

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Artūras Kneižys

**Bendrinis socialinių tinklų aplikacijų
programavimo sąsajų karkasas**

Magistro darbas

Darbo vadovas

doc. T. Blažauskas

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Artūras Kneižys

**Bendrinis socialinių tinklų aplikacijų
programavimo sąsajų karkasas**

Magistro darbas

Recenzentas

2011-05

dr. L. Motiejūnas

Vadovas

2011-05

doc. T. Blažauskas

Atliko

2006-05-27

IFM-9/2 gr. stud.
Artūras Kneižys

Kaunas, 2011

Summary

Today the social aspects of the internet are possibly the most important for most of the users. It seems like almost everyone has an account on Facebook, LinkedIn or some other social networking site. This means that incorporating your application and services into these social networking sites can bring you more publicity and users.

In this study we show how one could create a framework for working with multiple social network APIs. This framework is then used in conjunction with Ideas Bank system to explore the real world value of such functionality. While there are many examples of successful games or applications for simple users that incorporate into Facebook or other social networks, our example is that of a service made for business domain. Ideas Bank system is a way for people from management to get from their employees ideas for their projects or solutions to problems.

The testing of the created framework showed that it is easier to use than native Facebook API. Adding support for OpenSocial containers also proved to be relatively simple, as much of the code from Facebook integration could be reused, due to the unification of the APIs through the framework.

Finally, the results of the study on user behavior shows that as much as 25% - 30% of users came to the Ideas Bank system after finding out about it on Facebook, and 40% of all users tried to use the Facebook application, while 20% remained with it. However, these numbers might be too generous, since virtually no other form of advertising was done for the system, save for the word of mouth.

Turinys

1	Įvadas	3
1.1	Dokumento paskirtis	3
1.2	Santrauka.....	3
2	Paskirstytosios saityno sistemos	4
2.1	REST	4
2.2	XML-RPC	5
3	Socialinių tinklų aplikacijų programavimo sąsajos	5
3.1	Facebook platforma.....	5
3.1.1	Autentifikavimas.....	6
3.1.2	Graph API.....	8
3.2	OpenSocial	9
3.3	Apibendrinimas	11
4	Idėjų banko sistema	12
4.1	Projekto kūrimo pagrindas	12
4.1.1	Sistemos tikslai	12
4.1.2	Tikslinė auditorija.....	12
4.1.3	Rinkos apžvalga.....	12
4.1.4	Sąsajos su išorinėmis sistemomis aktualumas	12
4.2	Projekto įgyvendinimo fazės bei metodika	13
4.3	Projekto Apribojimai.....	14
4.3.1	Diegimo aplinka.....	14
4.3.2	Bendradarbiaujančios sistemos.....	14
4.3.3	Numatoma darbo vietos aplinka	14
4.4	Sistemos funkciniai reikalavimai	15
4.4.1	Veiklos kontekstas	15
4.4.2	Sistemos sudėtis.....	17

4.4.3	Egzistuojantys komponentai, kurie gali būti panaudoti.....	26
4.5	Architektūros specifikacija.....	27
4.5.1	Panaudojimo atvejų scenarijai	27
4.5.2	Sistemos statinis vaizdas.....	28
4.5.3	Sistemos dinaminis vaizdas	33
4.5.4	Sistemos komponentų fizinis išdėstymas	34
4.5.5	Duomenų bazės struktūra.....	35
5	Kokybės tyrimas	35
5.1	Darbo palengvinimas	36
5.2	Karkaso papildymas naujo socialinio tinklo palaikymu	37
6	Patobulinimai	39
6.1	Asinchroninis informacijos įkėlimas.....	39
6.2	Pakartotinis išorinės sistemos leidimų paprašymas	39
6.3	Idėjos vertinimo ir „man patinka“ susiejimas	40
7	Vartotojų elgesio tyrimas	40
8	Išvados	43
9	Literatūros sąrašas.....	44
10	TERMINŲ IR SANTRUMPŲ ŽODYNAS	46
11	Priedai	48
11.1	Surinkti vartotojų naudojimosi sistema duomenys.....	48

1 ĮVADAS

1.1 Dokumento paskirtis

Šio dokumento paskirtis yra aprašyti socialinių tinklų teikiamų aplikacijų programavimo sąsajų naudojimą bei sukurtą karkasą tam naudojimui apibendrinti bei palengvinti. Pasirinkus kelis socialinius tinklus, ištirti ir aprašyti karkaso pritaikymo tiems tinklams metodus ir sudėtingumą. Taip pat aprašyti ir įvertinti karkaso panaudojimą sukurtoje verslo sektoriaus realaus pasaulio programoje. Šiame dokumente atliekamas tyrimas, kaip sąryšis su socialiniais tinklais gali padėti populiarinti verslo sektoriaus saityno programą.

Šis dokumentas gali būti naudojamas kaip medžiaga tiriant, projektuojant, kuriant sistemas, naudojančias socialinių tinklų aplikacijų programavimo sąsajas. Atlikto tyrimo rezultatai gali būti panaudoti siekiant numatyti socialinių tinklų sąsajos poveikį sistemos populiarumui, tačiau tik jeigu kuriama sistema priklauso verslo sektoriui, ar yra su juo glaudžiai susijusi.

1.2 Santrauka

Šiuolaikiniame saityne vis daugiau dėmesio skiriama socialiniams jo aspektams. Bene kiekvienas aktyvus saityno vartotojas turi savo prieigą viename ar kitame socialiniame tinkle. Taip pat, daugelis tokių vartotojų savo prieigą pasitikrina net kelis kartus per dieną. Tai reiškia, kad įkėlus savo sistemos funkcionalumo dalį į tokius socialinius tinklus, bus pagerinamas ne tik sistemos plitimas, bet ir jos prieinamumas. Tai vartotojams suteikia galimybę lengviau matyti svarbias sistemos naujienas, būsenos pasikeitimus ir pan.

Daugelis sistemų, kurios galėtų pasinaudoti tokiais privalumais, kūrėjų apie socialinių tinklų panaudojimą net nepagalvoja. Daugelis tų, kurie svarsto tokią galimybę, nusprendžia, kad potenciali nauda nėra pakankamai aiški. Ta potenciali nauda dar labiau nublanksta, įvertinus galimus dažnus API pasikeitimus, o ypač jeigu tarp personalo nėra žmonių, dirbusių su socialinių tinklų sąsajomis.

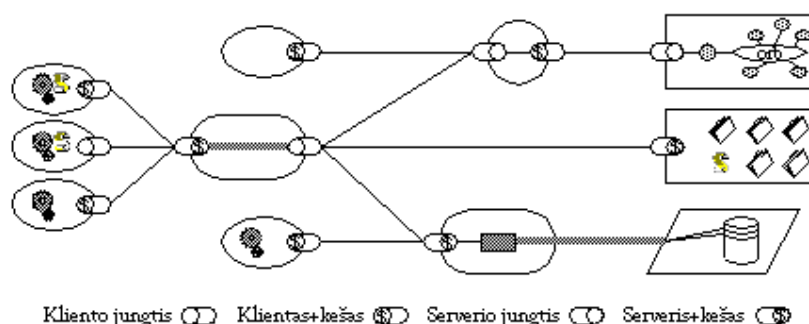
Šiame darbe siūlomas sprendimas, palengvinantis sąsajos su socialiniais tinklais kūrimą. Taip pat atlikto tyrimo rezultatai rodo, kad net verslo sektoriuje per socialinius tinklus atėjusių vartotojų kiekis gali siekti 30% visų vartotojų.

2 PASKIRSTYTOSIOS SAITYNO SISTEMOS

Paskirstytą sistemą sudaro daug nepriklausomų kompiuterių, bendraujančių tarpusavyje per kompiuterių tinklą (Tanenbaum, et al., 2007). Pradžioje panagrinėsime keletą galimų tokio bendravimo modelių: REST bei XML-RPC.

2.1 REST

REST (Representational State Transfer) nėra apibrėžiamas kaip protokolas. Tai yra architektūrinis stilius, ne įrankių rinkinys¹, rašo John Cowan internetiniame leidinyje „RESTful Web Services“ (Cowan, 2005).



1 pav. Pavyzdinė REST sistemos schema (Fielding, 2000)

REST tai **kliento – serverio architektūrinis stilius**. Tai reiškia, kad darbo sritys aiškiai atskiriamos. Šis stilius **neturi būsenų** – klientas kiekvienos užklauskos metu turi pasiūsti visą informacijai teisingai interpretuoti reikalingą informaciją, kad teisingai veiktų serveriui nesaugant jokios informacijos apie anksčiau to ar kurio kito kliento įvykdytas užklauskas. Tiek kliento, tiek serverio pusėje gali būti taikomas **kešavimas**. Tarp komponentų turi būti **vienodas ryšys**. Tai reiškia, kad ta pati REST jungtis gali aptarnauti kelis skirtingus serverius. REST sistema yra **sluoksniuota**, komponentams leidžiant „matyti“ tik tuos komponentus, kurie yra tame pačiame lygyje. REST palaiko kodo pateikimo į kliento sistemą (**Code-On-Demand**) stilių. Tai leidžia dalį programos pateiktį į vartotojo kompiuterį, taip sudarant sąlygas sistemos realaus laiko reakcijai į numatytus vartotojo veiksmus.

REST architektūra dažnai taikoma vien HTTP protokolui, kadangi REST reikalauja kūrimo, paklausimo, atnaujinimo bei trynimo operacijų, kurias HTTP turi kaip POST, GET, PUT ir DELETE atitinkamai (kai kuriuose šaltiniuose POST bei PUT yra sukeisti vietomis). Nors REST yra laikoma teoriškai atskirta nuo HTTP protokolo architektūrine kultūra, pats jos kūrėjas Roy Fielding, pasipiktinęs dėl neteisingo REST architektūrinio stiliaus pavadinimo naudojimo 2008 metais parašė straipsnį „REST APIs must be hypertext-driven“ (REST

¹ „An architectural style, not a toolkit“

saityno servisas turi būti pagrįsti hipertekstu) (Fielding, 2008). Tačiau turint galvoje, kad HTTP šiuo metu yra pagrindinis saityno protokolas, toks jo straipsnis yra pateisinamas.

Paprastai REST saityno serviso sąsaja susideda iš resursų rinkinio, nurodomo per URI (Pautasso, 2009).

2.2 XML-RPC

XML-RPC tai nuotolinio procedūrų paleidimo protokolas, naudojantis XML tų kvietimų užkodavimui ir HTTP kaip transporto mechanizmą²(Laurent, et al., 2001). Tai paprastas protokolas, leidžiantis iškviešti funkcijas (metodus) išorinėje sistemoje, nurodant joms reikiamus parametrus. Parametrai turi būti vieno iš 6 protokole aprašytų paprastų tipų, struktūros arba masyvo tipo. Paprastieji tipai: sveikas skaičius, loginė reikšmė, tekstas, slankaus kablelio skaičius, laikas bei data arba raidėmis (bei keliais specialiai parinktais simboliais) užkoduotas turinys. Struktūros tipus sudaro keli kintamieji, kurių kiekvienas taip pat gali būti tiek paprastojo tipo, tiek sudėtingojo. Masyvai taip pat gali turėti kelis bet kokio tipo elementus. Skirtumas tarp masyvo ir struktūros tas, kad struktūros kintamieji turi pavadinimus.

Kiekvieno funkcijos iškvietimo rezultatas – atgal gautas atsakymas. Jei funkcijos vykdymas pavyko, bus gražinama viena reikšmė – jos rezultatas. Jeigu įvyko klaida, gražinamas struktūros tipo kintamasis su klaidos kodu (sveikas skaičius) bei klaidos pranešimu (tekstas).

3 SOCIALINIŲ TINKLŲ APLIKACIJŲ PROGRAMAVIMO SĄSAJOS

Šiais laikais daugelis elektroninėje erdvėje aktyvių žmonių turi savo asmeninius puslapius MySpace, Facebook ar Twitter socialiniuose tinkluose. Verslo pasaulyje sekti savo pažintis bei kontaktus padeda LinkedIn. Šie socialiniai tinklai gali veikti kaip reklama bei informacijos apie asmenis šaltinis.

3.1 Facebook platforma

Facebook 2011 metų gegužės mėnesį turėjo virš 650 milijonų vartotojų visame pasaulyje ir virš 900 tūkstančių vartotojų Lietuvoje. Palyginus su Latvija bei Estija, tai gana didelis skaičius – šias šalis pagal vartotojų skaičių lenkiame atitinkamai 3,3 bei 2,3 karto (www.checkFacebook.com duomenys).

Facebook aplikacijų programavimo sąsaja leidžia trečioms šalims kurti Facebook aplikacijas. Šios aplikacijos veikia Facebook aplinkoje ir gali gauti duomenis apie vartotoją,

² „XML-RPC is a remote procedure call protocol which uses XML to encode its calls and HTTP as a transport mechanism.“

jo draugus, naujienas, ir panašiai. Aplikacijai prieinami duomenys priklauso nuo to, kokių priėjimų aplikacija vartotojo paprašė. Žinoma, vartotojas turi teisę nesuteikti prašomų priėjimų.

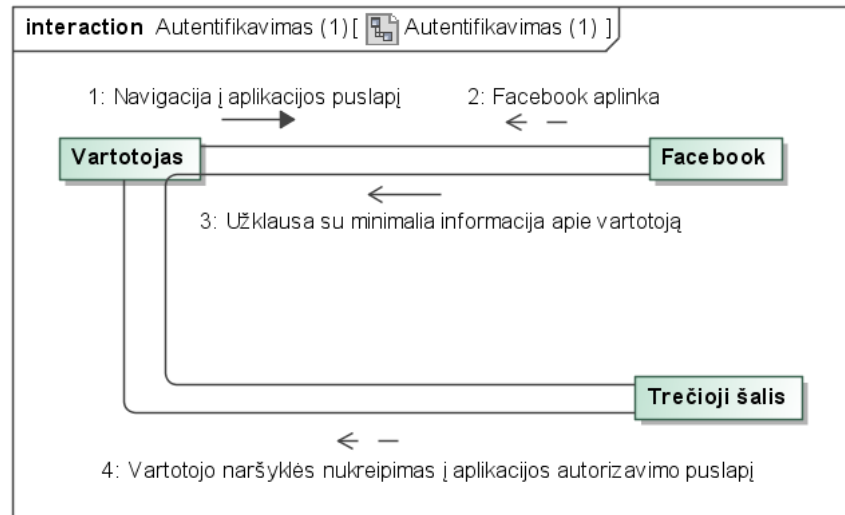
Programuojant sąsają su Facebook sistema svarbu sekti platformos būseną, kadangi Facebook vis prideda informacijos apie naujas žymes, paslaugų teikimo sąlygų pakeitimus (paprastai paaiškindami neaiškias vietas), sistemos sutrikimus bei, bene svarbiausia, platformos pasikeitimus, galinčius sugadinti jūsų aplikaciją³(Graham, 2008).

3.1.1 Autentifikavimas

Vartotojai gali prisidėti norimą aplikaciją prie savo Facebook puslapyje esančio aplikacijų (ar žaidimų, priklausomai nuo aplikacijos tipo) sąrašo. Tai atliekama nukreipus naršyklę aplikacijos adresu (pvz. http://www.Facebook.com/mano_aplikacija). Vartotojui pirmą kartą patekus į aplikacijos puslapį, aplikacija negauna beveik jokių duomenų apie vartotoją (2 pav.) Tokiu atveju aplikacijos dažniausiai pradeda autentifikavimo procesą.

Autentifikavimo proceso metu atliekamos 3 skirtingos funkcijos:

1. Vartotojo autentifikavimas;
2. Aplikacijos autorizacija;
3. Aplikacijos autentifikavimas.



2 pav. Facebook autentifikacija (1)

³ It's important to keep up-to-date with the platform status because Facebook is adding information about new tags, changes to the terms of service (typically clarifying hazy areas), systemwide outages, and, perhaps most important, platform changes that have the potential to break your application

Vartotojo autentifikavimas patvirtina, kad vartotojas yra tas, kuo dedasi(Facebook, 2011). Atliekant vartotojo autentifikavimą galimi du atvejai:

- Vartotojas neprisijungęs prie Facebook sistemos;
- Vartotojas jau yra prisijungęs.

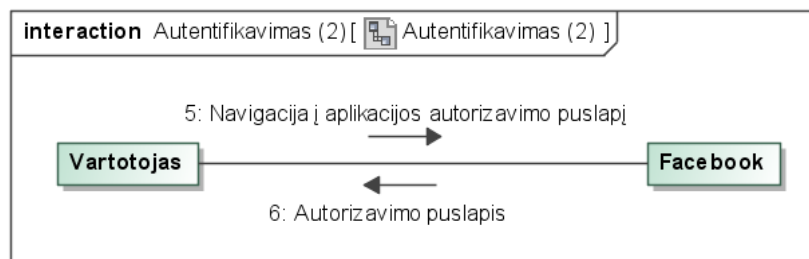
Tuo atveju, kai vartotojas dar nėra prisijungęs, jam pateikiamas prisijungimo langas, kur jis įveda savo el. pašto adresą ir slaptažodį. Jeigu vartotojas jau yra prisijungęs, tuomet patikrinamas prisijungimo sausainėlis (angl. *cookie*), išsaugotas vartotojo naršyklėje.

Aplikacijos autorizacijos metu vartotojui pateikiamas aplikacijai reikalingų leidimų sąrašas ir vartotojas nusprendžia, ar suteikti aplikacijai tuos leidimus (3 pav.) Apie aplikacijai reikalingus leidimus Facebook sužino iš aplikacijos perduoto parametro `scope` (apimtis). Prie aplikacijos nurodyto reikalingų leidimų sąrašo Facebook visuomet automatiškai prideda pagrindinės informacijos apie vartotoją (tokios kaip vardas, profilio paveikslėlis, lytis ir kita vieša informacija) skaitymo leidimą.

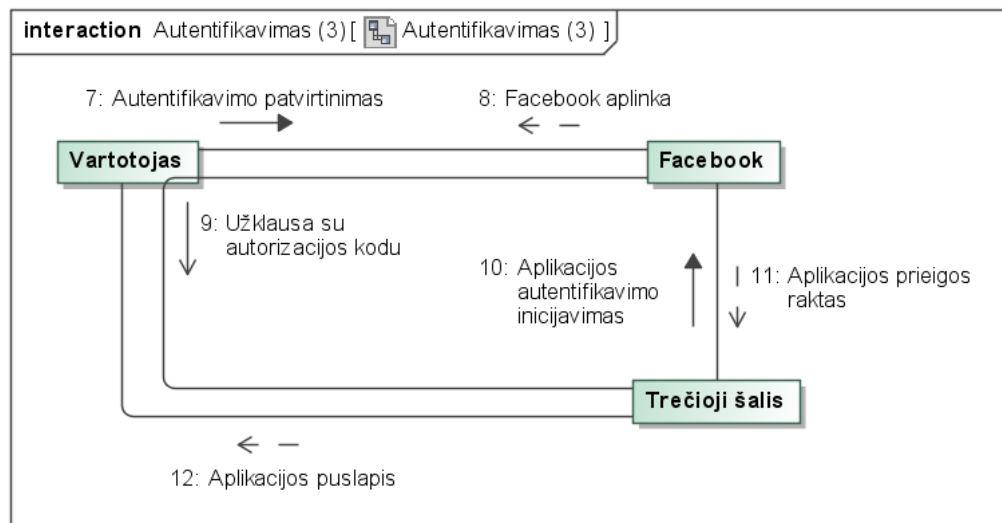
Sėkmingai įvykdžius autorizaciją, Facebook sistema nukreipia vartotojo naršyklę atgal į aplikacijos puslapį. Šiame puslapyje atidaroma aplikacija gauna autorizacijos kodą (4 pav.)

