

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

MULTIMEDIJOS INŽINERIJOS KATEDRA

Almantas Šilinskas

AJAX technologijų analizė ir demonstravimas

Magistro darbas

Darbo vadovas:

Doc. Armantas Ostreika

Kaunas, 2007

Turinys

1	Įvadas	5
2	Ajax technologijų analizė	6
2.1	Pagrindiniai Ajax požymiai.....	6
2.2	AJAX naudojimas praktikoje	10
2.3	AJAX alternatyvos	12
2.4	AJAX technologijos sudėtinės dalys.....	13
2.5	Pagrindiniai Ajax tikslai.....	16
2.6	Turtingos vartotojo sąsajos programinės įrangos sukūrimas	19
2.7	Sritis, kuriose didžiausios Ajax technologijų perspektyvos.....	21
3	Projektinė dalis.....	24
3.1	Projektinė specifikacija	24
3.1.1	Sistemos paskirtis.....	24
3.1.2	Projekto kūrimo pagrindas (pagrindimas)	24
3.1.3	Sistemos tikslai (paskirtis)	25
3.1.4	Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys	25
3.1.5	Vartotojai	26
3.1.6	Projekto apribojimai.....	26
3.1.7	Sistemos kūrimo terminai	28
3.1.8	Sistemos kūrimo biudžetas	28
3.1.9	Svarbūs faktai ir prielaidos	29
3.1.10	Funkciniai reikalavimai.....	30
3.1.11	Produkto veiklos sfera.....	31
3.1.12	Panaudojimo atvejų sąrašas.....	32
3.1.13	Funkciniai reikalavimai ir reikalavimai duomenims.....	36
3.1.14	Reikalavimai duomenims	39
3.1.15	Nefunkciniai reikalavimai	39
3.1.16	Projekto išeiğa	41

3.1.17	Egzistuojantys sprendimai.....	41
3.1.18	Naujos problemos.....	41
3.1.19	Uždaviniai	43
3.1.20	Pritaikymas.....	44
3.1.21	Rizikos.....	44
3.1.22	Kaina	46
3.1.23	Vartotojo dokumentacija ir apmokymas	46
3.1.24	Perspektyviniai reikalavimai	48
3.1.25	Idėjos sprendimams.....	48
3.2	Sistemos paketai.....	49
3.2.1	HIBERNATE paketas.....	49
3.2.2	UTIL paketas	51
3.2.3	WEB paketas.....	53
3.2.4	PROXY paketas	56
3.3	Sistemos architektūra	60
3.3.1	Architektūros tikslai ir apribojimai.....	60
3.3.2	Panaudojimo atvejų vaizdas.....	62
3.3.3	Sistemos sąveikos diagrama	63
3.3.4	Sistemos statinis vaizdas.....	65
3.3.5	Paketų detalizavimas.....	66
3.3.6	Tapestry karkasas.....	67
3.3.7	Klasių diagramos	68
3.3.8	Hibernate karkasas	73
3.3.9	Sistemos dinaminis vaizdas	74
3.3.10	Išdėstymo vaizdas	76
3.3.11	Duomenų vaizdas	77
3.3.12	Kokybė	78
3.4	Testavimo medžiaga.....	79
4	Vartotojo dokumentacija.....	80
5	Produkto kokybės įvertinimas.....	82
6	Išvados	83

Literatūra.....	84
Terminų ir santraukų žodynas.....	85
Priedas Nr. 1. Ajax vartotojo sąsajos komponentai.....	86
Priedas Nr. 2. Ajax puslapių architektūra.....	103

1 Įvadas

Pastaruosius pusantrų metų kone didžiausią susidomėjimą interneto naršyklės pagrindu kuriamos programinės įrangos srityje kelia Ajax technologija. Šiai technologijai skiriamas nemažas dėmesys įvairiuose interneto naršyklės pagrindu veikiančios programinės įrangos kūrėjų forumuose ir atsiliepimuose. Ajax technologijos naudojimas daugelio specialistų nuomone, atveria plačias galimybes kurti naujos kartos veikiančios interneto naršyklės pagrindu programinę įrangą. Šiai programinės įrangos kartai, kai kurių autorių pastangomis, net buvo suteiktas naujas pavadinimas – web 2.0. Kadangi Ajax yra palyginti nauja technologija, tai nėra nustovėjusios vieningos nuomonės nei kas yra tas Ajax, nei nėra atlikta detalesnių studijų, kaip ji yra ir gali būti naudojama interneto naršyklės pagrindu veikiančios programinės įrangos kūrimo srityje.

Atsižvelgdamas į Ajax technologijai rodomą didžiulį dėmesį, aš savo darbe ir atliksiu bendrą Ajax technologijų veikimo bei jų naudojimo būdų ir galimybių analizę bei sukursiu naujo tipo Ajax technologijų pagrindu besiremiančią programinę įrangą – Paveiksliukų nutolusiam vartotojui demonstravimo realiu laiku programinę įrangą – „Online prezentacija“.

Tad mano darbo du pagrindiniai šio darbo uždaviniai yra sekantys:

1. Atlikti Ajax technologijų veikimo principų analizę ir nusistovėjusią jos taikymo praktiką bei galimybių įvertinimą.
2. Sukurti programinę įrangą „Prezentacija online“, įgalinančią demonstruoti perkeltus į serverį paveiksliukus fiziškai nutolusiems stebėtojams sinchroniškai realiu laiku per internetą, naudojant Ajax technologiją.

2 Ajax technologijų analizė

Neretai sakoma, kad programinės įrangos vartotojo sąsaja yra geriausia tokia, kurios vartotojai apskritai nepastebi ir kuri suteikia galimybę patogiai ir be vargo atlikti tuos veiksmus, kuriuos vartotojas duotuoju laiko momentu nori. Be abejo, tai pasiekti yra gana sunku ir neretai reikia nemažai laiko, kol atrandamas vienas ar kitas būdas kaip spręsti vienokią ar kitokią programinės įrangos vartotojo sąsajos netobulumą ar trūkumą. Tačiau būtent pastaruoju metu programinė įranga, veikianti interneto naršyklės pagrindu išgyvena didelį pakylėjimą ir naujų vartotojų sąsajų paiešką bei kūrimą. Šios rūšies programinei įrangai atskirų autorių dėka buvo netgi suteiktas pavadinimas web 2.0, o jos pagrindas yra tai, kad programinės įrangos kūrėjai galima atsakyti „pastebėjo“ būdą, kaip sukurti naujos kartos, pasižyminčios kur kas didesniu vartotojo sąsajos interaktyvumu bei patogumu, interneto naršyklės pagrindu veikiančią programinę įrangą. Už visa tai, iš esmės glūdi technologija (tiksliau būtų galima sakyti, atskirų technologijų rinkinys) pavadinimu AJAX (angl. Asynchronous JavaScript + XML). AJAX termino krikštąteviu neabejotinai pripažįstamas Jesse James Garrett, pirmasis pavartojęs šį terminą savo 2005 m. vasario 18 d. straipsnyje „Ajax: A New Approach to Web Applications“. Iki tol atskiros dalys to, kas dabar įvardinama kaip AJAX, buvo įvardijamos angliškai „Dynamic HTML“ ir „remote scripting“.

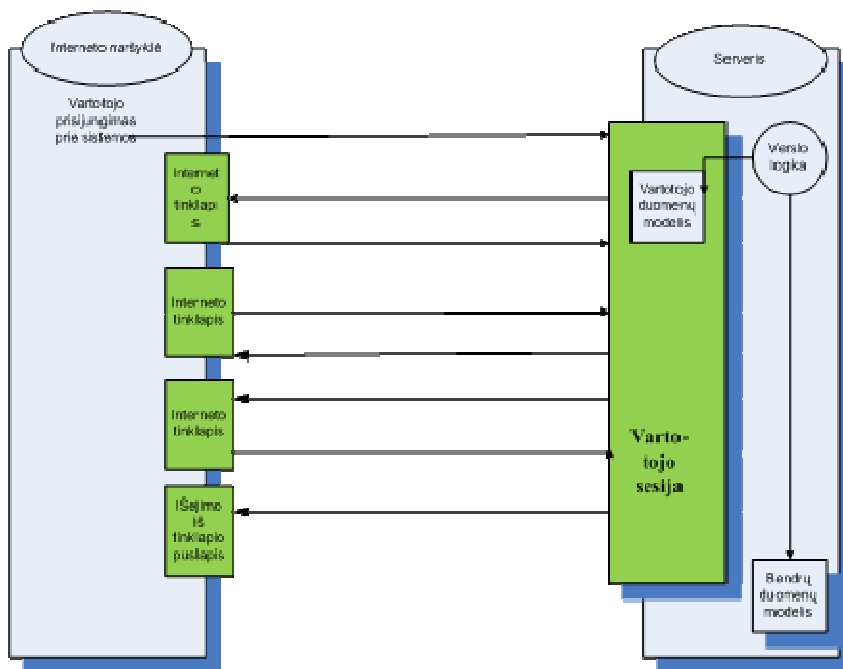
2.1 Pagrindiniai Ajax požymiai

Žemiau yra pateikiami keturi pagrindiniai požymiai, nusakantys bendrą AJAX veikimo esmę ir išskirtinumą:

- **Interneto naršyklėje veikia programinės įrangos aplikacija, o ne tiktai atvaizduojamas turinys**

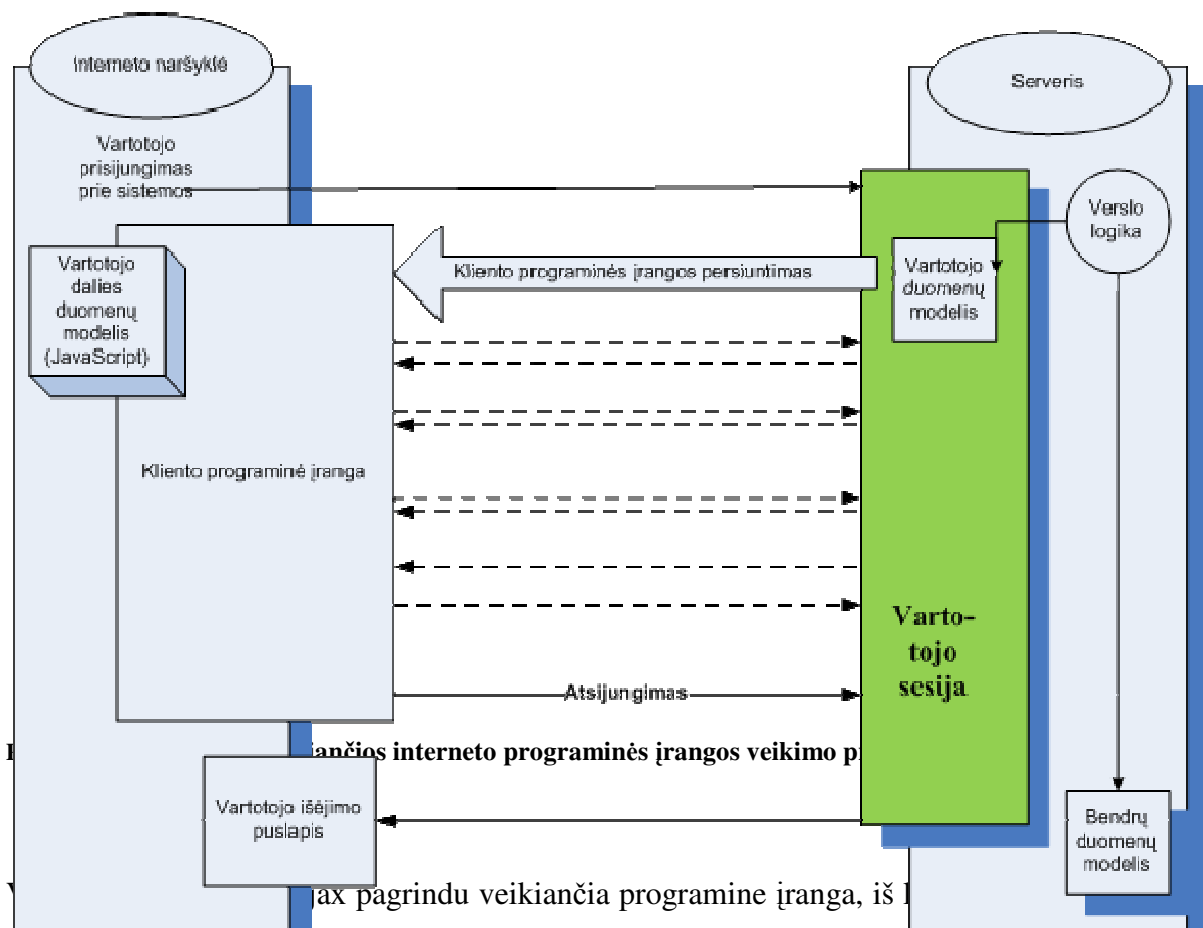
Klasikinėje, prieš pradėdant naudoti AJAX technologijas, interneto naršyklės paskirtis visoje interneto naršyklės pagrindu veikiančioje programinėje įrangoje buvo tiktai gauti ir teisingai atvaizduoti iš serverio gaunamą puslapio turinį. Visa informacija, susijusi su vartotojo darbu

su programine įranga, yra laikoma serveryje, paprastai tam kartui atskiram vartotojui sukurtoje sesijoje. Kiekvieną kartą vartotojui paspaudus ant atskiros nuorodos, atskiras dokumentas yra siunčiamas vartotojui, o interneto naršyklė paprasčiausiai „išmeta“ iš vartotojo ekrano prieš tai buvusį puslapį ir pateikia naujai gautą. Paprastai tariant, interneto naršyklės „klasikinėje“ interneto programinėje įrangoje vaidmuo yra gana „bukas“ ir ji tikrai atvaizduoja informaciją, nesiimdama pati kokių nors „veiksmų“. Klasikinės interneto naršyklės programinės įrangos veikimo principas pateiktas žemiau esančiame paveikslėlyje Pav. 1:



Pav. 1. Klasikinės interneto programinės įrangos veikimo principas.

Tuo tarpu Ajax pagrindu veikiančioje programinėje įrangoje dalis aplikacijos veikimo logikos yra perduodama interneto naršyklei, kaip parodyta žemiau pateiktame paveikslėlyje Pav. 2:

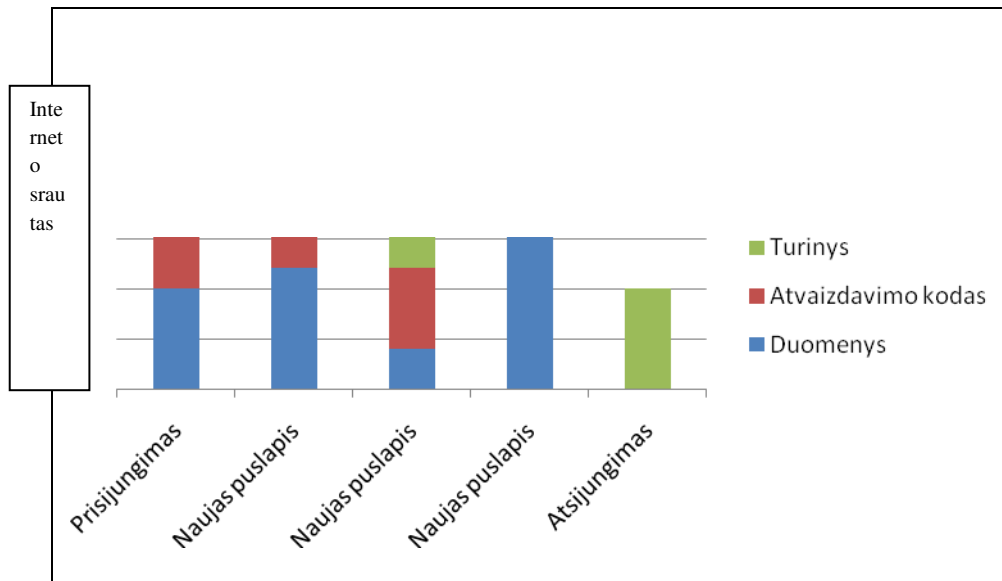


veikiančia interneto programinės įrangos veikimo p
 ax pagrindu veikiančia programine įranga, iš
 programine įranga yra persiunčiama ne paprastas, tikrai informaciją pateikiantis puslapis
 kaip „klasikinės“ interneto programinės įrangos atveju, bet sudėtingesnis dokumentas,
 turintis savyje nemažą dalį JavaScript kodo. Po to vartotojas ir dirba su šiuo dokumentu, o jis
 visas pats per naują nėra persiunčiamas, o persiunčiama tikrai vartotojo veiksmų iššaukiama
 dalis informacijos arba atnaujinami tam tikri puslapio elementai. Todėl tai jau galima vadinti
 tam tikru „klientu“, o ne paprasčiausiu informaciją pateikiančiu puslapiu.

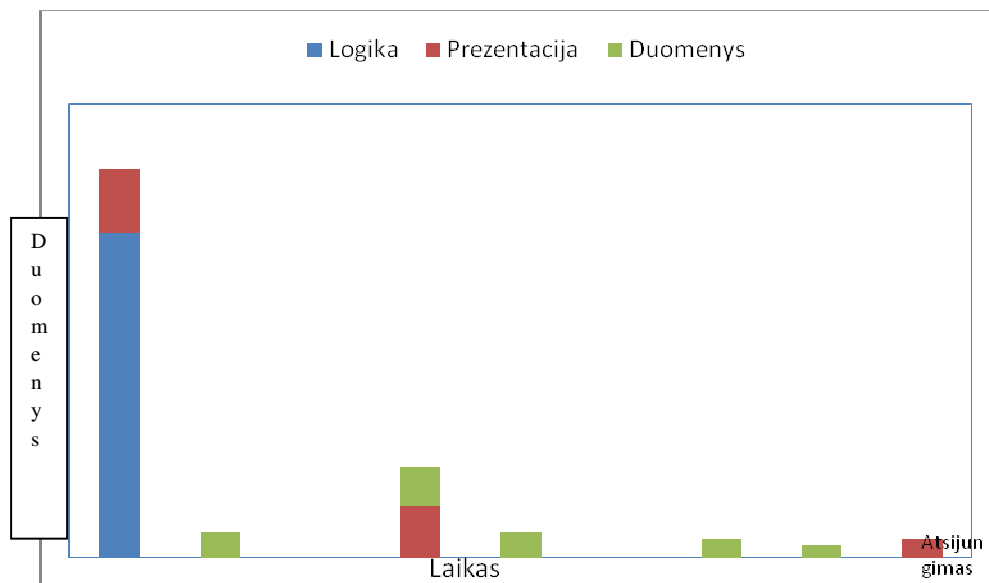
- **Serveris persiunčia duomenis, o ne turinį**

Kaip jau minėjau aukščiau “klasikinės” interneto programinės įrangos atveju klientui nusiunčiamas puslapio turinys, suformuotas iš atskirų dizaino ir navigacijos elementų bei duomenų. AJAX interneto programinės įrangos atveju, viskas vyksta šiek tiek “racionaliau”. Šiuo atveju kliento programinė įranga nusiunčia asinchroninę užklausą serveriui, o serveris persiunčia atgal ne visą klientui pateikiamą informaciją ir navigacijos elementus, bet tikrai atnaujintos informacijos dalį. Tiek “klasikinės” interneto programinės, tiek AJAX pagrindu

sukurtos interneto programinės įrangos persiunčiamos informacijos atvaizduoti žemiau pateiktuose paveikslukuose:



Pav. 3 Įprastinės interneto programinės įrangos interneto srautas



Pav. 4 Ajax programinės įrangos interneto srautas

Kaip matyti iš pateiktų aukščiau paveikslukų, AJAX atveju pradžioje yra klientui persiunčiamas “sunkesnis” ir nemažai informacijos savyje turintis klientas. Po to sekantis bendravimas su serveriu yra daug racionalesnis ir efektyvesnis bei klientui yra persiunčiama tiktai atsinaujinusi informacija. Atitinkamai, jeigu vartotojas atlieka tiktai vieną ar du

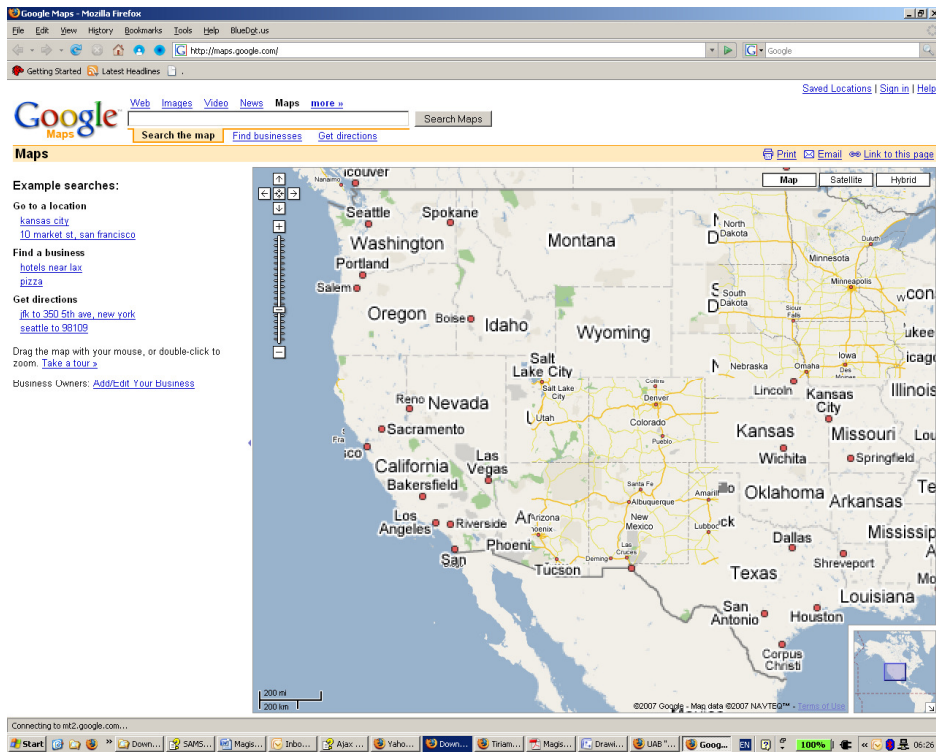
veiksmus su programine įranga, “klasikinės” interneto programinės įrangos perduodamas srautas gali būti mažesnis, tačiau kai vartotojas ilgiau dirba vienos sesijos metu su programine įranga, AJAX programinės įrangos atveju bendras persiųstų duomenų kiekis bus mažesnis nei “klasikinės” interneto programinės įrangos atveju.

- **Vartotojo darbas su AJAX programine įranga yra išsistinis ir nepertraukiamas**

Standartinėje interneto naršyklės pagrindu veikiančioje programinėje įrangoje, vartotojui atlikus tam tikrus veiksmus ir gaunant nesvarbu kokio dydžio naują informaciją, yra “perkraunamas visas tinklapis. Tuo tarpu Ajax technologijų pagrindu veikiančioje programinėje nauja informacija yra persiunčiama, nereikalaujant perkrauti visą puslapį, todėl vartotojas gali dirbti su Ajax pagrindu sukurta programine įranga, neblaškydamas savo dėmesio. Tai daro, skirtingai nei standartinės interneto naršyklės pagrindu sukurtos programinės įrangos atveju, jo darbą su programine įranga išsistiniu ir nepertraukiamu.

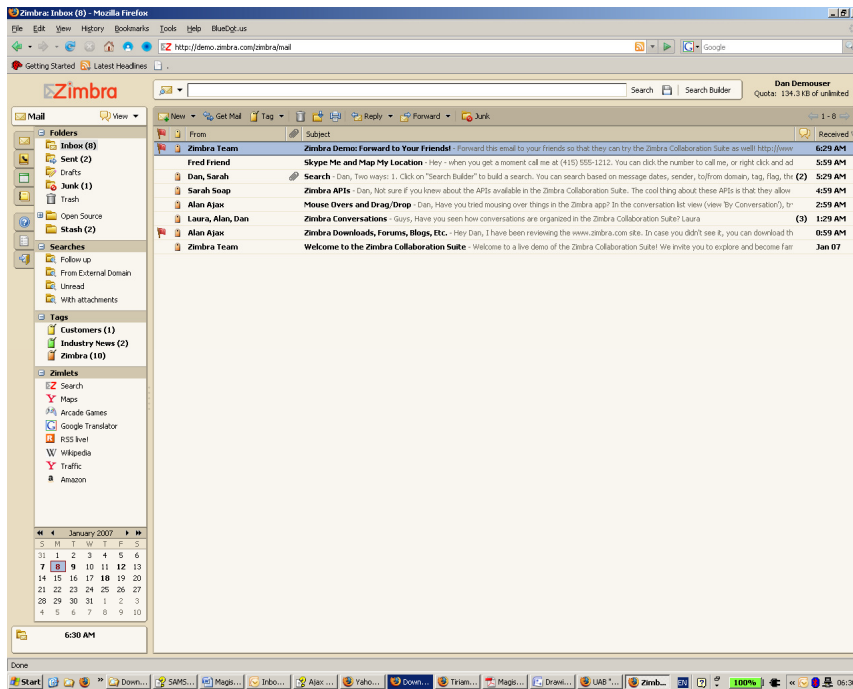
2.2 AJAX naudojimas praktikoje

Ajax technologijos labai sparčiai plinta. Kone labiausiai aprašytas atvejis, kuris tam tikra prasme ir sąlygojo Ajax išpopuliarėjimą buvo Google Maps, kurio bendras vartotojo sąsajos vaizdas, pateiktas žemiau.



Pav. 5 Google Maps vartotojo sąajos bendras vaizdas

Kitas įdomus pavyzdys yra žimbra elektroninio pašto klientas, kuriame taip pat gausu Ajax technologijų. Žimbra vartotojo grafines sąajos vaizdas pateiktas žemiau esančiame paveikslėlyje:



Pav. 6 Zimbra vartotojo sąsajos vaizdas.

2.3 AJAX alternatyvos

AJAX nėra vienintelis būdas kurti interaktyvią ir turtingą interneto programinės įrangos vartotojo sąsają. Kitos dvi pagrindinės šiam tikslui naudojamos alternatyvos yra:

- **Flash technologijos, sukurtos bendrovės „Macromedia“**

Macromedia bendrovės sukurtos ir išpopuliarintos Flash technologijos įgalina kurti vartotojo sąsają, panaudojant suspaustą vektorinės grafikos formatą. Su flash'u sukurta programinė įranga yra siunčiama vartotojui srautu (angl. streaming) ir pradeda leisti vartotojui, dar ją ne visą parsisiuntus. Flash technologija taip pat yra tam tikra prasme programuojama su ActionScript, savo esme panašiu į JavaScript. Iš dalies, vienas iš pagrindinių minusų Macromedia Flash technologijos yra tai, kad vartotojas turi instaliuoti Macromedia Flash plug-in'ą. Macromedia Flash technologija plačiau naudojama ten kur ypač svarbu dizainas ir yra plačiausiai naudojama kuriant įvairias prezentacijas bei interneto žaidimus.

Turtingos vartotojo sąsajos programinės įrangos kūrimui yra skirtos dvi praplečiančios technologijos – tai Macromedia Flex ir atvirojo kodo Laszlo suite. Jos abi savyje turi supaprastintus serverio aplinkos kūrimo karkasus (angl. framework), naudojančius Java / Java 2 Enterprise Edition (J2EE) serverio dalyje technologiją.

