

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Ramūnas Kiupelis

**Nekilnojamo turto administravimo proceso  
grafinio vizualizavimo modulis**

Magistro darbas

Vadovas

prof. Rimantas Butleris

2010-05

**KAUNAS, 2010**

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Ramūnas Kiupelis

**Nekilnojamo turto administravimo proceso  
grafinio vizualizavimo modulis**

Magistro darbas

Darbo konsultantas

Informacijos sistemų katedros lekt.

mag. Tomas Danikauskas

2010-05

Vadovas

prof. Rimantas Butleris

2010-05

Recenzentas

lekt. dr. Sigitas Drąsutis

2010-05

Atliko

IFM-4/4 gr. stud.

Ramūnas Kiupelis

2010-05-25

**KAUNAS, 2010**

## Turinys

Įvadas.....	6
1 Grafinio vizualizavimo modulio analizė .....	8
1.1 Analizės tikslas.....	8
1.2 Tyrimo sritis, objektas ir problema .....	8
1.3 Tyrimo objekto analizė .....	8
1.4 Vartotojų analizė .....	10
1.4.1 Vartotojų aibė, tipai ir savybės.....	10
1.4.2 Vartotojų tikslai ir problemos.....	10
1.5 Esamų sprendimų analizė.....	11
1.5.1 Technologijų analizė .....	11
1.5.1.1 SVG .....	12
1.5.1.2 Google Maps.....	12
1.5.1.3 Microsoft Silverlight.....	13
1.5.2 Programavimo kalbos pasirinkimas .....	14
1.5.3 Esamų produktų palyginimas .....	17
1.6 Siekiamas sprendimas .....	21
1.7 Darbo tikslas ir uždaviniai .....	22
1.8 Analizės išvados.....	22
2 Sistemos reikalavimai.....	23
2.1 Reikalavimų specifikacija .....	23
2.2 Dalykinės srities modelis .....	32
3 Sistemos projektas .....	33
3.1 Sistemos architektūros projektas.....	33
3.1.1 Sistemos loginė architektūra.....	33
3.1.2 Vartotojo paslaugos (sąsaja, sąsajos navigavimo planas) .....	33
3.1.3 Veiklos paslaugos .....	35
3.1.4 Duomenų paslaugos.....	35
3.2 Detalus projektas .....	36
3.3 Sistemos elgsenos modelis.....	39
3.4 Duomenų bazės schema .....	43
3.5 Realizacijos modelis.....	45
4 Sistemos realizacija .....	46
4.1 Sistemos veikimo aprašymas .....	46
4.2 Testavimo modelis bei duomenys, kontrolinis pavyzdys .....	50

4.3	Realizacijos apibendrinimas.....	57
5	Ekspirimentinis sistemos tyrimas .....	58
5.1	Savybių analizė .....	58
5.2	Komponento diegimas mašinų stovėjimo aikštelėje.....	61
6	Išvados.....	66
7	Literatūra .....	67
8	Terminų ir santrumpų žodynelis.....	68
9	Priedai.....	70
9.1	Priedas Nr. 1 (SQL programinis kodas).....	70

# GRAPHICAL VISUALIZATION MODULE OF REALTY ADMINISTRATION PROCESS

## SUMMARY

A graphic representation of data is increasingly popular on the Internet, in all security systems, property systems, statistical systems, asset management, trying to convey an image to the user-friendly interface that is clearly understandable not only to a developer, but also to a consumer that is using system not for the first time, that is using very often. A system can be perfect, with lots of features, but if its user interface is not acceptable for users, they won't use a product, and others will refuse to take such a product. In order to avoid that it is necessary to optimize system management.

The aim of this work is to design and develop the graphical visualization component of realty administration process. This component is implemented into system as a universal module and it can be adapted for other systems, such as rent items associated to the plans or parking places.

**Keywords:** Graphical visualization module, Component Programming, Realty administration.

## Įvadas

Kasmet didėjant informacijos svarbai, jos poreikis vis labiau auga. Informacija greičiausiai sklinda internete, tampa visiems prieinama, ją gali pasiekti didelis žmonių kiekis vienu metu. Natūraliai atsiranda vis naujų veiklos procesų, kurie perkeliami į internetą, nes tai palengvina komunikavimą tarp paslaugų tiekėjo ir kliento, arba tiesiog tarp dviejų veiklos procesų dalyvių. Internetas tapo vienu iš pagrindinių informacijos apsikeitimo būdų.

Grafinis duomenų vaizdavimas vis labiau populiarėja internete: visos apsaugos, nekilnojamojo turto, statistikos sistemos stengiasi perteikti vaizdą naudodamos vartotojui priimtina sąsają, kuri būtų suprantama, ne tik pačiam programuotojui, bet ir vartotojui, naudojančiam sistemą ne pirmą kartą, t. y. naudojančiam ją labai dažnai.

Pastaruoju metu Lietuvoje atsirado labai daug komercinės paskirties nekilnojamojo turto, kurį vienaip ar kitaip tenka administruoti įmonės vadovams. Ir tik tarptautinės kompanijos, valdančios nekilnojamą turtą, jau turi reikiamą programinės įrangos infrastruktūrą, daugeliu atvejų ji kuriama elementariausių kompiuterinių programų pagalba, bet ir tai yra žingsnis į priekį, nes vyksta dalinis įmonės veiklos procesų kompiuterizavimas.

Ši problema būdinga ne tik nekilnojamojo turto rinkai, bet ir kitoms verslo sritims: mašinų stovėjimu aikštelės, policininkų prižiūrimos teritorijos ir kt.

Šio darbo tikslas – sukurti grafinį įskiepi, Nekilnojamojo turto informacinės sistemos (IS), kuris bus naudojamas ne tik šioje sistemoje, bet ir kitose dalykinėse srityse, kuriose reikalinga valdyti grafinius objektus.

**Darbo tyrimo sritis** – grafinės informacijos atvaizdavimas internete, susiejant grafinius objektus su duomenų bazių duomenimis.