Aplikacijos autentifikavimo metu patvirtinama, kad aplikacija yra ta, kuria dedasi. Tam į Facebook serverį aplikacija siunčia autorizacijos kodą bei slaptą raktą, žinomą tik Facebook sistemai ir aplikacijai (4 pav.) Priešingai nei vartotojo autentifikavime ar aplikacijos autorizacijoje, vartotojo kompiuteris šioje operacijoje nedalyvauja. Taip slaptas aplikacijos raktas yra išlaikomas paslapyje. Į aplikacijos pasiųstą autentifikavimo užklausą Facebook sistema atsako prieigos raktu ir jo galiojimo laiku. Aplikacijos autorizacijos metu paprašius specialaus *offline_access* leidimo, prieigos raktas bus sugeneruotas nenurodytam laiko tarpui.

Vartotojas turi pasirinkimą bet kada pašalinti aplikaciją iš savo aplikacijų sąrašo. Tokiu atveju Facebook sistema aplikacijai apie tai praneša iš anksto nurodytu adresu perduodama vartotojo identifikacijos numerį. Gavusi tokį pranešimą, aplikacija turėtų pašalinti tam vartotojui išsaugotą prieigos raktą, kadangi Facebook sistemoje jis bus pažymėtas kaip nebegaliojantis.



3 pav. Facebook autentifikacija (2)



4 pav. Facebook autentifikacija (3)

3.1.2 Graph API

Facebook branduolyje yra socialinis grafas; žmonės ir sąryšiai su viskuo kas jiems rūpi⁴. Grafo (angl. *Graph*) API pateikia paprastą, nuoseklų Facebook socialinio grafo vaizdą, tolygiai atvaizduodamas objektus grafe (pvz. žmones, nuotraukas, įvykius ir puslapius) bei ryšius tarp jų (pvz. draugystės ryšius, bendrą turinį, nuotraukų žymes)⁵. (Facebook, 2011)

Knygoje *Analyzing Social Media Networks with NodeXL* autorius rašo:

Ar turite vieną grupę draugų? Ar žmonės iš mokyklos pažįsta jūsų draugus iš koledžo? Ar visi jūsų draugai yra dalis tos pačios grupės, ar skirtingų grupių? Facebook gali turėti atsakymą⁶ (Hogan, 2010).

Grafo API veikia 2.1 REST dalyje aprašytu principu. Informacija skaitoma ar ieškoma kreipiantis reikiamais adresais naudojant HTTP GET metodą. Informacija pateikiama Facebook sistemai naudojant HTTP POST metodą. Informacijos trynimasis atliekamas paprastai retai naudojamu DELETE metodu. Tiesa, kadangi kai kurie HTTP klientai gali nepalaikyti DELETE metodo, trynimą taip pat galima atlikti POST metodu, perduodant papildomą argumentą `method=delete`. Grafo API rezultatus grąžina JSON objektais.

JSON – tai JavaScript objektų užrašymo būdas (angl. JavaScript Object Notation), tačiau jis yra pakankamai paprastas. Būtent dėl šio paprastumo JSON yra patogu naudoti ir kitose programavimo kalbose. JSON objektai atvaizduojami kaip vardo-reikšmės poros.

⁴ At Facebook's core is the social graph; people and the connections they have to everything they care about

⁵The Graph API presents a simple, consistent view of the Facebook social graph, uniformly representing objects in the graph (e.g., [people](#), [photos](#), [events](#), and [pages](#)) and the connections between them (e.g., friend relationships, shared content, and photo tags).

⁶ Do you have a single group of friends? Do people from high school know your buddies from college? Are your friends all a part of the same cluster or different clusters? Facebook may have the answer.

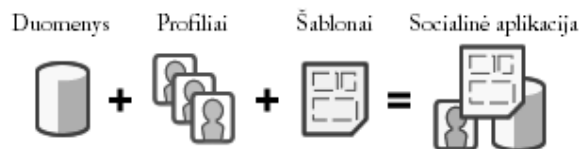
3.2 OpenSocial

Tai interneto giganto Google socialinių tinklų API specifikacija. Jawad Shuaib savo prezentacijoje „Facebook vs. Open Social” neblogai susumuoja kas tai yra:

- Google atsakas į Facebook aplikacijų kūrimo platformą;
- „Parašyk kartą, paleisk betkur“;
- „Google Widgets pripompuoti steroidų“;
- Išleista per anksti(Shuaib, 2007).

Iš tikro, tai labai galinga bei didelė sistema. Tiksliau specifikacija, kadangi iki šiol nei vienas socialinis tinklas nėra šios specifikacijos iki galo įgyvendinęs. Tačiau tokių tinklų, kurie naudoja OpenSocial yra tikrai daug: tai ir LinkedIn, ir MySpace, Orkut, Friendster, hi5 bei Ning. Taip pat OpenSocial palaiko ne tik socialiniai tinklai, pavyzdžiui Oracle, iGoogle ir net Yahoo!. Viename oficialiame sąrašė suskaičiavau 34 paprastiem vartotojam prieinamus tinklapius, turinčius dalinį OpenSocial palaikymą bei 14 tinklapių, prieinamų programuotojams (kai kurie tinklalapiai yra abiejuose sąrašuose) (OpenSocial, 2009).

OpenSocial specifikacijoje aprašomi trys socialinės aplikacijos architektūros šablonai, tačiau paprastai struktūra būna panaši (5 pav.)



5 pav. Standartinė socialinių aplikacijų struktūra (komponentai)

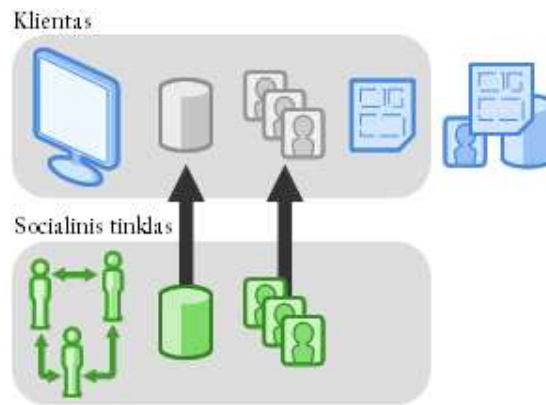
OpenSocial „ekosistemoje” šie komponentai gali ateiti iš skirtingų vietų (6 pav.).



6 pav. Galimi komponentų šaltiniai

„Social Mashup“

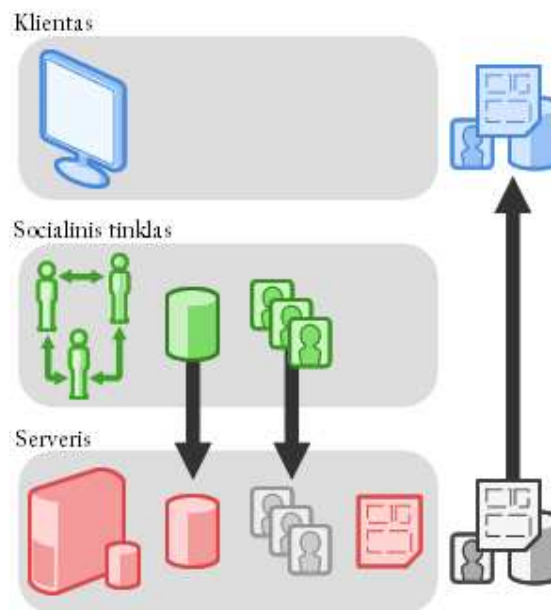
Toks architektūros šablonas tinka nedidelėms aplikacijoms kurti. Tam pakanka vartotojo kompiuteryje paleidžiamo javascript ir / arba flash kodo. Šio šablono schema pateikta 7 pav.



7 pav. "Social Mashup" šablonas

Socialinės aplikacijos

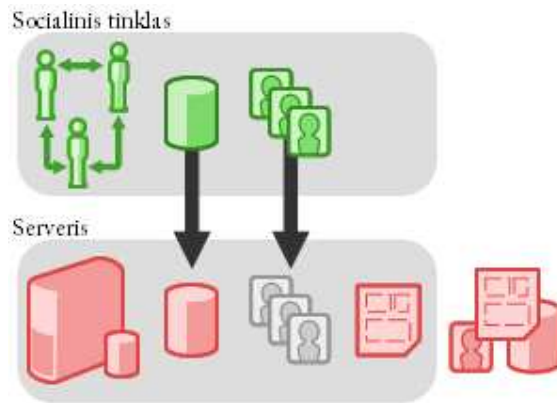
Socialinė aplikacija veikia socialinio tinklo viduje, tačiau pasikliauja išoriniu serveriu duomenų apdorojimui ir puslapio pateikimui. Tokios aplikacijos suteikia daugiau funkcionalumo, tačiau didėjant vartotojų skaičiui gali iškilti plečiamumo problemų. Šablono schema pateikta 8 pav.



8 pav. Socialinės aplikacijos šablonas

Socialiniai tinklalapiai

Šis OpenSocial šablonas gana stipriai skiriasi nuo kitų dviejų, kadangi jo paskirtis – papildyti išorinę sistemą socialinių tinklų savybėmis. Tai reiškia, kad aplikacija veikia už socialinio tinklo ribų. Iš socialinio tinklo ji per REST arba RPC priegą paima jai reikiamus socialinius duomenis. Šio šablono schema pateikta 9 pav.



9 pav. Socialinio tinklalapio šablonas

Deje, šiuo metu OpenSocial dar jokiam tinklalapyje nėra iki galo įgyvendinta. Tai didelė ir galinga specifikacija, tačiau kol kas ją atitinkančios didelės ir galingos programinės įrangos nėra. Tiesą sakant, OpenSocial galimybių gausa gali pasunkinti tinklapių kūrimą. Nors ir specifikacijos, tokios kaip OpenSocial, aprašo metodus kaip integruoti tik tą specifikacijos dalį, kuri reikalinga tam tikrai sistemai, šios dalys vis tiek apima daug panaudos atvejų, kurie gali būti nebūtini tai sistemai ar aplikacijai⁷ (LeBlanc, 2011).

Kita vertus, OpenSocial turi neblogus įrankius, leidžiančius paspartinti programuotojų darbą kuriant aplikacijas. Ne taip senai išleistas OSDE (OpenSocial Development Environment – OpenSocial programavimo aplinka) įgalina programuotoją programuoti bei testuoti savo aplikacijas be OpenSocial konteinerio. OSDE parašyta kaip dar vienas Eclipse šeimos (tai bene labiausiai paplitusi nemokama programinės įrangos kūrimui palengvinti skirtų įrankių platforma) įrankis. Jis turi net dažnai pasikartojančių aplikacijos dalių kūrimui palengvinti skirtus vedlius (OpenSocial Blog, 2009).

3.3 Apibendrinimas

Panagrinėjome du skirtingus socialinių tinklų servisus. Facebook API yra labiau išbaigtas, tačiau atitinkamai jame yra daugiau apribojimų. Bene didžiausias skirtumas yra tame, kad Facebook API tinka tik Facebook socialiniam tinklui. O štai Google OpenSocial sistemą naudoja virš 30 puslapių. Tačiau nei viename jų ji nėra iki galo įgyvendinta, be to gana dažnai išlenda įgyvendinimo klaidos. Visgi jei reiktų rinktis vieną API, dėl dydžio, paplitimo, pagalbinių įrankių bei tikslinės auditorijos (LinkedIn), rinkčiausi programuoti OpenSocial, tačiau realybė tokia, kad didžiausias vartotojų padengiamumas bus pasiekiamas su Facebook sistema.

⁷ Even though specifications like OpenSocial define methods for integrating only the portions of the specifications that are required for the particular implementation, these pieces still encompass a lot of use cases that may not be needed for a custom container or application solution.

4 IDĖJŲ BANKO SISTEMA

Idėjų banko sistema šiame darbe bus naudojama kaip socialinio karkaso (Soc. karkaso) testavimo platforma. Vienas iš esminių šios sistemos požymių yra tai, kad ji priklauso verslo sektoriui.

4.1 Projekto kūrimo pagrindas

Atsižvelgdami į rinkos poreikį inovacijoms, bei neišnaudojamus potencialius rinkos dalyvius, t.y. pavienius vartotojus gebančius pasiūlyti tikslingas idėjas, sukūrėme produkto koncepciją, kurio pagalba bus sukurta terpė, kurioje vartotojai biržos principu galės pateikti idėjų, probleminių sričių užklausas, ir taip gauti kvalifikuotų specialistų idėjinius pasiūlymus, probleminių sričių sprendimus.

4.1.1 Sistemos tikslai

Sistemos paskirtis – patogus įrankis kurio pagalba vartotojų grupė gali komunikuoti tarpusavyje, keistis patirtimi ir naudinga informacija:

- vadovai – idėjų paklausos grupė – vartotojams leidžiama skelbti idėjų, probleminių sričių sprendimų užklausas, nominuoti, reitinguoti ir komentuoti pateiktas idėjas ir sprendimus;
- darbuotojai – idėjų pasiūlos grupė – vartotojams leidžiama skelbti idėjas ir sprendimus aktyvioms užklausoms, komentuoti, reitinguoti kitų vartotojų idėjų pasiūlą.

4.1.2 Tikslinė auditorija

Idėjų banko internetinė sistema skirta organizacijoms, atstovaujančioms dideles vartotojų grupes. Šio produkto panaudojimas būtų ypač aktualus toms organizacijoms, kurių vartotojai pasiskirstę problematiniuose Lietuvos rajonuose. Idėjų banko integracija į organizaciją siekiama tiesiogiai paspartinti inovacinę organizacijos plėtrą.

4.1.3 Rinkos apžvalga

Lietuvos regiono mastu, tai būtų inovacinis produktas, nes šiuo metu vietinė rinkos pasiūloje analogiškų įrankių nėra. Pasauliniu mastu, galime paminėti kelias organizacijas, profesionaliai užsiimančias idėjų surinkimu ir jų sisteminiu, jas apžvelgsime preliminarioje konkurencinėje analizėje.

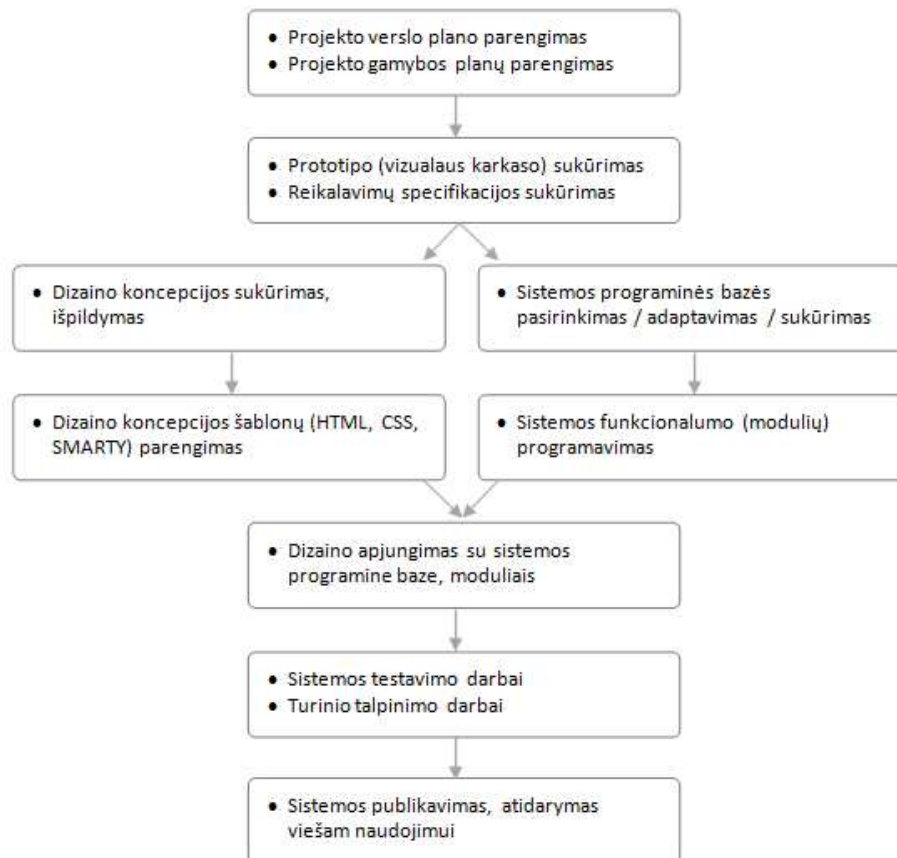
4.1.4 Sąsajos su išorinėmis sistemomis aktualumas

Sistemos prijungimu prie populiariausių socialinių tinklų puslapių siekiama reklamuoti sistemą, taip pasiekiant internete aktyvių žmonių, ar net tikslinę įmonių vadovų

auditoriją. Pavyzdžiui, pagal „Anderson Analytics“ rinkos tyrimų agentūros 2008 metais atlikto tyrimo duomenis, beveik 60% LinkedIn socialinio tinklo orientuoto į darbo reikalų pažintis vartotojų gauna didesnes nei vidutinės pajamas bei užima vadovaujančias ar konsultacines pareigas⁸ (O'MALLEY, 2008). Įmonės idėjų banko statistikos įdėjimas į įmonės profilį, pavyzdžiui, facebook socialiniame tinkle, taip pat gali būti gera reklama pačiai įmonei. Dar geresnių rezultatų būtų galima pasiekti leidžiant įmonėms socialiniuose tinkluose automatiškai talpinti viešas firmos užduotis, klausimus bei žmonių idėjas.

Taip pat integracija su Idėjų banko sistema gali būti naudinga ir patiems socialiniams tinklams. Pavyzdžiui, Justas, savo straipsnyje „LinkedIn panaudojimo scenarijai“ rašo „pastebėjau, jog vienos lietuviškos atrankų bendrovės atstovė jau suprato LinkedIn naudą ir puikiai renka konktatus, o vėliau jais pasinaudoja siūsdama darbo pasiūlymus“ (Marketer.lt, 2008). Vartotojas, pateikdamas savo idėjų statistiką, suteikia papildomą informaciją tokiems darbo pasiūlymus platinantiems žmonėms.

4.2 Projekto įgyvendinimo fazės bei metodika



10 pav. Projekto įgyvendinimo fazės

⁸ „Nearly 60% of the professional-focused social network's [LinkedIn] users have high personal incomes and hold executive-level or consultant positions, according to a new study from market research firm Anderson Analytics.“

Projektuojant informacinę sistemą atsižvelgiama į pagrindinius aspektus, vartojimo paprastumą (angl. *usability*), bei informacijos suradimo ir naršymo paprastumą (angl. *findability*).

Projektuojama pasitelkiant UML modeliavimo kalbą (naudojant MagicDraw programą), Volere reikalavimų užrašymo šablonus. Duomenų bazė projektuojama naudojant esybių – ryšių diagramas.