- **Java Web Start ir susijusios technologijos**

Java Web Start technologija įgalina Java programavimo kalbos pagrindu kurti interneto ryšio pagrindu veikiančią programinę įrangą, kuri suinstaliuota į kliento kompiuterį gali susirasti, paleisti ir naudoti procesus, esančius serverio programinėje įrangoje. Tradiciškai, Web Start programinės įrangos vartotojo sąsaja kuriama su Java Swing priemonėmis. Paminėtina ir tai, kad IBM bendrovės sukurtas Standard Widget Toolkit (SWT) taip pat gali veikti su Web Start.

Tačiau, kadangi mano šio darbo pagrindinis tikslas yra išanalizuoti bei pademonstruoti Ajax technologijų naudojimą, tai šių aukščių paminėtų technologijų nesirinksiu. Be to, jos tam tikra prasme ir netiktų, nes mano pagrindinis siekis yra sukurti tokią programinę įrangą, kuri nereikalautų iš vartotojų instaliuotis jokios kitos papildomos programinės įrangos, išskyrus tikrai turėti interneto naršyklę ir interneto ryšį.

2.4 AJAX technologijos sudėtinės dalys

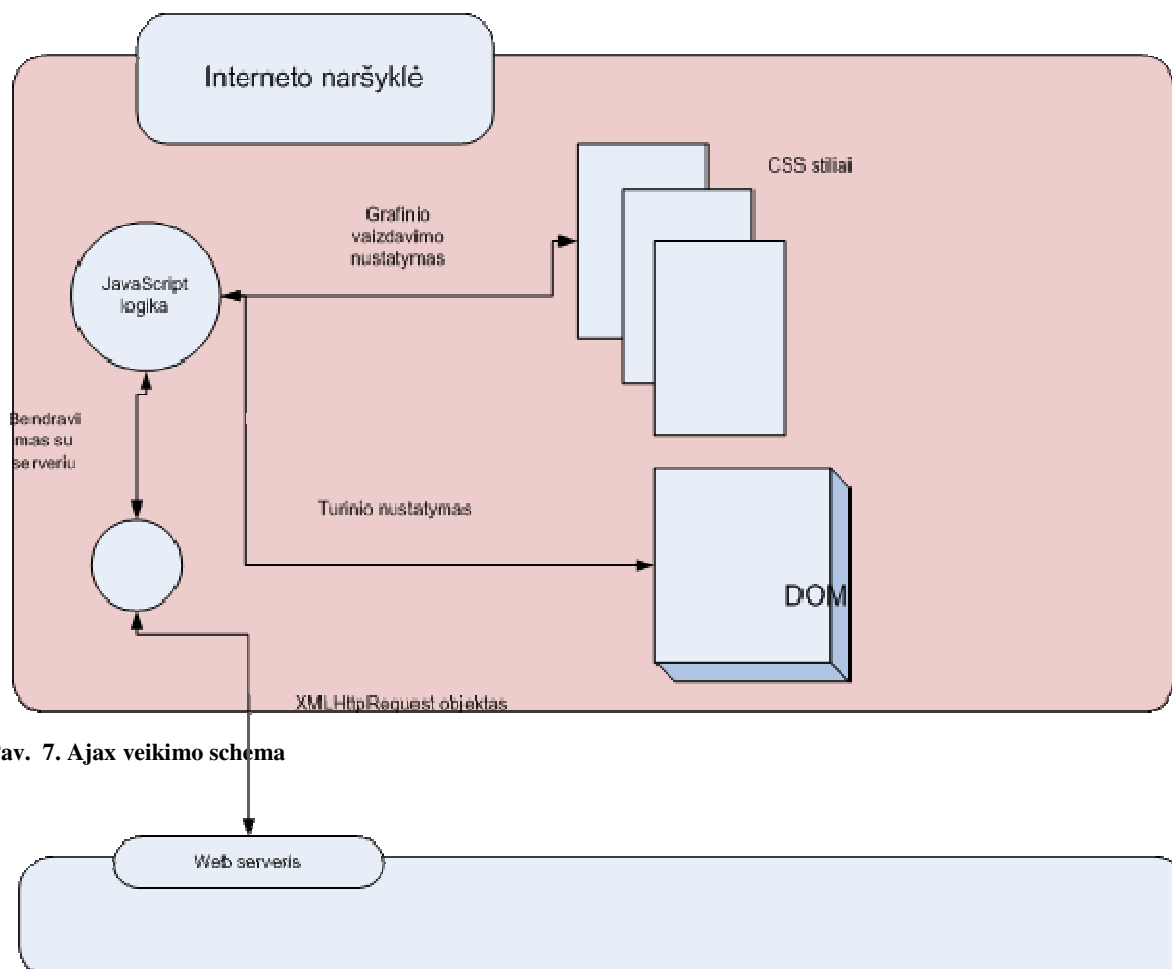
Tai, kas vadinama AJAX yra keturių technologijų, kurios papildo viena kitą, rinkinys. Šios technologijos yra:

JavaScript	JavaScript yra bendro pobūdžio skripto programavimo kalba, pritaikyta naudoti programinės įrangos viduje. JavaScript interpretatorius, esantis interneto naršyklėje, leidžia kuriamai programai naudoti interneto naršyklėje realizuotas galimybes. Pati kliento programinė įranga yra kuriama su JavaScript.
Cascading Style Sheets (CSS)	CSS suteikia galimybę aprašyti bendrai naudojamus tinklapio vizualaus atvaizdavimo stilius ir atskirus elementus. CSS suteikia galimybę paprastai naudoti / keisti vieningą stilių. AJAX pagrindu sukurtoje programinėje įrangoje, vartotojo grafinė aplinka yra nusakoma CSS pagalba.

Dokumentų Objekto Modelis (angl.Document Object Model) (DOM)	DOM išskaido atskirų puslapių struktūrą į atskirus programuojamus objektus, kuriuos galima valdyti JavaScript pagalba. DOM pagalba yra įmanoma modifikuoti ir keisti vartotojo sąsają iš karto, realiai keičiant atskiras sudėtines puslapio dalis.
XMLHttpRequest objektas	XMLHttpRequest objektas leidžia programuotojams paimti duomenis iš serverio vartotojui vizualiai nematant šio proceso. Paimamų duomenų formatas paprastai yra XML, bet AJAX gerai veikia ir su tekstiniais duomenimis. Be to, nors XMLHttpRequest yra labiausia lankstus ir bendras būdas duomenims paimti, galima naudoti ir kitus duomenų paėmimo būdus.

AJAX programinėje įrangoje JavaScript yra tarsi ta medžiaga, kuri suklijuoja visus komponentus į vieną visumą ir apibrėžia procesų eigą bei aplikacijos veikimo verslo logiką. Vartotojo sąsaja yra atnaujinama ir valdoma JavaScript pagalba per Dokumentų objekto modelį - DOM.

Žemiau yra pateikta bendra AJAX veikimo schema:



Pav. 7. Ajax veikimo schema

2.5 Pagrindiniai Ajax tikslai

Nepaisant įvairiausių vertinimų ir nepaprastai didelių kai kurių informacinių technologijų specialistų lūkesčių, Ajax pagrindinis tikslas yra pagerinti vartotojo bendravimo su programine įranga sąsają. Vartotojo sąsajos pagerinimus galima išskirti į dvi pagrindines kategorijas: padaryti vartotojo atliekamą atlikimą paprastesnį ir patogesnį bei sukurti tokią programinę įrangą, kuri įgalintų vartotoją atlikti veiksmus, kurių nebuvo galima prieš tai atlikti. Kuriant interneto programinę, esamų vartotojų atliekamų veiksmų pagerinimą galima išskirti į du pagrindinius tikslus: padidinti aplikacijų interaktyvumą ir sumažinti laiką, reikalingą tam tikram veiksmui atlikti. Ne intraneto aplikacijoms, dar yra svarbu sumažinti perduodamų duomenų srauto dydį, sumažinti tinklapio perkrovimų skaičių ir pagerinti bendrą vartotojo sąsają.

- **Interaktyvumo padidinimas**

Vienas iš bendrų tikslų, kurių siekiama bendrai naudojant JavaScript tinklapiuose, yra interaktyvumo padidinimas. Net ir nenaudojant Ajax, galima vartotojui pateikti grįžtamojo ryšio informaciją, nepersiuočiant viso tinklapio informacijos, kai vartotojas pildo tam tikrą formą ar užveda su pele ant nuorodos. Ši papildoma informacija padeda vartotojams geriau orientuotis tinklapyje ir jų neapkrauna pertekline informacija. Naudojant AJAX, galima pateikti ne statinę, bet dinamiškai generuojamą ir pateikiamą informaciją.

Vienas iš tokių taikymo pavyzdžių yra Google suggest, kai renkant paieškos žodžius, paieškos sistema realiu laiku atlieka paiešką ir parodo ne tik tai siūlomus žodžius ir atskiras frazes, bet ir rodo, koks skaičius paieškos rezultatų bus pateiktas. Žemiau yra pateiktas paveiksliukas, kaip Google Suggest paieška atrodo:

Vienas didžiausių interneto programinės įrangos trūkumų yra tai, kad daugiažingsnis procesas tampa daugiaminučiu procesu, tai yra kuo daugiau žingsnių, tuo daugiau reikia laiko vartotojui atlikti norimus veiksmus. Netgi esant greitam interneto ryšiui, kiekvieno puslapio perkrovimas reikalauja vienos ar dviejų sekundžių naujam puslapiui sugeneruoti ir interneto naršyklėje atvaizduoti. Esant lėtesniam interneto ryšio greičiui, laikas, reikalingas, sugeneruoti atskiriems puslapiams, padvigubėja arba netgi patrigubėja. AJAX naudojimas įgalina panaikinti atskirų puslapių perkrovimo būtinybę, ir reikalauja laiko tiktai papildomos informacijos ar duomenų persiuntimui, kas sutaupo nemažai laiko, reikalingo duomenims perduoti ir vartotojo naršyklėje atvaizduoti.

Be abejo, yra ir daugiau situacijų, kai AJAX gali sumažinti veiksmų atlikimo laiką. Be to, neretai reikia surasti veiksmus ar operacijas, kurios reikalauja daugiausiai laiko ir išanalizavus, pasirinkti būdus kaip šį laiką sumažinti. Kaip pavyzdys, galėtų būti situacija, kai reikia suteikti vartotojams galimybę pasirinkti iš didelio variantų sąrašo. Siunčiant visą sąrašą vartotojui, tai reikalautų nemažai laiko, kol visas sąrašas yra persiunčiamas. Labai ilgas iškrentantis sąrašas yra taip pat nepatogus naudoti. Naudojant standartines HTML galimybes, daugeliu atveju būdavo pridėdama paieškos galimybė surasti norimą pasirinkti reikšmę. Kad išvengti viso puslapio perkrovimo, paieškos modulis būdavo pateikiamas iššokančiame lange, tačiau nepaisant to vienos reikšmės pasirinkimas užimdavo 5-30 sekundžių. Žinoma, jeigu vienoje opcijoje reikia pasirinkti 1-2 reikšmes, tai gal ir nėra didelė problema. Tačiau problema pasireikšdavo tada, kai reikėdavo iš vieno sąrašo pasirinkti 20 ar daugiau reikšmių. AJAX suteikia galimybę vykdyti norimų pasirinkti reikšmių paiešką realiu laiku. Vartotojo sąsaja AJAX atveju atrodytų labai panašiai, kaip aukščiau paminėtu Google Suggest atveju. Atlikti tyrimai, rodo kad AJAX panaudojimas pasirinkti norimų 20 reikšmių užimtų 1 minutę, kaip tuo tarpu standartinėmis HTML galimybėmis padaryta programinė įranga įgalintų tai padaryti ne mažiau kaip per 5 minutes.

Naudojant AJAX, galima pagreitinti procesus, suteikiant vartotojams galimybę redaguoti informaciją bei tam tikrus elementus ar objektus vizualiai perkelti į kitą ekrano vietą. Redaguoti elementą galima, neperkraudant iš naujo visą puslapį. Vartotojui pateikus norimą informaciją, ši persiunčiama į serverį, kai tuo metu vartotojas jau gali „dirbti“ su kita tinklapio informacija ar atlikti kitus veiksmus. Vizualus objektų perkėlimas į kitą ekrano

vietą taip pat sutaupo daug vartotojo laiko, nes yra daug paprasčiau matyti tikslą AJAX pagrindu sukurtose sistemose nei be AJAX technologijų sukurtose interneto aplikacijose.

- **Interneto srauto sumažinimas**

Kadangi AJAX technologijų pagrindu sukurtoje interneto programinėje įrangoje duomenys yra persiunčiami mažesnėmis dalimis, tai savo ruožtu sumažina ir persiunčiamos informacijos bendrą srautą. Jeigu yra mokama už srauto kiekį, tai AJAX taikymas savo ruožtu sumažina ir aplikacijos vartotojų bei aplikacijos teikėjų (jeigu mokama už serverio išsiunčiamą ir/ar gaunamą srautą) kaštus interneto paslaugų teikėjams. Be abejo, jeigu interneto naršyklės pagrindu veikianti programinė įranga yra naudojama organizacijos viduje, šis “sutaupymas” nėra aktualus bei interneto greitis greičiausiai yra pakankamas užtikrinti greitą programinės įrangos veikimą iš vartotojo pusės.

Be abejo, reikia pažymėti, kad interneto srauto sumažinimas toli gražu nėra svarbiausias kriterijus, kuriant šiuolaikines interneto aplikacijas ir, nors AJAX technologijos suteikia tokią galimybę atskirais atvejais sumažinti bendrą reikalingą interneto srautą, AJAX technologijų naudojimas daugeliu atveju netgi padidins interneto srauto naudojimą, siekiant vartotojui suteikti patogesnę programinės įrangos vartotojo aplinką. Be to, naudojant AJAX technologijas, nebūtina iš karto vartotojui, pradėjus naudoti programinę įrangą, persiųsti visus reikalingus duomenis ir informaciją iš karto, bet galima tai daryti tai palaipsniui, bei persiųsti papildomus reikalingus duomenis ir informaciją, to net nepastebint pačiam vartotojui.

2.6 Turtingos vartotojo sąsajos programinės įrangos sukūrimas

Aukščiau įvardinti trys argumentai už AJAX naudojimą skirti daugiau patobulinti esamą interneto naršyklės pagrindu veikiančią programinę įrangą, tačiau AJAX naudojimas suteikia galimybę sukurti visiškai naujo kokybinio lygio interneto naršyklės pagrindu veikiančią programinę įrangą. Tokios aplikacijos literatūroje yra vadinamos turtingos vartotojo sąsajos programine įranga. Turtingos vartotojo sąsajos programine įranga yra paprastai vadinama tokia programine įranga, kuri savo vartotojo sąsaja ir veikimu siekia atkartoti įprastą atskirame kompiuteryje veikiančią programinę įrangą, savo ruožtu išsaugant interneto

programinės įrangos pranašumus, kaip paprastas diegimas ir palaikymas. Be abejo, net ir kuriant turtingos vartotojo sąsajos programinę įrangą yra svarbu taip pat padidinti vartotojo sąsajos interaktyvumą bei sumažinti laiką reikalingą atlikti tam tikrus veiksmus. Tačiau, žinoma, neretai tenka rinktis ir ieškoti kompromiso tarp turtingos vartotojo sąsajos ir interneto naršyklės pagrindu veikiančios programinės įrangos bendro greičio.

Be abejo, turtingos vartotojo sąsajos programinės įrangos patį veikimą reikia lyginti ne su prieš tai vyraavusiomis interneto naršyklės pagrindu veikiančiomis interneto aplikacijomis, bet su analogiškais desktopinėmis aplikacijomis. Be abejo, sukurti tokias interneto aplikacijas, kurios savo veikimu būtų analogiškos desktopinėms aplikacijoms, nėra paprasta, nes desktopinės aplikacijos gali naudoti dideles “vietines” duomenų bazines, siekiant sumažinti reikalingų duomenų perduoti ir gauti kiekį, kai tuo tarpu AJAX pagrindu veikianči programinė įranga paprastai apsieičia su serveriu mažomis duomenų “porcijomis” bei tuos duomenis “cashuoja”. Todėl, kai kada, kai reikia realiu laiku perduoti didelius duomenų srautus, gali būti neįmanoma atkartoti kompiuterio pagrindu veikiančios programinės įrangos greitį ir efektyvumą. Lengviausia atkartoti internete tokią kompiuterio pagrindu veikiančią programinę įrangą, kuri naudoja didelę duomenų bazę, bet jos nereikia visos klientui dirbant su programine įranga. Tokiu atveju internete reikia pasiekti tiksliai vartotojo sąsajos patogumą ir interaktyvumą.

Daugelis turtingos vartotojo sąsajos programinės įrangos naudoja daugiau interneto ryšio srauto nei analogiška standartinė interneto naršyklės pagrindu veikianči programinė įranga. Todėl tokia programinė yra labai tinkama kurti vidinėms organizacijų informacinėms sistemoms, nes tinkle pralaidumas šiuo atveju nėra problema.

Turtingos sąsajos interneto aplikacijas lengva sukurti kaip atsvarą jau egzistuojančiai kompiuterio pagrindu veikiančiai programinei įrangai, kaip pvz. elektroninio pašto klientinę programinę įrangą ar ataskaitas teikiančią programinę įrangą. Tačiau, standartinė interneto naršyklės pagrindu veikiančiai programinei įrangai, kaip elektroninė parduotuvė ar katalogas, sukurti turtingos sąsajos programinę įrangą dažniausiai nėra paprasta.

2.7 Sritis, kuriose didžiausios Ajax technologijų perspektyvos

Atlikus bendrą analizę, galima išskirti sekančias sritis, kur Ajax technologijų pritaikymas turi prasmę žiūrit iš verslo pusės perspektyvų:

1. **Programinės įrangos paslaugų teikėjai** (angl. ASP - Application Service Provider). Šiuo metu pasaulyje yra gana populiarus verslo modelis teikti programinės įrangos nuomos paslaugas internetu. Tarp tokių teikiamų paslaugų vertėtų paminėti ne tik GMail ir Yahoo!Mail, bet ir Salesforce.com, openair.com, kayak.com ir kitas. Pagrindinis argumentas už Ajax technologijų naudojimą šioje srityje yra tai, kad jų naudojimas gali pagreitinti darbą su teikiama programine įranga, ir tai sąlygoti vartotojų sutaupyta laiką bei padidėjusį efektyvumą, dirbant su programine įranga. Be to, nederėtų neįvertinti ir to, kad šių paslaugų vartotojai gali labai greitai pakeisti tokius paslaugus teikėjus, nes nereikia instaliuoti naują programinę įrangą. Be to, paprastai mokestis už tokias paslaugas yra mokamas už naudojimosi laiką, o atskirais atvejais ir visai nemokamas. Tai savo ruožtu taip pat skatina ir pačius programinės įrangos paslaugų teikėjus kuo greičiau tobulinti siūlomą savo programinę įrangą bei tuo pačiu suteikti didesnę patenkinimą vartotojams.
2. Nepriklausomi programinės įrangos kūrėjai (angl. Independent Service Vendor – ISV), kurie turi jau sukūrę interneto naršyklės pagrindu veikiančias vartotojo sąsajas savo sukurtai programinei įrangai. Be abejo, šioje srityje Ajax galimybės yra didelės, sukuriant naujas programinės įrangos sąsajas ir tobulinant jau esamas. Savo ruožtu, žinoma, Ajax technologijų didelę plėtrą gali sąlygoti vis stiprėjanti konkurencija bei didėjantys vartotojų lūkesčiai.
3. Nepriklausomi programinės įrangos kūrėjai (angl. Independent Service Vendor – ISV), kurie dar neturi sukūrę savo programinės įrangos atitikmenų ar plėtinių, veikiančių interneto naršyklės pagrindu. Ajax technologijos atveria galimybę sukurti „desktop“ programinės įrangos perkėlimą į interneto naršyklės pagrindu veikiančią programinę įrangą. Pirmieji šio proceso ženklai jau matyti, ir yra sukurtos pirmosios versijos programinės įrangos, kaip teksto redaktoriai, elektroninės lentelės ir kitos.
4. Elektroninė komercija (angl. E-commerce). Nors Ajax technologijos palengvina vartotojo “darbą” su programine įranga bei šiuo atveju padidina elektroninės

parduotuvės pardavimus, svarbu ir tai, kad elektroninės parduvotuvės vartotojo sąsaja atitiktų šiandienos reikalavimus ir tam tikra prasme madą, nes pirmas įspūdis bei tinklapio dizainas neretai lemia, ar klientas pirks šioje elektroninėje parduotuvėje, ar ne. Todėl labai tikėtina, kad Ajax technologijos ir toliau sparčiau plis elektroninės komercijos srityje.

5. Finansinių paslaugų sektorius. Finansinių paslaugų sektoriuje vis daugiau ir daugiau paslaugų teikiama per internetą bei šis kanalas jau daugeliui finansinių institucijų tapo pagrindiniu paslaugų teikimo kanalu. Tai turėtų neišvengiamai skatinti toliau vystyti ir sparčiai plėtoti interneto naršyklės pagrindu veikiančią programinę įrangą bei tuo pačiu diegti naujas Ajax technologijomis paremtas vartotojo sąsajas.
6. Bendruomenių internete formavimosi portalai. Ajax technologijos įgalina ištrinti ribą tarp forumų ir pokalbių internete realiu laiku, o tai skatina nuolat ieškoti naujų ir plėtoti esamas galimybes ir paslaugas. Be to, ši rinka yra labai dinamiškai ir sparčiai besivystanti, todėl labai tikėtina, kad atsiras naujų rinkos žaidėjų, siekiančių išnaudoti Ajax technologijų teikiamus pranašumus bei savo ruožtu skatins judėti į priekį ir dabar jau stiprias rinkoje pozicijas užėmusias bendroves.
7. Turinio teikėjai ir media bendrovės. Turinio teikėjai ir media bendrovės, kaip laikraščiai, žurnalai, interneto naujienų portalai nuolat ieško naujų formų ir galimybių labiau “išnaudoti” sukuriama turinį, todėl labai tikėtina, kad šios srities bendrovės skirs nemažą dėmesį ir Ajax technologijoms bei stengsis jas pritaikyti, suteikiant vartotojams didesnę pasitenkinimą ir galimybes.

Priede 1, aš pateikiu realius pavyzdžius, demonstruojančius pagrindines Ajax programinės įrangos vartotojo sąsajas ir naudojimo principus. Jas šioje dalyje nusprendžiau pademonstruoti atskirai, nes manyčiau taip jų naudojimas bus aiškesnis ir logiškesnis nei stengtis kuo daugiau Ajax galimybių įdėti į mano kuriamą „Prezentacija Online“ programinę įrangą, nes tos galimybės būtų tikrai galimybės dėl galimybių ir apčiuopiamos naudos vartotojams nesuteiktų. Be to, Priede 2, aš pateikiu Ajax interneto tinklapių architektūros analizę ir pademonstruoju realius pavyzdžius. „Prezentacija Online“ programinės įrangos sukūrimu, aš noriu pademonstruoti, kaip Ajax technologijų pagrindu galima kurti naujo tipo inovatyvią programinę įrangą, kurios prieš atsirandant Ajax technologijoms nebuvo įmanoma sukurti.

3 Projektinė dalis

3.1 Projektinė specifikacija

3.1.1 Sistemos paskirtis

Projektas „Online Presentacija“ yra skirtas sukurti programinę įrangą, įgalinančią demonstruoti paveikslukais perteikiamą informaciją internetu fiziškai nutolusiems vartotojams realiu laiku, nereikalaujant instaliuotis jokios kitos papildomos programinės įrangos, o tikrai naudojant įprastinę interneto naršyklės programinę įrangą. Sistema „Online Presentacija“ yra skirta naudoti šioms pagrindinėms vartotojų grupėms:

- Telepardavimų specialistams
- Nuotoliniam mokymui ir pagalbos teikimui
- Vykdyti telekonferencijas

Sukurta sistema įgalins telepardavimų, mokymų (konsultacijų) bei telekonferencijų metu ne tikrai perduoti informaciją žodžiu, bet ir pademonstruoti vizualiai prezentacijas realiu laiku ir sinchroniškai, nereikalaujant iš vartotojų instaliuotis ar būti susiinstaliavus kokią nors papildomą programinę įrangą, išskyrus įprastinę interneto naršyklę ir interneto ryšį.

Be to, pažymėtina, kad pati sukurta sistema bus savo esme demonstracinė sistema, įgalinanti pademonstruoti tokios sistemos sukūrimo galimybes ir įgalinanti panaudoti AJAX technologijas.

3.1.2 Projekto kūrimo pagrindas (pagrindimas)

Projektą vykdau aš pats - Kauno Technologijos Universiteto IFN4/1 grupės studentas Almantas Šilinskas. Be to, man šis projektas yra tuo pačiu ir Informacinių technologijų Magistro darbas.

3.1.3 Sistemos tikslai (paskirtis)

Pagrindinis kuriamos sistemos tikslas yra sukurti sistemą, įgalinančią demonstruoti prezentacijas sinchroniškai ir realiu laiku nutolusiems vartotojams, nereikalaujant iš vartotojų instaliuoti jokios kitos papildomos programinės įrangos, išskyrus turimą interneto naršyklę.

3.1.4 Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys

- Užsakovas.

Sistemos „Online prezentacija“ užsakovas yra UAB „MediaWorks“, kuri planuoja šią sistemą naudoti kaip prototipą ir galimybių analizę, kuriuos pagrindu vėliau galėtų būti sukurta komercinė programinė įranga, skirta teikti paslaugas potencialiems vartotojams.

UAB „MediaWorks“ asmuo, atsakingas už sistemos kūrimą ir tolimesnį jos vystymą, yra UAB „MediaWorks“ projektų vadovas Stasys Martišius.

- Pirkėjas.

Sistemos „Prezentacija Online“ pirkėjais bus pardavimus telefonu (angl. telesales) vykdančios įvairių bendrovių darbuotojų. Taip pat sistema bus naudojama ir telekonferencijų ir distanciniam mokymui vykdyti.

- Kiti sprendimus priimančiosios asmenys.

Kitus sprendimus, priimančiosios asmenys: UAB „MediaWorks“ vadovybė ir Kauno Technologijos Universiteto dėstytojai.

3.1.5 Vartotojai

Vartotojo kategorija	Sprendžiami uždaviniai	Patirtis dalykinėje srityje	Patirtis informacinėse technologijose	Papildomos charakteristikos	Prioriteto grupė
Administratorius	Sistemos administravimas ir sistemos vartotojų teisių nustatymas	Patyręs	Patyręs	Turi apsimokyti administruoti sistemą ir ją prižiūrėti.	Nesvarbūs
Prezentacijų teikėjas	Prezentacijų teikimas realiu laiku ir sinchroniškai	Patyręs	Naujokas	Turi parengti prezentaciją	Svarbiausi
Demonstracijos stebėtojai	Prezentacijų teikiamos prezentacijos stebėjimas	Naujokas	Naujokas	Turi priimti kvietimą peržiūrėti prezentaciją.	Svarbiausi

Lentelė 1 Sistemos vartotojai

3.1.6 Projekto apribojimai

3.1.6.1 Apribojimai sprendimui

Sistema turi būti dinamiška Nors sistema ir gali neturėti pilno funkcionalumo, bet turi veikti be sutrikimų.

