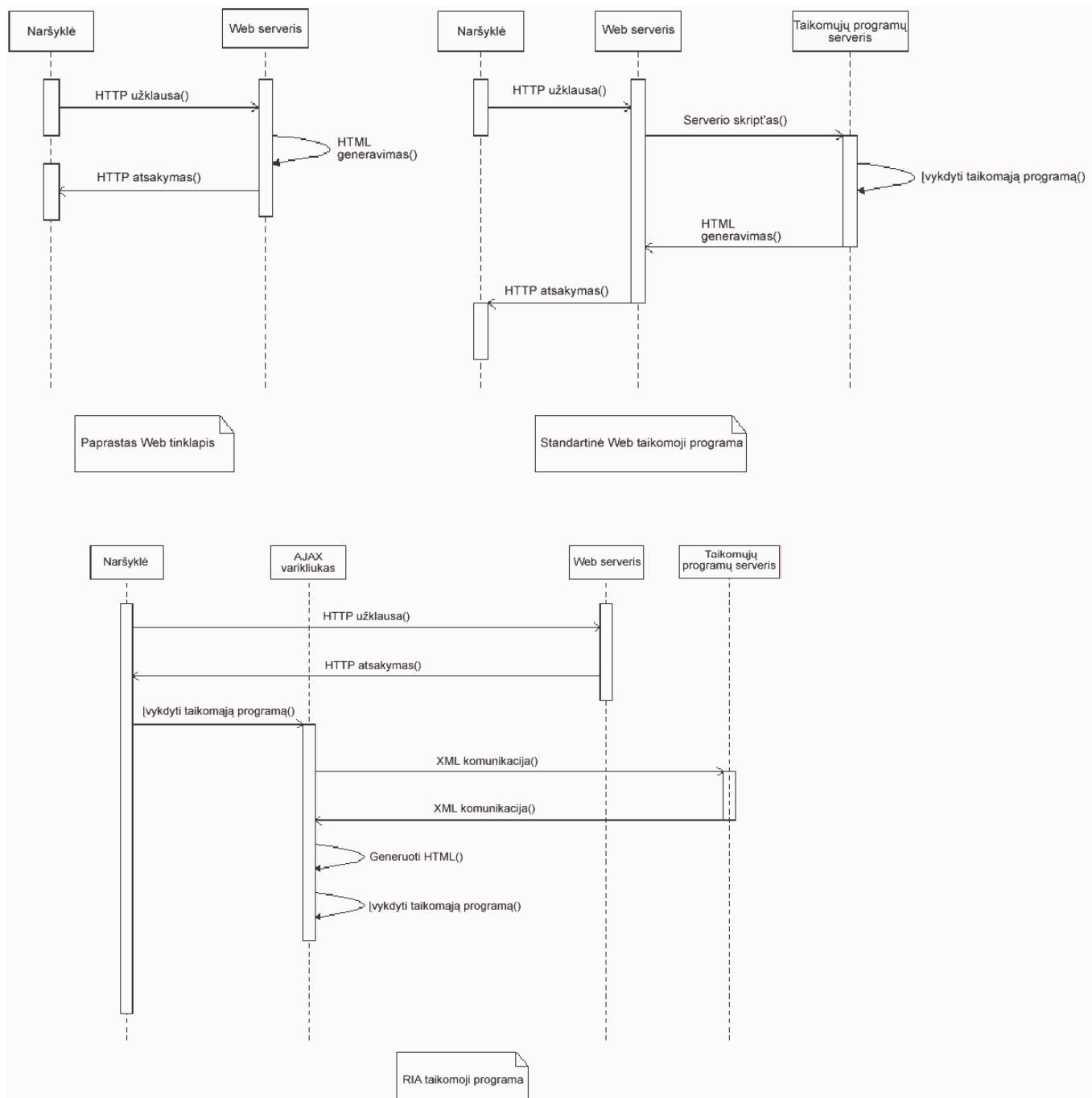
Asinchroninė komunikacija - AJAX pagrįstose taikomiosiose programose “užklausa/atsakymas” gali būti asinchroninis, atskiriant vartotojo ir serverio interakcijas. To pasekoje, vartotojas gali toliau naudoti taikomąją programą be pertraukimų, kol kliento programa užklausia informacijos iš serverio. Kai iš serverio gaunama nauja informacija, atnaujinamos tik su šia informacija susijusios vartotojo interfeiso dalys.



4.5 pav. Ajax varikliuko paslaugos

4.3. Sistemos elgsenos modelis

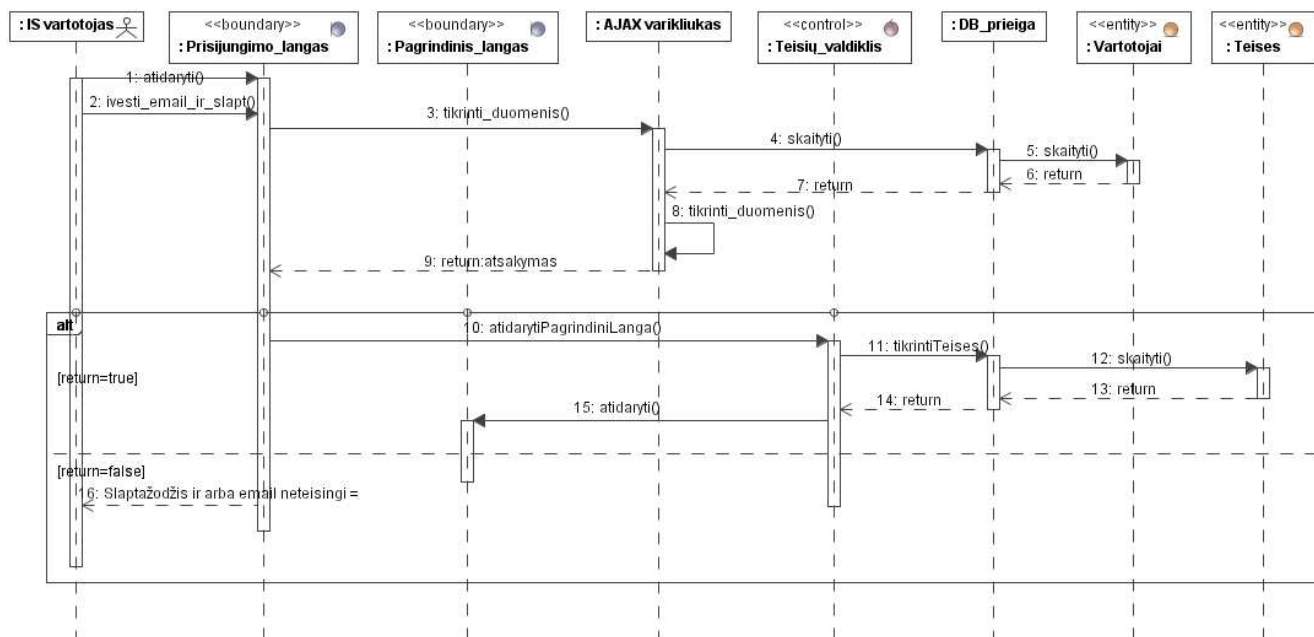
Pradžioje palyginimui pateikiamos (4.6 pav) paprasto Web tinklapio, standartinio Web taikomųjų programų ir RIA taikomųjų programų sekų diagramos, parodančios kaip vyksta užklauso/atsakymo dialogas tarp naršyklės ir serverio.



4.6 pav. Paprasto Web tinklapio, standartinio Web taikomųjų programų ir RIA taikomųjų programų dialogas

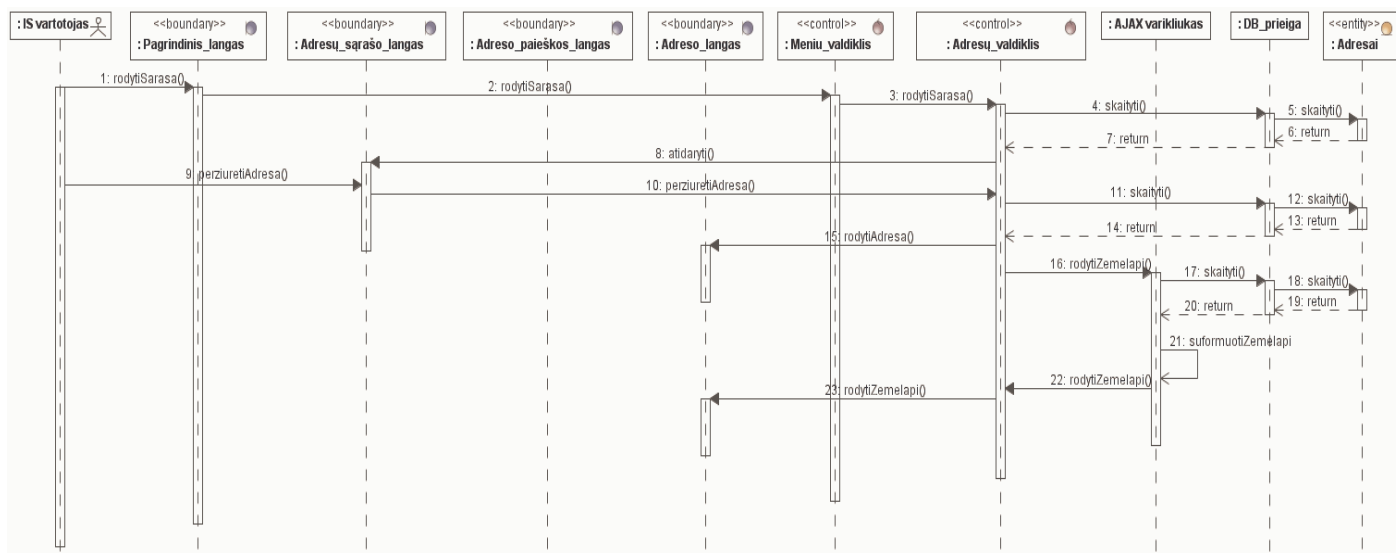
Iš diagramų galima pamatyti, jog RIA taikomųjų programų sekų diagrama skiriasi tuom, kad atsiranda papildomas elementas – AJAX varikliukas, suteikiantis taikomajai programai papildomą funkcionalumą, kurio naršyklė nepalaiko. Antras skirtumas – tai pastebimas asinchroninis komunikavimo procesas tarp naršyklės ir serverio, dėl ko nereikia perkrauti viso naršyklės turinio.

Šiuo principu sudarytos žemiau pavaizduotos svarbiausios ir įdomiausios projekto lygio sekų diagramos. Be to, kadangi kai kurie panaudojimo atvejai yra tarpusavyje labai panašūs pagal atliekamus veiksmus, tai jų sekų diagramos nekartojamos.



4.7 pav. Prisijungimo prie sistemos sekų diagrama

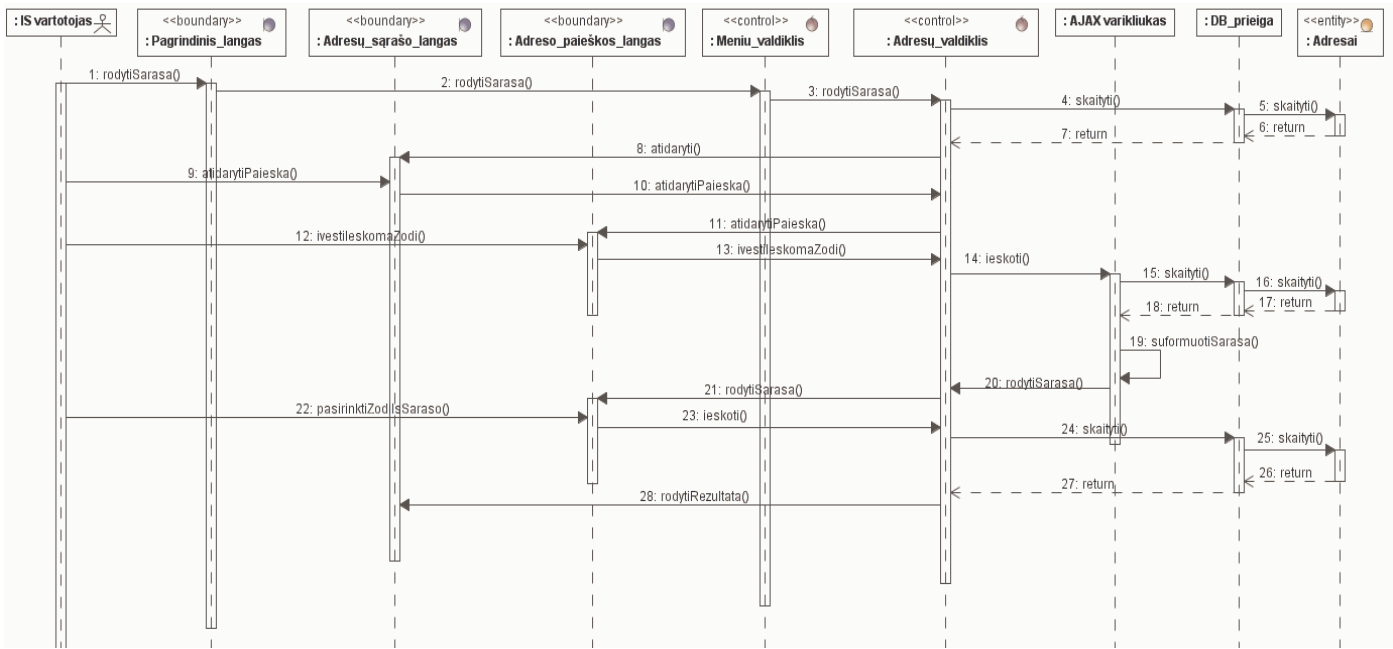
Kiekvieną kartą jungiantis prie sistemos yra patikrinami vartotojo įvesti duomenys. Jei prisijungimo duomenys yra teisingi, atidaromas pagrindinis langas. Jungiantis prie sistemos patikrinamos vartotojo teisės, nuo kurių priklauso, kokiomis sistemos funkcijomis vartotojas gali naudotis. Visi veiksmai atliekami kreipiantis į Ajax varikliuką, kuris asinchroniškai tikrina, ar realiu laiku įvedami prisijungimo duomenys yra teisingi.



4.8 pav. Kontakto peržiūros sekų diagrama

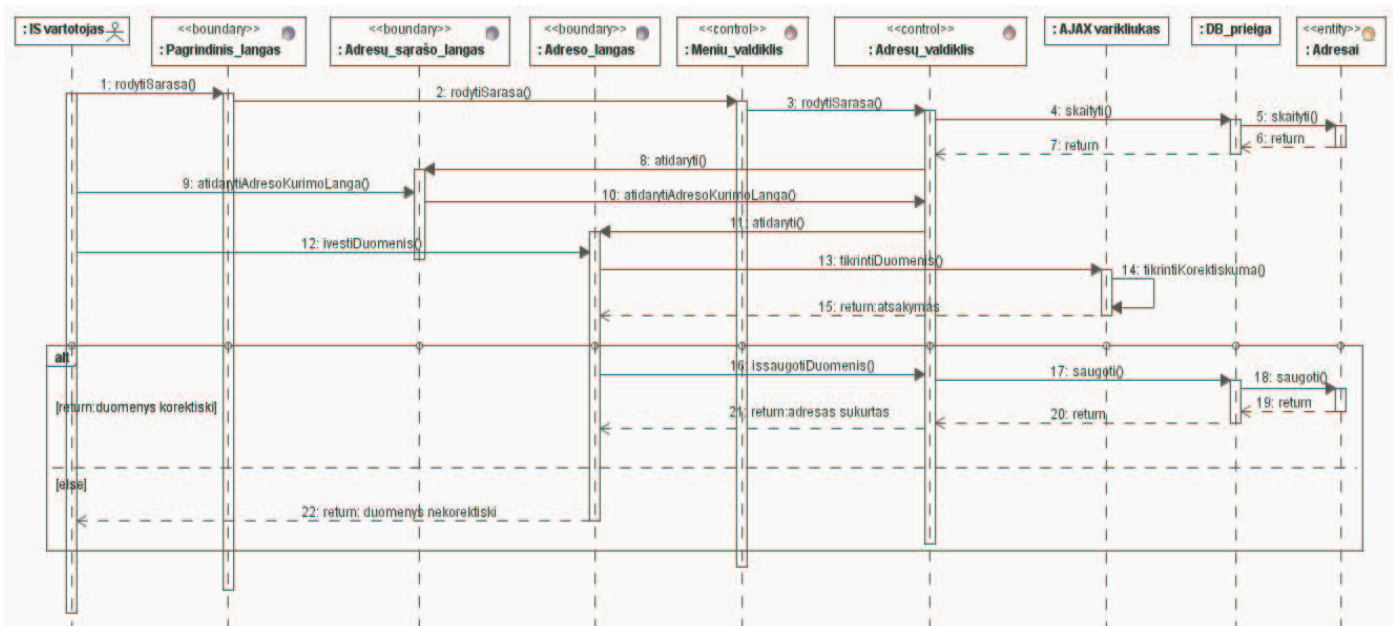
Pasirinkus meniu punktą „Adresų knygelė“ iš duomenų bazės nuskaitymas sąrašas vartotojo susikurtų kontaktų. Peržiūrai pasirenkamas norimas kontaktas. Iš duomenų bazės nuskaitymi pasirinkto kontakto duomenys ir pavaizduojami ekrane. Tuo pačiu metu Ajax varikliukas kreipiasi į duomenų

bazę, nuskaito reikalingus duomenis vietovės nustatymui, suformuoja žemėlapi ir jį atvaizduoja ekrane. Pakeitus kontakto duomenis varikliukas iš naujo suformuoja žemėlapi ir pateikia naujas koordinatas.



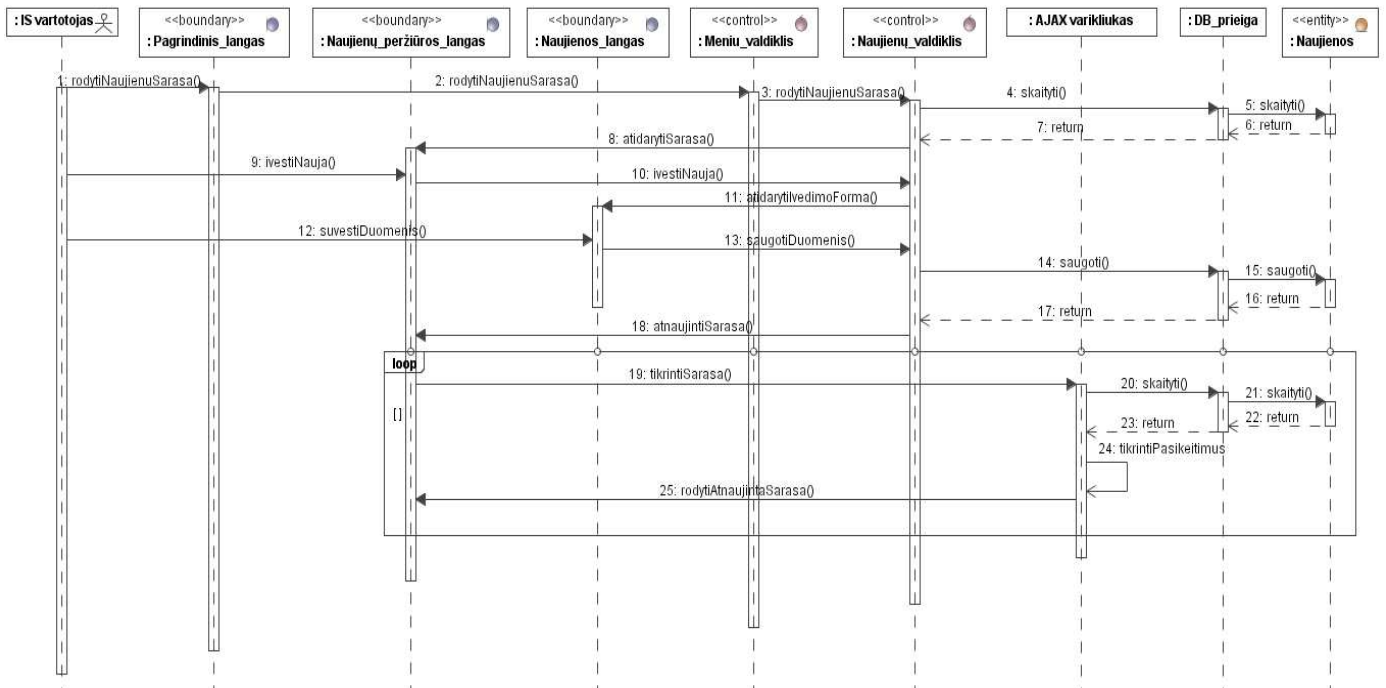
4.9 pav. Paieškos vykdymo sekų diagrama

Pasirinkus meniu punktą „Adresų knygelė“ iš duomenų bazės nuskaitomas sąrašas vartotojo susikurtų kontaktų. Pasirenkamas kontakto paieškos punktas ir atidaroma paieškos forma. Pasirenkamas paieškos kriterijus ir vedamas paieškos žodis. Realiu laiku vedant paieškos žodį Ajax varikliukas kreipiasi į duomenų bazę ir ieško įvedamų simbolių atitikmenų. Jei randami tam tikri įrašai, tai jų sąrašas akimirksniu pateikiamas vartotojo lange. Vartotojui pasirinkus pageidaujamą įrašą ir patvirtinus paiešką išvedamas pageidautas kontaktas. Analogiškai paieška yra vykdoma ir kitose sistemos dalyse.



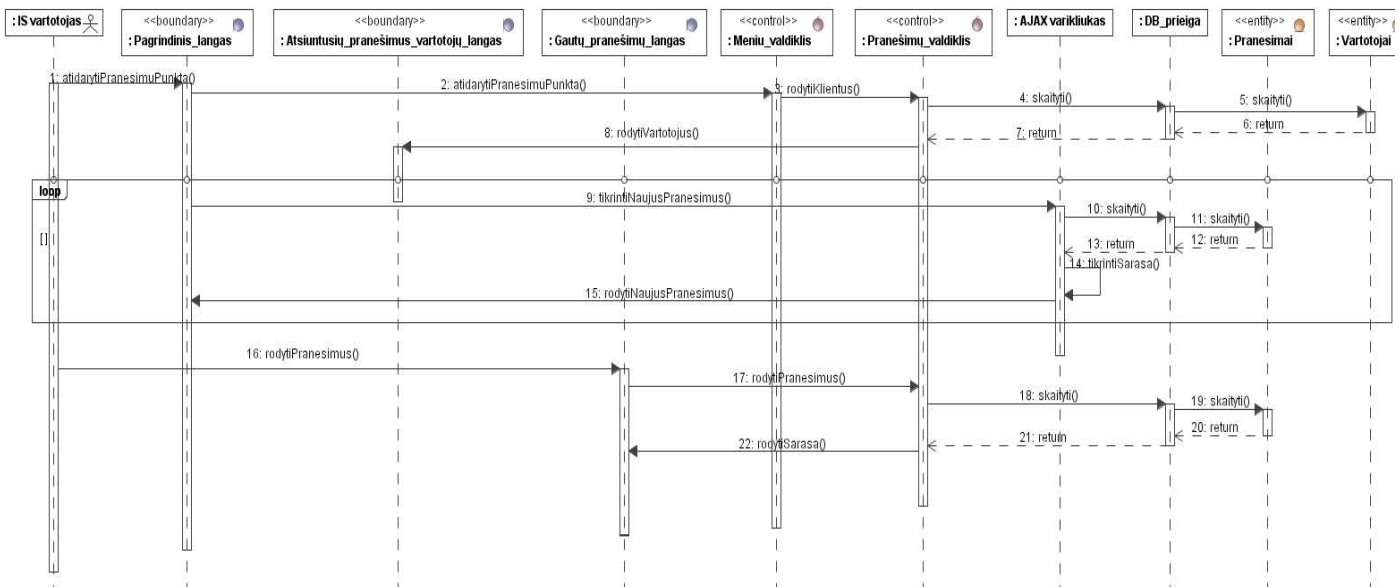
4.10 pav. Įvedamų duomenų validacijos sekų diagrama

Pasirinkus meniu punktą „Adresų knygelė“ iš duomenų bazės nuskaitymas sąrašas vartotojo susikurtų kontaktų. Pasirenkamas naujo kontakto sukūrimo punktas ir atidaroma duomenų įvedimo forma. Įvedami reikalingi duomenys į formos laukus. Realiu laiku vedant duomenis Ajax varikliukas tikrina, ar duomenys yra teisingi. Jei aptinkami klaidingi duomenys, akimirksniu ekrane parodomas klaidos pranešimas, neišsiunčiant formos duomenų. Analogiškai validacija yra vykdoma ir kitose sistemos dalyse.



