

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Tomas Selickas

**Duomenų bazės turinio publikavimo
interaktyviuose tinklapiuose galimybių tyrimas**

Magistro darbas

Darbo vadovė :

doc. dr. Rita Butkienė

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

**Duomenų bazės turinio publikavimo
interaktyviuose tinklapiuose galimybių tyrimas**

Magistro darbas

Darbo vadovė :

doc. dr. Rita Butkienė
2011-05-

Recenzentas :

doc. dr. Armantas Ostreika
2011-05-

Atliko:

IFM-9/4 gr. studentas
Tomas Selickas
2011-05-

Kaunas, 2011

Turinys

SUMMARY	5
TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS	6
1. ĮVADAS	7
2. ANALIZĖS DALIS	8
2.1. TYRIMO TIKSLAS IR UŽDAVINIAI	8
2.2. ANALIZĖS TIKSLAS	8
2.3. TYRIMO OBJEKTO ANALIZĖ	8
2.3.1. Lietuvos ornitologų draugijos veiklos analizė	8
2.3.2. Exhibit sistemos analizė.....	10
2.3.3. Google API analizė.....	17
2.4. VARTOTOJŲ ANALIZĖ	17
2.4.1. Vartotojų aibė, tipai ir savybės.....	17
2.4.2. Vartotojų tikslai ir problemos.....	18
2.5. ESAMŲ SPRENDIMŲ ANALIZĖ	18
2.5.1. Duomenų saugojimo formatai	18
2.5.1.1. JSON formatas.....	18
2.5.1.2. Reliacinės duomenų bazės	21
2.5.2. Kiti duomenų atvaizdavimo ir filtravimo sprendimai	21
2.5.2.1. Flamenco	21
2.5.2.2. mSpace.....	22
2.5.2.3. jSpace.....	24
2.6. ANALIZĖS APIBENDRINIMAS.....	26
2.7. SIEKIAMAS SPRENDIMAS	27
2.8. ANALIZĖS IŠVADOS	28
3. INFORMACINĖS SISTEMOS REIKALAVIMŲ SPECIFIKACIJA IR ANALIZĖ	29
3.1. REIKALAVIMŲ SPECIFIKACIJA	29
3.1.1. Panaudojimo atvejai.....	29
3.1.2. Vartotojo sąsajos modelis.....	33
3.2. DALYKINĖS SRITIES MODELIS.....	37
3.3. DUOMENŲ BAZĖS SCHEMA	40
4. INFORMACINĖS SISTEMOS PROJEKTAS	42
4.1. PROJEKTO TIKSLAS.....	42
4.2. SISTEMOS LOGINĖ ARCHITEKTŪRA	42
4.3. SISTEMOS REALIZACIJOS MODELIS	43
5. LIETUVOS ORNITOFAUNOS REGISTRAVIMO SISTEMOS REALIZACIJA	47
5.1. VEIKIMO APRAŠYMAS	47
5.1.1. Sistemos paskirtis	47
5.1.2. Neregistruoto vartotojo veiklos	48
5.1.1. Prisijungusio sistemos vartotojo veiklos.....	50
5.2. TESTAVIMO MODELIS	54
5.3. TESTAVIMO DUOMENYS IR REZULTATAI.....	54
6. EKSPERIMENTINIS SISTEMOS TYRIMAS	58
6.1. EKSPERIMENTO PLANAS	58
6.2. EKSPERIMENTO REZULTATAI.....	58
6.2.1. Dinamiškai kintančių duomenų atvaizdavimo su Exhibit įrankiu tyrimas	58
6.2.2. Exhibit įrankio greitaveikos tyrimas.....	58
6.3. SISTEMOS VEIKIMO IR SAVYBIŲ ANALIZĖ, KOKYBĖS KRITERIJŲ ĮVERTINIMAS	61
6.4. SISTEMOS TAIKYMO REKOMENDACIJOS.....	64
7. IŠVADOS	65
8. LITERATŪRA	66

9.	PRIEDAI	68
1	PRIEDAS. KOMPIUTERIZUOJAMŲ PANAUDOS ATVEJŲ SPECIFIKACIJA	68
2	PRIEDAS. SUPROJEKTUOTOS DUOMENŲ BAZĖS LENTELIŲ ATRIBUTŲ APRAŠYMAS	77
3	PRIEDAS. DUOMENŲ BAZĖS UŽKLAUSA <i>EXHIBIT</i> DUOMENIMS FORMUOTI	83
4	PRIEDAS. EXHIBIT ĮRANKIO VEIKIMO SPARTOS PRIKLAUSOMYBĖ.....	84

Research on the Possibilities of Publishing Data Base content in Interactive Web Pages

Summary

There is common situation where are much of important data on the website, but the data are given in an inconvenient form to use. The lack of information interactivity determines complicated understanding and acquisition of the given information, also it directly determines the decline of website visitors. The being of the mentioned situation is topical to websites where collecting and publishing a lot of specific data.

An aim of this research is to find a way to realize qualitatively publishing and visualization of the data stored in information system database.

An analysis of the present decisions showed that *Exhibit* is great tool to solve existing problem. But this tool have a limitation. Exhibit is working with static (stored in files) information. Also creation of internal *Exhibit* data structure takes too long time.

In this research were found and adapted facilities that let to use an *Exhibit* tool for publishing dynamic information. Also was improved method for faster internal *Exhibit* data structure creation.

Keywords: Exhibit, JSON, database data publishing, data visualization, Symfony.

Terminų ir santrumpų žodynėlis

1 lentelė Terminų ir santrumpų paaiškinimai

Santrumpa, terminas	Paaiškinimas
Ornitofauna	Paukščių rūšių visuma
MySQL	Viena iš reliacinių duomenų bazių valdymo sistemų, dirbanti SQL kalbos pagrindu.
SQL (angl. <i>Structured Query Language</i>)	Struktūrizuota užklausų kalba
DBVS	Duomenų bazės valdymo sistema
SIMILE (angl. <i>Semantic Interoperability of Metadata and Information in unLike Environments</i>)	Projektas, kuris orientuojasi į konkurencingų, atviro kodo įrankių kūrimą, kurie suteikia informacijos valdymo, vizualizavimo funkcionalumus.
JSON (angl. <i>JavaScript Object Notation</i>)	Nesudėtingas duomenų apsikeitimo formatas, kuris išpopuliarėjo ir tapo alternatyva populiariam ir daugelyje sričių naudojamam XML formatui
HTML (angl. <i>Hyper text Markup Language</i>)	Kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete
AJAX (angl. <i>Asynchronous JavaScript and XML</i>)	Svetainių programavimo technologija, kuri naudojama maksimaliam interaktyvumui pasiekti
RDF (angl. <i>Resource Description Framework</i>)	Standartas, skirtas duomenų apsikeitimui internete
MIT (angl. <i>Massachusetts Institute of Technology (MIT)</i>)	Masačusetso technologijų institutas
TSV (angl. <i>Tab Separated Values</i>)	failų formatas, kuris skirtas tekstiniu formatu aprašyti duomenų lenteles
3store	C kalbos bibliotekų branduolys naudojamas MySQL DBVS, kad saugoti RDF duomenis
RDQL (angl. <i>RDF Data Query Language</i>)	Užklausų kalba skirta dirbti su RDF duomenimis
SPARQL (angl. <i>SPARQL Protocol and RDF Query Language</i>)	Užklausų kalba skirta dirbti su RDF duomenimis
SOAP (angl. <i>Simple Object Access Protocol</i>)	Protokolas skirtas apsikeisti struktūrine informacija internete
LOD	Lietuvos ornitologų draugija
LOFK	Lietuvos ornitofaunistinė komisija
UTM (angl. <i>Universal Transverse Mercator</i>)	Tam tikra geografinių koordinatų sistema
Symfony	Atviro kodo internetinių tinklapių kūrimo karkasas, paremtas PHP programavimu
MVC (angl. <i>Model View Controller</i>)	Trijų lygių architektūra, kurią sudaro duomenų atvaizdavimo, skaičiavimų vykdymo ir duomenų lygis
PHP (angl. <i>Hypertext Preprocessor</i>)	Plačiai paplitusi dinaminė interpretuojama programavimo kalba, kuri specialiai pritaikyta internetinių svetainių kūrimui
JavaScript	Objektiškai orientuota skriptų programavimo kalba
SPSS (angl. <i>Statistical Package for the Social Science</i>)	Statistinės informacijos apdorojimo programinis paketas
Statistinis reikšmingumas (p)	Skaičius, kuris nurodo ar dviejų testavimo imčių įverčių skirtumas yra statistiškai reikšmingas ar ne.

1. Įvadas

Kuriamų internetinių tinklapių profesionalumą apsprendžia ne tik pateikiamo turinio formos išskirtinumas, bet ir tokių esminių dalykų, kaip filtravimas, grupavimas, rikiavimas, įvairių žemėlapių ar diagramų pritaikymas. Tačiau, dažnai būna taip, kad internetiniame tinklapyje yra daug svarbių duomenų, tačiau jie nėra pateikti lengvai suprantamoje formoje. Būtent pateiktos informacijos interaktyvumo stoka, sąlygoja ne tik esamos informacijos sudėtingesnę suvokimą ar įsisavinimą, bet taip pat tiesiogiai siejasi su lankytojų srauto mažėjimu. Pastarosios situacijos buvimas ypač aktualus internetiniams tinklapiams, kuriuose kaupiama ir publikuojama daug specifinės srities duomenų.

Šiuo metu rinkoje pateikiami įvairūs programiniai sprendimai, siūlantys vieno ar kitokio pobūdžio interaktyvią duomenų analizės realizaciją. Duomenų grafinės analizės atžvilgiu didelį žingsnį padarė *Google API*, *Yahoo*, siūlantys ne tik atvaizdavimo žemėlapyje paslaugas, bet ir įvairius grafinius, tai yra diagramomis paremtus duomenų atvaizdavimo sprendimus. Tačiau, šių sprendimų panaudojimas tik dalinai išsprendžia problemą, nes turi būti galimybė netik atvaizduoti duomenis, bet juos taip pat filtruoti, grupuoti ir kitomis dinaminėmis priemonėmis valdyti jų kiekį. Taigi, grafinio ir kitos formos struktūrizuotas duomenų atvaizdavimo susiejimas su filtrais leidžia ganėtinai išsamiai nagrinėti ir peržiūrėti duomenis. Tokio pobūdžio sprendimų svarba vis labiau auga, o naujos technologijos leidžia vis lanksčiau šiuos sprendimus realizuoti.

Šiame darbe siekiama išanalizuoti įrankius, kurie skirti interaktyviam duomenų atvaizdavimui, bei praplėsti pasirinkto įrankio funkcionalumą taip, kad atsirastų galimybė duomenų bazėje saugomus duomenis analizuoti įvairiais pjūviais. Taigi, suformuluota tokia tiriamojo darbo sritis, objektas ir problema:

Sritis – internetinių IS duomenų filtravimo ir vizualizavimo priemonės ir jų taikymas.

Objektas – internetinės IS duomenų bazės duomenų filtravimas ir vizualizavimas.

Problema – sudėtingesnių internetinių informacijos kaupimo sistemų realizavimo metu dažnai susiduriama su sukauptų duomenų nagrinėjimo sunkumais. Šių sunkumų priežastis – sudėtingumas realizuojant išsamias duomenų filtravimo, grupavimo, rūšiavimo bei duomenų vizualizavimo funkcijas, leidžiančias duomenis išnagrinėti įvairias pjūviais.

Analizės dalyje nagrinėjamos *Exhibit* ir panašios paskirties įrankių galimybės esamai problemai spręsti. Lyginami šių įrankių privalumai ir trūkumai, tam kad būtų pasirinktas ir pritaikytas labiausiai reikalavimus tenkinantis sprendimas.

Projektinėje dalyje pateikiami kuriamos sistemos reikalavimai, aprašoma sistemos dinamika, architektūra ir siekiamas sprendimas.

2. Analizės dalis

2.1. Tyrimo tikslas ir uždaviniai

Tikslas – rasti ir pritaikyti sistemų kūrėjo darbo sąnaudoms neimlų įrankį ar įrankius, leidžiančius realizuoti dinamišką ir patogų atributinių duomenų analizavimą ir vizualizavimą.

Uždaviniai :

- Išanalizuoti kompiuterizuojamą sritį;
- Išanalizuoti egzistuojančius problemos sprendimo būdus;
- Realizuoti informacinę sistemą, kuri pademonstruotų pasirinktos priemonės taikymą.
- Eksperimentiškai ištirti sukurtą sprendimą.

2.2. Analizės tikslas

Esminis analizės tikslas – rasti būdą, kuris pagreitintų informacinės sistemos duomenų atvaizdavimo ir vizualizavimo realizaciją. Taigi, analizės eiga susidės iš tokių dalių:

1. kompiuterizuojamos srities analizės;
2. *Exhibit* įrankio analizės;
3. kitų egzistuojančių sprendimų analizės;
4. informacijos vizualizavimo priemonių analizės;

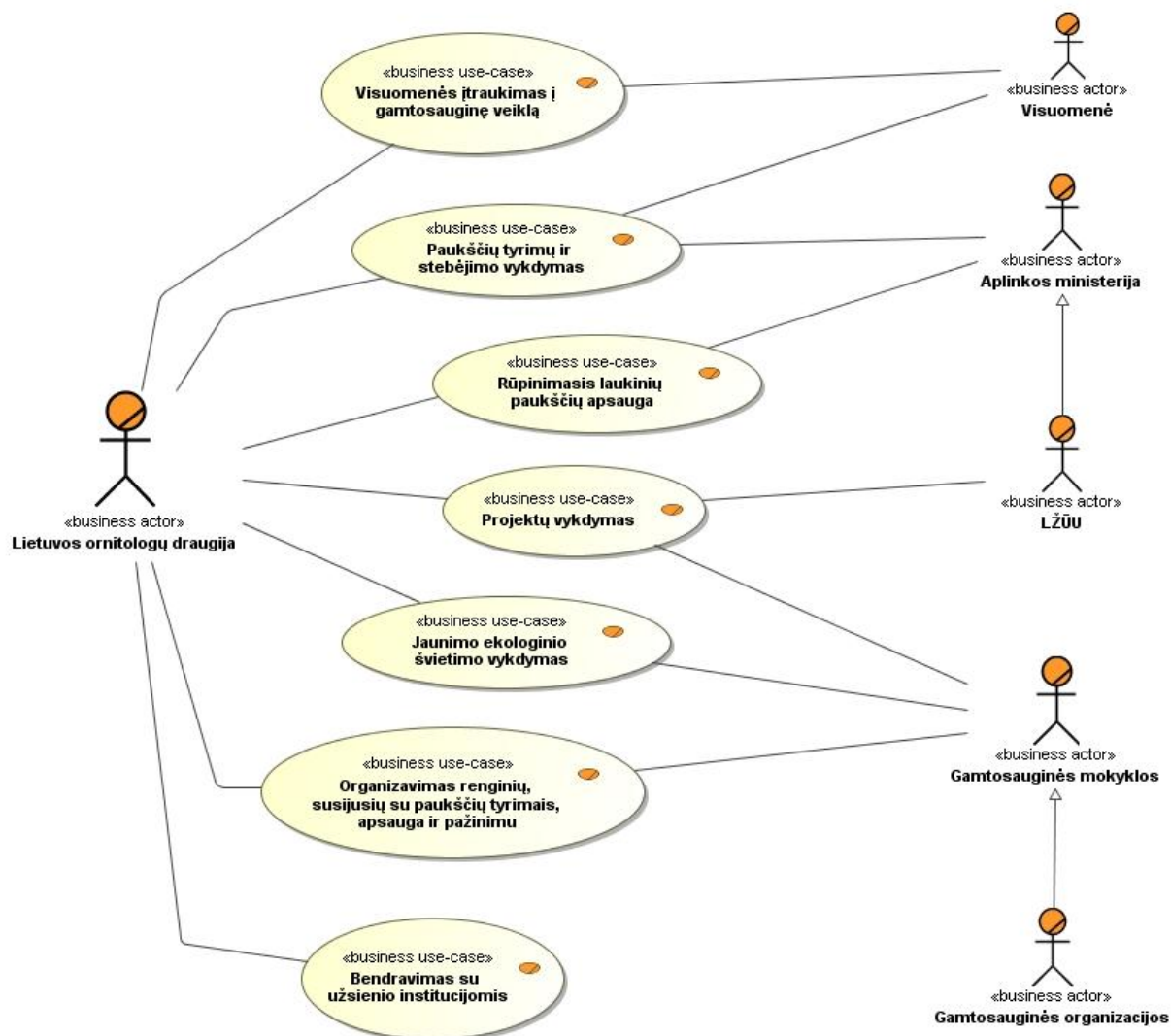
2.3. Tyrimo objekto analizė

2.3.1. Lietuvos ornitologų draugijos veiklos analizė

Lietuvos ornitofaunos registravimo informacinės sistemos kūrimo tikslas nėra vienareikšmis. Svarbiausia, kad tokios sistemos sukūrimas leistų populiarinti paukščių tyrimus, stebėjimus, o tuo pačiu ir gamtos apsaugą Lietuvoje. Be to, iki šiol nebuvo sukurta vieša informacinė sistema, kuri būtų skirta paukščių stebėjimams Lietuvos teritorijoje registruoti, kaupti, susisteminti, analizuoti ir pateikti. Pastarųjų galimybių realizavimas, leistų atlikti Lietuvos paukščių gausos ir jos dinamikos tyrimus, kurie reikalingi ne tik retų ar jau nykstančių paukščių apsaugos organizavimui, bet ir pačios aplinkos kokybiškumui tirti. Be to, struktūrizuota informacija leistų formuoti Lietuvoje gyvenančių paukščių atlasą, bei pasitarnautų priimant vienokius ar kitokius gamtosaugai būtinus sprendimus.

Norint tinkamai suprojektuoti sistemą, reikia įsigilinti į ornitologų draugijos veiklą. Taigi, 1 pav. pateikta supaprastinta Lietuvos ornitologų draugijos veikla. Joje yra išskirta

keliolika esminių veiklų. Kartu su veiklomis yra pavaizduoti aktoriai, kurie tiesiogiai ar netiesiogiai siejasi su Lietuvos ornitologų draugijos veikla.

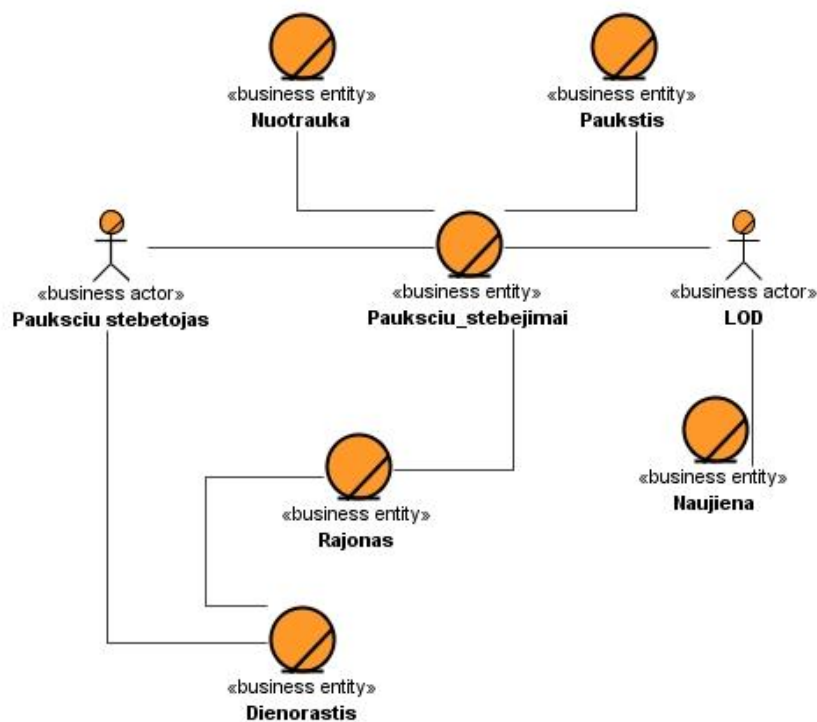


1 pav. Lietuvos ornitologų draugijos veiklos panaudos atvejų diagrama

Panaudos atvejų diagramos esminė veikla – „Paukščių tyrimų ir stebėjimų vykdymas“. Būtent šią veiklą numatoma kompiuterizuoti. Dėl to būtina gerai išsiginėti į šią specifinę ir konkrečią sritį, nes kuriama sistema turėtų tapti stebėjimo duomenų saugykla, tai yra iš dalies pakeisti duomenų saugojimą popieriniame variante ar kitokio pobūdžio failuose, bei tapti duomenų analizės įrankiu, leidžiančiu sukauptus duomenis apžvelgti įvairiais pjūviais.

Žiūrint iš abstraktesnės pozicijos, planuojamos kurti informacinės sistemos paskirtis – surinkti duomenis apie Lietuvos teritorijoje pastebėtus ar joje gyvenančius paukščius. Taigi, esminė informacija būtų: paukščio rūšies identifikavimas, pastebėtų paukščių kiekis, vietovės pavadinimas. Tačiau, be šios esminės informacijos yra svarbūs ir kiti paukščių stebėjimo niuansai: paukščių elgesys, stebėto paukščio išvaizda, giesmės pobūdis, reakcija į žmogų ir

taip toliau. Todėl ganėtinai svarbu struktūrizuoti kaupti tokio pobūdžio informaciją, kuri taip pat ganėtinai svarbi atliekant tiek lokalius (tam tikros vietovės), tiek globalius ornitofaunos tyrimus. 2 pav. pateiktas veiklos esybių modelis, kurio turinys supaprastintai atspindi kaupiamos informacijos pobūdį, bei sukuria pagrindą vystyti šį modelį iki išbaigtos duomenų bazės schemas, kuri leistų kausti reikiamo išsamumo informaciją apie atliktus paukščių stebėjimus.



2 pav. Paukščių stebėjimų veiklos esybių modelis

2.3.2. Exhibit sistemos analizė

Exhibit – tai atviro kodo įrankis, kurio tikslas supaprastinti duomenų nagrinėjimo procesą interneto tinklalapyje, tai yra realizuoti duomenų filtravimą ir atvaizdavimą įvairiais pjūviais. Pirmoji šio įrankio versija pasirodė 2006 metų rugsėjo mėnesį. Jo sumanytoju laikomas David Huynh, kuris šį projektą buvo pavadinęs „*rubik*“, bet vėliau, dėl problemų susijusių su autorinėmis teisėmis, pervadino. Šiuo metu *Exhibit* yra sudedamoji kito didelio projekto pavadinimu *SIMILE*¹ dalis. Šiam projektui vadovauja Masačusetso technologijų institutas (angl. *Massachusetts Institute of Technology (MIT)*). Be *Exhibit*, *MIT* mokslų daktaras David Huynh iniciavo tokius projektus, kaip *Babel*, *Potluck*, *Timeline*, *Ajax*, *Sifter*.

Šio įrankio sukūrimą įtakėjo keletas internetinių tinklapių trūkumų, susijusių su duomenų panaudojimu. Pirmasis trūkumas – duomenų saugojimas ir jų naudojimas tinklalapyje. Tarkime turime standartinį tinklapį, kurio tikslas atvaizduoti statinę ar retai

¹ *SIMILE* (angl. *Semantic Interoperability of Metadata and Information in unLike Environments*)

kintančią informaciją. Taigi, tokios sistemos duomenys dažniausiai saugomi tekstiniame ar kitokio tipo faile. Viso to pasėkoje atsiranda problemos susijusios su duomenų nagrinėjimu ir vizualizavimu. Taip pat, gali būti taip, kad kuriama sudėtingesnė sistema, kurioje turi būti realizuotos įvairios filtravimo, rūšiavimo, duomenų vizualizavimo funkcijos. Nors sistemoje duomenys gali retai ar visai nekisti, sistemos kūrimas vis vien turi remtis duomenų saugojimu duomenų bazėje, kadangi taip paprasčiau realizuoti pastarąsias funkcijas. Tačiau, tokių sistemų kūrimas užima nemažai laiko ir brangiai kainuoja, kadangi projektuojama duomenų bazės schema, realizuojamos įvairios duomenų bazės užklausos, testuojamas korektiškas sistemos veikimas skirtingose naršyklėse, reikalinga nuolatinė sistemos priežiūra [Huynh, 2007]. Antrasis trūkumas – internetinių sistemų struktūra. Pastebėta, kad didelį pranašumą teikia trijų lygių (angl. *three-tier*) struktūros panaudojimas, kuris susideda iš: duomenų lygio, sistemos logikos lygio bei informacijos vaizdavimo lygio. Toks architektūrinis sprendimas teikia didelį pranašumą, kadangi duomenų ir informacijos atvaizdavimo lygiai yra atskirti. Taigi, naudojantis tuo pačiu informacijos atvaizdavimo šablonu galima atvaizduoti skirtingus duomenų rinkinius arba tuos pačius duomenis atvaizduoti skirtingais būdais ir tokiu būdu pasiekti didesnę sistemos efektyvumą. Tačiau, tokios struktūros palaikymas turi savų niuansų, nes reikia palaikyti korektišką sistemos veikimą tarp skirtingų interneto serverių, naršyklių ar platformų.

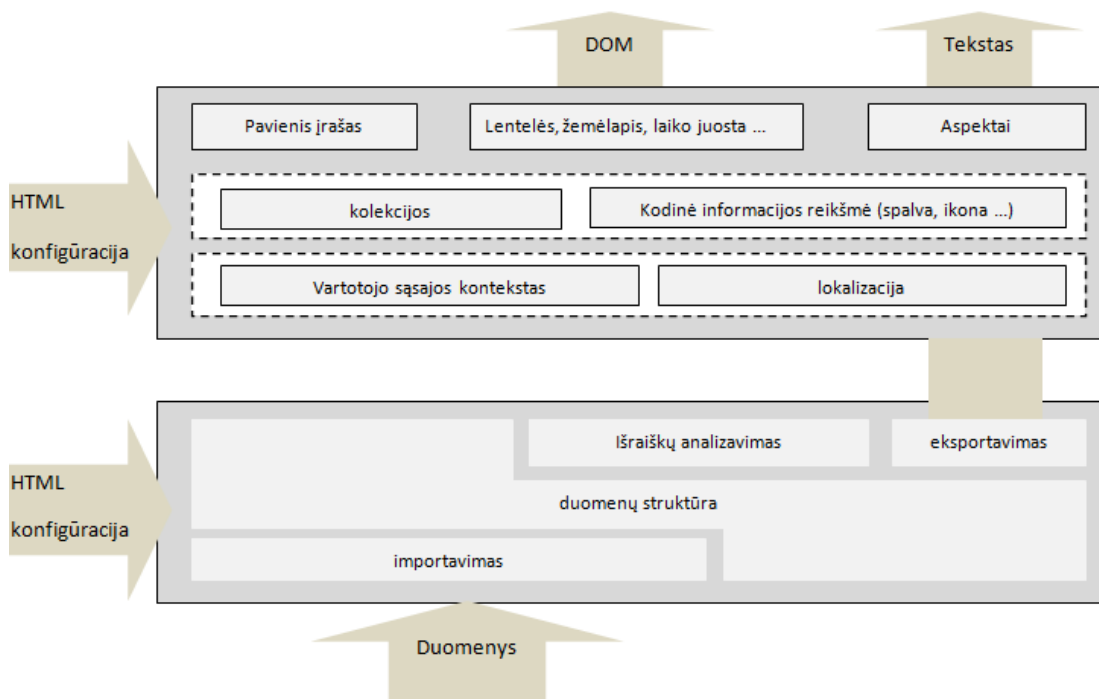
Apibendrintai galima teigti, kad sukurtasis įrankis *Exhibit* į tekstą orientuotiems (angl. *text-centric*) tinklapiams suteikia tokias galimybes, kurias turi tinklapiai besiremiantys į duomenis orientuota (angl. *data-centric*) architektūra. Be to išvengia su pastarąja architektūra susijusių niuansų.

Principinė *Exhibit* veikimo schema pateikta 3 paveikslėlyje. Šioje schemoje galima pastebėti, kad į *Exhibit* įrankį patenka dvejopa informacija, tai yra duomenys *JSON*² formatu ir *HTML*³ kalba aprašyti vartotojo sąsajos parametrai. *HTML* duomenys nusako, kokie filtrai, duomenų atvaizdavimo būdai bus naudojami atvaizduojant perduodamą informaciją. Taigi, duomenų užkrovimui naudojamos importavimo funkcijos, kurios pasiremdamos *AJAX*⁴ *POST* metodika užsikrauna siunčiamus duomenis. Vėliau iš esamų duomenų sukuriamas objektų masyvas, kurio struktūra primena orientuotą grafą. Vėliau šie duomenys susiejami su filtrais ir atvaizduojami vartotojo naršyklėje.