3.1.6.2 Diegimo aplinka

Kuriama sistema bus diegiama tarnybinėje stotyje arba kitaip pavadinus - serveryje. Sistema nevaldys jokių mechaninių ar kitokių prietaisų. Pačios stoties charakteristikos priklausos nuo vartotojo kiekio orientuoto į sistemą. Kadangi tai bus daugiau tiriama sistema tai didelio pajėgumo serveris nėra būtinas. Minimalūs serverio reikalavimai : ne lėtesnis nei amd Athlon xp 1400+ procesorius, ne mažiau 1Gb atminties, bei ne mažiau 10 Gb kietojo disko talpos arba šiuos minimalius reikalavimus atitinkantis kitos architektūros serveris.

3.1.6.3 Bendradarbiaujančios sistemos

Sistema veiks kartu su duomenų baze, o kitų bendradarbiaujančių sistemų nebus.

3.1.6.4 Komerciniai specializuoti programų paketai

Į kuriama sistemą neplanuojama įtraukti jokių papildomų specializuotų komercinių programinių paketų. Duomenų valdymui bus naudojama Hibernate technologija (karkasas). Kadangi java yra nemokama, o dabar jau ir atvirojo kodo, bei turi daugybę nemokamų priemonių kūrimui nereikia naudoti jokių komercinių paketų.

3.1.6.5 Numatoma darbo vietos aplinka

Darbo vieta nėra tiksliai nustatoma, nes vartotojas galės tiek namų aplinkoje, tiek bibliotekoje naudotis kuriama sistema. Patikslinant ten kur yra prieiga prie interneto. Spausdintuvas nebus reikalingas. Vartotojo patogumui geriausiai turėti kuo didesnės raiškos vaizduoklį, kad galėtų matyti daugiau informacijos puslapyje vienu metu. Realiai sistema normaliai atliks funkcijas bet kokioje aplinkoje.

3.1.7 Sistemos kūrimo terminai

Stadija	Paleidimo data	Priežastys	Vėlavimo pasekmės
Sukurta sistema.	2006-06-02	1 sistema parengta testavimui	Vėluos kitų sistemų sukūrimas.
Sistemos ištestavimas	2006-11-14	Visos sistemos ištestuotos	Vėluos diegimas.
Diegimas ir paleidimas bandomajai eksploatacijai.	2006-12-20	Sistemos ištestuotos ir įdiegtos į serverį.	Vėluos galutinė sistemų paleidimo data.

Lentelė 2 Sistemos kūrimo terminai

3.1.8 Sistemos kūrimo biudžetas

Kadangi tai nėra realaus užsakymo projektas – biudžetą galima priskirti turimiems vidiniams resursams ir tariamai galima įvertinti kiek kainuotų realus projektas.

Išlaidos	Vienetas	Vienetų skaičius	Vieneto kaina, Lt	Viso, Lt
1. Žmonių ištekliai				
Analitikas	Darbo diena	30	100	3000
Programuotojas	Darbo diena	35	70	2450
Testuotojas	Darbo diena	10	70	700
Dizaineris	Darbo diena	5	60	300
Iš viso žmonių išteklių		85		6450
2 Įranga ir prekės				
Kompiuteris	Vienetas	1	3000	3000
Programinė įranga	Vienetas	1	2000	2000
Vieta tarnybinėje stotyje	Metai	1	200	200
Popierius	Vienetas	1	10	10
Rašymo priemonės	Rinkinys	1	100	100
Viso :				5310
3 Kiti kaštai ir paslaugos				
Patalpos (nuoma, elektra ir t.t)	Mėnesis	4	200	800

Toneris	Vienetas	1	200	200
Viso:				1000
4. Iš viso kiti kaštai ir paslaugos				1000
5. Viso tiesioginiai projekto kaštai				11760
Viso:				12760

Lentelė 3 Projekto biudžetas

3.1.9 Svarbūs faktai ir prielaidos

Analizuojant galimybes kurti planuojamą programinę įrangą, buvo nerasta atitinkama alternatyva rinkoje. Kadangi tai būtų kuriamas naujas produktas, kuriam nėra labai aiškiai apibrėžtų reikalavimų ir niekas nėra išdirbęs procesų bei pats projektas nemaža dalimi yra skirtas „ištestuoti“ dar neatrastas galimybes, tai suteikia projektui nemažą aspektą inovatyvumo ir rizikos.

Pagrindines kuriamos programinės įrangos alternatyvas galima būtų išskirti sekančias:

Prezentacijų pateikimą nerealiu laiku ir nesinchroniškai, t.y. demonstruotojas gali pateikti stebėtojų nuorodą, kur peržiūrėti ar iš kur parsisiųsti prezentaciją ir po to ją komentuoti, sakydamas „Ar matote, ar ne vienokią ar kitokią informaciją“. Toks būdas yra netinkamas telepardavimuose, nes didžioje dalis potencialių pirkėjų paprastai neperžiūri visai arba peržiūri po daugelio paraginimų ir atidėliojimu, kai tuo tarpu mano kuriama programinė įranga įgalins betarpiškai demonstruoti prezentaciją realiu laiku.

„Application sharing“. Šiuo metu yra keletas technologijų, kurios įgalina demonstruoti tam tikros programinės įrangos vaizdą nutolusio vartotojo ekrane. Tokiu būdu, iš principo, būtų galima perduoti visą Microsoft Powerpoint programinės įrangos arba Open Office Impress vaizdą kitame kompiuteryje. Tačiau tokia programinė įranga reikalauja labai didelio kompiuterinės įrangos našumo bei didelio interneto ryšio pralaidumo. Be visa to, ji dar reikalauja ir iš kliento susiinstaliuoti tam tikrą programinę specialią programinę įrangą ir stebėtojo programinėje įrangoje. Pažymėtina ir tai, kad paprastai ji instaliuojama „nesiafišuojant“ arba pabrėžiant, kad ji iš karto išinstaliuojama, kai baigiasi prezentacija.

Pagrindinis mano numatomos sukurti programinės įrangos pranašumas ir būtų tas, kad jai nereikėtų jokios papildomos programinės įrangos tiek iš demonstruotojo, tiek iš „stebėtojo“ pusės ir jiems pakaktų naudoti tikrai įprastinėmis interneto naršyklėmis, netgi neprisirišant prie kurio nors vieno gamintojo.

3.1.10 Funkciniai reikalavimai

3.1.10.1 Veiklos sfera (The scope of the work)

3.1.10.2 Veiklos kontekstas

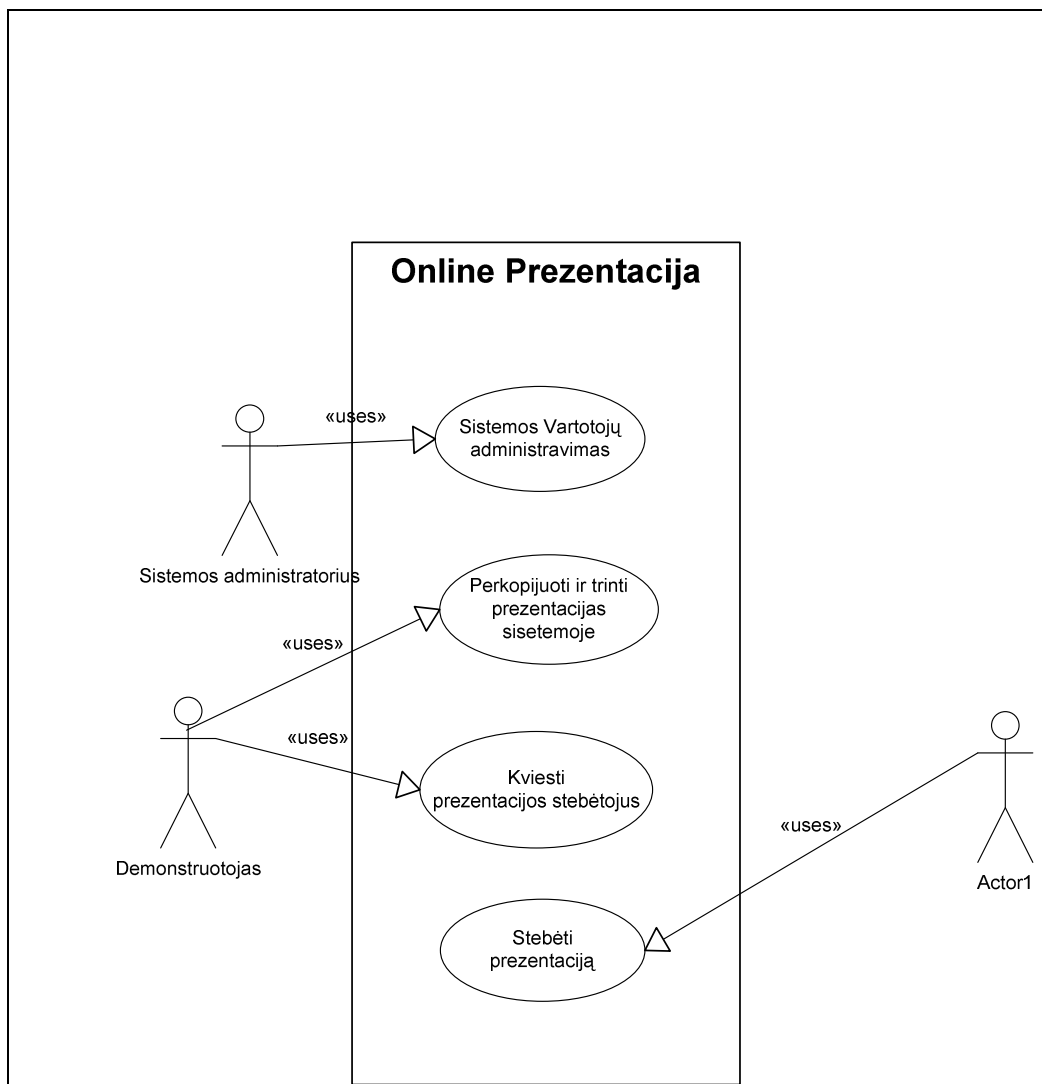
Kuriama sistema neturi ryšių su kitomis sistemomis.

3.1.10.3 Veiklos padalinimas.

Eil.nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Vartotojas, norintis teikti prezentacijas, užsiregistruoja sistemoje.	Vartotojo informacija (In)
2	Prisiregistravusiam vartotojui išsiunčiamas elektroninis pranešimas apie sėkmingą registraciją	Prisiregistravusiam vartotojui siunčiama elektroninis laiškas (Out)
3	Vartotojas nukopijuoja Paveiksliukus į sistemą „Prezentacija Online“	Prezentacija perkopijuojamas ir iškonvertuojama į formatą, tinkamą demonstravimui (In)
4	Vartotojas, teikiantis prezentaciją, kviečia prie sistemos prisijungti ir stebėti prezentaciją prezentacijos stebėtojus	Išsiunčiamas laiškas / pranešimas, kaip prisijungti prie demonstruojamos prezentacijos (Out)
5	Prezentacijos teikėjas teikia norimą (-as) prezentaciją (-as)	Demonstruojama prezentacija (Out)

3.1.11 Produkto veiklos sfera

3.1.11.1 Sistemos ribos



Pav. 9. Vartotojų panaudojimo atvejų diagrama

3.1.12 Panaudojimo atvejų sąrašas

1. PANAUDOJIMO ATVEJIS: **Naujo vartotojo – prezentacijos teikėjo registracija**

Vartotojas/aktorius: Prezentacijos teikėjas

Aprašas: Naujas vartotojas prisiregistruoja, užpildydamas reikalaujamos formos anketą.

Prieš sąlyga: Turi būti naujas vartotojas, prieš tai nesiregistravęs.

Sužadinimo sąlyga: Atsirado naujas asmuo, norintis naudotis sistema.

Po-sąlyga: Naujai užregistruota anketa.

2. PANAUDOJIMO ATVEJIS: **Registracijos patvirtinimo gavimas**

Vartotojas/aktorius: Prezentacijos teikėjo registracijos patvirtinimas

Aprašas: Sėkmingai užsiregistravęs sistemos vartotojas, kaip prezentacijos teikėjas, gauna elektroninio pašto žinutę apie sėkmingą registraciją ir demonstruotojo teisių suteikimą

Prieš sąlyga: Vartotojas nėra prieš prisiregistravęs prie sistemos

Sužadinimo sąlyga: Vartotojas pateikia visus reikalingus registracijai duomenis

Po-Sąlyga: Vartotojas turi teisę teikti prezentacijas

3. PANAUDOJIMO ATVEJIS: **Vartotojas perkelia parengtą prezentaciją į „Prezentacija Online“ sistemą**

Vartotojas/aktorius: Prezentacijos teikėjas

Aprašas: Prezentacijos teikėjas perkelia prezentaciją paveikslukus į „Prezentacija Online“ sistemą.

Prieš sąlyga: Tai gali padaryti tiktai vartotojai, turintys teisę teikti prezentacijas.

Sužadinimo sąlyga: Prezentacijos teikėjas, paspaudžia nuorodą, skirtą naujos prezentacijos įkėlimui.

Po-sąlyga: Prezentacija parengta demonstravimui.

4. PANAUDOJIMO ATVEJIS: **Kviečiami vartotojai, stebėti prezentaciją**

Vartotojas/aktorius: Prezentacijos teikėjas

Aprašas: Prezentacijos teikėjas pakviečia norimus vartotojus stebėti prezentaciją ir suteikia jiems tokias teises

Prieš sąlyga: Nėra

Sužadinimo sąlyga: Prezentacijos teikėjas įveda reikalingą informaciją apie norimą pakviesti stebėti prezentaciją vartotoją.

Po-sąlyga: Prie sistemos prisijungia vartotojai, norintys stebėti prezentaciją.

5. PANAUDOJIMO ATVEJIS: **Prezentacijos teikimas**

Vartotojas/aktorius: Prezentacijos teikėjas

Aprašas: Prezencacijos teikėjas teikia prezentaciją.

Prieš sąlyga: Prie kanalo (prezencacijos teikimo) prisijungę pakviesti prezencacijos stebėtojai.

Sužadavimo sąlyga: Prezencacijos teikėjas pradeda demonstruoti norimas skaidres.

Po-sąlyga: Pademonstruota norimiems stebėtojams prezentacija.

6. PANAUDOJIMO ATVEJIS: **Prezencacijos stebėjimas**

Vartotojas/aktorius: Prezencacijos stebėtojas.

Aprašas: Prezencacijos stebėtojas stebi prezentaciją.

Prieš sąlyga: Prezencacijos stebėtojas yra pakviestas stebėti prezentaciją ir prisijungia prie norimos stebėti prezencacijos sistemoje „Prezencacija Online“

Sužadavimo sąlyga: Prezencacijos stebėtojas prisijungia prie „Prezencacija Online“ sistemos ir Prezencacijos teikėjas pradeda demonstruoti norimas skaidres.

Po-sąlyga: Peržiūrėta prezentacija

7. PANAUDOJIMO ATVEJIS: **Vartotojų valdymas**

Vartotojas/aktorius: Administratorius

Aprašas: Administratorius gali pašalinti vartotoją, laikinai uždrausti jam teisę teikti prezencijas.

Prieš sąlyga: Kažkuris vartotojas pažeidė nustatytus apribojimus.

Sužadavimo sąlyga: Vartotojas nesumokėjo už sistemos teikiamas paslaugas arba

pažeidė sistemos vartojimo sąlygas.

Po-sąlyga: Atlikti pakeitimai vartotojuose.

3.1.13 Funkciniai reikalavimai ir reikalavimai duomenims

3.1.13.1 Funkciniai reikalavimai

<u>Reikalavimas#:</u>	1	<u>Reikalavimo tipas:</u>	1	<u>Panaudojimo atvejis #:</u>	1
<u>Aprašymas:</u>	Galimybė užsiregistruoti sistemoje kaip Prezentacijų teikėju.				
<u>Pagrindimas:</u>	Sistema skirta prezentacijų teikimui ir reikalinga prezentacijų teikėjų registravimas bei minimalios informacijos apie vartotoją - Prezentacijos teikėją turėjimas.				
<u>Šaltinis:</u>	Užsakovas.				
<u>Tikimo kriterijus:</u>	Sistema leidžia naujiems vartotojams tapti prezentacijų teikėjais.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>	Nėra				
<u>Istorija:</u>	Užregistruotas 2006-03-15				

<u>Reikalavimas#:</u>	2	<u>Reikalavimo tipas:</u>	1	<u>Panaudojimo atvejis #:</u>	2
<u>Aprašymas:</u>	Reikia prisiregistravusiam vartotojui, kaip prezentacijos teikėjui, išsiųsti elektroninį pranešimą apie sėkmingą registraciją bei kitą papildomą informaciją.				
<u>Pagrindimas:</u>	Reikalinga galimybė nusiųstu patvirtinimą apie sėkmingą registraciją ir papildomą informaciją, kaip naudotis sistema. Be to, būtina įsitikinti, ar įvestas elektroninio pašto adresas yra				

	teisingas.		
<u>Šaltinis:</u>	Užsakovas		
<u>Tikimo kriterijus:</u>	Išsiunčiamas elektroninis pranešimas apie sėkmingą registraciją.		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>	Nėra		
<u>Istorija:</u>	Užregistruotas 2006-03-15		

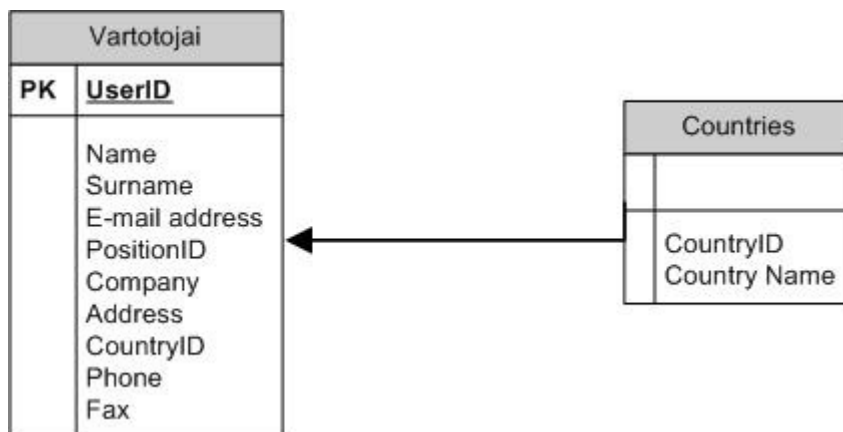
<u>Reikalavimas#:</u>	3	<u>Reikalavimo tipas:</u>	1	<u>Panaudojimo atvejis #:</u>	3
<u>Aprašymas:</u>	Reikia leisti vartotojui, turinčiam prezentacijos teikimo teises, įkelti paveikslukus į „Prezentacija Online“ sistemą.				
<u>Pagrindimas:</u>	Prezentacijos teikėjas neturės, ką demonstruoti, jeigu nebus į sistemą „Prezentacija Online“ įkėlęs jokios prezentacijos.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Sistema leidžia vartotojui įkelti naują prezentaciją, kada tik jis tą pageidauja daryti, jeigu yra neviršijęs Sistemos administratoriaus nustatyto galimų prezentacijų kiekio.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>	Nėra				
<u>Istorija:</u>	Užregistruotas 2006-03-15				

<u>Reikalavimas#:</u>	4	<u>Reikalavimo tipas:</u>	1	<u>Panaudojimo atvejis #:</u>	7
<u>Aprašymas:</u>	Administratorius turi galimybę uždrausti prezentacijos teikėjui				

	turėti teisę teikti prezentacijas.		
<u>Pagrindimas:</u>	Vartotojas yra nesumokėjęs už sistemos teikiamas paslaugas arba pažeidęs sistemos vartojimo taisykles ir reglamentą.		
<u>Šaltinis:</u>	Užsakovas		
<u>Tikimo kriterijus:</u>	Administratoriui leidžiama uždrausti prezentacijos teikėjui teikti prezentacijas.		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	4
<u>Priklausomybės:</u>	2 reikalavimas	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>	Nėra		
<u>Istorija:</u>	Užregistruotas 2006-03-15		

<u>Reikalavimas#:</u>	5	<u>Reikalavimo tipas:</u>	1	<u>Panaudojimo atvejis #:</u>	4
<u>Aprašymas:</u>	Prezentacijos teikėjas, gali kviesti stebėtojus trimis būdais: <ul style="list-style-type: none"> ▪ Nusiųsdamas el. pašto adresu nuorodą, kaip prisijungti prie „Online Prezentacija“ sistemos ▪ Nukopijuodamas interneto nuorodą ir pateikdamas ją vartotojui per pasirinktą interneto tekstinių žinučių sistemą ▪ Suteikdamas vartotojui prisijungimo sesijos kodą prisijungti pagrindiniame sistemos tinklapio puslapyje 				
<u>Pagrindimas:</u>	Prezentacijos teikėjas turi pakviesti prezentacijos stebėtojus, kad jie žinotų kaip prisijunti prie prezentacijos stebėjimo ir galėtų ją stebėti				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Sistema leidžia prisijungti prie prezentacijos ir ją stebėti, kai to pageidauja prezentacijos teikėjas.				
<u>Užsakovo tenkinimas:</u>	4	<u>Užsakovo netenkinimas:</u>	3		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>	Nėra				
<u>Istorija:</u>	Užregistruotas 2006-03-10				

3.1.14 Reikalavimai duomenims



Pav. 10. Koncepcinis duomenų bazės modelis

3.1.15 Nefunkciniai reikalavimai

3.1.15.1 Reikalavimai sistemos išvaizdai (Look and feel)

Sąsaja turi pirmiausiai būti patogi vartotojui. Ji turi būti novatoriška ir graži. Bet papildomi pagražinimai neturi užgožti pagrindinių funkcijų aiškumo. Sąsaja neturi būti įkyri. Kadangi ši sistema orientuota ne tik jaunimui – ji turi būti solidi, kad neatrodytų per daug vaikiška. Spalvos turi derintis ir nesipjauti tarpusavyje. Sąsaja taip pat turėtų būti sąveikaujanti ir bendradarbiaujanti su vartotoju. Puslapiai turi būti dinamiški, kad kuo mažiau laukti naujų puslapių atsidarymo. Statinė informacija neturėtų būti perkraunama pastoviai.

3.1.15.2 Reikalavimai panaudojamumui (Usability)

Sistema bus valdoma pele ir klaviatūra. Naudojimas bus paprastas, nereikės ilgai galvoti ką daryti, todėl apsimokymas (jei jis tik bus reikalingas) naudotis sistema bus lengvas ir greitas. Sistema turės aiškius meniu pavadinimus ir ikonas, todėl nesunku bus nuspėti, kur kas ką

reiškia. Vartotojui padarius klaidą, sistema stengsis nuspėti ką norėjo vartotojas padaryti ir informuos jį apie tai. Sistema veiks lietuvių kalba su galimybe ateityje pridėti dar kelias kalbas.

3.1.15.3 Reikalavimai vykdymo charakteristikoms

Komandų vykdymo greitis priklausys nuo prisijungusių vartotojų skaičiaus, tarnybinės stoties apkrautumo ir galingumo bei tinklo apkrautumo. Kadangi sistema pradžioje nebus komercinė – spartos problemos gali būti jaučiamos. Duomenų bazės apimtis priklausys nuo vartotojų skaičiaus, leidžiamų prezentacijų skaidrių bei jų dydžių skaičiaus ir laiškų skaičiaus vienam vartotojui, kuri sistema saugos. Kalbėjimo realiu laiku sparta priklausys nuo tinklo apkrautumo.

3.1.15.4 Reikalavimai veikimo sąlygoms (Operational)

Sistema veiks tarnybinėje stotyje, todėl jos veikimo sąlygos sutampa su tarnybinės stoties veikimo sąlygomis. Vartotojai galės naudotis sistema įprastais kompiuteriais, turinčiais interneto prieigą.

3.1.15.5 Reikalavimai saugumui

Sistemoje saugoma informacija yra konfidenciali, todėl jos negali matyti pašaliniai asmenys neprisiregistravę prie sistemos. Sistema turi teisingai priimti pateikiamus duomenis ir juos išsaugoti. Vartotojui panorėjus panaikinti savo duomenis arba ištrinti turimus prezentacijų failus – sistema turi leisti tą padaryti.

3.1.15.6 Kultūriniai – politiniai reikalavimai

Sistemai nekeliama jokie esminiai kultūriniai – politiniai reikalavimai, išskyrus tai, kad sistema turi palaikyti daugiakalbiškumą.

3.1.16 Projekto išeiga

3.1.16.1 Atviri klausimai

Atvirų klausimų nėra.

3.1.17 Egzistuojantys sprendimai

3.1.17.1 Pagamintos sistemos, kurios gali būti nupirktos

Analogišką sistemą, kuri atitiktų keliamus reikalavimus, rasti nepavyko.

3.1.17.2 Pagaminti komponentai, kuria gali būti panaudoti

Pagamintų komponentų neaptikta, kuriuos konkrečiai būtų galima panaudoti sistemoje. Reikalingas papildomas komponentų pakeitimas.

3.1.17.3 Galimas pakartotinis panaudojimas

Sistema galės būti pakartotinai naudojama neribotą laiko ir kartų skaičių. Be to, tikimasi, kad šio projekto metu sukurtus kai kuriuos komponentus bus galima sėkmingai panaudoti ir kitose kuriamose sistemose.

3.1.18 Naujos problemos

3.1.18.1 Problemos diegimo aplinkai

Sistema diegiama lengvai ir problemų diegimas neturėtų kelti. Diegimo metu gali išaiškėti faktas, kad tarnybinė stotis yra per silpna numatytam vartotojų skaičiui.

3.1.18.2 Įtaka jau kitoms įdiegtoms sistemoms

Sistema realiai neturės įtakos kitoms įdiegtoms sistemoms, nebent jau būtų naudojama tokia pati duomenų bazė ar web tarnybinė stotis. Tada juos reiktų paruošti naujos sistemos darbu ir suteikti atitinkamas teises. Konfliktų kilti neturėtų.

3.1.18.3 Neigiamas vartotojų nusiteikimas

Vartotojams gali nepatikti sistemos išvaizda, ar jiems gali pritrūkti kelių jiems įprastų kitose sistemose esančių funkcijų. Spartos bei stabilumo problemos taip pat aktualios. Jie gali nenorėti naudotis sistema. Taip nutikus būtina surinkti vartotojų nepatogumus ir juos ištaisyti bei informuoti vartotojus, kad diskomfortas ištaisytas.

3.1.18.4 Kliudantys diegimo aplinkos apribojimai

Sistemai diegti planuojama tarnybinė stotis gali būti prijungta prie per lėto ir nestabilaus interneto, kad būtų galima sistemoje naudoti realaus laiko pokalbių funkciją. Tarnybinė stotis taip pat gali būti nepakankamai apsaugota nuo įsilaužimų arba būti nestabili savo darbu ir būti dažnai nepasiekiamas.

3.1.18.5 Galimos naujos sistemos sukeltos problemos

Sistemos didžiausia galima problema yra ta, kad vartotojai gali naudoti labai didelės apimties prezentacijų failus.