4.3 Projekto Apribojimai

4.3.1 Diegimo aplinka

Sistema diegiama serveryje, turinčiame apache, PHP 5.2 ir MySQL programinę įrangą. Sistema neturėtų pasikliauti HTTPS protokolo palaikymu serveryje. Vartotojo kompiuteriuose sistemos pateikiamas internetinis puslapis turi gerai veikti per Internet Explorer 7 bei 8, Mozilla Firefox, Opera, Safari bei Google Chrome interneto naršyklės. Sistema taip pat turėtų būti paruošta integravimui į socialinių tinklų puslapius (pagrindė Facebook, su galimybe praplėsti sistemos funkcionalumą pridėdant naujų).

4.3.2 Bendradarbiaujančios sistemos

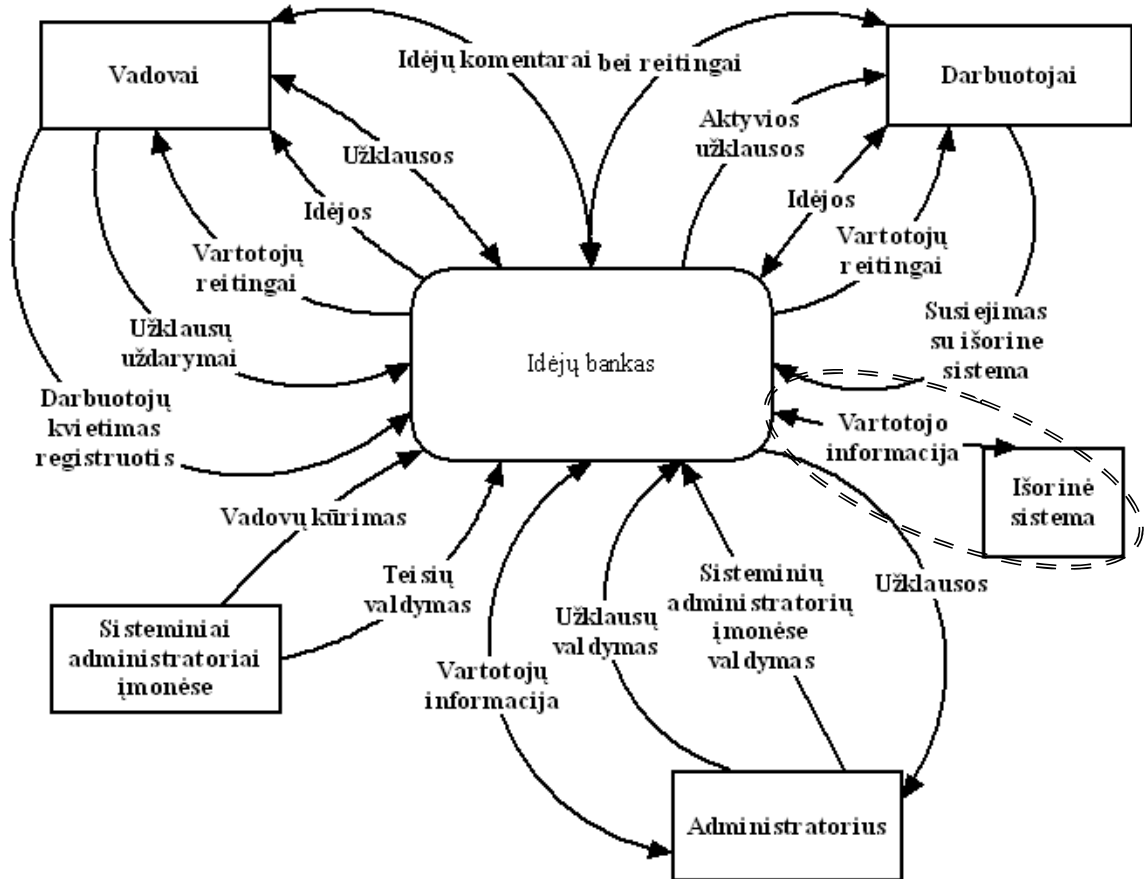
Sistema turi integruotis į Facebook sistemą. Vartotojai turėtų galėti įsidėti informaciją apie savo profilį sistemoje į savo Facebook puslapį. Turėtų būti paruošti programiniai įrankiai, palengvinantys analogišką funkcionalumą teikiančių sąsajų su kitomis išorinėmis sistemomis programavimą. Numatomas sistemų bendravimas per HTTP ar HTTPS protokolus.

4.3.3 Numatoma darbo vietos aplinka

Vartotojų darbo vietai reikalingas tik kompiuteris su interneto ryšiu ir šiuolaikine interneto naršykle (neįskaitant Internet Explorer 6).

4.4 Sistemos funkciniai reikalavimai

4.4.1 Veiklos kontekstas



11 pav. Idėjų banko veiklos kontekstas

Toliau (1 lentelė) kaip įvykiai su savo informacijos srautais išrašomi veiklos konteksto schemeje (11 pav.) pavaizduoti veiksmai.

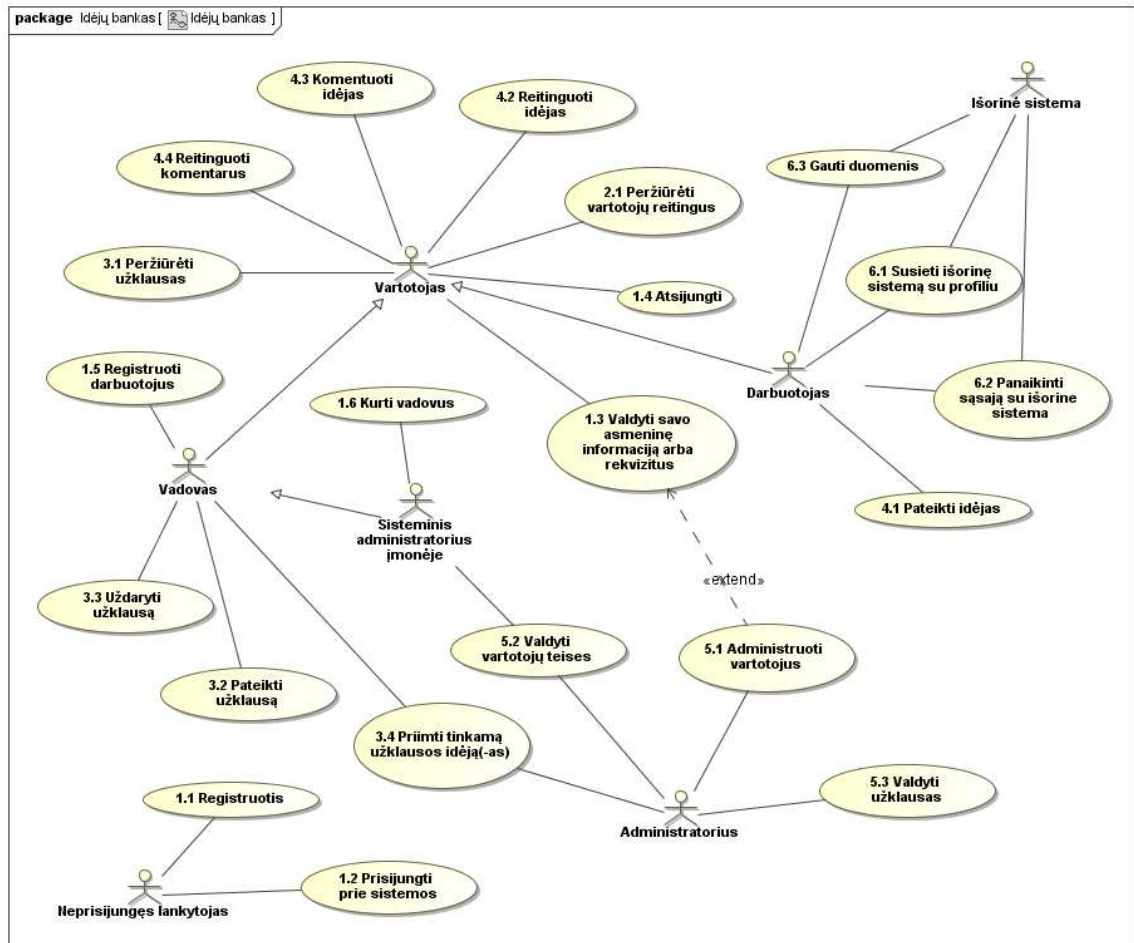
1 lentelė Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/Išeinantys informacijos srautai
1	Administratorius sukuria naują sisteminį administratorių (įmonę)	Sisteminio administratoriaus duomenys (<i>in</i>)
2	Sisteminis administratorius įmonėje sukuria vadovą	Vadovo duomenys (<i>in</i>)
3	Vadovas pateikia naują užklausą	Duomenys apie užklausą (<i>in</i>)
4	Vadovas pakviečia registruotis naują vartotoją	Vartotojo vardas bei el. paštas (<i>in</i>) Pakvietimo žinutė siunčiama nurodytu adresu (<i>out</i>)

5	Vartotojas pateikia idėją užklausiai	Idėja (<i>in</i>)
6	Vartotojas peržiūri norimas užklausas	Užklausių filtras (<i>in</i>) Duomenys apie užklausas ir jų idėjas (<i>out</i>)
7	Vartotojas paprašo vartotojų reitingų	Vartotojų reitingai (<i>out</i>)
8	Vartotojas įvertina idėją	Idėja, įvertinimas (<i>in</i>) Perskaičiuotas idėjos įvertinimas (<i>out</i>)
9	Vartotojas parašo komentarą idėjai	Idėja, komentaras (<i>in</i>)
10	Vartotojas įvertina komentarą	Komentaras, įvertinimas (<i>in</i>) Perskaičiuotas komentaro įvertinimas (<i>out</i>)
11	Užklaustos savininkas pasirenka sprendimą	Užklausa, sprendimas (<i>in</i>)
12	Užklaustos savininkas uždaro įvykdytą užklausą (tikslą)	Užklausa (<i>in</i>)
13	Vartotojas susieja išorinę sistemą su jo profiliu Idėjų banke	Susiejimo duomenys iš išorinės sistemos (<i>in</i>) Susiejimo duomenys išorinei sistemai (<i>out</i>)
14	Idėjų banko sistema paprašo duomenų apie vartotojo profilį	Vartotojo profilis iš išorinės sistemos (<i>in</i>)
15	Administratorius redaguoja vartotojų duomenis	Seni vartotojų duomenys (<i>in</i>) Nauji vartotojų duomenys (<i>out</i>)
16	Administratorius valdo užklausas	Seni užklaustos duomenys (<i>in</i>) Nauji užklaustos duomenys (<i>out</i>)

4.4.2 Sistemos sudėtis

4.4.2.1 Sistemos ribos



12 pav. Idėjų banko panaudos atvejų diagrama

4.4.2.2 Panaudojimo atvejų sąrašas

1.1 Registruotis

Aktorius	Neprisijungęs lankytojas
Aprašas	Leidžia lankytojui užsiregistruoti sistemoje.
Prieš sąlyga	Tokio vartotojo dar nėra sistemoje.
Sužadinimo sąlyga	Lankytojas pasirenka užsiregistravimo mygtuką.
Po-sąlyga	Sistemoje atsiranda naujas idėjų paklausos arba idėjų pasiūlos grupei priklausantis vartotojas.

1.2 Prisijungti prie sistemos

Aktorius	Neprisijungęs lankytojas
Aprašas	Leidžia lankytojui prisijungti prie sistemos.
Prieš sąlyga	Vartotojas, kuriuo nori prisijungti lankytojas, turi egzistuoti sistemoje.
Sužadinimo sąlyga	Vartotojas nurodo prisijungimo vardą ir slaptažodį bei paspaudžia prisijungimo mygtuką.
Po-sąlyga	Sistema identifikuoja lankytoją kaip konkretų registruotą vartotoją ir toliau leidžia vykdyti jam leidžiamus veiksmus.

1.3 Valdyti savo asmeninę informaciją arba rekvizitus

Aktorius	Vartotojas
Aprašas	Leidžia vartotojui redaguoti asmeninę informaciją arba rekvizitus (priklausomai nuo to, ar vartotojas yra fizinis asmuo, ar įmonė).
Prieš sąlyga	Vartotojas turi būti prisijungęs.
Sužadinimo sąlyga	Vartotojas iš meniu pasirinko profilio valdymą.
Po-sąlyga	Vartotojo profilis pakeistas.

1.4 Atsijungti

Aktorius	Vartotojas
Aprašas	Atjungia vartotoją nuo sistemos, kad jis saugiai galėtų palikti darbo vietą.
Prieš sąlyga	Vartotojas turi būti prisijungęs.
Sužadinimo sąlyga	Vartotojas paspaudžia atsijungimo nuo sistemos mygtuką.
Po-sąlyga	Vartotojas sistemai tampa neprisijungusiu lankytoju.

1.5 Registruoti darbuotojus

Aktorius	Vadovas
Aprašas	Pakviečia naujus vartotojus registruotis į sistemą.
Prieš sąlyga	Vartotojas turi būti prisijungęs bei turėti vadovo ar aukštesnes teises.
Sužadinimo sąlyga	Vartotojas pasirenka vadovų registravimo funkciją
Po-sąlyga	Sistemoje sukuriamas naujas vartotojas. Jis el. paštu gauna pakvietimą prisijungti prie sistemos ir papildyti duomenis.

1.6 Kurti vadovus

Aktorius	Sisteminis administratorius įmonėje
Aprašas	Pakviečia į sistemą naujus vadovus
Prieš sąlyga	Vartotojas turi būti prisijungęs bei turėti sistemos administratoriaus statusą.
Sužadinimo sąlyga	Vartotojas pasirenka vadovų registravimo funkciją
Po-sąlyga	Sistemoje sukuriamas naujas vartotojas. Jis el. paštu gauna pakvietimą prisijungti prie sistemos ir papildyti duomenis. Vartotojui automatiškai priskiriamas vadovo statusas

2.1 Peržiūrėti vartotojų reitingus

Aktorius	Vartotojas
Aprašas	Pateikia vartotojų reitingus.
Prieš sąlyga	Vartotojas turi būti prisijungęs.
Sužadinimo sąlyga	Rodoma vartotojo informacija (pavyzdžiui prie jo rašytos idėjos); Vartotojas peržiūri kito vartotojo profilį.
Po-sąlyga	Vartotojas mato informaciją apie vartotojo reitingus.

3.1 Peržiūrėti užklausas

Aktorius	Vartotojas
Aprašas	Pateikia informaciją apie atfiltruotas užklausas.
Prieš sąlyga	Vartotojas turi būti prisijungęs arba jis turi būti administratorius.
Sužadinimo sąlyga	Vartotojas iš meniu pasirenka užklausų sąrašą.
Po-sąlyga	Vartotojas mato norimų užklausų informaciją.

3.2 Pateikti užklausa

Aktorius	Vadovas
Aprašas	Leidžia vartotojui pateikti naują užklausa sistemai.
Prieš sąlyga	Vartotojas turi būti prisijungęs.
Sužadinimo sąlyga	Vartotojas paspaudžia užklauskos pateikimo mygtuką.
Po-sąlyga	Nauja užklausa atsiranda sistemoje.

3.3 Uždaryti užklausa

Aktorius	Vadovas
Aprašas	Leidžia vartotojui uždaryti užklausa.
Prieš sąlyga	Vartotojas turi būti prisijungęs ir būti norimos uždaryti užklauskos savininku.
Sužadinimo sąlyga	Vartotojas, iš jo užklauskos peržiūros, pasirenka uždarymo veiksmą.
Po-sąlyga	Užklausa uždaroma (tampa neaktyvia).

3.4 Priimti tinkamą užklauskos idėja(-as)

Aktorius	Vadovas; Administratorius
Aprašas	Leidžia vartotojui uždaryti užklausa, nustatant jam labiausiai patikusias idėjas.
Prieš sąlyga	Vartotojas turi būti prisijungęs ir būti norimos uždaryti užklauskos savininku; Užklausa turi turėti bent vieną idėją.
Sužadinimo sąlyga	Vartotojas pasirenka idėją kaip tinkamą.
Po-sąlyga	Užklausa uždaroma; pasirinktos idėjos pažymimos kaip jos „sprendimas“ arba „sprendimo dalis“.

4.1 Pateikti idėjas

Aktorius	Darbuotojas
Aprašas	Leidžia vartotojui pateikti savo idėją pasirinktai užklausiai.
Prieš sąlyga	Vartotojas turi būti prisijungęs ir pasirinkęs aktyvią užklausa.
Sužadinimo sąlyga	Vartotojas paspaudžia idėjos pateikimo mygtuką pasirinktos užklauskos peržiūroje.
Po-sąlyga	Prie užklauskos atsiranda įvesta idėja.

4.2 Reitinguoti idėjas

Aktorius	Vartotojas
Aprašas	Leidžia vartotojui balsuoti už ar prieš ne jo idėją, taip keičiant jos reitingą.
Prieš sąlyga	Vartotojas turi būti prisijungęs ir nebūti pasirinktos idėjos autoriumi; Vartotojas turi būti dar nebalsavęs už pasirinktą idėją.
Sužadinimo sąlyga	Vartotojas prie idėjos pasirenka, ar balsuos už, ar prieš.
Po-sąlyga	Idėjos reitingas atitinkamai pasikeičia.

4.3 Komentuoti idėjas

Aktorius	Vartotojas
Aprašas	Leidžia vartotojui parašyti komentarą prie norimos idėjos.
Prieš sąlyga	Vartotojas turi būti prisijungęs.
Sužadinimo sąlyga	Vartotojas prie idėjos paspaudžia komentaro rašymo mygtuką.
Po-sąlyga	Prie idėjos išsaugomas ir matomas parašytas komentaras.

4.4 Reitinguoti komentarus

Aktorius	Vartotojas
Aprašas	Leidžia vartotojui balsuoti už ar prieš ne jo komentarą, taip keičiant jo reitingą.
Prieš sąlyga	Vartotojas turi būti prisijungęs ir nebūti pasirinkto komentaro autoriumi; Vartotojas turi būti dar nebalsavęs už pasirinktą komentarą.
Sužadinimo sąlyga	Vartotojas prie komentaro pasirenka, ar balsuos už, ar prieš.
Po-sąlyga	Komentaro reitingas atitinkamai pasikeičia.

5.1 Administruoti vartotojus

Aktorius	Administratorius
Aprašas	Suteikia administratoriui galimybę matyti sistemos vartotojų sąrašą, keisti jų duomenis taip, kaip jie tai gali daryti patys bei aktyvuoti ar deaktyvuoti vartotojus.
Prieš sąlyga	-
Sužadinimo sąlyga	Administratorius iš meniu punkto pasirenka vartotojų sąrašo punktą.
Po-sąlyga	Administratorius mato sistemos vartotojų sąrašą; jo keisti vartotojų duomenys išsaugomi sistemoje.

5.2 Valdyti vartotojų teises

Aktorius	Sistemos administratorius įmonėje; Administratorius
Aprašas	Suteikia administratoriui galimybę matyti sistemos vartotojų teises ir keisti jas
Prieš sąlyga	-
Sužadinimo sąlyga	Administratorius iš vartotojų sąrašo pasirenka vartotojo teisių valdymo punktą.
Po-sąlyga	Administratorius mato sistemos vartotojų sąrašą; jo keisti vartotojų duomenys išsaugomi sistemoje.