² JSON (angl. *JavaScript Object Notation*)

³ HTML (angl. *Hyper text Markup Language*)

⁴ AJAX (angl. *Asynchronous JavaScript and XML*)



3 pav. *Exhibit* įrankio principinė veikimo schema

3 pav. viršuje esančią vartotojo sąsają sudaro dvi esminės sudedamosios dalys: saugomos informacijos vaizdas ir duomenų filtrai (angl. *facet*). Šių komponentų išdėstymą tinklapyje kontroliuoja pats kūrėjas [Huynh, 2007]. Duomenų vizualizavimui naudojami įvairios priemonės: žemėlapis, laiko juostos, histogramos, lentelės, miniatiūros (angl. *thumbnail*). Grafinio vaizdavimo būdą ir parametrus apibrėžia pats programuotojas. Be to, jei siūlomų galimybių neužtenka, *Exhibit* vartotojo sąsaja gali būti išplėsta sistemos kūrėjo pasirinktomis ir pritaikytomis vizualizavimo priemonėmis. Žemiau esančiame paveikslėlyje pateiktas *Exhibit* įrankio duomenų analizės ir vizualizavimo pavyzdys su pagrindinių sąsajos komponentų paaiškinimais.

PAUKŠČIŲ STEBĖJIMAI

Paieška

ieškoti

Rajonas	Stebėjimo tipas	RK	IUCN	Metai	Mėnuo
1 Alytaus r.	7 Fenologiniai paukščių stebėjimai	39 0	1 EN	123 2009	36 Gegužė
37 Ignalinos r.		1 I	1 GE	111 2010	10 Sausis
2 Kelmės r.	2 Lizdai	5 R	44 LC	46 2011	
1 Klaipėdos r.		1 V			
1 Plungės r.					
1 Radviliškio r.					

Duomenų atvaizdavimo formos

PAUKŠČIŲ STEBĖJIMO ŽEMĖLAPIS • STEBĖJIMŲ KIEKIŲ ŽEMĖLAPIS • LAIKO DIAGRAMA • KALENDORIUS • STEBĖJIMŲ SĄRAŠAS • DETALI INFORMACIJA

Aspektų sąrašas

Klykuolė

Data: 2011-01-21

UTM kvadratas: FG4b1

Rajonas: Raseinių r.

Paukščių kiekis: 8

Paukščio lytis: Patinas

Stebėjimo tipas: Fenologiniai paukščių stebėjimai

Stebėtojas: Iina

Lytis 2/3

- 22 Nenustatyta
- 30 Patelė
- 16 Patinas

Paukščio amžius

- 4 1cy+
- 8 2cy+
- 3 3cy
- 8 3cy+
- 6 Nesubrendęs

Aktyvus filtras

Biotopas 2/3

- 24 Miestas
- 20 Mišrus miškas
- 26 Pajūris
- 17 Pievos
- 23 Pušymas

Augalas

- 46 nenustatytas

4 pav. Exhibit įrankio panaudojimo pavyzdys

Norint pradėti naudoti *Exhibit* įrankį, pirmiausia reikia suformuoti duomenų failą ir apibrėžti duomenų vaizdavimo struktūrą. Duomenys turi būti saugomi *JSON* formatu, tačiau jei informacija saugoma kitokiu formatu, pvz.: *Bibtex*, *Excel*, *RDF/XML*, *N3*, tai jie nesunkiai gali būti konvertuojami į reikiamą formatą naudojantis *Babel*⁵ failų konvertavimo įrankiu. Programos kode 5 pav. pavaizduoti keli Exhibit įrankio komponentai: filtravimo aspektai (angl. *facets*), grafinio vaizdo apibrėžimo komponentai (angl. *views*), pavienių elementų vaizdavimo apibrėžimo komponentai (angl. *lens*). Be šių komponentų yra ir duomenų

⁵ <http://simile.mit.edu/babel/>

rinkiniai (angl. *collections*) ir pagalbiniai elementai pavadinti *coders*. Kiekvienas iš šių komponentų turi savo paskirtį:

1. *Collections* – apibrėžia duomenų rinkinį pagal tam tikrą kriterijų. Kiekvienas rinkinys apibrėžiamas identifikatoriumi, kuris specifikuojamas naudojant *HTML id* atributą.
2. *Facets* komponento tikslas – realizuoti duomenų filtravimą.
3. *Views* – apibrėžia duomenų vaizdavimą.
4. *Lenses* – skirtas atvaizduoti vienam duomenų elementui.
5. *Coders* skirtas elementų rinkiniui paversti į tam tikrą vizualią formą, pvz.: spalvą, ikoną ir pan.

```

0 10 20 30 40 50 60 70 80 90 100 110
4 <?php /*----- TEXT SEARCH -----*/ ?>
5 <div class="search_section" >
6   <div class="text_search">
7     <div ex:role="facet" ex:facetClass="TextSearch"></div>
8   </div>
9   <div id="search_label" style="float:right;"><?php echo __("Search");?></div>
10  <div class="clear_both"></div>
11 </div>
12 <?php /*----- ASPECTS -----*/ ?>
13
14 <?php /*----- FACETS -----*/ ?>
15 <div class="form_position_for_view" id="form_facet" >
16   <div ex:role="facet" style="height: 13em;width:150px; float:left;margin-right: 7px;font-size: 12px;" ex:exp
17   <div ex:role="facet" style="height: 13em;width:140px; float:left;margin-right: 7px;font-size: 12px;" ex:exp
18   <div ex:role="facet" style="height: 13em;width:80px; float:left;margin-right: 7px;font-size: 12px;" ex:expr
19   <div ex:role="facet" style="height: 13em;width:70px; float:left;margin-right: 7px;font-size: 12px;" ex:exp
20
21   <div ex:role="facet" style="height: 13em;width:80px; float:left;margin-right: 7px;font-size: 12px;" ex:sel
22   <div ex:role="facet" style="height: 13em;width:130px; float:left;margin-right: 7px;font-size: 12px;" ex:sel
23 </div>
24 <?php /* for count map */ ?>
25   <div ex:role="coder" id="count-coder"
26     ex:coderClass="SizeGradient"
27     ex:gradientPoints="0, 20; 100, 50">
28   </div>
29 </div>
30 <div id="utm-map1" ex:role="viewPanel">
31   <div class="map-lens" ex:role="lens" id="map_lens_id">
32     <span ex:content=".b_name"></span><br/>
33
34     <?php echo __('Date');?>: <span ex:content=".date"></span><br/>
35     <?php echo __('UTM square');?>: <span ex:content=".utm_sq"></span><br/>
36     <?php echo __('District');?>: <span ex:content=".district"></span><br/>
37     <?php echo __('Bird amount');?>: <span ex:content=".amount"></span><br/>
38     <?php echo __('Bird sex');?>: <span ex:content=".sex"></span><br/>
39     <?php echo __('Watch type');?>: <span ex:content=".character"></span><br/>
40     <?php echo __('Birdwatcher');?>: <span ex:content=".watcher"></span><br/>
41
42   </div>
43   <div ex:role="view"
44     ex:viewClass="Map"
45     ex:label="<?php echo __('Birdwatch map');?>"
46     ex:latlng=".utm_c"
47     ex:colorCoder="vote-for-colors"
48     ex:center="55.0934975,24.1512325"
49     ex:zoom="7"
50     ex:scaleControl="false"
51     ex:iconFit="none"
52     ex:shape="square"
53     ex:bubbleWidth="200"
54     ex:icon="concat('<?php echo "http://".$_SERVER['SERVER_NAME'].':8080/images/'?>', 'bird', '.png')"
55     ex:shapeWidth="20"
56     ex:shapeHeight="20">
57   </div>

```

paieškos laukas
(facet)

aspektų sąrašas (facet)

reikšmės priskirimas gradientui
(coder)

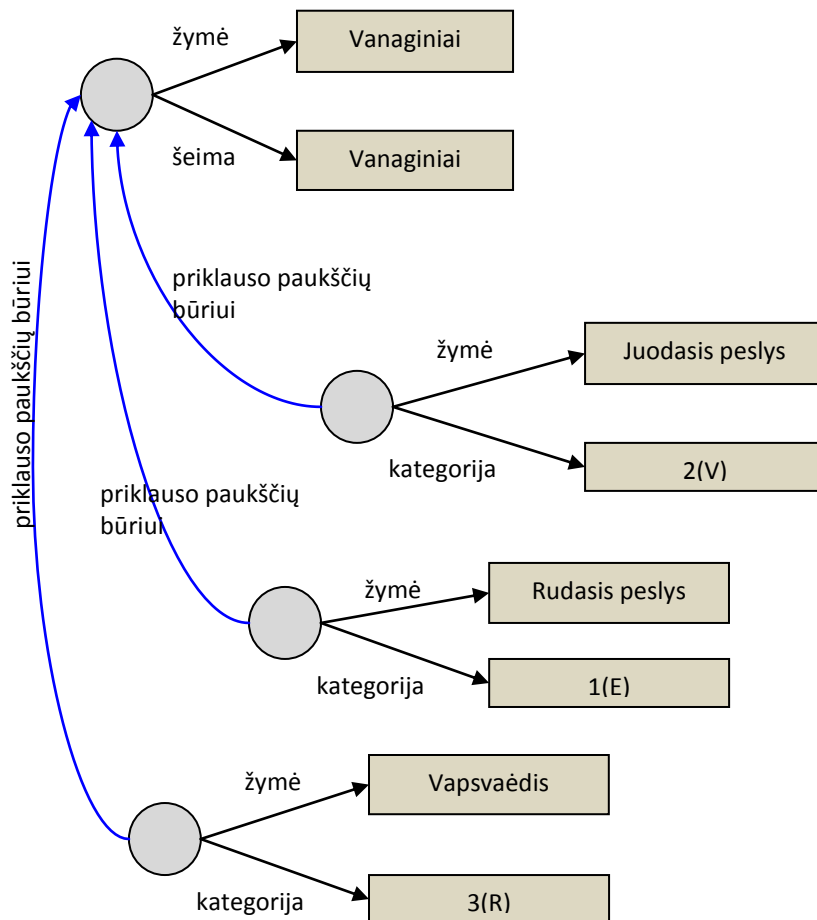
išvedamo
įrašo
duomenys
(lens)

duomenų
atvaizdavimo
formavimas
(view)

5 pav. HTML kalba aprašytos *Exhibit* vartotojo sąsajos pavyzdys

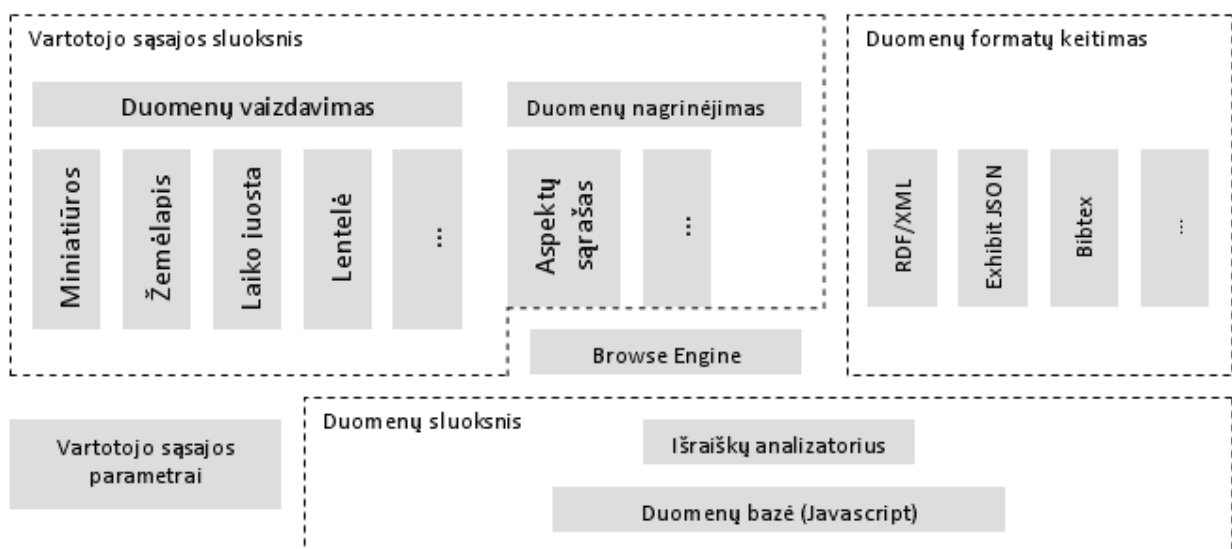
Svarbu atkreipti dėmesį į įrankio duomenų organizavimo modelį, kadangi nuo jo struktūros priklauso panaudojimo galimybės. Taigi, *Exhibit* duomenų modelis – orientuotas grafais, kurio mazgai – duomenų elementai, o lankai – elemento ypatybės. 6 pav. pavyzdyje pateikta sudėtingesnė duomenų struktūra, kurioje realizuoti ryšiai tarp duomenų elementų

(mėlyna linija), tačiau ryšių gali ir nebūti. Tokiu atveju, gaunamas elementų su viena ar daugiau ypatybių rinkinys.



6 pav. Exhibit duomenų modelio struktūra

Žvelgiant į *Exhibit* įrankio architektūrą 7 pav. pastebima, kad ji sudaryta iš dviejų pagrindinių sluoksnių: duomenų sluoksnio ir vartotojo sąsajos sluoksnio [Huynh, 2007].



7 pav. Exhibit architektūros modelis

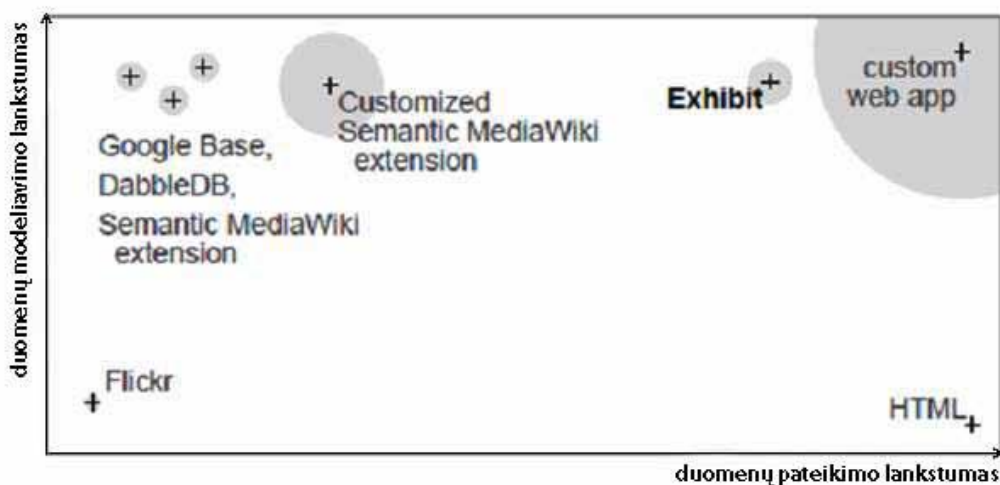
Duomenų sluoksnis susideda iš duomenų struktūros, kuri realizuota *JavaScript* kalba ir išraiškų analizatoriaus. Su šiuo sluoksniu susijęs duomenų formatų keitimo pasluoksnis.

Vartotojo sąsajos sluoksnis sudarytas iš tokių pagrindinių dalių: duomenų grafinio atvaizdavimo ir duomenų nagrinėjimo dalies, kurioje realizuojami įvairūs duomenų filtrai. Už šio sluoksnio korektišką veikimą ir atvaizdavimą atsakingi *vartotojo sąsajos parametrai* ir *Browse Engine* komponentai.

Tokio pobūdžio architektūros naudojimas *Exhibit* įrankiui suteikia esminių ypatybių:

1. Įrankio nereikia įdiegti, konfigūruoti ir programiniu būdu prižiūrėti;
2. *Exhibit* naudotojas nesunkiai gali plėsti įrankio funkcionalumą;
3. *Exhibit* naudojimui tinka esamos interneto naršyklės;
4. *Exhibit* panaudojimui nereikia turėti labai stiprių programavimo įgūdžių, pakanka išmanyti *HTML*;

8 paveikslėlyje pateikta *Exhibit* įrankio padėtis rinkoje, kai lyginamos duomenų valdymo ir jų vizualizavimo galimybės [Huynh et al., 2007].



8 pav. Exhibit duomenų modeliavimo ir pateikimo lankstumas kitų įrankių atžvilgiu

Šioje diagramoje duomenų atvaizdavimo technologijos lyginamos dviem aspektais: duomenų modeliavimo ir duomenų pateikimo lankstumu. Taigi *Flickr* įrankiui, bei įvairiems internetinių (angl. *web*) albumų generatoriams yra būtinos griežtos duomenų modeliavimo sąlygos ir kartu pasiūlomas nedidelis duomenų atvaizdavimo lankstumas. *DabbleDB*, *Google Base* ir *Semantic MediaWiki* leidžia susikurti lanksčius duomenų modelius, tačiau siūlo nedideles galimybes atvaizduojant duomenis. Kai tik norima kuriam nors įrankiui suteikti didesnes duomenų atvaizdavimo galimybes, iškart prireikia įrankio įdiegimo, duomenų bazės administravimo ar nemažų programavimo įgūdžių [Huynh et al., 2007].

2.3.3. Google API analizė

Prieš penketą metų pasirodžiusi *Google Maps* paslauga sukėlė tikrą perversmą skaitmeninių žemėlapių ir kartografijos srityje. Naudojimas skaitmeniniu žemėlapiu tapo laisvai prieinamas visiems piliečiams. Didžiulis populiarumas davė ankstiną šio įrankio tobulėjimui ir šiuo metu žemėlapių panaudojimas asmeniniuose ar komerciniuose tinklalapiuose nieko nebestebina.

Vienas iš šio įrankio privalumų, kad duomenys ir programos kodas apdorojami naršyklėje, tai yra ne serveryje, nes *Google Maps API*, kaip ir *Google visualization API* yra *JavaScript* biblioteka [Anslow et al., 2009]. Viena iš naudingų ypatybių, kad vartotojas gali spragtelėti ant žemėlapių žymės ir suaktyvinti informacijos, susijusios su konkrečiu tašku, pasirodymą. Informacija gali apimti kelias eilutes arba pasirodyti, kaip išsami išsklotinė su nuotraukomis, nuorodomis ir pan. Tokia informacijos pateikimo forma leidžia suteikti tinklapiui interaktyvumą. Be to interaktyvumas dar labiau padidinamas, leidžiant vartotojui manipuliuoti žemėlapiu tai yra jį paslinkti, sumažinti, padidinti. Ši paslauga leidžia ne tik atvaizduoti objektą konkrečioje dvimatės erdvės vietoje, bet ir paskaičiuoti atstumus tarp žymių, atlikti paiešką, susikurti specifines žemėlapių žymes ir pan.

Šiuo metu *Google Maps API* tobulinimas susijęs su žemėlapių užkrovimo naršyklėje laiko trumpinimu bei objektų žymių atvaizdavimo optimizavimu. Pastarąją problemą išprovokavo situacija, kai reikia atvaizduoti didelį kiekį žymių, pvz. 1000 ir daugiau. Dėl per didelio kiekio ne tik sulėtėja darbas su žemėlapiu, bet ir prarandamas atvaizduojamos informacijos aiškumas.

Nagrinėjant platesnes *Google* galimybes reiktų atkreipti dėmesį *Google Visualization API*. Šio įrankio gali prisireikti, kai standartiniai *Exhibit* duomenų vizualizavimo sprendimai netenkina. Taigi, *Google Visualization API* siūlo keliolika duomenų atvaizdavimo būdų, kurie nesunkiai pritaikomi.

2.4. Vartotojų analizė

2.4.1. Vartotojų aibė, tipai ir savybės

Kuriama informacinė sistema orientuota ne tik į ornitologijos srities specialistus, bet ir į plačiąją visuomenę. Tokiu būdu populiarinant Lietuvos gamtosauuginę veiklą ir pritraukiant daugiau paukščių stebėjimams neabejingų asmenų. Iš 2 pav. pateikto veiklos esybių modelio, preliminariai galima išskirti keturis vartotojų lygius: sistemos lankytojas, registruotas dalyvis, Lietuvos ornitologų draugijos narys ir administratorius. Šiuo atveju, mums svarbiausi yra

pirmieji trys lygiai, nes sistemos kūrimas yra orientuotas būtent į juos. Kiekvienas iš šių vartotojų tipų pasižymi tam tikromis savybėmis.

Informacinės sistemos realizavimo eigoje nereiktų pamiršti atkreipti dėmesį į būsimų vartotojų kompiuterines žinias, ornitologijos srities išmanymą tai yra terminų, sutrumpinimų, kategorijų pažymėjimų suprantamumą. Be jau išvardintų vartotojų grupių reiktų nepamiršti įvairių gamtosauginių institucijų, kurioms taip pat galėtų būti naudinga susisteminta Lietuvos paukščių monitoringo informacija. Todėl reiktų išlaikyti tam tikrą srities profesionalumo bei duomenų korektiškumo palaikymo lygį.

2.4.2. Vartotojų tikslai ir problemos

Vartotojai gali turėti keletą tikslų pvz.: dalyvauti paukščių registravimo procese, peržiūrėti stebėjimo duomenis, bendrauti su paukščių stebėtojais. Vartotojų problemų atsiradimą gali įtakoti įvairūs veiksniai, kaip pvz.: ornitologijos srities žynių stygius, kompiuterinės žinios, amžius, kalbos barjeras, negalia ir taip toliau. Viso to pasėkoje sektų padidėjusi netikslių duomenų įvedimo tikimybė. Taigi, kuriant sistemą į tai reiktų atsižvelgti, nes kaip minėta ankščiau, stengiamasi į paukščių stebėjimus įtraukti ne tik ornitologijos specialistus, bet ir visuomenę.

2.5. Esamų sprendimų analizė

2.5.1. Duomenų saugojimo formatai

2.5.1.1. JSON formatas

*JSON*⁶ apibūdinamas kaip nesudėtingas duomenų apskeitimo formatas, kuris išpopuliarėjo ir tapo alternatyva populiariam ir daugelyje sričių naudojamam *XML* formatui [Alexander, K., 2008]. Be to, tai tekstu grįstas (angl. *text-based*), nuo kalbos nepriklausomas (angl. *language-independant*) duomenų apskeitimo formatas, kurį žmogus gali lengvai suprasti ir koreguoti. *JSON* išsivystė iš *JavaScript* programavimo kalbos. Šis formatas specifiukuotas RFC-4627 aprašyme, kurį parengė vyriausias *JavaScript* architektas D. Crockford. Išsami specifikacija pateikta EMCA262 standarte.

Naudojant *JSON* formatą, duomenų objektai apibrėžiami naudojant tris parametrus: tipus (angl. *types*), ypatybes (angl. *properties*) ir elementus (angl. *items*). Pirmieji du parametrai skirti aprašyti duomenų struktūrai ir jie duomenų faile nėra privalomi. Paskutinis parametras yra privalomas, nes aprašo pačius duomenis.

⁶ JSON (angl. *JavaScript Object Notation*)

Atskirą parametro *items* įrašo rinkinį galima interpretuoti, kaip vienos duomenų bazės lentelės ar konkrečios lentelės atributo įrašą tekstiniame dokumente. Kiekvienas elementų rinkinys susideda iš grupės skirtingų objekto ypatybių, kurioms suteikiamos tam tikros reikšmės 9 pav. Esminis dalykas, kad aprašant elementų rinkinį įrašas *label* yra privalomas. Taip pat būtinas rinkinio unikalus identifikatorius, kad nesidubliuotų įrašai. Identifikatoriumi gali būti ypatybė *ID*, jei jos nėra, tuomet identifikatoriumi laikoma ypatybė *label*.

```
{
  items: [
    {
      "label": "Nendrine lingė",
      "Buryis": "Vanaginiai",
      "Seima": "Vanaginiai",
      "Kiekis": "3",
      "Data": "2009.06.02",
      "KORD": "26.21114,55.29875"
    }
  ]
}
```

9 pav. *Exhibit JSON* duomenų failo įrašo pavyzdys

Kaip paminėta anksčiau, parametrai *types* ir *properties* nėra privalomi, tačiau, jie leidžia apibrėžti aprašomus elementus ir jų ypatybes. Elementų rinkiniui 9 pav. pritaikius ypatybę *type*, galima priskirti tipus, pvz.: daugiskaitos reikšmę. Naudojant parametą *properties* elementų rinkinio įrašams galima nustatyti vieną iš keturių galimų įrašo tipo reikšmių tai yra suteikti skaičiaus, *url* (angl. *Uniform Resource Locator*), datos ar tekstinę reikšmę 10 pav.

```

{
  types: {
    'Paukstis':{
      pluralLabel: 'Pauksčiai'
    },
  },
  properties:{
    "label": {valueType: "item"},
    "Data": {valueType: "date"},
    "Kiekis":{valueType: "number"}
  },
  items: [
    {
      "label": "Nendrine linge",
      "Buryys": "Vanaginiiai",
      "Seima": "Vanaginiiai",
      "Kiekis": "3",
      "Data": "2009.06.02",
      "KORD": "26.21114,55.29875",
      "type": "Paukstis"
    }
  ]
}

```

10 pav. *Exhibit* JSON duomenų failo pavyzdys, kai aprašomi duomenys ir jų struktūra

Užsikraudamas *JSON* formatu aprašytų duomenų failą, *Exhibit* įrankis realizuoja aspektais (angl. *facet*) paremtą duomenų nagrinėjimą. Šis duomenų nagrinėjimo būdas pasižymi vartotojui patogia (angl. *user-friendly*) informacijos pateikimo forma. Ji labai patogi, kai reikia nagrinėti plačiai analizuojamą sritį aprašančius duomenis [Hildebrand et al., 2006]. Aspektais (angl. *facet*) paremtas duomenų nagrinėjimas išsiskiria tuo, kad vartotojas duomenų filtravimą gali pradėti nuo bet kurio aspekto ir tęsti duomenų filtravimą naudodamas kitus aspektus, pvz. informacija apie paukščius gali būti filtruojama pagal užregistravimo teritoriją, paukščių šeimą ir paukščio rūšį. Vartotojas filtravimą gali pradėti nuo bet kurio aspekto pvz. nuo paukščių šeimos ir tęsti duomenų filtravimą pagal teritoriją ir paukščio rūšį. Eiliškumo seka priklauso nuo vartotojo, kadangi jis sprendžia, koks pradinis kriterijus jam svarbiausias tai yra teritorija, paukščių rūšis ar paukščių šeima. Apie šį aspektų (angl. *facet*) pranašumą analizuojant duomenis rašo [Huynh, D.F., 2007]. Norint paspartinti duomenų filtravimą, galima atlikti duomenų paiešką pagal raktinį žodį, tai yra į tam skirtą lauką įvedus konkretų žodį iš ieškomos informacijos aibės. Tokiu būdu išrenkami informacijos rinkiniai, kuriuose rastas paieškos žodis. Aspektų (angl. *facet*) turinys gali nuolatos pildytis, jei duomenų struktūra papildoma nauju įrašu, tai yra aspektų turinys gali priklausyti nuo duomenų kiekio, pvz. jei įvedamas paukščio stebėjimas teritorijoje, kuri nebuvo paminėta ankstesniuose įrašuose, tuomet automatiškai įvedus stebėjimą, naujos teritorijos pavadinimas atsiranda filtro, skirto filtruoti duomenis pagal teritoriją, turinyje.

2.5.1.2. Reliacinės duomenų bazės

Reliacinės duomenų bazės sumanytoju laikomas IBM darbuotojas E. Codd. Jis 1970 metais apibrėžė ir pristatė savo idėją. Šiuo metu reliacinę duomenų bazę galima apibrėžti, kaip grupę duomenų elementų, kurių rinkiniai patalpinti formaliai apibrėžtose lentelėse. Naudojantis struktūrizuota užklausų kalba (*SQL*), vartotojas gali pasiekti duomenų bazėje kaupiamus duomenis, juos trinti, įterpti, modifikuoti, peržiūrėti.

Kaip jau minėta, reliacinėje duomenų bazėje duomenys saugomi dvimatėse lentelėse. Duomenų lentelės eilutės vadinamos kortežais, o stulpeliai – atributais. Kiekviena dvimatė lentelė turi turėti unikalų įrašo identifikatorių, kuris gali būti ir sudėtinis, tai yra susidedantis iš kelių atributų reikšmių. Duomenų lentelės pasižymi tokiais savybėmis: visi įrašai turi tą pačią struktūrą; visuose įrašuose toks pat skaičius laukų; lentelėje negali būti identiškų įrašų; įrašų ir laukų išdėstymo tvarka lentelėje neturi reikšmės; kiekviena lentelė turi turėti unikalų vardą.