Šiame darbe išanalizuota esama ir sukurtos sistemos, padaryta esamos sistemos analizė, iš analizės parengtas projektas, pagal projektą parengta realizacija. Realizacijoje yra 7 žingsniai, kuriais sekant galima dirbti su pačia realizacija. Šie žingsniai aprašyti sistemos realizacijos veikimo apraše. Sistemai projektuoti panaudota CASE priemonė Magic Draw UML 16.6, kuri leido greičiau ir efektyviau sudaryti ir analizuoti esamą sistemą bei kuriamą komponentą.

Šis komponentas – tik karkasas, kurį galima toliau tobulinti pagal įvairius poreikius ir norus, jei reikia rodyti visą planą vienu mygtuko paspaudimu, ar net tiesiogiai keisti objektų būsenas. Tai tik keletas išvardintų patobulinimo atvejų.

Darbas susideda iš devynių skyrių.

Pirmajame skyriuje išanalizuojama esama sistema: suformuluojama tyrimo sritis, tyrimo objektas ir analizės tikslas. Iš nagrinėtų sistemų, programavimo kalbų ir esamų sprendimų sudaromos išvados, pagal kurias atliekamas tolimesnis darbas.

Antrajame skyriuje pateikti sistemos funkciniai, nefunkciniai reikalavimai ir dalykinės srities modelis.

Trečiajame skyriuje pateiktas sistemos projektas, architektūra, detalus sistemos projektas su sekų diagramomis, vartotojo sąsajos navigacijos planas, robastiškumo diagramos, taip pat ir transformuotas esybių modelis į duomenų bazės modelį.

Ketvirtajame skyriuje pateikta sistemos realizacija, sistemos veikimo aprašymas, testavimas ir sistemos realizacijos apibendrinimas.

Penktajame skyriuje – sistemos eksperimentas.

Šeštajame skyriuje suformuluotos darbo išvados.

Septintajame skyriuje išvardinti literatūros šaltiniai, kuriais buvo remtasi rašant šį darbą.

Aštuntajame skyriuje – terminų ir santrumpų žodynas.

Devintajame skyriuje pateikti darbo priedai.

## 1 Grafinio vizualizavimo modulio analizė

Šiame skyriuje pateikiama analizės tikslai, tyrimo sritis, tyrimo objektas ir problema. Analizuojami esami sprendimai ir pateikiamos analizės išvados.

### 1.1 Analizės tikslas

Išanalizuoti esamus sprendimus, išnagrinėti kiekvieno savybes ir funkcijas;

Rasti problemines sritis esamoje sistemoje ir optimizuoti procesus;

Pasirinkti iš esamų sprendimų tinkamiausią problemai pašalinti.

### 1.2 Tyrimo sritis, objektas ir problema

**Tyrimo sritis:** Grafinės informacijos atvaizdavimas internete, susiejant grafinius objektus su duomenų bazių duomenimis.

**Tyrimo objektas:** Nekilnojamojo turto administravimo internetinė sistema.

**Tyrimo problema ir numatomas sprendimas:**

- Grafinės informacijos apdorojimo našumo stoka (netenkinantis spartumas) – optimizuoti funkcijų veikimą;
- Funkcionalumo stoka netenkina naujai susidariusių funkcinių ir nefunkcinių reikalavimų – papildyti informacinės sistemos funkcionalumą.

### 1.3 Tyrimo objekto analizė

Pagrindinis sistemos funkcionalumas – pateikti vartotojams informaciją apie įmonę. Prisijungę vadybininkai administruoja šią informaciją, taip pat naudojami vienu iš papildomų funkcionalumų, norėdami informuoti įmonės partnerius apie įsiskolinimus ar apie naujas vietas, esančias jų teritorijoje.

**Tyrimo objektas:** grafinės informacijos atvaizdavimas internete, suteikiant interaktyvumo – susiejant su duomenų baze.

**Sistemoje realizuotos šios funkcijos:**

- Pasirinkti paveikslėlį iš failų katalogo, kuris buvo įkeltas per ftp arba įkeltas per sistemą;
- Papildomų mažų paveikslėlių pasirinkimas, kuris yra panašus kaip prieš tai minėtas „Pasirinkti paveikslėlį“;
- Žymėti atitinkamus plotus (stačiakampiu, apskritimu, daugiakampiu);
- Perkelti į HTML formatą (perkėlus į HTML formatą nesimato, kokie plotai pažymėti, tačiau užvedus pelytę ant atitinkamo ploto galima pamatyti „rankytę“, kuri nuves į tą nuorodą, kurią pasirinko vadybininkas).

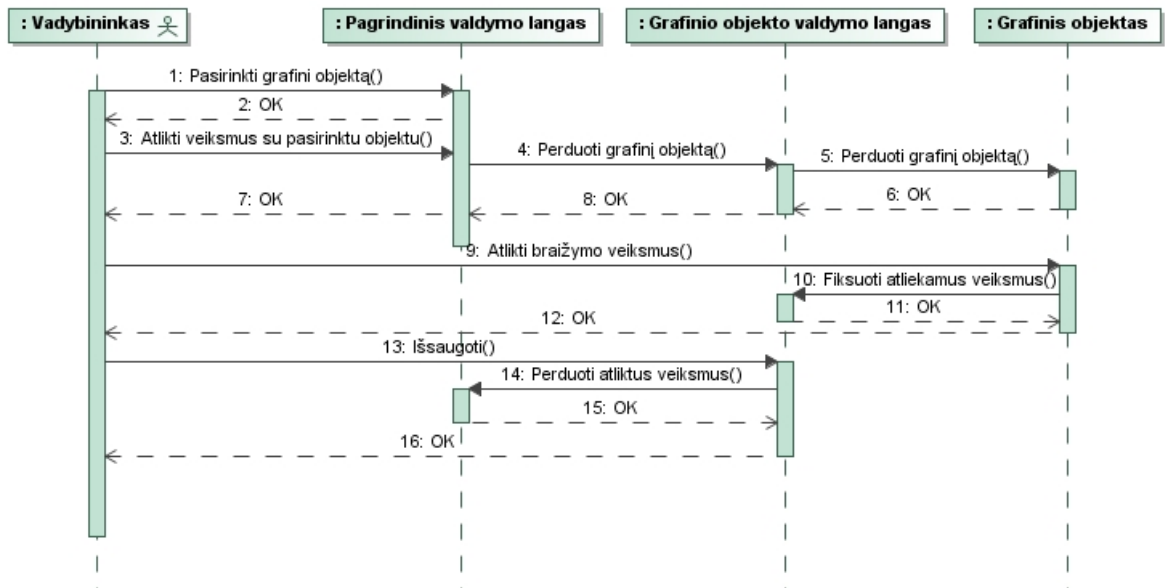
**Sistemos analizė.** Grafinės informacijos atvaizdavimas internete atliekamas su grafinio objekto valdymu ir WYSIWYG redaktoriumi. Pasirinkus paveikslėlį iš pagrindinio WYSIWYG valdymo lango perduodame jį į grafinio objekto valdymo langą naudodamiesi