4.11 pav. Duomenų atnaujinimo realiu laiku sekų diagrama

Pasirinkus meniu punktą „Naujienos“ iš duomenų bazės nuskaitymas sąrašas administratoriaus susikurtų naujienų. Tuo metu Ajax varikliukas cikliška kartas nuo karto kreipiasi į duomenų bazę ir tikrina, ar naujienų sąrašas neįvyko kokių nors pasikeitimų. Jei pasikeičia duomenys (sukuriamas nauja naujiena, pakoreguojama arba pašalinama), realiu laiku kliento lange pasikeičia naujienų sąrašas. Pasirenkamas naujos naujienos sukūrimo punktas ir atidaroma duomenų įvedimo forma. Įvedami reikalingi duomenys ir forma išsiunčiama. Sukurta nauja naujiena akimirksniu pasirodo kliento atidarytame naujienų sąrašas.

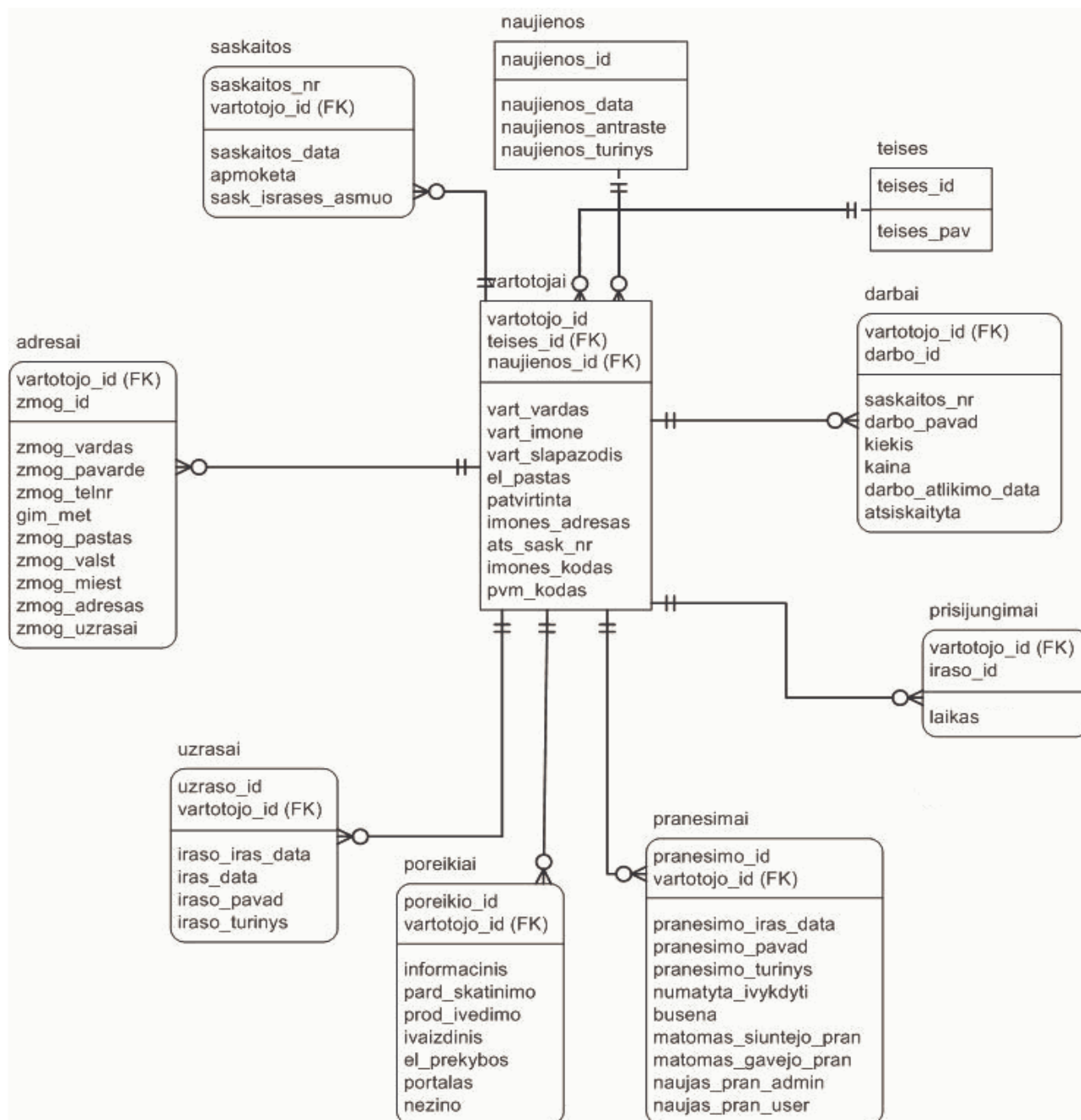


4.12 pav. Pranešimo realiu laiku apie gautas naujas žinutes sekų diagrama

Ajax varikliukas cikliškai kartas nuo karto kreipiasi į duomenų bazę ir tikrina, ar neatsirado naujo pranešimo įrašas. Kai tik parašomas naujas pranešimas, realiu laiku Ajax varikliukas suformuoja naujų pranešimų sąrašą ir sistemos vartotojui ekrane pasirodo pranešimas apie gautas naujas (neperskaitytas) žinutes. Vartotojui perskaičius žinutę Ajax varikliukas iš naujo performuoja sąrašą.

4.4. Duomenų bazės schema

Suprojektuota duomenų bazės schema pavaizduota 4.13 paveikslėlyje. DB loginė schema sudaryta iš dalykinės srities klasių modelio. Kiekviena lentelė, jos laukai bei galimos reikšmės aptartos smulkiau 4.2 lentelėje.



4.13 pav. Duomenų bazės schema

4.2 lentelė. DB lentelių specifikacija

Lentelė: „vartotojai“ - saugomi duomenys apie sistemos vartotojus		
Lauko pav	Paaiškinimas	Duomenų tipas
vartotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>pirminis raktas</i>)	Int (7)
vart_vardas	Vartotojo vardas, sudaro maks. 30 simbolių	Varchar (30)
vart_imone	Vartotojo įmonės pavadinimas, sudaro maks. 30 simbolių	Varchar (30)
vart_slaptazodis	Vartotojo slaptažodis, sudaro maks. 15 simbolių	Varchar (15)
el_pastas	Vartotojo elektroninis paštas, naudojamas kaip	Varchar (40)

	prisijungimo vardas, sudaro maks. 40 simbolių	
patvirtinta	Pažymi, ar vartotojas yra patvirtintas, jei taip reikšmė – 1, jei ne – 0	Tinyint(1)
imones_adresas	Vartotojo įmonės adresas	Text
ats_sask_nr	Atsakomosios sąskaitos numeris, maks. 20 simbolių; pirmos dvi raidės „LT“, likę – skaičiai.	Varchar (20)
imones_kodas	Įmonės kodas, maks. 9 simboliai, galimi tik skaičiai.	Int (9)
pvm_kodas	Įmonės PVM kodas, maks. 11 simbolių, pirmos dvi raidės „LT“, likę – skaičiai.	Varchar (11)
teises_id	Vartotojo teisės ID (<i>išorinis raktas</i>)	Int (1)
naujienos_id	Naujienos ID (<i>išorinis raktas</i>)	Int (10)
Lentelė: „pranesimai“- saugomi vartotojų siunčiami pranešimai		
pranesimo_id	Pranešimo ID, maks. 10 simbolių (<i>pirminis raktas</i>)	Int (10)
vartotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
pranesimo_iras_data	Pranešimo sukūrimo data	Datetime
pranesimo_pav_ad	Pranešimo antraštė, maks. 50 simbolių	Varchar(50)
pranesimo_turiny	Pranešimo turinys	Text
numatyta_ivykydyti	Kada numatyta įvykdyti užduotį, maks. 50 simbolių	Varchar(50)
busena	Užduoties įvykdymo būseną, galimos reikšmės: „Įvykdyta“ arba „Neįvykdyta“	Varchar(16)
matomas_siuntejo_pranesimas	Jei „1“, tai administratorius mato pranešimą, jei „0“ - nemato	Int (1)
matomas_gavejo_pranesimas	Jei „1“, tai klientas mato pranešimą, jei „0“ - nemato	Int (1)
naujas_pran_admin	Jei „1“, reiškia naują administratoriaus pranešimą, jei „0“ - perskaitytas pranešimas	Int (1)
naujas_pran_user	Jei „1“, reiškia naują kliento pranešimą, jei „0“ - perskaitytas pranešimas	Int (1)
Lentelė „adresai“- saugoma informacija apie kontaktus		
zmog_id	Kontakto ID, sudaro 7 skaitmenys(<i>pirminis raktas</i>)	Int (7)
vartotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
zmog_vard	Kontakto vardas, maks. 15 simbolių	Varchar(15)
zmog_pavard	Kontakto pavardė, maks. 20 simbolių	Varchar(20)
zmog_telnr	Kontakto telefono numeris, sudaro 11 skaitmenų	Int (11)
gim_met	Kontakto gimimo metai	Date
zmog_pastas	Kontakto elektroninis paštas, maks. 30 simbolių	Varchar(30)
zmog_valst	Kontakto valstybės pavadinimas, maks. 20 simbolių	Varchar(20)
zmog_miest	Kontakto miesto pavadinimas, maks. 20 simbolių	Varchar(20)
zmog_adresas	Kontakto adresas, maks. 30 simbolių	Varchar(30)
zmog_uzrasai	Trumpa pastaba apie kontaktą, maks. 255 simbolių	Varchar(255)
Lentelė „darbai“- saugomi klientam atlikti darbai		
darbo_id	Darbo ID, sudaro 7 skaitmenys(<i>pirminis raktas</i>)	Int (7)
vartotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
saskaitos_nr	Saskaitos, kuriai priskirtas darbas numeris, maks. 10 skaitmenų	Int (10)
darbo_pavard	Atlikto darbo pavadinimas	Text
kiekis	Atliktų darbų kiekis valandomis arba vienetais, maks. 10 skaitmenų	Int (10)

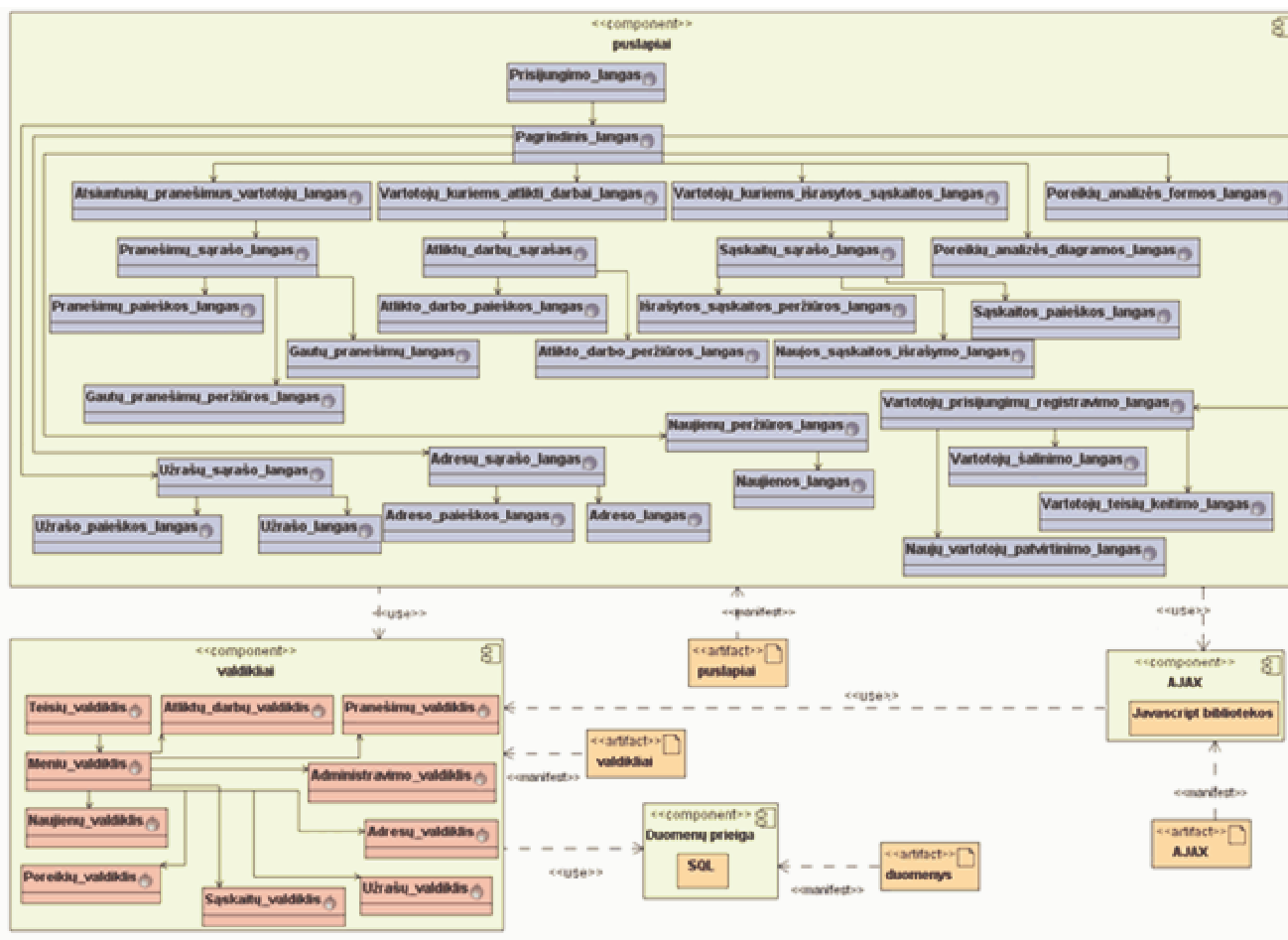
kaina	Vieneto kaina, maks. 10 skaitmenų	Int (10)
darbo_atlikimo_data	Darbo užregistravimo data.	Date
atsiskaityta	Pažymi, ar už šį darbą išrašyta sąskaita, jei taip reikšmė – 1, jei ne – 0	Tinyint(1)
Lentelė „naujienos“- saugomos klientam skelbiamos naujienos		
naujienos_id	Naujienos ID, sudaro 10 skaitmenų (<i>pirminis raktas</i>)	Int (10)
naujienos_data	Naujienos paskelbimo data	Date
naujienos_antrašte	Naujienos antraštė	Text
naujienos_turinys	Naujienos turinys	Text
Lentelė „poreikiai“- saugoma informacija apie klientų pasirenkamą tinklapio tipą		
poreikio_id	Poreikio ID, sudaro 10 skaitmenų (<i>pirminis raktas</i>)	Int (10)
virtotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
informacinis	Pasirinktas tiklapio tipas - informacinis	Char(1)
pard_skatinimo	Pasirinktas tiklapio tipas – pardavimų skatinimo	Char(1)
prod_ivedimo	Pasirinktas tiklapio tipas – produkto įvedimo į rinką	Char(1)
ivaizdinis	Pasirinktas tiklapio tipas - įvaizdinis	Char(1)
el_prekybos	Pasirinktas tiklapio tipas – elektroninės prekybos	Char(1)
portalas	Pasirinktas tiklapio tipas - portalas	Char(1)
nezino	Pasirinktas tiklapio tipas - nezino	Char(1)
Lentelė „prisijungimai“- saugoma informacija apie klientų prisijungimus prie IS		
iraso_id	Prisijungimo ID, sudaro 10 skaitmenų (<i>pirminis raktas</i>)	Int (10)
virtotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
laikas	Prisijungimo laikas	Datetime
Lentelė „saskaitos“- saugoma informacija apie klientam išrašytas sąskaitas		
saskaitos_nr	Sąskaitos numeris sudaro 10 skaitmenų (<i>pirminis raktas</i>)	Int (10)
virtotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
saskaitos_data	Sąskaitos išrašymo data	Date
apmoketa	Pažymi, ar apmokėta sąskaita, jei „1“ - taip, „0“ - ne	Tinyint(1)
sask_israses_asmuo	Sąskaitą išrašiusio asmens vardas, pavardė ir pareigos	Text
Lentelė „teises“- vartotojų teisės		
teises_id	Vartotojo teisės ID (<i>pirminis raktas</i>)	Int (1)
teises_pav	Teisės pavadinimas	Varchar(5)
Lentelė „uzrasai“- saugomi vartotų užrašai		
uzraso_id	Užrašo ID, sudaro 10 skaitmenų (<i>pirminis raktas</i>)	Int (10)
virtotojo_id	Vartotojo ID, sudaro 7 skaitmenys(<i>išorinis raktas</i>)	Int (7)
iraso_iras_data	Užrašo įrašymo data	Date
iras_data	Įvykio, apie kurį gaunamas priminimas data	Date
iraso_pavad	Užrašo antraštė, maks. 50 simbolių	Varchar(50)
iraso_turinys	Užrašo turinys	Text

5. Eksperimentinė realizacija

Eksperimentinės realizacijos dalyje nubraižyta sistemos komponentų diagrama, sudarytas sistemos įdiegimo modelis, apibrėžti reikalavimai sistemos funkcionalumo palaikymui, aprašytas sistemos veikimas bei atliktas testavimas.

5.1. Sistemos komponentai

Komponentų diagramoje (5.1 pav.) CRM informacinė sistema padalinama į fizinius komponentus bei nurodoma sąveika tarp jų. Kiekvieno komponento realizacijos rezultatas yra artefaktas, kuris bus įdiegiamas fiziniuose įrenginiuose. Kadangi sistema realizuojama *PHP* programavimo kalba, *HTML* puslapiai sudaro vieną komponentą, kurį realizuojame artefaktu *puslapiai*. O veiklos logika realizuojama *PHP* klasėmis, kurios sudaro artefaktą *valdikliai*. Dinamiškos sistemos komunikavimas vyksta netiesiogiai su *PHP* klasėmis, o per *AJAX* komponentą. Kai gaunama užklausa iš naršyklės, ji per *HTML* puslapius (*puslapiai*) pereina į *AJAX* komponentą, o ne į *valdikliai*. Taip realizuojama *AJAX* varikliuko funkcija. Nedinamiškos funkcijos vykdomos kreipiantis tiesiogiai į *valdikliai*.



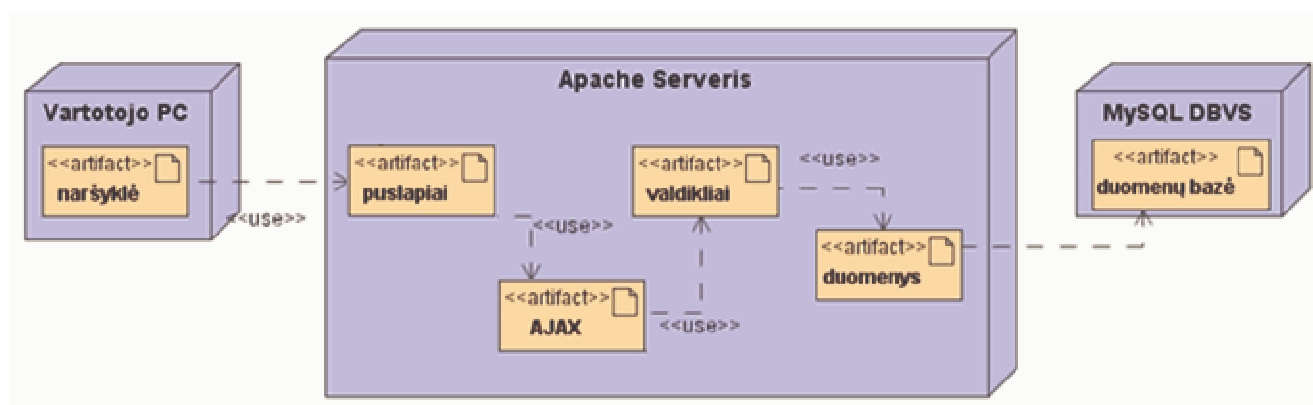
5.1 pav. Sistemos komponentų diagrama

5.2. Sistemos įdiegimo modelis

CRM sistemos įdiegimo diagrama (5.2 pav.) rodo fizinį techninių įrenginių išdėstymą bei vykdomųjų programų komponentų paskirstymą juose.

Informacinės sistemos įdiegimo diagrama susideda iš trijų dalių: vartotojo personalinio kompiuterio su interneto naršykle, *Apache* serverio, kuriame vykdomas vartotojo sąsajos generavimas bei veiklos logika, ir *MySQL* duomenų bazės, kurioje saugomi informacinės sistemos duomenys.

Vartotojas naudodamasis personaliniu kompiuteriu ir jame įdiegta programine įranga (Interneto naršykle) prisijungia prie informacinės sistemos, kuri yra operuojama *Apache* serverio pagalba, atlieka nustatytus veiksmus, o sistema išsaugo pakeitimus *MySQL* duomenų bazėje.



5.2 pav. Sistemos įdiegimo diagrama

5.3. Reikalavimai sistemos funkcionavimo palaikymui

Sistemos palaikymui serverio pusėje turi būti įdiegtas Apache Web serveris. Serverio platforma (operacinė sistema) gali būti bet kokia, palaikanti Apache Web serverį. Web serveris turi būti su PHP serverio programavimo kalbos ir MySQL duomenų bazių valdymo sistemos palaikymu. Visi IS sisteminiai failai įkeliami į serverio darbinę stotį. Serverio techninės įrangos reikalavimai yra tokie kaip paminėtų technologijų funkcionavimo minimalūs techniniai reikalavimai. Siekiant maksimalaus sistemos efektyvumo, serveris turi veikti 24 valandas per parą, 7 dienas per savaitę.

Kliento pusėje turi būti įdiegta interneto naršyklė, pageidautina Internet Explorer 5 ar vėlesnė versija. Interneto naršyklėje turi būti įjungtas Javascript palaikymas. Kliento techninės įrangos reikalavimai sutampa su interneto naršyklės minimaliais techniniais reikalavimais. Kliento pusėje jokie sistemos failai nėra įkeliami.

Web serveris, kuriame veikia sistema, gali būti įdiegtas ir kliento kompiuteryje, tačiau siekiant maksimalaus sistemos efektyvumo, toks kompiuteris privalo būti pasiekiamas iš interneto 24 valandas per parą, 7 dienas per savaitę.

Serveris, kuriame diegiama sistema, turi palaikyti „cronjobs“ funkciją, tam kad būtų galima planuoti failo, reikalingo priminimų funkcijai įgyvendinti, vykdymą nustytu laiku.