Reliacinės duomenų bazės pasižymi tuo, kad duomenų bazėje esančios lentelės susiejamos tarpusavyje, taip suformuojant reliacinį modelį. Šiam modeliui būdingas tam tikras duomenų pertekliškumas, galintis atsirasti tiek dėl pirminio rakto suformavimo, tiek dėl ryšių tarp lentelių formavimo. Iš esmės tiek reliacinėms, tiek kitokio pobūdžio duomenų bazėms būdingas duomenų integruotumas, nepertekliškumas (siekiama vengti duomenų dubliavimosi), nepriklausomumas (duomenų apdorojimo programos nesikeičia, kai modifikuojami duomenys).

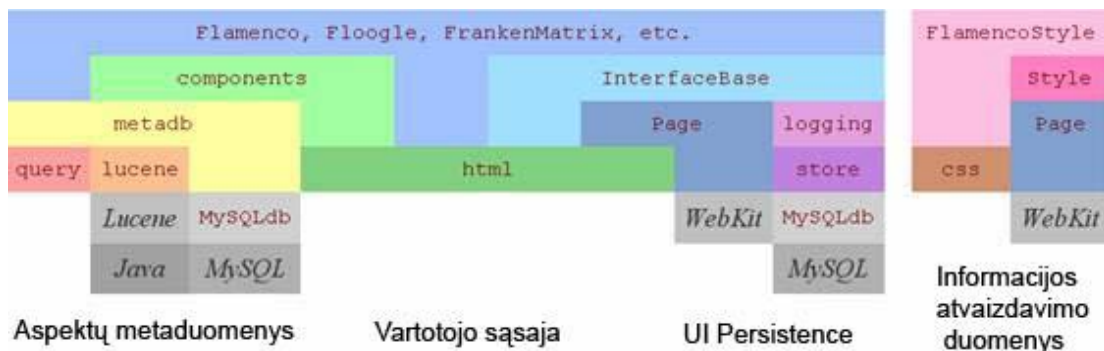
2.5.2. Kiti duomenų atvaizdavimo ir filtravimo sprendimai

2.5.2.1. Flamenco

Flamenco – tai internetiniu pagrindu (*angl. web-based*) sukurta sąsaja, skirta patogiai ir efektyviai nagrinėti didelės apimties duomenis. Šis įrankis paremtas aspektų (*angl. facet*) panaudojimu. Asmuo norėdamas naudotis *Flamenco* įrankiu, privalo atsisiųsti ir jį įsidiegti. Be to, įrankio veikimui būtina *UNIX* ar *MACOS* operacinė sistema su *Apache HTTP Server* ir duomenų bazės valdymo sistema *MySQL*. Šis įrankis pradėtas kurti nuo 2002 metų ir pirmoji šio įrankio versija pasirodė 2006 metais. Įrankį pradėjo kurti septyni studentai vadovaujami *UC (University of California) Berkeley* mokyklos profesoriaus Marti Hearst.

Flamenco įrankis sukurtas naudojant *Python* programavimo kalbą ir naudoja *Webware* programų paketą. Įrankio architektūra pateikta 11 pav. Pilka spalva išskirti komponentai, kurie nėra nagrinėjamo įrankio sudėtinė dalis. *Flamenco* įrankio branduolys yra *metadb* modulis, kuris įgyvendina aspektais (*angl. facet*) paremtą duomenų nagrinėjimą ir

informacijos paieškos funkcionalumą, kuris realizuojamas naudojant *Lucene* programą, kurią vartotojas turi pats įdiegti, taip praturtindamas sistemos funkcionalumą. *HTML* kodui generuoti naudojamas *html* modulis. Vartotojo sąsajos formavimui skirtas *components* modulis, kuris naudojamas duomenų bazėje saugomą informaciją, suformuoja dalį *Flamenco* įrankio vartotojo sąsajos. Likusias dalis apjungia *Flamenco* modulis taip suformuodamas pilną vartotojo sąsają. Vartotojo sąsajos stiliaus parametrus dinamiškai sugeneruoja *FlamencoStyle* modulis. *WebKit* (*Webware* sudėtinė dalis) modulis atsakingas už internetinio puslapio ir stiliaus apjungimą ir realizaciją.



11 pav. *Flamenco* įrankio architektūra

Norint, kad įrankis vykdytų duomenų filtravimą, rūšiavimą - reikia sukurti meta-duomenų failus, kurie aprašytų, kurie duomenys bus naudojami formuojant filtrus, o kurie bus naudojami informacijos atvaizdavimui. Meta-duomenų failai aprašomi *TSV*⁷ formatu. Iš viso turi būti sukurti 7 skirtingos paskirties duomenų failai, kurie pateikia: atributų sąrašą, aspektų (angl. *facet*) sąrašą, duomenų elementus, konkrečių aspektų struktūrą, aspektų ryšį su duomenų elementais, informaciją, kaip rūšiuoti duomenis, bei informaciją, leidžiančią realizuoti duomenų paiešką pagal raktinį žodį.

2.5.2.2. mSpace

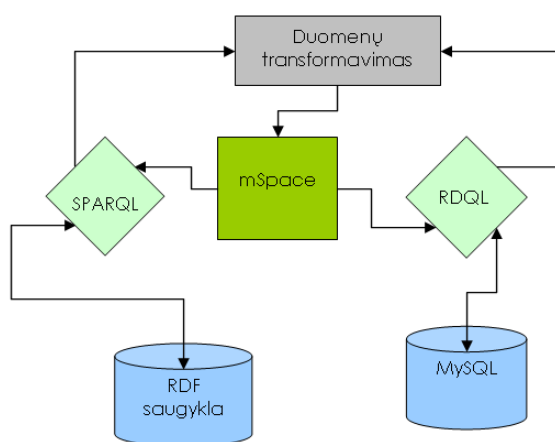
mSpace – tai įrankis skirtas didelės apimties informacijos analizavimui tai yra filtravimui, grupavimui, paieškai. Įrankiui reikalinga *MACOS* operacinė sistema su *Apache 2.2.3*, *PHP 5.2.1* ir *MySQL 5.0.38* programomis. Duomenys turi būti pateikiami struktūrizuota forma tai yra *RDF* formatu. Gautieji tripletai turi būti saugomi specialioje duomenų bazėje. Sistemos kūrimo pradininkai Russell A., Harris C., Smith D. A., Owens A. Įrankis plėtojamas nuo 2001 metų. Yra dvi versijos: laisvai platinama ir komercinė.

Šiame įrankyje, kaip paminėta [Smith et al. 2009] literatūroje, turi būti apibrėžta duomenų struktūros schema, pagal kurią suformuojami aspektais paremti duomenų filtrai. *mSpace* architektūroje naudojama sudėtinė (angl. *multicolumn*) filtravimo struktūra [Smith et

⁷ TSV - failų formatas, kuris skirtas tekstiniu formatu aprašyti duomenų lenteles

al., 2007]. Vartotojui pasirinkus, kurio nors filtro reikšmę, dešiniau esantys aspektų rinkiniai taip pat pasikeičia, tokiu būdu neleidžiant pasirinkti filtro reikšmės, kuri pateiktų nulinį rezultatą. Be to, patogiai vartotojo sąsaja leidžia lengvai manipuluoti filtrais, tai yra šalinti esamus ir sukurti naujus filtras, keisti jų hierarchinę struktūrą, tai yra išdėstymo eiliškumą vartotojo sąsajoje [schraefel et al., 2005 13].

mSpace architektūra sukurta taip, kad galėtų apjungti informaciją iš skirtingų šaltinių, tokiu būdu išplečiant įrankio funkcionalumą. Taip pat šio įrankio struktūra pasižymi tuo, kad yra galimybė pakoreguoti architektūrą taip, kad būtų naudojami duomenys ne iš *RDF* duomenis saugomų ir *SPARQL* užklausas naudojančių duomenų saugyklų, bet iš reliacinės duomenų bazės (*MySQL*) įdiegus 3store⁸ papildymą ir naudojant *RDQL* užklausas 12 pav.

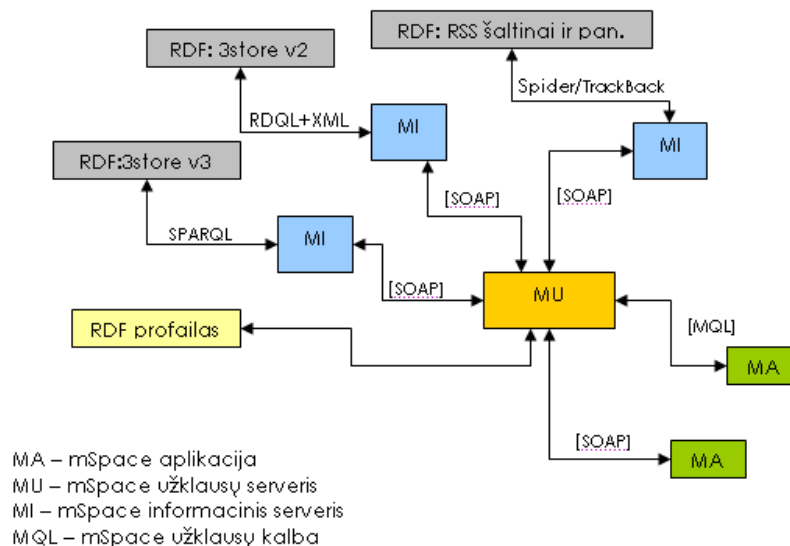


12 pav. Principinė *mSpace* įrankio struktūros schema

Kaip rašoma [schraefel et al., 2005 12], sistemos veikimas susideda iš tokių veiksmų grandinės: vartotojas pasirenka filtrų reikšmės; iš pasirinktų reikšmių suformuojama užklausa; užklausa siunčiamą į *RDF* tripletų saugyklą; suformuoti duomenys, paprastai *XML* formatu, gražinami internetiniu pagrindu (angl. *web-based*) paremtai *API*, kuri suformuoja galutinį vaizdą, generuodama *XHTML* ir *CSS*.

Kaip buvo minėta ankščiau, *mSpace* įrankis gali būti skirtas nagrinėti paskirstytiems duomenis, tai yra duomenims, kurie pasklidę po skirtingas saugyklas. Principinė aptartosios *mSpace* architektūros schema pateikta 13 pav.

⁸ 3store – tai C kalbos bibliotekų branduolys naudojamas *MySQL* DBVS, kad saugoti *RDF* duomenis



13 pav. *mSpace* architektūra naudojant MA, MI ir MU sluoksnius

Šioje schemoje pateikta trijų lygių architektūra, kuri susideda iš *mSpace* aplikacijos, *mSpace* užklausų ir *mSpace* informacijos lygio. MI^9 sluoksnis suteikia galimybę turėti keliolika duomenų šaltinių. Architektūra sukurta taip, kad būtų galimybė naudojantis *RDQL* ar *SPARQL* užklausas išskirti reikiamus duomenis. Vienas iš esminių architektūros sluoksnių yra *MU*, kuris nustato, kuri *MI* naudoti norint gauti reikiamus duomenis. Ryšiui tarp *MA* ir *MU* palaikyti naudojamas *SOAP*¹⁰ ir *HTTP* protokolas. Šių protokolų panaudojimas suteikia galimybę visiems serveriams būti pasiekiamiems per internetą ir išlaikyti operacinį suderinamumą. Paprastumo dėlei *MI* galima interpretuoti kaip atskirą tripletų saugyklą, kadangi atskiras *MI* palaiko individualų *HTTP-RDQL* ryšį tarp *RDF* duomenų.

2.5.2.3. *jSpace*

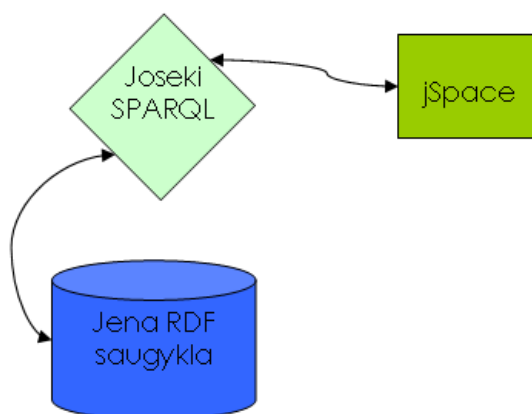
Šio įrankio atsiradimą įtakojo *NASA* užsakymas, kadangi, kaip teigiama [Grove, 2008], jų kaupiami duomenys pasklidę įvairiose duomenų bazėse, todėl reikėjo sistemos kuri apjungtų visus duomenis su galimybe juos nagrinėti tai yra filtruoti, peržiūrėti ir pan. Taigi, atsižvelgus į reikalavimus *Clark & Parsia LLC* sukūrė *jSpace* įrankį. Pradžioje šis įrankis buvo *mSpace* įrankio kopija, tačiau evoliucionavo į labiau vartotojo sąsajos atžvilgiu išplėstą įrankį, skirtą aspektų (angl. *facet*) pagalba naršyti duomenis. Įrankio vartotojo sąsaja pasižymi tokiomis savybėmis, kaip galimybė laisvai keisti filtrų rinkinių padėtį (angl. *drag and drop*), suskleisti filtrų rinkinius, duomenis susieti su žemėlapiu.

jSpace įrankio veikimas paremtas *SPARQL* užklausų ryšiu su *RDF* duomenų saugykla. Be to, norint, kad įrankis veiktų reikia apibrėžti aspektus (angl. *facet*), kurie bus naudojami

⁹ Santrumpų reikšmės pateiktos 13 pav.

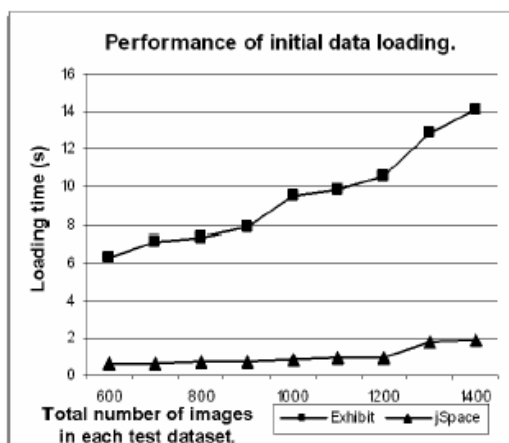
¹⁰ *SOAP* (angl. *Simple Object Access Protocol*)

informacijos filtravimui ir atvaizdavimui [Zhao et al., 2008]. Principinė *jSpace* architektūros schema pateikta 14 pav.

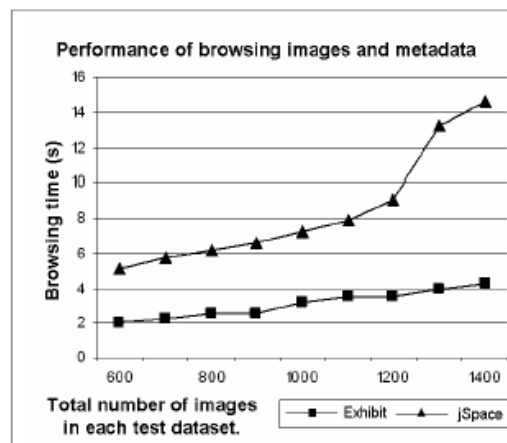


14 pav. Principinė *jSpace* įrankio schema

Labai panaši *jSpace* įrankio architektūra buvo aprašyta [Zhao et al., 2008] straipsnyje. Pastarojoje literatūroje atlikto tyrimo metu buvo lyginti dviejų įrankių tai yra *jSpace* ir *Exhibit* galimybės atvaizduoti ir filtruoti mokslinės informacijos duomenis (nuotraukas). Gauto tyrimo rezultatai pateikti 15 pav.



15a pav. *jSpace* ir *Exhibit* duomenų užkrovimo laikas



15b pav. *jSpace* ir *Exhibit* duomenų filtravimo laikas

Tiek 15a pav., tiek 15b pav. pavaizduota dviejų duomenų filtravimo įrankių veikimo greičio priklausomybė nuo nagrinėjamų duomenų kiekio. 15a pav. palygintas *Exhibit* ir *jSpace* įrankių duomenų užkrovimo laikai. *Exhibit* atveju užkrovimo laikas skaičiuojamas įvertinant kiek laiko reikia JSON formate saugomiems duomenis užkrauti į *Exhibit* duomenų struktūrą. *jSpace* atveju – kiek laiko užtrunka nusiųsti SPARQL užklausą ir gauti atsakymą. 15b pav. lyginami įrankių duomenų filtravimo sparta, tai yra kiek laiko užtrunka duomenų filtravimo operacijos įvykdymas pasirinkus vienokią ar kitokią filtro reikšmę. Iš gautų rezultatų galima spręsti, kad *Exhibit* silpnoji vieta – duomenų įkėlimas į vidinę duomenų struktūrą, o stiprioji – įkeltų duomenų apdorojimas.

2.6. Analizės apibendrinimas

Esamų sprendimų analizės metu *Exhibit* įrankį siekta palyginti dvejopai. Pirmasis analizės etapas, buvo skirtas išanalizuoti *Exhibit* įrankio privalumus ir trūkumus susijusius su duomenų saugojimo formatu. Antrasis etapas susidėjo iš panašios paskirties įrankių analizės ir tų įrankių stipriųjų ir silpnųjų pusių išskirimo.

Exhibit įrankio duomenų šaltinis turi būti tekstiniame ar kitokio pobūdžio faile saugomi duomenys. Tai byloja, kad įrankis nėra skirtas vienu metu apdoroti didelio kiekio duomenų, nes paprasčiausiai duomenų užkrovimas ir tolesnis jų naudojimas užimtų nemažai laiko. Įrankio veikimo laiką įtakoja ne tik duomenų įkėlimas, bet ir aspektų (angl. *facet*), duomenų vaizdavimo parametrų, duomenų filtravimo organizavimas. 4 priede pateikta informacija, apie *Exhibit* įrankio užkrovimo laiko priklausomybę nuo duomenų kiekio, naršyklės ir kompiuterio parametrų, kuri buvo pateikta [Huynh, 2007] literatūroje.

Analizės metu paaiškėjo, kad kiekvieną *JSON* faile saugomų elementų rinkinį galima interpretuoti, kaip atskiros duomenų bazės lentelės įrašų kortežą, su tam tikromis atributų reikšmėmis, kas leidžia manyti, kad galima organizuoti visų ar dalies duomenų perkėlimą į *JSON* failą, siekiant organizuoti duomenų filtravimo ir rezultatų atvaizdavimo funkcionalumą. Taigi, *JSON* failas galėtų būti naudojamas, ne kaip sistemoje kaupiamų duomenų saugykla, bet kaip duomenų, kurie reikalingi analizei ir peržiūrai rinkinys. *Exhibit* įrankis pasižymi ir tuo, kad gali apjungti kelis atskirus *JSON* failus. Taigi, teoriškai atsiranda galimybė duomenis užkrauti nedidelėmis porcijomis, taip pagreitinant įrankio veikimą. Be to, organizuojant duomenų iš reliacinės bazės generavimą į *JSON* formatą, būtų galima išspręsti informacijos prieinamumo problemą, kuri atsirastų realizuojant informacinę sistemą su kelių lygių vartotojais. Taip pat atsiranda galimybė pašalinti duomenų įterpimo, koregavimo, šalinimo dinamiškumo stoką, kuri atsiranda, kai visi nagrinėjami duomenys saugomi tik *JSON* formatu.

Antroji analizės dalis susidėjo iš trijų įrankių, skirtų duomenų filtravimui ir atvaizdavimui analizės. Šie įrankiai, kaip ir *Exhibit* naudoja aspektus (angl. *facet*) organizuojant duomenų grupavimą ir filtravimą. Didžiausias skirtumas, nuo pastarojo įrankio, kad nagrinėjamoji grupė įrankių duomenis pasiima ne iš tekstinio failo, bet iš duomenų bazės tai yra naudojant SPASQL, RDSQL ar SQL užklausas. Be to, šie įrankiai skirti nagrinėti didelius kiekius duomenų. Tačiau, be šio privalumo šie įrankiai turi keliolika esminių trūkumų lyginant su *Exhibit*. Esminis trūkumas, kad šie įrankiai turi būti įdiegti, be to tai turi būti atlikta tam tikrose operacinių sistemų aplinkose su tam tikrų papildomų programų įdiegimu. Tuo tarpu *Exhibit* nereikalauja įdiegimo, o kad šį įrankį naudoti užtenka išmanyti *HTML* kalbą. Be to, pastarasis įrankis išsiskiria duomenų atvaizdavimo funkcionalumu, nes

yra galimybė duomenis atvaizduoti diagramomis, žemėlapyje, laiko juostoje, lentelės pavidalu. Panašios paskirties įrankių analizės rezultatai pateikti 2 lentelėje. Pliuso ženklas reiškia, kad einamasis kriterijus yra įrankyje, minusas, kad jo nėra.

2 lentelė *Exhibit* įrankio palyginimas su kitais panašios paskirties įrankiais

Kriterijus \ Įrankis	<i>Exhibit</i>	<i>Flamenco</i>	<i>mSpace</i>	<i>jSpace</i>
Nebūtinai diegimas į operacinę sistemą	+	-	-	-
Būtinai naudojamų duomenų apibrėžimas	+	+	+	+
Realizuotas įvairiapusis duomenų vizualizavimas	+	-	-	+
Galimybė padidinti įrankio funkcionalumą	+	+	+	-
Darbas su dideliu kiekiu duomenų	-	+	+	+
Galimybė pačiam programuotojui formuoti vartotojo sąsają, laukų išdėstymą lange, stilių	+	-	-	-
Galimybė įrankį tiesiogiai naudoti formuojant tinklapį	+	-	-	-
Galimybė vartotojui manipuluoti vartotojo sąsaja tai yra pašalinti, įsikelti naujus duomenų filtrus	-	-	+	+

2.7. Siekiamas sprendimas

Darbo metu siekiama praplėsti pasirinkto įrankio funkcionalumą taip, kad atsirastų galimybė duomenų bazėje saugomus duomenis analizuoti įvairiais pjūviais. Šis įrankis bus panaudotas realizuojant Lietuvos paukščių registravimo sistemą, kuri pasižymėtų plačiomis duomenų atvaizdavimo, duomenų filtravimo, rūšiavimo, grupavimo galimybėmis, tokiu būdu palengvinant ir supaprastinant duomenų nagrinėjimą. Taigi, realizuojama sistema turi tenkinti tokius reikalavimus:

1. Interaktyvumas;
2. Įvairiapusis duomenų vizualizavimas;
3. Galimybė dirbti su dideliu kiekiu duomenų;

2.8. Analizės išvados

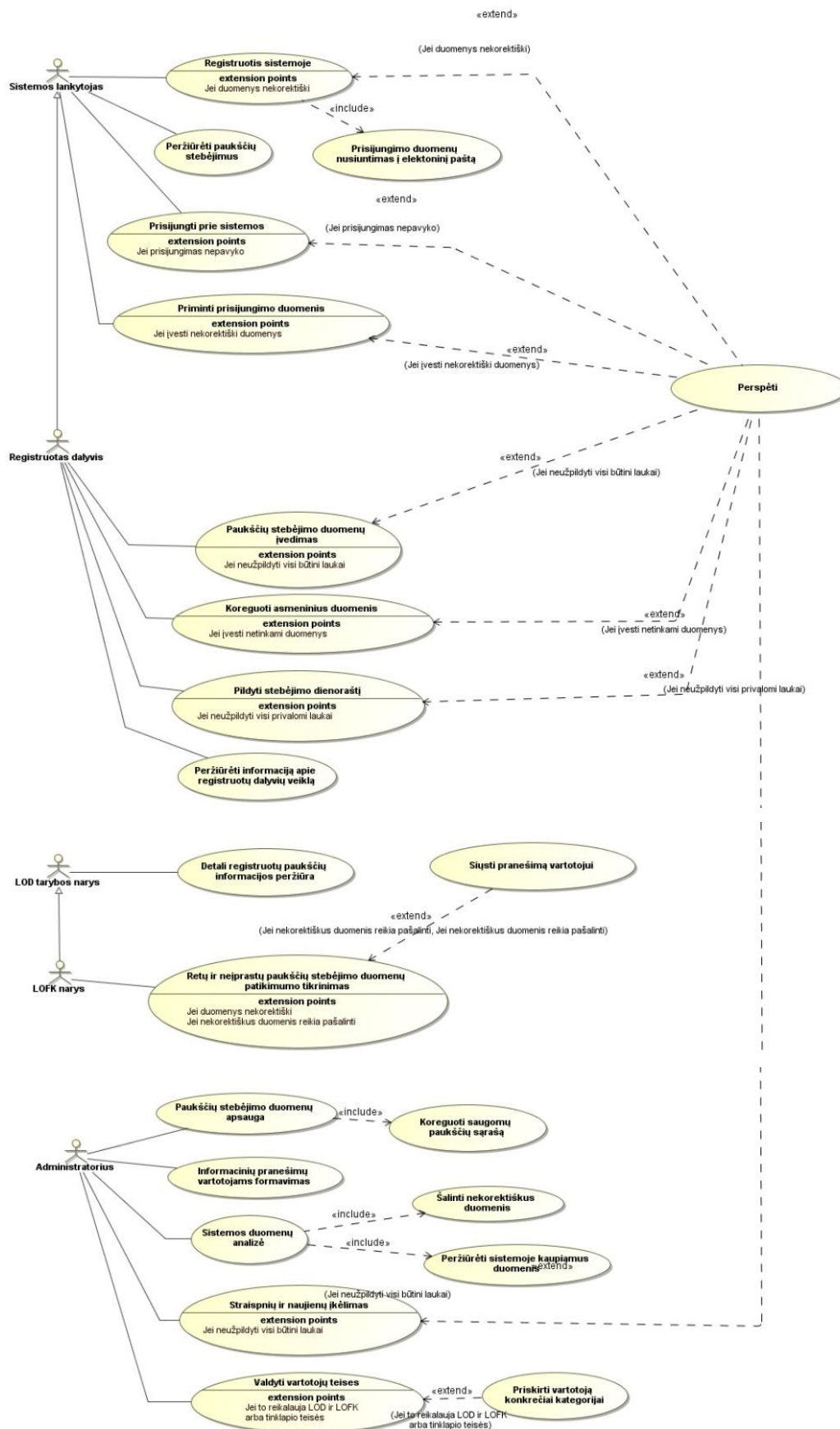
1. Lyginant su panašios paskirties įrankiais nustatyta, kad *Exhibit* įrankio panaudojimas pagreitintų ir supaprastintų duomenų filtravimo ir vizualizavimo realizavimą bei leistų sukurti vartotojui priimtinesnį ir patogesnį duomenų nagrinėjimą.
2. *Exhibit* įrankio panaudojimo trūkumai: neužtikrinamas duomenų privatumas, duomenys saugomi statinėje formoje, ilgai formuojama vidinė duomenų struktūra.
3. *Exhibit* įrankio siūlomus privalumus galima bandyti susieti su duomenų bazės siūlomais pranašumais.

3. Informacinės sistemos reikalavimų specifikacija ir analizė

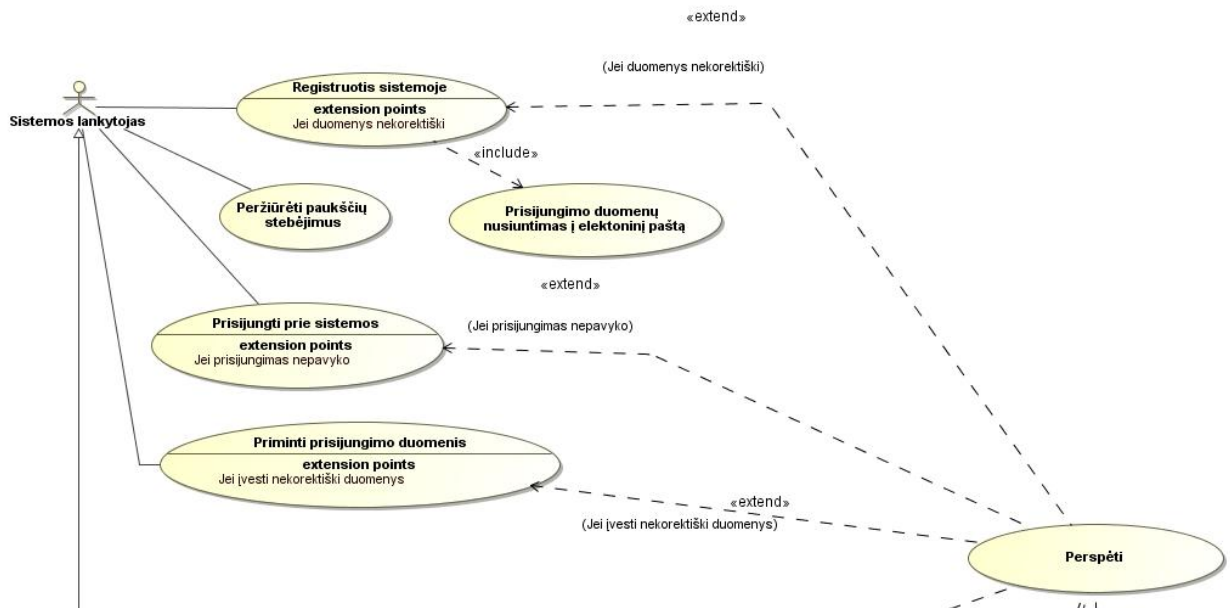
3.1. Reikalavimų specifikacija

3.1.1. Panaudojimo atvejai

Analizuojant sistemai keliamus funkcinius reikalavimus buvo išskirtos penkios vartotojų grupės, kurioms būdingas tam tikras leistinių vykdyti funkcijų (veiksmų), bei apribojimų sąrašas 16 pav. Iš 17 pav. pateiktos neprisijungusio aktorius galimų veiklų aibės matoma, kad apsilankęs vartotojas gali peržiūrėti naujienų skiltį, registruotis arba prisijungti prie sistemos. Šio lygio vartotojas turi galimybę peržiūrėti paukščių stebėjimus, tačiau jam pateikiama informacija yra ribota tiek duomenų detalumu, tiek informacijos kiekiu.