5.3 Valdyti užklausas

Aktorius	Administratorius
Aprašas	Suteikia administratoriui galimybę matyti užklausų sąrašą bei keisti jų duomenis.
Prieš sąlyga	-
Sužadinimo sąlyga	Administratorius iš meniu pasirenka užklausų sąrašo punktą.
Po-sąlyga	Administratorius mato sistemos vartotojų sąrašą; jo keisti vartotojų duomenys išsaugomi sistemoje.

6.1 Susieti išorinę sistemą su profiliu

Aktorius	Darbuotojas (inicijuoja); Išorinė sistema (dalyvauja)
Aprašas	Leidžia vartotojui susieti savo profilį Idėjų banko sistemoje su profiliu išorinėje sistemoje, suteikiant išorinei sistemai teisę gauti duomenis apie vartotoją.
Prieš sąlyga	Vartotojas turi būti prisijungęs; Išorinė sistema turi būti paruošta susiejimui.
Sužadinimo sąlyga	Vartotojas iš savo profilio pasirenka susiejimo su išorine sistema mygtuką.
Po-sąlyga	Idėjų banko sistema bei išorinė sistema gali identifikuoti viena kitą; Idėjų banko sistema gali gauti duomenis apie vartotoją iš išorinės sistemos.

6.2 Panaikinti sąsają su išorine sistema

Aktorius	Darbuotojas (inicijuoja); Išorinė sistema (dalyvauja)
Aprašas	Leidžia vartotojui panaikinti jo profilio sąryšį su išorine sistema.
Prieš sąlyga	Vartotojas turi būti susiejęs savo profilį su išorine sistema.
Sužadinimo sąlyga	Vartotojas savo profilyje pasirenka konkretaus sąryšio su išorine sistema naikinimo mygtuką.
Po-sąlyga	Idėjų banko sistema nebegauna duomenų iš išorinės sistemos.

6.3 Gauti duomenis

Aktorius	Darbuotojas (inicijuoja); Išorinė sistema (dalyvauja)
Aprašas	Gauna vartotojo duomenis (nuotrauką ir pan.) iš išorinės sistemos
Prieš sąlyga	Vartotojas turi būti susietas su išorine sistema.
Sužadinimo sąlyga	Po profilių susiejimo Idėjų banko sistemoje trūksta duomenų.
Po-sąlyga	Idėjų banko sistema gauna duomenis apie vartotoją.

4.4.2.3 Funkciniai reikalavimai

Čia Volere šablonu aprašomi aktualiausi funkciniai reikalavimai

Reikalavimas #	1	Reikalavimo tipas	auth.	Įvykis/panaudojimo atvejis #	1.1
Aprašymas	Vartotojo registracija sistemoje, pateikiant norimus prisijungimo vardą ir slaptažodį bei pasirenkant norimą vartotojo grupę (idėjų pasiūlos ar paklausos).				
Pagrindimas	Sistemai reikalingas vartotojų sąrašas jų informacijos kaupimui bei saugumui užtikrinti.				
Šaltinis	Neprisijungęs lankytojas				
Tikimo kriterijus	Registruotis galima tik su dar sistemoje neužregistruotu prisijungimo vardu. Po registracijos vartotojas turi galėti prisijungti prie sistemos.				
Užsakovo tenkinimas	0	Užsakovo netenkinimas	5		
Priklausomybės	2,40,41	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				
Reikalavimas #	2	Reikalavimo tipas	auth.	Įvykis/panaudojimo atvejis #	1.2
Aprašymas	Vartotojo prisijungimas prie sistemos, pateikus prisijungimo vardą ir slaptažodį.				
Pagrindimas	Sistema turi sekti bei riboti vartotojų veiksmus pagal jų informaciją sistemoje.				
Šaltinis	Neprisijungęs lankytojas				
Tikimo kriterijus	Prisijungti gali tik teisingus duomenis nurodęs vartotojas. Po prisijungimo sistema turi susieti darbo sesiją su vartotojo duomenimis.				
Užsakovo tenkinimas	0	Užsakovo netenkinimas	5		
Priklausomybės	1, 3	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				
Reikalavimas #	3	Reikalavimo tipas	auth.	Įvykis/panaudojimo atvejis #	1.4
Aprašymas	Vartotojo atsijungimas nuo sistemos, norint baigti darbą su ja.				
Pagrindimas	Vartotojas turi galėti atsijungti nuo sistemos dėl saugumo (jei naudoja viešą kompiuterį ar kas nors gali pasinaudoti jo kompiuteriu) ar norėdamas prisijungti su kitu vartotoju.				
Šaltinis	Prisijungęs vartotojas				
Tikimo kriterijus	Vartotojui atsijungiant jo darbo sesija tampa nebesusieta su jo vartotojo duomenimis ir jis sistemoje tampa neprisijungusiu lankytoju.				
Užsakovo tenkinimas	0	Užsakovo netenkinimas	4		

Priklausomybės Istorija	2	Užregistruotas 2009-10-01	Konfliktai	Nėra
Reikalavimas # Aprašymas Pagrindimas Šaltinis Tikimo kriterijus Užsakovo tenkinimas	4	Reikalavimo tipas asm.d Vartotojo asmeninių duomenų redagavimas, jei vartotojas – fizinis asmuo. Kadangi sistema susijusi su vartotojo reputacija, galimomis darbo galimybėmis bei mokamais kreditais, tikslinga saugoti vartotojo vardą bei pavardę, telefono numerį, adresą ir pan. Fizinis asmuo – vartotojas Išsaugoti duomenys turi atitikti vartotojo atliktus pakeitimus.	Įvykis/panaudojimo atvejis #	1.3
Priklausomybės Istorija	5,28	Užregistruotas 2010-03-05	Konfliktai	Nėra
Reikalavimas # Aprašymas Pagrindimas Šaltinis Tikimo kriterijus Užsakovo tenkinimas	5	Reikalavimo tipas asm.d Vartotojo rekvizitų redagavimas, jei vartotojas – sistemos administratorius. Įmonės informacija (pavadinimas, kodas, adresas, tel. nr.) reikalinga administratorių bendravimui su ja. Juridinis asmuo – sistemos administratorius Išsaugoti duomenys turi atitikti vartotojo atliktus pakeitimus.	Įvykis/panaudojimo atvejis #	1.3
Priklausomybės Istorija	4,28	Užregistruotas 2010-03-05	Konfliktai	Nėra
Reikalavimas # Aprašymas Pagrindimas Šaltinis Tikimo kriterijus Užsakovo tenkinimas	9	Reikalavimo tipas reit. Aukščiausius reitingus turinčių vartotojų sąrašas su reitingais Norint skatinti vartotojų dalyvavimą pateikiant geras idėjas, visiems vartotojams rodomas aukščiausius reitingus turinčių vartotojų sąrašas. Idėjų banko sistema Vartotojams matoma aukščiausių reitingų savininkų lentelė.	Įvykis/panaudojimo atvejis #	2.1
Priklausomybės Istorija	3	Užregistruotas 2010-03-05	Konfliktai	Nėra
Reikalavimas # Aprašymas Pagrindimas Šaltinis Tikimo kriterijus Užsakovo tenkinimas	10	Reikalavimo tipas reit. Kiekvienoje vietoje prie vartotojo rodomas jo reitingas Norint skatinti vartotojų dalyvavimą pateikiant geras idėjas, visiems vartotojams matomas kiekvieno vartotojo reitingas. Idėjų banko sistema Kiekvienoje vietoje, kur matomas vartotojas, rodomas ir jo reitingas.	Įvykis/panaudojimo atvejis #	2.1
Priklausomybės Istorija	1	Užregistruotas 2010-03-05	Konfliktai	Nėra
Reikalavimas # Aprašymas Pagrindimas Šaltinis Tikimo kriterijus Užsakovo tenkinimas	12	Reikalavimo tipas užkl. Užklausų sąrašo peržiūrėjimas Kad vartotojai galėtų rašyti savo idėjas, jie pirma turi matyti užklausas. Taip pat vartotojams prieš rašant užklausą būtų pravartu paieškoti atsakymo tarp jau išspręstų užklausų. Idėjų banko sistema Vartotojas ar Administratorius mato užklausų sąrašą.	Įvykis/panaudojimo atvejis #	3.1
Priklausomybės Istorija	13,14,43	Užregistruotas 2009-10-01	Konfliktai	Nėra

Reikalavimas #	13	Reikalavimo tipas	užkl.	Įvykis/panaudojimo atvejis #	3.1
Aprašymas	Užklausų sąrašo filtravimas				
Pagrindimas	Sistemoje esant dideliame kiekiui užklausų, norint rasti dominančią, reikalinga filtravimo (paieškos) galimybė.				
Šaltinis	Vartotojas arba Administratorius				
Tikimo kriterijus	Pateikiamas filtro kriterijus atitinkančių užklausų sąrašas				
Užsakovo tenkinimas	3	Užsakovo netenkinimas	4		
Priklausomybės	12,14,43	Konfliktai	Nėra		
Istorija	Užregistruotas 2010-03-05				

Reikalavimas #	14	Reikalavimo tipas	užkl.	Įvykis/panaudojimo atvejis #	3.2
Aprašymas	Naujos užklausos pateikimas				
Pagrindimas	Vartotojui neradus informacijos jį dominančiu klausimu tarp sistemoje esančių užklausų, jam reikia galimybės pačiam įrašyti savo užklausą.				
Šaltinis	Vadovas				
Tikimo kriterijus	Nauja užklausa išsaugoma sistemoje aktyvioje būsenoje ir gali būti rasta sąrašė.				
Užsakovo tenkinimas	2	Užsakovo netenkinimas	5		
Priklausomybės	12,13,43	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				

Reikalavimas #	16	Reikalavimo tipas	užkl.	Įvykis/panaudojimo atvejis #	3.3
Aprašymas	Užklausos uždarymas				
Pagrindimas	Vartotojas gali rasti idėją užklausiai kitu keliu arba užklausa tapti neaktyvia, jeigu niekas neturi jai gerų idėjų. Tokiu atveju vartotojui reikia galimybės uždaryti užklausą, kad ji nebesimaišytų sąrašė.				
Šaltinis	Vadovas – užklausos savininkas				
Tikimo kriterijus	Po uždarymo užklausa įgauna uždarytos neišspręstos užklausos būseną.				
Užsakovo tenkinimas	2	Užsakovo netenkinimas	2		
Priklausomybės	14,20,21,43	Konfliktai	Nėra		
Istorija	Užregistruotas 2010-03-05				

Reikalavimas #	18	Reikalavimo tipas	užkl.	Įvykis/panaudojimo atvejis #	3.4
Aprašymas	Tinkamos užklausos idėjos parinkimas				
Pagrindimas	Vartotojams pateikus tinkamą savininkui užklausos idėją, užklausa turi būti pažymėta kaip išspręsta.				
Šaltinis	Vadovas – užklausos savininkas				
Tikimo kriterijus	Pasirinkta idėja tampa užklausos sprendimu, o užklausa uždaroma kaip išspręsta.				
Užsakovo tenkinimas	3	Užsakovo netenkinimas	5		
Priklausomybės	14,20,43	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				

Reikalavimas #	20	Reikalavimo tipas	idėja	Įvykis/panaudojimo atvejis #	4.1
Aprašymas	Idėjų pateikimas				
Pagrindimas	Vartotojai turi galėti pateikti neuždarytomis užklausoms idėjas.				
Šaltinis	Darbuotojas				
Tikimo kriterijus	Prie užklausos matoma vartotojo įvesta naujiena.				
Užsakovo tenkinimas	1	Užsakovo netenkinimas	5		
Priklausomybės	16,18,21	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				

Reikalavimas #	21	Reikalavimo tipas	reit.	Įvykis/panaudojimo atvejis #	4.2
Aprašymas	Ne savo idėjos reitingavimas				
Pagrindimas	Vartotojai turi galėti balsuoti už arba prieš kiekvieną ne savo idėją, kad padėtų užklaustos savininkui išrinkti tinkamą idėją, leistų uždaryti užklausa be gerų idėjų bei keistų bendrą idėjos autoriaus reitingą, taip mažinant prastų idėjų kiekį sistemoje.				
Šaltinis	Vartotojas				
Tikimo kriterijus	Atitinkamai perskaičiuoti idėjos ir jos autoriaus reitingai.				
Užsakovo tenkinimas	3	Užsakovo netenkinimas	4		
Priklausomybės	Visi reit. grupės reitingai, 16,20	Konfliktai	Nėra		
Istorija	Užregistruotas 2010-03-05				

Reikalavimas #	22	Reikalavimo tipas	kom.	Įvykis/panaudojimo atvejis #	4.3
Aprašymas	Idėjų komentavimas				
Pagrindimas	Vartotojams reikėtų galimybės plačiau išreikšti savo nuomonę apie idėją ar ją praplėsti.				
Šaltinis	Vartotojas				
Tikimo kriterijus	Prie idėjos matomas naujas komentaras.				
Užsakovo tenkinimas	2	Užsakovo netenkinimas	1		
Priklausomybės	20,42	Konfliktai	Nėra		
Istorija	Užregistruotas 2010-03-05				

Reikalavimas #	28	Reikalavimo tipas	asm.d	Įvykis/panaudojimo atvejis #	5.1
Aprašymas	Vartotojų administravimas				
Pagrindimas	Vartotojai gali nurodyti neteisingą ar net kitus žeidžiančią informaciją savo profilyje, todėl administratoriui gali tekti tokią informaciją pataisyti. Taip pat gali prireikti patikrinti įmonės nurodytus rekvizitus.				
Šaltinis	Administratorius				
Tikimo kriterijus	Administratorius mato bei gali redaguoti vartotojų duomenis.				
Užsakovo tenkinimas	1	Užsakovo netenkinimas	1		
Priklausomybės	4,5	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01 Papildytas įmonės rekvizitų redagavimu 2010-03-05				

Reikalavimas #	30	Reikalavimo tipas	išor.	Įvykis/panaudojimo atvejis #	6.1
Aprašymas	Išorinės sistemos susiejimas su vartotojo profiliu				
Pagrindimas	Norint pateikti išorinei sistemai vartotojo duomenis, reikia kad idėjų banko sistema žinotų, kurio vartotojo duomenis pateikti.				
Šaltinis	Darbuotojas				
Tikimo kriterijus	Po susiejimo vartotojas savo profilyje mato susietą sistemą. Tai sistemai paprašius duomenų, idėjų banko sistema juos pateikia.				
Užsakovo tenkinimas	4	Užsakovo netenkinimas	2		
Priklausomybės	31,44	Konfliktai	Nėra		
Istorija	Užregistruotas 2009-10-01				

Reikalavimas #	31	Reikalavimo tipas	išor.	Įvykis/panaudojimo atvejis #	6.2
Aprašymas	Šąsajos su išorine sistema panaikinimas				
Pagrindimas	Duomenų teikimas išorinėms sistemoms bus neteisėtas, jeigu vartotojas negalės to teikimo nutraukti.				
Šaltinis	Darbuotojas				
Tikimo kriterijus	Po šąsajos panaikinimo, išorinei sistemai paprašius duomenų, jie nėra pateikiami.				

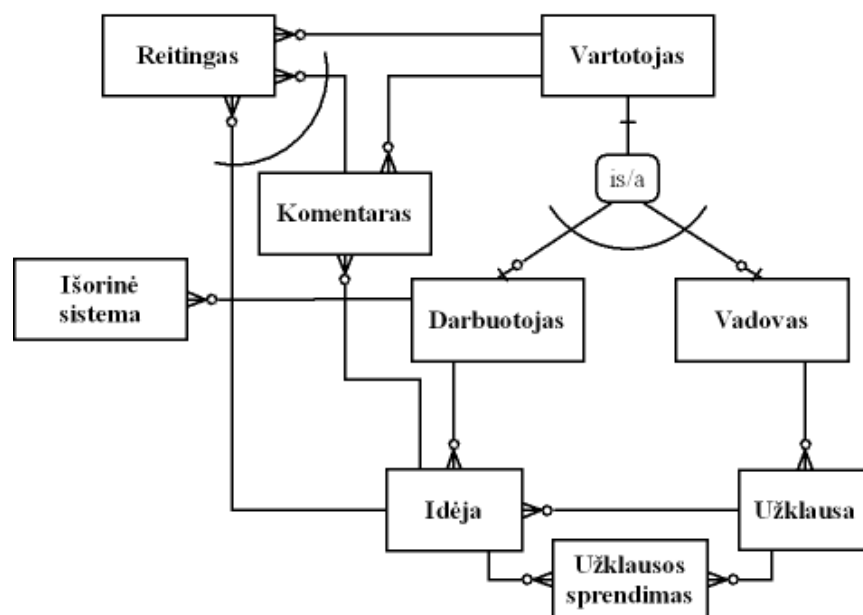
Užsakovo tenkinimas	1	Užsakovo netenkinimas	3
Priklausomybės	30	Konfliktai	Nėra
Istorija	Užregistruotas 2009-10-01		
Reikalavimas #	40	Reikalavimo tipas	asm.d Įvykis/panaudojimo atvejis # 1.5
Aprašymas	Darbuotojų registravimas.		
Pagrindimas	Įmonės saugumo sumetimais nauji įmonės darbuotojai sistemoje gali būti registruojami tik jau užregistruotų įmonės vadovų.		
Šaltinis	Įmonės vadovas		
Tikimo kriterijus	Vartotojas išsaugomas sistemoje, jam el. paštu nusiunčiamas pakvietimas prisijungti ir užpildyti likusius duomenis.		
Užsakovo tenkinimas	3	Užsakovo netenkinimas	4
Priklausomybės	1,41,43	Konfliktai	Nėra
Istorija	Užregistruotas 2010-11-15		
Reikalavimas #	41	Reikalavimo tipas	asm.d Įvykis/panaudojimo atvejis # 1.6
Aprašymas	Vadovų registravimas.		
Pagrindimas	Įmonės saugumo sumetimais nauji įmonės vadovai sistemoje gali būti registruojami tik jau užregistruoto įmonės sisteminio administratoriaus.		
Šaltinis	Įmonės sisteminis administratorius		
Tikimo kriterijus	Vadovas išsaugomas sistemoje, jam el. paštu nusiunčiamas pakvietimas prisijungti ir užpildyti likusius duomenis.		
Užsakovo tenkinimas	3	Užsakovo netenkinimas	4
Priklausomybės	1,40,43	Konfliktai	Nėra
Istorija	Užregistruotas 2010-11-15		
Reikalavimas #	42	Reikalavimo tipas	reit. Įvykis/panaudojimo atvejis # 4.4
Aprašymas	Ne savo komentaro reitingavimas		
Pagrindimas	Vartotojai turi galėti balsuoti už arba prieš kiekvieną ne savo komentarą, kad paryškintų gerus ir paslėptų prastus komentarus.		
Šaltinis	Vartotojas		
Tikimo kriterijus	Atitinkamai perskaičiuoti komentaro ir jo autoriaus reitingai.		
Užsakovo tenkinimas	3	Užsakovo netenkinimas	3
Priklausomybės	Visi reit. grupės reitingai,22	Konfliktai	Nėra
Istorija	Užregistruotas 2010-11-15		
Reikalavimas #	43	Reikalavimo tipas	asm.d Įvykis/panaudojimo atvejis # 5.2
Aprašymas	Vartotojų teisių valdymas.		
Pagrindimas	Įmonės darbuotojai gali keisti pareigas, todėl sistemoje reikalinga galimybė pakeisti vartotojo atliekamą vaidmenį.		
Šaltinis	Įmonės sisteminis administratorius		
Tikimo kriterijus	Įmonės darbuotojas įgauna ar praranda keistas teises, t.y. pasikeičia jam matomų ir leidžiamų funkcijų aibė.		
Užsakovo tenkinimas	2	Užsakovo netenkinimas	3
Priklausomybės	1,40,41	Konfliktai	Nėra
Istorija	Užregistruotas 2010-11-15		
Reikalavimas #	43	Reikalavimo tipas	užkl. Įvykis/panaudojimo atvejis # 5.3
Aprašymas	Užklausų valdymas		
Pagrindimas	Gali atsirasti poreikis pagedaguoti užklausą taip, kaip paprastaiems vartotojams nėra leidžiama (pvz. pakeisti vadovą). Tokiu atveju tai turi atlikti administratorius.		
Šaltinis	Administratorius		
Tikimo kriterijus	Užklausa pasikeičia pagal atliktus pakeitimus.		