3.1.19 Uždaviniai

3.1.19.1 Sistemos pateikimo žingsniai (etapai)

Sistemos pateikimo žingsniai yra sekantys:

Eil. Nr.	Etapas	Rezultatai	Igyvendinimo data
1.	Reikalavimų specifikuavimas	Reikalavimų specifikacija	2006-03-20
	Architektūros specifikuavimas	Architektūros specifikacija	2006-04-25
	Detalios architektūros specifikuavimas	Detali architektūros specifikacija	2006-05-15
	Sistemos sukūrimas	Sukurta sistema	2006-09-25
	Sistemos testavimas, pataisymai	Testų – pataisymų rezultatai ir planai	2006-10-15
	Sistemos administratorių apmokymas	Vartotojo dokumentacija	2006-10-25
	Sistemos reklamavimas	Reklaminiai bukletai ir informacija internete	2006-11-15

Lentelė 4. sistemos pateikimo žingsniai

3.1.19.2 Vystymo etapai

Kiekviename vystymo etape bus reikalinga ta pati techninė įranga. Tai elementarus stalinis kompiuteris. Nuo jo spartos pačio projekto kūrimo sparta priklausys nedaug.

Pagrindiniai sistemos vystymo etapai yra pateikti žemiau:

Vystymo etapas	Kas vystoma?
1	Reikalavimų ir specifikacijos parengimas
2	Sistemos sukūrimas

3	Sistemos testavimas
4	Sistemos bandomoji eksploatacija.
5	Sistemos „produkcijos“ versijos įdiegimas.

Lentelė 5. Sistemos pateikimo etapai

3.1.20 Pritaikymas

3.1.20.1 Reikalavimai esamų duomenų perkėlimui

Kadangi kuriama sistema yra nauja, jokių duomenų perkelti į ją nereikia.

3.1.20.2 Reikalingas duomenų transformavimas perkeliant į naują sistemą

Kadangi kuriama sistema yra nauja, jokių duomenų transformuoti nereikia.

3.1.21 Rizikos

3.1.21.1 Galimos sistemos kūrimo rizikos

Rizika	Tikimybė	Padariniai
Blogai suprasti reikalavimai	0,2	Sukurta sistema neatlieka visų funkcijų arba atliks to, kas nebuvo numatyta.
Programuotojo liga, trauma	0,5	Darbų vėlavimas. Kokybės praradimas.
Nepavykusi bandomoji eksploatacija.	0,5	Tiriamųjų išvadų nepilnumas. Neteisingai įvertinta technologija.

Lentelė 6 Rizikos

3.1.21.2 Atsitiktinumų (rizikų) valdymo planas

Rizika	Sprendimo planas
Blogai suprasti reikalavimai	Reikalavimų papildymas, pakartotinė analizė. Trūkstumų panaudojimo atvejų įterpimas.
Nežinomos Hibernate ir AJAX technologijos gali sąlygoti sistemos su jomis nesukūrimą.	Didesnis laiko skyrimas technologijai perprasti bei sistemai realizuoti. Konsultacijos su šias technologijas išmanančiais specialistais bei reikalingos papildomos literatūros paieška.
Programuotojo liga, trauma	Surasti, kad galėtų padėti tęsti darbus.
Nepavykusi bandomoji eksploatacija.	Bandyti ieškoti naujų testuotojų, naujų tarnybinių stočių, jei pastarosios yra per silpnos.

Lentelė 7 Rizikų sprendimo planai

3.1.22 Kaina

Kadangi sistemą Almantas Šilinskas kuria kaip savo Magistrinio darbą, todėl sistema UAB „MediaWorks“ nekainuos nieko finansinių kaštų. Bendri sistemos kaštai yra sekantys:

Faktorius	Vieneto kaina	Kiekis	Kaina, Lt
Veiklos įvykių skaičius	80	11	880
Panaudojimo atvejų skaičius	250	10	2500
Funkcinių reikalavimų skaičius	100	13	1300
Nefuncinių reikalavimų skaičius	60	8	480
Viso:			5160

Lentelė 8 Kainos skaičiavimas 1 būdu

Išlaidos	Vienetas	Vienetų skaičius	Vieneto kaina, Lt	Viso, Lt
1. Žmonių ištekliai				
Analitikas	Darbo diena	30	100	3000
Programuotojas	Darbo diena	35	70	2450
Testuotojas	Darbo diena	10	70	700
Dizaineris	Darbo diena	5	60	300
Iš viso žmonių išteklių	85			6450

Lentelė 9 Kainos skaičiavimas 2 būdu

3.1.23 Vartotojo dokumentacija ir apmokymas

Reikalingas bus tik administratorių apmokymas, kaip administruoti sistemą. Planuojama, kad vartotojai išmoks naudotis sistema patys, nes bus pateikta detali informacija, kaip naudotis sistema ir išsamus vartotojo vadovas.

Dokumentacijos lygis	Ar dalyvaus vartotojai ?	Kas atsakingas?	Pateikimo forma
Technologinis	Ne	Sistemas kūrėjas	Elektroninė-popierinė
Prezentacijos teikėjo demonstruojamas	Taip	Sistemas kūrėjas	Elektroninė
Prezentacijos stebėtojas	Taip	Sistemas kūrėjas	Elektroninė
Administratoriaus	Ne	Sistemas kūrėjas	Elektroninė-popierinė
Visos sistemas	Ne	Sistemas kūrėjas	Elektroninė
Reklaminis	Ne	Sistemas kūrėjas	Popierinė-elektroninė

Lentelė 10 Dokumentacija

3.1.24 Perspektyviniai reikalavimai

Jei sistema turės pasisekimą ir vartotojai pageidaus, kad ji būtų išplėsta naujomis funkcijomis – reikalinga, kad geriausia technologija realizuota sistema būtų lanksti išplėtimams. Numatoma, kad gali reikėti, esant techninėms galimybėms, suteikti galimybę demonstruoti kitų tipų dokumentų (Adobe Acrobat, Microsoft Word ir Excel) prezentacijas bei suteikti galimybę demonstruoti prezentacijas visiems atskiros sesijos vartotojams. Paliekamas ateičiai ir sistemos daugiakalbiškumo klausimas.

3.1.25 Idėjos sprendimams

Reikia išspręsti klausimą ar reikalinga koduoti vartotojų teikiamas prezentacijas realiu laiku. Taip pat aktualios ir spartos problemos kai sistemos vartotojų skaičius sparčiai išauga. Be to, aktualios yra ir prezentacijų skaidrių skaičiaus, dydžio ir išsaugojimo būdo parinkimo problemos.

3.2 Sistemos paketai

3.2.1 HIBERNATE paketas

Hibernate paketas yra paketas, atsakingas už duomenų operacijas su duomenų baze.

Hibernate.User	
Klasifikacija	Klasė.
Apibrėžimas	Vartotojo atributai ir jų pasiekimai rašymo metu.
Atsakomybės	Ši klasė skirta aprašyti vartotojui.
Struktūra	<p>Klasė susideda iš 2 funkcijų :</p> <ul style="list-style-type: none">• Funkcija user() Funkcijos paskirtis yra pateikti 1 useri pagal vardą ir slaptažodį.• Funkcija getUsers() Funkcijos paskirtis yra atrinkti vartotojus, kurie yra nepatvirtintoje būsenoje <p>Atributus paveldi iš AbstractUser klasės</p>
Sąveikavimas	Sąveikauja su AbstractUser klase
Resursai	Karkaso bibliotekos.
Skaičiavimai	Nėra skaičiavimų.
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

Hibernate.SessionFactory	
Klasifikacija	Klasė.
Apibrėžimas	Duomenų operacijos su duomenų baze, prisijungimo palaikymas ir t.t
Atsakomybės	Ši klasė yra pagrindinė duomenų valdymo klasė, kuri sujungia duomenų lygį su kitais lygiais. Klasė atsako už prisijungimą, transakcijų bei sesijų valdymą. Vartotojas neprieina prie šių klasių tiesiogiai.
Apribojimai	Apribojimai prisijungimo baseinui (pool) yra 5, prisijungimų skaičius prie duomenų bazės yra 10, spartinimo atminties lygis „level 2“.Visos Hibernate karkaso išlygos galimos šioje klasėje. Pavadinsime apibendrintai Hibernate exception . Tai gali būti neprisijungimo prie db klaida, rašymo klaida, transakcijos klaida, duomenų trūkumo klaida ir t.t.
Struktūra	<p>Klasė susideda iš 8 funkcijų ir 6 atributų:</p> <ul style="list-style-type: none"> • Funkcija void init(String path) Funkcijos paskirtis yra inicializuoti klasę, nurodant kelią prie reikiamų xml failų.Galima PathNotFound išlyga. • Funkcija Session currentSession() Funkcijos paskirtis yra atidaryti dabartinę sesiją. Galima Hibernate exception. • Funkcija void closeSession() Funkcijos paskirtis yra uždaryti aktyvią sesiją. Galima Hibernate exception • Funkcija void beginTransaction() Funkcijos paskirtis yra pradėti transakciją. Galima Hibernate exception • Funkcija void commitTransaction() Funkcijos paskirtis yra įvykdyti transakciją. Hibernate exception • Funkcija void rollBackTransaction() Funkcijos paskirtis yra atšaukti padarytus pakeitimus. Galima Hibernate exception. • Funkcija String getObjectPath()

	<p>Funkcijos paskirtis yra gauti spartinančiojo Hibernate užkrovimo failo vietą.</p> <ul style="list-style-type: none"> • Funkcija void setObjectPath(String objectPath) <p>Funkcijos paskirtis yra nustatyti Hibernate spartinančio užkrovimo failo vietą.</p> <ul style="list-style-type: none"> • Atributas String CONFIG_FILE_LOCATION <p>Atributas saugo Hibernate valdymo failo vietą.</p> <ul style="list-style-type: none"> • Atributas ThreadLocal threadSession <p>Atributas saugo sesijos gijų informaciją.</p> <ul style="list-style-type: none"> • Atributas ThreadLocal threadTransaction <p>Atributas saugo sesijos transakcijų gijų informaciją.</p> <ul style="list-style-type: none"> • Atributas Configuration cfg <p>Atributas saugo užkrautą Hibernate konfigūraciją iš failo</p> <ul style="list-style-type: none"> • Atributas SessionFactory sessionFactory <p>Atributas saugo sesijos kūrimo fabriko objektą.</p> <ul style="list-style-type: none"> • Atributas String objectPath <p>Atributas saugo spartinančiojo Hibernate užkrovimo failo vietą</p>
Sąveikavimas	Sąveikauja su User klase
Resursai	Karkaso bibliotekos.
Skaičiavimai	Nėra skaičiavimų.
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

3.2.2 UTIL paketas

Util paketas apima pagalbinės sistemos priemones, pavyzdžiui elektroninio pašto žinučių siuntimą.

Util.SendMail	
Klasifikacija	Klasė.
Apibrėžimas	Elektroninio pašto žinučių siuntimas.
Atsakomybės	Klasė suteikia galimybę siųsti elektroninio pašto žinutes naujai prisiregistravusiems sistemos vartotojams – prezentacijų teikėjams ir pakviestiems prezentacijų stebėtojams
Apribojimai	Apribojimų nėra.
Struktūra	<p>Klasė susideda iš penkių funkcijų ir devynių atributų.</p> <ul style="list-style-type: none"> • Funkcija connect () Funkcijos paskirtis atsidaryti socketą ir nusistatyti „streamus“. • Funkcija doCommand(String) Funkcijos paskirtis yra siųsti komandą į socketą. • Funkcija getResponse() Funkcijos paskirtis yra gauti atsakymą iš elektroninio pašto serverio. • Funkcija sendMessage() Funkcijos paskirtis yra pasiųsti elektroninio pašto žinutes. • Funkcija toBase64() Funkcijos paskirtis yra konvertuoti stringą į Base64 kodą. • Atributas host Jo paskirtis yra saugoti pasirinkto mail serverio adresą. • Atributas recipient Jo paskirtis yra saugoti vartotojo, kuriam siunčiamas elektroninis pranešimas vardą. • Atributas sender Jo paskirtis yra saugoti elektroninės žinutės siuntėjo vardą/pavadinimą. • Atributas to

	<p>Jo paskirtis yra saugoti elektroninio pašto adresą, kuriam siunčiamas pranešimas.</p> <ul style="list-style-type: none"> • Atributas message <p>Jo paskirtis yra saugoti siunčiamo elektroninio pranešimo teksto turinį.</p> <ul style="list-style-type: none"> • Atributas sessionSock <p>Jo paskirtis yra saugoti socketo duomenis.</p> <ul style="list-style-type: none"> • Atributas inReader <p>Jo paskirtis yra skaityti duomenis iš socketo.</p> <ul style="list-style-type: none"> • Atributas outStream <p>Jo paskirtis yra rašyti duomenis į socketą.</p> <ul style="list-style-type: none"> • Atributas outCStream <p>Jo paskirtis rašyti elektroninio pašto žinutės turinį.</p>
Sąveikavimas	Klasė suteikia galimybę siųsti elektroninius pranešimus ir sąveikauja su Client klase.
Resursai	Karkaso bibliotekos.
Skaičiavimai	Tai paprasta duomenų įvedimo-redagavimo klasė. Jokių skaičiavimo algoritmų realiai nėra.
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

3.2.3 WEB paketas

Web paketas apima serverio ir interneto karkaso klases, puslapių užkrovimą, vykdo klientų prisijungimą bei komandų parsinimą.

web.Server	
Klasifikacija	Klasė.
Apibrėžimas	Klasė jungianti visus kanalus sistemoje, taip pat saugo pakvietimus, bei unikalius id.
Atsakomybės	Klasė atsakinga už visos sistemos kanalus ir pakvietimus prezentacijoms.
Apribojimai	-
Struktūra	<p>Klasė susideda iš 11 funkcijų ir 3 atributų:</p> <ul style="list-style-type: none"> • Funkcija int getId () Funkcijos paskirtis yra grąžinti unikalų id. • Funkcija int getCurrentId () Funkcijos paskirtis yra grąžinti esamą unikalų id. • Funkcija Invitation getInvitation (String invId) Funkcijos paskirtis yra gauti pakvietimą, kurio identifikatorius yra invId • Funkcija void addChannel(Channel ch) Funkcijos paskirtis yra sukurti naują kanalą serveryje. • Funkcija void addInvitation(String id) Funkcijos paskirtis yra sukurti naują kvietimą su identifikatoriumi id • Funkcija Channel getChannel(Client owner) Funkcijos paskirtis yra gauti savininko owner kanalą. • Funkcija void addInvitation(Invitation inv) Funkcijos paskirtis yra sukurti naują kvietimą inv • Funkcija void removeInvitation (Invitation inv) Funkcijos paskirtis yra pašalinti inv kvietimą • Funkcija void removeInvitation (Invitation inv) Funkcijos paskirtis yra pašalinti inv kvietimą. Galima NamingException, kuri yra tiesiog ignoruojama.

	<ul style="list-style-type: none"> • Funkcija Client getChannelByOwner (Client client) Funkcijos paskirtis yra gauti kanalą palei savininką client. • Funkcija HashMap getInvitations () Funkcijos paskirtis yra gauti visus pakvietimus. • Atributas HashMap channels Atributas saugo serverio kanalus. • Atributas HashMap invitations Atributas saugo informaciją apie serverio pakvietimus. • Atributas int uniqueId Atributas saugo informaciją apie serverio unikalų identifikatorių pakvietimams.
Sąveikavimas	Klasę naudoja Client , Proxy klasės.
Resursai	-
Skaičiavimai	Skaičiavimai neatliekami
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

web.Invitation	
Klasifikacija	Klasė.
Apibrėžimas	Klasė saugo pakvietimų informaciją
Atsakomybės	Klasė atsakinga kuriamų pakvietimų informacijos saugojimą ir gavimą.
Apribojimai	-
Struktūra	<p>Klasė susideda iš 10 funkcijų ir 5 atributų:</p> <ul style="list-style-type: none"> • Funkcija Boolean getActive() Funkcijos paskirtis yra grąžinti klasės atributo active reikšmę • Funkcija void setActive(Boolean active)

	<p>Funkcijos paskirtis yra klasės atributo active reikšmei nustatyti.</p> <p>Visiems kitiems atributams yra tokie patys metodai su get ir set pradžiomis.</p> <ul style="list-style-type: none"> • Atributas String id Atributas saugo pakvietimo identifikatoriaus reikšmę • Atributas Channel channel. Atributas saugo informaciją apie pakvietimo kanalą. • Atributas String clientName() Atributas saugo informaciją apie kviečiamo kliento vardą • Atributas String password Atributas saugo informaciją apie kviečiamo kliento slaptažodį • Atributas Boolean active Atributas saugo informaciją apie pakvietimo aktyvumą.
Sąveikavimas	Klasę naudoja Client,Proxy,Server klasės.
Resursai	-
Skaičiavimai	Skaičiavimai neatliekami
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

3.2.4 PROXY paketas

Proxy paketas vykdo funkcijas, susijusias su kliento dalimi, demonstruojamų prezentacijų kopijavimą ir kitus veiksmus. Be to, jis palaiko Ajax klientų veikimą, juos kuria ir naikina.

web.proxy.Channel	
Klasifikacija	Klasė.

Apibrėžimas	Skaidrių rodytojo ir stebėtojų ryšio kanalas.
Atsakomybės	Klasė suteikia galimybę klientui (skaidrių rodytojui) žinoti visus jo prezentacijos stebėtojus, siųsti jiems komandas.
Apribojimai	Klientas galės būti tik vieno kanalo savininkas.
Struktūra	<p>Klasė susideda iš 8 funkcijų ir 2 atributų:</p> <ul style="list-style-type: none"> • Funkcija Client getClient(String name) Funkcijos paskirtis yra gražinti kanalo vartotoją, kurio name yra paduodamas per parametą. • Funkcija void addClient(Client c) Funkcijos paskirtis yra įtraukti į kanalą vartotoją c. • Funkcija void setOwner(Client client) Funkcijos paskirtis yra nustatyti kanalo savininką į client • Funkcija Client getOwner() Funkcijos paskirtis yra gauti kanalo savininką. • Funkcija void sendToAll(String message) Funkcijos paskirtis yra išsiųsti komandą message visiems kanale esantiems vartotojams ir pačiam savininkui. • Funkcija void sendToAllExceptOwner (String message) Funkcijos paskirtis yra išsiųsti komandą message visiems kanale esantiems vartotojams išskyrus kanalo savininką. • Funkcija void sendAllViewersToOwner () Funkcijos paskirtis yra parodyti pristatytojui skaidrių, kurie vartotojai prisijungę prie jo rodomos sesijos. • Funkcija void leave (Client client) Funkcijos paskirtis yra vartotojui client palikti kanalą. • Atributas Vector clients

	<p>Atributas saugo kanalo stebėtojų sąrašą.</p> <ul style="list-style-type: none"> • Atributas Client owner <p>Atributas saugo informaciją apie kanalo savininką (tą, kas sukūrė kanalą)</p>
Sąveikavimas	Klasę naudoja Client ir Server klasės.
Resursai	-
Skaičiavimai	Skaičiavimai neatliekami
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

web.proxy.Proxy	
Klasifikacija	Klasė.
Apibrėžimas	Klasė jungianti Ajax klientus.
Atsakomybės	Klasė atsakinga už Ajax klientų palaikymą sujungtais , tikrinimą ar jie aktyvus, ryšio tikrinimą.
Apribojimai	-
Struktūra	<p>Klasė susideda iš 7 funkcijų ir 1 atributo:</p> <ul style="list-style-type: none"> • Funkcija void init() <p>Funkcijos paskirtis yra paleisti sesijos palaikytojo giją.</p> <ul style="list-style-type: none"> • Funkcija void leaveProxy (String sesionId) <p>Funkcijos paskirtis yra palikti sesiją vartotojui, kurio sesijos identifikatorius yra sessionId.</p> <ul style="list-style-type: none"> • Funkcija Client getClient(String sessionId) <p>Funkcijos paskirtis yra grąžinti klientą palei jo sesijos identifikatorių sessionId.</p> <ul style="list-style-type: none"> • Funkcija void addClient(String sessionId,Client c)

	<p>Funkcijos paskirtis yra įtraukti naują klientą c su sesijos id sessionId</p> <ul style="list-style-type: none"> • Funkcija Client join(String sessionId, String identity,boolean invite) Funkcijos paskirtis yra bandyti prijungti naują klientą, kurio sesijos id yra sessionId, identifikuojamas pagal identity ir žiūrima ar tai pakviestas klientas ar ne pagal invite. • Funkcija Vector pingClients() Funkcijos paskirtis yra gauti nebeaktyvių klientų sąrašą, meta ConcurrentModificationException, kuri yra apdorojama kitoje klasėje. • Vidinė klasė SessionKeeper Klasės paskirtis yra palaikyti ryšį tarp pristatytojo ir jo klientų. • Funkcija void run() Funkcijos paskirtis yra vykdyti pingClients funkciją, bet išmesti iš sąrašo neaktyvius vartotojus.Gali įvykti InterruptedException ir ConcurrentModificationException, išlygos neapdorojamos. • Atributas Hashtable sessions Atributas saugo sesijų lentelę.
Sąveikavimas	Klasę turi ryšį su Invitation bei Server klasėmis
Resursai	-
Skaičiavimai	Skaičiavimai neatliekami
Sąsaja	Sąsajos aprašymas atitinka komponento struktūrą, kurioje pateikti visi atributai, bei funkcijos su aprašymais.

3.3 Sistemos architektūra

Ši dalis skirta apibrėžti kuriamos sistemos „Online prezentacija“ architektūrai. Taip pat bus parodoma sistemos struktūra, išdėstymas. Šį dokumentą naudos sistemos programuotojas, kuris kurs sistemą. Dokumentas jam padės suprasti sistemos struktūrą ir jos suskirstymą į modulius (paketus) ir t.t. Taip pat šį dokumentą galės naudoti testuotojai, kuriems šis dokumentas padės geriau orientuotis sistemoje testuojant ją. Taip pat šį dokumentą galės naudoti žmonės, kurie nori kažkokių techninių žinių, sprendimų ar panašiai.

Architektūrai pateikti yra reikalingi jos statiniai, dinaminiai vaizdai ir išdėstymo. Statinius vaizdus sudaro sistemos struktūros paketų modelis, klasių diagramos. Dinaminius vaizdus sudaro veiklos, būsenų, sąveikos modeliai. Išdėstymui (angl. Deployment) parodyti naudojamas išdėstymo modelis. Kad geriau suprasti sistemą, pateikiamas jos panaudojimo atvejų vaizdas – panaudojimų atvejų modelis (ang. „User case“).

3.3.1 Architektūros tikslai ir apribojimai

Kadangi sistemą buvo nuspręsta kurti su JAVA programavimo kalba, kad būtų pakankamas objektiškumo lygis ir galimybė sistemą naudoti ant skirtingų operacinių sistemų, tai kone pagrindinis architektūros apribojimas būtent ir bus java JDK 1.5 platforma. Kadangi nagrinėjamos technologijos yra nemokamos ir karkasai nemokami, galima juos pilnai išnaudoti (ne kokios tai demo versijos) karkasų įvairių galimybes ir funkcijas. Taip pat bus svarstoma galimybė panaudoti jau sukurtus atvirojo kodo komponentus, jeigu jie atitiks kuriamos sistemos poreikius. Ta prasme, kad bus naudojami tik pačių karkasų komponentai ir galimybės. Toliau pateikiamas planuojamas sistemų sukūrimo ir įdiegimo grafikas.

Stadija	Paleidimo data
Sistemos sukūrimas	2006-10-20
Sistemos testavimas	2006-11-14
Diegimas ir paleidimas bandomajai eksploatacijai.	2006-11-21

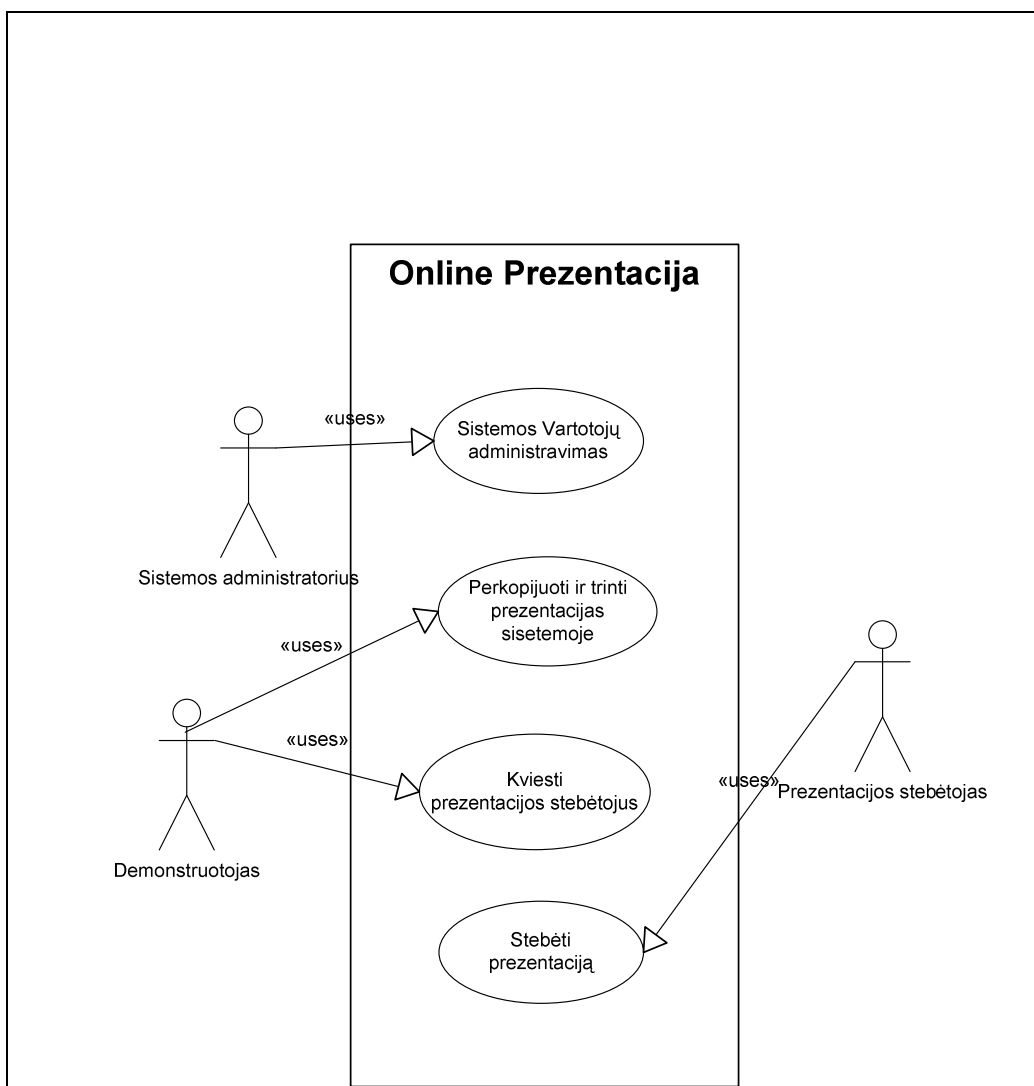
Lentelė 11 Darbų grafikas.

Architektūros pagrindiniai tikslai yra padėti geriau suprasti nagrinėjamus karkasus, susidaryti bendrą sistemos statinį ir dinaminį vaizdus, apibrėžti apribojimus ir sužinoti kokius elementus, klases ir panašiai reikės įgyvendinti.

Projektuojant sistemą buvo naudotas Omondo EclipseUML Free Editon programinės įrangos įrankis. Tai yra nemokama paketo versija.