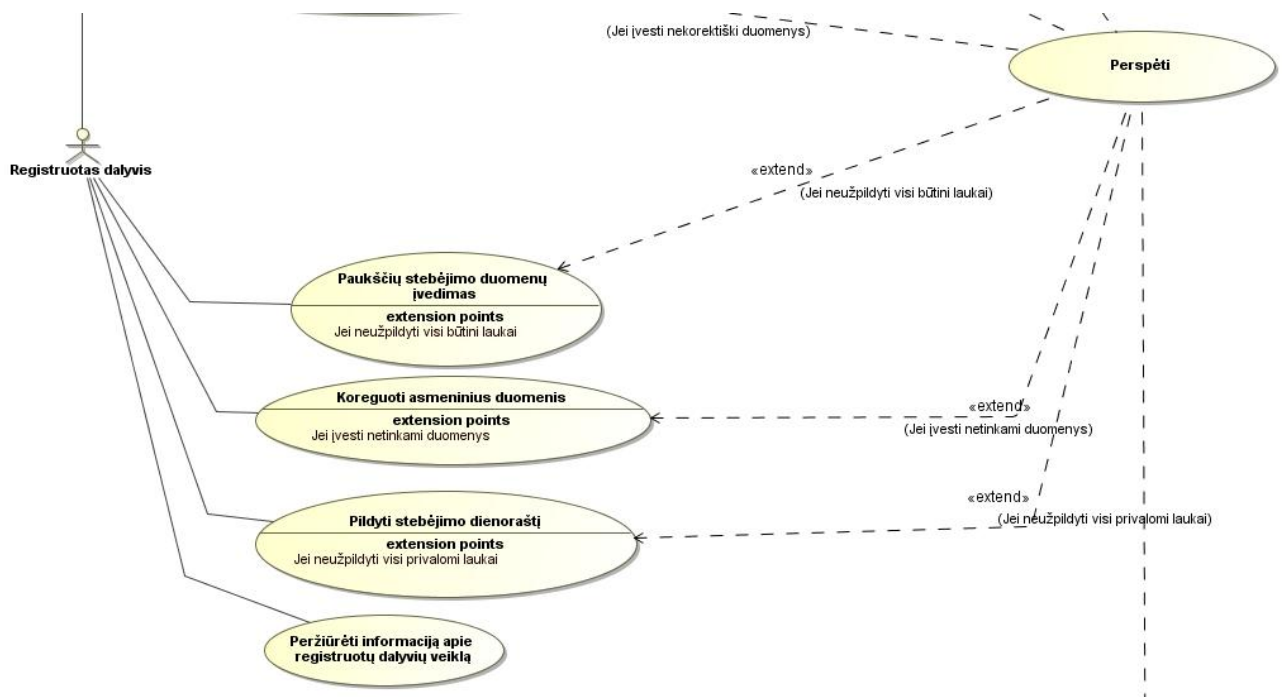


16 pav. Struktūrinis informacinės sistemos panaudos atvejų diagramos vaizdas



17 pav. Panaudos atvejų diagrama nusakanti neregistruoto ar neprisijungusio vartotojo galimybes

Antroji vartotojų grupė – registruoti sistemos vartotojai. Jų galimybės labai stipriai išsiplečia lyginant su pastarąja vartotojų grupe 18 pav. Atsiranda galimybė įvesti stebėjimą, peržiūrėti išsamesnę informaciją apie konkretų stebėjimą, gauti pilną informaciją apie ankstesnius stebėjimus. Tačiau informacija apie retus ar nykstančius paukščius turi tokius pat matomumo apribojimus, kurie suteikti neprisiregistravusiems sistemos lankytojams. Be šių galimybių registruotas dalyvis gali pildyti savo stebėjimų dienoraščius, peržiūrėti informaciją apie kitų dalyvių veiklą, tai yra kiek konkretus registruotas dalyvis stebėjo paukščių, kokias rūšis jam pavyko užregistruoti ir panašiai.



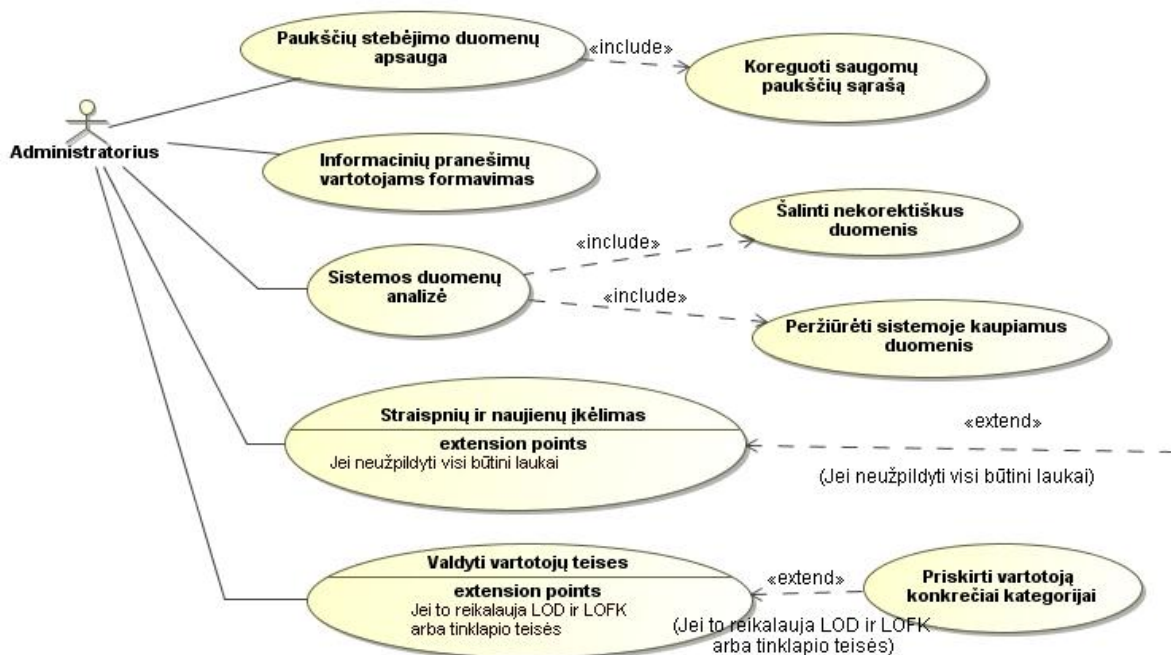
18 pav. Panaudos atvejų diagrama nusakanti registruoto vartotojo galimybes

19 pav. esančioje diagramoje išskirta *LOD* ir *LOFK* vartotojų grupė, kuriai suteiktos papildomos galimybės susijusios su išsamia suvestų duomenų peržiūra bei informacijos apie retus ar nykstančius paukščius analize, tai yra *LOFK* atstovai peržiūri pateiktą informaciją apie retą ar nykstantį paukštį ir analizuoja ar pateikti duomenys yra korektiški.



19 pav. Panaudos atvejų diagrama nusakanti LOD ir LOFK galimybes

Už informacinės sistemos korektišką veikimą ir priežiūrą atsakingas administratorius 20 pav. Jo veikla apima nekorektiškos informacijos šalinimą, naujienų įkėlimą, vartotojų teisių valdymą ir pan.



20 pav. Panaudos atvejų diagrama nusakanti administratoriaus funkcijas

Taigi, suformuotas esybių-klasių modelis, bei panaudos atvejų diagrama leidžia susidaryti pakankamai išsamų vaizdą apie kuriamos sistemos funkcionalumą, saugomos informacijos struktūrą ir pobūdį. Gautoji informacija leidžia pradėti kurti prototipinį sistemos modelį, kurio tolimesnis vystymasis leistų sukurti išbaigtą ir nurodytus reikalavimus

tenkinančią sistemą. Detalesnė informacija apie kompiuterizuojamus panaudos atvejus pateikta 1 priede, kompiuterizuojamų panaudos atvejų specifikacijoje.

3.1.2. Vartotojo sąsajos modelis

Informacinės sistemos vartotojo sąsajos modelis pavaizduotas 21 paveiksle. Iš šios diagramos galima išskirti dvejopą informacijos pasiekimo būdą, tai yra naudojantis meniu ir tinklapio viršutinėje, bei apatinėje dalyje esančiomis nuorodomis. Naudojantis meniu skiltyje esančiomis nuorodomis galima prieiti prie paukščių registracijos įvedimo formų, peržiūrėti registruotų dalyvių statistinius duomenis, pildyti ir peržiūrėti jau įvestus paukščių stebėjimo dienoraščius, peržiūrėti paukščių nuotraukų galeriją, bei prieiti prie prisiregistravusio dalyvio asmeninės informacijos, tai yra registracijos duomenų, asmeninių stebėjimų statistikos, įkeltų nuotraukų peržiūros. Kitose tinklapio dalyse esančios nuorodos yra informacinio pobūdžio. Vienos nuorodos pateikia statinę informaciją pvz., „Kontaktai“ – pateikia kontaktinę informaciją; „Apie sistemą“ – pateikiama sistemos sukūrimo esmė ir tikslas. Tačiau nuoroda susijusi su įvestų stebėjimų peržiūra, skirta atvaizduoti kintančiai ir nuolatos atsinaujinančiai informacijai. Pastaroji informacija atvaizduojama įvairiais pjūviais, su galimybe duomenis filtruoti, rikiuoti, bei grupuoti.

TERITORIJA IR ORO SĄLYGOS

Oro sąlygos	
Temperatūra	<input type="text"/>
Debesuotumas	<input type="text"/>
Krituliai	<input type="text"/>
Vėjas	<input type="text"/>
Teritorijos duomenys	
Data	<input type="text"/>
Stebėjimo pradžia	<input type="text"/>
Stebėjimo pabaiga	<input type="text"/>
Utm	<input type="text"/>
Rajonas	<input type="text"/>

Filtrai

Žemėlapis

22 pav. Paukščių stebėjimo registracijos forma (teritorijos nurodymas ir oro sąlygų įvedimas)

STEBĖTI PAUKŠČIAI IR STEBĖJIMO TIPAS

Stebėjimo pobūdis	<input type="text"/>
Pauks	<input type="text"/>
Kiekis	<input type="text"/>
Matomumas	<input type="text"/>

23 pav. Paukščių stebėjimo registracijos forma (stebėjimo pobūdžio ir stebėto paukščio pasirinkimas)

INFORMACIJA APIE PAUKŠTĮ

Esminė informacija

Lytis

Amžius

Kiekis

Biotopas

Vietovės pavadinimas

Laikas

Papildoma informacija

Ar buvo stebėtas?

Ar buvo išgirstas?

Ar buvo girdetas tuoktuvų balsas?

Buvo matyta už (metrų)

Buvo išgirsta už (metrų)

Ar skrido?

Skrydžio kryptis

Ar ilsėjosi?

Ar maitinosi?

Ar vyko tuoktuvės?

Nuotrauka

Komentaras

24 pav. Paukščių stebėjimo registracijos forma (pavienių paukščių stebėjimo forma)

Magistrinio darbo vienas iš tikslų – duomenų analizei, atvaizdavimui ir nagrinėjimui panaudoti *Exhibit* įrankį. 25 pav. pateiktas interaktyvus paukščių duomenų atvaizdavimas naudojant *google maps*, kuris susietas su *Exhibit* aspektais. Kadangi informacija apie paukščio stebėjimo faktą yra gana plati, atsiranda noras ir poreikis suvestą informaciją peržiūrėti įvairias pjūviais tai yra peržiūrėti informaciją pagal tam tikrą Lietuvos raudonosios knygos kategoriją, pagal paukščių būrį, individą, stebėjimo teritoriją ir taip toliau. Be to *Exhibit* leidžia informaciją susieti ne tik su žemėlapiu, bet ir atvaizduoti grafikais, kalendoriumi, interaktyviai rikiuojamu/grupuojamu sąrašu ir panašiai.

PAUKŠČIŲ STEBĖJIMAI

Ieškoti

Rajonas	Stebėjimo tipas	RK kategorija IUCN	Metai

Įvairūs informacijos atvaizdavimo būdai

Būrys

Seima

Paukštis

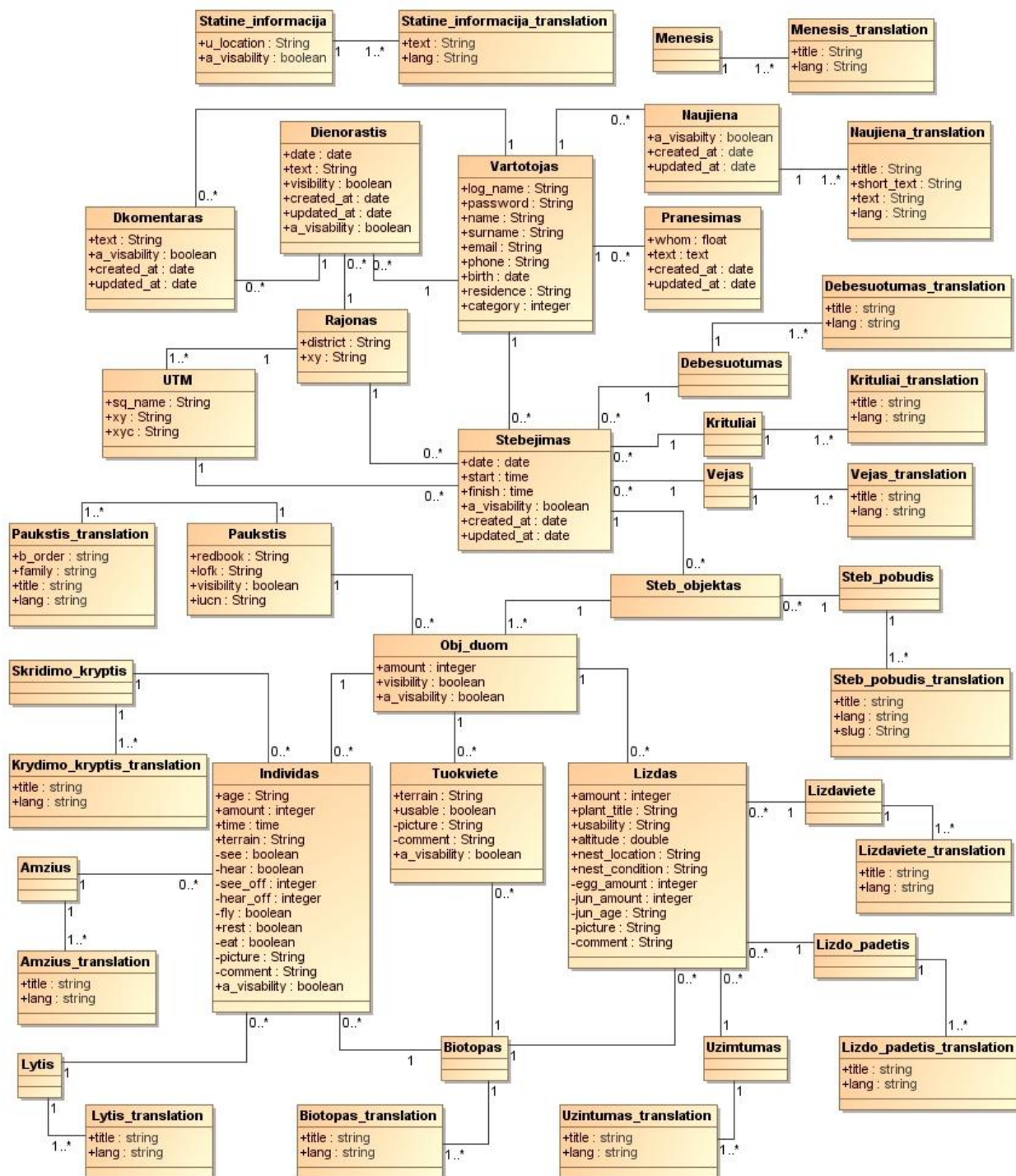
25 pav. Paukščių stebėjimo duomenų atvaizdavimas ir nagrinėjimas naudojant *Exhibit*

3.2. Dalykinės srities modelis

Dalykinės srities analizės bei funkcinį reikalavimų išgavimo metu buvo formuojamas informacinės sistemos konceptų modelis, kuris vystymosi eigoje transformavosi į kompiuterizuojamos srities esybių modelį. Šio modelio 26 pav. pagrindu ateityje bus formuojama duomenų bazės schema.

Taigi, aiškumo dėlei esybių modelio diagramą galima padalinti į dvi dalis. Viena esybių-klasijų modelio dalis atsakinga už informaciją, kuri susijusi su mažai kintančios informacijos saugojimu, tai yra informacijos, kurios prisiregistravęs dalyvis neįvedinėja. Kaip pavyzdį galima paimti informaciją susijusią su stebimo paukščio teritorija, tai yra rajonų bei *UTM*¹¹ kvadratų lentelės, kuriose informacija užpildoma iš anksto. Kitaip sakant *UTM* kvadratas geografiškai susiejamas su žemėlapiu, o po to su konkrečiu rajonu ar rajonais.

¹¹ *UTM* (angl. *Universal Transverse Mercator*) – tam tikra geografinių koordinatų sistema



26 pav. Esybių, apie kurias planuojama kaupti informaciją, UML klasių diagrama

Antroji esybių diagramos dalis skirta saugoti informacijai, kuri susijusi su paukščių stebėjimais. 26 pav. pateiktoje diagramoje reiktų išskirti tris lenteles „Individias“, „Tuokvietė“ ir „Lizdas“. Šios lentelės atsakingos už konkrečią su stebimu paukščiu susijusią informaciją. Be to, šios lentelės nebūtinai kiekvieno paukščio stebėjimo įvedimo metu bus pildomos. Lentelių pildymas priklauso nuo pasirinkto stebėjimo pobūdžio, kuris įtakoja, kurią iš šių lentelių paukščio stebėtojas turės užpildyti. Siekiant optimizuoti formų pildymo procesą, informacija suskaidyta taip, kad vartotojui tektų kuo mažiau įvedinėti pasikartojančią

informaciją, tokiu būdu sutaupant netik stebėtojo laiką, bet ir mažinant informacijos kartojimąsi informacinėje sistemoje.

3 lentelėje trumpai aptariamos 26 pav. pavaizduotos esybės, tai yra kokią informaciją jos saugo. Reiktų pabrėžti, kad šiame poskyryje nebus gilinamasi į esybių atributų reikšmių aiškinimą, tačiau informacija apie atributų paskirtį ir tipą yra pateikta 2 priede.

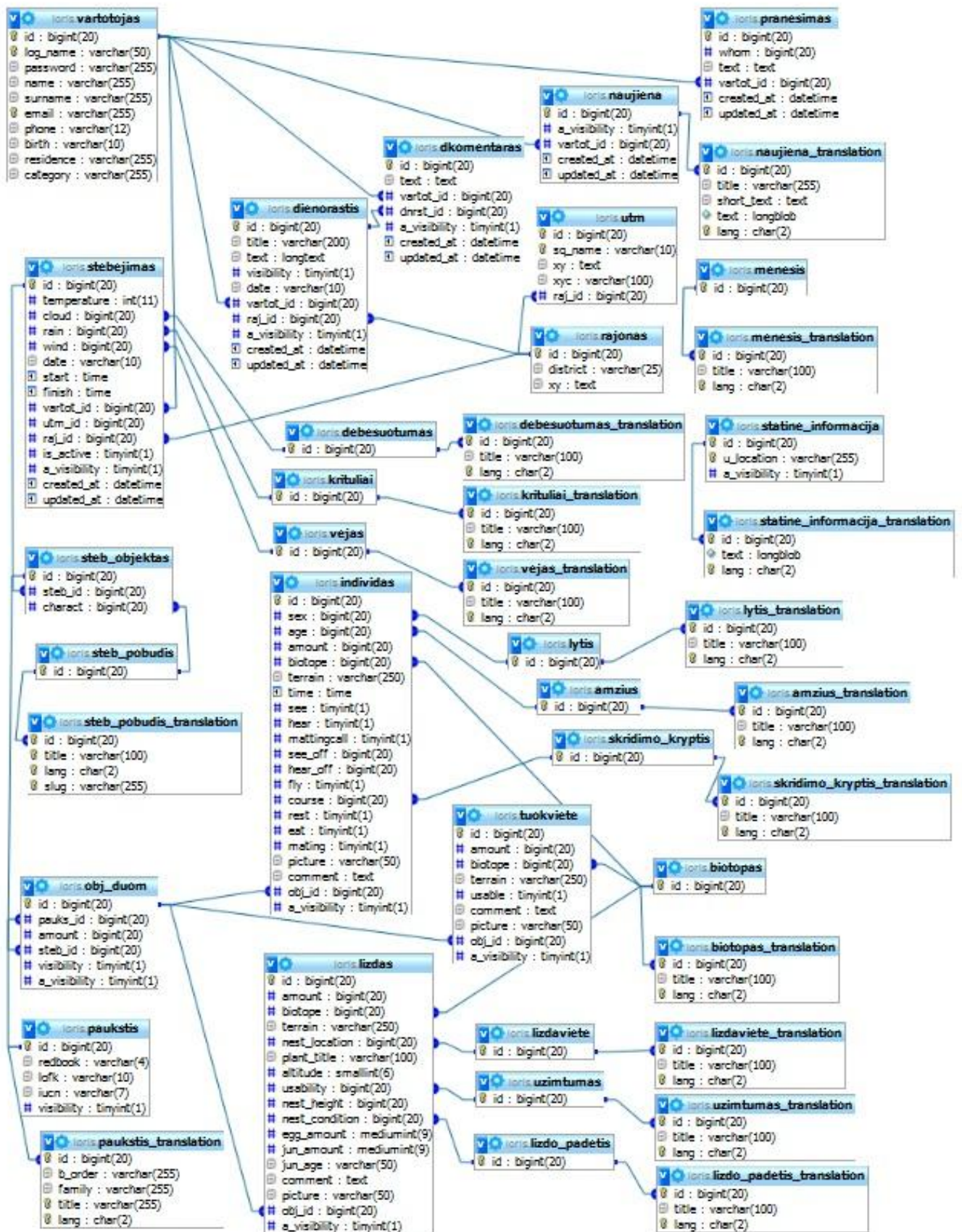
3 lentelė Klasių diagramos elementų saugomos informacijos pobūdžiai

Esybė	Saugoma informacija
<i>Vartotojas</i>	Informacija apie sistemoje prisiregistravusius dalyvius.
<i>Dienorastis</i>	Registruotų vartotojų įvesti dienoraščiai.
<i>Dien_komentar</i>	Dienoraščių komentarai.
<i>Pranesimas</i>	Įvairaus pobūdžio pranešimai, kuriuos suformuoja sistemos administratorius ir kurie yra skirti konkrečiam vartotojui.
<i>Rajonas</i>	Lietuvos rajonų sąrašas.
<i>UTM</i>	Lietuvos teritorijoje esančių UTM kvadratų (10x10km) duomenys.
<i>Stebėjimas</i>	Konkrečios teritorijos, kurioje atliekami stebėjimai duomenys, apimantys oro sąlygas, stebėjimo datą, stebėjimo laikotarpį.
<i>Debesuotumas</i>	Galimi debesuotumo atvejai.
<i>Debesuotumas_translation</i>	Debesuotumo atvejų vertimai.
<i>Krituliai</i>	Galimi kritulių buvimo atvejai.
<i>Krituliai_translation</i>	Kritulių buvimo atvejų vertimai.
<i>Vejas</i>	Vėjo būsenos atvejai.
<i>Vejas_translation</i>	Vėjo būsenų vertimai.
<i>Steb_objektas</i>	Stebėjimo atvejo susiejimas su stebėjimo pobūdžiu.
<i>Steb_pobudis</i>	Galimi stebėjimo pobūdžiai, pvz. Pavieniai stebėjimai, paukščių sankaupos, lizdavietės ir taip toliau
<i>Steb_pobudis_translation</i>	Galimų stebėjimo pobūdžių vertimai.
<i>Paukstis</i>	Lietuvoje užregistruotos paukščių rūšys ir informacija apie jas.
<i>Paukstis_translation</i>	Verčiama informacija apie paukštį, tai yra , būrys, šeima, bei paukščio pavadinimas kitomis kalbomis.
<i>Obj_duom</i>	Stebėjimo pobūdžio susiejimas su konkrečiu individu.
<i>Individas</i>	Informacija apie konkretų paukštį.
<i>Skridimo_kryptis</i>	Skridimo kryptis
<i>Skrydimo_kryptis_translation</i>	Skridimo krypties vertimai.
<i>Amžius</i>	Galimas paukščio amžius.
<i>Amžius_translation</i>	Galimo amžiaus pavadinimo vertimas į kitas kalbas.
<i>Lytis</i>	Paukščio lytis
<i>Lytis_translation</i>	Paukščio lyties pavadinimo vertimas
<i>Biotopas</i>	Biotopo tipai
<i>Biotopas_translation</i>	Biotopo tipų vertimai.
<i>Tuokvietė</i>	Informacija apie stebėtą paukščių tuoktuvių vietą.
<i>Lizdas</i>	Informacija apie stebėtas paukščių lizdavietes.
<i>Lizdaviete</i>	Lizdo sukrovimo vieta.

<i>Lizdaviete_translation</i>	Lizdo sukrovimo vietos vertimai.
<i>Lizdo_padetis</i>	Galimas lizdo išsidėstymas ant tam tikro objekto.
<i>Lizdo_padetis_translation</i>	Lizdo išsidėstymo pobūdžio vertimai
<i>Uzimtumas</i>	Lizdo užimtumo būseną.
<i>Uzimtumas_translation</i>	Lizdo užimtumo būsenos vertimai.
<i>Statine_informacija</i>	Statinės informacijos tekstas.
<i>Statine_informacija_translation</i>	Statinės informacijos teksto vertimai.
<i>Menesis</i>	Mėnesių pavadinimai.
<i>Menesis_translation</i>	Mėnesių laukų vertimai.
<i>Naujiena</i>	Naujienos identifikatorius, bei naujienos sukūrimo laikas.
<i>Naujiena_translation</i>	Su naujiena susiję duomenys.

3.3. Duomenų bazės schema

Atsižvelgiant į 26 pav. pateiktą esybių klasių diagramą, realizuota informacinės sistemos duomenų bazės schema 27 pav. Šios schemos realizavimui pasirinkta *MySQL* duomenų bazė, kurios naujausia versija leidžia atvaizduoti suprojektuotos duomenų bazės schemą bei sukurtus ryšius tarp lentelių. Kiekvieno lentelės atributo paskirtis aprašyta 2 priedo 1 lentelėje. Joje pateikiamas atributo pavadinimas, tipas bei trumpas aprašymas.



27 pav. Realizuotos sistemos duomenų bazės schema

4. Informacinės sistemos projektas

4.1. Projekto tikslas

Projekto tikslas – suprojektuoti tokio pobūdžio informacinę sistemą, kuri pademonstruotų *Exhibit* įrankio teikiamus informacijos peržiūros ir nagrinėjimo privalumus, bei atskleistų esamus įrankio trūkumus. Taip pat, pademonstruoti surastą ir pritaikytą atsiradusios problemos sprendimą.