Užsakovo tenkinimas	2	Užsakovo netenkinimas	4
Priklausomybės	12,13,14,16,18	Konfliktai	Nėra
Istorija	Užregistruotas 2010-11-15		

Reikalavimas #	44	Reikalavimo tipas	išor.	Įvykis/panaudojimo atvejis #	6.3
Aprašymas	Išorinės sistemos informacijos paėmimas				
Pagrindimas	Norint palengvinti darbuotojų informacijos įvedimą, dalis informacijos gali būti paimama iš išorinių sistemų.				
Šaltinis	Sistema				
Tikimo kriterijus	Trūkstama informacija perkelta iš išorinės sistemos.				
Užsakovo tenkinimas	4	Užsakovo netenkinimas	2		
Priklausomybės	30	Konfliktai	Nėra		
Istorija	Užregistruotas 2010-11-15				

4.4.2.4 Reikalavimai duomenims

Reikalavimai duomenims pateikiami kaip esybių-ryšių diagrama (13 pav.)



13 pav. Reikalavimai duomenims

4.4.3 Egzistuojantys komponentai, kurie gali būti panaudoti

Sistemoje numatoma naudoti nemokamus viešai prieinamus programinius komponentus programavimui palengvinti:

- **jQuery** – javascript paketas, palengvinantis dinaminio turinio pateikimą vartotojams;
- **xajax** – PHP paketas, palengvinantis asinchroninį turinio pateikimą vartotojams;
- **Smarty** – šablonų varikliukas, skirtas atskirti svetainės logiką nuo dizaino (html šablonų).
- **OpenSocial PHP Client** – didelė biblioteka, skirta darbo su OpenSocial API palengvinimui

- Taip pat gali būti panaudotas užsakovo turimas internetinių puslapių branduolys.

4.5 Architektūros specifikacija

Pagrindiniai reikalavimai sistemos architektūrai:

- Sistema turi būti sudaryta pagal MVC (Model-View-Controller) šabloną – taip tikimasi palengvinti potencialių naujų darbuotojų įsitraukimą į darbą;
- Sistema turi galėti bendrauti su išorinėmis sistemomis, tokiomis kaip Facebook;
- Turi būti paruošti metodai papildyti sistemos palaikomų išorinių sistemų sąrašą;
- Sistema turėtų veikti viename Linux serveryje, su galimybe perkelti duomenų bazės funkcijas į atskirą serverį.

MVC šablonas naudojamas todėl, kad tai vienas labiausiai visur minimų šablonų. Jis atskiria vartotojo sąsajos elementus į tris atskiras dalis (iš (Fowler, 2003)):

- Modelis – tai objektas, atstovaujantis kokią nors informaciją apie sritį. Tai nevizualinis objektas, turintis savyje visus duomenis ir elgesį, išskyrus tai, kas priklauso vartotojo sąsajai;
- Šablonas (angl. *view*) atspindi modelio atvaizdavimą vartotojo sąsajoje. [...] Šablonas skirtas tik informacijos pateikimui; informacijos pasikeitimai yra apdorojami trečio MVC trijulės nario – kontrolierio (valdiklio);
- Kontroleris paima vartotojo įvestis, manipuliuoja modeliu ir sukelia atitinkamus šablono pasikeitimus.

4.5.1 Panaudojimo atvejų scenarijai

Čia nagrinėjami susiję su išorinėmis sistemomis panaudojimo atvejų scenarijai.

7.1 – Susieti išorinę sistemą su profiliu

Pagrindinis scenarijus:

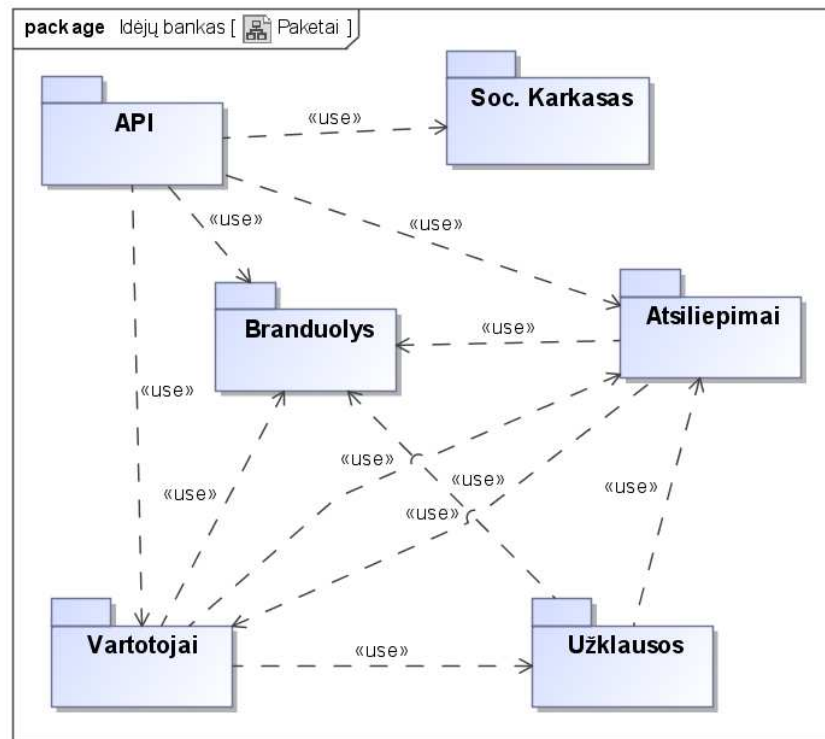
1. Idėjų pasiūlos grupės vartotojas iš meniu pasirenka naujos išorinės sistemos prijungimą
2. Pasirinkęs, kokią išorinę sistemą prijungs, jis patenka į prijungimo formą
3. Priklausomai nuo sistemos, Idėjų banko sistema automatiškai prisijungia prie išorinės sistemos arba duoda vartotojui tolesnes prijungimo instrukcijas
4. Įsitikinęs duomenų teisingumu susietoji sistema yra išsaugoma Idėjų banko sistemoje prie to vartotojo.

7.3 – Gauti susieto vartotojo reitingus

Pagrindinis scenarijus:

1. Išorinėje sistemoje vartotojui reikia parodyti Idėjų banko teikiamą informaciją
2. Išorinė sistema kreipiasi į Idėjų banko sistemą su Idėjų banko vartotoją identifikuojančia informacija
3. Idėjų banko sistema perduoda išorinei sistemai suformuotus reikiamo vartotojo reitingus.

4.5.2 Sistemos statinis vaizdas



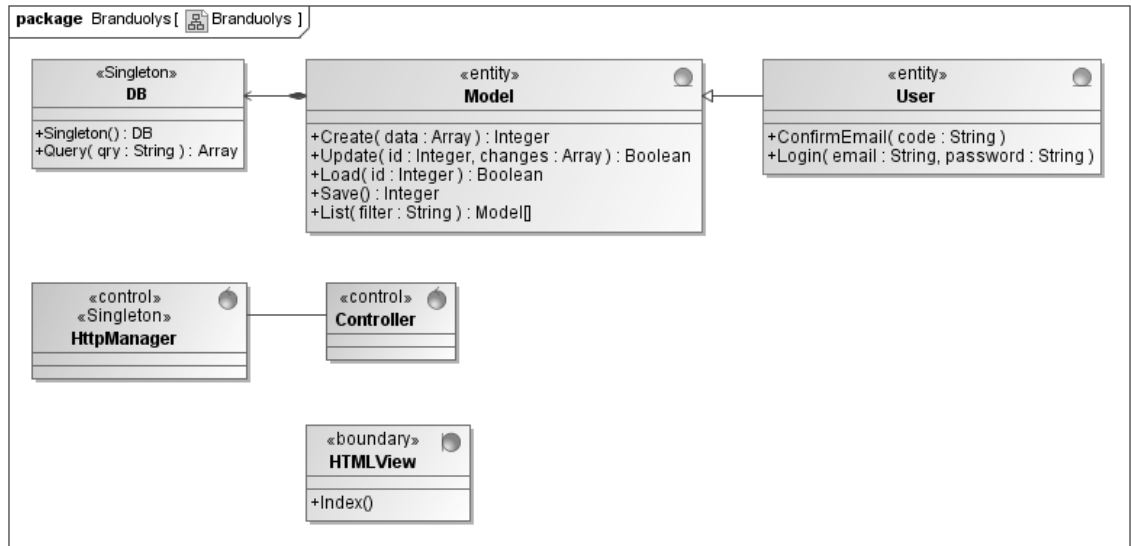
14 pav. Sistemos paketų vaizdas

Sistema suskaidyta į branduolį ir 5 funkcinius paketus:

1. Branduolio užduotis – suteikti funkcijų bazę, reikalingą kitų paketų darbui;
2. Vartotojų paketo užduotis – dirbti su sistemos vartotojų informacija, t.y. su jų registracija, informacijos peržiūra, el. pašto patvirtinimu ir t.t.;
3. Užklausų paketas yra skirtas su pagrindine sistemos informacija – užklausomis bei jų kategorijomis ir idėjomis;
4. API paketas – tai Idėjų banko sistemos dalis, skirta darbui su Soc. karkaso paketu. Pagrindinė priežastis, kodėl naudojamas toks papildomas paketas, yra tai, kad taip palengvinamas reikiamų pakeitimų įvedimas;
5. Soc. karkaso paketo užduotis – palengvinti darbą su išorinėmis sistemomis. Šis paketas padeda programuotojui dirbti su skirtingomis išorinėmis sistemomis.

4.5.2.1 Branduolys

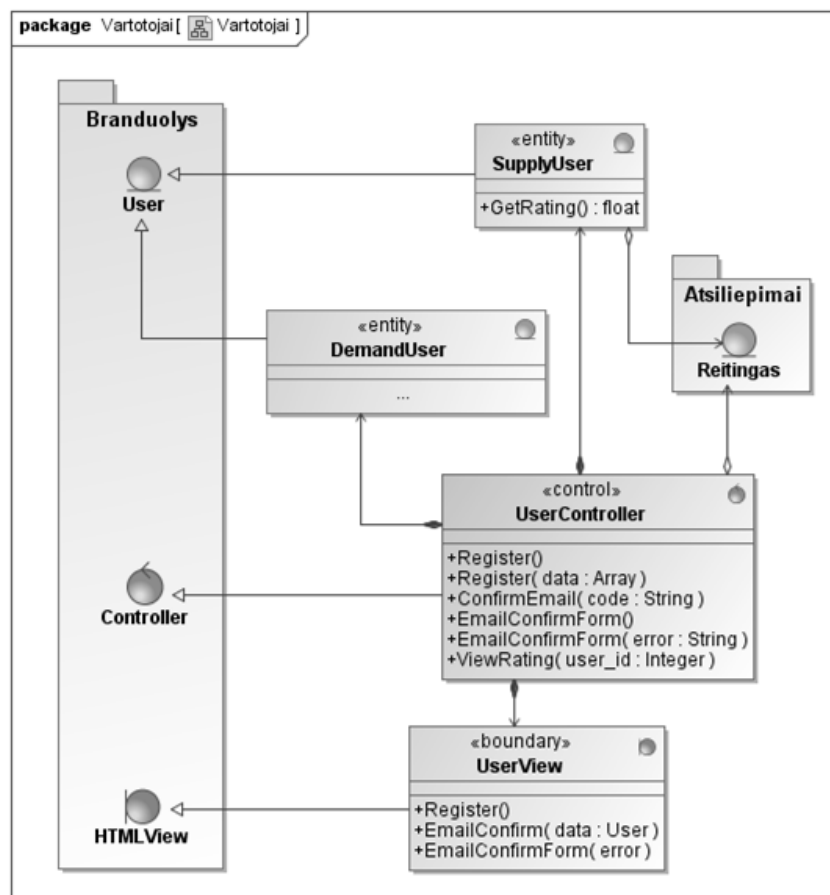
Šiam paketui priklauso bazinės klasės, kurios reikalingos sistemos veikimui bei yra paveldimos kitų paketų klasių.



15 pav. Branduolio paketo klasių diagrama

4.5.2.2 Vartotojai

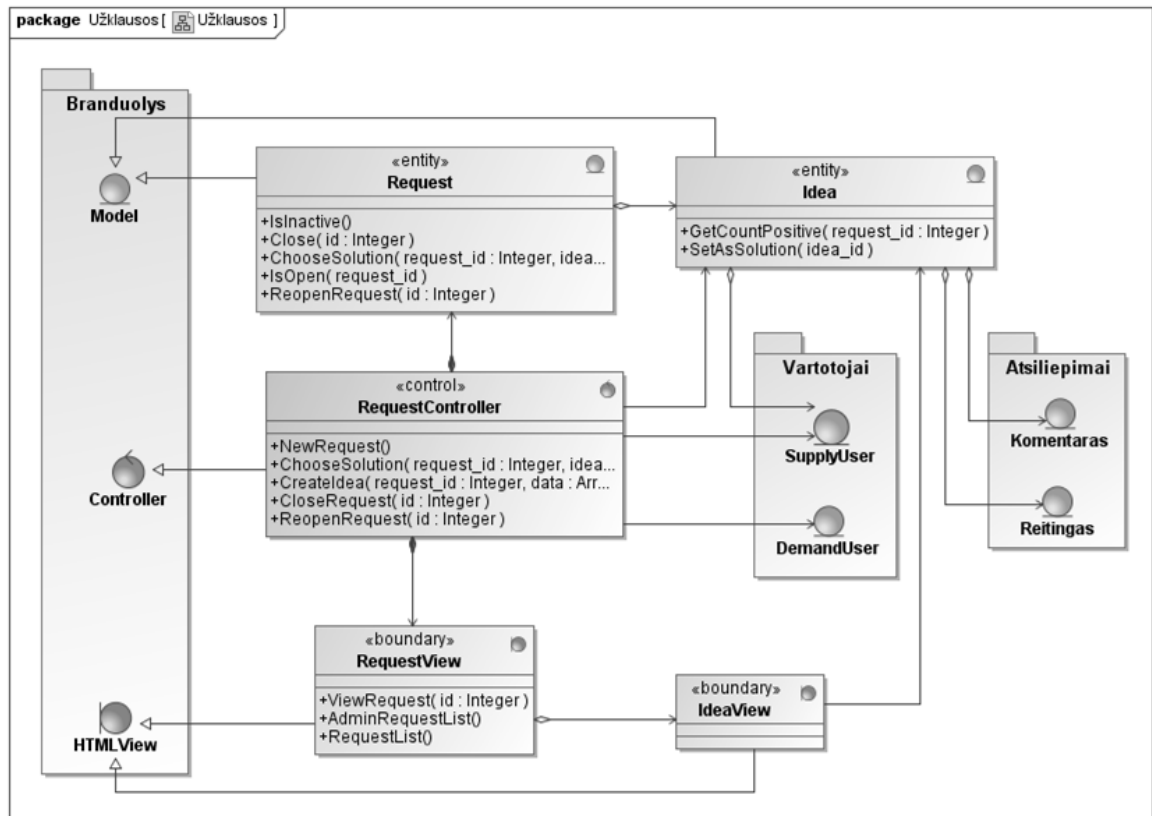
Šio paketo užduotis – dirbti su sistemos vartotojais.



16 pav. Vartotojų paketo klasių diagrama

4.5.2.3 Užklauso

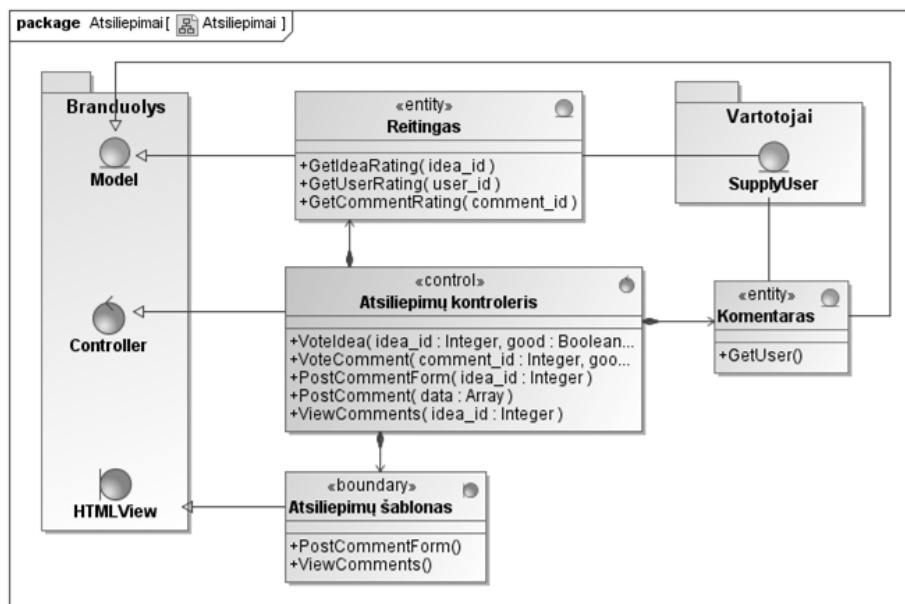
Paketas skirtas darbui su pagrindine sistemos informacija – užklausomis bei jų kategorijomis ir idėjomis.