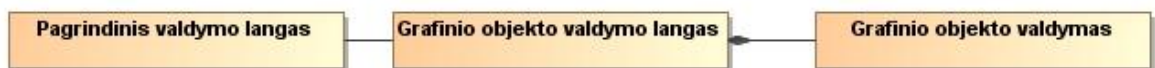
vienu iš valdymo mygtukų. Grafinio objekto valdymo lange atliekame pasirinktus veiksmus ir išsaugojimo mygtuku grįžtame į pagrindinį WYSIWING redaktoriaus langą. Visa tai pavaizduota 1.3.1 pav. ir 1.3.1 lentelėje. Sąveikos tarp objektų pavaizduotos 1.3.2 pav.

**1.3.1 lentelė** *Dabartinės sistemos sekų diagramos specifikacija*

<b>Aktorius</b>		Vadybininkas
<b>Sužadinimo sąlyga</b>		Nori redaguoti grafinio objekto būseną
<b>Prieš sąlyga</b>		Vartotojas turi būti prisijungęs; Turi egzistuoti grafinis objektas
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	
	<b>Apima PA</b>	
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1. Vadybininkas pasirenka grafinį objektą		Sistema grąžina OK
2. Vadybininkas pasirenka atlikti veiksmus su objektu.		2.1. Sistema perduoda parametrus į grafinio objekto valdymo langą. 2.2. Sistema perduoda parametrus iš grafinio objekto valdymo lango į grafinio objekto valdymo sistemą. 2.3. Sistema grąžina OK
3. Vadybininkas atlieka braižymo veiksmus.		3.1. Sistema fiksuoja atliekamus veiksmus 3.2. Sistema grąžina OK.
4. Vadybininkas pasirenka išsaugoti		4.1. Sistema perduoda atliktus veiksmus. 4.2. Sistema grąžina OK.
<b>Po sąlyga</b>		Matomas pakeistas grafinis objektas
<b>Alternatyvūs scenarijai</b>		



1.3.1 pav. Dabartinės sistemos sekų diagrama



1.3.2 pav. Objektų sąveikos modelis

## 1.4 Vartotojų analizė

### 1.4.1 Vartotojų aibė, tipai ir savybės

Sistemoje yra 3 tipų vartotojai: **administratoriai, vadybininkai ir viešas vartotojas.**

**Administratoriai** – prieina prie visų esamų temų ir leidžia priėjimus tiek administratoriams, tiek vadybininkams, tiek viešiesiems vartotojams prieigas prie sistemos puslapių. Administratoriai taip pat gali leisti prieiga prie tam tikrų sistemos modulių.

**Vadybininkai** – gali atlikti veiksmus, tuos kuriuos leidžia administratorius.

**Viešas vartotojas** – gali tik peržiūrėti leistinas temas.

### 1.4.2 Vartotojų tikslai ir problemos

Patobulinus jau esamą sprendimą, ir sukūrus papildomas funkcijas šios problemos bus išspręstos (žr. lentelę 1.4.2).

1.4.2 lentelė Vartotojų tikslai ir problemos

<b>Problema</b>	<b>Kaip viskas vyksta dabar</b>
Lėtas duomenų su pažymėtais plotais išsaugojimas	<ul style="list-style-type: none"> <li>• Paruošia planą saugojimui</li> <li>• Daro matomo vaizdinio kopiją</li> <li>• Persikelia vaizdinį į vaizdo redagavimo programą.</li> <li>• Perkelia planą atgal į serverį.</li> <li>• Iš naujo sudėlioja nuorodas.</li> </ul>
Blogai atvaizduoja pašto programose paveikslėlius	Dedant paprastą paveikslėlį į planą paveikslėlis persimeta į apačią.
Vartotojas negali pasirinkti srities žymėjimo būdo.	Pažymėtą plotą spalvina tik viena spalva, pilka. Persidengiantys plotai tamsina sritį.
Nėra galimybės grafiniams objektams uždėti būseną	Šios funkcijos nėra sistemoje.
Nėra galimybės susieti grafinius objektus su kitais duomenimis.	Šios funkcijos nėra sistemoje.
Nėra galimybės suskirstyti objektų duomenis į grupes	Šios funkcijos nėra sistemoje.

## 1.5 Esamų sprendimų analizė

### 1.5.1 Technologijų analizė

Daugelis komponentinių programavimo elementų, paremtų programine įranga, vartotojo sąsajos modeliais, komponentais, sąsajoms pagrįstą projektavimą ir hibridiniais komponentais (sistemas), turi tiek bendrų, tiek skirtingų savybių.

Hibridinės sistemos remiasi komponentais paremtomis technologijomis. Šio tipo projektai savo apimtimi yra pakankamai maži ir galima naudoti testais paremta kūrimo technologija. Tai yra kiekvienu etapu galime patikrinti komponento efektyvumą: jei komponentas dirba ne efektyviai arba klaidingai, grįžtame į pradinę stadiją su tam tikru komponento sukurtu kapitalu ir tęsiame tolimesnį darbą.

Sąsajomis grindžiamas projektavimas ir programavimas buvo sukurtas dėl apribojimų, kurie atsiranda naudojant ryšį tarp klasių ar paveldėjimą. Sąsajos projektavimą ir programavimą padaro sudėtingesniu, bet patirtos sąnaudos atsiperka, kuriant didelės apimties informacijos sistemas [7, 11, 12].