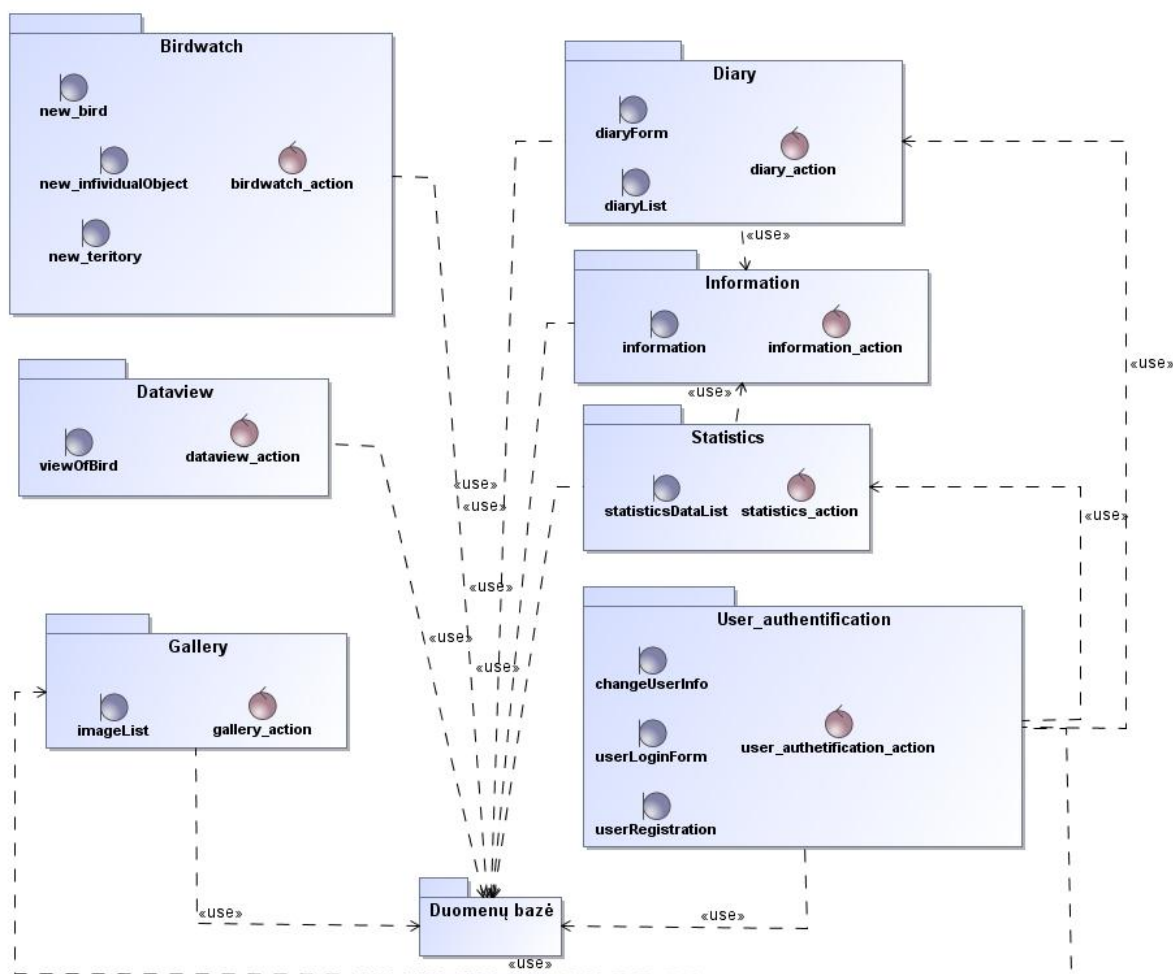
4.2. Sistemos loginė architektūra

Lietuvos ornitofaunos registravimo sistemos kūrimui naudojamas *Symfony* karkasas (angl. *framework*), kuris paremtas trijų lygių architektūros naudojimu. Taigi, kiekvienam sistemos projektavimo metu išsiskirtam posistemiiui taikomas trijų lygių architektūros modelis 28 pav.



28 pav. Loginė trijų lygių architektūra

Kuriamų posistemių priklausomybė ir vidinė struktūra, kuri bus realizuota programuojant informacinę sistemą, pateikta 29 pav. Pastarajame paveikslėlyje matoma, kad informacinės sistemos viešoji dalis (angl. *frontend*) suskaidyta į septynis posistemius. Kiekvienas posistemis atsakingas už tam tikrą informacinės sistemos sritį, tai yra „*Birdwatch*“ posistemis atsakingas už paukščių stebėjimo formos realizaciją, „*Diary*“ – už įvestų dienoraščių atvaizdavimą ir naujų dienoraščių sukūrimą, „*Information*“ – už statinės ar mažai kintančios informacijos atvaizdavimą, „*Dataview*“ – skirta atvaizduoti įvestiems stebėjimams, „*Statistics*“ – atvaizduoja įvestų stebėjimų statistiką, tai yra kiek ir ką pavyko pastebėti sistemoje prisiregistravusiems paukščių stebėtojams, „*Gallery*“ – atvaizduoja paukščių registracijos metu įkeltas nuotraukas, „*User_authentication*“ – posistemis atsakingas už registruoto vartotojo informaciją bei prisijungimo ir registracijos formų atvaizdavimą ir korektišką veikimą.

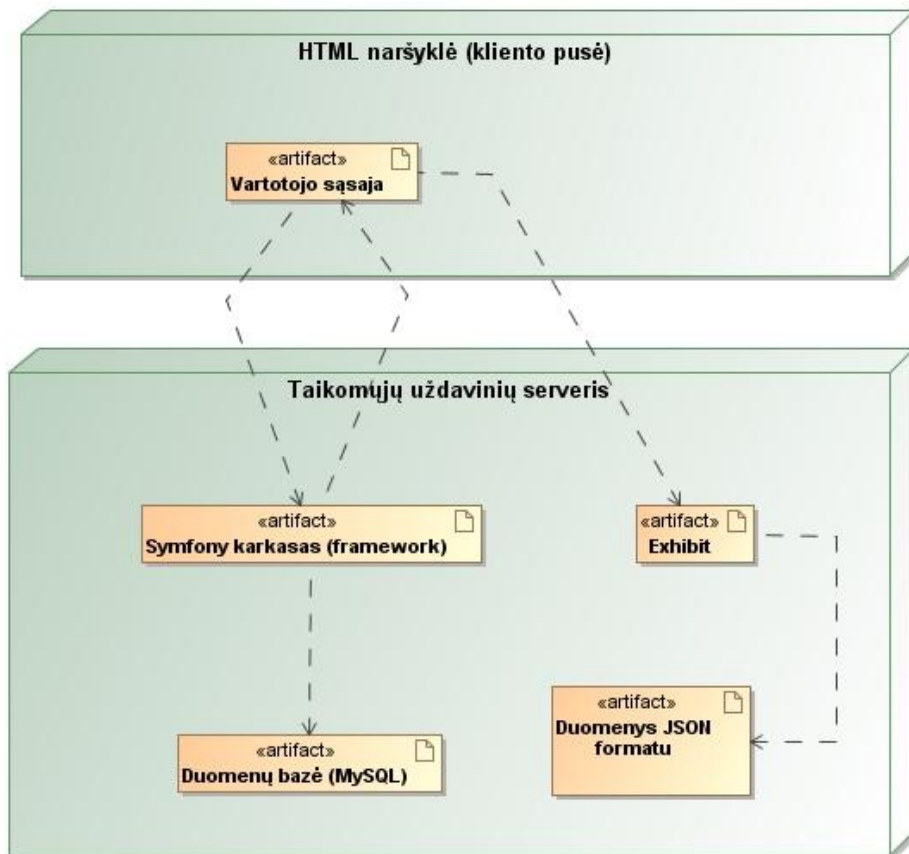


29 pav. Paukščių registravimo sistemos architektūra

4.3. Sistemos realizacijos modelis

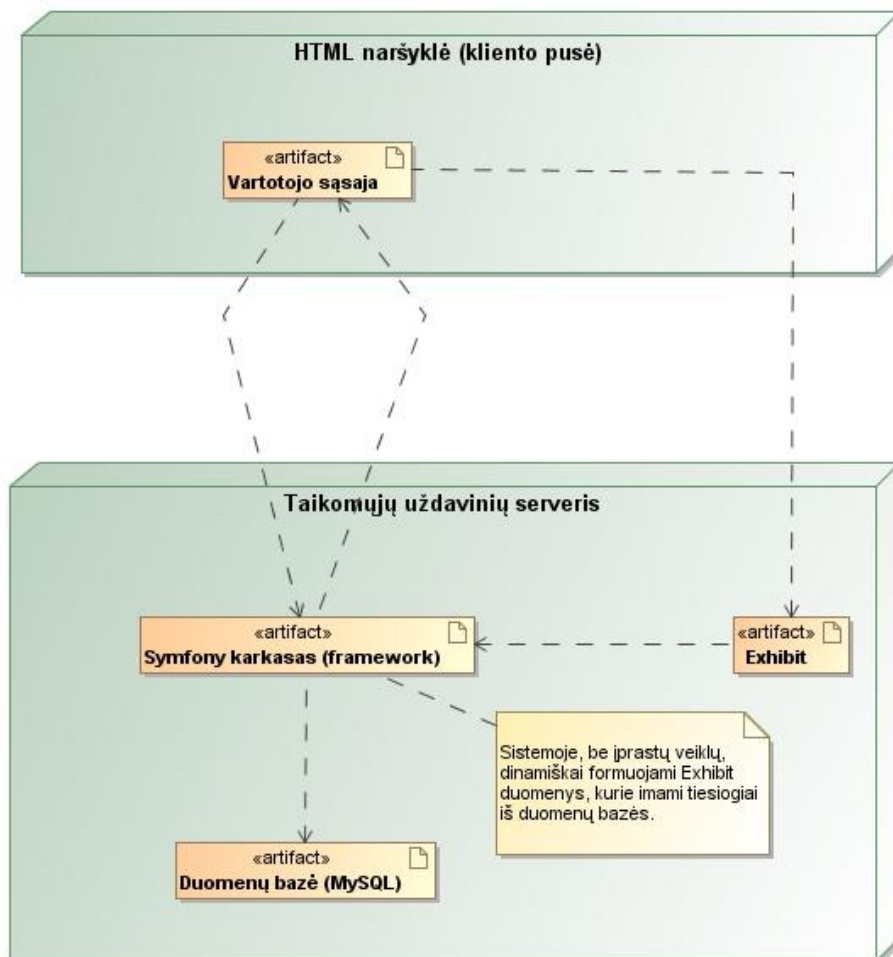
Kaip buvo minėta anksčiau, sistemai realizuoti bus naudojama trijų lygių *MVC* (angl. *Model View Controller*) architektūra. Siekiant dar labiau paspartinti sistemos kūrimą, o kartu įnešant dar daugiau naujoviškumo bus panaudotas *Symfony* karkasas, kuris paremtas objekciniu *PHP* kalbos naudojimu.

Realizuojamos sistemos fizinė architektūra pateikta 30 pav. Ji susideda iš dviejų dalių. Pirmoji dalis susideda iš serveryje saugomos informacinės sistemos, jos duomenų bazės, *Exhibit* įrankio, kuris bus naudojamas duomenų filtracijai ir vizualizacijai realizuoti. Antroji dalis susideda iš kliento pusės, tai yra *web* naršyklės, kuri atvaizduoja gaunamą informaciją ir siunčia vartotojo komandas atgal į informacinę sistemą.



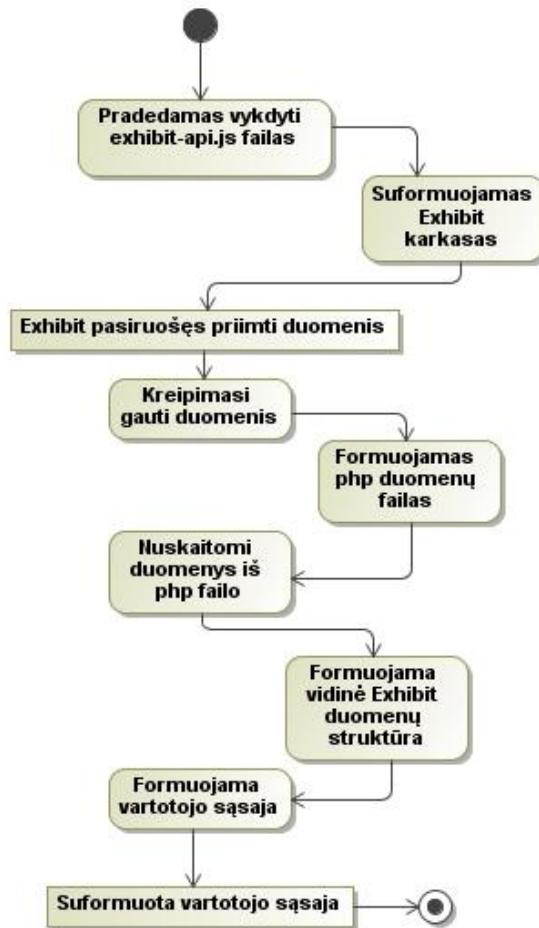
30 pav. Informacinės sistemos realizacijos modelis, kai nėra realizuotas *Exhibit* duomenų dinamiškumas

Šiame modelyje vis dar yra matomas *Exhibit* duomenų failas. Tačiau jo buvimas yra tik santykinis, kadangi galutinėje sistemos realizacijoje šio failo nebereikės, nes visi reikalingi duomenys bus tiesiogiai formuojami iš duomenų bazės ir dinamiškai užkraunami į vidinę *Exhibit* duomenų bazę 31 pav.



31 pav. Siekiamas informacinės sistemos realizacijos sprendimas ir *Exhibit* duomenų dinamiškumo sukūrimas

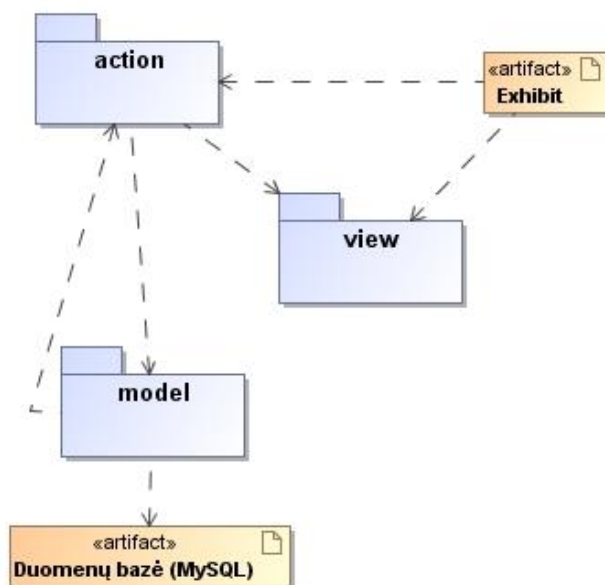
Kaip jau buvo minėta anksčiau, *Exhibit* struktūra susideda iš keliasdešimt *javascript* failų, kurių kiekvienas turi būti užkraunamas vartotojo pusėje. Šakninis *Exhibit* įrankio failas – *exhibit-api.js*. Būtent nuo šio failo iškvietimo pradedama formuoti sistemos struktūra. Norint atskleisti praplėstą *Exhibit* funkcionalumą 32 pav. pateikta supaprastinta duomenų užkrovimo į *Exhibit* eiga.



32 pav. Siekiamas informacinės sistemos realizacijos sprendimas ir *Exhibit* duomenų dinamiškumo sukūrimas

Taigi, iš pat pradžių paleidžiamas vykdyti *exhibit-api.js* failas, kuris inicijuoja kitų failų vykdymą, bei apibrėžia pradinis *Exhibit* parametrus. Tie parametrai, tai įrankio vertimų kalba, stilius ir panašiai. Kitų inicijuotų *javascript* failų įvykdymo paskirtis - baigti formuoti vidinę *Exhibit* struktūrą, peržiūros formų, filtrų, vidinės duomenų struktūros karkasą.

Taigi, priklausomai nuo naršyklės tipo ir nuo internetinio tinklo spartos, pastarųjų veiksmų įvykdymas gali užtrukti įvairiai. Tačiau, šį laiką galima sumažinti panaudojus kešavimą, įvairias *javascript* turinio suspaudimo funkcijas ir pan. Baigus formuoti vidinę *Exhibit* struktūrą, suformuojamas kreipimasis nuskaityti duomenis. Standartiškai, *Exhibit* įrankis tiesiogiai kreiptųsi į *JSON* failą, tai yra į failą, kuriame saugomi duomenys *JSON* formatu. Tačiau, magistrinio darbo metu padaryta taip, kad būtų kreipiamasi į *symfony* sistemos veiklą (angl. *action*). Ši veikla savo ruožtu kreipiasi į metodą, kuris pagal užduotus parametrus suformuoja duomenų bazės užklausą ir gražina rezultatą *JSON* formatu. Gautasis rezultatas pasiunčiamas į pastarosios veiklos šabloną (angl. *template*). Šio šablono turinys nuskaitymas su *AJAX XMLHttpRequest*. Vėliau iš nuskaitytų duomenų suformuojamas objektas, iš kurio baigiama formuoti vidinę *Exhibit* įrankio duomenų struktūra (orientuotas grafas) 33 pav.



33 pav. Exhibit kreipimasis gauti duomenis ir duomenų nuskaitymas

Kai jau pilnai suformuota vidinė duomenų struktūra pradedamas vartotojo sąsajos formavimas, tai yra duomenų filtrų užpildymas duomenimis, duomenų atvaizdavimo formų užpildymas duomenimis, informacijos atvaizdavimo stilių taikymas ir panašiai. Šio žingsnio trukmė priklauso nuo sąsajos sudėtingumo, bei nuo atvaizduojamų duomenų kiekio.

5. Lietuvos ornitofaunos registravimo sistemos realizacija

5.1. Veikimo aprašymas

5.1.1. Sistemos paskirtis

LORIS – Lietuvos ornitofaunos registravimo informacinė sistema 34 pav. Šios sistemos paskirtis – kaupti informaciją apie Lietuvoje pastebėtus paukščius, bei jų veiklos žymes, tai yra lizdavietes, tuokvietes, migraciją ir panašiai. Sistema taip pat suteikia galimybę peržiūrėti įvestus stebėjimus.

Siekiant sukurti kuo patogesnę ir vartotojui patrauklesnę informacijos peržiūros galimybę, naudojamas *Exhibit* įrankis. Šio įrankio pagalba vartotojas keliais funkciniais žingsniais gali nusifiltruoti pateiktus duomenis, bei juos peržiūrėti įvairiais pjūviais, tai yra duomenys gali būti atvaizduojami žemėlapyje, diagramomis, laiko juostoje, kalendoriuje, lentelėje, sąrašu. Be to informacijos pateikimas yra interaktyvus, o tai dar labiau suteikia patrauklumo duomenų nagrinėjimui.

Be šių esminių funkcionalumų, sistema sukuria galimybę įkelti paukščių nuotraukas, pildyti stebėjimų dienoraštį, peržiūrėti naujienas.

LORIS
Lietuvos ornitofaunos registravimo sistema

Prisijungti » | Esi naujas paukščių stebėtojas? Pradėti čia »

PRADŽIA | STEBĖJIMŲ PERŽIŪRA | TIKLAPIO STRUKTŪRA | KONTAKTAI

Aktyviausi rajonai:

1. Ignalinos r.	(21)
2. Šalčininkų r.	(20)
3. Biržų r.	(18)
4. Šilutės r.	(18)
5. Švenčionių r.	(18)
6. Panevėžio r.	(16)
7. Vilkaviškio r.	(16)
8. Kauno r.	(15)
9. Vilniaus r.	(15)
10. Kelmės r.	(14)

Šiandien užregistruota:
paukščių stebėjimų - 0
paukščių rūšių - 0

Iš viso registruota stebėjimų: 519

Paukščių registracija ▶

Dienoraštis ▶

Statistika ▶

Galerija ▶

Asmeninis profilis ▶

PASKUTINĖS NAUJIENOS

STEBĖJIMAI

REČIAUSI SAVAITĖS PAUKŠČIAI

kas, kada ir kur stebėjo rečiausius paukščius Lietuvoje

34 pav. LORIS informacinė sistema

5.1.2. Neregistruoto vartotojo veiklos

Neregistruotam arba neprisijungusiam prie sistemos vartotojui suteikiamos tik informacijos peržiūros galimybės. Be to, tam tikros informacijos peržiūros matomumą gali apriboti pats informacijos pateikėjas arba sistemoje automatiškai taikomi konkretūs apribojimai, pvz. neprisijungęs vartotojas gali peržiūrėti, tik paskutinius 10 paukščių stebėjimų ir pan.

Taigi, neregistruotas arba neprisijungęs prie sistemos vartotojas, registracijos ir/arba prisijungimo veiksmus gali atlikti užpildęs 35 pav. ir/arba 36 pav. esančią duomenų įvedimo formą.

PRISIJUNGTI

Prisijungimo vardas

Slaptažodis

[Užmiršote savo slaptažodį?](#)

35 pav. Sistemos vartotojo prisijungimo forma

REGISTRACIJA

Vartotojo duomenys

Prisijungimo vardas

Vardas

Pavardė

E-paštas

Telefonas

Gimimo diena 

Gyvenamoji vieta

Saugumo duomenys

Slaptažodis

36 pav. Sistemos vartotojo registracijos forma

Neprisijungęs vartotojas gali peržiūrėti registruotų dalyvių pildytus paukščių stebėjimo dienoraščius, su sąlyga, kad šio dienoraščio savininkas nėra uždėjęs informacijos matomumo apribojimo 37 pav.

DIENORAŠČIŲ SĄRAŠAS

2011-03-07 / JONAVOS R.

Informacija informacija

Nuo sienos penkiasdešimties kilometrų atstumu gyvenantiems žmonėms leidimai kainuos 20 eurų, galios penkerius metus. „Tiek dokumentų blankai, tiek vienoje, tiek kitoje pusėje yra pagaminti, informacinės sistemos yra paruoštos ir testuojamos. Juridine prasme baltarusių pusė pasakė, kad mus informuos, kada jie bus pasirengę ir mums galioja tos pačios taisyklės“, – „Panora ...

tomas

[more »](#)

[First page](#) [Previous page](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next page](#) [Last page](#)

37 pav. Sistemoje pateikto dienoraščio pavyzdys

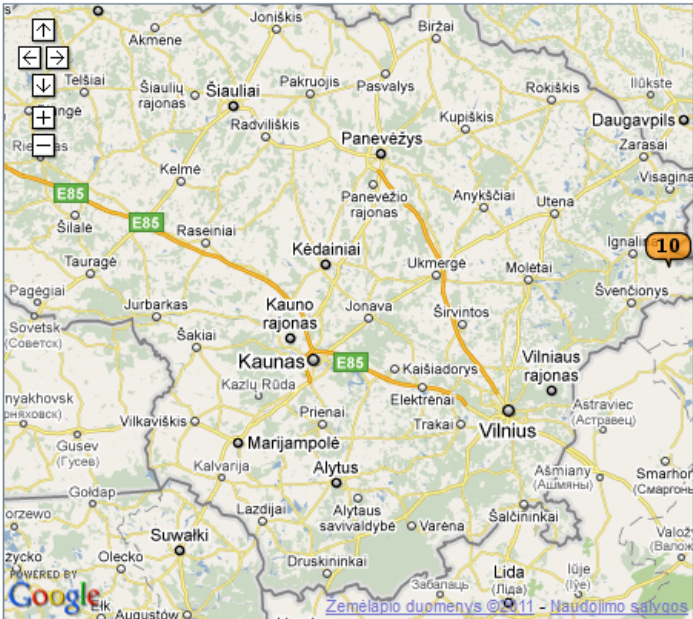
Kaip jau buvo minėta anksčiau, neprisijungęs sistemos dalyvis gali peržiūrėti paukščių stebėjimo informaciją, tačiau yra ribojamas tos informacijos kiekis ir pilnumas 38 pav. Šiuo atveju yra taikomi tokie apribojimai: pateikiama 10 paskutinių stebėjimų, nerodoma išsami informacija apie konkretų stebėjimą.

PAUKŠČIŲ STEBĖJIMAI

Ieškoti

Rajonas	Stebėjimo tipas	RK	IUCN	Metai	<input checked="" type="checkbox"/> 1	Mėnuo
10 Ignalinos r.	3 Paukščių sankaupos 4 Tuokvietės 3 Žasų ir gervių migracija	8 0 1 I 1 V	10 LC	10 2011	<input checked="" type="checkbox"/>	10 Gegužė

[PAUKŠČIŲ STEBĖJIMO ŽEMĖLAPIS](#) • [STEBĖJIMŲ KIEKIŲ ŽEMĖLAPIS](#) • [LAIKO DIAGRAMA](#) • [KALENDORIUS](#) • [STEBĖJIMŲ ŠARAŠAS](#) • [DETALE INFORMACIJA](#)



Taksonomijos filtras

Papildomų parametų filtras

Būrys

- 1 Gandriniai
- 9 Nariniai

Šeima

- 1 Garniniai
- 9 Nariniai

Paukštis

- 1 Baltoji pelėda
- 1 Kanadinė berniklė
- 1 Mėlyngurklė
- 1 Nendrinė vištelė
- 1 Pelėsakalis
- 1 Pilkasis qarnys

38 pav. Paukščių stebėjimų atvaizdavimas žemėlapyje

Taigi, neregistruotam arba neprisijungusiam prie sistemos vartotojui suteikiamos nors ir apribotos, tačiau pakankamai išsamios informacijos peržiūros galimybės. O tai leidžia sužinoti naujausią informaciją apie paukščių stebėjimus, bei motyvuoja registruotis sistemoje ir išplėsti savo veiklos galimybes.

5.1.1. Prisijungusio sistemos vartotojo veiklos

Iš 16 pav. esančios panaudos atvejų diagramos (detaliau 17-20 pav.) galima pastebėti, kad sistemoje galimi 4 registruotų dalyvių tipai. Kurių kiekvieno veikla praplečiama arba konkrečios veiklos vykdymo galimybe, arba informacijos pateikimo išsamumu.

Taigi, prisijungęs prie sistemos vartotojas gauna galimybę netik peržiūrėti daugiau paukščių stebėjimų, bet galimybę įvesti savo paukščių stebėjimus 39-41 pav.

TERITORIJA IR ORO SĄLYGOS

Oro sąlygos

Temperatūra

Debesuotumas

Krituliai

Vėjas

Teritorijos duomenys

Data

Stebėjimo pradžia :

Stebėjimo pabaiga :

Utm

Rajonas

Rajonas 1

- 9 Akmenės r.
- 18 Alytaus r.
- 18 Anykščių r.
- 20 Biržų r.
- 13 Druskininkų r.
- 7 Elektrėnų r.
- 22 Ignalinos r.
- 9 Jonavos r.
- 14 Joniškio r.
- 16 Jurbarko r.
- 10 Kaišiadorių r.
- 6 Kalvarijos r.
- 21 Kauno r.
- 7 Kazlų Rūdos r.



39 pav. Pirmasis paukščių stebėjimų duomenų įvedimo žingsnis

STEBĖTI PAUKŠČIAI IR STEBĖJIMO TIPAS

Stebėjimo pobūdis

Pauks

- Rudakaklis naras
- Juodakaklis naras
- Ledinis naras
- Geltonsnapis naras
- Mažasis kragas
- Ausuotasis kraogas

Kiekis

Matomumas

40 pav. Antrasis paukščių stebėjimų duomenų įvedimo žingsnis

INFORMACIJA APIE PAUKŠTĮ

Esminė informacija

Lytis	Unspecified ▾
Amžius	Ad. ▾
Kiekis	<input type="text"/>
Biotopas	Fir forest ▾
Vietovės pavadinimas	<input type="text"/>
Laikas	<input type="text"/> : <input type="text"/>

Papildoma informacija

Ar buvo stebėtas?	<input type="checkbox"/>
Ar buvo išgirstas?	<input type="checkbox"/>
Ar buvo girdetas tuoktuvių balsas?	<input type="checkbox"/>
Buvo matyta už (metrų)	<input type="text"/>
Buvo išgirsta už (metrų)	<input type="text"/>
Ar skrido?	<input type="checkbox"/>
Skrydžio kryptis	- ▾
Ar ilsėjosi?	<input type="checkbox"/>
Ar maitinosi?	<input type="checkbox"/>
Ar vyko tuoktuvės?	<input type="checkbox"/>
Nuotrauka	<input type="button" value="Pasirinkti failą"/> Nepasirinktas joks failas
Komentaras	<input type="text"/>

41 pav. Trečiasis paukščių stebėjimų duomenų įvedimo žingsnis

Be to suteikiama galimybė pildyti paukščių stebėjimo dienoraštį. Dienoraščio pildymo forma pateikiama 42 pav.

SUKURTI DIENORAŠTĮ

Įvedama informacija

Rajonas: Akmenės r.

Data:

Antraštė:

Tekstas

Path:

Matomumo parametrai

Matomumas

Suteikti galimybę neregistruotiems dalyviams matyti šį dienoraštį

42 pav. Paukščių stebėjimo dienoraščio pildymo forma

Kaip žinoma, informacinė sistema turi dvi informacijos valdymo posistemės, tai yra viena posistemė skirta išoriniam sistemos vartotojui (angl. *frontend*), o kita skirta sistemoje kaupiamos informacijos valdymui (angl. *backend*). Pirmosios posistemės esminiai langai pateikti šio skyriaus 34 – 42 pav.

Didesnes teises turintys sistemoje registruoti dalyviai gali dirbti su sistemoje saugoma informacija, ją koreguoti ir kitaip valdyti. Vidinės sistemos langas pateiktas 43 pav.