17 pav. Užklauso paketo klasių diagrama

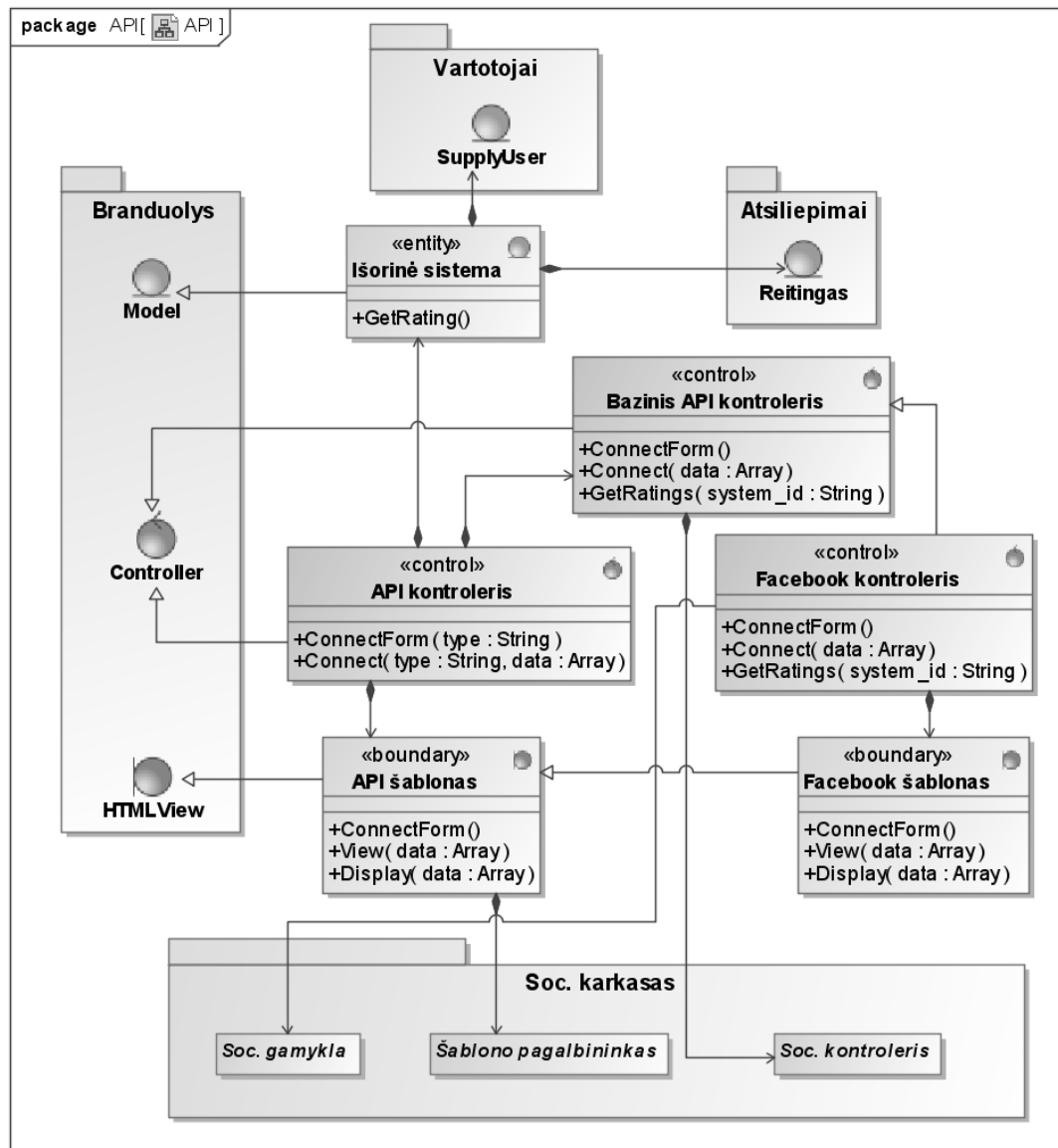
4.5.2.4 Atsiliepimai

Atsiliepimų modulis skirtas darbui su idėjoms vartotojų priskirtais reitingais bei komentarais.



18 pav. Atsiliepimų paketo klasių diagrama

4.5.2.5 Idėjų banko API paketas



19 pav. API paketo klasių diagrama

APIController

Klasifikacija	Klasė
Paveldi	Controller
Apibrėžimas	Kontroleris, skirtas bendravimo su išorinėmis sistemomis nustatymams redaguoti
Atsakomybės	Išorinių sistemų sąsajos su Idėjų banko vartotoju nustatymų redagavimas. Taip pat išorinės sistemos tipo nustatymas iš jos užklauso (radus vartotojo sukurtą sąsajos aprašą - ExternalSystems DB lentelės įrašą) ir valdymo perdavimas atitinkamam kontrolieriui

AbstractApiController

Klasifikacija	Klasė
Paveldi	Controller
Apibrėžimas	Abstraktus kontrolieris, skirtas bendrauti su išorine sistema. Turėtų būti paveldimas konkrečios išorinės sistemos
Atsakomybės	Išorinių sistemų sąsajos su Idėjų banko vartotoju pagrindinių metodų aprašas ir bazinis funkcionalumas

FacebookApiController

Klasifikacija	Klasė
Paveldi	AbstractApiController
Apibrėžimas	Facebook kontrolieris, skirtas bendrauti su išorine Facebook sistema, perduodant duomenis Soc. karkaso klasei
Atsakomybės	Facebook sistemos sąsajos su Idėjų banku kontrolieris. Pritaiko abstraktų funkcionalumą Facebook sistemos užklausoms ir perduoda informaciją Soc. Karkasui

ExternalSystem

Klasifikacija	Klasė
Paveldi	Model
Apibrėžimas	Dirba su vartotojų sąsajų su išorinėmis sistemomis nustatymais bei agreguoja duomenis, paruošdama juos teikti išorinei sistemai
Atsakomybės	Vartotojų sąsajos nustatymų duomenų bazėje saugojimas ir redagavimas. Vartotojų reitingų informacijos agregavimas teikimui išorinei sistemai
Resursai	Tiesiogiai dirba su ExternalSystems duomenų bazės lentele

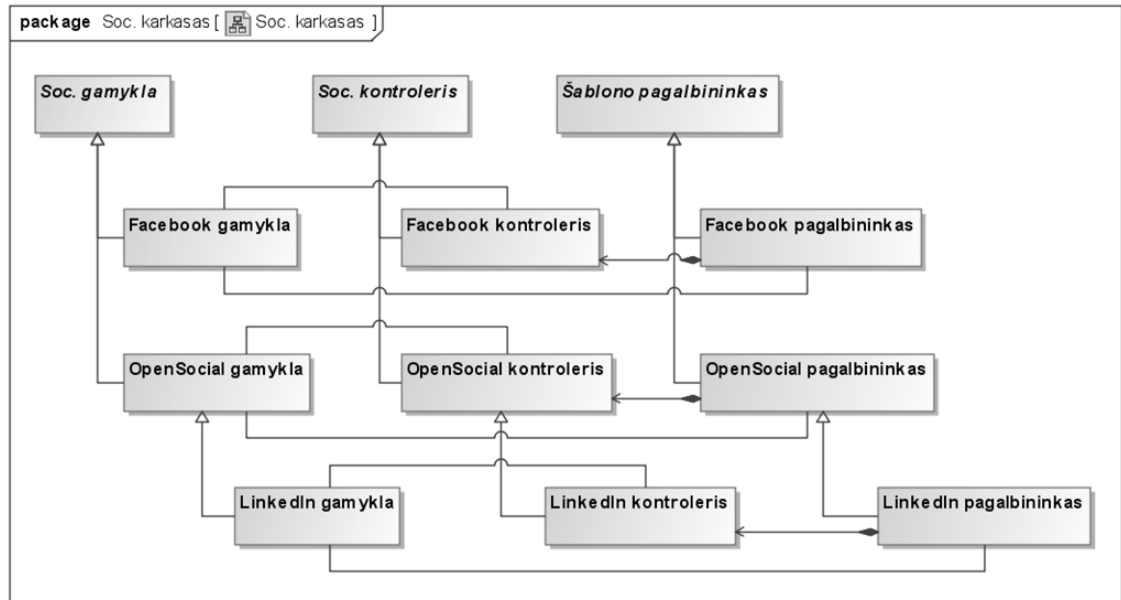
APIView

Klasifikacija	Klasė
Paveldi	HTMLView
Apibrėžimas	Abstraktus API šablonas, paveldimas konkrečių išorinių sistemų darbui pritaikytų šablonų
Atsakomybės	Išorinių sistemų sąsajos su Idėjų banko vartotoju pagrindinių šablonų aprašas

FacebookAPIView

Klasifikacija	Klasė
Paveldi	APIView
Apibrėžimas	Facebook šablonas, skirtas pateikti duomenis Facebook išorinei sistemai priimtina forma. Naudoja Šablono pagalbininko klasę iš Soc. karkaso paketo.
Atsakomybės	Kontrolierio perduotus duomenis paruošti pateikimui Facebook išorinei sistemai

4.5.2.6 Soc. karkaso paketas



20 pav. Soc. karkaso paketo klasių diagrama

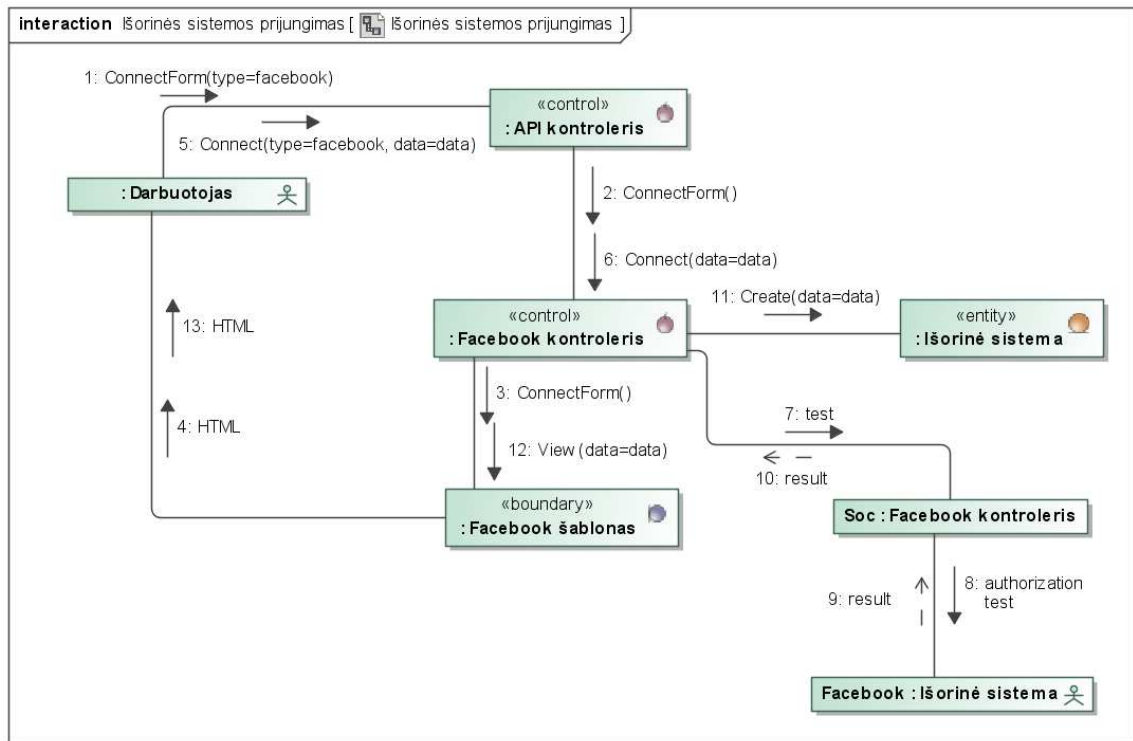
Klasių apibrėžimas

Yra trys pagrindiniai socialinio karkaso paketo iš išorės prieinamų klasių tipai:

- Kontroleriai (valdikliai) – priima užklausas ir jas apdoroja. Taip pat paima informaciją iš išorinės sistemos bei ją nuskaito į patogią formą;
- Pagalbininkai (angl. *helpers*) – naudojami šablonuose – suteikia priėjimą prie duomenų bei saugo reikiamus komunikavimo su išorine sistema per vartotoją duomenis (pvz. kelius iki Facebook Grafo API reikiamų taškų);
- Gamyklos (angl. *factories*) – padeda sukurti reikiamus kontrolierio ir pagalbininko objektus.

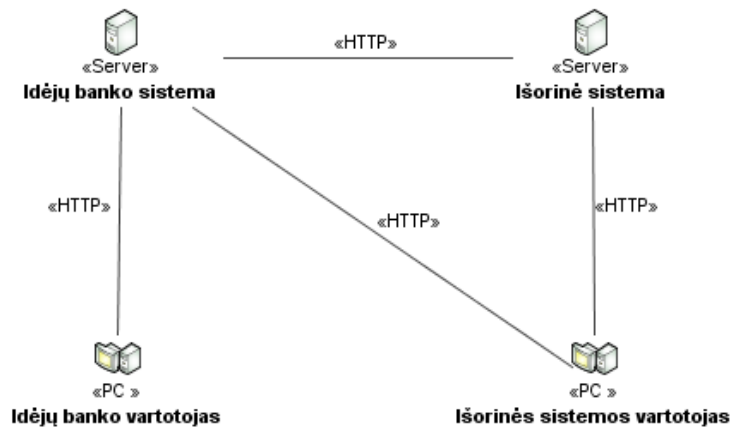
4.5.3 Sistemos dinaminis vaizdas

21 pav. pavaizduotas vienas iš svarbiausių Idėjų banko ir išorinės sistemos bendradarbiavimo atvejų – jo metu susiejami šių dviejų sistemų vartotojai.



21 pav. Išorinės sistemos prijungimas iš Idėjų banko pusės

4.5.4 Sistemos komponentų fizinis išdėstymas



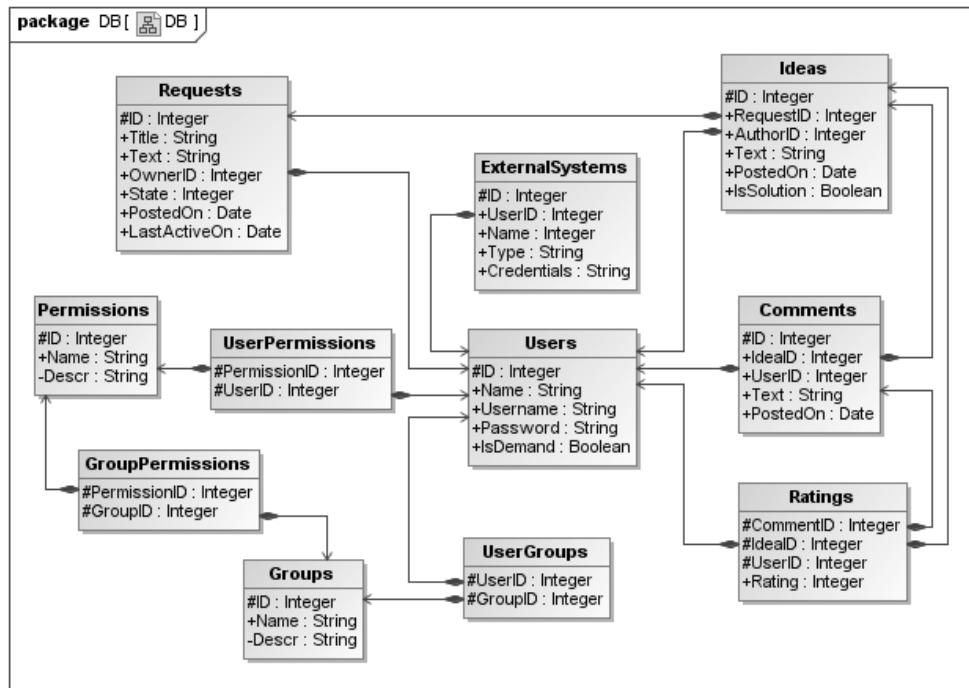
22 pav. Išdėstymo (angl. deployment) vaizdas

Pagrindinė Idėjų banko sistemos dalis (t.y. kodas, failai ir duomenų bazė) bus laikoma viename serveryje. Sistemos sugeneruoti HTML, JavaScript ir CSS kodai bus pateikiami vartotojo kompiuteryje esančiai naršyklei. Numatoma, kad serveris su išorinės sistemos serveriu bei tiesiogiai su jos vartotojais bendraus HTTP protokolu.

Paprastai kai vartotojas yra prisijungęs prie išorinės sistemos ir atsidaręs Idėjų banko aplikaciją, užklauskos Idėjų banko sistemai ateis iš išorinės sistemos, tačiau ne tiesiogiai, o per vartotojo naršyklę. Dėl šios priežasties ateinantys duomenys turėtų būti užkoduoti, kad nebūtų galima apgauti Idėjų banko sistemos jai pateikiant sufabrikuotus duomenis.

Vartotojui dirbant tiesiogiai su Idėjų banko sistema, gali prireikti, kad jo atliekami veiksmai tiesiogiai atsispindėtų jo išorinės sistemos puslapyje (jeigu jis pavyzdžiui pateikė naują idėją ir yra pažymėjęs, kad nori, kad jos automatiškai atsidurtų jo Facebook naujienose). Tokiu atveju Idėjų banko sistema su išorine sistema susisiečia tiesiogiai.

4.5.5 Duomenų bazės struktūra



23 pav. Duomenų bazės vaizdas

Išorinių sistemų lentelėje (ExternalSystems) saugoma sąryšio su išorinių sistemų vartotojais informacija:

- Vartotojo ID (UserID) – Idėjų banko vartotojo, su kuriuo susietas šis išorinės sistemos vartotojas, identifikacijos numeris;
- Vardas (name) – saugo išorinės sistemos vartotojo vardą;
- Tipas (Type) – nurodo, kokia tai išorinė sistema;
- Sąryšio duomenys (Credentials) – saugo informaciją, reikalingą ryšiui su išorine sistema sėkmingai sudaryti. Šio lauko turinys priklauso nuo išorinės sistemos bei Soc. karkaso kontrolerio klasės. Paprastai šiame lauke saugomas tekstu paverstas asociatyvus masyvas.

5 KOKYBĖS TYRIMAS

Pagrindinis kokybės tyrimo objektas – socialinių tinklų aplikacijų programavimo sąsajų karkasas. Svarbiausi požymiai, kuriais turėtų pasižymėti šis karkasas:

- Jo naudojimas turi palengvinti projektuotojų bei programuotojų darbą su išorinių sistemų aplikacijų programavimo sąsajomis;
- Karkasas turėtų būti nesunkiai ir aiškiai papildomas naujomis socialinių tinklų aplikacijų programavimo sąsajomis.

5.1 Darbo palengvinimas

Sukūrus Idėjų banko sistemą buvo sukurtos funkcijos, skirtos gauti reikiamą informaciją pateikimui išorinėms sistemoms:

- Įvertintų idėjų sąrašo;
- Vartotojo idėjų ir jų vertinimų sąrašo;
- Vartotojo užklausų sąrašo.

Kitame etape buvo sukurtas Soc. karkaso Facebook funkcionalumas. Šios Soc. karkaso versijos funkcionalumas apėmė:

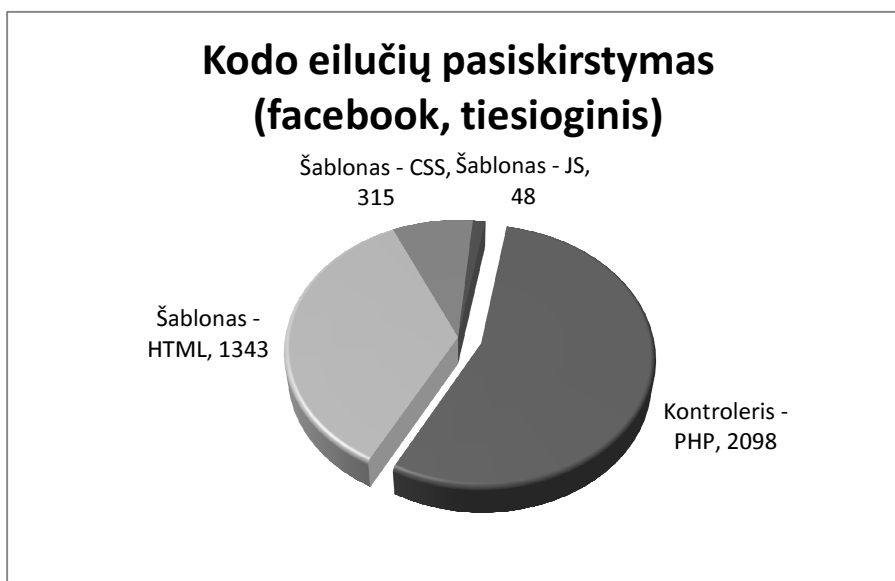
- Automatinį aplikacijos įkėlimą į išorinę sistemą (autorizacijos proceso automatizavimas bei gautų raktų pateikimas karkasą naudojančiai sistemai saugojimui);
- Automatinį autorizacijos proceso pakartojimą nustojus galioti gautiems raktams;
- Automatinį pranešimo apie aplikacijos pašalinimo iš išorinės sistemos apdorojimą – išsaugotų raktų pašalinimą iš raktų sąrašo;
- Prisijungusio prie išorinės sistemos vartotojo atrinkimą iš Facebook su kiekviena užklausa siunčiamos koduotos informacijos;
- Facebook grafo API prieigą
 - Įrašų skaitymo, nuskaitant atsaką į PHP masyvą patogiam priėjimui;
 - Įrašų publikavimo, perduodant informaciją PHP masyve.