Pakartotinio naudojimo metodologija ir transformavimo sistemos, metodologijos tikslas, atrodo, taip pat aiškus – siekiama didesnio proceso našumo ir efektyvumo, aukštesnės gaminio kokybės ir trumpesnio kelio į rinką, kad gaminys greičiau pasiektų vartotoją.

Pakartotinio naudojimo technologiją sudaro dvi dalys: kompozicinis pakartotinis naudojimas (angl. coomposition reuse), ir transformacinis pakartotinis naudojimas generuojant (angl. generative reuse). Atitinkamai jų tyrimo objektai – komponentika ir generatoriai (transformavimo sistemos). Jų sukūrimo metodologinis pagrindas – srities analizė [9].

#### **1.5.1.1 SVG**

SVG – standartas, pradėtas kurti 1998 metais, po to kai buvo paskelbti VML (Vector Markup Language) ir PGML formatai. Kūrimui vadovavo Chrisas Lilley, pirmoji standarto versija (1.0) paskelbta 2001 metų rugsėjo mėnesį, 2003 metų sausį paskelbta naujausia (1.1) versija. 2003 metais taip pat paskelbtos ir standarto versijos, skirtos mobiliems įrenginiams – SVG Tiny bei SVG Basic. Šiuo metu kuriamos SVG Full 1.2 bei SVG Tiny 1.2 standartų versijos.

Formatas dar nėra palaikomas taip, kaip populiarūs senesni formatai (GIF, PNG, JPEG). Daugeliui naršyklių reikia įdiegti specialius išplėtimus, norint, kad šiose būtų vaizduojami SVG vaizdai, todėl dažnai SVG vaizdus naudojančios svetainės verčia tokius vaizdus į populiariesnius ne vektorinius formatus.

SVG numatyti trijų rūšių objektai:

- vektorinės figūros, sudarytos iš atkarpų ir kreivių;
- ne vektorinės iliustracijos;
- tekstai.

Objektai gali būti grupuojami, transformuojami (pasukami, ištempiami), sujungiami, galima keisti kiekvieno objekto savybes (pavyzdžiui, alfa permatomumą, spalvą, linijos storį ar tipą).

Interaktyvių SVG vaizdų animacija gali būti kuriama tiesiogiai pridėdant animacijos elementus ar aprašant skriptus, naudojant pasirinktą skripto kalbą bei SVG dokumento objektinį modelį (DOM), leidžiantį pasiekti bet kurį SVG dokumento elementą ar jo savybę.

#### **1.5.1.2 Google Maps**

Google Maps (anksčiau buvo žinomas kaip Google Local) yra internetinis žemėlapis. Žemėlapių kūrė Google. Google Maps leidžia žiūrėti palydovines fotografijas, eismo informaciją JAV, Jungtinėje Karalystėje (Didžiojoje Britanijoje) ir Prancūzijoje. Galima surasti kelionės maršrutą, važiuojant automobiliu ar viešuoju transportu. Prisijungus prie

Google, galima žymėti įvairias žemėlapių vietas ir išsaugoti savo darbą. Google žemėlapių teikia ir kitokias galimybes:

- Pažymėti tam tikrus plotus;
- Pridėti įvairių spalvų žymeklius;
- Surasti trumpiausią kelią;
- Nustatyti kelionės maršrutą, maršrutai atvaizduojami keliais pagal kelių eismo taisykles;
- Žemėlapių veikia GPS vietovės nustatymo telefonuose, kitaip vadinamuose TomTom;
- Rodo keliones maršrutą važiuojant mašina;
- Labai lankstus programinis valdymas;
- Galimybė integruoti vaizdo įrašus;
- Žemėlapis žemėlapyje.

Google Maps buvo papildytas nauju žemėlapių tipu, kuris pavadintas "terrain". Šiame žemėlapyje matomas beveik trimatis reljefo vaizdas. Turbūt net nemačiusieji realaus kraštovaizdžio galės įsivaizduoti, kur lygumos, o kur kalnai. Daug įdomesnis vaizdas matomas pasirinkus Google mylimiausias vietas. (Google Favorite Places): čia matomi net 3D pastatų vaizdai. Kitiems miestams mažiau pasisekė, juose matomos tik gatvės. Žinoma, Lietuva patenka į tą nelaimingų šalių sąrašą.

Kai kuriuose miestuose leidžiama pasijusti taip, lyg būtumei pačiame mieste ir keliautum pėsčias. Tai įgyvendinta naudojant paveikslėlius: kiekvieno žingsnio metu matome vieną vaizdą, kuri galime apsukti 360 kampu.

Vienas didžiausių Google TomTom trūkumų – nemalonus planavimas, nes nedideliame ekrane planuoti nepatogu. Greit turėtų atsirasti patogesnių įrankių, skirtų bent jau TomTom. Patys pirmieji žingsniai jau žengti – padaryta TomTom ir Google Maps integracija.

### **1.5.1.3 Microsoft Silverlight**

„Microsoft Silverlight“ yra tarpsisteminė naršyklė, įvairioms aplinkoms tinkantis papildinys, pateikiantis daugialypės terpės perteikimą, paremtą naujos kartos „Microsoft.NET“, ir turingas interaktyvias žiniatinklio taikomąsias programas. Šiame straipsnyje aprašoma „Silverlight 3“ versija.

„Silverlight 3“ yra daugiau nei 50 naujų funkcijų, tarp jų išskirtinos šios savybės:

- Padidintas saugumas ir patikimumas;
- Pagerintas medijos ir vaizdo įrašų palaikymas, įskaitant H.264 vaizdo įrašų palaikymą;
- Galimybė leisti „Silverlight“ programas už naršyklės ribų;

- Pagerinta grafika, jos charakteristikos ir kūrybiniai efektai, tarp jų – ir „Perspective 3D“ grafikos palaikymas;
- Pagerintos turiningų interneto programų (RIA) kūrimo galimybės.
- „Silverlight 3“ yra suderinama su ankstesnėmis žiniatinklio programomis, kurios buvo įdiegtos ankstesnėse „Silverlight“ versijose.

„Microsoft Silverlight“ prilygsta „Adobe Flash“, tačiau apie 80% vaizdo turinio internete remiasi būtent „Adobe Flash“ technologija.