5.4. Sistemos veikimo aprašymas

Pagrindiniai darbo principai

Prieš pradėdant darbą sistemoje, reikia prie jos prisiregistruoti, t.y. suvesti registravimosi duomenis. Suvedus korektišką informaciją, registracijos forma išsiunčiama sistemos administratoriui, kuris el. paštu gauna pranešimą, jog tam tikras vartotojas prisiregistravo sistemoje ir laukia patvirtinimo. Prisijungti prie sistemos galima tik kai administratorius patvirtina registraciją.

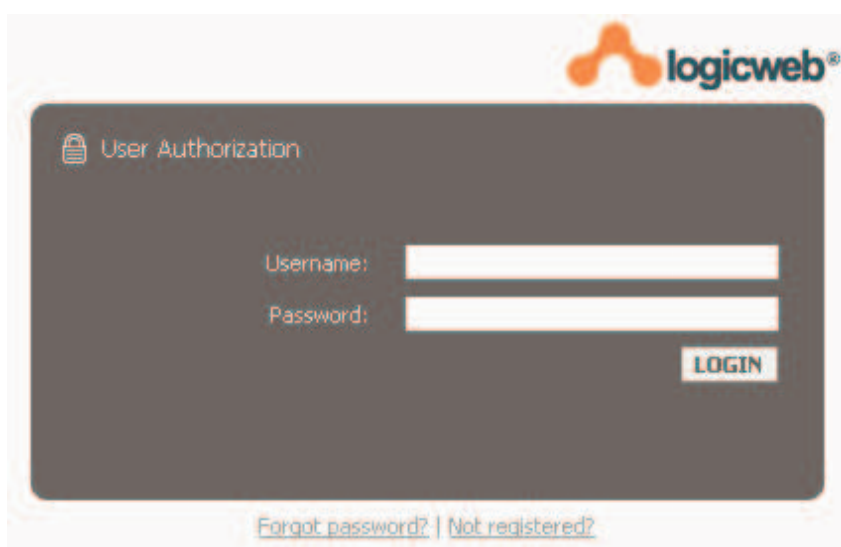
Prisijungus prie sistemos iš karto matomas naujienų langas. Visas sistemos meniu išdėstytas vertikaliai kairėje ekrano pusėje. Meniu punktai yra aiškiai suformuluoti pagal atitinkamas sistemos teikiamas paslaugas, todėl žinant, kokia funkcija norima pasinaudoti, tiesiog pasirenkamas atitinkamas meniu punktas. Detalesnis navigavimas aprašomas žemiau atskiriems vartotojų tipams.

Yra įgyvendinta pakankamai griežta apsauga nuo nekorektiškų duomenų įvedimo, todėl netyčia suvedus netinkamus duomenis ekrane išvedamas perspėjantis pranešimas. Tuo atveju, jeigu užmirštamasis prisijungimo prie sistemos slaptažodis, yra slaptažodžio priminimo funkcija. Tereikia įvesti savo el. pašto adresą, kuriuo registravotės sistemoje ir prisijungimo duomenys atsiunčiami į el. pašto dėžutę.

Baigiant darbą sistemoje būtina paspausti mygtuką „Atsijungti“ tam, kad panaikinti prisijungimo sesiją.

Administratoriaus atliekamų funkcijų aprašymas

Darbo pradžioje reikia registruotis sistemoje. Naudojantis interneto naršykle reikia įvesti adresą, kuriuo pasiekama informacinė sistema. Sistema yra patalpinta šiuo adresu: <http://www.deform-group.com/mag/>. Naršyklės lange matomas vaizdas pavaizduotas (5.3 pav):



5.3 pav. Prisijungimo prie sistemos langas

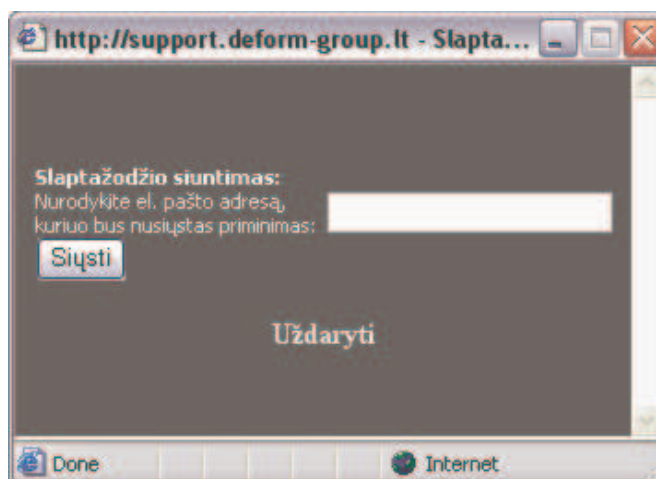
Pasirenkame nuorodą „Not registered?“, atsidaro langas (5.4 pav):



5.4 pav. Registracijos sistemoje langas

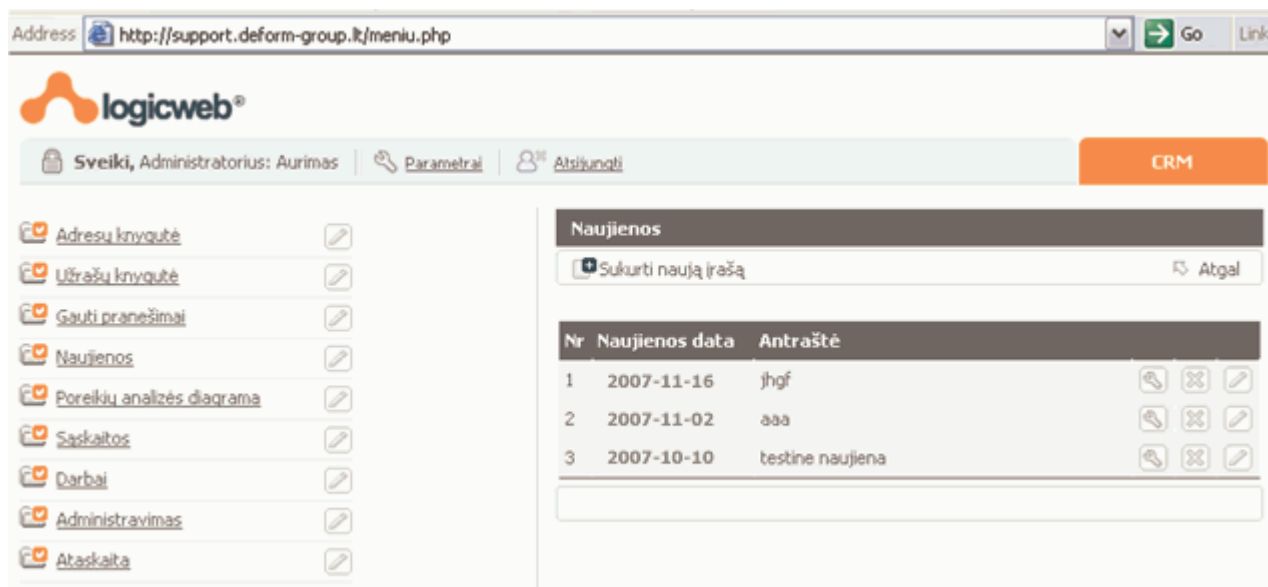
Į atsidariusią formą reikia įvesti reikalingus duomenis ir spausti mygtuką „Registruotis“. Sėkmingai išsiuntę registracijos duomenis gausite el. pašto pranešimą, kuriame rašoma, jog Jūs sėkmingai prisiregistravote sistemoje ir turite laukti administratoriaus patvirtinimo. Kai jau anksčiau registruotas administratorius patvirtina Jus sistemoje, gaunate el. pašto pranešimą apie patvirtinimą ir galite jungtis prie sistemos.

Jei pamiršote slaptažodį, galima pasinaudoti slaptažodžio priminimo funkcija, tam spaudžiama nuoroda „Forgot password?“. Atsidariusiame lange (5.5 pav.) įvedamas el. paštas, kuriuo Jūs registravotės sistemoje ir spaudžiame „Siųsti“. Netrukus gausite į nurodytą el. pašto adresą pranešimą su Jūsų prisijungimo duomenimis.



5.5 pav. Slaptažodžio priminimo langas

Norint prisijungti prie sistemos, į prisijungimo formą (5.3 pav.) reikia įvesti prisijungimo vardą (tai Jūsų el. paštas) ir slaptažodį, paspausti mygtuką „LOGIN“. Prie sistemos prisijungiamo administratoriaus teisėmis. Prisijungus matomas sistemos administratoriui teikiamų paslaugų langas (5.6 pav):



5.6 pav. Sistemos paslaugų meniu prisijungus administratoriaus teisėmis

Lango viršutinėje dalyje, po sistemos logotipu, matome prisijungusio vartotojo informaciją, parametų keitimo ir atsijungimo nuo sistemos meniu nuorodą. Po ja, kairėje lango pusėje vertikaliajose pozicijoje yra pagrindinis valdymo (paslaugų) meniu. Trumpai aptarsime kiekvieną paslaugą:


Adresų knygtė – tai abiemis vartotojų tipams galiojanti paslauga. Čia galima sukurti naują kontaktą, peržiūrėti, redaguoti bei šalinti esamą, atlikti paiešką, eksportuoti visą adresų knygelę. Adresų knygtės pagrindinis langas pavaizduotas 5.7 paveikslėlyje:





5.7 pav. Adresų knygelės pagrindinis langas

Paspaudus ant lentelės antraštės pavadinimo sąrašas atitinkamai surikiuojamas pagal tą laiką. Paspaudus punktą „Sukurti naują įrašą“ atidaromas naujo kontakto sukūrimo langas pavaizduotas 5.8 pav. Į formą įvedame reikiamus duomenis ir spaudžiame „Saugoti“.

5.8 pav. Naujo kontakto sukūrimo langas

Paspaudus punktą  atidaromas kontakto redagavimo langas, kuris yra analogiškas kontakto sukūrimo langui (5.8 pav.). Pakoreguojame formos duomenis ir spaudžiame „Saugoti“.

Paspaudus  mygtuką yra pašalinamas kontaktas. Paspaudus mygtuką  atidaromas kontakto peržiūros langas pavaizduotas 5.9 pav.

5.9 pav. Kontakto peržiūros langas

Čia taip pat galima pasirinkti įrašo šalinimo ir redagavimo funkcijas.

Norėdami ieškoti norimo kontakto, pasirenkame nuorodą „Paieška“. Atidaromas langas pavaizduotas 5.10 pav. Pasirenkame pagal kokį kriterijų norime ieškoti, įvedame paieškos žodį ir

spaudžiame „Ieškoti“. Į ekraną išvedamas rastų kontaktų sąrašas, su kuriuo galima atlikti aukščiau aprašytas operacijas.

PAIEŠKA

Duomenų formatai:
Vardai, pavardės, valstybės, miestai turi būti sudaryti tik iš raidžių;
Telefono numeris tik iš skaičių.
Gimimo metų pvz.: 1984, 1984-01, 1984-01-01
Adresų pvz.: Balt, Baltijos, Baltijos 14, Baltijos 14-5

Pasirinkite pagal ką ieškosite: Pavarde

Įveskite paieškos žodį:

Ieškoti

5.10 pav. Kontakto paieškos langas

Norėdami išeksportuoti duomenis, pasirenkame punktą „Duomenų eksportas“. Atsidaro sisteminis langas, kuris klausia, kur norime išsaugoti duomenis, nurodome saugojimo vieta ir spaudžiame „Save“ mygtuką.

Užrašų knygutė – tai abiemis vartotojų tipams galiojanti paslauga. Čia galima sukurti naują užrašą, peržiūrėti, redaguoti bei šalinti esamą, atlikti paiešką. Šios funkcijos veikimo principas yra analogiškas „Adresų knygelės“, todėl smulkiau galimi veiksmai neaprašinėjami.

Atsakymas į klientų pranešimus – čia galimos tokios funkcijos: klientui siunčiamas atsakas apie gautos užduoties įvykdymo laitarpį ir būseną, pranešimo šalinimas, paieška, peržiūra.

Pasirinkus meniu punktą „Gauti pranešimai“ išvedamas klientų, atsiuntusių bent vieną pranešimą, sąrašas, kuris pavaizduotas 5.11 pav. Jei yra naujų pranešimų, išvedamas pranešimas ir vartotojas paryškinamas raudona spalva.


Pasirinkite klientą:

Atgal

Viso neperskaityta pranešimų: 1

Nr	Vardas	Imonė	El. paštas
1 (1)	aurimas	ACATSEON	aurimas@acatseon.lt
2	petras	OHO	vadaz@one.lt
3	Petras	UAB "Kalbiai"	aurimas.juncis@stud.ktu.lt

5.11 pav. Klientų sąrašo langas

Norint pasirinkti klientą, spaudžiama ikona  ir išvedamas pasirinkto kliento atsiųstų pranešimų langas, kuris pavaizduotas 5.12 pav. Nauji pranešimai pažymėti raudonai.

Kliento atsiųsti pranešimai		
Paieška		Atgal
Nr	Būsena	Antraštė
1	Įvykdyta	RE: nauja info
2	Įvykdyta	RE: test

5.12 pav. Kliento atsiųstų pranešimų sąrašas


Norint atsakyti klientui pasirenkama ikona  ir atidaroma atsakymo į pranešimą forma, pavaizduota 5.13 pav.

RE: nauja info



PLAČIAU ATSAKYTI ŠALINTI

Numatyta įvykdyti: 2 val bėgyje

Pastabos: viskas gerai

 Siųsti

5.13 pav. Atsakymo į pranešimą forma

Įvedamas numatomas užduoties įvykdymo terminas bei pastabos (jei reikia) ir spaudžiamas mygtukas „Siųsti“. Norint tik peržiūrėti gautą pranešimą (5.14 pav.) arba jį pašalinti, atitinkamai pasirenkamas punktas „Plačiau“ ir „Šalinti“ arba sąrašo lange (5.12 pav) ikonos  ir  .

Norint pakeisti užduoties įvykdymo būseną atidaromas peržiūros langas (5.14 pav.) ir paspaudžiama ant būsenos pavadinimo, kuris po paspaudimo atitinkamai pasikeičia į „Įvykdyta“ arba „Neįvykdyta“.

RE: test

PLAČIAU ATSAKYTI ŠALINTI

Būsena: **Įvykdyta**

Pranešimo įrašymo data: 2007-11-16 10:23:54

Pranešimo antraštė: RE: test

Numatyta įvykdyti: siandien

Pranešimo turinys: none

REPLAYED TO CLIENT:
Užrašai vedami čia

5.14 pav. Pranešimo peržiūros ir būsenos keitimo langas

Norėdami ieškoti norimo pranešimo, sąrašo lange pasirenkame nuorodą „Paieška“. Atidaromas analogiškas kontaktų paieškai langas, kuris pavaizduotas 5.10 pav. skiriasi tik paieškos kriterijai. Pasirenkame pagal kokį kriterijų norime ieškoti, įvedame paieškos žodį ir spaudžiame „Ieškoti“. Į ekraną išvedamas rastų pranešimų sąrašas, su kuriuo galima atlikti aukščiau aprašytas operacijas.

Naujienų skelbimas – čia galimos tokios funkcijos: naujienos peržiūra, naujos naujienos kūrimas, esamos redagavimas ir šalinimas. Kadangi navigacija tokia pati kaip ir „Adresų knygelės“, tai nuosekli veiksmų seka nebekartojama, o sistemos langų paveikslėliai neberodomi (juos galima pasižiūrėti aukščiau).

Pasirinkus meniu punktą „Naujienos“ analogiškai kaip ir „Adresų knygelėje“ matomas jau paskelbtų naujienų sąrašas, kurio pasirinktą elementą galima redaguoti, šalinti ar tiesiog peržiūrėti. Sukuriant naują naujieną pasirenkamas punktas „Sukurti naują įrašą“, įvedama naujienos antraštė, turinys ir spaudžiamas mygtukas „Skelbti“.

Kliento tinklapio tipo pasirinkimo analizės diagrama – paprasčiausiai pasirenkamas meniu punktas „Poreikių analizės diagrama“ ir ekrane išveda klientų pasirenkamo tinklapio tipo analizės diagrama grafiniame pavidale.

Sąskaitos – čia galimos tokios funkcijos: sąskaitos peržiūra, naujos sąskaitos išrašymas, apmokėtos šalinimas ir paieška. Pasirinkus meniu punktą „Sąskaitos“ analogiškai kaip ir punkte „Gauti pranešimai“ matomas klientų, kuriems yra išrašytos sąskaitos, langas (5.11 pav.). Pasirenkamas pageidaujamas klientas ir atidaromas konkrečiam klientui išrašytų sąskaitų sąrašo langas. Norint peržiūrėti sąskaitą pasirenkama atitinkama, jau anksčiau aprašyta, ikona ir ekrane išvedamas sąskaitos turinys (5.16 pav.). Norėdami ieškoti norimos sąskaitos, jų sąrašo lange pasirenkame nuorodą „Paieška“. Atidaromas analogiškas kontaktų paieškai langas, kuris pavaizduotas 5.10 pav. skiriasi tik paieškos kriterijai. Norint išrašyti naują sąskaitą pasirenkamas klientų sąrašo esantis punktas „Išrašyti naują sąskaitą“. Atidaroma nauja forma (5.15 pav.).

5.15 pav. Sąskaitos išrašymo forma

PVM SAŠKAITA FAKTŪRA
Nr. 19

Sąskaitos išrašymo data: 2007-11-17

Paslaugos tiekėjas: UAB "Reklamos vizija"
Adresas: Lietuvių g. 1, 44169 Kaunas
Ats. sąskaitos nr.: LT123456789012345678
Bankas: AB bankas "Bankas"
Banko kodas: 12345
Įmonės kodas: 123456789
PVM mok. kodas: LT123456789

Paslaugos užsakovas: UAB "Kalbiai"
Adresas: mano gatve 4
Ats. sąskaitos nr.: LT123456789023456785
Įmonės kodas: 7171717
PVM mok. kodas: LT254367890

Nr.	Pavadinimas	Kiekis	Kaina	Suma
1	Už nuotraukų sukėlimą	20	5	100
2	Naujos informacijos sukėlimas	2	60	120

Iš viso:	220
PVM(18%):	39.6
Bendra suma:	259.6

Sąskaitą išrašė: programuotojas Aurimas Junčis

5.16 pav. Išrašyta sąskaita

Jei yra neapmokėtų sąskaitų, tai jos sąraše būna pažymėtos raudonai ir jei norime patvirtinti sąskaitos apmokėjimą, tiesiog paspaudžiame ant atitinkamos sąrašo eilutės ir būseną „Ar apmokėta“ pasikeičia iš „Ne“ į „Taip“. Pašalinti galima tik apmokėtas sąskaitas.

Darbu registravimas – čia galimos tokios funkcijos: registruoti naują darbą, peržiūrėti, redaguoti bei šalinti esamą, atlikti paiešką. Taigi šios funkcijos yra analogiškos „Adresų knygelės“, todėl smulkiau galimi veiksmai neaprašinėjami. Skiriasi tik tai, jog prieš administruojant atliktus darbus reikia pasirinkti iš sąrašo klientą (tai pat kaip ir dirbant su sąskaitom). Tolimesni veiksmai yra identiški tik skiriasi laukų reikšmės ir įvedama ar koreguojama informacija. Išskirtina tik tai, jog darbų, kuriems jau išrašytos sąskaitos redaguoti nebegalima, o pašalinti galima tik pašalinus sąskaitą.

Klientų administravimas – čia galimos tokios funkcijos: patvirtinti naują vartotoją, pašalinti esamą, keisti teises ir peržiūrėti prisijungimų statistiką. Norint atlikti administravimą pasirenkamas meniu punktas „Administravimas“ iš karto patenkama į vartotojų prisijungimų prie sistemos registravimo langą:

PRISIJUNGIMŲ STATISTIKA		VARTOTOJŲ PATVIRTINIMAI	VARTOTOJŲ TEISĖS	VARTOTOJŲ ŠALINIMAS
Nr	Vardas	Įmonė	El. paštas	Prisijungimai
1	Aurimas	ADMINAS	aj@deform-group.com	78
2	Petras	UAB "Kalbiai"	aurimas.juncis@stud.ktu.lt	2
3	aurymka	ADVISION	aurimas@advision.lt	41
4	petras	OHO	vadaz@one.lt	2

[Neaktyvūs vartotojai](#)
5.17 pav. Administravimo langas

Paspaudus nuorodą „*Neaktyvūs vartotojai*“ parodomi prie sistemos nei karto neprisijungę vartotojai. Pasirinkus punktą „*Vartotojų patvirtinimai*“ išvedamas sąrašas prisiregistravusių, bet dar laukiančių patvirtinimo vartotojų sąrašas. Norimus klientus galima patvirtinti stulpelyje „*Patvirtinti?*“ paspaudus „*Taip*“ . Pasirinkus punktą „*Vartotojų teisės*“ galima keisti sistemos vartotojų teises, tiesiog ties norimu vartotoju pasirenkamas teisės, kurias jam norime suteikti pavadinimas. Vartotojus šalinti galima punkte „*Vartotojų šalinimas*“. Pasirenkam norimo pašalinti vartotojo eilutėje esantį mygtuką „*Taip*“.

Atliktų darbų ataskaitos spausdinimas - spausdinamas nurodytų datų laikotarpyje atliktų darbų sąrašas. Tam pasirenkame meniu punktą „*Ataskaita*“, atsidariusioje formoje pasirenkame už kokį laikotarpį norima spausdinti ataskaitą ir spaudžiame „*Tvirtinti*“. Ekrane gauname tokią ataskaitą:

Per pasirinktą laikotarpį klientam atlikti darbai				
Laikotarpis nuo: 2007-8-1 iki: 2007-9-1				
ADVISION				
Nr.	Darbo pavadinimas	Kiekis	Kaina	Suma
1	Už kazka	34	5	170
2	Atlikti pakeitimai	4	60	240
			Iš viso:	410
			PVM(18%):	73.8
			Bendra suma:	483.8
OHO				
Nr.	Darbo pavadinimas	Kiekis	Kaina	Suma
1	Testinis darbas	4	20	80
			Iš viso:	80
			PVM(18%):	14.4
			Bendra suma:	94.4

5.18 pav. Per pasirinktą laikotarpį atliktų darbų ataskaita

Parametru keitimas – tai vartotojų prisijungimo ir informacijos apie įmonę keitimas. Pasirenkame sistemoje esantį punktą „Paramentai“, kurio forma pavaizduota 5.19 pav.



5.19 pav. Parametru koregavimo forma

Skliausteliuose rodoma esama informacija, kurią norint pakeisti įvedama nauja informacija į šalia esančius laukus.

Atsijungimas nuo sistemos - pasirinkus meniu punktą „Atsijungti“, sistema pereina į prisijungimo lango būseną. Atsijungimo metu sunaikinama darbo sesija, kurioje saugomi tam tikri kliento duomenys, tarp kurių – prisijungimo vardas bei slaptažodis.

Kliento atliekamų funkcijų aprašymas

Dauguma aukščiau aprašytų kliento funkcijų yra identiškos administratoriaus funkcijoms, todėl čia apžvelgiami tik kai kurie jų ypatumai, o naudojimąsis lieka analogiškas.

Savo funkcionalumu besiskiriančios paslaugos:

- a) Naujienų skaitymas – klientams naujienų negalima sukurti, redaguoti ar šalinti. Naujienas galima tik skaityti.
- b) Pranešimų siuntinėjimas – yra beveik analogiška administratoriaus funkcijai, tačiau klientui apribojami kai kurie veiksmai, t.y. negalima keisti užduoties įvykdymo termino ir būsenos. Toliau visi veiksmai atliekami taip pat.
- c) Sąskaitų peržiūra - klientams sąskaitų negalima sukurti, redaguoti ar šalinti. Sąskaitas galima tik peržiūrėti.
- d) Projekto užduoties įvertinimo formos pildymas - tai išskirtinė funkcija, galiojanti tik klientams. Pirmą kartą prisijungęs klientas gali užpildyti projekto užduoties įvertinimo formą, kurią užpildžius ir patvirtinus, sistema apdoroja (konvertuoja į PDF formatą) ir

administratorius gauna elektroniniu paštu prisegtą failą. Norint pildyti formą pasirenkamas meniu punktas „Pildyti analizės formą“. Užpildžius ir išsiuntus formą, šitas meniu punktas dingsta iš sistemos.