Vartotojai Pranešimai vartotojams Dienoraščiai Rajonai UTM kvadratai Paukščių stebėjimai

Stebėjimo objektai Stebėjimo objekto duomenys Paukščiai Individas Lizdas Tuokvietė

Vartotojų sąrašas

<input type="checkbox"/>	Id	Prisijungimo vardas	Vardas	Pavardė	E-paštas	Veiksmai
<input type="checkbox"/>	1	tomas	tomas	tomas	tomas12@gmail.com	Redaguoti ✖ Šalinti
<input type="checkbox"/>	2	lina	lina	lina	lina@lina.lt	Redaguoti ✖ Šalinti
<input type="checkbox"/>	3	rokas	rokas	rokas	rokas@rokas.lt	Redaguoti ✖ Šalinti
<input type="checkbox"/>	4	topas	topas	topas	tomas@topas.lt	Redaguoti ✖ Šalinti
<input type="checkbox"/>	5	topas1	topas	topas	topas@topas.lt	Redaguoti ✖ Šalinti
<input type="checkbox"/>	6	as	as	as	tomasas@topas.lt	Redaguoti ✖ Šalinti
<input type="checkbox"/>	7	as1	as1	as1	as1@as.lt	Redaguoti ✖ Šalinti

7 results

Choose an action [+ Naujas](#)

Prisijungimo vardas

Vardas

Pavardė

E-paštas

Gimimo diena is empty

Category

43 pav. Administravimo langas

43 pav. esantis vaizdas, parodo struktūrą, kuri analogiškai kartojasi kituose moduluose, kurie skirti valdyti konkrečios duomenų bazės lentelės turinį. Taigi, standartiškai pradiniame puslapyje pateikiamas lentelėje esančios informacijos sąrašas, informacijos filtras, bei galimybė šalinti arba koreguoti esančią informaciją. Tarkim pasirinkus koreguoti konkretų įrašą, atidaroma duomenų koregavimo forma, kuri įgalina atlikti šį veiksmą 44 pav.



Koreguojamas vartotojas "tomas"	
Prisijungimo vardas	tomas
Slaptažodis	
Vardas	tomas
Pavardė	tomas
E-paštas	tomas12@gmail.com
Telefonas	545465464
Gimimo diena	1985-01-31
Gyvenamoji vieta	dsfsdfdsfdasf
Category	paprastas

✖ Šalinti Back to list Išsaugoti

44 pav. Informacijos redagavimo langas

5.2. Testavimo modelis

Pagrindinis testavimo tikslas – įsitikinti, kad tinkamai veikia realizuotos informacinės sistemos dalys, kuriose panaudotas ir pritaikytas *Exhibit* įrankis. Tačiau ne ką mažiau svarbios ir kitos sistemos dalys, kadangi nuo jų priklauso netik visos sistemos, bet ir *Exhibit* įrankio korektiškas veikimas.

5.3. Testavimo duomenys ir rezultatai

Exhibit įrankio analizės metu sužinota, kad norint nesudėtingai atvaizduoti duomenis su šiuo įrankiu, reikia suformuoti duomenų aibę taip, kad ji atitiktų *JSON* duomenų struktūrą 9 pav. Kadangi informacinės sistemos duomenys saugomi duomenų bazėje, reiktų suformuoti duomenų bazės užklausą norimiems duomenis gauti. Tokiu būdu atsirastų galimybė visą laiką manipuluoti nuolatos atsinaujinančiais ir atsirandančiais duomenimis.

Magistriniame darbe buvo paminėta, kad paukščių registravimo sistema realizuota naudojant *Symfony* karkasą (angl. *framework*), bei *MySQL* duomenų bazę. Taigi, paukščių stebėjimo duomenims gauti buvo suformuota duomenų bazės užklausa 45 pav., kurios

pakitimų, reiktų sugeneruoti tekstinį duomenų failą *JSON* formatu. Tačiau šitokio pobūdžio sprendimas yra labai neefektyvus ir griozdiškas.

Tiriamąjį darbą metu šiai problemai spręsti, buvo panaudotas *PHP* failas. Tam tikslui, standartinė nuoroda į *JSON* duomenų failą

`<link href="duomenys.js" type="application/json" rel="exhibit/data"/>` pakeista į nuorodą `<link href="<?php echo '/birdwatch/bwData?user_name=' . $user_name; ?>" type="application/json" rel="exhibit/data" />`. Taigi, šiuo atveju kreipiamasi į *bwData* veiklą (angl. *action*), kuri savo ruožtu kreipiasi į funkciją, kuri gražina suformuotą *Exhibit* duomenų eilutę. O gautosios duomenų eilutės turinys atvaizduojamas *PHP* faile. Vėliau šis failas *Exhibit* duomenų užkrovimo metu nuskaitomas kaip *JSON* duomenų šaltinis.

Sprendimo veikimui pademonstruoti į esamą paukščių stebėjimo duomenų aibę bus įtrauktas naujas stebėjimas, kuris vėliau bus peržiūrimas su *Exhibit* įrankiu.

Taigi, pradžioje turima tokia informacija: iš viso registruoti 697 paukščių stebėjimai 47 pav. Eksperimento metu bus įvestas naujas paukščių stebėjimas Ignalinos rajone.

AKTYVIAUSI RAJONAI:

1. Ignalinos r.	(27)
2. Švenčionių r.	(26)
3. Varėnos r.	(25)
4. Vilniaus r.	(25)
5. Ukmergės r.	(23)
6. Prienų r.	(21)
7. Biržų r.	(20)
8. Lazdijų r.	(20)
9. Panevėžio r.	(20)
10. Šalčininkų r.	(20)

Šiandien užregistruota:
paukščių stebėjimų - 0
paukščių rūšių - 0

Iš viso registruota stebėjimų: 697

PAUKŠČIŲ STEBĖJIMAI

leškoti

Rajonas	Stebėjimo tipas	RK	IUCN	Metai	Mėnuo
4 Alytaus r.	20 Metų paukštis	5 E	1	1396 2009	118 Sausis
8 Druskininkų r.	11 Paukščių sankaupos	3	EN	1486 2010	
6 Ignalinos r.	6 Pavieniai stebėjimai	Ex	109	118 2011	
5 Jonavos r.	19 Peledu	1 I	LC		
3 Joniškio r.		10 R	5		
3 Kaišiadorių r.		2 V	NT		

47 pav. Pradinė duomenų aibė

Po paukščių stebėjimo įvedimo Ignalinos rajone turime tokią informaciją 48 pav.

AKTYVIAUSI RAJONAI:

1. Ignalinos r.	(28)
2. Švenčionių r.	(26)
3. Varėnos r.	(25)
4. Vilniaus r.	(25)
5. Ukmergės r.	(23)
6. Prienų r.	(21)
7. Biržų r.	(20)
8. Lazdijų r.	(20)
9. Panevėžio r.	(20)
10. Šalčininkų r.	(20)

Šiandien užregistruota:
paukščių stebėjimų - 0
paukščių rūšių - 0

Iš viso registruota stebėjimų: 698

48 pav. Duomenų aibė po naujo stebėjimo įvedimo

Norint peržiūrėti įvestus stebėjimo duomenis pasirenkamas meniu laukas „Stebėjimų peržiūra“. Po šio pasirinkimo išvedama įvestų stebėjimų, kurioje yra ir naujai įvestas stebėjimas, aiš 49 pav. Naujai įvestas stebėjimas nufiltruotas, naudojant „Metai“ ir „Mėnuo“ filtrus.

Aktyviausi rajonai:

1. Ignalinos r. (28)
2. Švenčionių r. (26)
3. Varėnos r. (25)
4. Vilniaus r. (25)
5. Ukmergės r. (23)
6. Prienų r. (21)
7. Biržų r. (20)
8. Lazdijų r. (20)
9. Panevėžio r. (20)
10. Šalčininkų r. (20)

Šiandien užregistruota:
paukščių stebėjimų - 0
paukščių rūšių - 0

Iš viso registruota stebėjimų: 698

PAUKŠČIŲ STEBĖJIMAI

leškoti

Rajonas	Stebėjimo tipas	RK	IUCN	Metai	Mėnuo
1 Ignalinos r.	1 Pavieniai stebėjimai	1 0	1 LC	124 2009 <input type="checkbox"/>	1 Gegužė <input checked="" type="checkbox"/>
				140 2010 <input type="checkbox"/>	118 Sausis <input type="checkbox"/>
				1 2011 <input checked="" type="checkbox"/>	

[PAUKŠČIŲ STEBĖJIMO ŽEMĖLAPIS](#) • [COUNT MAP](#) • [TIMEPLOT](#)
• [CALENDAR](#) • [MINIATIŪROS](#) • [DETAILS](#)

Taksonomijos filtras
Papildomų parametų filtras



49 pav. Naujo stebėjimo nufiltravimas naudojant *Exhibit* įrankį

Taigi, šio pavyzdžio rezultatai leidžia teigti, kad informacinės sistemos sritis atsakinga už paukščių stebėjimo duomenų įvedimą ir atvaizdavimą veikia korektiškai. Analogišku būdu patikrinti ir kiti sistemos funkcionalumai, tai yra: autorizacija, nuotraukų galerija, statistinės informacijos atvaizdavimas, dienoraščių skiltis, naujienų skiltis ir taip toliau.

6. Eksperimentinis sistemos tyrimas

6.1. Eksperimento planas

Magistrinio darbo eksperimento tyrimui įgyvendinti buvo realizuota Lietuvos paukščių registravimo sistema. Šios sistemos kaupiamų duomenų analizei ir atvaizdavimui pritaikytas *Exhibit* įrankis. Magistrinio darbo metu buvo siekiama dvejopai praplėsti *Exhibit* įrankio galimybes, tai yra šį įrankį pritaikyti dinamiškai atsirandantiems duomenims analizuoti bei pagretinti *Exhibit* įrankio veikimą.

Taigi, eksperimentas susideda iš dviejų dalių: dinamiškai kintančių duomenų atvaizdavimo bei *Exhibit* įrankio greitaveikos padidėjimo patikrinimo. Eksperimento metu operuojama sugeneruota duomenų aibe, kuri buvo sugeneruota taip, kad kuo labiau atitiktų realiai suvedamų duomenų struktūrą ir sudėtį.

6.2. Eksperimento rezultatai

6.2.1. Dinamiškai kintančių duomenų atvaizdavimo su *Exhibit* įrankiu tyrimas

Kaip buvo minėta ankščiau, tyrimo realizavimui sukurta Lietuvos paukščių registravimo sistema. Informacinės sistemos duomenys saugomi suprojektuotoje duomenų bazėje 27 pav. Šios duomenų bazės, o kartu ir visos sistemos branduolį sudaro tos duomenų bazės lentelės, kurios atsakingos už paukščių stebėjimo duomenų saugojimą. Taigi, esminis *Exhibit* įrankio pritaikymas ir būtų sistemoje užregistruotų paukščių stebėjimų duomenų vizualinis atvaizdavimas su galimybe interaktyviai manipuluoti gauta duomenų aibe.

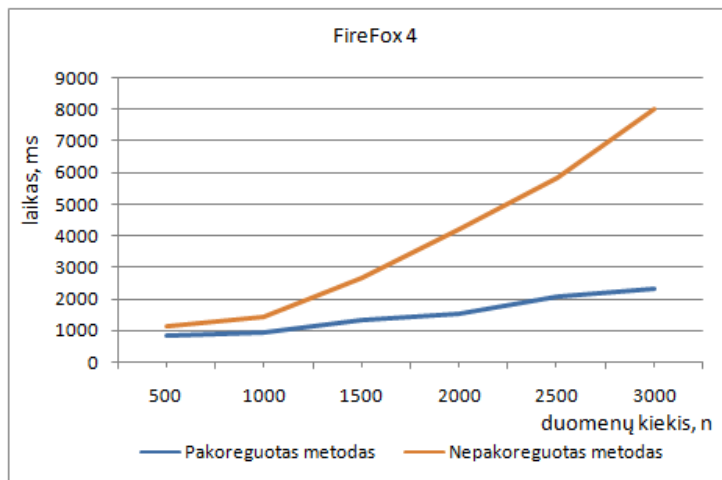
Informacinės sistemos testavimo skyriuje aprašyta paukščių stebėjimo duomenų įvedimo eiga. Šiuo pavyzdžiu siekta ne tik parodyti, kad korektiškai veikia duomenų įvedimas, *Exhibit* įrankis, bet taip pat akcentuoti, kad yra pasiūlytas ir išbandytas sprendimas, leidžiantis *Exhibit* įrankį panaudoti dinamiškai kintamiems duomenis vizualizuoti, filtruoti. Taigi testavimo eigą galima sutapatinti su eksperimentu, o gautus rezultatus įvertinti, kaip atitinkančius sistemai iškeltus dinamiškumo reikalavimus.

6.2.2. *Exhibit* įrankio greitaveikos tyrimas

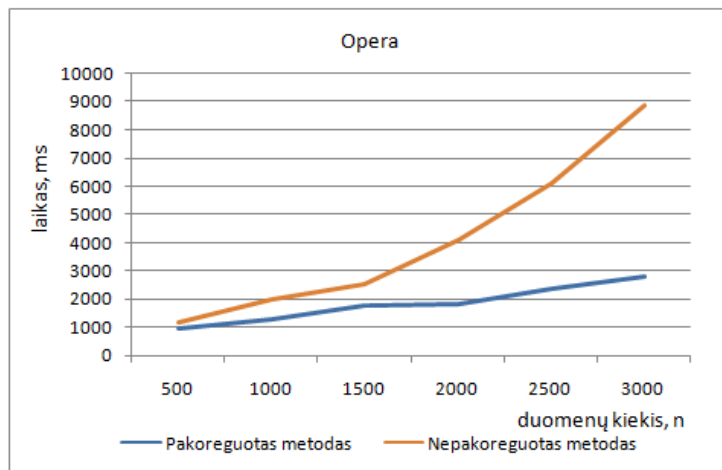
Šio eksperimento tikslas – nustatyti *Exhibit* duomenų struktūros suformavimo greičio skirtumus naudojant esamą *Exhibit* duomenų struktūros formavimo metodą ir naudojant pakoreguotą duomenų struktūros suformavimo metodą.

Žemiau esančiose 50 pav. – 52 pav. diagramose pateikta duomenų struktūros suformavimo greičio priklausomybė nuo duomenų kiekio bei duomenų formavimo metodo.

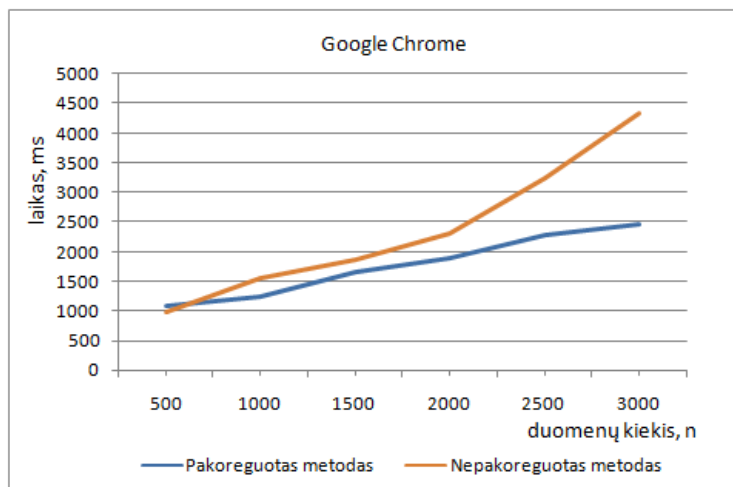
Mėlyna spalva pavaizduotas duomenų struktūros suformavimo greitis naudojant pakoreguotą metodą, o kita spalva vaizduojamas duomenų formavimo greitis ir jo priklausomybė nuo laiko naudojant standartinį metodą.



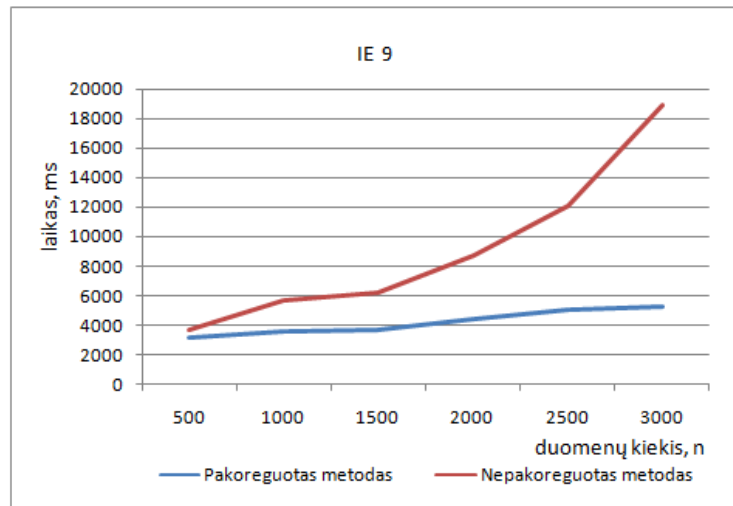
50 pav. Exhibit duomenų struktūros formavimo laikas naudojant Firefox 4 naršyklę



51 pav. Exhibit duomenų struktūros formavimo laikas naudojant Opera naršyklę

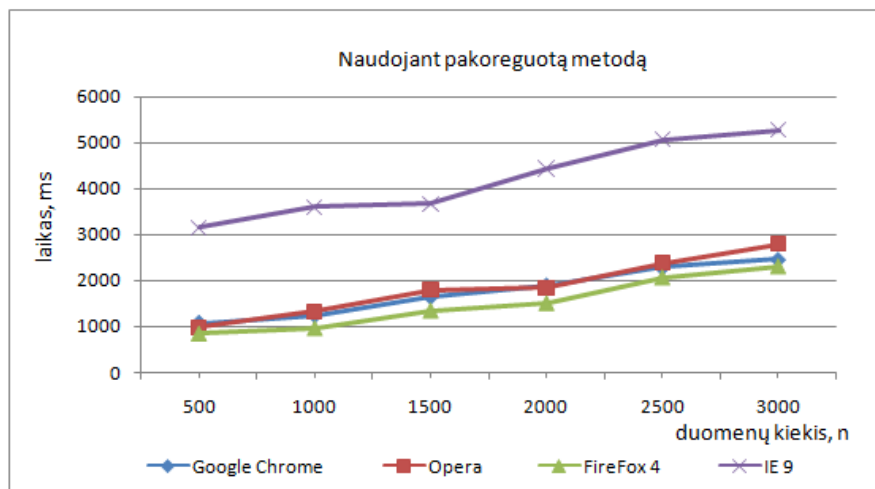


52 pav. Exhibit duomenų struktūros formavimo laikas naudojant Google Chrome naršyklę

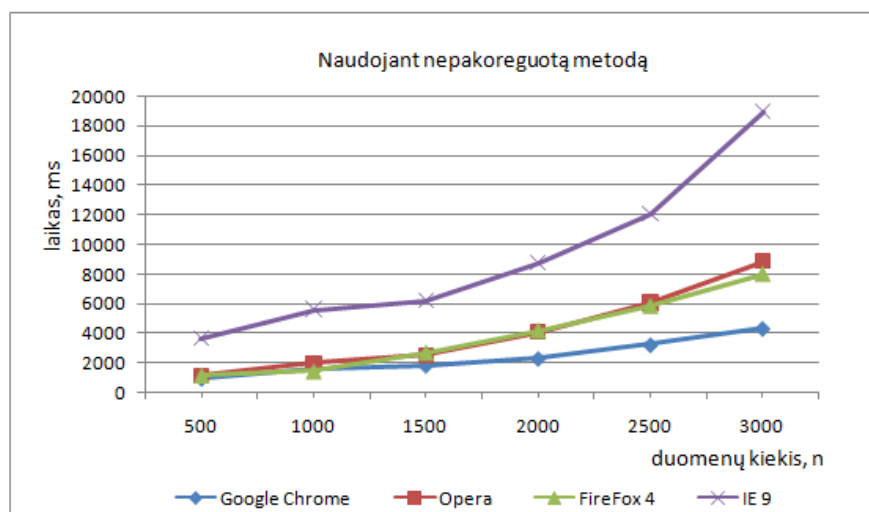


53 pav. Exhibit duomenų struktūros formavimo laikas naudojant Internet Explorer 9 naršyklę

54 pav. ir 55 pav. tarpusavyje palyginti skirtingų naršyklių veikimo greičių vidurkiai naudojant pakoreguotą ir nepakoreguotą duomenų struktūros formavimo metodą.



54 pav. Duomenų struktūros formavimo greitis tarp skirtingų naršyklių naudojant pakoreguota metodą



55 pav. Duomenų struktūros formavimo greitis tarp skirtingų naršyklių naudojant nepakoreguota metodą

6.3. Sistemos veikimo ir savybių analizė, kokybės kriterijų įvertinimas

Informacinės sistemos testavimo ir 13.2 skyriuje paminėta, kad pasiūlytas *Exhibit* duomenų formavimo sprendimas leidžiantis įrankyje atvaizduoti dinamiškai kintančią informaciją. Buvo realizuota galimybė informaciją gauti tiesiogiai iš duomenų bazės, suformavus SQL užklausą. Taigi, testavimo metu pavyko įrodyti, kad įvedus naują paukščių stebėjimą, iš karto galima jį peržiūrėti su *Exhibit* įrankiu.

Sekančio eksperimento atlikimui buvo naudojama *Windows 7* operacinė sistema, bei (*Intel (R) Celeron (R) CPU 2,66 GHz, 766 MB RAM*) techniniais parametrais pasižymintis kompiuteris. Tyrimo metu buvo eksperimentuojama su trimis interneto naršyklėmis, tai yra „*Google Chrome*“, „*Opera*“ ir „*Firefox 4*“.

Iškeltos hipotezės:

H_0 : $t_{\text{nepakoreguotas}} \leq t_{\text{pakoreguotas}}$

$H_1(H_a)$: $t_{\text{nepakoreguotas}} > t_{\text{pakoreguotas}}$

Nepriklausomi kintamieji:

○ Duomenų įterpimo metodai

1. Nepakoreguotas įterpimo metodas (56 pav.)
2. Pakoreguotas įterpimo metodas (57 pav.)

○ Internetinės naršyklės

1. Google Chrome
2. Opera
3. Firefox 4

○ Paukščių stebėjimų kiekis

Priklausomas kintamasis:

○ *Exhibit* duomenų struktūros suformavimo greitis

```
Exhibit.Database._indexPut = function(index, x, y, z) {
    var hash = index[x];
    if (!hash) {
        hash = {};
        index[x] = hash;
    }

    var array = hash[y];
    if (!array) {
        array = new Array();
        hash[y] = array;
    } else {
        for (var i = 0; i < array.length; i++) {
            if (z == array[i]) {
                return;
            }
        }
    }
    array.push(z);
};
```

56 pav. Nepakoreguotas metodas

```
Exhibit.Database._indexPut = function(index, x, y, z) {
    var hash = index[x];
    if (!hash) {
        hash = {};
        index[x] = hash;
    }
    var array = hash[y];
    if (!array) {
        array = new Array();
        hash[y] = array;
    }
    array.push(z);
};
```

57 pav. Pakoreguotas metodas

Eksperimento metu buvo apdorojami tokie paukščių stebėjimo duomenų kiekiai: 500, 1000, 1500, 2000, 2500, 3000. Vieno paukščių stebėjimo įrašo pavyzdys pateiktas 58 pav.

```
{
  "label": "1",
  "utm_sq": "LA1e2",
  "utm_xy": "54.6468985,24.1361093|54.6495142,24.2722186
           |54.5605849,24.2722186|54.5579692,24.1361093",
  "utm_c": "54.6037417,24.20416395",
  "district": "Prienų r.",
  "b_name": "Žalioji pečialinda",
  "order": "Nariniai",
  "family": "Nariniai",
  "rb": "0",
  "iucn": "LC",
  "date": "2010-07-16",
  "character": "LOFK pranešimas",
  "year": "2010",
  "month": "Liepa",
  "day": "16",
  "watcher": "rokas",
  "location": "MKAFTRFB",
  "amount": "10",
  "sex": "Nenustatyta",
  "age": "1cy+",
  "biotope": "Ežeras"
}
```

58 pav. Vieno paukščių stebėjimo įrašo struktūra

Siekiant gauti kuo tikslesnius rezultatus, eksperimentas esant tiems patiems parametrui buvo kartojamas 5 kartus. Taip pat reiktų pažymėti, kad eksperimento metu buvo skaičiuojami trys skirtingi laiko intervalai, tai yra *Exhibit* įrankio užkrovimo laikas, *Exhibit* duomenų struktūros suformavimo laikas, bei vartotojo sąsajos generavimo laikas. Šiam eksperimentui svarbiausias vidurinis laikas, kadangi jis tiesiogiai priklausomas nuo 56 pav. ir 57 pav. pateikto metodo veikimo spartos. Atlikto eksperimento duomenims įvertinti buvo naudojamas *Wilcoxon* testas, kuris paskaičiuotas naudojantis *SPSS*¹² (17.0 versijos) įrankiu. *Wilcoxon* testas leidžia palyginti dvi tarpusavyje susijusias duomenų imtis, kai yra nedidelis bandymų kiekis (mano atveju atlikti 5 bandymai). Vertinant rezultatų patikimumą, statistinėje analizėje remtasi $p=0,05$ reikšmingumo lygmeniu.

Atlikto eksperimento rezultatai leidžia spręsti, kad pakoreguotas *Exhibit* duomenų formavimo metodas leido gerokai paspartinti duomenų struktūros suformavimą (50 pav., 51 pav., 52 pav., 53 pav.). Naudojant *Google Chrome* naršyklę statistiškai reikšmingas skirtumas ($p<0,05$) nustatytas užkraunant 2000, 2500, 3000 stebėjimo įrašų kiekius. O užkraunant 500, 1000 ir 1500 statistiškai reikšmingas skirtumas nenustatytas ($p>0,05$). Eksperimentą atliekant su *Opera* naršykle statistiškai reikšmingas ($p<0,05$) skirtumas nustatytas užkraunant didesnius nei 1000 duomenų įrašų kiekius. Tačiau užkraunant 500 stebėjimų duomenų kiekį statistiškai reikšmingumas nenustatytas ($p>0,05$). Naudojant *Firefox 4* naršyklę, statistiškai reikšmingas skirtumas ($p<0,05$) nustatytas visose eksperimentiniuose duomenų kiekiuose. Atliekant

¹² *SPSS* (angl. *Statistical Package for the Social Science*)

eksperimentą su IE-9 naršykle statistiškai reikšmingas skirtumas ($p < 0,05$) nustatytas užkraunant 1000, 1500, 2000, 2500, 3000 stebėjimo įrašų kiekius. O užkraunant 500 stebėjimo įrašų kiekį, skirtumas tarp duomenų struktūros suformavimo greičių nėra reikšmingas ($p > 0,05$).

Gautų rezultatų analizė patvirtino, kad pakoreguotas metodas leido pagreiti duomenų struktūros formavimą. Taigi, patvirtinama $H_1(H_a)$: $t_{\text{nepakoreguotas}} > t_{\text{pakoreguotas}}$ hipotezė. Be to, gauti rezultatai leidžia pastebėti, kad kuo didesnis duomenų kiekis, tuo stipriau pasireiškia pakoreguoto metodo efektyvumas (50 pav. 51 pav., 52 pav., 53 pav.). Be to reiktų paminėti, kad didelę įtaką turi ne tik duomenų kiekis, bet ir kiek įrašų yra viename duomenų elemente, kokio tipo naršyklė naudojama (54 pav., 55 pav.).

Žemiau esančioje 4 lentelėje apibendrinta, kuriuos informacinės sistemos reikalavimus magistrinio darbo metu pavyko įgyvendinti. Taigi, *Exhibit* įrankį praplėtus duomenų bazės teikiamais privalumai pavyko išpildyti 1, 2, 3 informacinės sistemos reikalavimus. Būtent realizuotas duomenų suformavimo būdas leido pasiekti, kad vartotojui pateikiama informacija atitiktų reikalaujamą duomenų saugumą (išsamumą), privatumą ir dinamiškumą, tai yra priklausomai nuo vartotojo kategorijos, pateikiamas tam tikras informacijos detalumas bei manipuluojama nauja informacija. *Exhibit* įrankio pagalba realizuota išsamios duomenų analizės vartotojo sąsaja. *Exhibit* įrankio duomenų formavimo algoritmo analizė ir šio algoritmo sudėtingumo įvertinimas leido manyti, kad šio algoritmo patobulinimas pagreitins duomenų suformavimo laiką, o tuo pačiu metu atsiras galimybė manipuluoti didesniu kiekiu duomenų. Atliktas eksperimentas patvirtino atliktų įvertinimų teisingumą, ko pasekoje atsirado galimybė manipuluoti didesniu kiekiu duomenų. Taigi, ketvirtasis realizuotos informacinės sistemos reikalavimas išpildytas dalinai, kadangi suteikta galimybė dirbti su didesniu nei įprastai duomenų kiekiu. Naudojant labai didelius duomenų kiekius, tai yra 10000 įrašų ir daugiau, vartotojui teks ilgai laukti, kol bus suformuota vidinė duomenų struktūra bei grafinė vartotojo sąsaja.