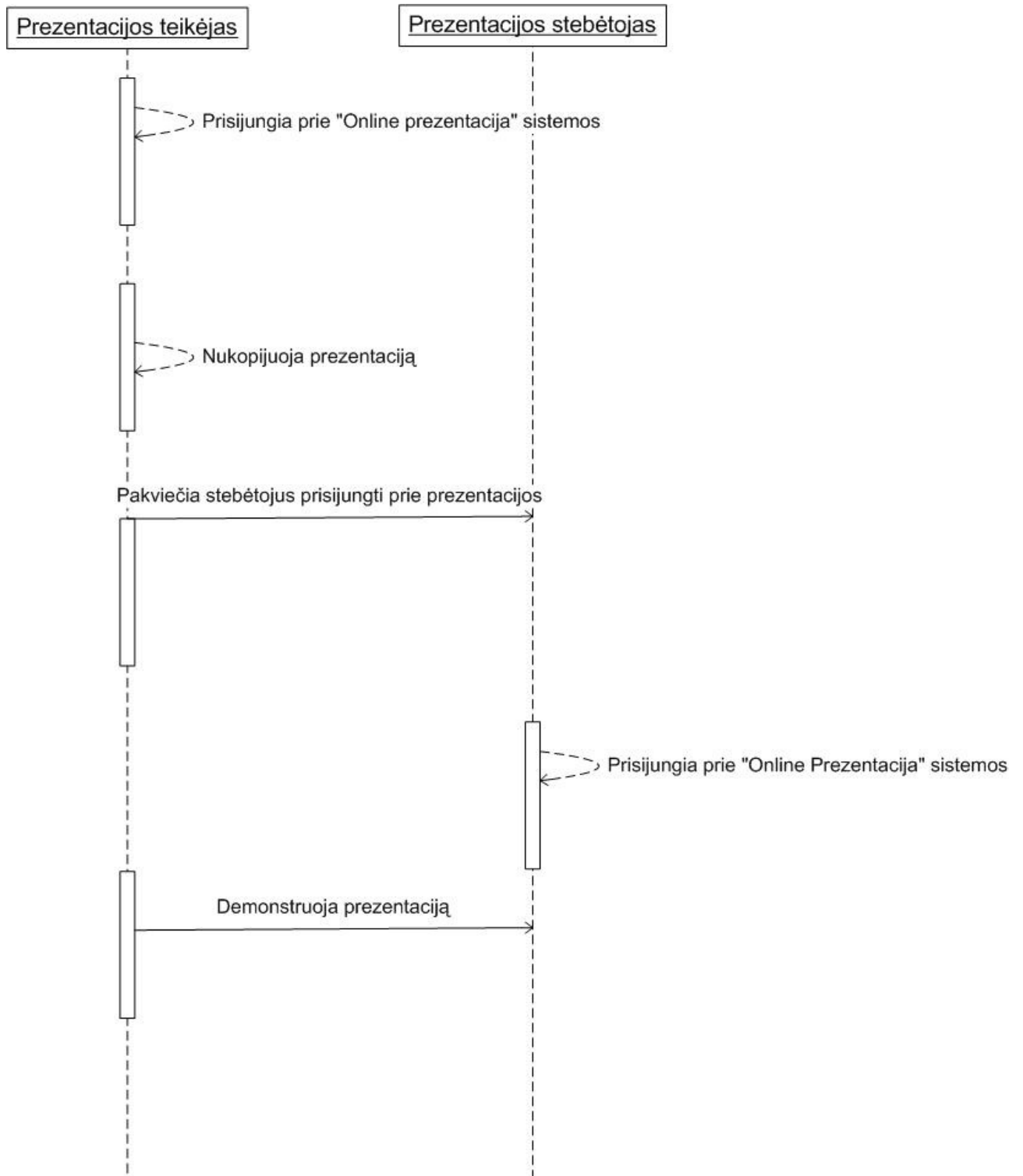
3.3.2 Panaudojimo atvejų vaizdas

Žemiau yra pateikta sistemos „Online prezentacija“ panaudojimo atvejų (angl. „User case“) diagrama.



Pav. 11 Panaudojimų atvejų modelis (use case)

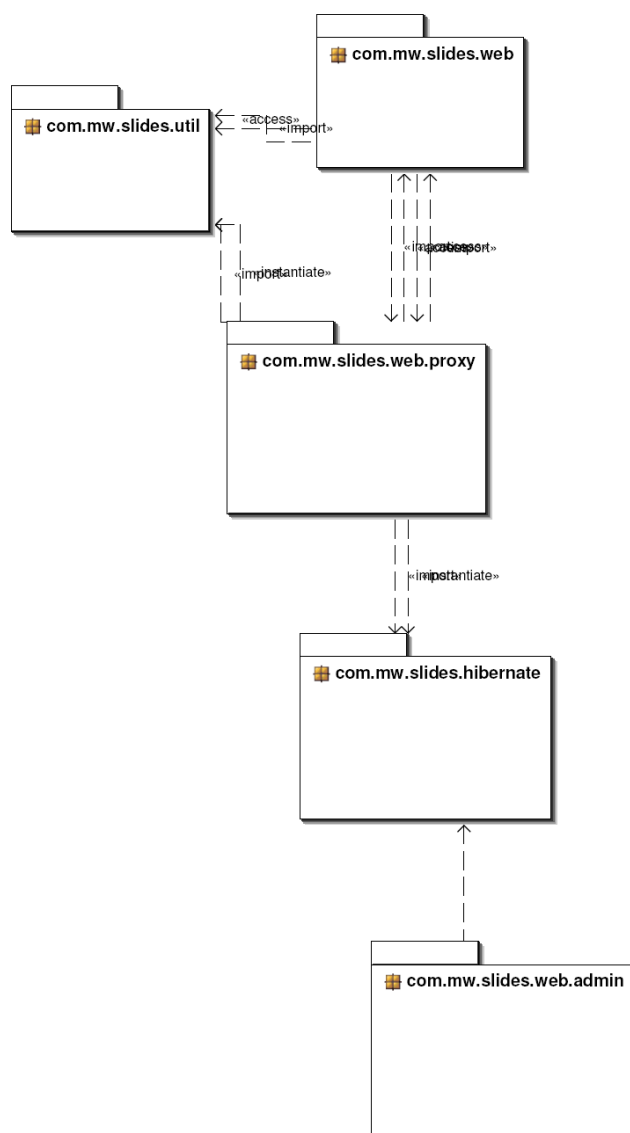
3.3.3 Sistemos sąveikos diagrama



Pav. 12 Sistemos sąveikos diagrama

3.3.4 Sistemos statinis vaizdas

Apžvalga



Pav. 13 Sistemos paketai (packages)

3.3.5 Paketų detalizavimas

Hibernate – atlieka duomenų apsikeitimo su duomenų baze funkcijas. Užklauso, duomenų rašymas ir t.t. (*Hibernate* karkasas)

Util – pagalbinės sistemos priemonės, kaip elektroninio pašto žinučių siuntimas.

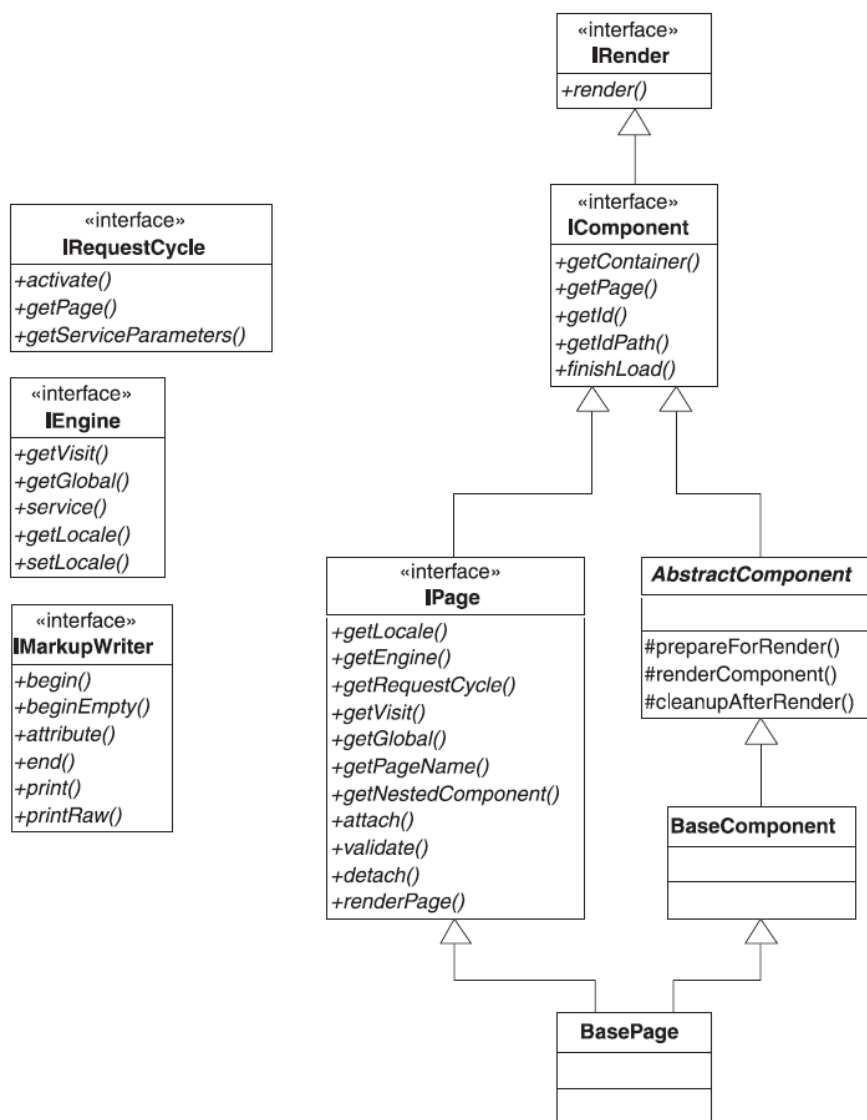
Web – serverio ir gebinio frameworko klasės, užkraunami puslapiai, vykdomas prisijungimas, komandų parsinimas.

Admin – klientų administravimas

Proxy – kliento dalis, demonstruojamų prezentacijų ir failų uploadinimas, kanalų valdymas, visi kiti veiksmai.

3.3.6 Tapestry karkasas

Bendrai Tapestry Karkaso schema (jo interfeisai, klasės ir metodai) yra pateikta žemiau:



Pav. 14 Tapestry karkaso pagrindiniai interfeisai, klasės ir metodai

Bus kuriami puslapiai, įgyvendinantys **IPage**. Atvaizdavimo pakete bus html puslapiai, su specialiom Tapestry žymėmimis. Veiklos logika bus suprogramuota puslapiuose. Duomenų valdymui bus naudojamas Hibernate karkasas.

3.3.7 Klasių diagramos

Hibernate – atlieka duomenų apsikeitimo su duomenų baze funkcijas. Užklauso, duomenų rašymas ir t.t. (Hibernate karkasas)

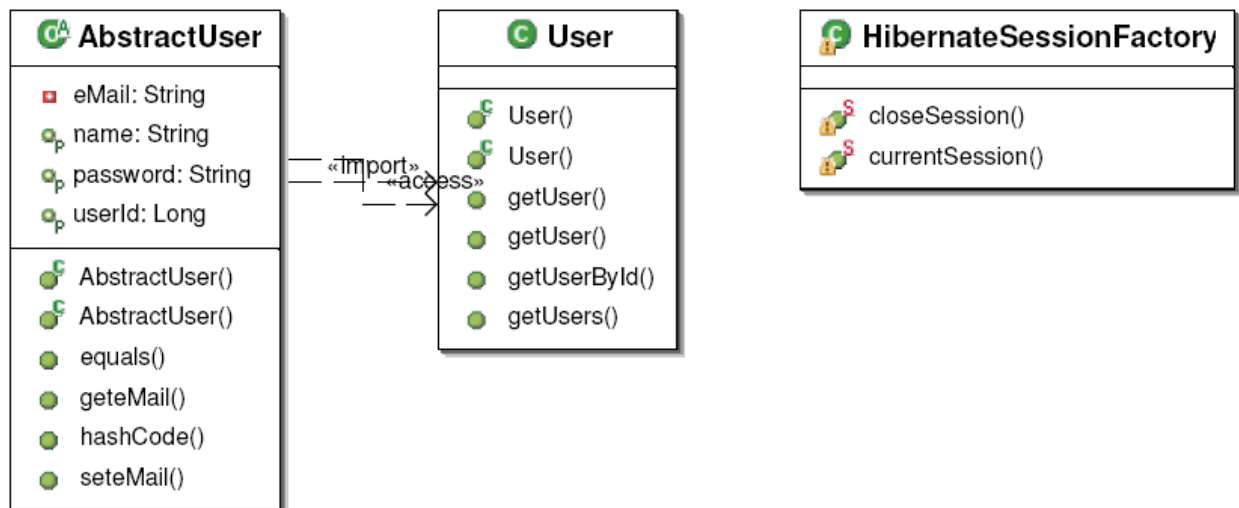
Util – pagalbines sistemos priemones, kaip elektroninio pašto žinučių siuntimas.

Web – serverio ir gebinio frameworko klasės, užkraunami puslapiai, vykdomas prisijungimas, komandų parsinimas.

Admin – klientų administravimas

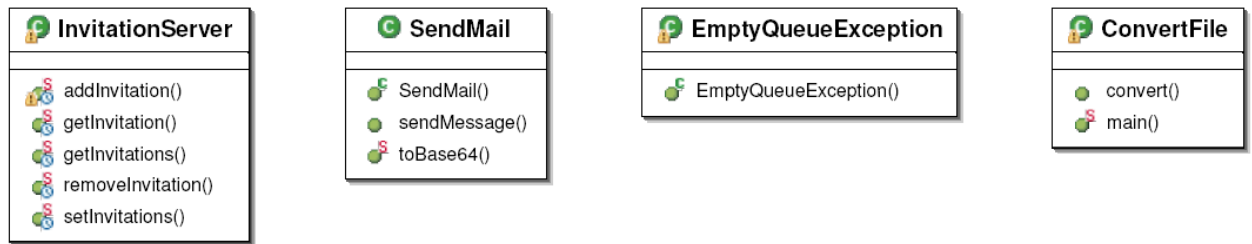
Proxy – kliento dalis, demonstruojamų prezentacijų ir failų uploadinimas, kanalų valdymas, visi kiti veiksmai.

3.3.7.1 Hibernate klasių diagrama.



Pav. 15 Hibernate klasių diagrama

3.3.7.2 Util klasių diagrama.



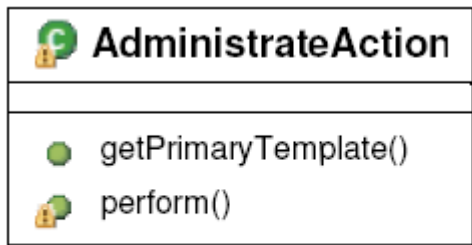
Pav. 16 Util klasių diagrama

3.3.7.3 . Web klasių diagrama.



Pav. 17 Web klasių diagrama

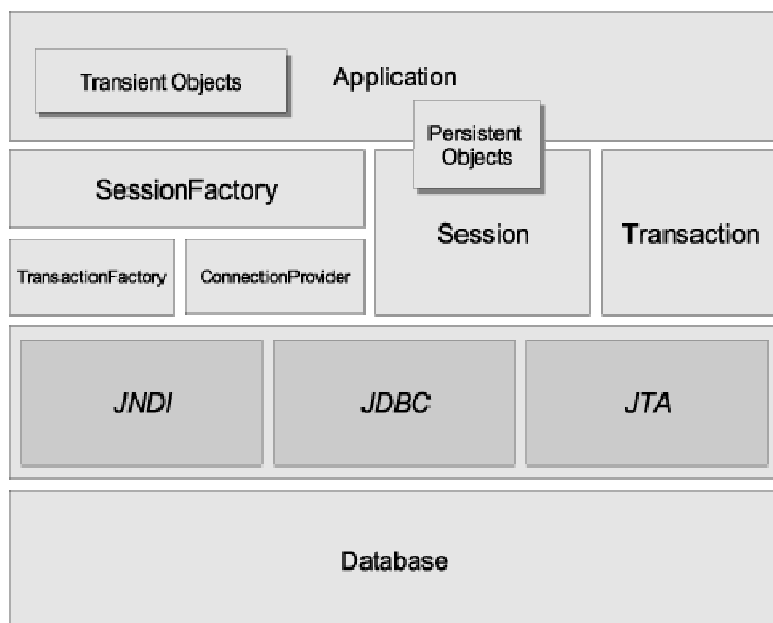
3.3.7.4 Admin klasių diagrama.



Pav. 18 Admin klasių diagrama

3.3.8 Hibernate karkasas

Žemiau yra pateikta bendra Hibernate karkaso architektūra.



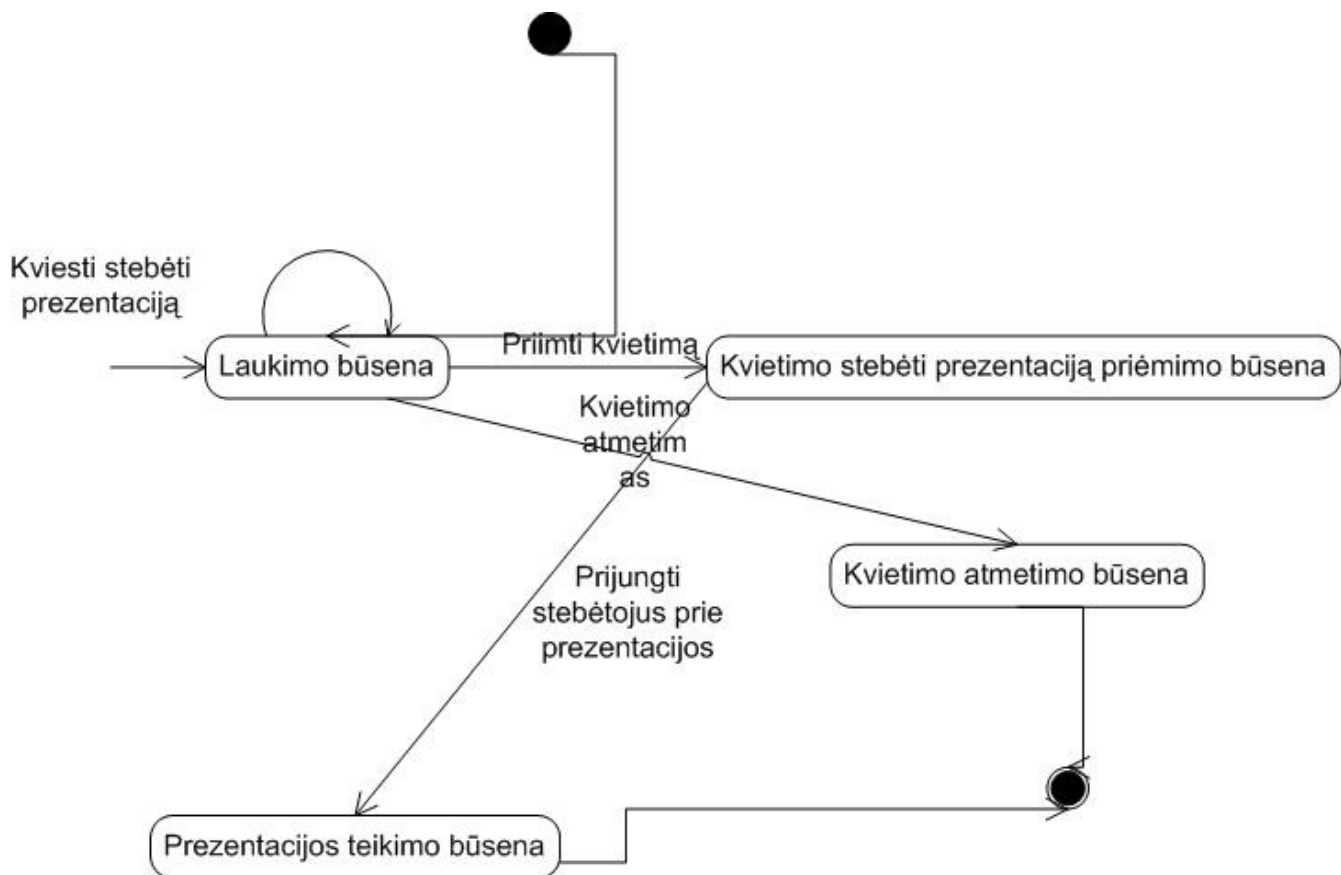
Pav. 20 Hibernate karkaso architektūra

Duomenų lygis bus pasiekiamas per „Session“ ir „Persistent Objects“ klases.

Kadangi naudosime Tapestry karkaso pagrindu kuriamoje sistemoje Ajax, pateikta viršuje diagrama parodo skirtumus tarp paprastos web aplikacijos ir Ajax web aplikacijos.

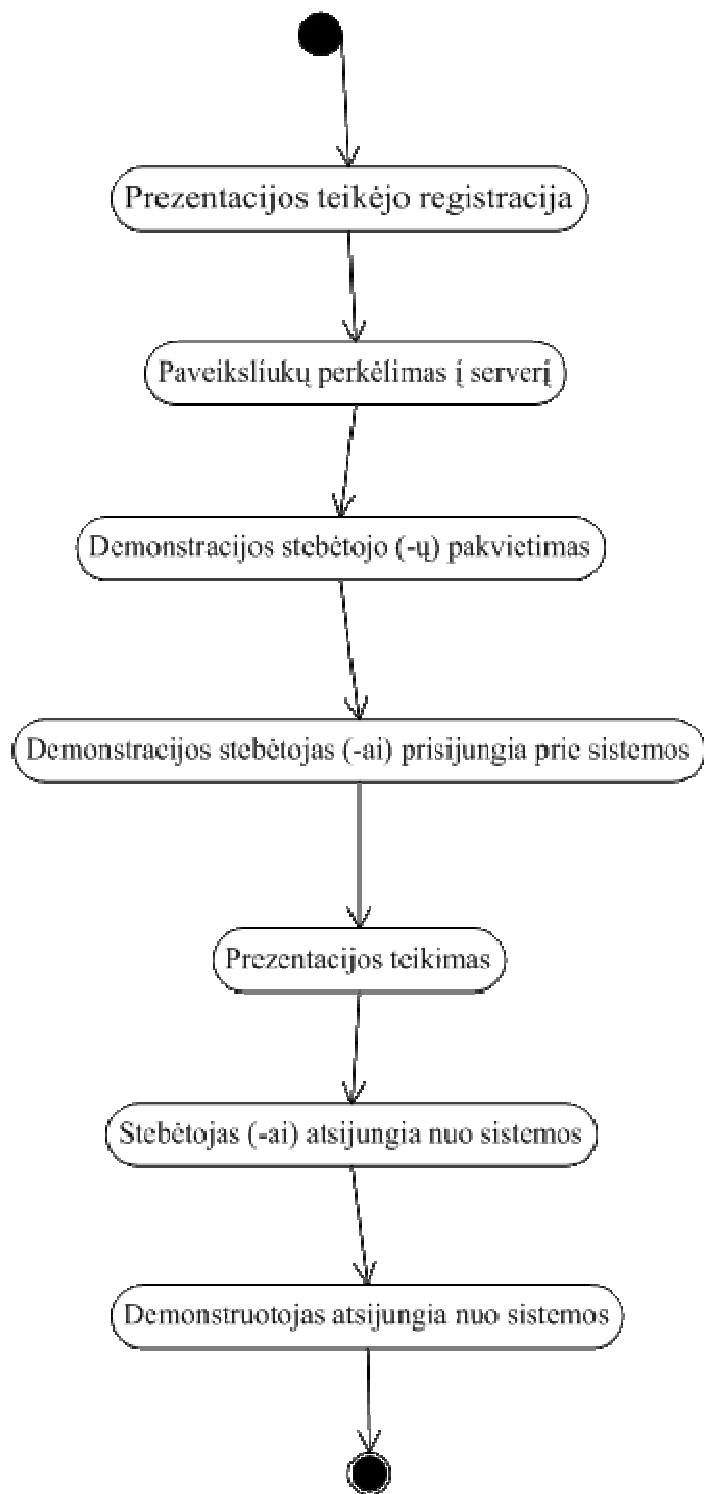
3.3.9 Sistemos dinaminis vaizdas

3.3.9.1 Sistemos prezentacijų teikimo būsenų diagrama



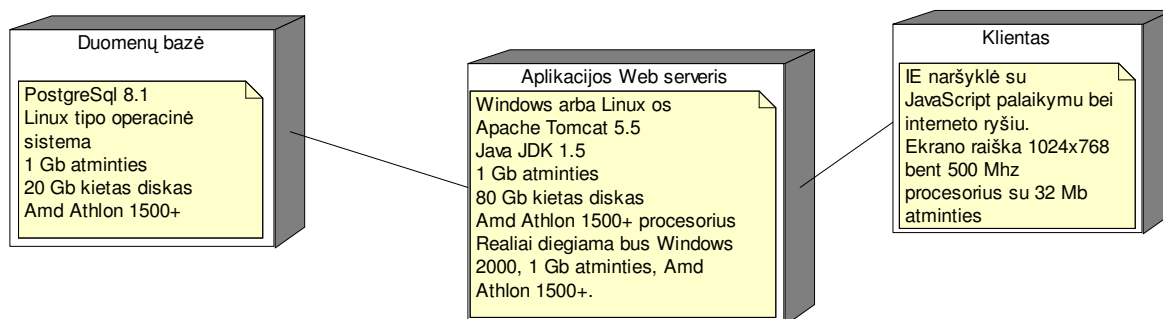
Pav. 21 Sistemos prezentacijų teikimo būsenų diagrama

3.3.9.2 Veiklos diagrama



Pav. 22 Veiklos diagrama

3.3.10 Išdėstymo vaizdas

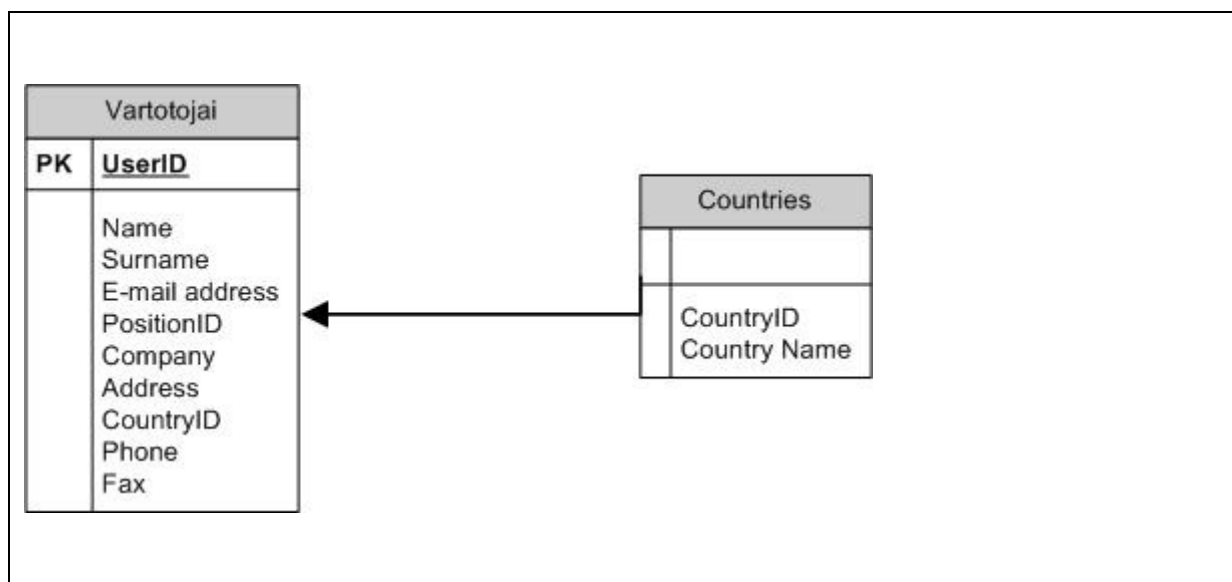


Pav. 23 Išdėstymo diagrama

3.3.11 Duomenų vaizdas

Duomenų struktūra

Sistemos „Prezentacija Online“ duomenų bazėje bus saugoma informacija apie vartotojus, turinčius ar praityje turėjusius teisę demonstruoti prezentacijas „Prezentacija Online“ sistemos pagalba. Jos duomenų struktūra yra pateikta



Pav. 24 Duomenų struktūra

3.3.12 Kokybė

Tapstry karkasas yra komponentinio interneto puslapių karkasas, turintis labai dideles išplečiamumo, daugkartinio panaudojimo galimybes. Užtenka sukurti komponentą ir jį galima naudoti keliose vietose. Kadangi duomenų apsikeitimui bus naudojamas Hibernate karkasas, kuris yra labai lankstus pačios duomenų bazės pokyčiams, tai realiai sukurta sistema bus lanksti struktūros pakeitimams. Java platformos internetinės sistemos gali būti lengvai pernešamos supakavus visą sistemą į specialius paketus. Užtenka tik paketą permesti į kitą vietą ir sistema veiks kitoje tarnybinėje stotyje.