### **1.5.2 Programavimo kalbos pasirinkimas**

Renkantis ir nagrinėjant atskirai kiekvieną programavimo kalbą, - atitiktum pilnam funkcionalumui nėra. Bet galima padaryti visą funkcionalumą tiek su FLASH, tiek su FLEX, tiek su JAVA APLET. Tada iškyla klausimas: kuris jų efektyvesnis, patikimesnis ir su kuriuo yra geriau realizuoti?

JAVA APLET naudojamas tik įdiegiant JAVA į operacinę sistemą. Todėl reikalingas papildomas nustatymas – ar yra įdiegta APLET [4].

FLEX naudoja FLASH PLAYER, sistemoje užima virš 10 kartų mažiau atminties, negu JAVA APLET. FLEX dažniau naudojamas objektiniam programavimui, sąsajoms su MYSQL, PHP, HTML ir kitomis programavimo kalbomis. Trūkumas – negalima kurti MOTIONS pasirinkus kokį nors įrankį, nes viskas aprašoma tik ACTIONSCRIPT, MXML [3].

FLASH, ypač dabartinė išleista versija CS5, labai pažengė į priekį lyginant su ankstesnėmis versijomis. Daugelis stebisi, kam tiek daug funkcijų joje, jei daugiau nei pusė dažniausiai nenaudojamos, o tos funkcijų, kurių reikia, nerandamos, ir todėl reikia ieškoti alternatyvių sprendimų.

Norit pagerinti efektyvumą, reikia tobulinti ne tik pačius metodus, bet ir pasirinkti tinkamiausią programavimo kalbą, kuria galima įvykdyti norimus veiksmus; taip pat reikia galvoti apie tolimesnį sistemos tobulinimą. Nagrinėsime tik WEB sąsajas: FLASH, FLEX ir JAVA APLET.

FLASH privalumai [6]:

- FLASH yra gana universali programa, leidžianti dirbti su vaizdo įrašais, paveikslėliais bei vektorine grafika ir animacija;
- OS suderinamumas;
- FLASH - vienas iš geriausių pasirinkimų, kai reikia interaktyvumo, pavyzdžiui. žaidimams kurti.

FLASH trūkumai [6]:

- Puslapio apkrova - kuo daugiau FLASH'inių elementų viename puslapyje, tuo lėčiau veikia puslapis;
- FLASH yra sunkiai suderinama su interneto paieškos programomis. Anksčiau FLASH sukurtos svetainės nebuvo indeksuojamos, nes paieškos sistemos tiesiog neperskaitydavo teksto, esančio FLASH'e. Svarbu žinoti, kad paieškos sistemos (taip pat ir žmogus), sunkiai randa reikiamą informaciją, jei nėra paieškos pačiame FLASH'e;
- Daugumoje FLASH puslapių sudėtinga pastebėti tekstinę informaciją, todėl kad ją užgožia per didelis naudojamos animacijos / vizualinių specialiųjų efektų kiekis;
- FLASH animacijas šiuo metu palaiko tik keletas telefonų modelių.

FLEX privalumai [6]:

- Naudoja FLASH grotuvo technologiją;
- Teikia realaus laiko prieigą prie daugumos išorinių šaltinių, tokių kaip duomenų bazės;
- FLEX jau nebėra visiškai atviro kodo.

FLEX Trūkumai [6]:

- Reikia išmokti XML ir ACTIONSCRIPT kalbą, norint ką nors kurti;
- Ribotos vaizdavimo galimybės.

Java APPLELET gali turėti bet kurį arba visus iš šių privalumų [4]:

- Yra paprasta, dirba Linux, Windows ir MAC OS t. t.;
- Tas pats JAVA APPLELET gali naudoti visas JAVA versijas. Tačiau, jei privestas ilgai laukti, kol parsisiųs;
- JAVA palaiko dauguma naršyklių;
- Kadangi JAVA APPLELET kešuoja (angl. "cache") dauguma interneto naršyklių, todėl jie bus greitai paleisti, sugrįžus atgal į tą patį internetinį puslapį, tačiau gali užstrigti keše, ir dar gali kilti problemų, kai išleidžiamos naujos JAVA versijos;
- JAVA gali turėti visišką priėjimą prie sistemos įrenginių, jeigu vartotojas sutinka;
- JAVA APPLELET galima pagerinti naudojant: po pirmojo paleidimo, kiti pasileidžia greičiau, tačiau JVM reikia perkrauti kiekvieną kartą, kai naršyklė paleidžiama iš naujo;
- JAVA APPLELET gali veikti tokiu greičiu, kuris yra panašus į (tačiau paprastai lėtesnis nei) kitas, toks kaip C++, tačiau daug kartų greičiau nei JAVASCRIPT;
- Kūrėjai gali kurti ir derinti APPLELET tiesiogiai, tiesiog sukurdami pagrindines klases ir iškviisdami init () bei start () APPLELET'e, tokiu būdu užtikrinant plėtrą savo įprastoje J2SE kūrimo aplinkoje. Visgi atliekant testavimą reikia pakartotinai

testuoti su APPLETT-VIEWER programa arba interneto naršykle, siekiant užtikrinti, kad jos atitiktų saugumo apribojimus.

JAVA APPLETT gali turėti bet kurį iš šių trūkumų [4]:

- JAVA APPLETT reikalauja JAVA programinės įrangos. JAVA APPLETT negali veikti tol, kol nėra paleista pati JAVA aplinka operacinėje sistemoje, ir tai gali turėti reikšmingų pasekmių paleisties metu;
- Kartais dėl saugumo apribojimų negalima įgyvendinti arba sunkiai pavyksta atlikti užsibrėžtą tikslą su JAVA APPLETT;
- Kai kurios organizacijos leidžia įdiegti programinę įrangą tik administratoriams. Todėl daugelis vartotojai negali matyti APPLETT pagal nutylėjimą;
- JAVA APPLETT gali reikalauti specialios JRE aplinkos.

Algoritminė sparta: iš visų nagrinėtų programavimo kalbų sparčiausias yra JAVA, nes galima naudoti C++ kalbos algoritmus.

Sąsaja su duomenų baze: tiesiogines DB sąsajas galima daryti su JAVA. Su FLASH ir FLEX tiesioginės sąsajos padaryti negalima, tačiau įmanoma ją padaryti naudojant PHP.