5.5. Testavimas

Testavimo modelyje atsispindi sistemos testavimo veiksmų seka bei testavimo duomenys. Pradedant naudotis sistema, reikia prie jos prisiregistruoti, sistemos adresas yra <http://www.deform-group.com/mag/>. Užpildant registracijos formą būtina laikytis nurodytų duomenų įvedimo reikalavimų, t.y. vartotojo vardas turi būt sudarytas tik iš raidžių, el. paštas turi atitikti tinkamą formatą (pvz.: vardas@domain.lt), adresą gali sudaryti bet kokie simboliai, atsakomosios sąskaitos numerį sudaro 18 skaitmenų (sistema automatiškai prieš skaitmenis prirašo raides „LT“), įmonės kodą sudaro 9 skaitmenys, PVM kodą sudaro 9 skaitmenys (sistema automatiškai prieš skaitmenis prirašo raides „LT“), įvestas ir pakartotas slaptažodžiai turi sutapti. Nesilaikant šių reikalavimų sistema gražino klaidos pranešimą. Be to, kai buvo bandoma antrą kartą registruotis tuo pačiu el. paštu, sistema neleido to padaryti ir išvedė apie tai pranešimą. Įmonės pavadinimą, kliento vardą ir slaptažodį turi sudaryti ne mažiau kaip 3 simboliai, be to reikėjo laikytis ir kitų nurodytų simbolių ilgių apribojimų, antraip registravimasis buvo nesėkmingas.

Prisijungti prie sistemos galima skirtingomis vartotojo teisėmis, klaidingai įvedus prisijungimo vardą ir/arba slaptažodį realiu laiku buvo išvestas atitinkamas pranešimas. Užmiršus prisijungimo slaptažodį, galima pasinaudoti slaptažodžio priminimo f-ja. Tiesiog į laukelį įvedamas prisijungimo vardas (tai el. paštas, kuriuo registruojamasi sistemoje). Jei sistema neranda savo duomenų bazėje tokio el. pašto arba jis įvestas nekorektiškai, apie tai buvo išvedamas pranešimas.

Prisijungus administratoriaus teisėmis, buvo ištestuotos šios paslaugos:

- a) administruoti vartotojus - su sistemos vartotojais buvo bandoma atlikti šiuos veiksmus: patvirtinti prisiregistravusius vartotojus, šalinti vartotojus, priskirti vartotojams teises (administratoriaus arba kliento), peržiūrėti vartotojų prisijungimų statistiką. Pabandyta atsijungti ir vėl prisijungti prie sistemos, patikrinta statistika – pastebėta, jog testuojamo vartotojo prisijungimų skaičius vienetu padidėjo. Šalinant klientus ar keičiant teises, sistema pasiteiravo, ar tikrai mes tai norime padaryti, patvirtinus, buvo atliktas atitinkamas veiksmas. Bandant pašalinti administratoriaus teises turintį vartotoją sistema liepė prieš tai pakeisti jo teises į paprasto vartotojo. Pakeitus vartotojo teises stulpelyje „Teisės“ buvo pakeistas pavadinimas atitinkamai į „user“ arba „admin“. Patvirtinus vartotoją įrašas apie laukiančius patvirtinimo vartotojus dingo.
- b) registruoti klientam atliktus darbus – su darbais buvo bandoma atlikti tokius veiksmus: sukurti naują, peržiūrėti, šalinti ir redaguoti esamą, atlikti paiešką. Registruojant naują darbą

įvedame darbo pavadinimą, kurį gali sudaryti bet kokie simboliai; įvedame kiekį (valandomis ar vienetais) ir vieneto kainą, kurie turi būti sudaryti tik iš skaičių, antraip buvo išvedamas klaidos pranešimas. Naujas darbas atsirando darbų sąrašė. Užregistruotą darbą pakoregavome laikydamiesi tokių pačių duomenų įvedimo taisyklių. Korekcijos išsaugotos. Pasirinkus įrašo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – įrašas buvo pašalintas. Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. Rašant žodį realiu laiku buvo pateikiamas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą buvo išvestas sąrašas rastų įrašų, nesėkmės atveju gavome atitinkamą pranešimą.

- c) išrašyti sąskaitas - su sąskaitomis buvo bandoma atlikti tokius veiksmus: išrašyti naują, peržiūrėti, patvirtinti apmokėjimą, šalinti, atlikti paiešką. Išrašant sąskaitą iš sąrašo buvo pasirenkamas klientas, įrašoma sąskaitą išrašančio asmens vardas, pavardė ir pareigos ir patvirtinus veiksmą ekrane išvesta sąskaita. Ekrane akimirksniu pasirodė sąrašas neapmokėtų sąskaitų. Jei nebuvo nei vieno kliento, kuriam būtų atlikta darbų, tai patvirtinimo mygtukas, iškrentantis meniu ir asmens įvedimo langas buvo neaktyvūs. Pasirinkus sąskaitos apmokėjimo patvirtinimo opciją, sąskaitos apmokėjimo būseną pasikeitė į „Taip“ ir atitinkamas įrašas iš neapmokėtų sąskaitų sąrašo dingo realiu laiku. Pasirinkus sąskaitos šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip ir jei sąskaita yra apmokėta – sąskaita buvo pašalinama, priešingu atveju gautas pranešimas apie sąskaitos neapmokėjimą. Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. Rašant žodį realiu laiku buvo pateikiamas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą išvestas sąrašas rastų įrašų, nesėkmės atveju gautas atitinkamas pranešimas.
- d) skelbti naujienas - su naujienomis buvo bandoma atlikti tokius veiksmus: sukurti naują, peržiūrėti, šalinti ir redaguoti esamą. Skelbiant naują naujieną buvo įvedama naujienos antraštė ir naujienos turinys, kurį gali sudaryti bet kokie simboliai. Nauja naujiena atsirado naujienų sąrašė. Paskelbtą naujieną pakoregavome laikydamiesi tokių pačių duomenų įvedimo taisyklių. Korekcijos išsaugotos. Pasirinkus įrašo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – įrašas pašalinamas. Atliekant šiuos veiksmus kliento lange realiu laiku pasikeitė naujienų sąrašas.
- e) spausdinti darbų atlikimų ataskaitą - įvestas norimas laikotarpis, jei nurodytu laikotarpiu nebuvo atliktų darbų, gavome apie tai pranešimą, priešingu atveju ekrane buvo atspausdinama ataskaita.
- f) atsakyti klientam apie užduočių įvykdymo laikotarpį ir būseną – pasirinktas kliento atsiųstas pranešimas, peržiūrėtas, paspausta ikona „atsakyti“ ir įvestas užduoties įvykdymo laikotarpis, pastabos ir išsiųsta klientui. Klientui akimirksniu buvo pateikiamas pranešimas

pagrindiniame lange, jog yra naujas administratoriaus pranešimas. Jei administratorius pasirinko užduoties įvykdymo patvirtinimo opciją, tai būseną pasikeitė į „Įvykdyta“. Pasirinkus pranešimo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – įrašyta pranešimo šalinimo žymė ir administratorius pranešimo sąrašė nebematė, o klientas vis dar matė, kol taip pat nepasirinko šalinimo funkcijos (tuomet įrašas pašalintas iš duombazės). Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. Rašant žodį realiu laiku pateiktas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą išvestas sąrašas rastų įrašų, nesėkmės atveju gautas atitinkamas pranešimas.

Prisijungus kliento teisėmis, buvo ištestuotos šios paslaugos:

- a) skaityti naujienas – klientas pasirinko iš sąrašo norimą naujieną ir perskaitė pateiktą informaciją. Nei šalinti, nei redaguoti naujienos jis negalėjo. Kai administratorius sukūrė naują naujieną, pakoregavo esamą arba pašalino, tai kliento ekrane akimirksniu pasikeitė naujienų sąrašas.
- b) siųsti administratoriui pranešimą (užduotį) – sukuriant naują pranešimą buvo įvedama pranešimo antraštė, kurią gali sudaryti ne daugiau 50 bet kokių simbolių ir įvedamas turinys. Viršijus antraštės duomenų ilgį, gauntas sistemos pranešimas. Jei duomenys suvesti teisingai, pranešimas (užduotis) buvo išsiunčiamas. Administratorius pagrindiniame lange realiu laiku gavo pranešimą, jog gauta nauja žinutė. Po to klientas galėjo sekti informaciją apie užduoties įvykdymo laikotarpį ir būseną. Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. Rašant žodį realiu laiku pateiktas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą išvestas sąrašas rastų įrašų, nesėkmės atveju gautas atitinkamas pranešimas. Pasirinkus pranešimo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – buvo įrašyta pranešimo šalinimo žymė ir klientas pranešimo sąrašė nebematė, o administratorius vis dar matė, kol taip pat nepasirinko šalinimo funkcijos (tuomet įrašas buvo pašalinamas iš duombazės).
- c) ieškoti ir peržiūrėti išrašytas sąskaitas (šalinti sąskaitas klientas neturi teisės). Pasirinkęs sąskaitų peržiūros punktą klientas gavo pranešimą apie neapmokėtas sąskaitas. Kai administratorius patvirtino sąskaitos apmokėjimą, pranešimo lango duomenys realiu laiku pasikeitė.
- d) pildyti projekto užduoties įvertinimo formą - naujai prisiregistravęs vartotojas užpildė projekto užduoties įvertinimo formą. Jei privalomi laukai buvo paliekami tušti, ar buvo netinkamo formato, forma buvo neišsiunčiama, priešingu atveju – gautas pranešimą apie atliktą sėkmingą operaciją. Antrą kartą formos pildyti nebuvo leista (menui punktas dingsta).

Ištestuotos bendros visiems sistemos vartotojams funkcijos:

- a) naudojimąsis adresų knygele – sukuriant naują adresą formoje buvo įvedami reikiami duomenys, jei duomenys buvo nekorektiški arba neužpildyti privalomi laukai, apie tai realiu laiku (dar prieš išsiunčiant duomenis) buvo išvedamas pranešimas. Jei viskas gerai, kontaktas buvo išsaugomas ir parodomas sąrašė. Pabandyta redaguoti kontaktą – jei duomenys buvo korektiški, korekcijos buvo išsaugomos, priešingu atveju gautas atitinkamas pranešimas apie klaidas. Pasirinkus kontakto peržiūros punktą, buvo pateikiamas korektiškas kontakto gyvenamosios vietos žemėlapis, pagal suvestus adresu duomenis. Pasirinkus įrašo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – įrašas buvo pašalinamas. Pasirenkus duomenų eksportavimo funkciją – sistema paklausė, kur mes norime išsaugoti duomenis, pasirenkus norimą vietą duomenys buvo korektiškai išsaugomi html formatu. Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. Rašant žodį realiu laiku buvo pateikiamas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą buvo išvedamas sąrašas rastų įrašų, nesėkmės atveju gautas atitinkamas pranešimas.
- b) naudojimąsis užrašų knygele – sukuriant naują užrašą formoje buvo įvedami reikiami duomenys, jei duomenys buvo nekorektiški arba neužpildyti privalomi laukai, apie tai realiu laiku (dar prieš išsiunčiant duomenis) buvo išvedamas pranešimas. Jei viskas gerai, užrašas buvo išsaugomas ir parodomas sąrašė. Pabandyta redaguoti užrašą – jei duomenys buvo korektiški, korekcijos buvo išsaugomos, priešingu atveju gautas atitinkamas pranešimas apie klaidas. Pasirinkus įrašo šalinimą sistema pasiteiravo, ar mes tikrai to norime, jei taip – įrašas buvo pašalinamas. Įvedėme įvykio datą ir patikrinom ar gausime apie jį pranešimą. Rezultate gavome pranešimą el. paštu vieną dieną prieš nurodytą įvykio datą. Ši funkcija suveikė teisingai. Norėdami atlikti paiešką pasirenkame kriterijų, įvedame nurodyto formato ieškomą žodį. . Rašant žodį realiu laiku buvo pateikiamas pasiūlymų sąrašas, iš kurio pasirinkus pageidaujamą įrašą ir išsiuntus formą buvo išvedamas sąrašas rastų įrašų, nesėkmės atveju gautas atitinkamas pranešimas.
- c) parametrų keitimas – buvo įvedami nauji vartotojo parametrai, jei šie buvo korektiški ir buvo įvestas teisingas senas slaptažodis bei sutapo įvestas naujas ir pakartotas slaptažodžiai, tai nauji nustatymai buvo išsaugomi, kitu atveju apie nekorektiškus duomenis realiu laiku (dar prieš išsiunčiant duomenis) buvo pateikiamas klaidos pranešimas.
- d) atsijungus nuo sistemos buvo sugrįžtama į prisijungimo langą, o paspaudus naršyklėje „back“ mygtuką sistema daugiau nefunkcionavo. Reiškia sistema apsaugota nuo nesankcionuoto naudojimosi.

6. CRM sistemos patobulinimo įvertinimas

Šiame skyriuje atliekamas dinamiškų sistemos savybių tyrimas ir įvertinimas pagal devynis scenarijus, pateikiama įvertinimų suvestinė ir išvados bei suformuluotos RIA taikomųjų programų savybių pritaikymo CRM ir kitoms organizacijų sistemoms rekomendacijos.

6.1. Dinamiškų sistemos savybių tyrimas ir įvertinimas

Dinamiškų sistemos savybių tyrimui ir įvertinimui pasirinktos kelios esminės 2.2. poskyryje išskirtos RIA taikomosioms programoms būdingos savybės:

- 1) raiškios sąveikos galimybės
- 2) kliento pusės biznio logika
- 3) dalinis puslapio atnaujinimas
- 4) retos serverio iteracijos (round-trips)
- 5) spartus veikimas
- 6) patogumas/aiškumas naudoti

Bandysime nustatyti kiek sistemoje yra dinamiškų savybių, kurios taip pat rodys sistemos dinamiškumo lygį. Vertinimo skalė pasirinkta nuo 0 iki 10. Aptarsime skalės reikšmes kiekvienai savybei 6.1 lentelėje.

6.1 lentelė Savybių vertinimo aprašymas

Nr.	Savybė	Vertinimo aprašymas
1.	Raiškios sąveikos galimybės	Įvertinama, ar sistemos komponentų tarpusavio sąveika yra raiški. Tarkim, jei tarp susijusių sistemos komponentų pasikeičia tam tikra informacija ir ji iškart realiu laiku atvaizduojama puslapyje, tai šią savybę galima vertinti 10. Jei norint pamatyti pakeitimus vartotojas turi atlikti papildomus veiksmus, tai sąveika nėra raiški ir savybė įvertinama žemu balu.
2.	Kliento pusės biznio logika	Įprastose Web sistemose kliento pusės biznio logika yra labai ribota, o Web 2.0 kliento pusėje įkeliamas Ajax varikliukas, kuris yra tarpininkas tarp kliento ir serverio, o jame yra atliekama dalis verslo logikos. Todėl, jeigu konkrečioms sistemos funkcijoms atlikti panaudojamas Ajax varikliukas, tai ši savybė vertinama aukščiausiais balais, priklausomai nuo biznio logikos sudėtingumo ir panaudojamumo.
3.	Dalinis puslapio atnaujinimas	Įvertinama, ar yra realizuotas dalinis puslapio atnaujinimas. Jei vykdant operacijas atnaujinama tik dalis puslapio, tai suteikiami aukštesni balai. Jei vykdant operacijas, perkraunamas visas puslapis, tai savybės įvertinimas lygus 0.
4.	Retos serverio	Kuo retesnės serverio iteracijos, tuo sistema veikia greičiau. Jei vykdant

	iteracijos	tam tikrą scenarijų yra labai dažnai kreipiamasi į serverį, tai ši savybė vertinama 0. Ir atvirkščiai, jei iteracijos yra retos, tai vertinama 10.
5.	Spartus veikimas	Kuo didesnis veikimo spartos įvertinimas – tuo dinamiškesnė yra sistema. Veikimo sparta ne visada būtinai parodys tik dinamiškų komponentų veikimą, bet kartu ir nedinamiškų, jei jos yra sistemoje. Tarkime įvertiname veikimo spartą reikšme 0. Tai reikš, kad sistema veikia labai lėtai, jame visai nėra naudojamas Ajax ir apskritai jis nėra tinkamas naudoti. Jei įvertinimas 10 – sistema veikia labai greitai, beveik visos jo pagrindinės funkcijos naudoja Ajax. Kiti įvertinimai tarp 0 ir 10 parodys bendrą sistemos veikimo spartos įvertinimą, kartu įvertinus visų funkcijų veikimo spartą.
6.	Patogumas/aiškumas naudoti	Parodo, kiek sistemos veikimas yra aiškus ir patogus vartotojui. Ar sukelia kokių nesuprantamų sistemos veiksmų ir pan. Jei įvertintume sistemos veikimo aiškumą/patogumą nuliui – sistemos veikimas visiškai nesuprantamas, nepatogus vartotojui. Tai gali sukelti per sudėtingas funkcijų įvertinimas ir pan. Jei įvertiname sistemos naudojimosi patogumą 10 – tai reiškia, kad sistemos veikimas yra aiškus ir Ajax naudojimas jame nepasunkino vartotojo supratimo, kaip naudotis sistema ir kokius ji rezultatus pateikia.

Dinamiškumo pasiekimo nagrinėjimui sukuriame konkretūs sistemos atliekamų veiksmų scenarijai (jiems bus įvertintos aukščiau nagrinėtos savybės):

- 1) **Scenarijus1:** Vartotojo prisijungimo prie sistemos duomenų korektiškumo ir teisingumo tikrinimas realiu laiku.
- 2) **Scenarijus2:** Formos duomenų validacija realiu laiku (prieš išsiunčiant formos duomenis), kuriant naują įrašą.
- 3) **Scenarijus3:** Įrašų paieškos vykdymas. Vedant paieškos žodį sistema automatiškai pasiūlo galimus variantus, dar prieš išsiunčiant formos duomenis.
- 4) **Scenarijus4:** Momentinis kontakto buvimo vietos nustatymas ir atvaizdavimas integruotoje “Google Maps” sistemoje pagal įvestą vietovę.
- 5) **Scenarijus5:** Naujienų sąrašo atnaujinimas realiu laiku, administratoriui sukūrus naują arba pakoregavus naujieną.
- 6) **Scenarijus6:** RSS naujienų skaitymas.
- 7) **Scenarijus7:** Vartotojų administravimo meniu navigavimas, neperkraudant visos puslapio informacijos.
- 8) **Scenarijus8:** Momentinis pranešimas vartotojo lange, gavus naują žinutę iš kito vartotojo.

9) **Scenarijus9:** Momentinis pranešimas administratoriaus lange apie vartotojų neapmokėtas sąskaitas.

Visi eksperimentai buvo atlikti, Windows XP Home Edition OS, Intel(R) Celeron(R) CPU 2.80GHz ir 448MB operatyvios atminties, kompiuteryje. Interneto greitis – 1 Mb/s. Duomenų bazė ir informacinė sistema veikia nutolusiame serveryje. Visi skaičiavimai paimti iš sistemos ir serverio teikiamos tekstinės veikimo informacijos (logs).

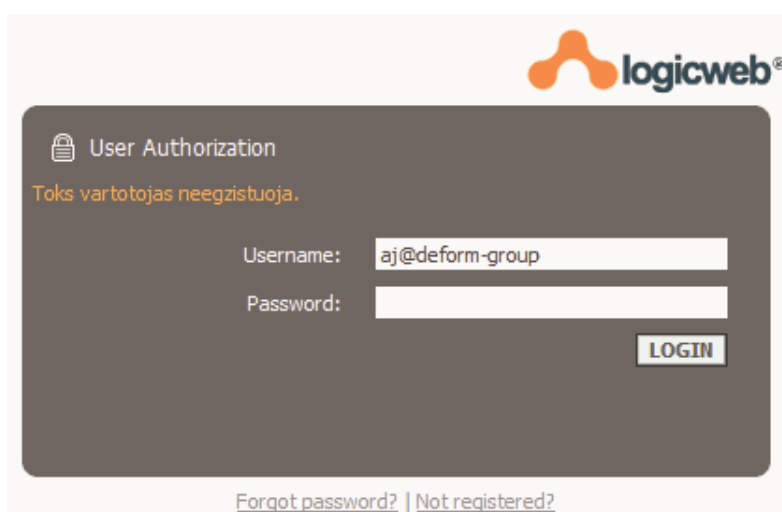
Priimsime, jog sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus, jei bendras dinamiškųjų sistemos savybių vidutinis įvertinimas bus **ne mažesnis nei 7** (iš 10 galimų).

Eksperimentus atliko ir įvertinimus skyrė trys įmonėje, kurios CRM sistema buvo tobulinama, dirbantys darbuotojai – ekspertai:

1. Projektų vadovas: Andrius Kučiauskas
2. Programuotojas: Justas Adžgauskas
3. Projektų vadovas – programuotojas: Aurimas Junčis

Vykdomas eksperimentas su **Scenarijus1:**

Prisijungimo prie sistemos lange vedamas prisijungimo vardas ir slaptažodis. Berašant duomenis, realiu laiku išvedamas pranešimas (6.1 pav), jog „Toks vartotojas neegzistuoja“, kol neįvedamas sistemoje jau prisiregistravęs vartotojas. Tuomet pranešimas dingsta ir toliau vedamas slaptažodis. Berašant slaptažodį, Ajax varikliukas realiu laiku tikrina ar slaptažodis atitinka vartotojo vardą. Kol neatitinka, rodomas pranešimas, jog „Klaidingas slaptažodis“. Tam, kad įsitikinti, jog vedami duomenys yra teisingi, vartotojui nereikia išsiųsti formos duomenų – taip sutaupomas laikas ir atsiranda patogumas naudotis sistema.



The image shows a web browser window displaying a login form for 'logicweb'. The form is titled 'User Authorization' and has a lock icon. A message in orange text says 'Toks vartotojas neegzistuoja.' Below this, there are two input fields: 'Username:' with the value 'aj@deform-group' and 'Password:'. A 'LOGIN' button is located to the right of the password field. At the bottom of the form, there are two links: 'Forgot password?' and 'Not registered?'.