3.4 Testavimo medžiaga

Dar kurdamas „Online prezentacija” programinę įrangą, aš testavau atskirus modulius:

- Vartotojo registracija
- Paveiksliukų perkopijavimas į serverį
- Prezentacijos stebėtojų pakvietimas
- Prezentacijos demonstravimas
- Vartotojų atsijungimas

Sukūręs pilnai integruotą sistemą, aš ją testavau kaip vieną visumą. Testavimo metu buvo naudojami skirtingi paveiksliukai prezentacijos bei simuliuojamas visas prezentacijos procesas.

Be funkcionalumo, aš dar testavau „Online Prezentacija“ sukurta programinę įrangą, kaip ji veikia su trimis labiausiai naudojamomis interneto naršyklėmis:

- Microsoft Internet Explorer (6.x, 7.x)
- Mozilla Firefox (1.x, 2.x)
- Opera (9.x)

Ištestavęs pats ir ištaisęs surastas klaidas, programą suinstaliavau į testinį serverį ir daviau kaip beta versiją testuoti UAB „MediaWorks”. Šio proceso metu vartotojai pateikinėjo pastabas ir komentarus ir aš atitinkamai taisiau bei tobulinau sukurta programą, kol neliko neišspręstų problemų ir pastabų.

Testavimui naudoju programavimo klaidų sekimo sistemą „Mantis”.

4 Vartotojo dokumentacija

Vartotojo dokumentacija yra pateikiama Priede Nr. 1.

Funkcinis sistemos aprašymas

Sistema „Online Presentacija“ suteikia galimybę demonstruoti prezentaciją, sudarytą iš atskirų paveikslėlių, nutolusiems vartotojams sinchroniškai ir realiu laiku per internetą

Sistema „Online presentacija“ suteikia demonstruotojui galimybę:

- Prisiregistruoti prie sistemos kaip demonstruotojui
- Perkopijuoti į serverį bei ištrinti paveikslukus, skirtus demonstruoti prezentaciją
- Kviesti prezentacijos stebėtojus stebėti prezentaciją realiu laiku
- Rodyti paveikslukus fiziškai nutolusiems stebėtojams realiu laiku
- Nutraukti prezentacijos rodymą stebėtojui ir jį atjungti iš sesijos
- Perkopijuoti failus į serverį ir leisti juos prezentacijų stebėtojams parsisiųsti

Prezentacijos stebėtojas gali atlikti sekančius veiksmus:

- Priimti demonstruotojo kvietimą prisijungti stebėti prezentaciją realiu laiku
- Parsisiųsti elektroniniu formatu failus, kuriuos pateikia prezentacijos stebėtojas

+

Kaip naudotis sistema „Online presentacija“

Vartotojas, demonstruojantis prezentaciją, užsiregistruoja kaip prezentacijos teikėjas ir prisijungia prie sistemos

- Vartotojas, demonstruojantis prezentaciją, spaudžia mygtuką pridėti naują paveiksluką ir prideda visus norimus demonstruoti paveikslukus
- Vartotojas, demonstruojantis prezentaciją, iškviečia norimus prezentacijos stebėtojus, išsiųsdamas elektroninio pašto pranešimą arba pasakydamas vartotojo prisijungimo vardą ir slaptažodį
- Vartotojas, stebėsiantis prezentaciją, spaudžia ant demonstruotojo atsiųstos nuorodos arba prisijungia prie sistemos su demonstruotojo jam suteiktu vartotojo vardu ir slaptažodžiu

- Demonstruotojas spaudžia ant norimo rodyti paveiksluko ir stebėtojas (-ai) mato tą paveiksluką savo monitoriaus ekrane.
- Demonstruotojas taip pat gali pažymėti failus, kaip galimus stebėtojams parsisiųsti, ir tada jie matys juos kaip nuorodas ir ant jų paspaudę, galės parsisiųsti į savo kompiuterį norimus failus.
- Norint atsijungti iš sistemos, reikia spausti mygtuką „atsijungti“.

Vartotojams sistemos diegtis ir ką nors instaliuoti nereikia, nes ji teikiama kaip paslauga.

Diegimas serveryje

Diegiant į serverį, reikia paprasčiausiai sukongigūruoti serverį ir perkopijuoti sukurtos programos failus. Reikalavimai serverio programinei įrangai:

- Apache Tomcat 5.x arba aukštesnė
- PostgreSQL RDBMS 8.x
- JRE 1.4 arba aukštesnė

5 Produkto kokybės įvertinimas

Atlikęs visus sistemos kūrimo ir testavimo darbus, aš įdiegiau programinės įrangos „Online Presentacija“ beta versiją UAB „MediaWorks“ skirtame serveryje ir buvo pradėta vykdyti bandomoji eksploatacija. Bandomosios eksploatacijos metu sistema buvo pradėta naudoti tarp skirtinguose Vilniaus ir Kauno bendrovės ofisuose dirbančių darbuotojų. Be to, palaipsniui buvo pradėta demonstruoti prezentacijas ir nesvarbiais komerciniais atvejais ne įmonės darbuotojams.

Bendri bandomosios eksploatacijos rezultatai buvo teigiami, kaip ir atsiliepimai apie patį sistemos veikimo principą. Vartotojai labai greitai įsisavino šios programinės įrangos veikimą ir neturėjo didesnių problemų ją naudodami. Kadangi šios bandomosios eksploatacijos pabaigoje jau nekilo jokių problemų su pačios sukurtos programinės įrangos veikimu, ir nebuvo jokių nusiskundimų, galima daryti prielaidą, jog sistema yra pakankamos kokybės ir tinkama naudojimui. Be to, be grįžtamojo ryšio palaikymo ryšio su sistemos vartotojais, buvo nuolat stebimas ir sistemos pats darbas bei atlikti „stress“ testai, kurie leidžia daryti prielaidą, kad sukurtos sistemos „Online Presentacija“ kokybė yra patenkinama.

Be to, pažymėtina ir tai, kad pilnai buvo įgyvendinti ir visi funkciniai ir nefunkciniai reikalavimai kelti sistemai „Presentacija Online“ projekto pradžioje.

6 Išvados

Šio magistro darbo pradžioje iškeltus uždavinius aš sėkmingai įgyvendinau ir atlikau tiek bendrą Ajax technologijų analizę, bet ir sėkmingai sukūriau “Online prezentacija” programinę įrangą, kuri turėtų pasitarnauti, kaip prototipas kuriant jos realizacijos pagrindu komercinę programinę įrangą, skirtą vykdyti prezentacijas nutolusiems vartotojams internetu.

Įvertindamas šio darbo metu įgytą patirtį ir žinias, aš galiu daryti sekančias pagrindines išvadas:

1. Ajax technologijomis veikiančią programinę įrangą sukurti nėra labai sudėtinga ir tai turėtų nesunkiai įsisavinti „vidutinės“ kvalifikacijos programinės įrangos kūrėjas.
2. Ajax technologijų pritaikymas su skirtingomis pagrindinėmis interneto naršyklėmis nėra lengvas, tačiau neabejotinai pasiekiamas, ką pavyko pasiekti ir man, kuriant programinę įrangą „Online prezentacija“.
3. Interneto tinklo patikimumas yra vienas svarbiausių reikalavimų, užtikrinančių nepertraukiamą ir stabilų darbą su Ajax technologijų pagrindu sukurta programine įranga.
4. Ajax technologijos neabejotinai suteikia programinės įrangos kūrėjams pranašumų prieš standartines „desktop“ tipo programinės įrangos kūrimo priemones srityse, kur yra pageidautinas vartotojų tarpusavio bendravimas ir informacijos apsikeitimas realiu laiku.
5. Ajax technologijų taikymas praktikoje dar tikrai nepasiekė nei savo brandos, nei galimybių ribų ir jau artimiausioje ateityje sąlygos naujos kartos ir aukštesnio interaktyvumo lygio programinės įrangos, veikiančios interneto naršyklės pagrindu, sukūrimą, ką sėkmingai pavyko pademonstruoti ir man pačiam.

Literatūra

1. Jesse James Garrett “Ajax: A New Approach to Web Applications” (<http://www.adaptivepath.com/publications/essays/archives/000385.php>)
2. Dave Crane, Eric Pascarello, Daren James „Ajax in Action“, Manning Publications Co. 2006
3. Leland Scott „Ajax/DHTML Library Scorecard: How Cross Platform Are They“, <http://www.musingsfrommars.org/2006/03/ajax-dhtml-library-scorecard.html>, 2006.03.04
4. Jakob Nielsen, <http://www.useit.com/papers/responsetime.html>
5. Straipsnis internete “Gates admits IE failings, looks to an AJAX future”, <http://news.zdnet.co.uk/internet/0,1000000097,39258532,00.htm>
6. Nicholas C. Zakas, Jeremy McPeak and Joe Fawcett, Professional Ajax, Wrox Press 2006
7. Joshua Eichorn, Understanding AJAX: Using JavaScript to Create Rich Internet Applications, Prentice Hall, 2006
8. Michael Mahemoff, Ajax Design Patterns, O’Reilly, 2006
9. Tinklapis: <http://www.ajaxian.com/>
10. S. Holzner, Ajax for Dummies, 2006
11. Wikipedia. Web Application
12. Tinklapis: http://en.wikipedia.org/wiki/Web_application
13. Baekdal. T. AJAX, XMLHttpRequest or AHAX 2005 (<http://www.baekdal.com/articles/Technology/ajax-xmlhttprequest-ahax/>)
14. W3C. W3C Document Object Model (<http://www.w3.org/DOM/>)
15. W3C. The XMLHttpRequest Object (<http://www.w3.org/TR/XMLHttpRequest/>)
16. McLaughlin. B. Advanced requests and responses in Ajax, (<http://www.ibm.com/developerworks/web/library/waajaxintro3/index.html>)
17. Apple Developer Connection Dynamic HTML and XML: The XMLHttpRequest Object (<http://developer.apple.com/internet/webcontent/xmlhttpreq.html>)
18. Tinklapis: <http://www.ajaxinfo.com>

Terminų ir santraukų žodynas

Terminas	Paaiškinimas
AJAX	Asynchronous JavaScript+CSS+DOM+XMLHttpRequest technologijų rinkinys.
GUI	(graphical user interface) Grafinis vartotojo interfeisas.
JSP	(java server page) normalus HTML puslapis su Java kodu. JSP kompiliuotojas JSP paverčia į Servletą.
Servlet	Serverio pusės Java programa, kuri suteikia papildomų savybių serveriui. Iš Servleto yra generuojamas HTML kodas.
HTML	(HyperText Markup Language). Web puslapio aprašymo kalba.
HTTP	(HyperText Transfer Protocol). Standartinis web puslapių perdavimo protokolas.
URL	(Uniform Resource Locator). Internetinis web resursų adresas, suprantamas ir interpretuojamas web naršyklių.
CSS	(Cascading Style Sheets). HTML dokumentų formavimo, stilizavimo kalba.
XML	(eXtensible Markup Language). Universali kalba.
API	(Aplication program interface) Aibė šablonų bibliotekose, kurios išplečia kalbos funkcionalumą.
EJB	(Java Enterprize beans)
JDBC	(Java database connectivity). Sun Microsystems standartas apibrėžiantis aplikacijų priėjimą prie duomenų bazių.
XLS	(Xml stylesheets), xml aprašomas puslaido stilius formavimnas ir panašiai.
Terminas	Aprašymas
DAO	(data access object) dizaino šablonas, atskiriantis žemo lygio duomenų prieigą, nuo aukšto lygio biznio logikos.
DOM	(Document Object Model). Medžio tipo struktūra, laikanti visus HTML objektus. Naudojama Javascript ir kitų skriptavimo kalbų, kad dinamiškai formuoti ir keisti HTML puslapius.
JDK	(Java develompent kit) – priemonės skirtos kūrimui java kalboje

Priedas Nr. 1. Ajax vartotojo sąsajos komponentai

Standartiniai vartotojo sąsajos komponentai, kaip HTML teksto įvedimo laukai, radiobutonai seniai nesikeitė, ir todėl daugelis programinės įrangos kūrėjų pradėjo ieškoti ir kurti naujus vartotojo grafinės sąsajos komponentus. Žemiau yra nagrinėjami bei demonstruojamas veikimas labiausiai išpopuliarėjusių ir pradėtų plačiausiai naudoti programinės įrangos kūrime Ajax vartotojo sąsajos komponentų.

1. Reikšmių pasirinkimo juosta (angl. Slider)

Pasirinkimo juosta įgalina vartotoją pasirinkti tam tikrą vieną reikšmę iš tam tikros aibės reikšmių. Pvz., tai galėtų būti panaudojama įvertinant paskaitos aktualumą ar supratimą.

Jos vartojimo atveju, vartotojas traukia žymeklį nustatytoje reikšmių aibėje ir taip pasirenka norimą reikšmę. Kaip atskiras variantas, gali būti du žymekliai, kurie įgalintų pasirinkti norimų reikšmių poaibį nustatytos reikšmių aibės, o ne vieną kurią nors reikšmę. Standartiniame HTML yra du būdai tai realizuoti: radiobutonai (angl. radiobuttons) ir selektoriai (angl. selectors). Abu jie leidžia pasirinkti tam tikrą konkrečią reikšmę iš pateiktų reikšmių aibės. Didžiausias pasirinkimo juostos privalumas yra tai, kad pasirenkamų reikšmių aibė yra nepertraukiama ir diskreti. Pažymėtina, kad terminas „nepertraukiama“ galbūt ir nėra visai tikslus, nes pasirinkimų negali būti daugiau nei yra pikselių pasirinkimo juostoje, tačiau bet kurio atveju paprastai yra keli šimtai galimų pasirinkti reikšmių ir galima teigti, kad šiuo atveju terminas nepertraukiamas yra gana tinkamas ir adekvatus realiame gyvenime. Pasirinkimo juostoje taip pat galima atvaizduoti ir diskrečius duomenis, pažyminti keletą konkrečių reikšmių. Tokiu atveju, kai vartotojas nustoja traukti žymeklį, jis iš karto „peršoka“ į artimiausios reikšmės vietą.

Reikšmių pasirinkimo juostos pagrindiniai pranašumai prieš standartiniame html realizuotų radiomygtukų ir selektorių yra sekantys:

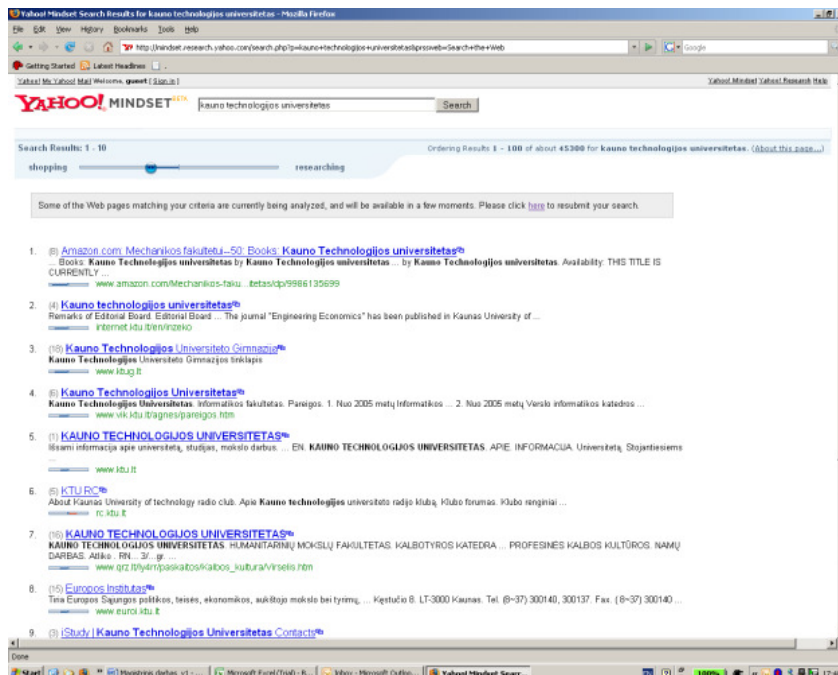
- Reikšmių pasirinkimo juosta pateikia geresnį ir aiškesnį vizualų reikšmių, esančių pasirinkimo spektre, atvaizdavimą
- Vartotojui yra paprastai greičiau pasirinkti norimą reikšmę pasirinkimo juostoje, perkeliant žymeklį į bet kurią norimą vietą nei tiksliai pažymėti konkretų radiomygtuką ar naujame pranešimų lange

- Užpildant įvairias anketas ar pateikiant kelis vertinimus vienu metu, yra žymiai patogiau atsakančiajam į klausimus palyginti naudojamą vertinimo skalę ir palyginti atsakymus į skirtingus klausimus pagal vienodą vertinimo skalę.

Kuriant reikšmių pasirinkimo juostą, reikia nuspręsti sekančius pagrindinius dalykus:

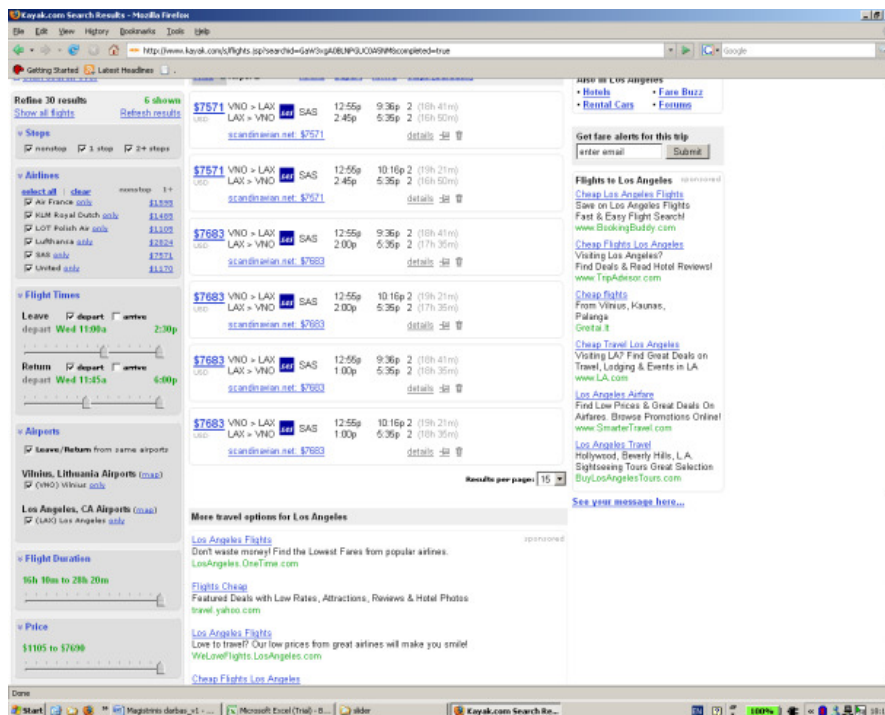
- Kaip juostą pateikti puslapyje: horizontaliai ar vertikalčiai
- Kokią skalę naudoti:
 - Kokybiniai vertinimai („Geras“, „Blogas)
 - Absoliutinės vertės
 - Procentai (paprastai nuo 0 iki 100)
- Kaip atvaizduoti skalę:
 - Nepateikti reikšmių pavadinimų
 - Pateikti reikšmių pavadinimą skalės pabaigoje
 - Pateikti reikšmių pavadinimus keliose skalės vietose

Vienas iš realių sistemų pavyzdžių, kaip reikšmių skalė yra panaudojama realiose interneto aplikacijose yra Yahoo!Mindset (<http://mindset.research.yahoo.com/>), kur galima surūšiuoti rezultatus atitinkamai pagal pirkimas (angl shopping) ir tyrimas (angl. researching). Yahoo!Mindset bendras vaizdas pateiktas žemiau:



Pav. 25 Yahoo! Mindset

Kitas vertas dėmesio pavyzdys yra kelionių paieškos tinklapyje www.kayak.com pateikiama galimybė pasirinkimo juostos pagalba, pagal pasirinktas laiko ribas, apriboti pateikiamų kelionių sąrašą. Be to, atitinkamai yra pateikiama galimybė riboti pateikiamas keliones pagal kainą ir skrydžio trukmę. Šio lango vaizdas pateiktas žemiau:



Pav. 26 Kayak.com

2. Progreso indikatorius (angl. Progress indicator)

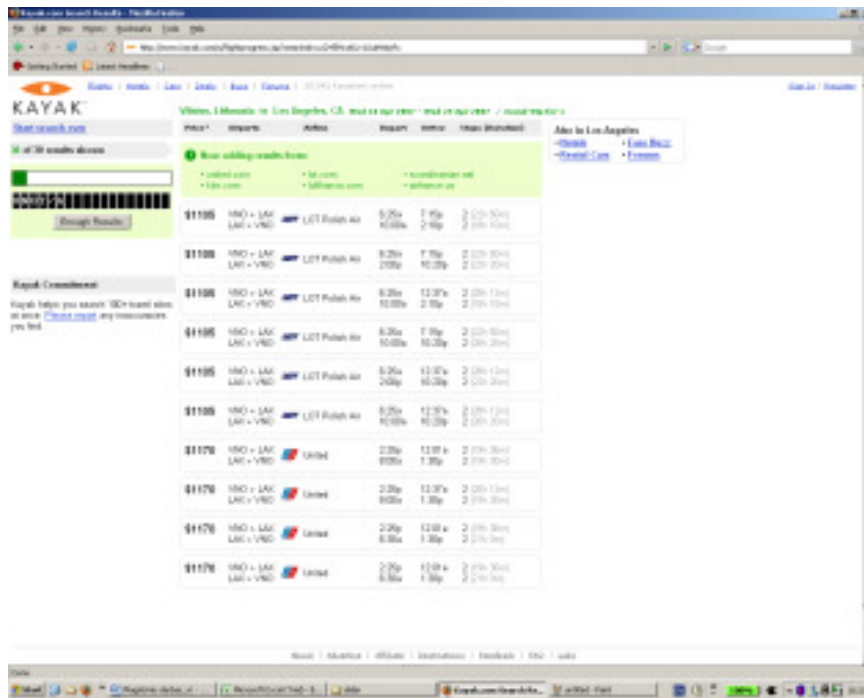
Progreso indikatorius yra skirtas pateikti vartotojui grįžtamąjį ryšį, kol yra „laukiama“ atsakymo iš serverio pusės arba informacijos „apsikeitimas“ vyksta vienu metu su kitais vartotojo vykdomais veiksmiais. Progreso indikatorius tuo pačiu išlaiko vartotojo dėmesį, pagerina vartotojo sąsają bei tam tikra prasme informuoja klientą, kad sistema yra „gyva“ ir kad atsakymas iš sistemos pusės dar negautas. Pagrindiniai progreso indikatoriaus variantai gali būti išskirti sekantys:

- Trumpesniems vėluojamos informacijos pateikimams:
 - Nedidelė žinutė, kad „informacija yra atnaujinama“
 - Animuotas paveikslėlis
- Ilgesniems vėluojamos informacijos pateikimams:
 - Skaitliukas, rodantis koks progresas buvo pasiektas
 - Rodomas vertinamas laikas, likęs informacijai pateikti
 - Pranešimų seka, rodanti koks dabar veiksmas yra vykdomas
 - Turinys, galintis sudomintis vartotoją (kaip pvz. Naudingas dienos patarimas) arba grafinis sistemos veikimo atvaizdavimas

Be abejo, galima naudoti ir kelis variantus vienu metu. Kodėl progreso indikatorius yra svarbus, atspindi ir žinomo programinės įrangos ir interneto aplikacijų navigacijos specialisto Jakobo Nielseno (Jakob Nielsen) pastebėjimai interneto aplikacijoms:

- 0.1 sekundės: vartotojai jaučiasi tiesiogiai ir betarpiškai dirbantys su sistemos objektais.
- 1 sekundė: Riba, kai vartotojai dar „jaučia“, kad laisvai dirba su sistema ir nereikia jiems begalo ilgai laukti, kol sulauks sistemos atsakymo.
- 10 sekundžių: Vartotojų praranda koncentraciją dirbdami su sistema, todėl, anot Jakobo Nielseno, visiems veiksmais, reikalaujantiems ilgesnio nei 10 s grįžtamojo ryšio iš serverio, yra reikalingas progreso indikatorius.