Grafikos apdirbimas: tiek su JAVA, tiek su FLASH, tiek su FLEX galima padaryti grafinę objektų sąsają, tačiau iš išvadintų programavimo kalbų FLASH ir FLEX sąsaja yra paprastesnė ir mažiau sudėtinga nei JAVA. FLASH leidžia naudoti tokius grafinius efektus kaip permatomumas, objektų judėjimas ir kiti.

Sudėtingumas: FLASH yra skirtas grafinių objektų valdymui, dėl to šis pasirinkimas yra daug svaresnis, negu JAVA ar FLEX programavimo kalbos.

Laiko sąnaudos: JAVA neturi tiek daug dokumentacijos apie grafinių objektų valdymą, taškų tampymą iš vienos vietos į kitą. FLEX daugiau pritaikytas tokių duomenų vaizdavimui, kaip lentelės sąrašams atvaizduoti, mažiau skirtas grafiniam vizualizavimui, todėl kuriant su FLEX laiko sąnaudos būtų dar didesnės nei su JAVA. FLASH turi daug dokumentacijos apie grafinių objektų valdymus, todėl tai padeda sumažinti laiko sąnaudas.

Spartumas atnaujinant: diegiant naujas funkcijas arba toliau tobulinant sistemą, t. y. FLASH komponente sudarant tokią struktūrą, galima mažomis pastangomis įdiegti papildomą funkciją daug greičiau, nei kitose programavimo kalbose, nes užtenka tik perkompiliuoti vieną failą ir galima naudotis toliau.

Išnagrinėjus programavimo kalbas sudaryta 1.5.3 lentelė.



1.5.3 lentelė Programavimo kalbų palyginimas

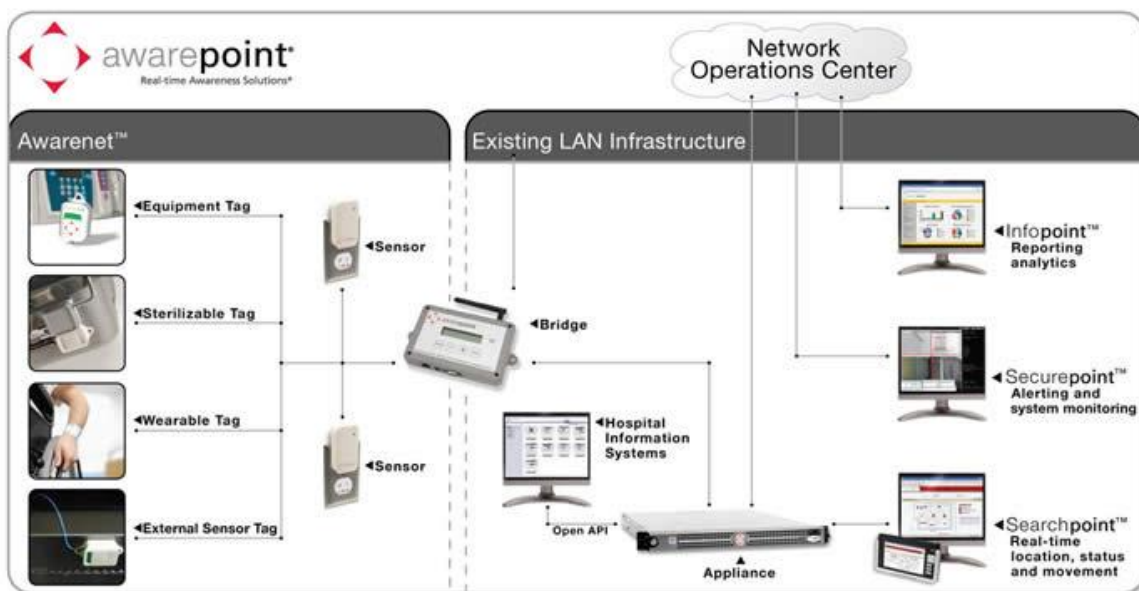
	JAVA	FLASH	FLEX	Komentaras
<b>Algoritminė sparta</b>	+	-	-	Sparčiausias JAVA, nes galima naudoti C++ kalbos algoritmus.
<b>Sąsaja su duomenų baze</b>	+	-	+	FLASH sąsaja galima padaryti naudojant PHP
<b>Grafikos apdirbimas</b>	+	+	+	Priklausomai nuo kiekvienos programavimo kalbos sudėtingesnis algoritmas ir pritaikymas
<b>Sudėtingumas</b>	+	-	+	Sudėtingiausi algoritmai yra JAVA, sekantis – FLEX.
<b>Laiko sąnaudos</b>	+	-	+	+ reiškia, kad sunaudosime daug laiko, t. y. reikės rašyti viską iš naujo, kurti naujus metodus ir naujas funkcijas.
<b>Spartumas atnaujinant</b>	-	+	-	Funkcijų tobulinimas atima mažiau laiko, negu jų sukūrimas.