Tai reiškia, kad trečiame etape sujungiant Idėjų banko ir Soc. karkaso funkcionalumą nereikėjo nuo nulio programuoti šių funkcijų – teliko tik jas panaudoti. Šiam sujungimui skirto kodo eilučių kiekio pasiskirstymas atvaizduotas 24 pav. Iš parašyto kodo proporcijų, priklausomai nuo programavimo kalbos, matome, kad daugiausiai darbo tenka įdėti programuojant socialiniame tinklapyje pateikiamo funkcionalumo išvaizdą, t.y. HTML bei CSS kodą. Žinoma, tai priklauso nuo reikiamo dizaino sudėtingumo.



24 pav. Idėjų banko ir Soc. karkaso sujungimo Facebook funkcionalumui kodo eilučių kiekio pasiskirstymas

Kad galima būtų įvertinti Soc. karkaso teikiamą naudą programuojant, tas pats funkcionalumas buvo įgyvendintas tiesiogiai, nenaudojant karkaso. Šiuo atveju kodas, anksčiau priklausęs Soc. karkasui, perėjo į kontrolerį (25 pav.) Taip pat, kadangi sąsajos su išorine sistema kodas nebeturi būti toks universalus, jo rezultatus paprasčiau naudoti iš HTML šablono. Tai paaiškina šiek tiek sumažėjusį HTML šablono kodo eilučių skaičių.



25 pav. Idėjų banko Facebook funkcionalumo nenaudojant Soc. karkaso kodo eilučių kiekio pasiskirstymas

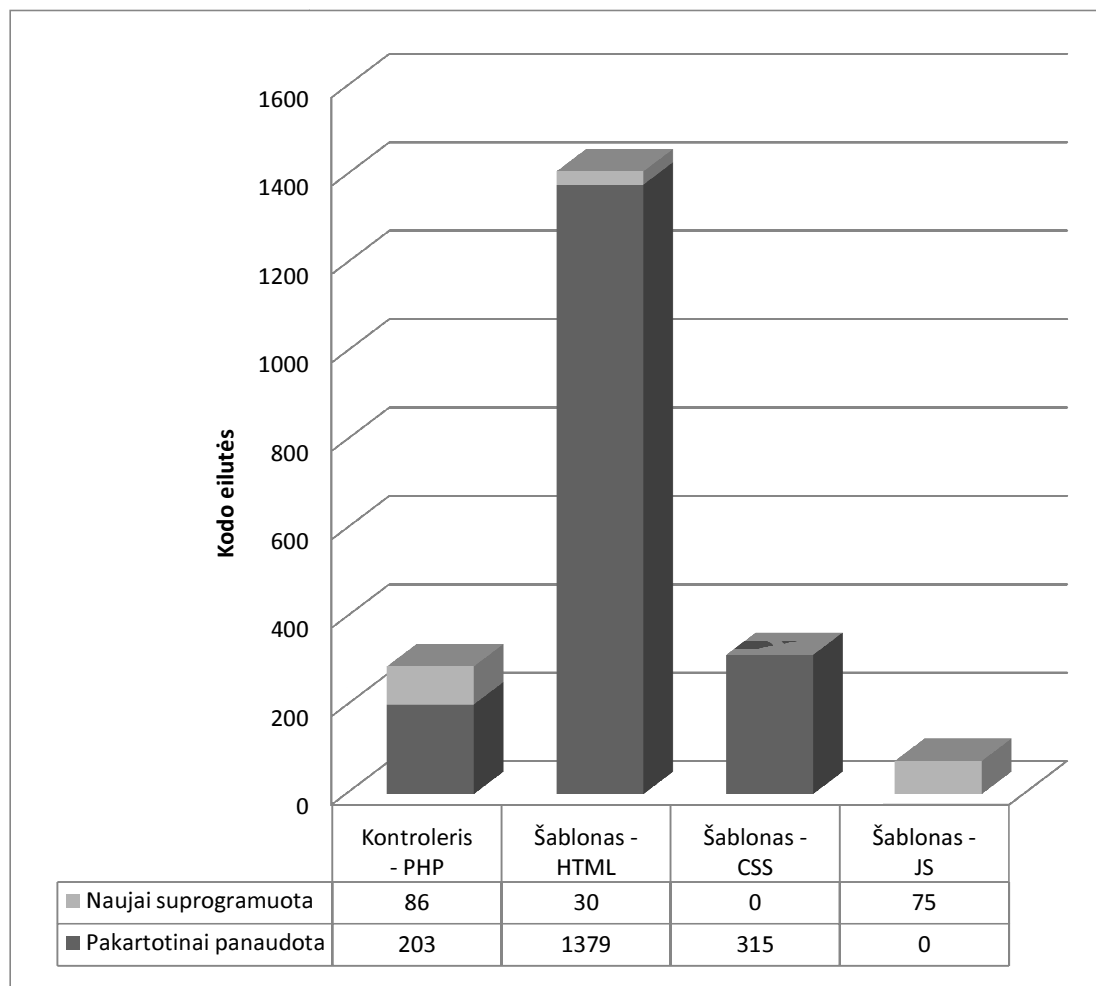
5.2 Karkaso papildymas naujo socialinio tinklo palaikymu

Šioje kokybės tyrimo dalyje prie Soc. karkaso pridėsime OpenSocial palaikymą. OpenSocial sritis apima daugelį skirtingų socialinių tinklų ir net kitokių puslapių. Vienas iš tokių „ne socialinių tinklų“ puslapių yra iGoogle – vieta, kur žmonės savo Google vartotojui gali priskirti įvairias aplikacijas (angl. *gadgets*) ir naudoti šį puslapį kaip savo namų puslapį.

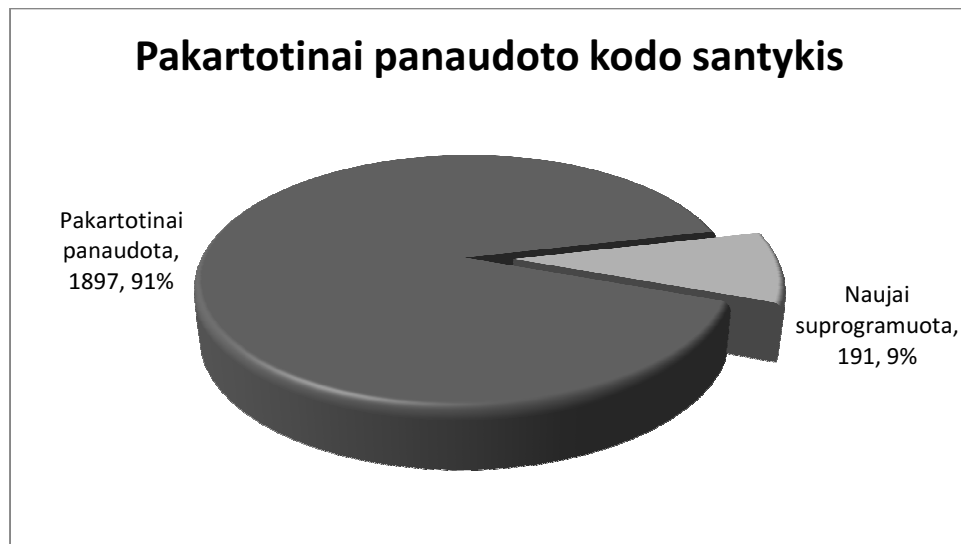
Pagal (Meoa, et al., 2010), vartotojai dažnai ištraukia į daugelį socialinių tinklų ir susikuria savo paskyrą kiekviename iš jų. Kadangi iGoogle nėra socialinis tinklapis, jam trūksta ir kai kurių funkcijų, kurias socialiniuose tinklalapiuose tikimės rasti. Pavyzdžiui, per iGoogle nėra galimybės ką nors publikuoti savo vartotojo puslapyje (paprasčiausiai dėl to, kad toks puslapis neegzistuoja). Dėl šios priežasties iGoogle funkcionalumas Soc. karkase bus ribojamas. Soc. karkaso funkcijos, skirtos informacijos publikavimui socialiniuose tinkluose nieko nedaro, o skaitymo funkcijos visuomet grąžina tuščią įrašą (tačiau ir nemeta klaidos).

Bendravimui su OpenSocial tinklapiais bus naudojama viešai prieinama nemokama biblioteka „OpenSocial PHP Client Library“. Šios bibliotekos funkcijas kvies specialiai tam parašytos Soc. karkaso OpenSocial kontrolerio klasės. Jos reikalingos tam, kad suvienodinti funkcijų aprašus su Facebook sistemos kontroleriu.

Prie Idėjų banko sistemos prijungiant naują Soc. karkaso funkcionalumą gali būti pakartotinai panaudotas seniau Facebook sistemos prijungimui parašytas kodas (26 pav.) Naujai parašyto ir pakartotinai panaudoto kodo santykis pateiktas 27 pav.



26 pav. Idėjų banko ir Soc. karkaso sujungimo OpenSocial funkcionalumui kodo eilučių kiekio pasiskirstymas



27 pav. Idėjų banko ir Soc. karkaso sujungimo OpenSocial funkcionalumui pakartotinis panaudojimas

6 PATOBULINIMAI

6.1 Asinchroninis informacijos įkėlimas

Sistemos testavimo metu pastebėta, kad sinchroninis kreipimasis į Facebook API kiekvieną kartą vartotojui įvedus naują idėją sukelia pastebimų sulėtėjimų. Veikimo greitis čia labiausiai priklauso nuo ryšio tarp Idėjų banko sistemos bei išorinės sistemos, o jo keisti beveik neįmanoma. Dėl šios priežasties nuspręsta įvesti galimybę duomenis išorinei sistemai siųsti asinchroniškai, kadangi daugeliu atvejų nebūtina žinoti, ar duomenys buvo perduoti sėkmingai. Tai įgyvendinta prie duomenų įkėlimo į išorinę sistemą pridedant papildomą požymį, o funkcijos užduotį per komandinę eilutę perduodant vykdyti atskiram PHP interpretatoriui.

6.2 Pakartotinis išorinės sistemos leidimų paprašymas

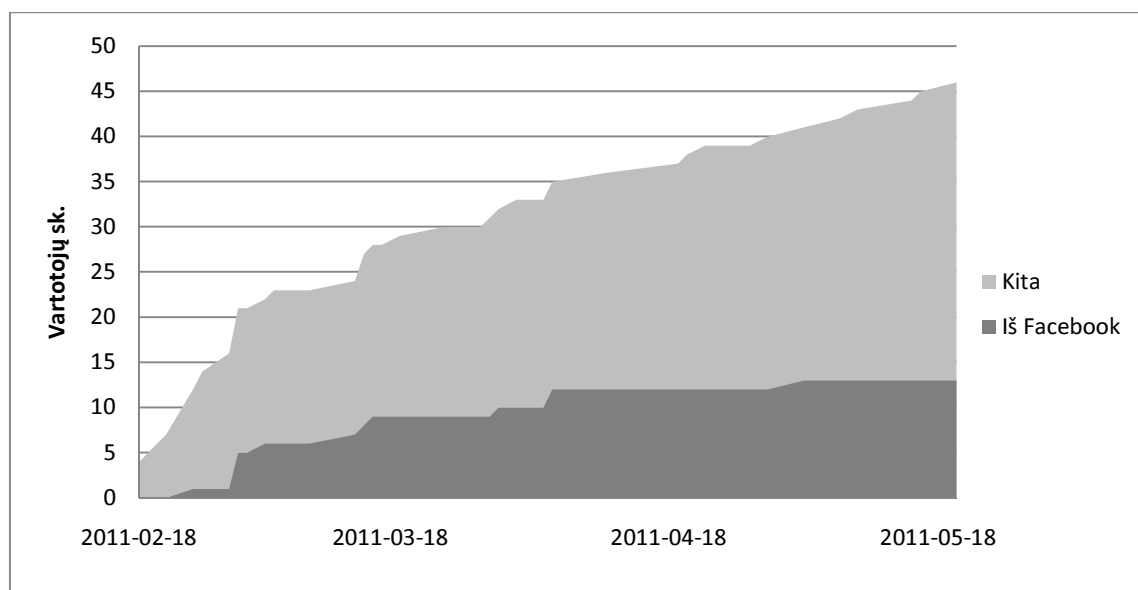
Pasikeitus sistemos reikalavimams, turi pasikeisti ir leidimai, kurių prašoma iš facebook vartotojo. Tuo tikslu prie kiekvieno išorinės sistemos vartotojo sąryšio informacijos, saugomos duomenų bazėje, pridedamas sąrašas leidimų, kurie buvo suteikti šio vartotojo mūsų aplikacijai. Idėjų banko Facebook kontrolerio kode įrašytas reikiamų leidimų sąrašas nusako dabartinės versijos teisingam veikimui reikalingus leidimus. Tada vartotojui užėjus į savo idėjų banko aplikaciją Facebook sistemoje, yra patikrinama, ar priskirti leidimai pilnai dengia reikiamus ir jeigu ne, vartotojo automatiškai paprašoma papildomų leidimų.

6.3 Idėjos vertinimo ir „man patinka“ susiejimas

Vartotojui pateikus idėją, ji gali automatiškai atsirasti jo Facebook profilyje. Tokiu atveju, kam nors tą idėją gerai įvertinus Idėjų banko sistemoje, Facebook profilyje prie tos idėjos galėtų automatiškai atsirasti ją įvertinusio asmens „man patinka“ (angl. *like*). Patobulinimas paliktas įgyvendinti vėlesnėse projekto stadijose.

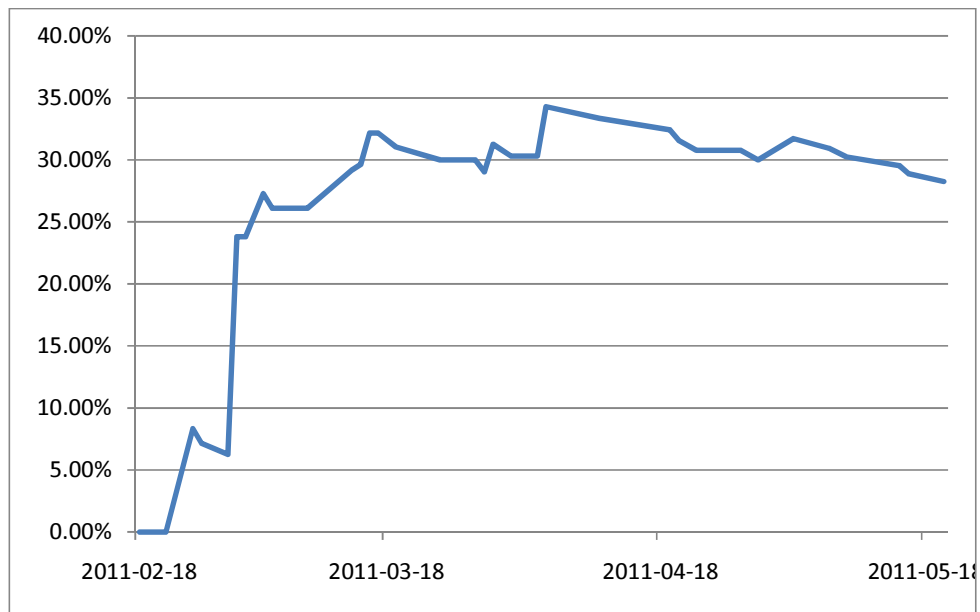
7 VARTOTOJŲ ELGESIO TYRIMAS

Idėjų banko sistema po paleidimo turėjo tik 7 pradinius vartotojus. Šių vartotojų buvo paprašyta išbandyti socialinių tinklų funkcionalumą bei pasikviesti prisijungti daugiau žmonių. Registruojantis prie sistemos buvo galima pasirinkti, ar vartotojas apie Idėjų banką sužinojo per Facebook. Taip surinkti duomenys, atvaizduoti 28 pav.



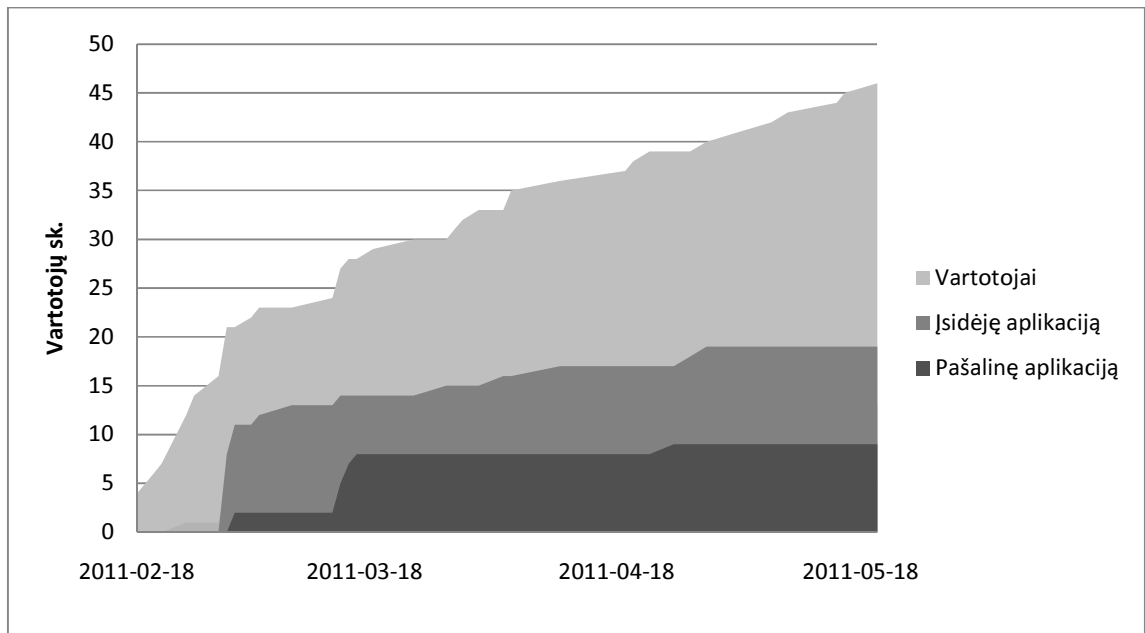
28 pav. Grafikas, parodantis kokia dalis žmonių apie Idėjų banką sužinojo iš Facebook

Pateiktame grafike matomas taškas, nuo kurio paleista Idėjų banko Facebook aplikacija. Tuo metu egzistuojantiems vartotojams buvo pasiūlyta išmėginti naują funkcionalumą bei paskleisti apie jį žinią Facebook'e. Tiriama laikotarpio pabaigoje apie 26% sistemos vartotojų buvo atėję iš Facebook, ir šis procentas jau buvo pradėjęs kristi. Proporcinis vartotojų, sužinojusių apie Idėjų banko sistemą per Facebook, kiekio kitimas atvaizduotas 29 pav.