Vienas iš progreso indikatoriaus pavyzdžių gali būti jau aukščiau mano minėto kelionių paieškos tinklapyje <http://www.kayak.com> naudojamas progreso statuso indikatorius, parodytas žemiau pateikiamame paveikslėlyje:



Pav. 27 Kayak.com

3. Navigavimas gilyn (ang. Drilldown)

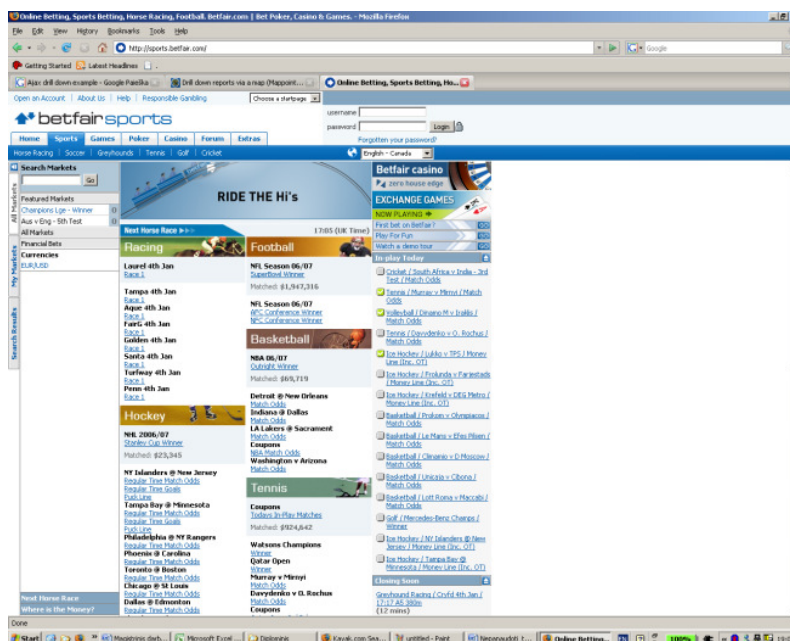
Pagrindinis šios funkcijos tikslas yra suteikti vartotojui patogią galimybę „judėti“ tarp skirtingų hierarchinės struktūros elementų grupių ir pabaigoje pasirinkti norimą elementą. Analitinėje Ajax literatūroje pavyko aptikti sekančias navigavimo gilyn alternatyvas:

- Išlaikyti navigavimą gilyn tam tikroje apibrėžtoje tinklapio srityje. Kiekvieną kartą paspaudus ant vienos hierarchinės struktūros elementų grupės, vartotojui rodoma tikta ta viena elementų grupė. Šio vaizdavimo būdo pagrindinis privalumas yra tai, kad kiekvienu atveju, pasirinkus tam tikrą naują hierarchinės struktūros elementų grupę, puslapis yra nekeičiamas, ir atvaizduojamos hierarchinės struktūros dalis nesikeičia.
- Parodyti visus hierarchinius lygius iš karto. Vartotojui pasirinkus tam tikrą kategoriją, hierarchinės struktūros dalis yra praplečiama ir yra išskleidžiama vartotojo pasirinkta hierarchinės struktūros elementų grupė.
- Kaip tam tikras antrojo varianto atvejis, yra kai visi lygiai yra išskleidžiami ir rodomi, tačiau vartotojas gali pasirinkti tik tai vieną variantą iš pasirinktojo meniu. Geriausia tokio meniu iliustracija galėtų būti Microsoft Windows Start meniu.

Realizuojant navigavimo gilyn vartotojo sąsajos komponentą, reikėtų apsispręsti dėl dviejų pagrindinių dalykų:

- Kokio gilumo turėtų būti hierarchija? Be abejo, kuo daugiau yra duomenų, tuo mažiau yra tikslinga persiųsti iš anksto duomenis vartotojui, jeigu jam tuo metu jų nereikia.
- Ar navigavimo gilyn pati hierarchija keičiasi? Jeigu hierarchija keičiasi, tai reikia paimti naujausius duomenis iš serverio, kai kiekviena kategorija yra atidaroma arba suaktyvinama. Pačiu kraštutiniu atveju, serveris netgi pats gali sugeneruoti duomenų hierarchiją pagal pareikalavimą.

Navigavimo gilyn Ajax pavyzdžiu galėtų būti tinklapio Betfair (<http://www.betfair.com>) kairiame šone esantis norimo įvykio, už kurį norima statyti, pasirinkimas.



Pav. 28 Betfair.com

4. Duomenų pateikimas elektroninių lentelių pavidalu (angl. Data Grid)

Šiuo vartotojo sąsajos komponentu yra siekiama suteikti galimybę vartotojui patogiai analizuoti ir lyginti pateikiamus duomenis. Pagrindinės šio vartotojo sąsajos komponento poreikio priežastys yra sekančios:

- Ajax programinė įranga neretai turi savo nuosavas duomenų bazes bei sąveikauja su kitomis išorinėmis duomenų bazėmis
- Vartotojams dažnai reikia analizuoti pateikiamus duomenis, patikrinti jų teisingumą, bandyti nustatyti prognozuojamas ateities tendencijas bei atlikti kitus veiksmus su šiais duomenimis

- Be abejo, yra labai sunku iš anksto prognozuoti, kokius veiksmus vartotojas norės atlikti su pateikiamais duomenimis

Ši problema Ajax programavimo priemonėmis sprendžiama sukuriant duomenų lenteles ir suteikiant vartotojui sekančias galimybes:

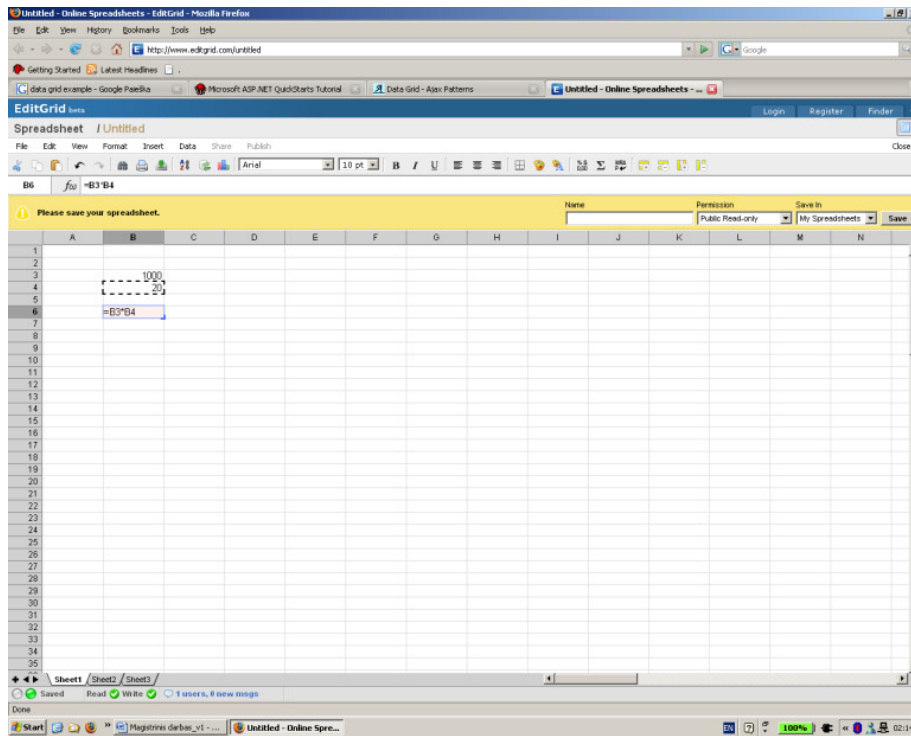
- Rūšiuoti duomenis
- Filtruoti duomenis
- Grupuoti duomenis
- Vykdyti agreguotus skaičiavimus
- Redaguoti pateikiamus duomenis

Panaudojant Web Remoting, pateikiami duomenys vartotojui atrodo kaip kokia virtuali darbo aplinka, suteikianti iliuziją, kad jis dirba su didžiuliais duomenų masyvais. Iš tiesų gi, duomenys yra laikomi serveryje, ir kiekvienas vartotojo veiksmas sąlygoja serverio užklausimą ir tam tikrų rezultatų persiuntimą vartotojui atgal. Užklausos paprastai vykdomos XML pranešimų, JSON pranešimų, arba HTML pranešimų pagalba.

Apsisprendžiant, kaip kurti duomenų pateikimą lentelių pavidalu, yra svarbu apsispręsti dėl sekančių dalykų:

- Reikia numatyti būdus, kaip bus tikrinamas vartotojo įvedamų duomenų teisingumas.
- Reikia atskiras celes paversti vartotojo sąsajos duomenų įvedimo ir redagavimo komponentais.

Viena iš įdomesnių duomenų lentelių realizacija yra Elitgrid duomenų įvedimo ir analizės lentelės (<http://www.editgrid.com/untitled>). Kaip atrodo elitgrid.com pateikta žemiau esančiame paveikslėlyje:



Pav. 29 Elitgrid

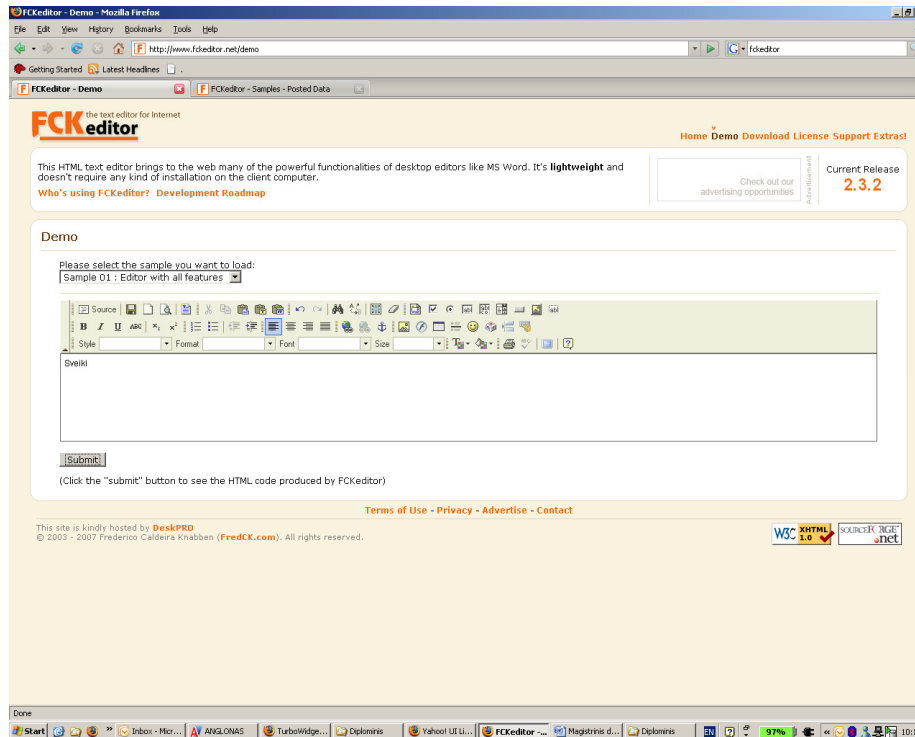
5. Turtingos sąajos teksto editorius (angl. Rich text editor)

Daugelyje interneto ir tuo pačiu Ajax interneto programinėje įrangoje reikia vartotojams suvesti daug teksto. Be to neretai vartotojai turi suvesti gražiai suformatuotą tekstą html, tačiau jų HTML žinios yra labai menkos. Todėl yra gana patogu įdiegti WYSIWYG (angl. What you see is what you get) vartotojo sąajos elementus arba teksto redaktorių. Paprastai tokie tekstiniai redaktoriai suteikia sekančias pagrindines galimybes:

- Kintančius teksto stilius, dydžius, paryškinimas, ir pan.
- Lanksčias spalvų schemas
- Įterptus paveikslėlius
- Įterptas lenteles
- Įvairius sąrašus
- Nustatyti tarpus tarp teksto

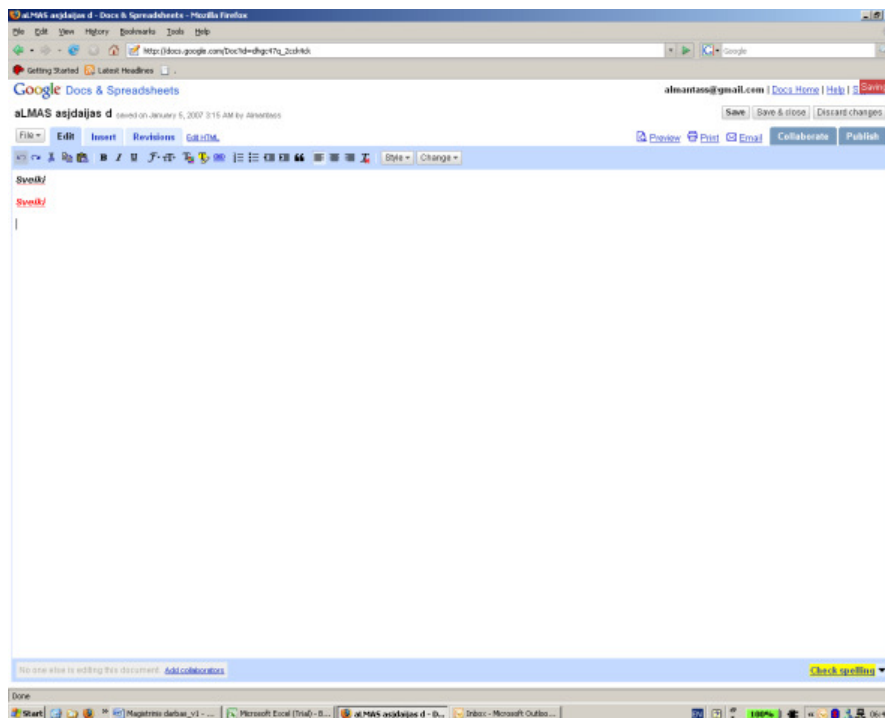
Taip pat paprastai gali panaikinti ar pakartoti paskutinį vykdytą veiksmą, nukopijuoti teksto dalį iš vienos vietos į kitą, išsaugoti, rasti rašybos klaidas ir atlikti kitus veiksmus.

Iš realių teksto redaktoriaus realizavimo, paminėtinas FCKeditor (<http://www.fckeditor.net/>), kuris suteikia galimybę rašyti html WYSIWYG. Šio redaktoriaus bendras vaizdas pateiktas žemiau esančiame paveikslėlyje:



Pav. 30 FCKeditor

Kitas paminėtinas projektas – Google docs, įgalinantis laisvai rašyti tekstus ir juos išsaugoti kaip atskirus failus, pasirenkant įvairius formatus, tame tarpe ir Microsoft Word, OpenOffice write, Adobe PDF ir kitais formatais.



Pav. 31 Google docs

6. Automatinis pasiūlymas pabaigti pildyti renkama tekstą (angl. Suggestion)

Šio Ajax vartotojo sąsajos komponento pagrindinis tikslas yra palengvinti vartotojui suvesti tekstinius duomenis, pasiūlant keletą variantų kaip užbaigti pradėtą spausdinti žodį ar sakinį.

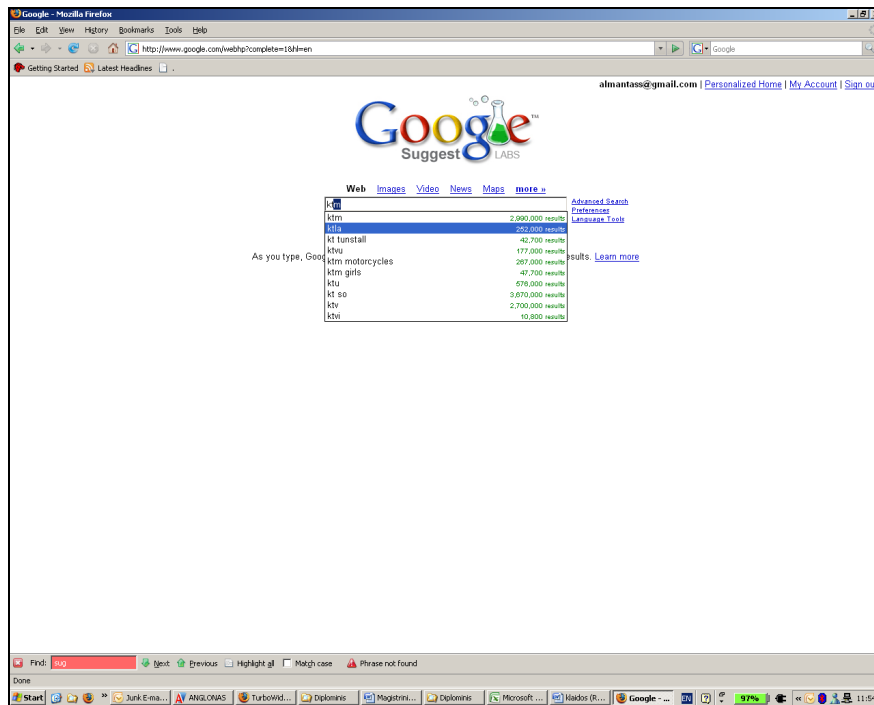
Pagrindiniai veiksniai sąlygojantys šio vartotojo sąsajos komponento poreikį yra sekantys:

- Vis daugiau ir daugiau interneto naršyklės pagrindu veikiančios programinės įrangos vartotojai suvedinėja informacijos teksto pavidalu
- Kai vartotojams pateikiamas tuščias laukas įvesti tekstui, vartotojai ne visada lengvai supranta, kokią ten informaciją reikia arba galima įvesti
- Nors daugelis programinės įrangos vartotojų šiandien yra įgūdę rinkti tekstus su klaviatūra greitai, nemaža dalis yra ir tokių, kurie yra labiau įgūdę dirbti su programa pelės pagalba
- Teksto rinkimo greitis vis dar yra problema, nes daugelis mąsto greičiau nei gali rinkti tekstą
- Vartotojai rinkdami tekstą daro klaidas

Kuriant automatinį pasiūlymą pabaigti pildyti renkama tekstą, reikia apsispręsti dėl sekančių pagrindinių dalykų:

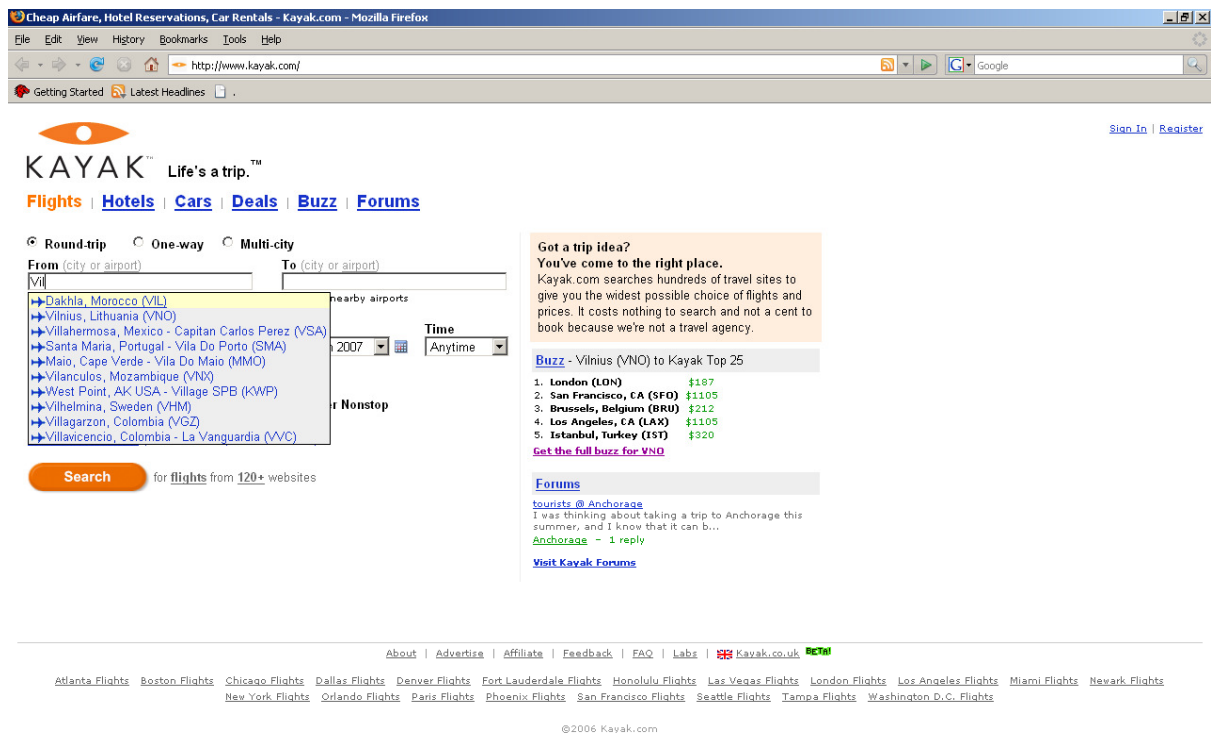
- Kokį siūlymo pabaigti pildyti renkama tekstą parinkimo algoritmą naudoti:
 - Naudoti istorinius duomenis
 - Personalizuoti. Užuoat pateikus tuos pačius variantus visiems vartotojams, parinkti konkrečius siūlymus kiekvienam vartotojo pagal to vartotojo istorinę patirtį
 - Naujesniems rezultatams suteikti didesnę svorį arba apskritai siūlyti pasirinkti tikrai tam tikro netolimo laikotarpio buvusius pasirinkimus
 - Naudoti interneto vartotojo interneto naršyklės „cache“ atmintyje išsaugotus tekstus ir pagal juos pagal pasirinktus algoritmus pateikti siūlymus užbaigti renkama tekstą
- Kokia tvarka siūlymai užbaigti renkama tekstą bus pateikti vartotojui. Galima būtų išskirti sekančius rūšiavimo būdus:
 - Istoriskai buvęs pasirinkimo dažnumas
 - Pagal tai, kaip neseniai pasirinkimo atvejai buvo
 - Pagal abėcėlę ir skaičių eiliškumą
 - Pagal tam tikrą specifinę programinės įrangos kūrime pasirinktą eiliškumą
- Kiek pasiūlymai užbaigti renkama tekstą vartotojui rodyti
- Kokia papildoma informacija vartotojui bus pateikiama su kiekvienu „pasiūlymu“

Klasikinis šio vartotojo grafinės sąsajos komponento pavyzdys yra Google Suggest, kur bene pirmą kartą viešai prieiname projekte buvo panaudotas šis Ajax vartotojo sąsajos komponentas. Google Suggest bendras vartotojo sąsajos vaizdas pateiktas žemiau:



Pav. 32 Google Suggest

Kitas pavyzdys galėtų būti kelionių paieškos Kayak tinklapis, kur sistema pateikia pasiūlymus, kaip užbaigti pildyti išvykimo ar atvykimo aerouostą:



Pav. 33 Kayak.com

7. Aktyvi paieška (angl. Live Search)

Pagrindiniai veiksniai, paskatinę tokio vartotojo sąsajos komponento atsiradimą:

- Ajax programinėje įrangoje vartotojams dažnai reikia atlikti paiešką
- Kai kada kone tiesiogine priežodžio prasme vartotojas „ieško adatos šieno vežime“
- Suteikti vartotojui galimybę vykdyti paiešką jo norimu būdu bei eksperimentuoti vykdamt paiešką

Standartiniu atveju vartotojo sąsajos komponentas “Greita paieška” realizuojamas sekančiai:

1. Vartotojas pakeičia komponento statusą, pvz. Įveda naują tekstinę ženklą ar pažymi pasirinkimo opciją
2. Visa paieškos informacija XMLHttpRequest pagalba nusiunčiama į serverį

3. Serverio programinė įranga nustato, kokia informacija turi būti (jeigu turi) parodyta vartotojui ir ji nusiunčiama vartotojui

4. Interneto naršyklė atnaujina vartotojui pateikiamus rezultatus

Pagrindiniai sprendimai, realizuojant komponentą “greita paieška” yra sekantys:

- Kaip interpretuoti nepilnai surinktą tekstą
- Kiek rezultatų vartotojui pateikti
- Ką daryti, kai per didelis skaičius galimų rezultatų

Geru aktyvios paieškos naudojimo pavyzdžiu gali būti Delta Vacations tinklapyje esanti viešbučių paieška. Pavyzdžiui, jeigu aš ieškau tam tikroje vietoje (mano pademonstruotu atveju Las Vegase), ir įvedu raide “c”, tai iš karto bus atrinkti tiksliai tie viešbučiai, kurie prasideda raide “c”.

The screenshot shows the Delta Vacations website interface. At the top, there is a navigation bar with links for Home, Destinations, Activities, Deals, Members, and Customer Care. Below this, there are links for Membership log in, My Profile, and Flight Hotel Car. The main content area is titled "Las Vegas, Nevada" and features a large image of the city skyline at night. Below the image, there is a text block describing the city and its attractions. A "Hotel Search" section is visible, with a search bar containing the letter "c" and a dropdown menu set to "featured pick". Below the search bar, two hotel listings are shown: "Caesars Palace" and "Circus Circus". Each listing includes a small image, a star rating, a brief description, and a "View More" link. A "PRICE NOW" button is visible at the bottom right of the listings.

Pav. 34 Delta vacations

8. Realus laiko komandinė eilutė (angl. Live Command-Line)

Pastaruoju metu nemaža dalis programinės įrangos remiasi vartotojo grafine sąsaja, paremta pelės navigacija ir paspaudimais. Tačiau ne visada tai yra pati geriausias ir patogiausias būdas dirbti su programine įranga:

- Komandinė eilutė paprastai suteikia daugiau galimybių nei tikrai kompiuterio pelės naudojimas navigacijai
- Paieškos laukai pastaruoju metu neretai tampa bendros paskirties komandinėmis eilutėmis

Tradicinė komandinė eilutė pasižymi tokiu trūkumu, kad paprastai vartotojo veiksmai turi būti paremti prieš tai vartotojo įgyta patirtimi ir įsisavintomis tam tikromis žiniomis. Iš esmės, kaip tai veikia, tai “vartotojas įveda komandą, o kompiuteris įvykdo tą komandą”. Tačiau tai nėra labai efektyvu.

Komandinės eilutės realizavimo sprendimas galėtų būti daug interaktyvesnis. Pavyzdžiui galėtų būti sekantys šio vartotojo sąsajos komponento patobulinimai:

- Patikrinti įvedimo metu, ar įvedama komanda gali būti duotuoju momentu įgyvendinta
- Pasiūlyti vartotojui pataisyti neteisingai įvestą komandą
- Pasiūlyti būdus ir variantus, kaip užbaigti vesti konkrečią komandą
- Padėti vartotojui įsivaizduoti, kas įvyks, kai komanda bus įvykdyta
- Pateikti grafinius vartotojo sąsajos komponentus, įgalinančius pateikti komandą
- Pateikti vizualizacijos priemones, įgalinančias vizualiai atvaizduoti komandą

Realizuojant šį Ajax vartotojo grafines sąsajos komponentą, svarbu atsakyti į sekančius klausimus:

- Kur dalinė analizė įvedamos komandos bus atlikta: interneto naršyklėje ar serveryje?
- Kiek plačiai informacijos pateiks serveris?