4 lentelė Reikalaujamų informacinės sistemos savybių išpildymas

<i>Nr.</i>	<i>Reikalaujamos sistemos savybės</i>	<i>Ar įgyvendinta</i>
1.	Atvaizduojamų duomenų saugumas, privatumas	taip
2.	Manipuluojama dinamiškai kintančia informacija	taip
3.	Galimybė duomenis peržiūrėti įvairiais pjūviais bei interaktyviomis duomenų atvaizdavimo formomis	taip
4.	Galimybė dirbti su dideliu kiekiu duomenų	iš dalies

6.4. Sistemos taikymo rekomendacijos

Atliktų eksperimentų rezultatai leidžia praplėsti iki tol buvusią *Exhibit* įrankio panaudojimo sritį. Be ankščiau buvusių *Exhibit* įrankio privalumų, atsiranda galimybė manipuluoti dinamiškai kintančia informacija, o sutrumpintas duomenų struktūros formavimo greitis leidžia manipuluoti didesniu kiekiu duomenų. Tačiau, kaip buvo paminėta ankščiau, nemažą įtaką turi duomenų sudėtis, tai yra duomenų elementų kiekis, elemento atributinių įrašų kiekis, naudojamos naršyklės tipas. Be to, *Exhibit* įrankis ir toliau gali būti tobulinamas, tai yra optimizuojamas šio įrankio veikimo algoritmas.

7. Išvados

1. *Exhibit* įrankio panaudojimo trūkumai: neužtikrinamas duomenų privatumas, duomenys saugomi statinėje formoje, ilgai formuojama vidinė duomenų struktūra.
2. *Exhibit* įrankio panaudojimas leido sukurti vartotojui priimtinesnį ir patogesnį duomenų nagrinėjimą.
3. *Exhibit* duomenų formavimo apjungimas su duomenų baze leidžia manipuliuoti dinamiškai kintančia informacija ir išlaikyti pageidaujamą informacijos privatumą.
4. *Exhibit* duomenų formavimo metodo pakoregavimas, leido pagreitinti šio įrankio vidinės duomenų struktūros suformavimą.
5. Naudojant *Symfony* karkasą sukurta Lietuvos ornitofaunos registravimo sistema, o joje kaupiamų duomenų vizualizavimui ir analizei pritaikytas *Exhibit* įrankis.
6. Eksperimento metu įsitikinta, kad *Exhibit* įrankis geba manipuliuoti dinamiškai kintančia informacija bei dirbti su didesniu kiekiu informacijos.
7. *Wilcoxon* testo rezultatai patvirtino, kad formuojant vidinę duomenų struktūrą greičių skirtumas naudojant pakoreguotą ir nepakoreguotą metodą yra statistiškai reikšmingas, kitaip sakant $p < 0,05$.

8. Literatūra

1. Alexander, K., (2008) RDF/JSON: A Specification for Serialising RDF in JSON, SFSW2008 (Scripting for the Semantic Web).
2. Anslow, C., Noble, J., Tempero, E. (2009) Web Software Visualization Via Google's Visualization API, NZCSRSC 2009 Auckland, New Zealand.
3. Erle, S., Gibson R. (2006) Google Maps Hacks, United States of America
4. Few, S. (2009) Introduction to Geographical Data Visualization, Visual Business Intelligence Newsletter.
5. Flamenco įrankį pristatantis tinklapis. [interaktyvus] [žiūrėta 2011-05-15]. Prieiga per internetą: <http://flamenco.berkeley.edu/>
6. Grove, M., (2008) Case Study: POPS – NASA's Expertise Location Service Powered by Semantic Web Technologies [interaktyvus] [žiūrėta 2011-05-15]. Prieinamas per internetą <http://www.w3.org/2001/sw/sweo/public/UseCases/Nasa/>
7. Harris, C., Owens, A., Russel, A. and Smith, D. A. (2004) mSpace: Exploring The Semantic Web, Technical Report UNSPECIFIED, ECS, University of Southampton.
8. Hildebrand, M., Ossenbruggen, J., Hardman, L. (2006) /facet: A Browser for Heterogeneous Semantic Web Repositories, The Semantic Web – ISWC 2006.
9. Hearst, M. A. (2008) UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In: Workshop on Computer Interaction and Information Retrieval
10. Huynh, D.F. (2007) User Interfaces Supporting Casual Data-Centric Interaction on the Web, Massachusetts Institute of Technology.
11. Huynh, D.F., Karger, D.R., Miller, R.C. (2007) Exhibit: Lightweight Structured Data Publishing, MIT Computer Science and Artificial Intelligence Laboratory.
12. mSpace įrankį pristatantis tinklapis. [interaktyvus] [žiūrėta 2011-05-15]. Prieiga per internetą: <http://mspace.fm/>
13. schraefel, m.c., Smith, D. A., Russel, A., Owens, A., Harris, C., Wilson, M. L., (2005) The Evolving mSpace Platform: Leveraging the Semantic Web on the trail of the Memex, Hypertext 5th konferencija, Zalcburgas.
14. schraefel, m.c., Smith, D. A., Russel, A., Owens, A., Harris, C., Wilson, M. L., (2005) The mSpace Classical Music Explorer: Improving Access to Classical Music for Real People, V MUSICNETWORK OPEN WORKSHOP: Integration of Music in Multimedia Applications, July 4 and 5, 2005, Vienna, Austria.

15. Smith, D.A., Owens, A., schraefel, m.c., Sinclair, P., Andre, P., Wilson, M.L, Russell, K., Lewis, P., (2007) Challenges in Supporting Faceted Semantic Browsing of Multimedia Collections, *SAMT2007*, Genuja, Italija
16. Smith, D.A., schraefel, m.c., (2009) Interactively using Semantic Web knowledge: Creating scalable abstractions with FacetOntology, *WWW2009*.
17. Zhao, J., Klyne, G., Shotton, D., (2008) Building a Semantic Web Image Repository for Biological Research Images, 5th European Semantic Web Conference, Tenerifè, Ispanija

9. Priedai

1 Priedas. Kompiuterizuojamų panaudos atvejų specifikacija

Panaudojimo atvejis 1: Registruotis sistemoje

Aktorius: sistemos lankytojas.

Tikslas: suteikti galimybę įvesti paukščių stebėjimo duomenis, peržiūrėti išsamesnę įvestų duomenų informaciją, pildyti stebėjimo dienoraščius.

Ryšiai su kitais PA: naudoja „Prisijungimo duomenis nusiųsti į elektroninį paštą“ ir yra praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas sėkmingai atsidaro informacinės paukščių registracijos sistemos tinklalapį.

Sužadinimo sąlyga: vartotojas pasirinko registracijos formos pildymo meniu punktą.

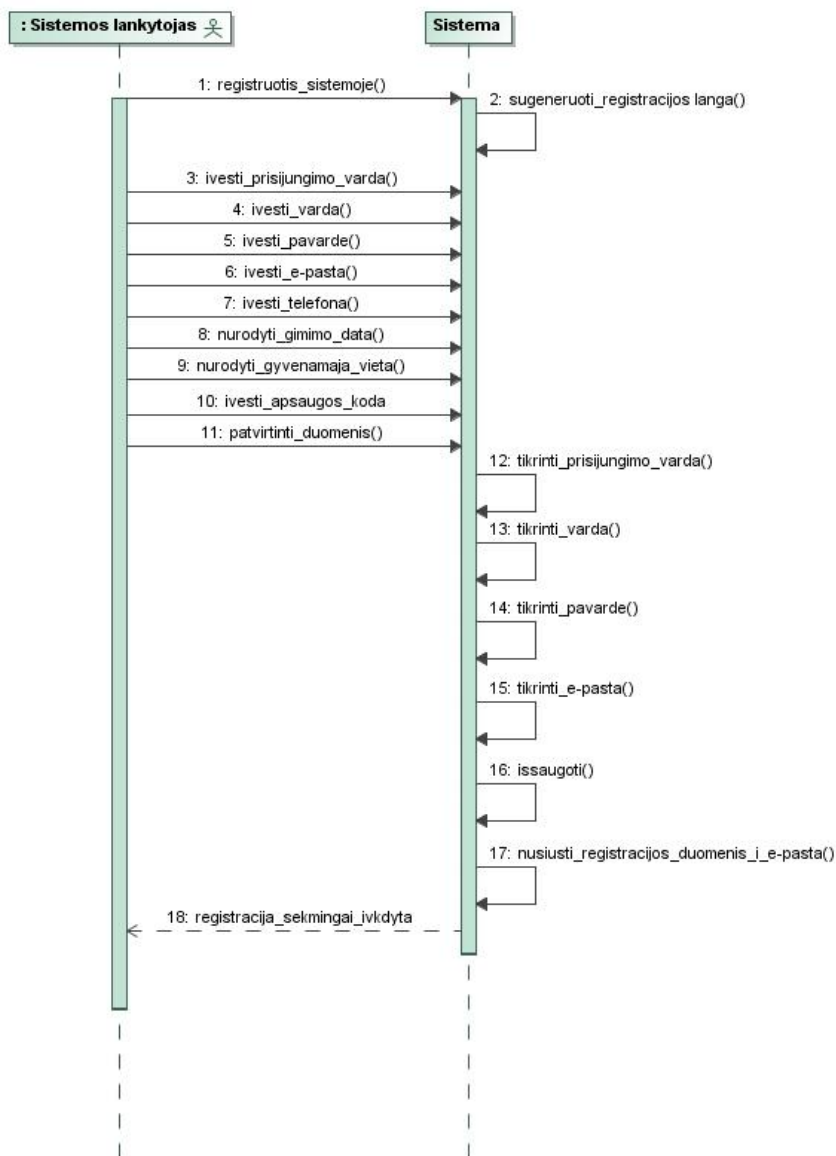
Po-sąlyga: vartotojas užregistruojamas sistemoje.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia nuorodą, kuri nekreipia į registracijos formos pildymo langą;
2. Į registracijos formą suvedami visi privalomi duomenys;
3. Vartotojui į elektroninį paštą nusiunčiami prisijungimo duomenys ir vartotojas prieregistruojamas sistemoje;

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka duomenų išsaugojimo funkcijos;



1 pav. Registravimosi sistemoje sekų diagrama

Panaudojimo atvejis 2: Prisijungimo duomenis nusiųsti į elektroninį paštą

Aktorius: sistemos lankytojas.

Tikslas: suteikti galimybę vartotojui prisijungti prie sistemos.

Ryšiai su kitais PA: -

Prieš-sąlygos: vartotojas sėkmingai suveda asmeninę informaciją.

Sužadinimo sąlyga: vartotojas pasirinko registracijos patvirtinimo funkciją.

Po-sąlyga: vartotojas užregistruojamas sistemoje ir išsiunčiamas pranešimas.

Pagrindinis scenarijus:

1. Vartotojas įveda registracijos duomenis;
2. Pasirenkama funkcija, kuri registruoja vartotoją;
3. Išsiunčiamas pranešimas;

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka duomenų išsaugojimo funkcijos;

Panaudojimo atvejis 3: Peržiūrėti paukščių stebėjimus

Aktorius: sistemos lankytojas.

Tikslas: suteikti galimybę peržiūrėti paskutiniuosius 10 stebėjimų, pateikiant tam tikros apimties informaciją, kuri tiesiogiai susijusi su paukščio rūšies paplitimu Lietuvos teritorijoje.

Ryšiai su kitais PA: -

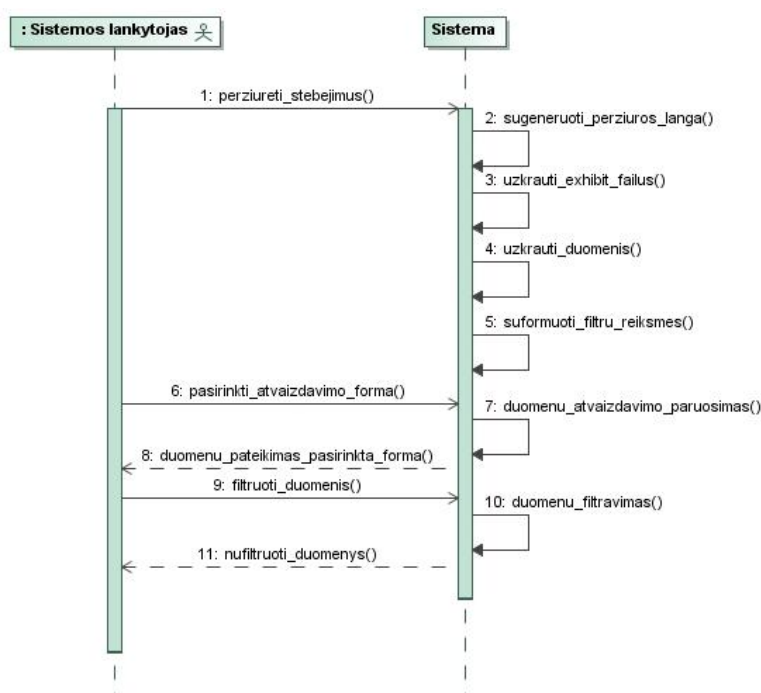
Prieš-sąlygos: vartotojas sėkmingai atsidaro informacinės paukščių registracijos sistemos tinklalapį.

Sužadinimo sąlyga: vartotojas pasirinko registruotų duomenų peržiūros meniu punktą.

Po-sąlyga: vartotojui atveriamas įvestų duomenų peržiūros langas.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia nuorodą, kuri nekreipia į įvestų duomenų peržiūros langą;
2. Vartotojui suteikiama galimybė įvairiais pjūviais peržiūrėti informaciją, ją filtruoti, grupuoti ir kitaip analizuoti;



2 pav. Įvestų paukščių stebėjimo duomenų peržiūrose sekų diagrama

Panaudojimo atvejis 4: Prisijungti prie sistemos

Aktorius: sistemos lankytojas.

Tikslas: suteikti galimybę įvesti paukščių stebėjimo duomenis, peržiūrėti išsamesnę įvestų duomenų informaciją, pildyti stebėjimo dienoraščius.

Ryšiai su kitais PA: praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas sėkmingai atsidaro informacinės paukščių registracijos sistemos tinklalapį.

Sužadinimo sąlyga: vartotojas pasirinko prisijungimo formos pildymo funkciją.

Po-sąlyga: vartotojas prijungimas prie sistemos.

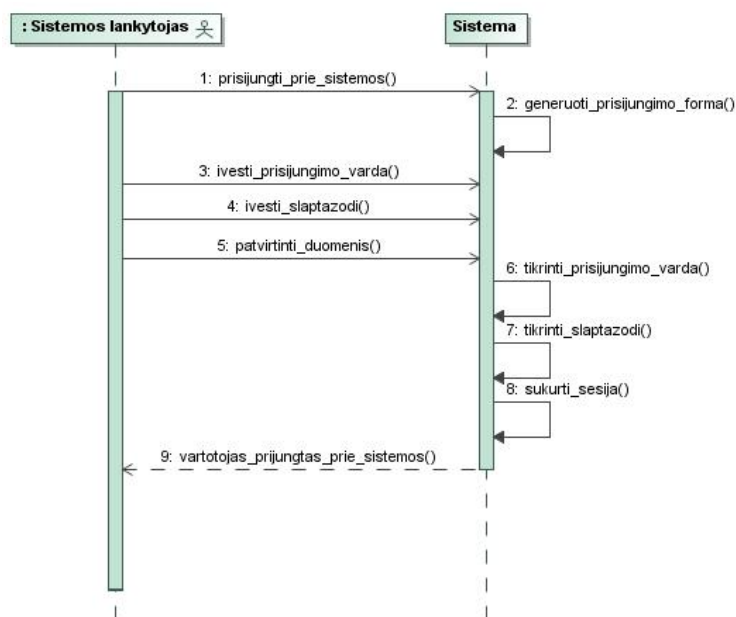
Pagrindinis scenarijus:

1. Vartotojas paspaudžia nuorodą, kuri nekreipia į prisijungimo formos pildymo langą;
2. Į prisijungimo formą suvedami visi privalomi duomenys;
3. Vartotojas prijungiamas prie sistemos;

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;

- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka prijungimo prie sistemos funkcijos;



3 pav. Prisijungimo prie sistemos sekų diagrama

Panaudojimo atvejis 5: Priminti prisijungimo duomenis

Aktorius: sistemos lankytojas.

Tikslas: atstatyti galimybę įvesti paukščių stebėjimo duomenis, peržiūrėti išsamesnę įvestų duomenų informaciją, pildyti stebėjimo dienoraščius.

Ryšiai su kitais PA: praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas prieš tai buvo sėkmingai užsiregistravęs sistemoje, bet dėl tam tikrų priežasčių negali prisijungti prie sistemos.

Sužadinimo sąlyga: vartotojas pasirinko prisijungimo duomenų priminimo funkciją.

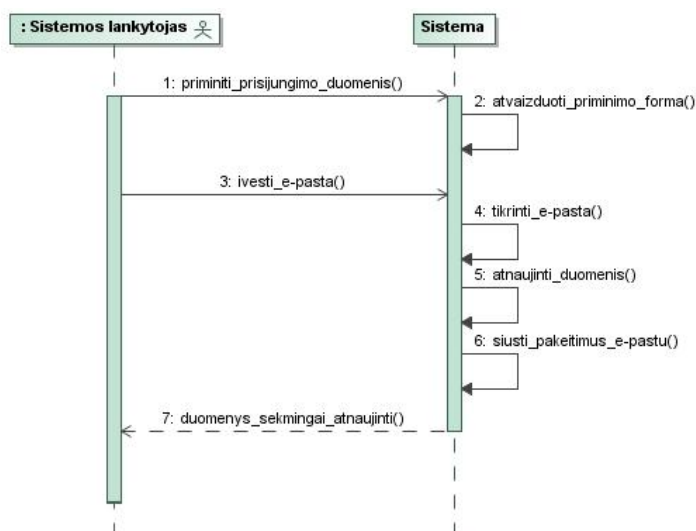
Po-sąlyga: vartotojui į elektroninį paštą nusiunčiama atnaujinta prisijungimo duomenų informacija.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia nuorodą, kuri atidaro prisijungimo duomenų priminimo formos pildymo langą;
2. Į prisijungimo formą suvedamas vartotojo elektroninio pašto adresas;
3. Vartotojui į elektroninį paštą nusiunčiamas naujai sugeneruoti reikiami duomenys;
4. Atnaujinami duomenų bazėje esantys vartotojo prisijungimo duomenys.

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka duomenų priminimo funkcijos;



4 pav. Prisijungimo duomenų priminimo sekų diagrama

Panaudojimo atvejis 6: Paukščių stebėjimo duomenų įvedimas

Aktorius: registruotas dalyvis.

Tikslas: suteikti galimybę įvesti paukščių stebėjimo duomenis.

Ryšiai su kitais PA: praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas sėkmingai prisijungia prie sistemos.

Sužadinimo sąlyga: vartotojas pasirinko paukščių stebėjimų registracijos formos pildymo meniu punktą.

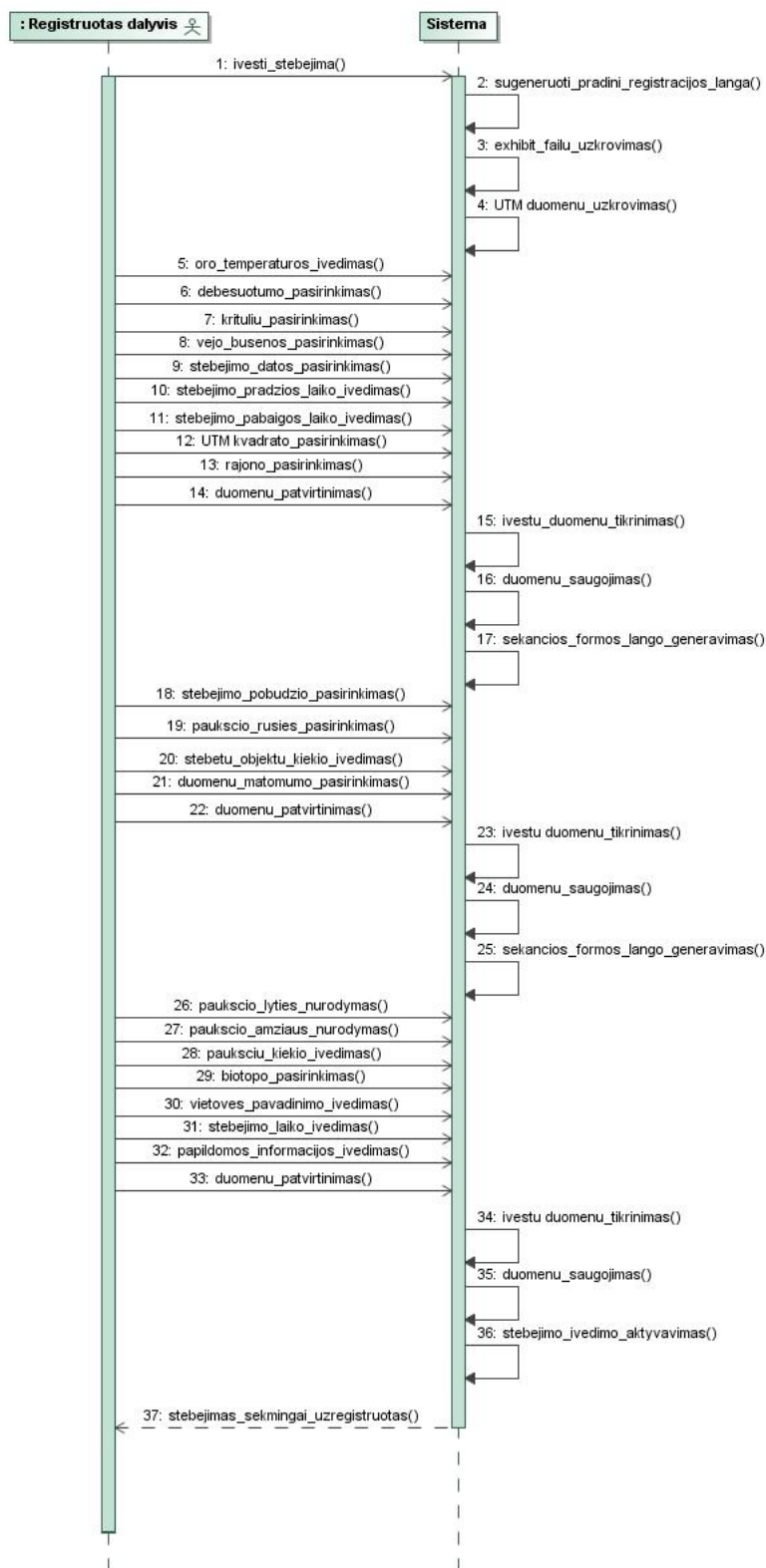
Po-sąlyga: sistemos duomenų bazėje išsaugojami įvesti duomenys.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia nuorodą, kuri nekreipia į pradinę paukščių registracijos formos dalį. Įvedama informacija apie teritoriją ir stebėjimo metu esančias oro sąlygas;
2. Vartotojas nukreipiamas į sekantį registracijos formos pildymo langą, kuriame pasirenkamas stebėjimo pobūdis ir apie kokią rūšį įvedama informacija;
3. Vartotojas nukreipiamas į galutinę registracijos formos dalį. Įvedama informacija apie konkretų nurodytos rūšies individą;
4. Sistemoje registruojamas sėkmingas stebėjimas;

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka duomenų išsaugojimo funkcijos;
- Toliau pildoma informacija apie einamąją paukščio rūšį;
- Toliau pildoma informacija apie einamojoje teritorijoje pastebėtas paukščių rūšis;
- Pildomas stebėjimas naujoje teritorijoje;
- Vartotojas nepasirenka duomenų išsaugojimo funkcijos;



5 pav. Paukščių stebėjimo duomenų įvedimo sekų diagrama

Panaudojimo atvejis 7: Pildyti stebėjimo dienoraštį

Aktorius: registruotas dalyvis.

Tikslas: suteikti galimybę įvesti įdomią, intriguojančią ar tam tikrus pastebėjimus bylojančią informaciją, kuri susijusi paukščių stebėjimais.

Ryšiai su kitais PA: praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas sėkmingai prisijungęs prie sistemos.

Sužadinimo sąlyga: vartotojas pasirinko dienoraščio pildymo meniu punktą.

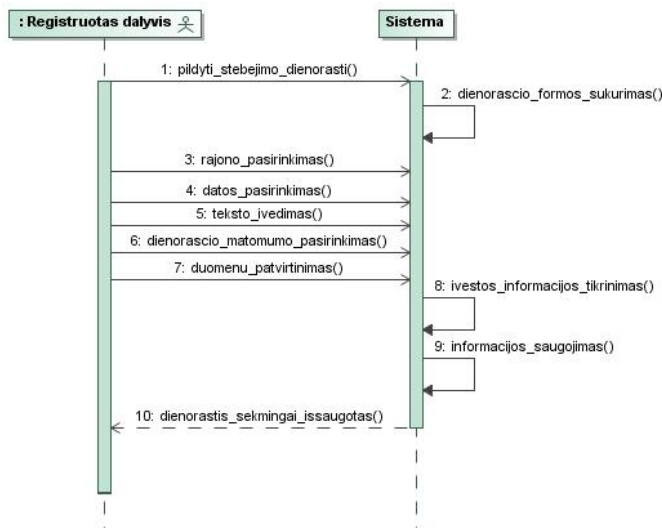
Po-sąlyga: dienoraščio informacija išsaugoma duomenų bazėje.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia meniu punktą, kuria atidaro dienoraščio pildymo langą;
2. Suvedama informacija į privalomus laukus;
3. Išsaugoma suvesta informacija;
4. Įvesta informacija atvaizduojama įvestų dienoraščių sąrašė.

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Nėra užpildyti visi privalomi formos laukai;
- Vartotojas nepasirenka duomenų priminimo funkcijos;



6 pav. Paukščių stebėjimo dienoraščio pildymo sekų diagrama

Panaudojimo atvejis 8: peržiūrėti informaciją apie registruotų dalyvių veiklą

Aktorius: registruotas dalyvis.

Tikslas: suteikti galimybę peržiūrėti kitų vartotojų statistinę informaciją.

Ryšiai su kitais PA: -.

Prieš-sąlygos: vartotojas sėkmingai prisijungęs prie sistemos.

Sužadinimo sąlyga: vartotojas pasirinko statistikos peržiūros meniu punktą.

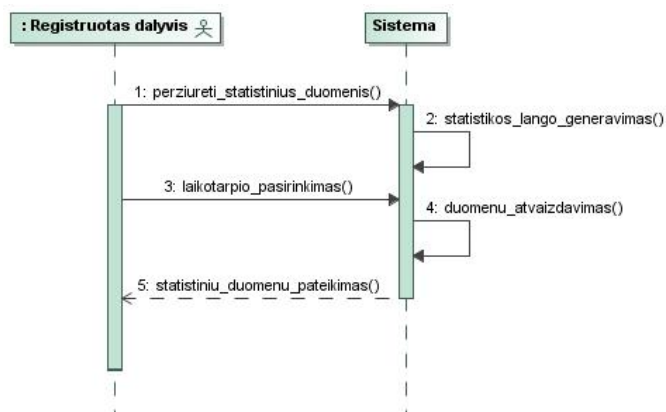
Po-sąlyga: pateikiama statistinė informacija apie tam tikrą stebėjimo pobūdį ir laikotarpį.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia meniu punktą, kuria atidaro statistikos peržiūros langą;
2. Pasirenkama, apie kurį stebėjimo pobūdį norima gauti informaciją;
3. Pasirenka, kurių metų informacija būtų pateikiama;
4. Pateikiama statistinė informacija.

Alternatyvūs scenarijai:

- Vartotojas nepasirenka stebėjimo pobūdžio;



7 pav. Statistinių duomenų peržiūros, apie registruotų dalyvių veiklą, sekų diagrama

Panaudojimo atvejis 9: Koreguoti asmeninius duomenis

Aktorius: registruotas dalyvis.

Tikslas: suteikti galimybę koreguoti įvestus asmeninius duomenis.

Ryšiai su kitais PA: praplėstas panaudos atveju „Perspėti“.

Prieš-sąlygos: vartotojas sėkmingai prisijungęs prie sistemos.

Sužadinimo sąlyga: vartotojas pasirinko asmeninės informacijos peržiūros meniu punktą.