### 1.5.3 Esamų produktų palyginimas

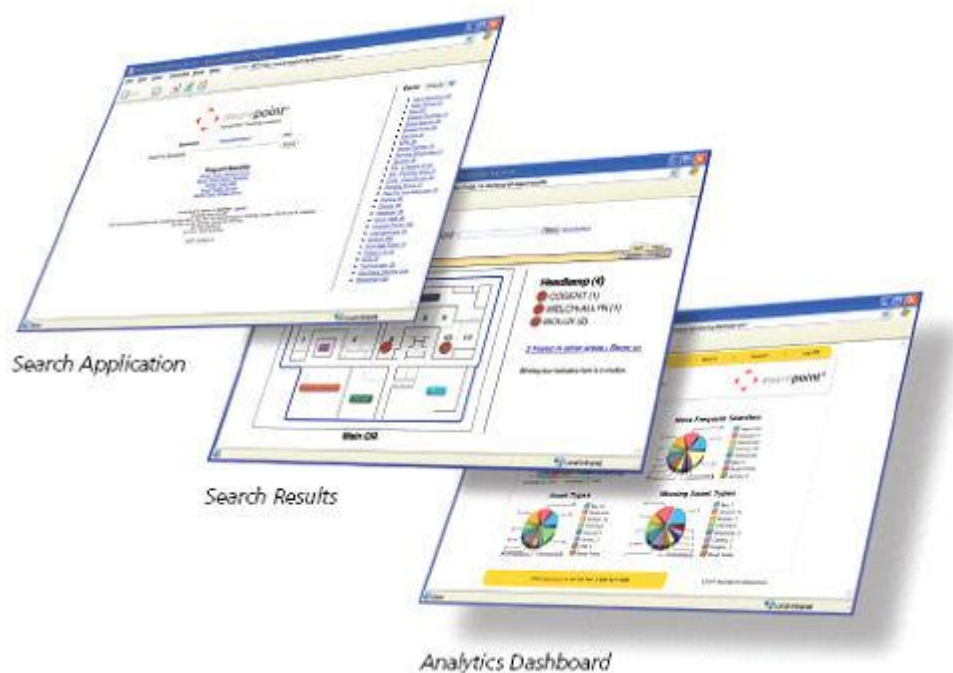
Darbe analizuojami trys esami sprendimai: Sensorių identifikavimo sistema (*Real-time Location Systems*) [1], Ligoninė turto valdymo sistema [5], *Locate and Manage Wireless Devices* [8]. Sudaryta šių sistemų palyginimo 1.5.3.1 lentelė.

Sensorių identifikavimo sistema naudoja papildomą techniką: sensorius, bevielį adapterį ir kitą įrangą, kuri pateikta 1.5.3.1 pav., taip pat pateiktos sistemos grafiniai vaizdai 1.5.3.2 pav.. Šio produkto pagrindiniai aspektai:

- Visų sensorių būsenas galima stebėti per interneto naršyklę nuotoliniu būdu;
- Leidžia atlikti kiekvieno sensoriaus analizę;
- Individualiai sudaromas planas pagal sutartį.



1.5.3.1 pav. Sensorių identifikavimo sistema



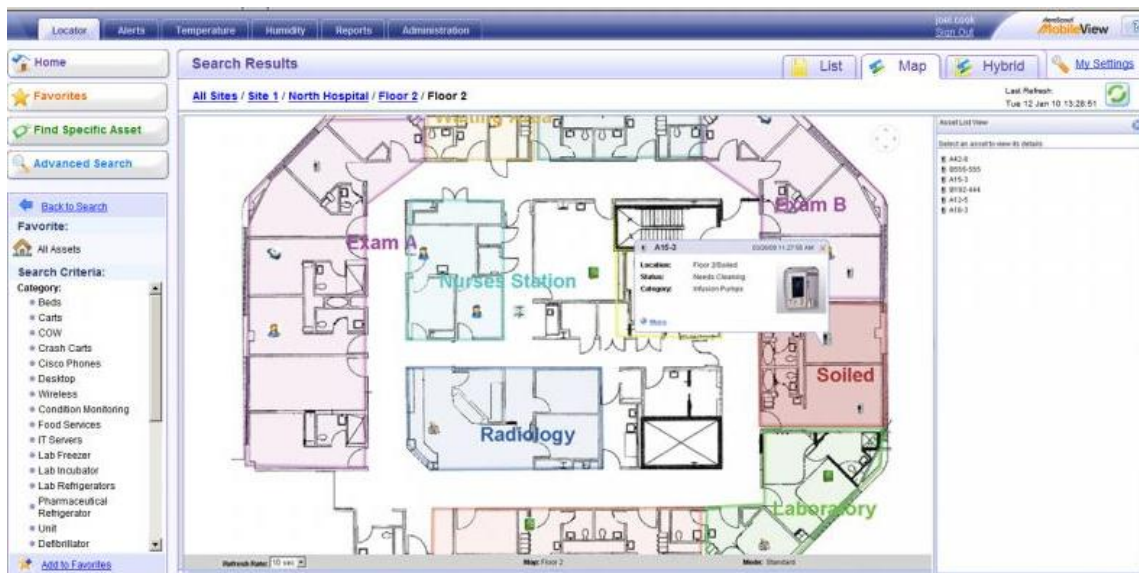
1.5.3.2 pav. Sensorių identifikavimo sistemos vaizdiniai

Ligoninės turto valdymo sistema (*Asset Management*) (žr. 1.5.3.3 pav.). Ši sveikatos turto stebėjimo ir valdymo sistema skirta stebėti įrangos buvimo vietą.

Sistemos privalumai:

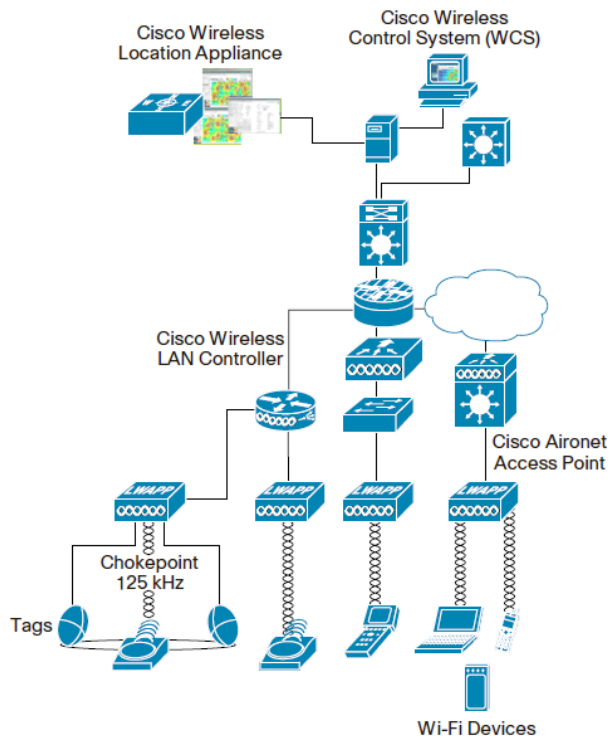
- Sumažina įrangos pirkimo ir nuomos išlaidas;
- Gerina darbuotojų našumą ir darbo eigą, sumažina įrangos paieškos ir užsakymo laukimo laiką;

- Didina įrangos panaudojimą;
- Automatizuoja atsargų valdymą;
- Sumažina įrangos kiekį ir tuo pačiu sumažina išlaidas įrangos priežiūrai.