Šios vartotojo sąsajos pilnai ajax realizacijų kol kas nepavyko aptikti, tačiau literatūroje yra neretai pateikiamas YubNub (<http://yubnub.org>) pavyzdys, kaip tam tikra komandinės eilutės vartotojo sąsajos realizacija. Ši programinė įranga leidžia įvesti komandas, kurios po to yra persiunčiamos vykdymui atitinkamiems servisams. Pvz. Įvedus raidę ” w” (kas yra oras – angl. weather) ir miesto pavadinimą, bus surastas ir pateiktas to miesto pavadinimas. Bendras YubNub vartotojo sąsajos vaizdas pateiktas žemiau:



w los angeles

Some examples of what you can do with YubNub:

[gim porsche 911](#) Do a Google Image search for Porsche 911's. [create](#) Create a new command **Cool!**
[random 100](#) Pick a random number between 1 and 100. [ls](#) List all available commands.
[man random](#) Display help for the "random" command. [xe -amount 100 -from USD -to EUR](#)

Popular Searches

Google	g	Google News	gnews	Yahoo!	y
Wikipedia	wp	Technorati	tec	Amazon	am
CNN	cnn	Weather for zip code	weather	eBay	ebay
AllMusic.com	allmusic	delicio.us tag	deli	Flickr	flk
ESPN	espn	Yahoo! Stock Quote	stock	Dictionary (Answers.com)	a

[More commands...](#)

[Yet more commands...](#)

[What Is This Thing?](#) | [List All Commands \(ls\)](#) **XML** | [Create a New Command](#) | [Installing YubNub](#) | [Advanced Syntax](#)
[Most-Used Commands](#) | [Golden Eggs \(Notable Commands\)](#) **XML** | [Jeremy's Picks](#)
[Community \(Google Group\)](#) | [Wiki](#) | [YubNub Blog](#) | [Press \(Technorati\)](#) | [Acknowledgements](#) | [Typos? Spam? Contact Jon](#)

Pav. 35 yubnub.org

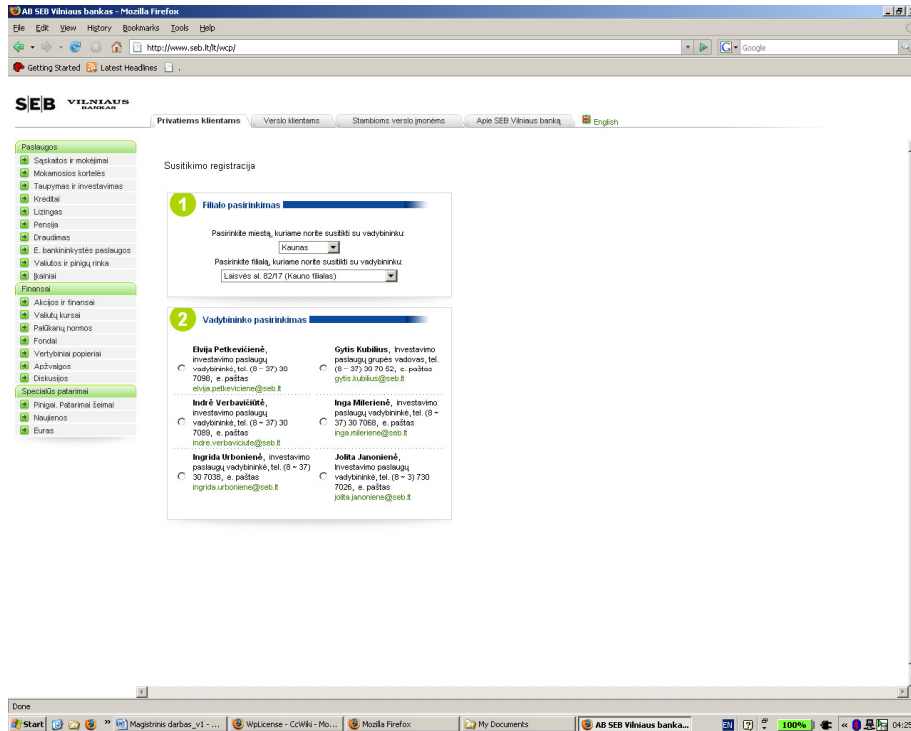
9. Dinaminė forma (angl. Live Form)

Pildant tokią formą atitinkami klausimai yra pateikiami vartotojui, priklausomai nuo to, kokius duomenis jis pateikė prieš tai. Šio vartotojo sąsajos komponento vystymąsi skatinai sekantys veiksniai:

- Informacijos pateikimas reikalauja tam tikro lankstumo ir priklausomai, koks atsakymas, tai gali sąlygoti vienokį ar kitokį klausimą
- Didžioji dalis pateikiamos informacijos turi būti tikrinama ir validuojama
- Vartotojams yra nepatogu, kai įvesta informacija yra patikrinama tik tai jau siunčiant galutinę formą.

Realus tokios vartotojo sąsajos pavyzdys galėtų būti AB SEB Vilniaus banko realizuota forma užsiregistruoti konsultacijai investicijų klausimais. Ši forma yra realizuota taip, kad

pasirinkus vieną opciją, sekantis pasirinkimo variantas pateikiamas jau priklausomai nuo to, kas buvo pasirinkta arba atsakyta prieš tai. Taip pavyzdžiui, pasirinkus miestą, pateikiami šiame mieste esantys SEB Vilniaus banko filialai, o pasirinkus konkretų filialą, toliau pateikiami šio filialo už konsultacijas atsakingų darbuotojų vardai ir pavardės bei kontaktinė informacija. Šios formos grafinis vaizdas pateiktas žemiau:



Pav. 36 SEB Vilniaus bankas

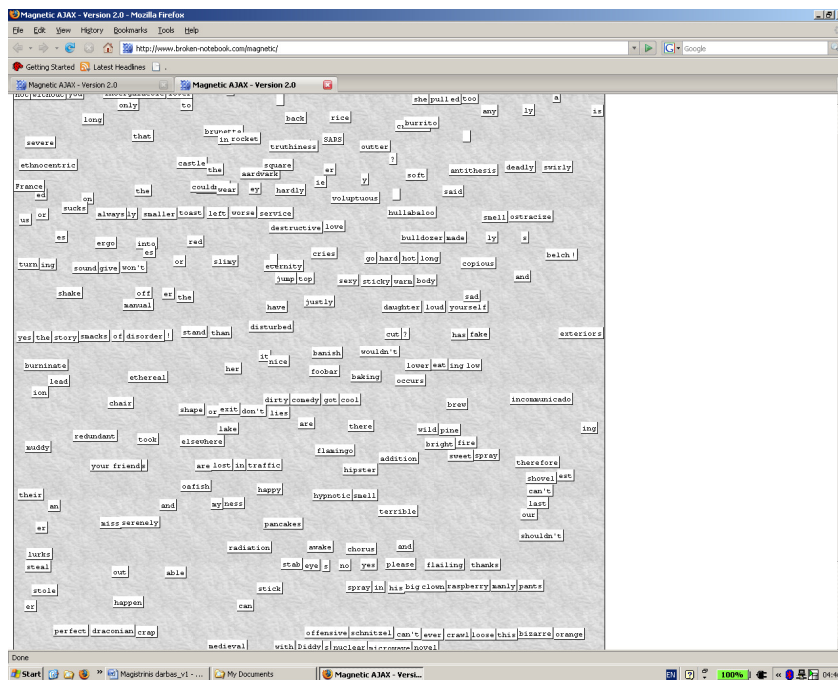
Priedas Nr. 2. Ajax puslapių architektūra

Šiame priede aš panagrinėsiu esamus ir labiausiai paplitusius Ajax puslapių tarpusavio susiejimo būdus.

1. Vizualaus perkėlimas į kitą ekrano vietą (angl. Drag and drop)

Drag-and-drop yra plačiau naudojamas ir pasiteisino desktopinėse aplikacijose. Drag-and-drop veikimas yra gana paprastas: vartotojas užveda pelės žymeklį ant tam tikro puslapio elemento, paspaudžia pelės kairįjį mygtuką ir, laikydamas nuspaudęs pelės kairįjį mygtuką, nutempia pasirinktą elementą pelės pagalba į norimą puslapio vietą, kur vartotoja atleidžia paspaustą pelės mygtuką. Realizuojant drag-and-drop funkciją atskirais atvejais gali būti prasminga apriboti elemento „judėjimo“ ribas, be to šis elementas gali būti „pilnai“ perkeltas į naują vietą arba tiktai jo kopija perkelta.

Vienas iš įdomiausių drag-and-drop funkcijos pritaikymo pavyzdžių yra aplikacija, simuliuojanti nedidelių magnetinių užrašų, paprastai klijuojamų ant šaldytuvų, veikimą interneto tinklapyje (<http://www.broken-notebook.com/magnetic>). Šios aplikacijos bendra vartotojo sąsaja pateikta žemiau:



Pav. 37 Broken-notebook

2. Turinio blokai (angl. Sprite)

Sprite gali būti apibrėžti, kaip maži, lankstūs, pranešimo tipo turinio blokai. Paprastai sprite yra paveikslukai, bet gali būti ir „div“ tipo elementai, turintys savyje tekstinę informaciją arba paveikslukus, arba ir tekstinę informaciją ir paveikslukus.

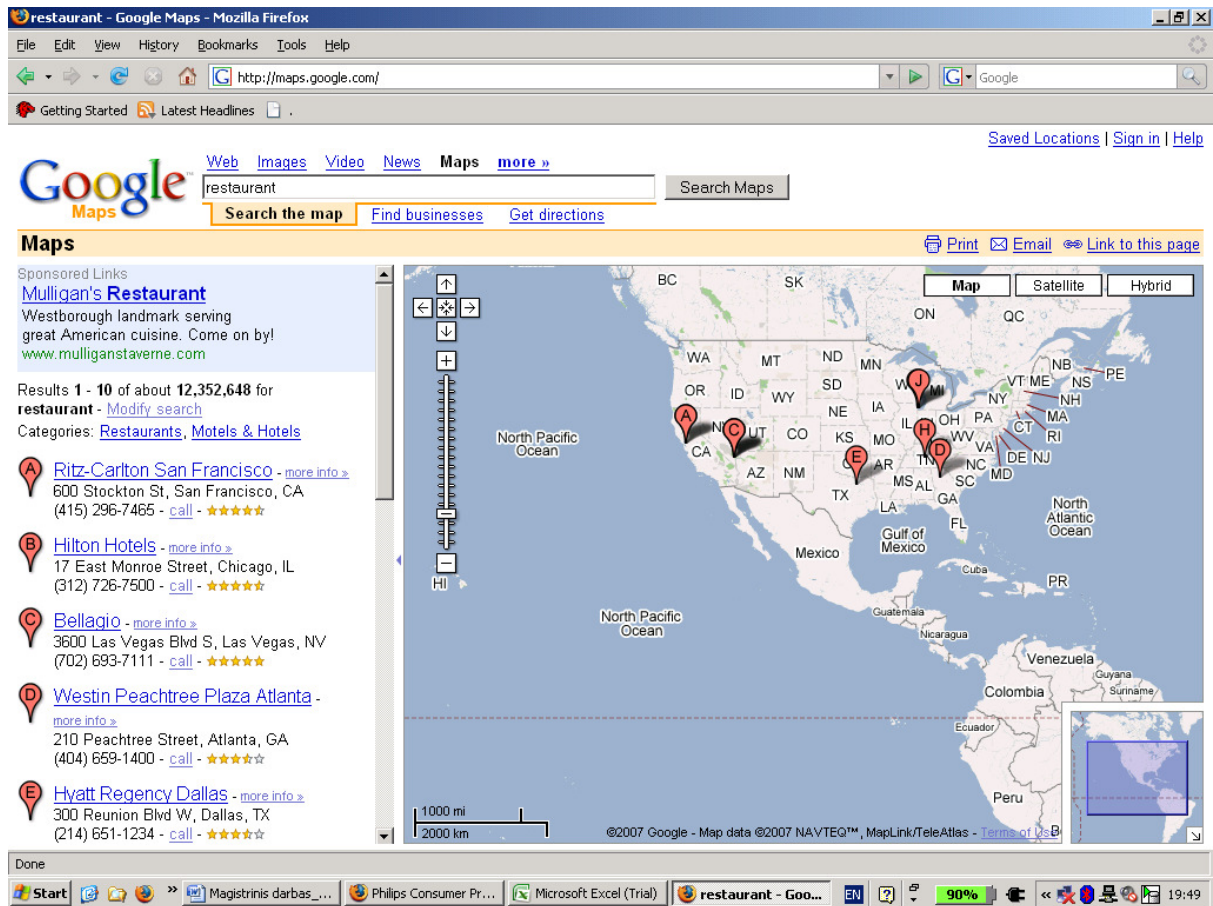
Svarbiausi sprendimai, kuriuos reikia priimti, norint įgyvendinti sprite sprendimus yra sekantys

- Kadangi sprite paprastai yra paveikslukai, kokį paveikslukų formatą (JPEG, GIF, PNG) naudoti. Pagrindiniai argumentai už vieną ar kitą paveiksluko formatą gali būti sąlygoti sekančių veiksnių:
 - Pernešamumas (angl. Portability)
 - Paveiksluko kokybė
 - Failo dydis
 - Permatomumas / Transparantiškumas (angl. Transparency)

Paanalizavus, įvairius pateikiamus argumentus už ir prieš, galima daryti išvadą, kad JPEG formatas nelabai tinka Sprites, nes nepalaiko permatomumo (transparantiškumo). Dėl šios JPEG formato savybės jis yra tinkamas tikrai santykinai dideliems, stačiakampio, nuotraukų tipo Sprite, o tokie Sprite praktikoje yra gana retai naudojami. PNG paveikslukų formatas turi tam tikrų techninių pranašumų prieš GIF kaip kintamas transparantiškumas, tačiau PNG formato nepalaiko Microsoft Internet naršyklė IE6 ir ankstesnės versijos. Visa tai apibendrinus, galima daryti išvadą, kad tinkamiausias paveikslukų formatas naudoti Sprite yra GIF, o JPEG ar PNG yra prasmė naudoti tikrai atskirais specifiniais atvejais.

- Nuspręsti, ar sprite bus animuotas. Nors animacija gali būti erzinanti ir dėmesį atitraukianti, ji, savo ruožtu, gali būti labai patogi tam tikrai informacijai ir žiniai atvaizduoti.

Vienu iš sprite naudojimo klasikinių pavyzdžių gali būti <http://maps.google.com>. Šio puslapio pavyzdys ir naudojami sprite lašeliai pateikiamas žemiau:



Pav. 38 Google maps

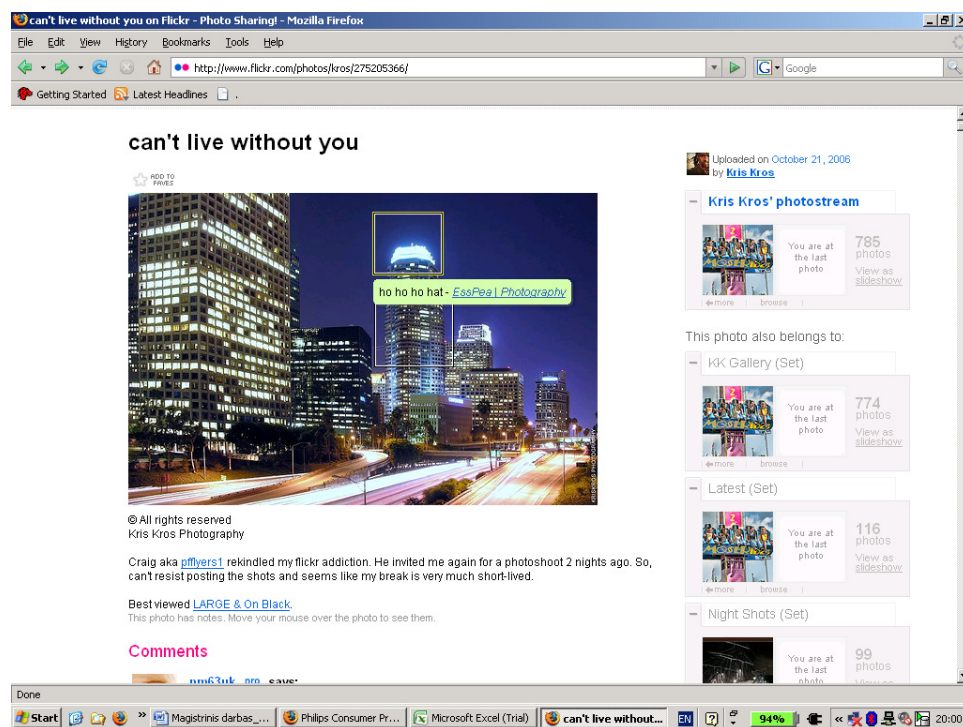
3. Iššokantis langas (angl. Popup)

Popup paprastai naudojamas, kai reikia atlikti papildomus veiksmus arba gauti tam tikros papildomos informacijos apie tam tikrus interneto tinklapio objektus. Be to, popup paprastai naudojama ir tada, kai ekrano turinys yra ribotas, o papildoma informacija arba galimi iššaukti veiksmus, tačiau jie nėra tokie svarbūs, kad būtų verta juos patalpinti pačiame tinklapyje.

Realizuojant atsidarančio lango vartotojo sąsają, svarbu priimti sekančius sprendimus:

- Kaip atrodys atsidarantis langas?
- Kaip vartotojos atsidarys ir uždarys atsidariusį langą?

Viena iš atsidarančio lango įdomesnių realizacijų yra Flickr (nuotraukų talpinimo) sistema, kur kiekvienas vartotojas gali prie matomos nuotraukos rašyti pastabas. Užvedus kompiuterio pelės žymeklį ant nuotraukos, atsiranda pastabos. Taip pat ir pats vartotojas gali rašyti čia pastabas.



Pav. 39 Flickr

4. Redaguojamo turinio blokai (angl. Malleable Content)

Interneto naršyklės pagrindu veikiančioje programinėje įrangoje dažnai reikia ne tiktai perskaityti ir gauti pateikiamą informaciją, bet ir pačiam vartotojui redaguoti pateikiamą informaciją. Be to, programinė įranga turėtų vartotojams leisti lengvai atpažinti blokus, kur galima redaguoti informaciją. Siekiant palengvinti ir padaryti patį turinio redagavimą lengvesniu ir patogesniu, ir naudojami taip vadinami redaguojamo turinio blokai tinklapyje.

Ajax technologijos įgalina sukurti tokią vartotojo grafinę sąsają, kuri leistų vartotojui lengvai atsidaryti ir redaguoti atskirus turinio blokus. Paprastai, redaguojamo turinio blokas yra div arba textarea elementas. Jį galima padaryti redaguojamu sekančiais būdais:

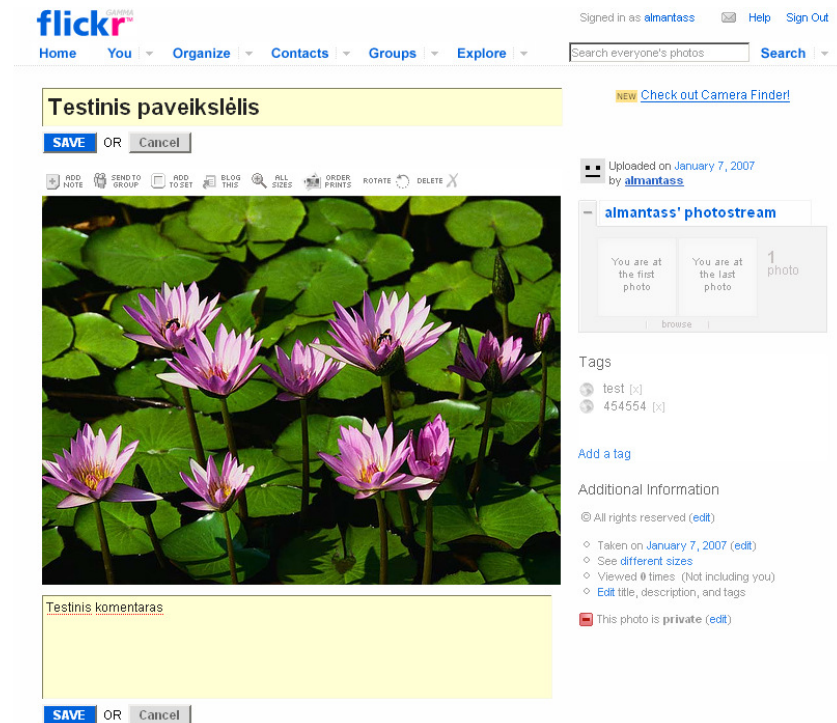
- Originalu turinio elementas yra papildomas turinio redagavimo elementais
- Vietoje esamo turinio sukuriama nauja div arba textarea tipo elementas
- Originalus elementas yra paslepiamas, o nerodomas div arba textarea elementas pateikiamas

Redaguojamų blokų pradėtas taikymas yra tam tikra prasme pasitraukimas nuo daugiau nei 10 metų internete naudotų įvedimo formų. Tačiau šis būdas padaro tam tikro turinio įvedimą ir redagavimą, o tuo pačių ir pačių formų įvedimą gerokai patogesniu ir lengvesniu naudoti vartotojams.

Kuriant redaguojamus turinio blokus, reikia apsispręsti dėl sekančių pagrindinių dalykų:

- Kaip vartotojai atpažins ir atsidarys redaguojamo turinio blokus
- Ar pakeistame turinio elemente bus išsaugota ir pateikiama informacija, kas pakeitė turinį
- Kokį turinį bus galima redaguoti ir kokios apimties jis bus
- Kaip vartotojas pažymės turinio redagavimo pabaigą
- Ar leisti vienam vartotojui redaguoti keleto skirtingų blokų informaciją vien metu.

Vienas iš paprasčiausių redaguojamo turinio blok dalies realizacijos pavyzdžių yra Flickr (<http://www.flickr.com>) pridėtos savo nuotraukos pavadinimo ir komentarų rašymas ir redagavimas. Bendras šios aplinkos vaizdas yra pateiktas žemiau esančiame paveikslėlyje.



Pav. 40 Flickr

5. Mikronuoroda (angl. Microlink)

Mikronuorodos programinėje įrangoje, sukurtoje Ajax technologinėmis priemonėmis, suteikia galimybę vartotojui naviguoti po programinę įrangą. Pagrindinės priežastys, skatinančios mikronuorodų naudojimą, yra sekančios:

- Ajax programinė įranga dažnai kaip atsaką į tam tikrą vartotojo veiksmą pateikia vartotojui naują turinį
- Siekiant paskatinti bendravimą su sistema, turinys turi būti vartotojui parodytas greitai ir be jokių trukdžių
- Puslapio perkrovimas iš naujo lėtina vartotojo darbą su interneto naršyklės pagrindu sukurta programine įranga

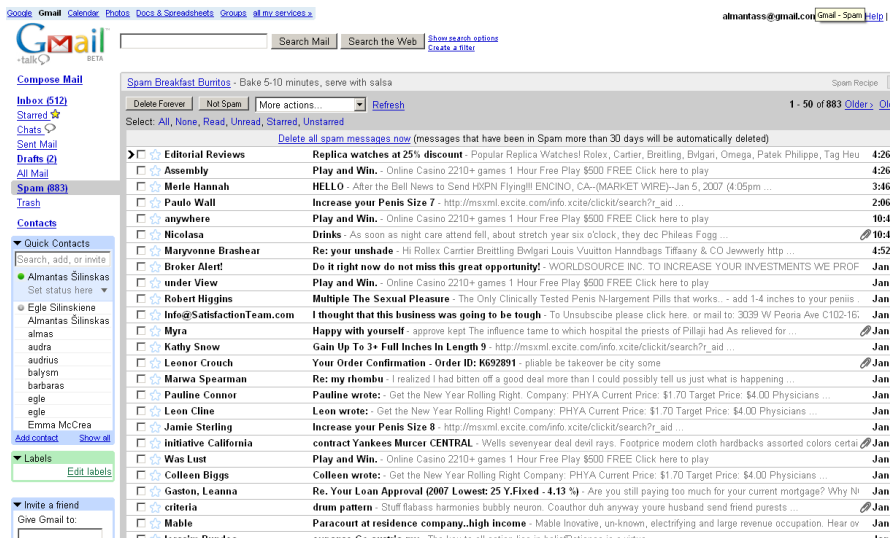
To pasėkoje išpopuliarėjo taip vadinamo mikronuorodos, kai atnaujinamas ir vartotojui persiunčiama nauja informacija ne viso tinklapio, o tiktai vartotojo veiksmų iššaukti tinklapio turinio pokyčiai. Mikronuorodos gali naudojami sekančiais būdais:

- Jos gali iššaukti naujo lango atsidarymą
- Jos gali leisti programinės įrangos vartotojui naviguoti programinės įrangos gilyn
- Mikronuodos gali suteikti galimybę jų pagalba naviguoti po programinę įrangą

Mikronuorodų naudojimas Ajax programinėje įrangoje suteikia sekančius pranašumas:

- Naujas turinys vartotojui pateikiamas greičiau
- Programinės įrangos veikimas atrodo nuoseklesniu
- Turinio pasikeitimai atrodo ryškesni

Kone žinomiausias mikronuorodų pavyzdys yra Gmail (<http://www.gmail.com>) programinė įranga. Google bendrovė čia naudoja mikronuorodas į gautų elektroninių pašto žinučių dėžutę, kontaktus, naujo laiško sukūrimą ir kitas. Bendras gmail kliento programinės dalies vaizdas pateiktas žemiau:



Pav. 41 Google Mail

6. Portletai (angl. Portlet)

Portletas yra tam tikras savarankiškas tinklapio komponentas, galintis veikti savarankiškai. Tam tikra prasme portletas yra tarsi atskira mini programa. Nors portletai yra naudojami jau daugiau nei penketą metų, kuriant interneto naršyklės pagrindu veikiančią programinę įrangą. Tačiau nepaisant pirminio susižavėjimo portletais, jie netapo labai paplitę ir populiarūs, Viena iš įvardijamų priežasčių, sąlygojusių mažesni naudojimą, nei pradžioje tikėtasi, yra vartotojo grafinė sąsaja. Todėl yra siejamos didelės viltys su Ajax technologijomis ir tikimasi, kad jos patobulins ir leis išpopuliarėti patiems portletams. Iš Ajax technologijų suteikiamų pranašumų portletų naudojimui išskiriami sekantios Ajax galimybės:

- XMLHttpRequest užklausa, įgalina neperkrauti viso puslapio, įvykus kiekvienam atskiram pokyčiui
- Galima pakeisti tinklapio vartotojo sąsajos vaizdą, priklausomai nuo vartotojo veiksmų ir serverio atsakymų
- Periodiniai atnaujinamai suteikia galimybę automatiškai atnaujinti atskirų portletų informaciją ir turinį, priklausomai nuo vartotojo bei kitų tuo metu programinę įrangą vartojančių vartotojų veiksmų

- Drag-and-drop galimybė suteikia galimybę parinkti atskirą portletą ir jį „nešioti“ po programinės įrangos vartotojo sąsają, kas daugelio analitikų nuomone suteikia puikią galimybę pačiam vartotojui susikurti jam labiausiai tinkančią ir priimtina vartotojo sąsają.

Geru Ajax portletų realizacijos realiu pavyzdžiu galėtų būti Google Homepage (<http://www.google.com/ig>) tinklapis. Iš esmės, Google Homepage visas sudarytas iš įvairių portletų. Atskirus portletus galima pridėti, paspaudus ant nuorodos “add stuff”. Paspaudus šią nuorodą galima prisidėti iš pateikiamo sąrašo įvairiausių portletų. Be to, vartotojas gali bet kada į savo pradinį puslapį pridėti naujus bei pašalinti jau esamus portletus. Google Homepage tinklapio bendras vaizdas pateiktas žemiau esančiame paveikslėlyje.

The screenshot shows the Google iGoogle homepage with the following components:

- Search Bar:** Google logo, search input field, and buttons for "Google Search" and "I'm Feeling Lucky".
- Navigation:** Links for Web, Images, Video, News, Maps, and more.
- Message:** "Interesting Photos of the Day has been removed. Undo"
- Home:** "Add a tab" button.
- Stock Market:**

DJIA	12398.01	-82.68	(-0.66%)
NASDAQ	2434.25	-19.18	(-0.78%)
NYSE	9025.04	-88.12	(-0.97%)

Delayed at least 15 minutes unless otherwise indicated.
Disclaimer
- Google Calendar:** Calendar for January 2007, showing dates from 24 to 31.
- Weather:** "01123", "Information is temporarily unavailable."
- NYT > Home Page:** Links to "Little Asia on the Hill", "Car Boom Puts Europe on Road to a Smoggy Future", "Police Ignore Serial Killings in Delhi Slum, Exposing Unequal Justice for India's Poor", "In Reversal, Toys 'R' Us Gives 3 Baby Prizes", "Seahawks 21, Cowboys 20: Season Slips Through Cowboys' Fingers".
- Gmail:** "Inbox (512)", "Hide preview".
- BBC News:** "Edition", "At least 12 dead in Iraq violence", "Live: FA Cup third round", "Tycoon may run for top ANC post", "Brown outlines 'patriotic vision'", "Winfrey offers students HIV tests".

Pav. 42 Google Homepage