6.1 pav. pagal Scenarijus1 atliekamo eksperimento vaizdas

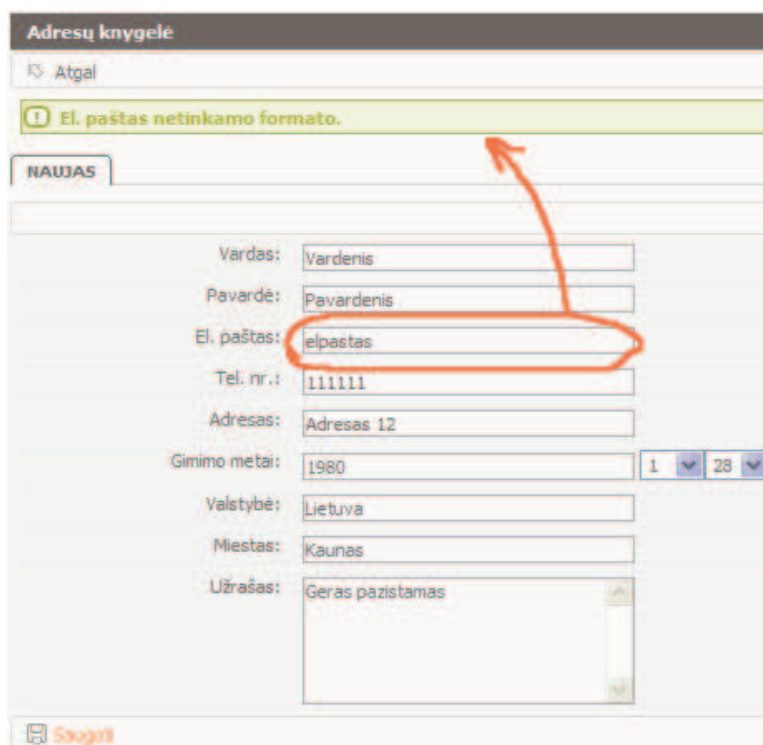
Savybių įvertinimas pagal eksperimento rezultatus:

6.2 lentelė Savybių įvertinimas pagal Scenarijus1

Scenarijus1						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	9	10	10	7	10	9
Antro eksperto įvertinimas	9	9	10	8	10	9
Trečio eksperto įvertinimas	9	10	9	7	10	9

Vykdomas eksperimentas su **Scenarijus2**:

Naujo įrašo kūrimo lange vedami pageidaujami duomenys. Berašant duomenis, realiu laiku Ajax varikliukas tikrina įvedamų duomenų korektiškumą, jei aptinkami klaidingi duomenys, išvedamas tai bylojantis pranešimas (6.2 pav). Pataisius klaidingus duomenis pranešimas akimirksniu dingsta. Tam, kad įsitikinti, jog vedami duomenys yra teisingi, vartotojui nereikia išsiųsti formos duomenų – taip sutaupomas laikas, vartotojo neerzina pilnas puslapio perkrovimas ir sistema tampa patogesne.



6.2 pav. pagal Scenarijus2 atliekamo eksperimento vaizdas

Savybių įvertinimas pagal eksperimento rezultatus:

6.3 lentelė Savybių įvertinimas pagal Scenarijus2

Scenarijus2						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	9	10	10	7	10	10
Antro eksperto įvertinimas	10	9	10	8	10	10
Trečio eksperto įvertinimas	9	10	10	8	10	9

Vykdomas eksperimentas su **Scenarijus3**:

Ieškomas norimas kontaktas. Pasirenkamas kriterijus, pagal kurį bus ieškoma – pagal „Vardą“. Tuomet vedamas paieškos žodis. Parašius jau pirmą raidę, realiu laiku Ajax varikliukas tikrina

įvedamus simbolius ir ieško panašių atitikmenų duomenų bazėje. Jei įrašai randami, jų sąrašas akimirksniu pateikiamas žemiau vedamų simbolių esančiam bloke (6.3 pav). Iš pateikto sąrašo galima pasirinkti konkretų ieškomą kontaktą. Taip žymiai pagerėja sistemos interaktyvumas ir dinaminės savybės, pagreitėja paieškos procesas ir apsaugojama nuo tuščių rezultatų gavimo.

6.3 pav. pagal Scenarijus3 atliekamo eksperimento vaizdas

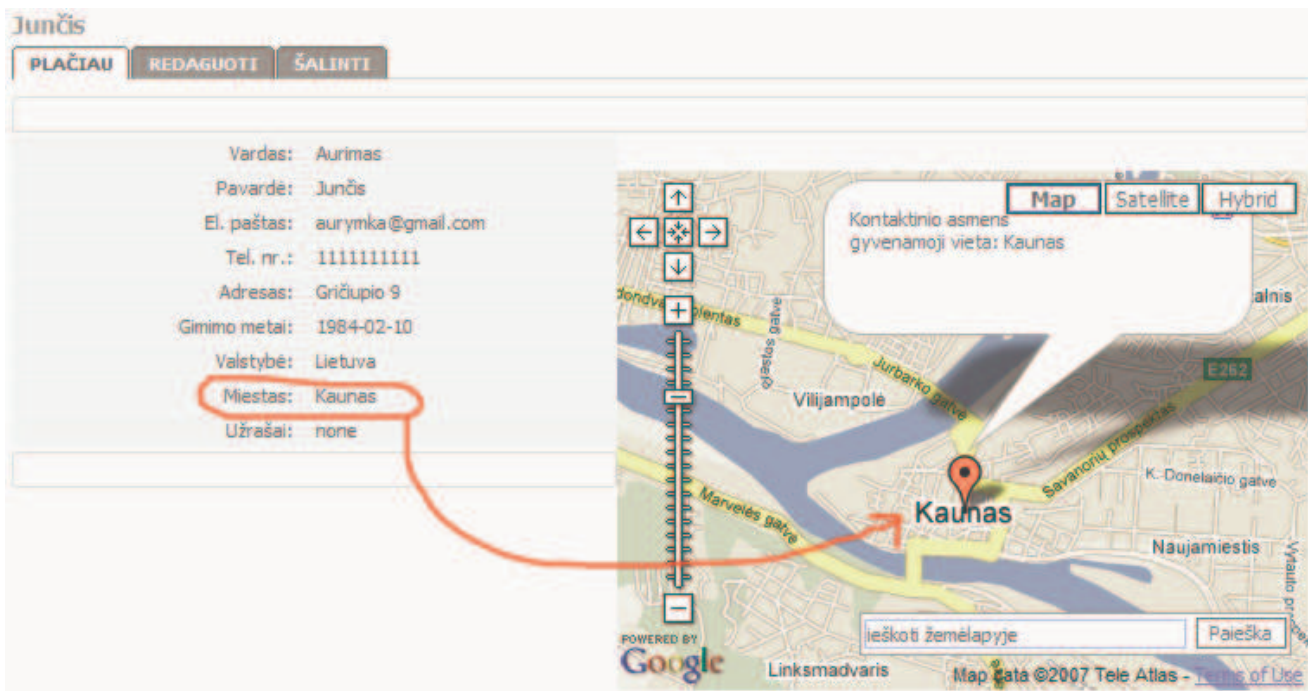
Savybių įvertinimas pagal eksperimento rezultatus:

6.4 lentelė Savybių įvertinimas pagal Scenarijus3

Scenarijus3						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	9	9	10	8	10	9
Antro eksperto įvertinimas	9	9	10	9	10	9
Trečio eksperto įvertinimas	9	9	9	7	10	10

Vykdomas eksperimentas su **Scenarijus4:**

Kuriant naują kontaktą, įvedama vietovė (miestas), kurioje gyvena asmuo. Pasirinkus kontakto peržiūros punktą pagal išsaugotus duomenis sugeneruojamas vietovės žemėlapis (6.4 pav) su pažymėta kontakto buvimo vieta. Pakeitus vietovės duomenis, žemėlapis sugeneruojamas iš naujo. Šio funkcionalumo trūkumas yra tas, kad integruota sistema neaptinka labiau sukonkretintų Lietuvos vietovių ir apsiriboja miestų masteliu. Ši sistema labai patogi vartotojams, pageidaujantiems žinoti savo adresų knygelėje esančių kontaktų buvimo vietą. Nereikia naudotis jokiais papildomomis priemonėmis, viskas matoma viename lange. Sistemai tai suteikia daugiau raiškumo.



6.4 pav. pagal Scenarijus4 atliekamo eksperimento vaizdas

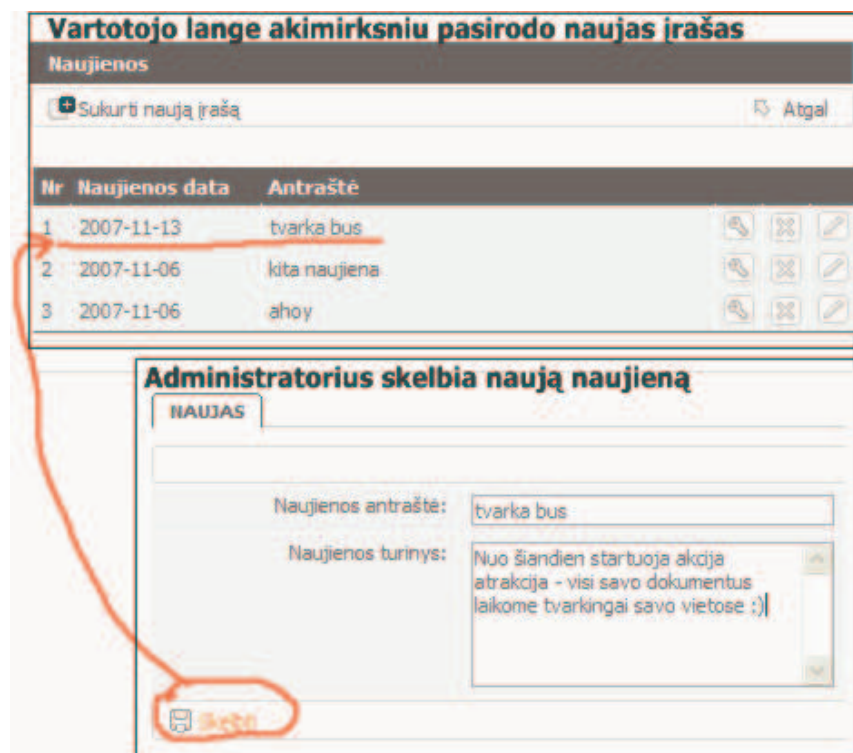
Savybių įvertinimas pagal eksperimento rezultatus:

6.5 lentelė Savybių įvertinimas pagal Scenarijus4

Scenarijus4						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	10	8	3	9	7	9
Antro eksperto įvertinimas	9	9	5	9	7	9
Trečio eksperto įvertinimas	10	9	4	8	7	10

Vykdomas eksperimentas su **Scenarijus5**:

Kiekvienas vartotojas, prisijungęs prie sistemos, mato naujienų sąrašą. Kai administratorius paskelbia naują naujieną arba pakoreguoja esamą, vartotojo lange esantis naujienų sąrašas akimirksniu atsinaujina (6.5 pav). Vartotojui nereikia perkrauti puslapio tam, kad pasitikrinti, ar yra naujų įrašų ar pakeitimų. Taip pasiekiamas didesnis interaktyvumas, dinamiškumas ir taupomi vartotojų atliekami veiksmai, siekiant gauti pageidaujamos informacijos.



6.5 pav. pagal Scenarijus5 atliekamo eksperimento vaizdas

Savybių įvertinimas pagal eksperimento rezultatus:

6.6 lentelė Savybių įvertinimas pagal Scenarijus5

Scenarijus5						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	10	10	10	6	8	10
Antro eksperto įvertinimas	10	9	10	6	8	9
Trečio eksperto įvertinimas	10	10	9	7	8	10

Vykdomas eksperimentas su **Scenarijus6:**

Kiekvienas vartotojas, prisijungęs prie sistemos, mato naujienų sąrašą, o po juo pateikiamas RSS naujienų blokas. Šiuo konkrečiu atveju skaitomos tinklapio delfi.lt IT kanalo naujienos. RSS skaitytuvas rodo pirmas penkias naujausias naujienų antraštes, trumpus aprašymus ir nuorodas į pilną straipsnio aprašymą. Siekiant dinamiškumo ekrane vienu metu rodoma tik viena naujiena, sekančios naujienos periodiškais, kas tris sekundes keičiasi. Taip sutaupoma vietos ekrane ir vartotojas neapkraunamas per dideliu informacijos kiekiu. Vartotojui ši naujienų skiltis yra patogi tuom, kad visuomet mato prieš akis naujausių straipsnių antraštes, o pasirinkęs sudominusį straipsnį gali perskaityti delfi.lt portale.

IT naujienos RSS formatu:

Rusai įsigijo „LiveJournal“

Blogų pionieriai „Six Apart“ skiniasi su svetaine „LiveJournal“ – dar viena novatoriška svetaine, suteikusia galimybę masėms rašyti internetinius dienoraščius, praneša „Associated Press“.

[Skaityti daugiau](#)

6.6 pav. pagal Scenarijus6 atliekamo eksperimento vaizdas

Savybių įvertinimas pagal eksperimento rezultatus:

6.7 lentelė Savybių įvertinimas pagal Scenarijus6

Scenarijus6						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	10	9	10	8	9	8
Antro eksperto įvertinimas	10	9	10	8	8	9
Trečio eksperto įvertinimas	10	10	10	7	9	9

Vykdomas eksperimentas su **Scenarijus7**:

Vartotojų administravimo dalyje naviguojant tarp skirtingų meniu punktų vyksta tik dalinis puslapio atnaujinimas, t.y. pasikeičia tik su pasirinktu meniu punktu susijusi informacija (6.7 pav). Likusi puslapio informacija lieka nepakitusi ir nebereikalingas pilnas puslapio perkrovimas.



6.7 pav. pagal Scenarijus7 atliekamo eksperimento vaizdas

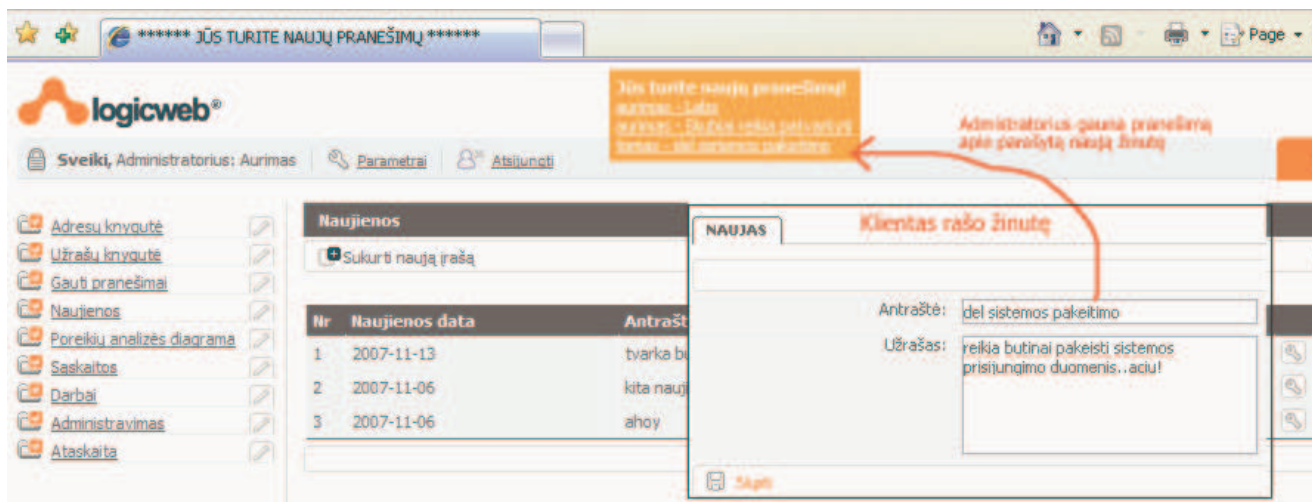
Savybių įvertinimas pagal eksperimento rezultatus:

6.8 lentelė Savybių įvertinimas pagal Scenarijus7

Scenarijus7						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	6	10	10	9	8	9
Antro eksperto įvertinimas	6	10	9	8	8	9
Trečio eksperto įvertinimas	6	10	10	9	9	9

Vykdomas eksperimentas su **Scenarijus8**:

Klientas parašo žinutę administratoriui. Administratoriaus sistemos darbiniam lange akimirksniu pasirodo pranešimas (6.8 pav) apie gautas naujas žinutes. Pasirinkęs pageidaujamą žinutę, administratorius gali ją perskaityti ir pranešimo sąrašas akimirksniu pasikeičia (perskaityta žinutė iš sąrašo dingsta). Taip administratorius sutaupo didelę laiko dalį, kadangi jam nebereikia karts nuo karto naviguoti per meniu punktus ir tikrinti, ar niekas neparasė naujo pranešimo. Tai viena geriausių ir svarbiausių sistemos patobulinimo funkcijų, suteikianti sistemai daug dinamiškumo, patogumo, raiškumo ir paspartinanti sąveiką tarp vartotojų.



6.8 pav. pagal Scenarijus8 atliekamo eksperimento vaizdas

Savybių įvertinimas pagal eksperimento rezultatus:

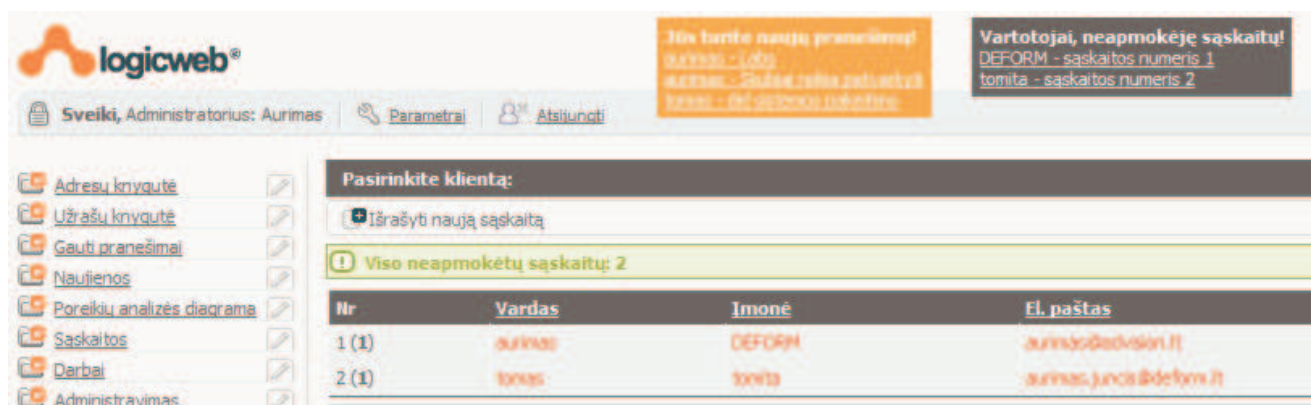
6.9 lentelė Savybių įvertinimas pagal Scenarijus8

Scenarijus8						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	10	10	10	7	10	10
Antro eksperto įvertinimas	10	10	10	8	10	9
Trečio eksperto įvertinimas	10	10	10	7	9	10

Vykdomas eksperimentas su **Scenarijus9**:

Administratorius užregistruoja klientui atliktus darbus ir už juos išrašo sąskaitą. Išrašytos sąskaitos statusas tampa „Neapmokėta“. Sąskaitos statusas pasikeičia į „Apmokėta“, kai administratorius pamato pavedimą ir apmokėjimą patvirtina prisijungęs prie sistemos. Kol yra

neapmokėtų sąskaitų, klientui sistemoje pateikiamas tai bylojantis pranešimas (6.9 pav), kurio turinys (sąskaitų sąrašas) kinta realiu laiku administratoriui patvirtinus apmokėjimą arba išrašius naują sąskaitą. Administratorius savo ruožtu taip pat mato klientus ir jų neapmokėtas sąskaitas, kurių sąrašas kinta realiu laiku, kai kitas administratorius patvirtina sąskaitos apmokėjimą. Ši funkcija pasitarnauja įmonės finansų valdymo efektyvumo didinimui ir sąskaitų apmokėjimo proceso stebėjimui bei kontrolei.



6.9 pav. pagal Scenarijus9 atliekamo eksperimento vaizdas

Savybių įvertinimas pagal eksperimento rezultatus:

6.10 lentelė Savybių įvertinimas pagal Scenarijus9

Scenarijus9						
Savybė	1	2	3	4	5	6
Pirmo eksperto įvertinimas	10	10	10	7	10	10
Antro eksperto įvertinimas	10	10	10	8	10	9
Trečio eksperto įvertinimas	10	10	10	7	9	10

6.2. Įvertinimų suvestinė ir išvados

Taigi įvertinome sukurtus dinamiškus sistemos elementus. Galime reziumuoti visus rezultatus. Kiekvieno scenarijaus savybės įvertinimas gaunamas sudėjus visų ekspertų įvertinimus ir padalinus iš ekspertų skaičiaus.

Apskaičiuojame kiekvienos dinamiškos savybės vidutinį įvertinimą sudedant visų ekspertų atliktų scenarijų vidutinius įvertinimus ir padalinant iš scenarijų skaičiaus. Vidutinis scenarijaus įvertinimas skaičiuojamas sudedant visus scenarijaus savybių įvertinimus ir padalinant iš savybių skaičiaus. Bendras vidutinis įvertinimas apskaičiuojamas sudedant visus savybių vidutinius įvertinimus ir padalinant iš savybių skaičiaus.

6.11 lentelė. Rezultatų suvestinė

Savybės	1	2	3	4	5	6	Vidutinis scenarijaus įvertinimas
Scenarijus1	9	9,67	9,67	7,33	10	9	$54,67 / 6 = 9,11$
Scenarijus2	9,33	9,67	10	7,67	10	9,67	$56,33 / 6 =$

							9,39
Scenarijus3	9	9	9,67	8	10	9,33	55 / 6 = 9,17
Scenarijus4	9,67	8,67	4	8,67	7	9,33	47,33 / 6 = 7,89
Scenarijus5	10	9,67	9,67	6,33	8	9,67	53,33 / 6 = 8,89
Scenarijus6	10	9,33	10	7,67	8,67	8,67	54,33 / 6 = 9,06
Scenarijus7	6	10	9,67	8,67	8,33	9	51,67 / 6 = 8,61
Scenarijus8	10	10	10	7,33	9,67	9,67	56,67 / 6 = 9,45
Scenarijus9	10	10	10	7,33	9,67	9,67	56,67 / 6 = 9,45
Vidutinis įvertinimas	83 / 9 = 9,22	86 / 9 = 9,56	82,67 / 9 = 9,19	69 / 9 = 7,66	81,33 / 9 = 9,04	84 / 9 = 9,33	
Bendras vidutinis įvertinimas	54 / 6 = 9 balai						

Taigi, iš 6.11 lentelės matome, kad operacijos su Ajax atliekamos daug greičiau. Kiek greičiau, lemia per kiek laiko pasileidžia pats puslapis, nes be Ajax pagalbos, norint atnaujinti dalį puslapio, reikia perkrauti visą puslapį iš naujo. Jei puslapyje daug operacijų – Ajax privalumas nenuginčytinas. Iš vertinimų sumų matome, kad dinamiškiausi yra pirmas, antras, trečias, aštuntas ir devintas scenarijai. Šiuose scenarijuose panaudotos funkcijos labiausiai pasitarnauja CRM sistemos dinaminės interneto sąsajos tobulinimui.

Galima teigti, jog sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus, kadangi bendras dinamiškųjų sistemos savybių vidutinis įvertinimas (9 balai) viršija prieš tai nustatytą ribinį 7 balų įvertinimą.

Atliktas vertinimas yra tik apytikslis, gali nedinamiškas puslapis turėti tokį panašų vertinimą, kaip ir dinamiškas, jei nedinamiškame puslapyje bus nedaug funkcijų, arba tos funkcijos bus paprastos ir puslapio persikrovimas bus greitas, arba jo visai nereikės, bus naudojami frame'ai.

Mažiausiai dinamiškas yra ketvirtasis scenarijus, kadangi integruotoji sistema dar nėra pakankamai lanksti ir tiksli. Be to, šiame scenarijuje yra daug nedinamiškų operacijų, nėra dalinio puslapio atnaujinimo ir pats žemėlapis užkrovimas yra pakankamai lėtas. Paminėtos savybės labai lėtina šio puslapio funkcionavimą, todėl ir vertinimas yra mažiausias. Galima daryti išvadą, kad patobulinta sistema yra dinamiška ir interaktyvi, nes pagrindinės, dažniausiai naudojamos funkcijos yra sukurtos naudojant Ajax technologijas. Likusios funkcijos, kurios veikia be Ajax pagalbos neįneša per daug nedinamiškumo sistemai, kad būtų ją galima laikyti nedinamiška.