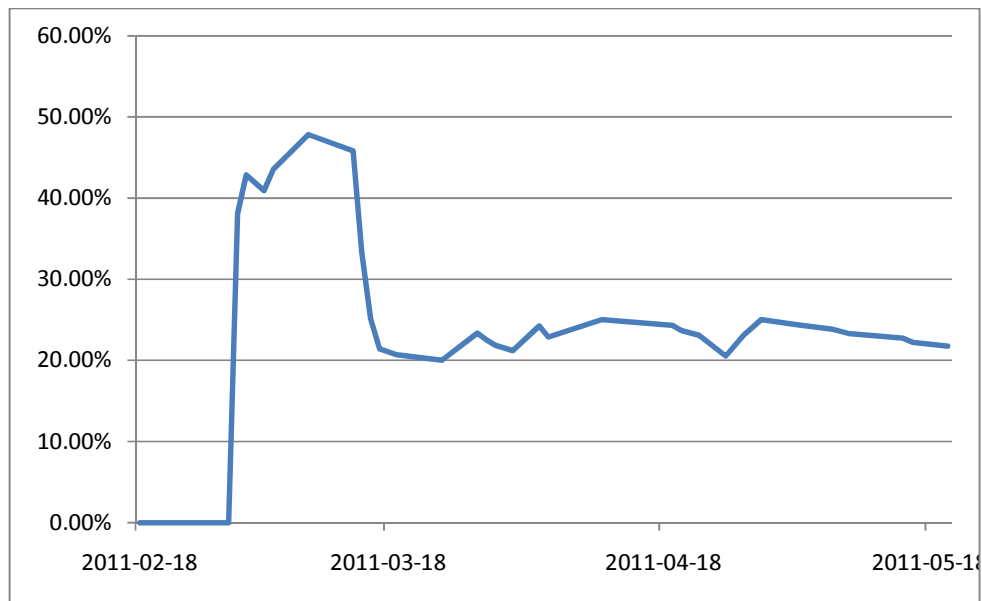


29 pav. Proporcinis vartotojų, sužinojusių apie Idėjų banko sistemą per Facebook, kiekio kitimas

Be šios informacijos Idėjų banko sistemoje taip pat buvo renkama informacija apie Facebook aplikacijos naudojimą. Buvo registruojami tokie įvykiai, kaip aplikacijos įsidėjimas į savo aplikacijų sąrašą bei jos pašalinimas iš aplikacijų sąrašo. Pašalinimu taip pat laikomas vartotojo nesutikimas pakartotinai suteikti aplikacijai reikalaujamų prieigos teisių.



30 pav. Facebook aplikacijos naudojimo statistika



31 pav. Facebook aplikaciją savo profilyje turinčių vartotojų procentinė išraiška

30 pav. ir 31 pav. matomas Facebook aplikacijos diegimų skaičiaus išaugimas, sutampantis su 28 pav. bei 29 pav. matomu Facebook vartotojų išaugimu. Čia taip atsispindi Facebook aplikacijos paleidimas bei informacijos apie tai išplatimas. Tačiau gana greitai po to stipriai padidėjo pašalinusių aplikaciją vartotojų. Taip atsitiko dėl to, kad prie Idėjų banko Facebook aplikacijos buvo pridėtas papildomas funkcionalumas, iš vartotojų reikalaujantis papildomų prieigos teisių. Pasikeitus reikalingoms prieigos teisėms, soc. karkasas automatiškai paprašė vartotojų suteikti naujas teises, tačiau didelė dalis vartotojų nesutiko to padaryti arba net rankiniu būdu pašalino aplikaciją iš savo Facebook aplikacijų sąrašo. Įdomu tai, kad jei pažiūrėsime į pirmą grafiką (28 pav.), maždaug toje pačioje vietoje, kaip ir aplikacijų pašalinimas, pamatysime vartotojų skaičiaus išaugimą. Gali būti, kad žmonėms pasipiktinus dėl reikalingų papildomų teisių, sistema susilaukė papildomo dėmesio.

8 IŠVADOS

- Idėjų banko sistema suprojektuota MVC principu. Tai padeda atskirti sistemos dizainą, duomenis ir valdantį kodą;
- Socialinių tinklų karkaso paketo (Soc. karkaso) duomenys suformatuojami, tačiau paties paketo nesaugomi. Šių duomenų saugojimo užduotis paliekama sistemai, naudojančiai karkasą. Taip projektuotojas nėra pririšamas prie vienos duomenų bazių valdymo sistemos;
- Soc. karkaso kontrolierio ir šablono pagalbininko objektai kuriami naudojant atitinkamą gamyklą. Taip Soc. karkasui leidžiama sukurti jau susietus reikiamus objektus bei atlikti kitus pasiruošimus. Kadangi OpenSocial yra bendra specifikacija, naudojama daugelio tinklapių, visai gali būti, kad nebus būtina kurti atskirų kontrolierių kiekvienam OpenSocial konteineriui. Tais atvejais, kai reikiamo kontrolierio ar šablono pagalbininko nėra, gamykla gali sukurti bendrinio funkcionalumo objektus;
- OpenSocial funkcionalumui įtraukti į Soc. karkasą panaudota OpenSocial PHP Client biblioteka. Tai turėtų pagreitinti atnaujinimus, jeigu kartais tokių atsirastų OpenSocial specifikacijoje. Taip pat ši biblioteka apima labai didelę OpenSocial funkcionalumo dalį;
- Soc. karkaso kokybės tyrimas parodė, kad programuojant sąsają su Facebook išorine sistema naudojantis Soc. karkasu galima sumažinti PHP kodo eilučių skaičių iki 8 kartų, palyginus su tuo atveju, jeigu būtų programuojama naudojantis tik viešai prieinama Facebook PHP kliento biblioteka. Kokybės tyrimas taip pat parodė, kad OpenSocial funkcionalumo pridėjimo metu didžiąją dalį kodo galima buvo kopijuoti iš Facebook versijos;
- Vartotojų elgesio tyrimas parodė, kad net verslo sektoriaus sistemos gali neblogai pasinaudoti socialinių tinklų viešumu, kad pritraukti naujų klientų – 25% - 30% vartotojų atėjo į sistemą apie ją sužinoję iš Facebook sistemos. Deja, iš surinktų duomenų neįmanoma pasakyti, kiek tokių vartotojų būtų buvę, jeigu Idėjų banko sistema nebūtų susieta su Facebook išorine sistema, arba jeigu ji būtų buvusi reklamuojama dar kaip nors kitaip, nei iš lūpų į lūpas. Taip pat reikia atkreipti dėmesį, kad surinktų duomenų imtis nebuvo labai didelė, todėl gautais rezultatais aklaai pasitikėti nederėtų.

9 LITERATŪROS SĄRAŠAS

1. **Cowan, J. 2005.** RESTful Web Services. [Tinkle] 2005 m. [Cituota: 2009 m. 11 23 d.] <http://home.ccil.org/~cowan/restws.pdf>.
2. **Facebook. 2011.** Facebook Developers. [Tinkle] 2011 m. [Cituota: 2011 m. 05 24 d.] <http://developers.facebook.com/docs/>.
3. **Fielding, R. 2000.** *Architectural Styles and the Design of Network-based Software Architectures* : Doktoro disertacija. University of California, 2000.
4. **Fielding, Roy. 2008.** REST APIs must be hypertext-driven. *Untangled*. [Tinkle] 2008 m. [Cituota: 2009 m. 11 23 d.] <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>.
5. **Fowler, M. 2003.** *Patterns of enterprise application architecture* : Addison-Wesley Professional, 2003.
6. **Graham, W. 2008.** *Facebook API: Developers Guide* : Apress, 2008.
7. **Hogan, B. 2010.** Chapter 11 - Visualizing and Interpreting Facebook Networks. [knygos aut.] D. Hansen, B. Shneiderman ir M. Smith. *Analyzing Social Media Networks with NodeXL*. Oxford : Elsevier, 2010.
8. **Laurent, S., Johnson, J. ir Dumbill, E. 2001.** *Programming Web Services with XML-RPC* : O Rilly, 2001.
9. **LeBlanc, J. 2011.** *Programming Social Applications* : O'Reilly Media, Inc., 2011.
10. **Marketer.lt. 2008.** LinkedIn panaudojimo scenarijai. [Tinkle] 2008 m. 11. [Cituota: 2009 m. 11 18 d.] <http://www.marketer.lt/linkedin-668.htm>.
11. **Meoa, P., et al. 2010.** *Recommendation of similar users, resources and social networks in a Social* : Elsevier, 2010.
12. **O'MALLEY, G. 2008.** Study: LinkedIn Users Have Higher Incomes. *MediaPost News*. [Tinkle] 2008 m. 11. [Cituota: 2009 m. 11 18 d.] http://www.mediapost.com/publications/?fa=Articles.showArticle&art_aid=94128.
13. **OpenSocial Blog. 2009.** Introducing the OpenSocial Development Environment. *OpenSocial Blog*. [Tinkle] 2009 m. 03. [Cituota: 2009 m. 11 23 d.] <http://blog.opensocial.org/2009/03/introducing-opensocial-development.html>.
14. **OpenSocial. 2009.** Main Page. *Open Social*. [Tinkle] 2009 m. 06. [Cituota: 2009 m. 11 20 d.] <http://wiki.opensocial.org>.
15. **Pautasso, C. 2009.** *RESTful Web service composition with BPEL for REST* : Elsevier, 2009.

16. **Shuaib, J. 2007.** Facebook vs. OpenSocial. *Slideshare*. [Tinkle] 2007 m. [Cituota: 2009 m. 11 23 d.] <http://www.slideshare.net/jawadshuaib/facebook-vs-open-social>.
17. **Tanenbaum, A. ir Steen, M. 2007.** *Distributed Systems Principles and Paradigms* : Prentice Hall, 2007.

10 TERMINŲ IR SANTRUMPŲ ŽODYNAS

Karkasas	Programinių priemonių (klasių, funkcijų) rinkinys, padedantis lengviau atlikti panašius ar pasikartojančius programavimo darbus
Vartotojas	Prie sistemos prisiregistravęs (priklausomai nuo konteksto ir prisijungęs) asmuo. Paprastai kalbant apie vartotojus turima galvoje ne konkretus asmuo, o su juo susietas sistemos objektas su savo duomenimis
Užklausa, tikslas	Vartotojo klausimas, problema, projektas ar produktas, kuriam spręsti ar tobulinti jis ieško idėjų ar tiesiog sprendimų
Idėja	Vartotojo idėja, pateikta kaip atsakymas į kurią nors užklausą
Užklauskos savininkas	Užklauskos autorius, t.y. vartotojas, kuris ją parašė
Sprendimas	Idėja ar kelios idėjos, užklauskos savininko pasirinkta kaip geriausia tinkamiausiai atsakanti į jo klausimą, išsprendžianti jo problemą ir pan.
Vadovas, idėjų paklauskos grupė	Šiai grupei priklausantys vartotojai ieško idėjų, t.y. gali teikti savo užklauskas
Darbuotojas, idėjų pasiūlos grupė	Šiai grupei priklauso vartotojai, galintys pateikti savo idėjas prie užklauskų
Komentaras	Idėjos komentaras, pateiktas vieno iš sistemos vartotojų
Balsas	Sistemos vartotojo pateiktas idėjos arba komentaro įvertinimas. Gali būti „už“ arba „prieš“
Komanda	Prie tikslo nurodyta sistemos vartotojų grupė. Tik šie vartotojai bei tikslo savininkas gali matyti tikslą
Socialinis tinklas	Tekste naudojamas internetinio socialinio tinklo prasme. Tai internetinis įrankis (svetainė), skirtas sudarinėti internetines bendruomenes, padėti žmonėms surasti bendraminčių, bendrauti, išreikšti save
Saitynas	Kitaip žinomas kaip internetas
Saityno servisas, API	Internetinės sistemos dalis (bei protokolas), skirta bendravimui (informacijos mainams) su kita internetine sistema
Aplikacija	(Tekste) programa (programinis kodas ir funkcionalumas), kurią galima įterpti į socialinį tinklą ar Idėjų banko sistemą žiniatinklio servisu pagalba
Išorinė sistema	Programinė sistema, fiziškai ar programiškai atskirta nuo atskaitos taško sistemos. Šios sistemos gali būti ir tame pačiame kompiuteryje

Asociatyvus masyvas	Tai duomenų elementų rinkinys, kur kiekvienas rinkinys turi savo vardą, pagal kurį jis gali būti nuskaitytas ar atnaujintas
OpenSocial konteineris	Taip OpenSocial sistemoje vadinamos sistemos, teikiančios API pagal OpenSocial specifikaciją
XML	(angl. Extensible Markup Language) – W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba ⁹
(X)HTML	HyperText Markup Language – Hiperteksto žymėjimo kalba – kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete. ¹⁰ X priekyje reiškia, kad naudojama XML standartus atitinkanti HTML atmaina
HTTP	Plačiausiai internete naudojamas hiperteksto ir su juo susijusių duomenų perdavimo protokolas
JavaScript	Internetinių puslapių dinaminio programavimo kalba, leidžianti atlikti HTML transformacijas bei dinaminį bendravimą su serveriu tiesiai vartotojo kompiuteryje
PHP	(angl. PHP - <i>Hypertext Preprocessor</i>) dinaminė interpretuojama programavimo kalba
MySQL	reliacinė duomenų bazių valdymo sistema, palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu
MVC	(angl. <i>Model-View-Controller</i>) – modeliavimo šablonas, dažnai naudojamas internetiniuose puslapiuose. Pagrįstas sistemos funkcijų skaidymu į modelius (duomenų), vaizdinius (pateikimo) ir kontrolierius (valdančiuosius)
JSON	(angl. JavaScript Object Notation) – JavaScript objektų užrašymo būdas, tačiau dėl paprastumo naudojamas ne tik JavaScript kalboje

⁹ Iš Vikipedijos (<http://lt.wikipedia.org/wiki/XML>)

¹⁰ Iš Vikipedijos (<http://lt.wikipedia.org/wiki/HTML>)

11 PRIEDAI

11.1 Surinkti vartotojų naudojimosi sistema duomenys

2 lentelė Vartotojų naudojimosi sistema duomenys

Nr.	Užsiregistravo	Sužinojo apie sistemą iš FB	Įsitraukė aplikaciją į FB	Pašalino aplikaciją iš FB
1	2011-02-18	Ne	2011-03-01	
2	2011-02-18	Ne	2011-03-02	
3	2011-02-18	Ne	2011-03-28	2011-03-17
4	2011-02-18	Ne	2011-03-09	
5	2011-02-20	Ne	2011-03-01	2011-03-15
6	2011-02-20	Ne	2011-03-01	2011-03-16
7	2011-02-21	Ne	2011-03-01	
8	2011-02-24	Ne		
9	2011-02-24	Ne		
10	2011-02-24	Taip	2011-03-01	
11	2011-02-24	Ne		
12	2011-02-24	Ne	2011-03-02	2011-03-15
13	2011-02-25	Ne		
14	2011-02-25	Ne		
15	2011-02-28	Ne	2011-03-01	2011-03-15
16	2011-02-28	Ne		
17	2011-03-01	Taip	2011-03-01	2011-03-02
18	2011-03-01	Taip	2011-03-01	
19	2011-03-01	Taip	2011-03-02	2011-03-02
20	2011-03-01	Taip	2011-03-05	2011-03-16
21	2011-03-01	Ne		
22	2011-03-04	Taip		
23	2011-03-05	Ne		
24	2011-03-14	Taip	2011-03-15	
25	2011-03-15	Ne		
26	2011-03-15	Ne		
27	2011-03-15	Taip		
28	2011-03-16	Taip		
29	2011-03-19	Ne		
30	2011-03-24	Ne		
31	2011-03-29	Ne		
32	2011-03-30	Taip	2011-04-04	2011-04-25
33	2011-04-01	Ne		
34	2011-04-05	Taip	2011-04-27	
35	2011-04-05	Taip	2011-04-11	
36	2011-04-11	Ne		
37	2011-04-19	Ne		
38	2011-04-20	Ne		
39	2011-04-22	Ne	2011-04-29	
40	2011-04-29	Ne		
41	2011-05-03	Taip		
42	2011-05-07	Ne		

43	2011-05-09	Ne		
44	2011-05-15	Ne		
45	2011-05-16	Ne		
46	2011-05-20	Ne		

3 lentelė Paruošti grafikų braižymui vartotojų naudojimosi sistema duomenys

Data	Vartotojai	Iš Facebook	Įsidėję aplikaciją	Pašalinę aplikaciją	% iš facebook	Naudojančių aplikaciją	% naudojančių aplikaciją
2011-02-18	4	0	0	0	0.00%	0	0.00%
2011-02-20	6	0	0	0	0.00%	0	0.00%
2011-02-21	7	0	0	0	0.00%	0	0.00%
2011-02-24	12	1	0	0	8.33%	0	0.00%
2011-02-25	14	1	0	0	7.14%	0	0.00%
2011-02-28	16	1	0	0	6.25%	0	0.00%
2011-03-01	21	5	8	0	23.81%	8	38.10%
2011-03-02	21	5	11	2	23.81%	9	42.86%
2011-03-04	22	6	11	2	27.27%	9	40.91%
2011-03-05	23	6	12	2	26.09%	10	43.48%
2011-03-09	23	6	13	2	26.09%	11	47.83%
2011-03-14	24	7	13	2	29.17%	11	45.83%
2011-03-15	27	8	14	5	29.63%	9	33.33%
2011-03-16	28	9	14	7	32.14%	7	25.00%
2011-03-17	28	9	14	8	32.14%	6	21.43%
2011-03-19	29	9	14	8	31.03%	6	20.69%
2011-03-24	30	9	14	8	30.00%	6	20.00%
2011-03-28	30	9	15	8	30.00%	7	23.33%
2011-03-29	31	9	15	8	29.03%	7	22.58%
2011-03-30	32	10	15	8	31.25%	7	21.88%
2011-04-01	33	10	15	8	30.30%	7	21.21%
2011-04-04	33	10	16	8	30.30%	8	24.24%
2011-04-05	35	12	16	8	34.29%	8	22.86%
2011-04-11	36	12	17	8	33.33%	9	25.00%
2011-04-19	37	12	17	8	32.43%	9	24.32%
2011-04-20	38	12	17	8	31.58%	9	23.68%
2011-04-22	39	12	17	8	30.77%	9	23.08%
2011-04-25	39	12	17	9	30.77%	8	20.51%
2011-04-27	39	12	18	9	30.77%	9	23.08%
2011-04-29	40	12	19	9	30.00%	10	25.00%
2011-05-03	41	13	19	9	31.71%	10	24.39%
2011-05-07	42	13	19	9	30.95%	10	23.81%
2011-05-09	43	13	19	9	30.23%	10	23.26%
2011-05-15	44	13	19	9	29.55%	10	22.73%
2011-05-16	45	13	19	9	28.89%	10	22.22%
2011-05-20	46	13	19	9	28.26%	10	21.74%