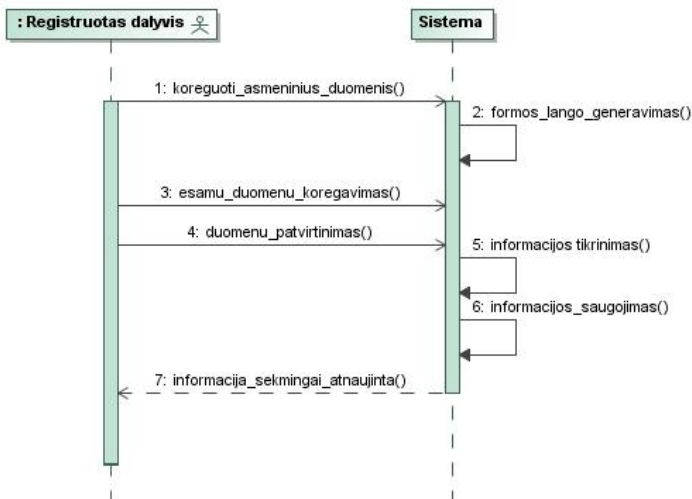
Po-sąlyga: atnaujinama asmeninė informacija.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia meniu punktą, kuris atidaro asmeninių duomenų peržiūros langą;
2. Koreguojami norimi įvestų duomenų formos laukų turiniai;
3. Išsaugomi pakeitimai.

Alternatyvūs scenarijai:

- Į formos laukus įvedami nekorektiški duomenys;
- Vartotojas nepasirenka duomenų atnaujinimo funkcijos;



8 pav. Asmeninių duomenų koregavimo sekų diagrama

Panaudojimo atvejis 10: Detali registruotų paukščių informacijos peržiūra

Aktorius: LOD tarybos narys.

Tikslas: suteikti galimybę peržiūrėti visą įmanomą suvestų duomenų informaciją, kuri susijusi užregistruotais paukščiu stebėjimais.

Ryšiai su kitais PA: -

Prieš-sąlygos: vartotojas sėkmingai prisijungęs prie sistemos ir identifikuojamas kaip LOD tarybos narys.

Sužadinimo sąlyga: vartotojas pasirinko įvestos informacijos peržiūros meniu punktą.

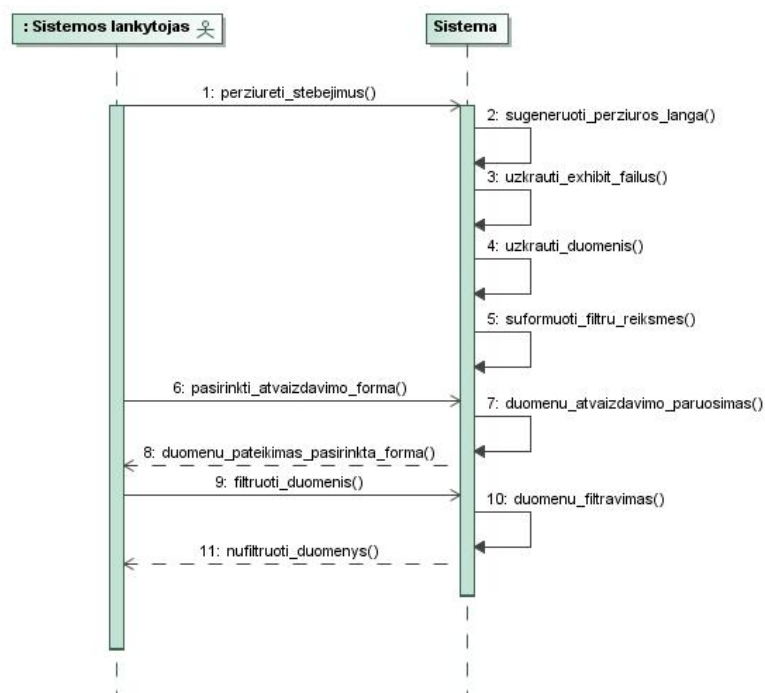
Po-sąlyga: pateikiama išsami įvestų paukščių stebėjimo duomenų informacija.

Pagrindinis scenarijus:

1. Vartotojas paspaudžia meniu punktą, kuria atidaro įvestų duomenų peržiūros langą;
2. Vartotojui suteikiama galimybė įvairiais pjūviais peržiūrėti informaciją, ją filtruoti, grupuoti ir kitaip analizuoti;

Alternatyvūs scenarijai:

- -



9 pav. Įvestų paukščių stebėjimo duomenų peržiūrose sekų diagrama

Panaudojimo atvejis 12: Perspėti

Aktorius: sistemos lankytojas, registruotas dalyvis.

Tikslas: suteikti galimybę išvesti informacinius pranešimus apie nekorektiškus duomenis.

Ryšiai su kitais PA: praplečia kitus PA atvejus.

Prieš-sąlygos: sistemoje nurodytos sąlygos, kurias turi tenkinti tam tikri informacijos įvedimo laukai.

Sužadinimo sąlyga: vartotojas įveda nekorektišką informaciją ir ją mėgina išsaugoti.

Po-sąlyga: išvedamas pranešimas.

Pagrindinis scenarijus:

1. Vartotojas įvedinėja duomenis į tam tikras informacijos įvedimo formas;
2. Išsaugomi pakeitimai;

Alternatyvūs scenarijai:

- Įvedami korektiški duomenys;

2 Priedas. Suprojektuotos duomenų bazės lentelių atributų aprašymas

1 lentelė Duomenų bazės lentelių atributai ir jų paskirtis

Atributas	Tipas	Paskirtis
<i>vartotojas</i>		
id	bigint(20)	Pirminis raktas
log_name	varchar(50)	Unikalus vartotojo prisijungimo vardas
password	varchar(255)	Slaptažodis
name	varchar(255)	Vartotojo vardas
surname	varchar(255)	Vartotojo pavardė
email	varchar(255)	Unikalus vartotojo e-pašto adresas
phone	varchar(12)	Telefono numeris
birth	varchar(10)	Vartotojo gimimo diena
residence	varchar(255)	Vartotojo gyvenamoji vieta
category	varchar(255)	Vartotojo kategorija
<i>dienorastis</i>		
id	bigint(20)	Pirminis raktas
text	text	Dienoraščio tekstas
visibility	tinyint(1)	Dienoraščio matomumo identifikatorius
date	varchar(10)	Dienoraščio parašymo diena
vartot_id	bigint(20)	Vartotojo identifikatorius
raj_id	bigint(20)	Rajono identifikatorius
created_at	datetime	Įrašo sukūrimo data
updated_at	datetime	Įrašo atnaujinimo data
<i>dkomentaras</i>		
id	bigint(20)	Pirminis raktas
text	text	Komentaro tekstas
vartot_id	bigint(20)	Vartotojo identifikatorius
dnrst_id	bigint(20)	Dienoraščio identifikatorius
vartot_id	bigint(20)	Vartotojo identifikatorius
created_at	datetime	Įrašo sukūrimo data
updated_at	datetime	Įrašo atnaujinimo data
<i>pranesimas</i>		
id	bigint(20)	Pirminis raktas
whom	bigint(20)	Kuriam vartotojui skirta informacija
text	text	Pranešimo tekstas
vartot_id	bigint(20)	Vartotojo identifikatorius
created_at	datetime	Įrašo sukūrimo data
updated_at	datetime	Įrašo atnaujinimo data
<i>stebėjimas</i>		
id	bigint(20)	Pirminis raktas
temperature	Int(11)	Temoeratūra
cloud	bigint(20)	Debesuotumo identifikatorius
rain	bigint(20)	Kritulių būsenos identifikatorius
wind	bigint(20)	Vėjo būsenos identifikatorius

Atributas	Tipas	Paskirtis
date	varchar(10)	Stebėjimo data
start	time	Stebėjimo pradžios laikas
finish	time	Stebėjimo pabaigos laikas
vardot_id	bigint(20)	Vartotojo identifikatorius
utm_id	bigint(20)	UTM kvadrato identifikatorius
raj_id	bigint(20)	Rajono identifikatorius
is_active	tinyint(1)	Stebėjimo korektiškumo identifikatorius
created_at	datetime	Įrašo sukūrimo data
updated_at	datetime	Įrašo atnaujinimo data
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
<i>debesuotumas</i>		
id	bigint(20)	Pirminis raktas
<i>debesuotumas_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Debesuotumo pobūdis
lang	char(2)	Kalbos ISO kodas
<i>krituliai</i>		
id	bigint(20)	Pirminis raktas
<i>krituliai_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Kritulių pobūdis
lang	char(2)	Kalbos ISO kodas
<i>vejas</i>		
id	bigint(20)	Pirminis raktas
<i>vejas_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Vėjo buvimo pobūdis
lang	char(2)	Kalbos ISO kodas
<i>rajonas</i>		
id	bigint(20)	Pirminis raktas
district	varchar(20)	Rajono pavadinimas
xy	text	Rajono koordinacijų eilutė
<i>UTM</i>		
id	bigint(20)	Pirminis raktas
sq_name	varchar(10)	UTM kvadrato kodas
xy	text	UTM koordinacijų eilutė
raj_id	bigint(20)	Rajono identifikatorius
<i>steb_objektas</i>		
id	bigint(20)	Pirminis raktas
steb_id	varchar(10)	Stebėjimo identifikatorius
charact	bigint(20)	Stebėjimo pobūdžio identifikatorius
<i>steb_objektas</i>		
id	bigint(20)	Pirminis raktas
steb_id	varchar(10)	Stebėjimo identifikatorius
charact	bigint(20)	Stebėjimo pobūdžio identifikatorius

Atributas	Tipas	Paskirtis
<i>steb_podudis</i>		
id	bigint(20)	Pirminis raktas
<i>steb_podudis_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Paukščių stebėjimo pobūdis
lang	char(2)	Kalbos ISO kodas
slug	varchar(100)	Stebėjimo pobūdžio pavadinimas, kuris skirtas nuorodai formuoti
<i>obj_duom</i>		
id	bigint(20)	Pirminis raktas
pauks_id	bigint(20)	Paukščio pavadinimo identifikatorius
amount	bigint(20)	Stebėtų paukščių ar kitų objektų kiekis
steb_id	bigint(20)	Stebėjimo objekto identifikatorius
visibility	tinyint(1)	Stebėjimo duomenų matomumas
a_visibility	boolean	Administratoriaus lygio matomumo ribojimas
<i>paukstis</i>		
id	bigint(20)	Pirminis raktas
red_book	varchar(255)	Raudonosios kygos kategorija
lofk	varchar(255)	LOFK kategorija
iucn	varchar(255)	IUCN kategorija
visibility	tinyint(1)	Stebėjimo duomenų matomumas
<i>paukstis_translation</i>		
id	bigint(20)	Pirminis raktas
b_order	varchar(255)	Būrys, kuriam priklauso paukštis
family	varchar(255)	Šeima, kuriam priklauso paukštis
title	varchar(255)	Paukščio pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>individas</i>		
id	bigint(20)	Pirminis raktas
sex	bigint(20)	Paukščio lyties identifikatorius
age	bigint(20)	Paukščio amžiaus identifikatorius
amount	bigint(20)	Stebėtų paukščių kiekis
biotope	bigint(20)	Biotopo identifikatorius
terrain	varchar(255)	Vietovės pavadinimas
time	time	Paukščio pastebėjimo laikas
see	tinyint(1)	Identifikatorius, nusakantis ar paukštis buvo stebėtas
hear	tinyint(1)	Identifikatorius, nusakantis ar paukštis buvo išgirstas
mattingcall	tinyint(1)	Identifikatorius, nusakantis ar buvo stebėtos paukščio tuoktuvės
see_off	tinyint(1)	Kokiu atastumu buvo stebėtas paukštis
hear_off	tinyint(1)	Kokiu atastumu buvo girdėtas paukštis
fly	tinyint(1)	Identifikatorius, nusakantis ar

Atributas	Tipas	Paskirtis
		paukštis skrido
course	tinyint(1)	Paukščio skridimo krypties identifikatorius
rest	tinyint(1)	Identifikatorius, nusakantis ar paukštis ilsėjosi
mating	tinyint(1)	Identifikatorius, nusakantis ar paukštis atleika tuoktuvių ritualą
comment	text	Komentaras
picture	varchar(50)	Nuotraukos pavadinimas
steb_id	bigint(20)	Stebėjimo objekto duomenų identifikatorius
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
<i>lytis</i>		
id	bigint(20)	Pirminis raktas
<i>Lytis_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Lyties pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>amzius</i>		
id	bigint(20)	Pirminis raktas
<i>amzius_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Amžiaus pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>skridimo_kryptis</i>		
id	bigint(20)	Pirminis raktas
<i>skridimo_kryptis_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Skridimo krypties pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>biotopas</i>		
id	bigint(20)	Pirminis raktas
<i>biotopas_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Biotopo pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>lizdas</i>		
id	bigint(20)	Pirminis raktas
amount	bigint(20)	Stebėtų lizdų kiekis
biotope	bigint(20)	Biotopo identifikatorius
terrain	varchar(255)	Vietovės pavadinimas
nest_location	bigint(20)	Lizdo padėties identifikatorius
plant_title	varchar(100)	Augalo ant kurio susuktas lizdas pavadinimas
altitude	smallint(6)	Lizdo sukrovimo aukštis
usability	bigint(20)	Lizdo naudojamumo identifikatorius
nest_height	bigint(20)	Lizdo aukštis

Atributas	Tipas	Paskirtis
nest_condition	bigint(20)	Lizdo padėties identifikatorius
egg_amount	mediumint(9)	Kiaušinių skaičius lizde
jun_amount	mediumint(9)	Jauniklių skaičius lizde
comment	text	Komentaras
picture	varchar(50)	Nuotraukos pavadinimas
steb_id	bigint(20)	Stebėjimo objekto duomenų identifikatorius
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
<i>lizdaviente</i>		
id	bigint(20)	Pirminis raktas
<i>lizdaviete_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Lizdo sukrovimo vietos pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>uzimtumas</i>		
id	bigint(20)	Pirminis raktas
<i>uzimtumas_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Lizdo užimtumo būseną
lang	char(2)	Kalbos ISO kodas
<i>lizdo_padetis</i>		
id	bigint(20)	Pirminis raktas
<i>Lizdo_padetis_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(100)	Lizdo padėties pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>tuokviete</i>		
id	bigint(20)	Pirminis raktas
amount	bigint(20)	Stebėtų tuokviečių kiekis
biotope	bigint(20)	Biotopo identifikatorius
terrain	varchar(255)	Vietovės pavadinimas
usable	tinyint(1)	Tuokvietės naudojamumo identifikatorius
comment	text	Komentaras
picture	varchar(50)	Nuotraukos pavadinimas
steb_id	bigint(20)	Stebėjimo objekto duomenų identifikatorius
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
<i>statine_informacija</i>		
id	bigint(20)	Pirminis raktas
u_location	varchar(255)	Vietos pavadinimas, su kuria susieta statinės informacija
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
<i>statine_informacija_translation</i>		
id	bigint(20)	Pirminis raktas

Atributas	Tipas	Paskirtis
text	text	Statinės informacijos tekstas
lang	char(2)	Kalbos ISO kodas
<i>menesis</i>		
id	bigint(20)	Pirminis raktas ir kartu mėnesio skaitinė vertė
<i>menesis_translation</i>		
id	bigint(20)	Pirminis raktas ir kartu mėnesio skaitinė vertė
title	varchar(255)	Mėnesio pavadinimas
lang	char(2)	Kalbos ISO kodas
<i>naujiena</i>		
id	bigint(20)	Pirminis raktas
a_visability	boolean	Administratoriaus lygio matomumo ribojimas
vartot_id	bigint(20)	Vartotojo identifikatorius
created_at	datetime	Įrašo sukūrimo data
updated_at	datetime	Įrašo atnaujinimo data
<i>naujiena_translation</i>		
id	bigint(20)	Pirminis raktas
title	varchar(255)	Naujienos pavadinimas
short_text	text	Trumpas naujienos aprašymas
text	longblob	Ilgas naujienos aprašymas
lang	char(2)	Kalbos ISO kodas

3 Priedas. Duomenų bazės užklausa *Exhibit* duomenims formuoti

```
$query = Doctrine_Query::create()
->from('Individas i')
->select('CONCAT (
    CONCAT("label:\'",i.id,"\'"),
    CONCAT("utm_sq:\'",u.sq_name,"\'"),
    CONCAT("utm_xy:\'",u.xy,"\'"),
    CONCAT("utm_e:\'",u.xyc,"\'"),
    CONCAT("district:\'",s.district,"\'"),
    CONCAT("b_name:\'",t.title,"\'"),
    CONCAT("order:\'",t.b_order,"\'"),
    CONCAT("family:\'",t.family,"\'"),
    CONCAT("rb:\'",p.redbook,"\'"),
    CONCAT("iucn:\'",p.iucn,"\'"),
    CONCAT("date:\'",s.date,"\'"),
    CONCAT("charact:\'",st.title,"\'"),
    CONCAT("year:\'",year(s.date),"\'"),
    CONCAT("month:\'",mt.title,"\'"),
    CONCAT("day:\'",day(s.date),"\'"),
    CONCAT("watcher:\'",v.log_name,"\'"),
    CONCAT("location:\'",i.terrain,"\'"),
    CONCAT("amount:\'",i.amount,"\'"),
    CONCAT("sex:\'",lt.title,"\'"),
    CONCAT("age:\'",amt.cstle,"\'"),
    CONCAT("biotope:\'",bt.title,"\'"),
    *)'
) as json,
i.id
')
```

```
->leftJoin('i.Obj_duom obj')
->leftJoin('i.Amzius am')
->leftJoin('am.Translation amt')
->leftJoin('i.Lytis l')
->leftJoin('l.Translation lt')
->leftJoin('i.Biotopas bt')
->leftJoin('bt.Translation btc')
->leftJoin('obj.Paukstis p')
->leftJoin('p.Translation t')
->leftJoin('obj.Steb_objektas sobj')
->leftJoin('sobj.StebPobudis stpob')
->leftJoin('stpob.Translation st')
->leftJoin('sobj.Stebejimas s')
->leftJoin('s.Vartotojas v')
->leftJoin('s.Raionas r')
->leftJoin('s.Utm u')
->leftJoin('Menezis m')
->leftJoin('m.Translation mt')
->where('amt.lang =?', $culture)
->andWhere('lt.lang =?', $culture)
->andWhere('btc.lang =?', $culture)
->andWhere('t.lang =?', $culture)
->andWhere('st.lang =?', $culture)
->andWhere('mt.lang =?', $culture)
->andWhere('m.id = month(s.date)')
->andWhere('i.a_visibility =?', 1)
->andWhere('obj.a_visibility =?', 1)
->andWhere('s.a_visibility =?', 1)

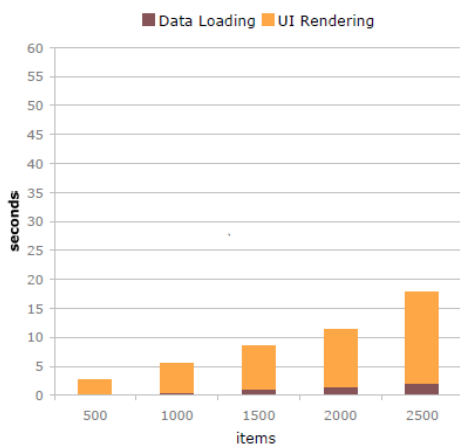
->orderBy('i.id ASC');
```

1 pav. Duomenų bazės užklausa, leidžianti gauti duomenis su dalinai suformuota JSON struktūra

4 Priedas. Exhibit įrankio veikimo spartos priklausomybė

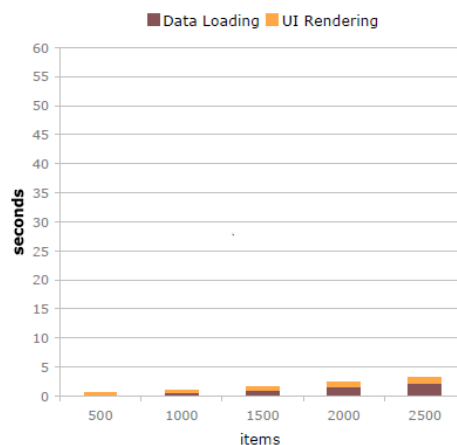
Exhibit įrankio veikimo spartos priklausomybė nuo naršyklės bei skirtingų operacinių sistemų pagal [Huynh, D.F., 2007] literatūrą.

Mac OSX 10.4.10, 2.16 GHz Intel Core 2 Duo, 2GB 667 DDR2 SDRAM



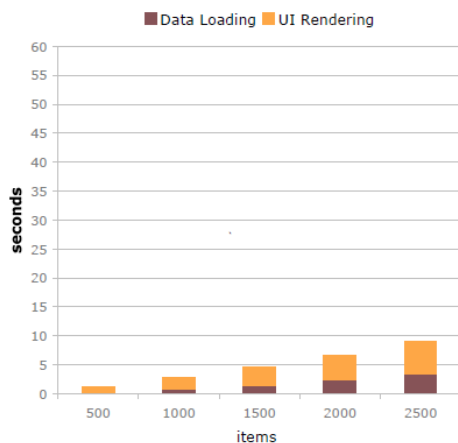
1a pav. Apdorojami visi elementai.

Naudojama *Firefox 2.0.0.5* versija



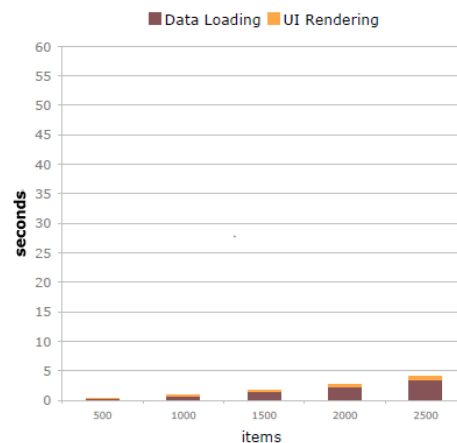
1b pav. Apdorojami pirmi 10 elementų.

Naudojama *Firefox 2.0.0.5* versija



2a pav. Apdorojami visi elementai.

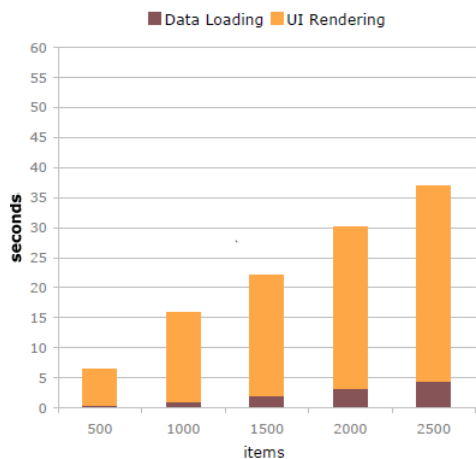
Naudojama *Safari 3.0.2* versija



2b pav. Apdorojami pirmi 10 elementų. Naudojama

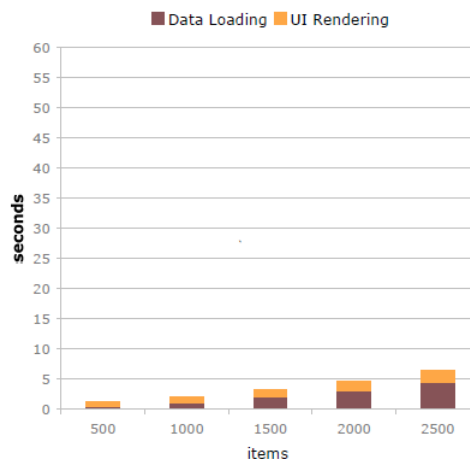
Safari 3.0.2 versija

Windows XP SP2, Pentium(R) 4 CPU 2.53 GHz, 1.00 GB RAM



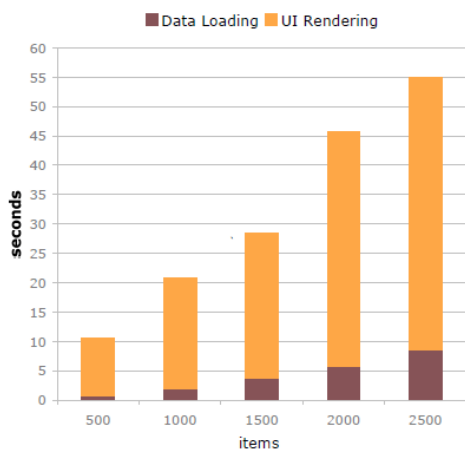
3a pav. Apdorojami visi elementai.

Naudojama *Firefox 2.0.0.5* versija



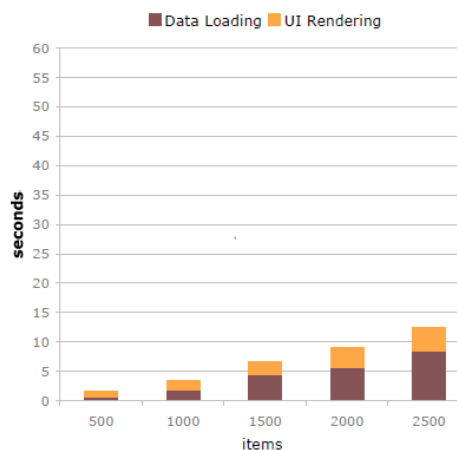
3b pav. Apdorojami pirmi 10 elementų.

Naudojama *Firefox 2.0.0.5* versija



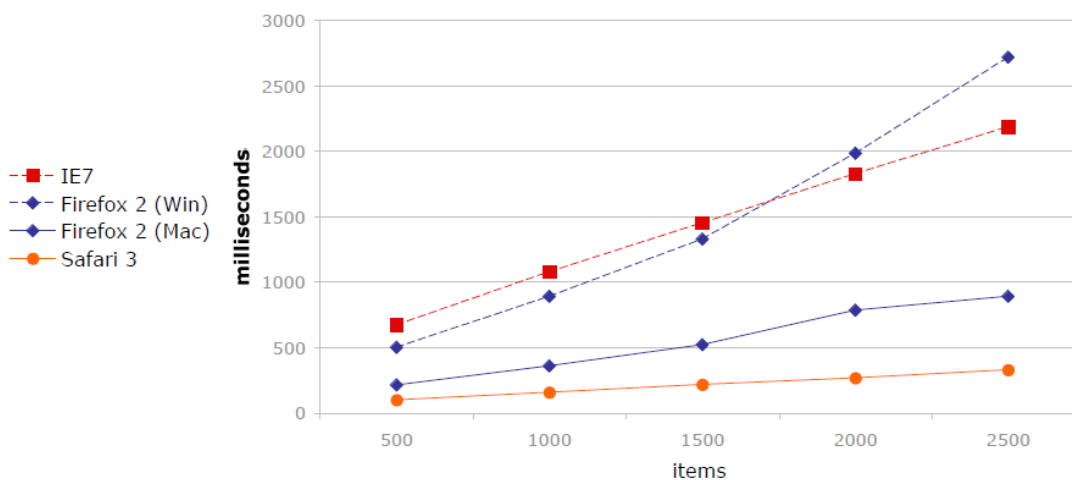
4a pav. Apdorojami visi elementai.

Naudojama *Internet Explorer 7.0.5730.11* versija

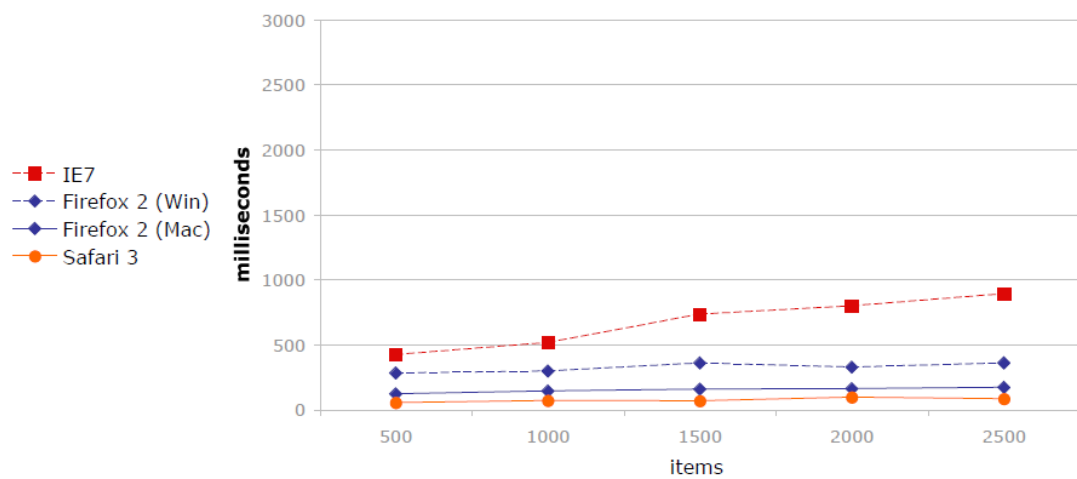


4b pav. Apdorojami pirmi 10 elementų.

Naudojama *Internet Explorer 7.0.5730.11* versija



5 pav. Exhibit įrankio veikimo greičio priklausomybė nuo įvairių naršyklių, kai apdorojami visi duomenys



6 pav. Exhibit įrankio veikimo greičio priklausomybė nuo įvairių naršyklių, kai apdorojami pirmi 10 elementų