CRM sistemos patobulinimo įvertinimas laiko atžvilgiu

Patobulinti egzistuojančią CRM sistemą apytiksliai užtruks 40 vieno žmogaus darbo valandų. Tam, kad sukurti tokią dinamišką CRM sistemą nuo nulio (pvz.: naudojant naują programavimo kalbą ar kokį nors naują dinaminį karkasą), apytiksliai prireiktų 160 vieno žmogaus darbo valandų. Taigi galima teigti, jog šiame darbe taikoma metodika leidžia sutaupyti 75 procentus darbo laiko. Tai yra pakankamai nebrangus būdas egzistuojančiai taikomajai programai suteikti daugiau dinamiškumo.

6.3. RIA taikomųjų programų savybių pritaikymo CRM ir kitoms organizacijų sistemoms rekomendacijos

Dinamiškų puslapių savybių tyrimo rezultatai parodė, jog Ajax panaudojimas nagrinėtoje CRM sistemoje įnešė daugiau patogumo, interaktyvumo, dinamiškumo, pagerino vartotojo sąsają, paspartino komunikavimo procesus.

Remiantis šiais rezultatais sudarytos RIA taikomųjų programų pritaikymo CRM sistemoms rekomendacijos panaudojant Ajax priemones:

- Vartotojo prisijungimo prie sistemos duomenų korektiškumą ir teisingumą tikrinti realiu laiku, dar prieš išsiunčiant apdorojimui prisijungimo prie sistemos formos duomenis.
- Kuriant naujus įrašus arba koreguojant esamus į formą įvedamų duomenų validaciją atlikti realiu laiku (prieš išsiunčiant formos duomenis apdorojimui).
- Vykdamas įrašų paiešką pasiūlyti įrašų, rastų duomenų bazėje realiu laiku pagal vedamus paieškos simbolius, galimus variantus pasirinkimui.
- Momentinio kontakto buvimo vietos nustatymui ir atvaizdavimui pagal įvestą vietovę integruoti "Google Maps" sistemą.
- Atlikti momentinį naujienų sąrašo atnaujinimą realiu laiku.
- Integruoti ir naudoti skaitytuvus RSS naujienų skaitymui.
- Navigacijai tarp meniu punktų naudoti dalinį informacijos ataujinimą, neperkraudant viso puslapio.
- Naudoti momentinius pranešimus vartotojo lange, sistemoje įvykus naujiems svarbiems įvykiams ar pasikeitimams.

Šias rekomendacijas tikslinga taikyti ir kitoms įmonių taikomosioms programoms, kadangi RIA technologijos ne tik padidina sąveikos dinamiškumą, bet ir sumažina administravimo sąnaudas. Interneto taikomosios programos diegiamos serveryje, todėl administratorius gali jas tvarkyti vienoje vietoje vieną kartą. Ankstesnės interneto technologijos neturėjo pakankamo raiškumo ir dinamiškumo, todėl nebuvo tinkamos visoms organizacijų sistemoms. RIA technologija suteikia galimybes visą įmonės programinę įrangą realizuoti vieninga interneto technologija.

7. Išvados

1. Literatūros šaltinių ir esamų sistemų analizė parodė, kad galima pagerinti CRM sistemų kokybę pritaikant raiškaus interneto technologijas (RIA), kurios įgalina padidinti vartotojų sąveikos dinamiškumą ir priartina vartotojo interneto sąsajos galimybes prie darbatalio programų.
2. Technologinių RIA sprendimų analizės rezultate nustatyta, kad šiam tikslui geriausiai tinka JavaScript/DHTML AJAX įgyvendinimo metodas (Asinchroninis JavaScript + XML), kadangi metodas gerai pritampa prie egzistuojančių HTML Web taikomųjų programų, jį naudojant nereikia kurti sistemos iš naujo, o galima praplėsti esamą realizaciją. Be to šis metodas yra tinkamas DHTML/PHP programuotojams.
3. Pasirinktos CRM sistemos analizės parodė, kad tikslinga atlikti tokių jos funkcijų patobulimus: vartotojo prisijungimo prie sistemos duomenų korektiškumo ir teisingumo tikrinimą, ir formų duomenų validaciją realiu laiku; automatinį galimų rezultatų variantų pasiūlymą paieškos metu; momentinį kontakto buvimo vietos nustatymą ir atvaizdavimą integruotoje “Google Maps” sistemoje pagal įvestą vietovę; naujienų sąrašo atnaujinimą realiu laiku; RSS naujienų skaitytuvą; dalinį puslapių atnaujinimą naudojant administravimo meniu; momentinius pranešimus vartotojui apie gautas žinutes bei administratoriui – apie vartotojų neapmokėtas sąskaitas.
4. Sistemos išplėtimo realizacijos testavimas patvirtino, kad patobulinimai veikia patikimai, jokių žymesnių naujos dinaminės sistemos sąsajos sutrikimų neaptikta.
5. Specialistų atliktas devynių tipinių sistemos veikimo scenarijų įvertinimas pagal 6 dinamiškumo kriterijus parodė, kad bendras dinamiškųjų sistemos savybių vidutinis įvertinimas 10 balų skalėje yra lygus 9. Todėl galima teigti, jog patobulinta sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus.
6. Šiame darbe taikoma metodika leido sutaupyti 75 procentus darbo laiko kuriant dinamišką CRM sistemą. Tai yra pakankamai nebrangus būdas egzistuojančiai taikomajai programai suteikti daugiau dinamiškumo.
7. Atliktas tyrimas įgalino sudaryti rekomendacijas RIA technologijoms taikyti CRM ir kitoms įmonių taikomųjų programų sistemoms, tuo padidinat jų dinamiškumą ir palengvinant administravimą.
8. Remiantis šiuo darbu paruoštas straipsnis, kuris pateiktas konferencijai IT 2008.

8. Literatūra

- [1] What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software, by Tim O'Reilly 09/30/2005. <http://tim.oreilly.com/news/2005/09/30/what-is-web-20.html>
- [2] J. Allaire. Macromedia Flash MX-A next-generation rich client. Technical report, Macromedia, March 2002.
- [3] A Fragment-Based Approach for Efficiently Creating Dynamic Web Content, by Jim Challenger, Arun Iyengar, Karen Witting, Cameron Ferstat, and Paul Reed, which appeared in Proceedings of INFOCOM 2005, 2005 IEEE.
- [4] Ajax: A New Approach to Web Applications, by Jesse James Garrett, February 18, 2005. <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [5] An Open Source AJAX Comparison Matrix, by Edmon Begoli. December 5, 2006. <http://www.devx.com/AJAXRoundup/Article/33209>
- [6] The State of Web Frameworks. Craig R. McClanahan, Sun Microsystems, Inc., February 25, 2006. URL: http://kr.sun.com/developers/PDFs/preso/Craig_JCO2006.pdf
- [7] AJAX Design Strategies. By Ed Ort and Mark Basler, October 2006. URL: <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/DesignStrategies/index.html>
- [8] N. Davies, M. Davy, G. Blair, and J. Mariani, Object invocation and management in the distributed multimedia information system, Information and Software Technology vol. 35 nr. 5 March (1993)
- [9] B. Gaines and M. Shaw, WebMap: Concept Mapping on the Web, World Wide Web Journal, vol. 1 nr.1 (1995) p.171-183.
- [10] T. Bryant and A. Evans, OO Oversold - Those objects of obscure desire, Information and Software Technology, vol. 36 nr. 1 (1994) p.35-42.
- [11] M. J. Jacobson, C. Maouri, P. Mishra, and C. Kolar, Learning with Hypertext Learning Environments: Theory, Design, and Research, Journal of Educational Multimedia and Hypermedia, vol 4(4) (2002), p.321-364.
- [12] D. B. Lange, An Object Oriented Design Method for Hypermedia Information Systems, Proceedings of the 27th Annual Hawaii Int. conf. on System Sciences (1994) p.366-375.
- [13] J. Nanard and M. Nanard, Hypertext Design environments and the Hypertext Design Process, Communications of the ACM, vol. 38, nr. 8, (1995) p.49-56.
- [14] Wikipedia. Ajax (programming). Žiūrėta 2007-05-22. Prieiga per internetą <http://en.wikipedia.org/wiki/AJAX>
- [15] A. Verickas. Web 2.0 – naujoji interneto karta. // Kompiuterija. PC World. 2007 Sausio 15. Žiūrėta 2006-11-19. Prieiga per internetą <http://www.kompiuterija.lt/zurnale/2007-01-15/web-20-%E2%80%93-naujoji-interneto-karta/>
- [16] D. Saffer. The Web 2.0 Experience Continuum. // Adaptive Path. 2005 Lapkričio 14. Žiūrėta 2006-11-15. Prieiga per internetą <http://www.adaptivepath.com/publications/essays/archives/000545.php>

9. Terminų ir santrumpų žodynas

Modelis – realaus pasaulio objekto atvaizdavimas vienu ar kitu požiūriu.

Duomenų bazė (DB) – priemonė duomenims ir jų apdorojimo funkcijoms laikyti.

Forma – rinkinys html elementų duomenims susrinkti.

JavaScript – Scenarijų rašymo kalba, leidžianti dirbti su dokumento objektiniu modeliu (DOM).

Naršyklė – grafinė sąsaja, skirta vartotojams bendrauti su internetu.

Serveris – kompiuteris, kuriame laikomi failai bei programos ir suteikiama galimybė klientams kreiptis į failus, programas ir išteklius.

Hipertekstas (hypertext) – informacijos elementų organizavimas į tarpusavyje susietas asociacijas.

Hipertekstas buvo pirmasis konceptas kuriant pasaulinį tinklą (WWW).

Vartotojo interfeisas – ekrano aplinka, suteikianti vartotojui galimybę matyti ir valdyti taikomosios programos veiksmus.

CRM (Customer Relationship Management) – ryšių su klientais valdymas. Tai kompleksinė sistema, leidžianti įvertinti ir maksimizuoti kiekvieno bendrovės kliento ekonominę vertę bei taikyti veiksmingus vertingiausių klientų lojalumo skatinimo metodus.

RIA (Rich Internet Applications) - raiškaus interneto taikomosios programos.

RSS (Really Simple Syndication) - naujienų publikavimo kalba.

AJAX (Asynchronous Javascript And XML) – asinchroninis Javascript + XML. Tai tam tikrų technologijų sukomponavimas, suteikiantis interaktyvumo, dinamiškumo.

CSS – Cascading Style Sheet (internetinių stilių aprašymai).

HTML – hiperteksto žymėjimo kalba. DHTML – dinaminė hiperteksto žymėjimo kalba.

DOM (Document Object Model) – dokumentų objektų modelis. Tai taikomųjų programų programavimo sąsaja skirta galiojantiems HTML ir taisyklingiems XML dokumentams.

API (Application Programming Interface) - taikomųjų programų programavimo sąsaja (interfeisas).

XML (Extensible Markup Language) - tai „žymėjimo“ kalba dokumentams, kuriuose saugoma struktūrizuota tekstinė informacija.

W3C (World Wide Web Consortium) – tai pagrindinė pasaulinio kompiuterių tinklo tarptautinių standartų organizacija.

Flash – interaktyvios multimedijos kūrimo įrankis.

XUL – tai XML vartotojo interfeiso „žymėjimo“ kalba.

IS – informacinė sistema.

WWW - pasaulinis kompiuterių tinklas.

HTTP – hiperteksto perdavimo protokolas.

DBVS – duomenų bazių valdymo sistema.

SQL – struktūrizuota užklausų kalba.

CGI – scenarijų rašymo technologija.

Priedai

Priedas 1. Straipsnis, paruoštas „IT 2008“ konferencijai.

DINAMINĖ INTERNETO SAŠAJA ĮMONIŲ KLIENTŲ-RYŠIŲ VALDYMO SISTEMOMS PATOBULINTI

Aurimas Juncis, Lina Nemuraitė

Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų g. 50, Kaunas

Straipsnyje nagrinėjamas antros kartos (Web 2.0) interneto atsiradimas, jam būdingos naujos technologijos ir pasikeitę reikalavimus interneto sistemoms. Pastebimas augantis poreikis darbui su dideliais duomenų kiekiais, naudojant paprastą naršyklę. Iškeliamas tikslas padidinti CRM sistemų dinamiškumą, išplečiant jas raiškių interneto programų savybėmis ir sudarant rekomendacijas esamoms sistemoms patobulinti. Detaliau aptariamas technologijų rinkinys, pavadintas Ajax (Asynchronous JavaScript and XML), kuris leidžia kurti dinamiškus interneto puslapius (sistemas). Pateikiamas dinamiškos sistemos scenarijų apibrėžimas ir sudaromos RIA taikomųjų programų savybių pritaikymo CRM ir kitoms organizacijų sistemoms rekomendacijos

1 Įvadas

Efektyvus klientų valdymas – dar sunkiai suprantama sąvoka dideliai daliai Lietuvos verslo atstovų. Ryšių su klientais valdymo sistemų, dar kitaip vadinamų CRM, diegimas kelia įmonės konkurencinį pranašumą rinkoje, leidžia didinti pardavimus ir įmonės pelną. Viena iš karščiausių tendencijų elektroninėje komercijoje ir mažmeninėje prekyboje yra Web 2.0 taikomųjų programų adaptacija ryšių su klientais valdymo procese. Vienos vairių dydžių ir sferų įmonės tai paprastai vadina protinga el. komercija, o kitos savo ruožtu įvardija tai kaip CRM 2.0. Daugelis pasaulyje lyderiaujančių organizacijų diegia Web 2.0 taikomąsias programas savo klientų-ryšių valdymo sistemose tam, kad klientams bedradarbiavimas būtų patogesnis, greitesnis, dinamiškesnis ir jie pasijustų esantys esminiais elementais visame verslo ryšyje.

Pastaruoju metu Web evoliucijoje vyksta gilūs pasikeitimai. Mes žengiame į neturintį precedento vartotojų interfeiso inovacijų periodą, kai Web programuotojai galiausiai gali sukurti tokio lygio Web taikomąsias programas, kokios egzistuoja lokaliuose darbastaliuose. Modernios Web 2.0 taikomosios programos vis labiau populiarėja įmonių tarpe, nors nebūtinai visi supranta, kas tai iš tikrųjų yra, todėl šia tema vyksta daug diskusijų. WEB 2.0 pirmą kartą paminėjo O'Reilly Media [1] 2005 metais, apibūdindamas kaip antrosios kartos internetu paremtas paslaugas, kurias sudaro visuomeninio tinklo svetainės, "wikis", komunikaciniai įrankiai ir dar daugybė kitų priemonių leidžiančių žmonėms bendrauti ir dalintis informacija internete anksčiau nežinomais būdais. Seniau internetas buvo išrinktųjų teisė, dabar jis visų, kiekvienas gali kurti interneto turinį, naujienas, prisidėti prie enciklopedijų kūrimo. Visas šias galimybes suteikia moderniosios Web 2.0 taikomosios programos.

Todėl šio darbo tyrimo yra raiškaus interneto taikomųjų programų (RIA) kūrimo technologijos ir jų taikymas tobulinant organizacijų CRM informacines sistemas. Tyrimo objektas – interneto vartotojo interfeiso kūrimo procesas, leidžiantis pagerinti dinamines vartotojo sąveikos su interneto sistema savybes.

Darbe siekiama pritaikyti RIA taikomųjų programų savybes CRM sistemai, suteikiant vartotojų interfeisui didesnę dinamiškumą ir naudojimo patogumą. Tam tikslui bus tiriamos galimybės pritaikyti analizės metu nustatytus RIA taikomųjų programų privalumus. CRM sistemoje bus įdiegtos raiškaus interneto taikomosios programos, kurių pagrindinis uždavinys – parodyti Web 2.0 taikomųjų programų pritaikymo galimybes įmonės vartotojo interfeiso tobulinime.

2 Dinamiškų sistemų kūrimo poreikis ir technologijos

Šiame skyriuje aprašomas antros kartos internetas, jo atnešti nauji poreikiai interneto sistemoms ir pagrindinės technologijos, kurios bus reikalingos kuriant dinamiškas sistemas.

2.1 Antros kartos internetas (Web 2.0)

Timas O'reilis savo straipsnyje [1] pabrėžia, kad naujos kartos interneto kompanijomis galima vadinti tik tas, kurios tenkina šias sąlygas:

- siūlo paslaugas (o ne supakuotą programinę įrangą) su nuo kainos priklausančiu plečiamumu;
- kontroliuoja unikalių, sunkiai atkuriamų duomenų šaltinius, kurie auga priklausomai nuo to, kiek žmonių juos naudoja;
- pasitiki naudotojais kaip pagalbinais programinės įrangos vystytojais;
- išnaudoja kolektyvinį protą;
- didina konkurenciją suteikdami galimybę klientams kurti savo paslaugas;
- kuria programinę įrangą, kuri neapsiriboja vienu prietaisu;
- lengvina naudotojo interfeisus, vystymo modelius ir verslo modelius.

AJAX yra vienas pagrindinių Web 2.0 taikomųjų programų komponentas tokiuose projektuose kaip *Flickr*, dalis *Yahoo!*, taip pat Google taikomoseose programose – *Gmail*, *Orkut*, *Google Maps*, *Google Docs* ir daugelyje kitų. AJAX

reiškia Asynchronous Javascript And XML. Iš tiesų, AJAX vadinama Web 2.0 (naujojo interneto) dalimi. Nors tai ir nėra technologija, tai greičiau tam tikrų technologijų sukomponavimas, tačiau suteikiantis tam tikrų galių. AJAX leidžia mums atlikti elementarų CSS ir XHTML panaudojimą, dinaminį duomenų vaizdavimą ir jų ryšius su DOM, manipuliavimą ir keitimąsi duomenimis su XMLHttpRequest, ir visa tai sujungia Javascript.

Tradiciniai Web tinklapiai veikia taip: vartotojas savo veiksmais išsiunčia HTTP serveriui užklausa, serveris atlieka veiksmus - paima duomenis, juos apdoroja, praeina pro kelias sistemas, ir tada gražina sugeneruotą Web puslapį vartotojui. Tokia yra Web specifika, tačiau nesunku suprasti, jog tokia schema ne visai tinkama programinės įrangos kūrimui. Kol serveris atlieka savo darbus, vartotojas laukia. Kiekviename žingsnyje jis turi laukti, kol užklausa nukeliauja, serveris atlieka skaičiavimus ir gražina tai, ką reikia. Be abejo, šiandien kuriant Web nuo pagrindų, vartotojas neturėtų būti verčiamas laukti. Kodėl viskas turi būti kraunama iš naujo, jei vartotojo interfeisas užkrautas ir jis realiai nesikeičia? O kodėl išvis vartotojui rodyti, jog siunčiamos užklauskos ar gaunami duomenys?

AJAX schemeje tarp vartotojo ir serverio įterpiamas AJAX varikliukas, kuris ir apsiima išspręsti šias problemas. AJAX varikliukas paprastai užkraunamas Javascript, kuris atsakingas ir už vartotojo interfeiso apdorojimą, ir komunikavimą su serveriu. Vartotojo bendravimas su serveriu tampa nesinchroniškas – tai yra, nepriklauso nuo ryšio su serveriu. Vartotojas nebemato tuščio lango, laukdamas, kol užsikraus puslapis. Kiekvienas vartotojo, kuris paprastai atlieka HTTP užklausa, veiksmas kreipiasi į AJAX varikliuko funkciją. Veiksmai, kurie nereikalauja susijungimo su serveriu, įvykdomi iškart varikliuko lygmenyje. Jei veiksmas reikalauja susijungimo su serveriu, užklauskos atliekamos asinchroniškai, dažniausiai naudojant XML, ir neįtraukiant į tai vartotojo.

2.2 Raiškaus interneto taikomosios programos (RIA)

Terminas „raiškaus interneto taikomosios programos“ RIA (Rich Internet Applications) buvo pristatytas Macromedia šaltinyje [2] 2002 metais tam, kad pažymėti tradicinio darbaltalio ir Web taikomųjų programų susivienijimą, stengiantis pasverti abiejų architektūrų privalumus ir iškylančius trūkumus.

Lentelėje (1 lentelė) pateiktas darbaltalio (įskaitant kliento serverį) ir bendrai priimtų HTML+HTTP Web taikomųjų programų palyginimas, išryškinant potencialą raiškaus interneto taikomųjų programų, kurios paremtos Web skirstymo modeliu, siūlo patobulintus interfeisus ir sumažina komunikacijos išlaidas.

1 lentelė. Darbaltalio ir bendrai priimtų HTML+HTTP Web taikomųjų programų palyginimas

Savybė	C / S, Darbaltalis	Web	RIA
Universalaus kliento panaudojimas (naršyklė)	Ne	Taip	Taip
Paprastas kliento įdiegimas	Ne	Taip	Taip
Raiškios sąveikos galimybės	Taip	Ribotos	Taip
Serverio pusės biznio logika	Taip	Taip	Taip
Kliento pusės biznio logika	Taip	Ribota	Taip
Dalinis puslapio atnaujinimas (refresh)	Taip	Ne	Taip
Retos serverio iteracijos (round-trips)	Taip	Ne	Taip
Serveris-klientas komunikavimas	Taip	Ne	Taip
Funkcionavimas atsijungus	Taip	Ne	Taip
Navigavimas klaviatūra	Taip	Taip	Taip

Pavaizduoti RIA privalumai yra tinkami daugeliui užduočių, tokių kaip duomenų rūšiavimas, filtravimas, sudėtingas vaizdų apdorojimas ir t.t. Paprastai klientas užkrauna raiškaus interneto taikomąją programą kartu su tam tikrais pradiniais duomenimis, tuomet ji, komunikuodama su serveriu tik tuomet, kai vartotojui prireikia sekančios informacijos arba kai jis turi išsiųsti duomenis, valdo duomenų interpretavimą ir įvykių apdorojimą. Yra siūloma keletas technologijų RIA kūrimui. Nors ir yra skirtingos daugeliu požymių, tačiau jos gali būti klasifikuojamos į keturias plačias kategorijas:

- Skriptais grindžiamos (Scripting-based), kur kliento pusės logika yra įgyvendinama naudojantis skriptais grindžiamomis kalbomis, tokiomis kaip JavaScript, ir interfeisai yra pagrįsti HTML ir CSS kombinacija;
- Įskiepiais pagrįstos (Plugin-based), kur pažangus duomenų pateikimas (rendering) ir įvykių apdorojimas yra leidžiamas naršyklės įskiepių interpretuojant XML, bendrų tikslų programas ar media failus (Flash, Flex, Laszlo, Xamlon);
- Naršyklė pagrįstos, kur raiški sąveika yra iš prigimties palaikoma kai kurių naršyklių, kurios interpretuoja deklaratyviais interfeiso apibrėžimo kalbas (XUL);
- Web pagrįstos darbaltalio technologijos, kur taikomosios programos yra parsiumčiamos iš Web, bet įvykdomos naršyklės išorėje (Java Web Start, Window Smart Client).

Įprastas savybių rinkinys gali būti aptariamas charakterizuojant RIA metodą conceptualaus modeliavimo ir kodo generavimo tikslais:

- RIA turėtų palaikyti kliento pusės apdorojimą ir sumažinti komunikaciją su serveriu iki minimumo;
- Taikomųjų programų duomenys galėtų būti saugomi skirtingo atkaklumo lygiuose tiek kliento, tiek serverio pusėje;

- Duomenų apdorojimas (pvz.: sukūrimas, redagavimas, filtravimas) galėtų įvykti tiek kliento, tiek serverio pusėje;
- Turėtų būti galimas tiek „online“, tiek „offline“ naudojimas.

2.3 AJAX (Asinchroninis JavaScript ir XML)

Šiame skyrelyje aprašomas pagrindinis dinamiškos sistemos technologijų rinkinys, pavadintas Ajax. Realiai jokios tikros dinamiškos sistemos negali nenaudoti Ajax, nes tai pagrindinė priemonė pagreitinti serverio ir kliento bendravimą.

2.3.1 AJAX įgyvendinimo metodai

Siekiant svetainėje išgauti kuo daugiau dinamiškumo yra dažniausiai naudojamas AJAX [4]. Yra keletas skirtingų AJAX kliento pusės varikliuko kūrimo technologijų, iš kurių JavaScript, Java ir Flash yra naudojamos dažniausiai:

- Java metodas (Asinchroninis Java + XML) – šis metodas paprastai naudoja naršykle pagrįstą lengvo svorio Java varikliuką kliento pusės apdorojimui, tokiam kaip vartotojo interfeiso pateikimas (rendering), dalinio ekrano atnaujinimo atlikimas ir asinchroninis serverio komunikavimas. Šiame metode vartotojo interfeisas gali būti apibrėžtas naudojant XML, tuo tarpu kliento pusės taikomosios programos logika yra suprogramuojama naudojant standartinę Java.

- JavaScript/DHTML metodas (Asinchroninis JavaScript + XML) - šis metodas paprastai naudoja naršykle pagrįstą lengvo svorio JavaScript biblioteką kliento pusės apdorojimui, tokiam kaip dalinio ekrano atnaujinimo atlikimas ir asinchroninis foninis tinklo komunikavimas. Šiame metode vartotojo interfeisas paprastai yra apibrėžiamas naudojant DHTML, o kliento pusės taikomosios programos logika yra suprogramuojama naudojant JavaScript.

- Flash metodas (Asinchroninis ActionScript + SWF) - šis metodas kliento pusės apdorojimui paprastai naudoja naršykle pagrįstą Flash varikliuką su ActionScript biblioteka. Šiame metode vartotojo interfeisas paprastai apibrėžiamas naudojant SWF, tuo tarpu kliento pusės taikomosios programos logika yra suprogramuojama naudojant ActionScript.

Kiekvienas iš šių metodų turi savo pliusų ir minusų, dėl kurių jie pritaikomi skirtingoms taikomosioms programoms. Lentelėje 2 pavaizduotos metodų stiprybės ir silpnybės.

2 lentelė. AJAX įgyvendinimo metodų palyginimas

Metodai	Privalumai	Trūkumai
Java	<ul style="list-style-type: none"> • Didelis našumas ir patikimumas • Lengvas palaikymas dėl objektiškai orientuoto programavimo • Tinkamas Java programuotojams 	<ul style="list-style-type: none"> • Reikalauja Java programavimo žinių • Egzistuojančią CRM sistemą reiktų kurti iš naujo
DHTML/JavaScript	<ul style="list-style-type: none"> • Gerai pritampa prie egzistuojančių HTML Web svetainių/Web taikomųjų programų • Tinkamas DHTML Web programuotojams • Nebrangus būdas egzistuojančiai taikomajai programai suteikti daugiau dinamiškumo 	<ul style="list-style-type: none"> • Šis metodas gali būti problematiškas didelės aprėpties biznio taikomose programose dėl JavaScript programavimo ir palaikymo sunkumų
Flash	<ul style="list-style-type: none"> • Patrauklios animacijos • Tinkamas grafikos projektuotojams 	<ul style="list-style-type: none"> • ActionScript kodas yra sunkiai palaikomas ir nėra suprojektuotas komandiniam darbui • Veikimo ir funkcionalumo apribojimai • Egzistuojančią CRM sistemą reiktų kurti iš naujo

Java metodas pasižymi geru funkcionalumu dėl binarinio vykdymo, multi siūlinio (multi-threaded) skaičiavimo ir Java virtualios mašinos industrinės stiprybės. Taikomosios programos, parašytos Java kalba, yra lengvai kuriamos ir palaikomos dėl objektiškai orientuotos programavimo prigimties. Šis metodas labiausiai tinka programuotojams, turintiems Java programavimo įgūdžių.

DHTML/JavaScript metodas labai gerai tinka egzistuojančiose HTML Web taikomose programose. HTML programuotojui yra lengviau programuoti su JavaScript ir tai yra ne toks brangus būdas egzistuojančiai taikomajai programai suteikti daugiau interaktyvumo. Paprastai šis metodas reikalauja parsisiųsti JavaScript bibliotekų, tam kad kompensuoti skirtingas kliento naršyklės galimybes. Gauta taikomoji programa bus tiek raiški, kiek tai leis naršyklė. Šis metodas gali būti problematiškas didelės aprėpties biznio taikomose programose dėl JavaScript programavimo ir palaikymo sunkumų, bei DHTML funkcionalumo apribojimų.

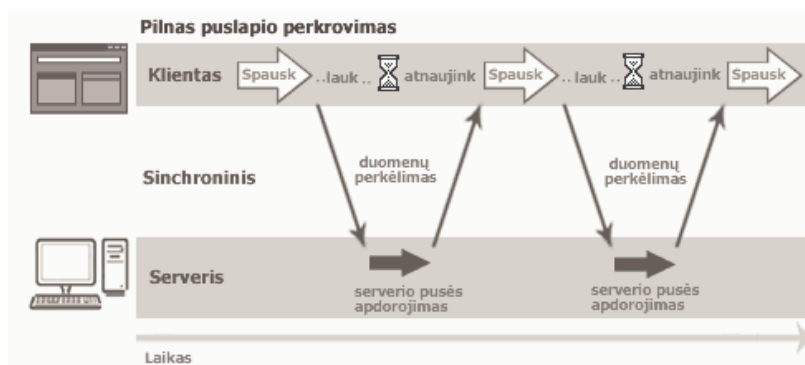
Flash metodas yra geras taikomosioms programoms, kurioms reikia patrauklaus vaizdo ir pojūčio. Tai natūralus pasirinkimas grafikos dizaineriams. Flash programuotojams suteikia galimybę kurti labai raiškų vartotojo interfeisą. Šiuo metu, Macromedia yra nusitaikiusi į tris Flash panaudojimo atvejus: valdomas pardavimas, valdomos formos ir pradinė vartotojo apdaila arba duomenų vizualizavimo taikomosios programos. Flash taikomosios programos gali pasiekti Java ir

.NET objektus, serveryje veikiančius asinchroniškai. Pagrindiniai šio metodo apribojimai yra ActionScript ir Flash varikliuko ribotos galimybės.

Pasirinktas JavaScript/DHTML AJAX įgyvendinimo metodas (Asinchroninis JavaScript + XML) dėl to, kad gerai pritaipama prie egzistuojančių HTML Web taikomųjų programų (automatizuota pirkėjų aptarnavimo sistema realizuota šiuo pagrindu) bei yra tinkamas DHTML Web programuotojams.

2.3.2 Architektūra, veikimo principai

Pagal savo kilmę Web buvo sukurtas HTML dokumentų skaitymui. Klasikinės Web taikomosios programos modelis yra perėmęs “spausk, lauk ir atnaujink” vartotojo interakcijos paradigmą ir sinchroninės “užklausa/atsakymas” komunikacijos mechanizmą (1 pav):



1 pav. Klasikinis Web taikomųjų programų modelis

- “Spausk, lauk ir atnaujink” paradigma: naršyklė atsako į vartotojo veiksmą atmesdama esamą HTML puslapį ir išsiųsdama HTTP užklausą atgal Web serveriui. Serveris užklausą apdoroja ir grąžina naršyklei naują HTML puslapio turinį. Naršyklė atnaujina ekraną ir parodo naują HTML puslapį.

- Sinchroninės „užklausa/atsakymas“ komunikacijos modelis: naršyklė visada inicijuoja užklausas, tuo tarpu Web serveris paprasčiausiai atsako į šias naršyklės užklausas. Web serveris niekada neinicijuoja užklausų – komunikacija visuomet inicijuojama viena kryptimi. „Užklausa/atsakymas“ yra sinchroninis ciklas, kurio metu vartotojas neprivalo laukti.

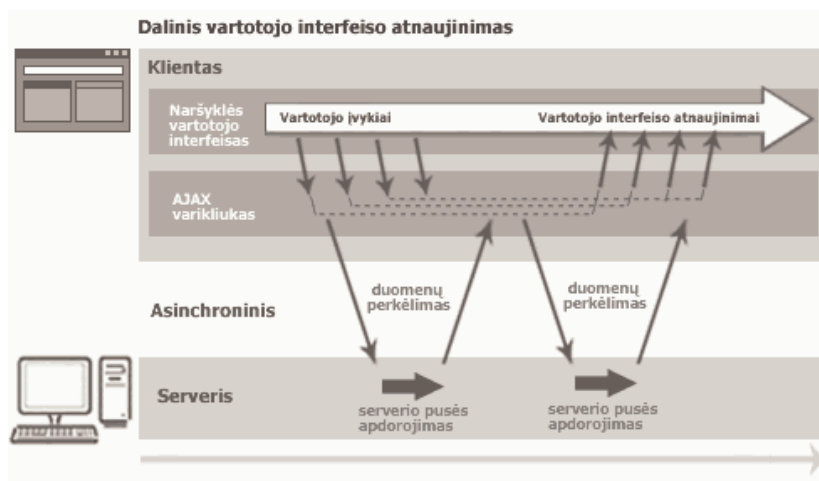
Tačiau šios dvi pagrindinės klasikinio Web modelio elgsenos yra netinkamos programinės įrangos taikomosioms programoms. Šiame kontekste klasikinis Web taikomųjų programų modelis sukelia daug problemų:

- lėtas vykdymas dėl “spausk, lauk ir atnaujink”;
- operacijos konteksto praradimas puslapio atnaujinimo metu;
- pernelyg didelė serverio apkrova ir pralaidumo suvartojimas dėl bereikalingų puslapio atnaujinimų;
- trūksta dviejų kryptių, realaus laiko komunikacijos sugebėjimų serverio inicijuotiems atnaujinimams atlikti.

Viso to rezultate turime lėtas, nepatikimas, mažo produktyvumo ir neefektyvias Web taikomąsias programas. Tam, kad pasiekti geresnes charakteristikas, daugiau interaktyvumo ir efektyvesnes Web taikomąsias programas, naudojamas AJAX taikomųjų programų modelis (2 pav):

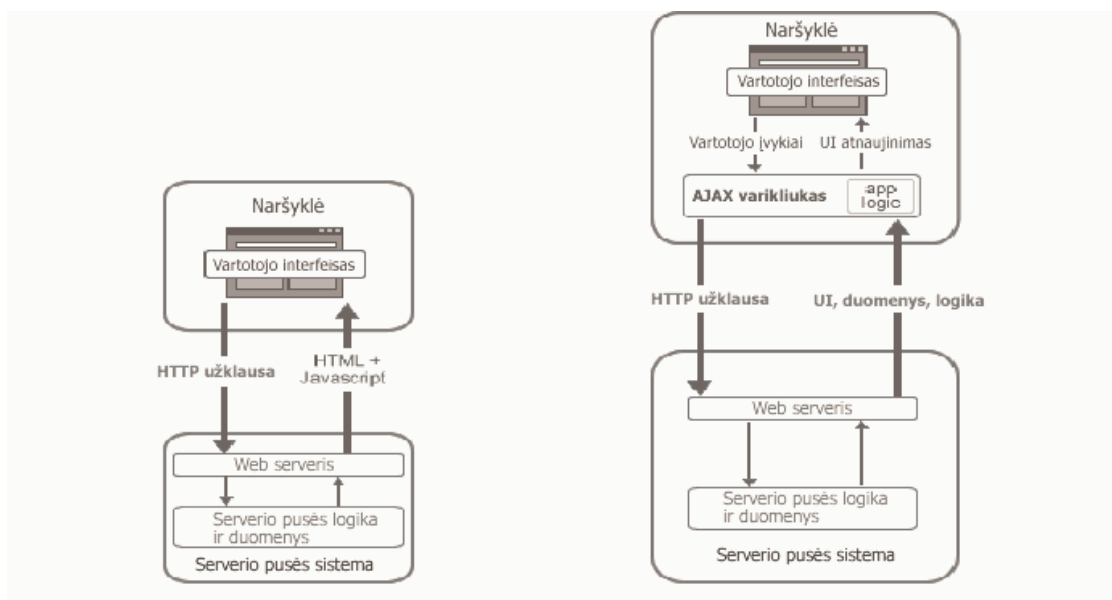
- “Dalinis ekrano atnaujinimas” pakeičia “spausk, lauk ir atnaujink” modelį: vykstant vartotojo interakcijoms AJAX pagrįstose taikomosiuose programose, atnaujinami tik naują informaciją turintys vartotojo interfeiso elementai. Likę interfeiso elementai išlieka nepakitę. Šis “dalinio ekrano atnaujinimo” sąveikos modelis ne tik įgalina nenutrūkstamo veikimo kontekstą, bet taip pat padaro įmanoma netiesinę darbų seką.

- Asinchroninė komunikacija pakeičia sinchroninę “užklausa/atsakymas” modelį: AJAX pagrįstose taikomosiuose programose “užklausa/atsakymas” gali būti asinchroninis, atskiriant vartotojo ir serverio interakcijas. To pasėkoje, vartotojas gali toliau naudoti taikomąją programą be pertraukimų, kol kliento programa užklausia informacijos iš serverio. Kai iš serverio gaunama nauja informacija, atnaujinamos tik su šia informacija susijusios vartotojo interfeiso dalys.



2 pav. AJAX pagrįstas Web taikomųjų programų veikimo modelis

Klasikinio Web ir AJAX architektūrų palyginimui pateiktas 3 paveikslas.



3 pav. Klasikinio Web ir AJAX architektūrų palyginimas

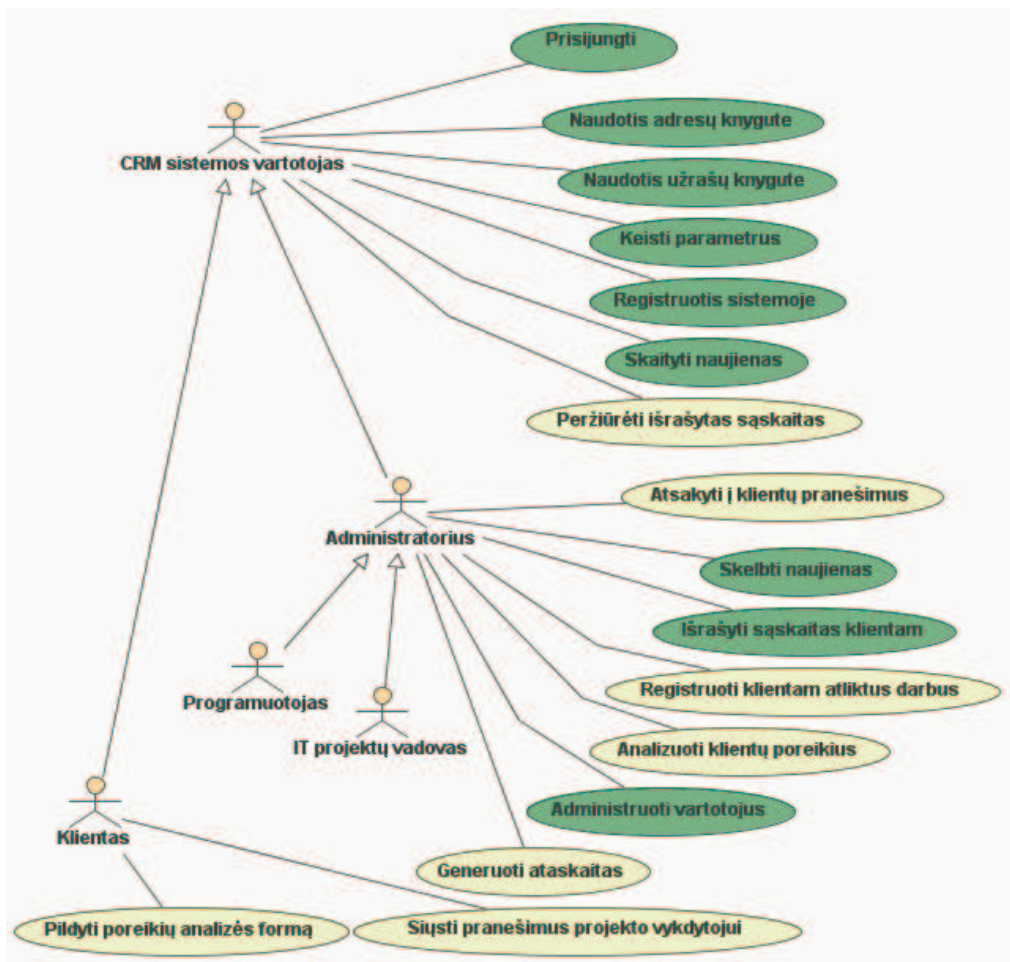
Programinės įrangos architektūros požiūriu AJAX skiriasi nuo klasikinės Web taikomųjų programų architektūros, atsiradus kliento pusės varikliukui, šiais atžvilgiais:

- Kliento pusės varikliukas naudojamas kaip tarpininkas tarp vartotojo interfeiso ir serverio;
- Vartotojo veikla priveda prie programų kreipimosi į kliento pusės varikliuką, vietoj puslapio užklauso tiesiai iš serverio;
- XML duomenų perdavimas tarp serverio ir kliento pusės varikliuko.

Kliento pusės varikliukas yra AJAX modelio pagrindas. Be šio varikliuko, kiekvienas vartotojo įvykis turi grįžti atgal į serverį apdorojimui. Vartotojo interakcija yra tvirtai surišta su serverio komunikacijomis - kliento pusės varikliukas panaikina šią priklausomybę. Šis varikliukas, dirbdamas Web naršyklėje, suteikia naršyklei papildomų sugebėjimų, kad atlikti „dalinį ekrano atnaujinimą“ vietoje „pilno puslapio atnaujinimo“.

2.4 Tobulinamos CRM sistemos panaudojimo atvejai

4 paveikslėlyje pavaizduotas kompiuterizuojamos sistemos panaudojimo atvejų modelis. Žalia spalva pažymėti panaudojimo atvejai rodo tobulinamas CRM sistemos funkcijas.



3 pav. Kompiuterizuojamos sistemos panaudojimo atvejų modelis

Esamos CRM sistemos trūkumai ir sprendimai, pašalinantys nustatytus trūkumus, pateikti 3 lentelėje.

3 lentelė. Esamos sistemos trūkumai ir jų šalinimas

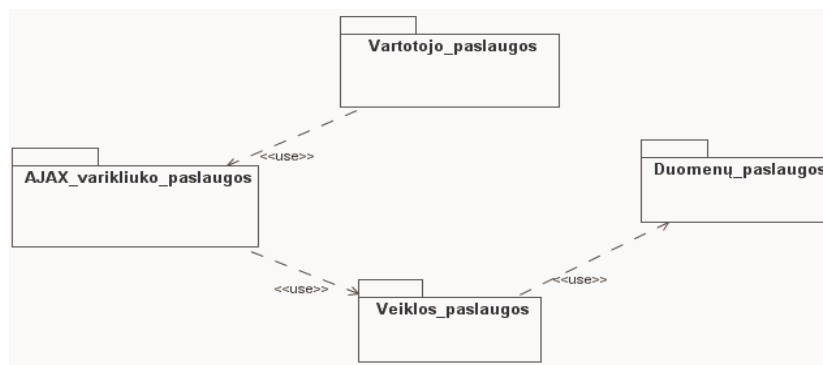
Trūkumas	Sprendimas
Nepatogus formos duomenų validavimas	Vartotojo prisijungimo prie sistemos duomenų tikrinimas ir formų duomenų validavimas realiu laiku
Neefektyvus įrašų paieškos vykdymo procesas	Automatinis galimų rezultatų variantų pasiūlymas paieškos metu
Nėra galimybės nustatyti kontakto buvimo vietos	Momentinis kontakto buvimo vietos nustatymas ir atvaizdavimas integruotoje "Google Maps" sistemoje pagal įvestą vietovę
Nevisuomet pastebimos paskutinės naujienos	Naujienų sąrašo atnaujinimas realiu laiku.
Nėra RSS naujienų skaitytuvo	Integruotas skaitytuvas RSS naujienų skaitymui.
Naviguojant per meniu reikia laukti kol persikrauna visa puslapio informacija	Dalinis informacijos atnaujinimas, neperkraunant viso puslapio navigavimo metu.
Nepatogu komunikuoti su klientais ir sekti sąskaitų apmokėjimą	Momentinis pranešimų pateikimas vartotojui, sistemoje įvykus naujiems svarbiems įvykiams ar pasikeitimams.

3 CRM sistemos tobulinimo projektas

Šiame skyriuje yra išdėstoma sistemos įgyvendinimui pasirinkto metodo pagrindinė esmė. Taip pat pavaizduojama sistemos architektūra, sudaromas sistemos elgsenos modelis.

3.1 Loginė sistemos architektūra

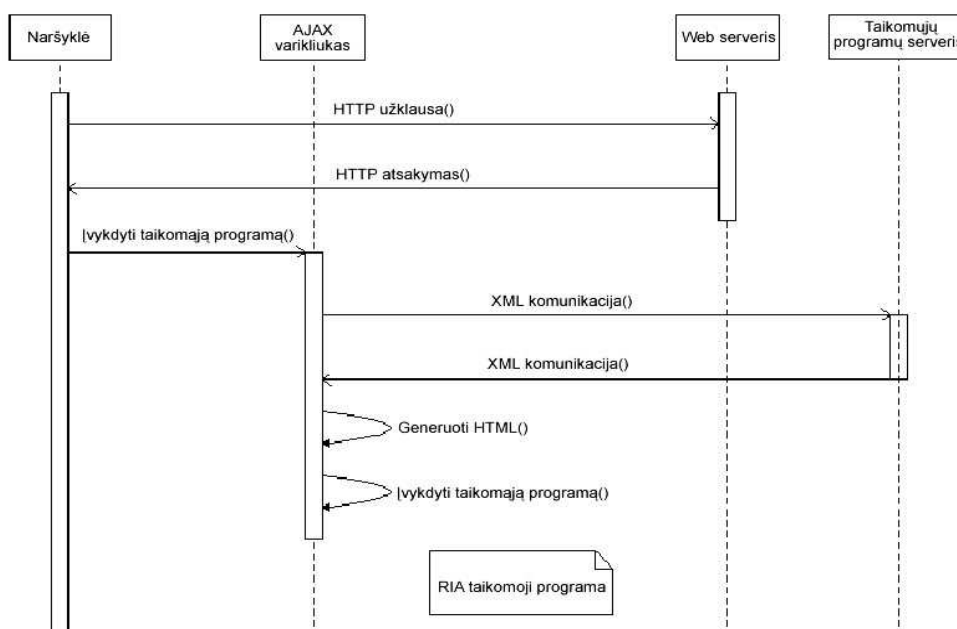
Sistemos architektūros kūrimui panaudotas tipinis trijų lygių architektūros modelis, kuris papildomas AJAX varikliuko paslaugomis (4 pav). Remiantis šiuo modeliu kuriamą sistemą sudaro vartotojo, veiklos bei duomenų paslaugos, kurios sugrupuojamos į atskirus paketus. Tarp vartotojo ir veiklos paslaugų įsiterpia AJAX varikliuko paslaugos.



4 pav. Loginis sistemos architektūros modelis

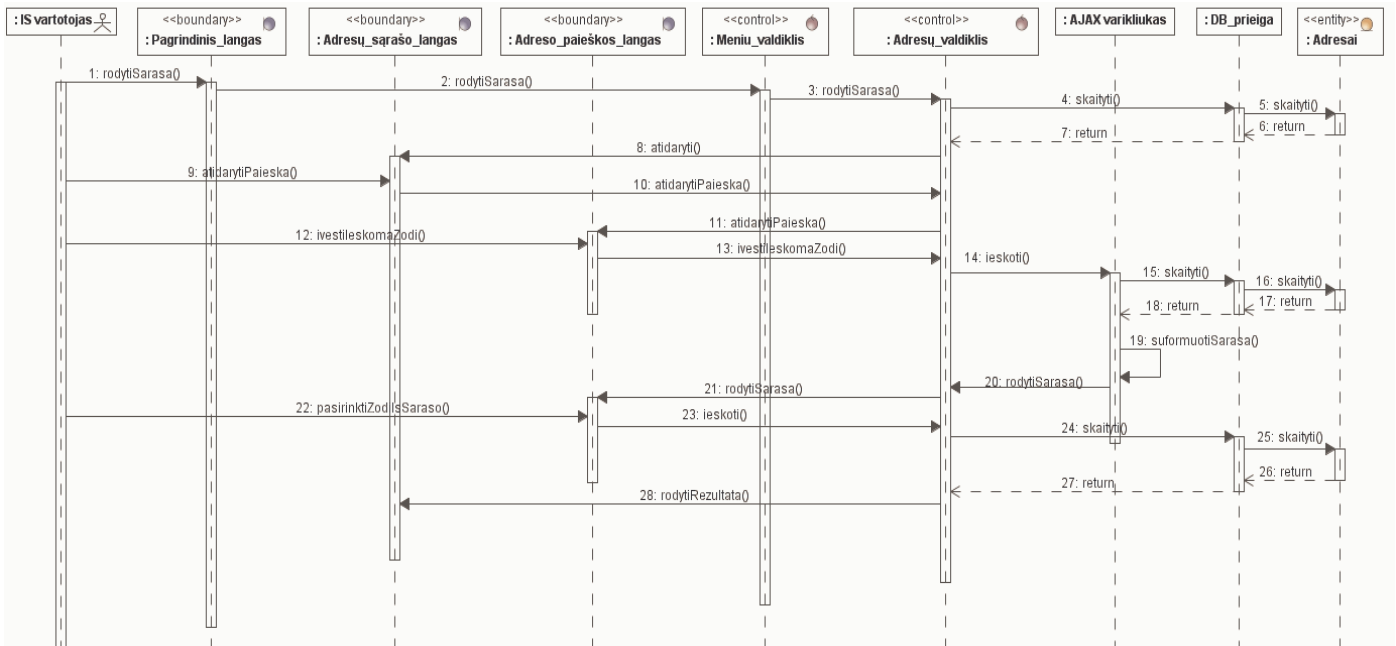
3.2 Sistemos elgsenos modelis

5 paveikslėlyje pateikiama principinė RIA taikomųjų programų sekų diagrama, parodanti kaip vyksta užklauso/atsakymo dialogas tarp naršyklės ir serverio. Iš diagramos galima pamatyti, jog RIA taikomųjų programų sekų diagramoje atsiranda papildomas elementas – AJAX varikliukas, suteikiantis taikomajai programai papildomą funkcionalumą, kurio naršyklė nepalaiko. Antra – pastebimas asinchroninis komunikavimo procesas tarp naršyklės ir serverio, dėl ko nereikia perkrauti viso naršyklės turinio.



5 pav. Principinė RIA taikomųjų programų sekų diagrama

Konkrečiai realizuotas kontakto paieškos vykdymas pavaizduotas 6 paveikslėlyje. Pasirinkus meniu punktą „Adresų knygelė“ iš duomenų bazės nuskaitomas sąrašas vartotojo susikurtų kontaktų. Pasirenkamas kontakto paieškos punktas ir atidaroma paieškos forma. Pasirenkamas paieškos kriterijus ir vedamas paieškos žodis. Realiu laiku vedant paieškos žodį Ajax varikliukas kreipiasi į duomenų bazę ir ieško įvedamų simbolių atitikmenų. Jei randami tam tikri įrašai, tai jų sąrašas akimirksniu pateikiamas vartotojo lange. Vartotojui pasirinkus pageidaujama įrašą ir patvirtinus paiešką išvedamas pageidautas kontaktas.



6 pav. Paieškos vykdymo sekų diagrama

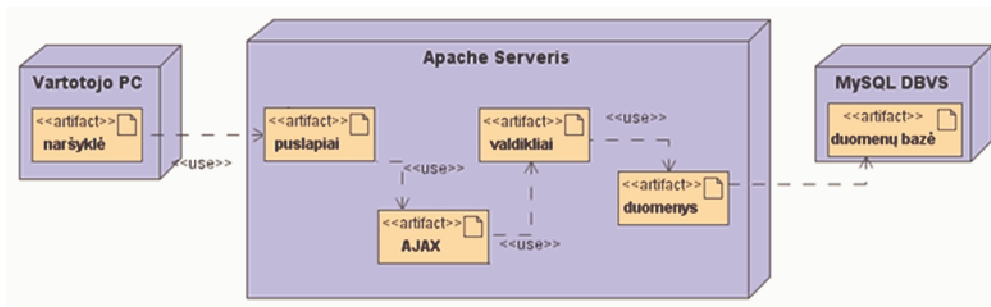
3.3 Sistemos įdiegimo modelis

CRM informacinė sistema padalinama į fizinius komponentus bei nurodoma sąveika tarp jų. Kiekvieno komponento realizacijos rezultatas yra artefaktas, kuris bus įdiegiamas fiziniuose įrenginiuose. Kadangi sistema realizuojama *PHP* programavimo kalba, *HTML* puslapiai sudaro vieną komponentą, kurį realizuojame artefaktu *puslapiai*. O veiklos logika realizuojama *PHP* klasėmis, kurios sudaro artefaktą *valdikliai*. Dinamiškos sistemos komunikavimas vyksta netiesiogiai su *PHP* klasėmis, o per *AJAX* komponentą. Kai gaunama užklausa iš naršyklės, ji per *HTML* puslapius (*puslapiai*) pereina į *AJAX* komponentą, o ne į *valdikliai*. Taip realizuojama *AJAX* varikliuko funkcija. Nedinamiškos funkcijos vykdomos kreipiantis tiesiogiai į *valdikliai*.

CRM sistemos įdiegimo diagrama (7 pav.) rodo fizinių techninių įrenginių išdėstymą bei vykdomųjų programų komponentų paskirstymą juose.

Informacinės sistemos įdiegimo diagrama susideda iš trijų dalių: vartotojo personalinio kompiuterio su interneto naršykle, *Apache* serverio, kuriame vykdomas vartotojo sąsajos generavimas bei veiklos logika, ir *MySQL* duomenų bazės, kurioje saugomi informacinės sistemos duomenys.

Vartotojas naudodamasis personaliniu kompiuteriu ir jame įdiegta programine įranga (Interneto naršykle) prisijungia prie informacinės sistemos, kuri yra operuojama *Apache* serverio pagalba, atlieka nustatytus veiksmus, o sistema išsaugo pakeitimus *MySQL* duomenų bazėje.



7 pav. Sistemos įdiegimo diagrama

4 Dinamiškų sistemos savybių tyrimas ir įvertinimas

Dinamiškų sistemos savybių tyrimui ir įvertinimui pasirinktos kelios esminės RIA taikomosioms programoms būdingos savybės: 1) raiškios sąveikos galimybės; 2) kliento pusės biznio logika; 3) dalinis puslapio atnaujinimas; 4) retos serverio iteracijos; 5) spartus veikimas; 6) patogumas/aiškumas naudoti.

Bandysime nustatyti kiek sistemoje yra dinamiškų savybių, kurios taip pat rodytų sistemos dinamiškumo lygį. Vertinimo skalę pasirinkta nuo 0 iki 10.

Dinamiškumo pasiekimo nagrinėjimui sukuriame konkretūs sistemos atliekamų veiksmų scenarijai (jiems bus įvertintos aukščiau nagrinėtos savybės):

- Scenarijus1:** Vartotojo prisijungimo prie sistemos duomenų korektiškumo ir teisingumo tikrinimas realiu laiku.
- Scenarijus2:** Formos duomenų validacija realiu laiku (prieš išsiunčiant formas duomenis), kuriant naują įrašą.
- Scenarijus3:** Įrašų paieškos vykdymas. Vedant paieškos žodį sistema automatiškai pasiūlo galimus variantus, dar prieš išsiunčiant formas duomenis.
- Scenarijus4:** Momentinis kontakto buvimo vietos nustatymas ir atvaizdavimas integruotoje “Google Maps” sistemoje pagal įvestą vietovę.
- Scenarijus5:** Naujienų sąrašo atnaujinimas realiu laiku, administratoriui sukūrus naują arba pakoregavus naujieną.
- Scenarijus6:** RSS naujienų skaitymas.
- Scenarijus7:** Vartotojų administravimo meniu navigavimas, neperkraunant visos puslapio informacijos.
- Scenarijus8:** Momentinis pranešimas vartotojo lange, gavus naują žinutę iš kito vartotojo.
- Scenarijus9:** Momentinis pranešimas administratoriaus lange apie vartotojų neapmokėtas sąskaitas.

Eksperimentus atliko ir įvertinimus skyrė trys įmonėje, kurios CRM sistema buvo tobulinama, dirbantys darbuotojai – ekspertai: Projektų vadovas: Andrius Kučiauskas; Programuotojas: Justas Adžgauskas; Projektų vadovas – programuotojas: Aurimas Junčis.

Priimsime, jog sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus, jei bendras dinamiškųjų sistemos savybių vidutinis įvertinimas bus **ne mažesnis nei 7** (iš 10 galimų).

Kiekvieno scenarijaus savybės įvertinimas gaunamas sudėjus visų ekspertų įvertinimus ir padalinus iš ekspertų skaičiaus. Apskaičiuojame kiekvienos dinamiškos savybės vidutinį įvertinimą sudedant visų ekspertų atliktų scenarijų vidutinius įvertinimus ir padalinant iš scenarijų skaičiaus. Vidutinis scenarijaus įvertinimas skaičiuojamas sudedant visus scenarijaus savybių įvertinimus ir padalinant iš savybių skaičiaus. Bendras vidutinis įvertinimas apskaičiuojamas sudedant visus savybių vidutinius įvertinimus ir padalinant iš savybių skaičiaus.

4 lentelė. Rezultatų suvestinė

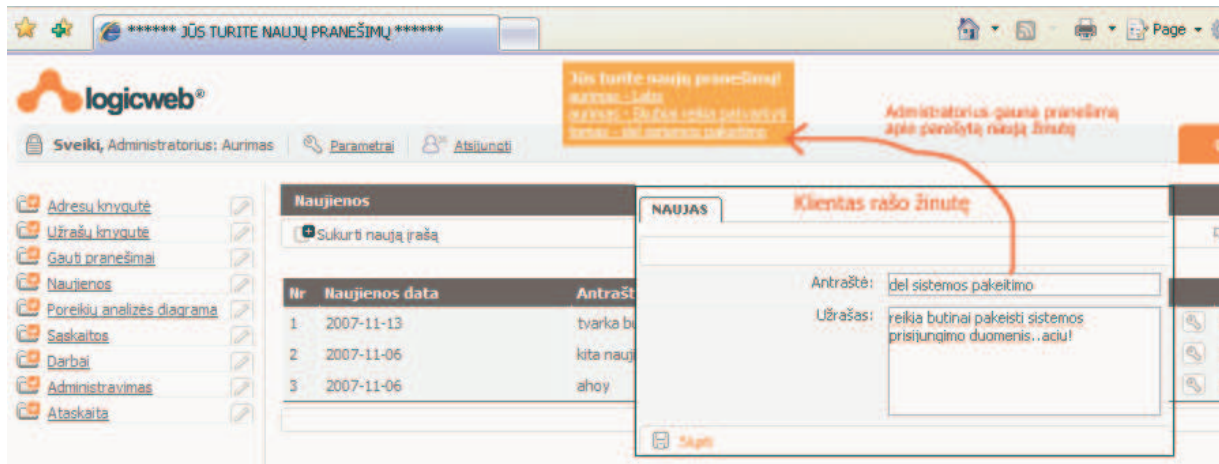
Savybės	1	2	3	4	5	6	Vidutinis scenarijaus įvertinimas
Scenarijus1	9	9,67	9,67	7,33	10	9	54,67 / 6 = 9,11
Scenarijus2	9,33	9,67	10	7,67	10	9,67	56,33 / 6 = 9,39
Scenarijus3	9	9	9,67	8	10	9,33	55 / 6 = 9,17
Scenarijus4	9,67	8,67	4	8,67	7	9,33	47,33 / 6 = 7,89
Scenarijus5	10	9,67	9,67	6,33	8	9,67	53,33 / 6 = 8,89
Scenarijus6	10	9,33	10	7,67	8,67	8,67	54,33 / 6 = 9,06
Scenarijus7	6	10	9,67	8,67	8,33	9	51,67 / 6 = 8,61
Scenarijus8	10	10	10	7,33	9,67	9,67	56,67 / 6 = 9,45
Scenarijus9	10	10	10	7,33	9,67	9,67	56,67 / 6 = 9,45
Vidutinis įvertinimas	83 / 9 = 9,22	86 / 9 = 9,56	82,67 / 9 = 9,19	69 / 9 = 7,66	81,33 / 9 = 9,04	84 / 9 = 9,33	
Bendras vidutinis įvertinimas	54 / 6 = 9 balai						

Taigi, iš 4 lentelės matome, kad operacijos su Ajax atliekamos daug greičiau. Kiek greičiau, lemia per kiek laiko pasileidžia pats puslapis, nes be Ajax pagalbos, norint atnaujinti dalį puslapio, reikia perkrauti visą puslapį iš naujo. Jei puslapyje daug operacijų – Ajax privalumas nenuginčytinas. Iš vertinimų sumų matome, kad dinamiškiausi yra pirmas, antras, trečias, aštuntas ir devintas scenarijai. Šiuose scenarijuose panaudotos funkcijos labiausiai pasitarnauja CRM sistemos dinaminės interneto sąsajos tobulinimui.

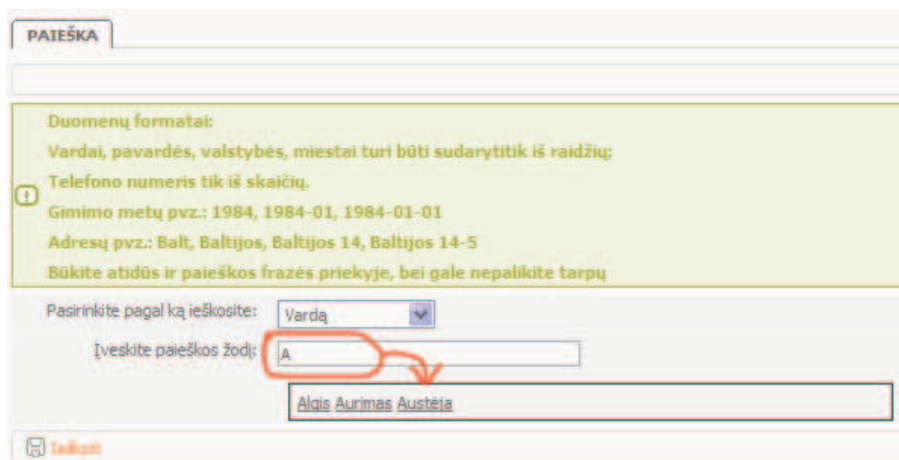
Galima teigti, jog sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus, kadangi bendras dinamiškųjų sistemos savybių vidutinis įvertinimas (9 balai) viršija prieš tai nustatytą ribinį 7 balų įvertinimą.

Mažiausiai dinamiškas yra ketvirtasis scenarijus, kadangi integruotoji sistema dar nėra pakankamai lanksti ir tiksli. Be to, šiame scenarijuje yra daug nedinamiškų operacijų, nėra dalinio puslapio atnaujinimo ir pats žemėlapis užkrovimas yra pakankamai lėtas. Paminėtos savybės labai lėtina šio puslapio funkcionavimą, todėl ir vertinimas yra mažiausias. Galima daryti išvadą, kad patobulinta sistema yra dinamiška ir interaktyvi, nes pagrindinės, dažniausiai naudojamos funkcijos yra sukurtos naudojant Ajax technologijas. Likusios funkcijos, kurios veikia be Ajax pagalbos neįneša per daug nedinamiškumo sistemai, kad būtų ją galima laikyti nedinamiška.

Keli patobulintos CRM sistemos scenarijų langai pavaizduoti 8 ir 9 paveikslėliuose.



8 pav. pagal Scenarijus8 atliekamo eksperimento vaizdas



9 pav. pagal Scenarijus3 atliekamo eksperimento vaizdas

Klientas parašo žinutę administratoriui. Administratoriaus sistemos darbiniam lange akimirksniu pasirodo pranešimas (8 pav) apie gautas naujas žinutes. Pasirinkęs pageidaujama žinutę, administratorius gali ją perskaityti ir pranešimo sąrašas akimirksniu pasikeičia (perskaityta žinutė iš sąrašo dingsta). Taip administratorius sutaupo didelę laiko dalį, kadangi jam nebereikia kartų nuo karto naviguoti per meniu punktus ir tikrinti, ar niekas neparašė naujo pranešimo. Tai viena geriausių ir svarbiausių sistemos patobulinimo funkcijų, suteikianti sistemai daug dinamiškumo, patogumo, raiškumo ir paspartinanti sąveiką tarp vartotojų.

Ieškomas norimas kontaktas. Pasirenkamas kriterijus, pagal kurį bus ieškoma – pagal „Vardą“. Tuomet vedamas paieškos žodis. Parašius jau pirmą raidę, realiu laiku Ajax varikliukas tikrina įvedamus simbolius ir ieško panašių atitikmenų duomenų bazėje. Jei įrašai randami, jų sąrašas akimirksniu pateikiamas žemiau vedamų simbolių esančiam bloke (9 pav). Iš pateikto sąrašo galima pasirinkti konkretų ieškomą kontaktą. Taip žymiai pagerėja sistemos interaktyvumas ir dinaminės savybės, pagreitinėja paieškos procesas ir apsaugojama nuo tuščių rezultatų gavimo.

5 Išvados

Literatūros šaltinių ir esamų sistemų analizė parodė, kad CRM sistemų kokybę galima pagerinti pritaikant raiškaus interneto technologijas (RIA). Technologinių RIA sprendimų analizės rezultate nustatyta, kad šiam tikslui geriausiai tinka JavaScript/DHTML AJAX įgyvendinimo technologija (Asinchroninis JavaScript + XML), kadangi ji gerai pritampa prie egzistuojančių HTML Web taikomųjų programų ir įgalina patobulinti esamą sistemą neperdarant visko nuo nulio.

CRM sistemos tobulinimas ir tyrimas parodė, kad tikslinga taikyti Ajax tokioms funkcijoms: Duomenų validavimui; Galimų sprendimų pasiūlymui paieškos metu; Kontaktų vietos nustatymui "Google Maps"; RSS naujienų skaitymui; Daliniam puslapių atnaujinimui administravimo meniu; Momentiniams pranešimams CRM vartotojams.

Specialistų atliktas devynių tipinių sistemos veikimo scenarijų įvertinimas pagal 6 dinamiškumo kriterijus parodė, kad sistemos dinamiškumas 10 balų skalėje yra lygus 9, o tai leidžia teigti, jog patobulinta sistema yra dinamiška ir tenkina RIA taikomųjų programų reikalavimus.

Šiame darbe taikoma metodika leido sutaupyti 75 procentus darbo laiko kuriant dinamišką CRM sistemą. Tai yra pakankamai nebrangus būdas egzistuojančiai taikomajai programai suteikti daugiau dinamiškumo.

Atliktas tyrimas įgalino sudaryti rekomendacijas RIA technologijoms taikyti CRM ir kitoms įmonių taikomųjų programų sistemoms, tuo padidinat jų dinamiškumą ir palengvinant administravimą.

Literatūros sąrašas

- [1] ANSI What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software, by Tim O'Reilly 09/30/2005. <http://tim.oreilly.com/news/2005/09/30/what-is-web-20.html>
- [2] J. Allaire. Macromedia Flash MX-A next-generation rich client. Technical report, Macromedia, March 2002.
- [3] A Fragment-Based Approach for Efficiently Creating Dynamic Web Content, by Jim Challenger, Arun Iyengar, Karen Witting, Cameron Ferstat, and Paul Reed, which appeared in Proceedings of INFOCOM 2005, 2005 IEEE.
- [4] Ajax: A New Approach to Web Applications, by Jesse James Garrett, February 18, 2005. <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [5] An Open Source AJAX Comparison Matrix, by Edmon Begoli. December 5, 2006. <http://www.devx.com/AJAXRoundup/Article/33209>
- [6] The State of Web Frameworks. Craig R. McClanahan, Sun Microsystems, Inc., February 25, 2006. URL: http://kr.sun.com/developers/PDFs/preso/Craig_JCO2006.pdf
- [7] AJAX Design Strategies. By Ed Ort and Mark Basler, October 2006. URL: <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/DesignStrategies/index.html>
- [8] Wikipedia. Ajax (programming). Žiūrėta 2007-05-22. Prieiga per internetą <http://en.wikipedia.org/wiki/AJAX>
- [9] A. Verickas. Web 2.0 – naujoji interneto karta. // Kompiuterija. PC World. 2007 Sausio 15. Žiūrėta 2007-11-19. Prieiga per internetą: <http://www.kompiuterija.lt/zurnale/2007-01-15/web-20-%E2%80%93-naujoji-interneto-karta/>

Dynamic Web interface for improvement of enterprise customer relationship management systems

This work is concentrated on creation of dynamic web systems. Web 2.0 brings new requirements to internet based systems (web pages). Users needs to work with large amounts of data (information) using simple web browser are increasing very fast. So there is a problem when we need to represent this data on web page. We need dynamic systems (pages) to do it fast and easy to operate. Dynamic system (page) is internet based system (web page), which most functions are using Ajax (asynchronous communication with server).

The real experimental system to increase a dynamism of Customer Relationship Management systems by extending them with Rich Internet Applications characteristics is created and analyzed. Recommendations to improve similar existing CRM systems are made.



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

Studentų g. 50, 51368 Kaunas. Tel.: +370 37 30 03 51 / 30 03 50.
Fax +370 37 30 03 52. <http://www.if.ktu.lt> E-mail rastif@ktu.lt

PAŽYMA

2008-01-11

Šiuo dokumentu patvirtinu, kad Aurimas Junčis ir Lina Nemuraitė konferencijos IT 2008 organizatoriams pateikė straipsnį „Dynamic Web Interface for Improvement of Enterprise Customer Relationship Management Systems“.

14-ta tarptautinė informacinių ir programinės įrangos technologijų konferencija „Informacinės technologijos 2008“ (IT 2008) įvyks Kaune 2008 m. balandžio 24-25 d. Konferencijos internetinis adresas <http://www.isd.ktu.lt/it2008/>.

Konferencijos IT 2008
organizacinio komiteto pirmininkas

Rimantas Butleris

UAB „REKLAMOS VIZIJA“

Kauno Technologijos Universiteto
Informatikos fakultetui

SISTEMOS PATOBULINIMO PAŽYMA

2008 m. sausio 10 d.
Kaunas

UAB "Reklamos vizija" užsakymu programuotojas Aurimas Junčis atliko "Deform Group Support" CRM sistemos patobulinimo darbus. Egzistuojančiai klientų-ryšių valdymo sistemai sukurta dinaminė sąsaja, suteikianti vartotojų interfeisui didesnę dinamiškumą ir naudojimo patogumą.

Tvirtinu, jog patobulinta sistema funkcionuoja teisingai ir tenkina iškeltus pradinčius reikalavimus.

I.e.p.direktorius



Audrius Karečka