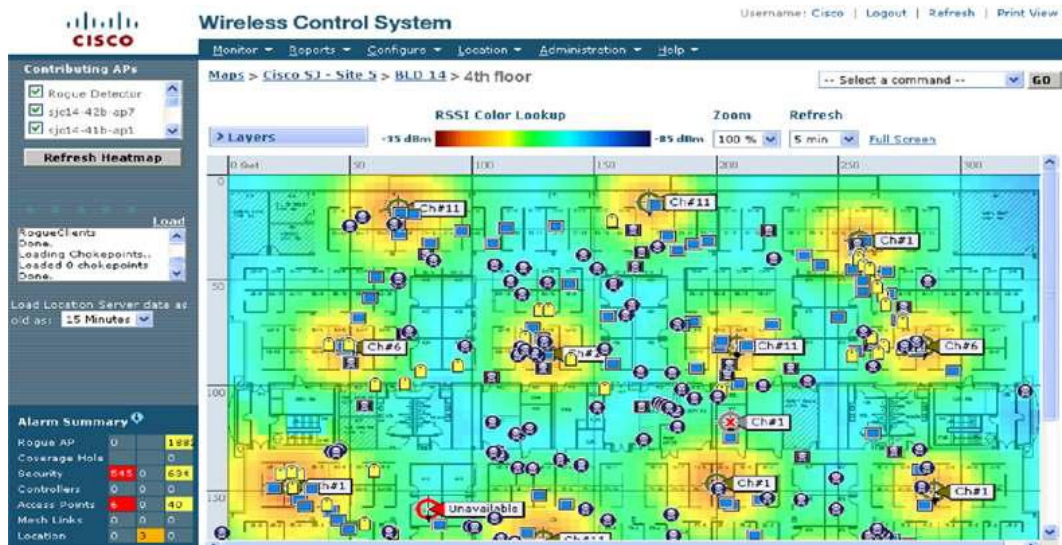


1.5.3.3 pav. Ligoninės turto valdymo sistema

*Locate and Manage Wireless Devices* sistemos architektūra pateikta 1.5.3.4 paveikslėlyje. Architektūroje vaizduojama kaip vienas su kitu komunikuoja pagrindiniai objektai, kaip visi duomenys gaunami sistemoje. Pačios sistemos grafinis vaizdas 1.5.3.5 pav. rodo wi-fi įrenginių stiprumo signalą, kiek aprėpia erdvės kiekvienas signalas ir koks jo stiprumas yra tam tikru atstumu nuo įrenginio.



1.5.3.4 pav. Cisco Location Solution Architecture



1.5.3.5 pav. Cisco Wireless Control System—Targeted View for Rogue Access Points and Devices

1.5.4 lentelė Esamų produktų palyginimas

<b>Kriterijai</b>	<b>Sensorių identifikavimo sistema</b>	<b>Ligoninės turto valdymo sistema</b>	<b>Locate and Manage Wireless Devices</b>	<b>Komentaras</b>
<b>Pavadinimas</b>				
<b>Veikia su daug planų</b>	+	+	+	Sistema palaiko ne vieną planą
<b>Yra dinamiškai keičiamas</b>	-	-	-	Planai keičiasi dažnai ir lengva juos keisti nereikia papildomų įgūdžių
<b>Valdomas per internetinę aplikaciją</b>	+	+	+	Sistemą galima valdyti internetinės sąsajos pagalba
<b>Galima susieti su teksto redaktoriumi</b>	-	-	-	Planus galima keisti per WYSIWYG redaktorių, išgeneruojant nuorodą į grafinį objektą
<b>Vaizduoja vietų užimtumą</b>	-	+	+	Fiksuoja plotų užimtumas
<b>Reikalauja papildomos įrangos</b>	+	+	+	Naudoja papildomą įrangą, neužteka vien internetinės sąsajos
<b>Planus galima siųsti elektroniniu paštu</b>	-	+	-	Pranešimai siunčiami nustatytiems asmenims

## 1.6 Siekiamas sprendimas

Kuriamojoje sistemoje bus siekiama įdiegti šiuos funkcionalumus:

- Plano sutalpinimas į vieną vaizdinį;
- Vartotojui rodomas objektų saugojimas;
- Saugojant objektus nenutrūkstamas saugojimas;
- Saugojimas vykdomas intervalais, o ne atliekant kiekvieną veiksmą;
- Pasirinkus atitinkamą būseną pasikeičia objekto spalva;
- Vadybininkų darbo optimizavimas, sumažinant veiksmų atlikimo seką.

## 1.7 Darbo tikslas ir uždaviniai

Darbo tikslas – sukurti grafinį įskiepį Nekilnojamojo turto informacinėje sistemoje (IS), kuris bus naudojamas ne tik šioje sistemoje, bet ir kitose dalykinėse srityse, kuriose turėtų būti valdomi grafiniai objektai.

Uždaviniai:

- Optimizuoti vadybininkų darbą;
- Išsaugoti duomenis duomenų bazėje;
- Suprojektuoti ir sukurti konfigūruojamą komponentą, kurį būtų galima naudoti ir kitose dalykinėse srityse.

## 1.8 Analizės išvados

1. Atsižvelgiant į išnagrinėtas metodologijas bus ruošiamas sistemos projektas. Šios metodologijos padės realizuoti komponentą, tinkamą pakartotiniam panaudojimui, diegiant į kitas sistemas, sudarant vartotojo sąsają ir testuojant realizaciją.
2. Išanalizavus programavimo kalbas, pasirinkta Adobe Flash, kadangi ji užima mažiau atminties, negu JAVA applet. Adobe Flex naudojamas tokių duomenų vaizdavimui, kaip duomenys, pateikiami lentelės forma.
3. Iš surastų realizuotų sistemų tokių kaip Sensorių identifikavimo sistema, Ligoninės turto valdymo sistema ir *Locate and Manage Wireless Devices*, visos yra visiškai pritaikytos konkrečiai dalykinei sričiai ir šių sistemų ar jų dalies adaptavimas, sprendžiant nekilnojamojo turto administravimo problemas, yra neįmanomas.

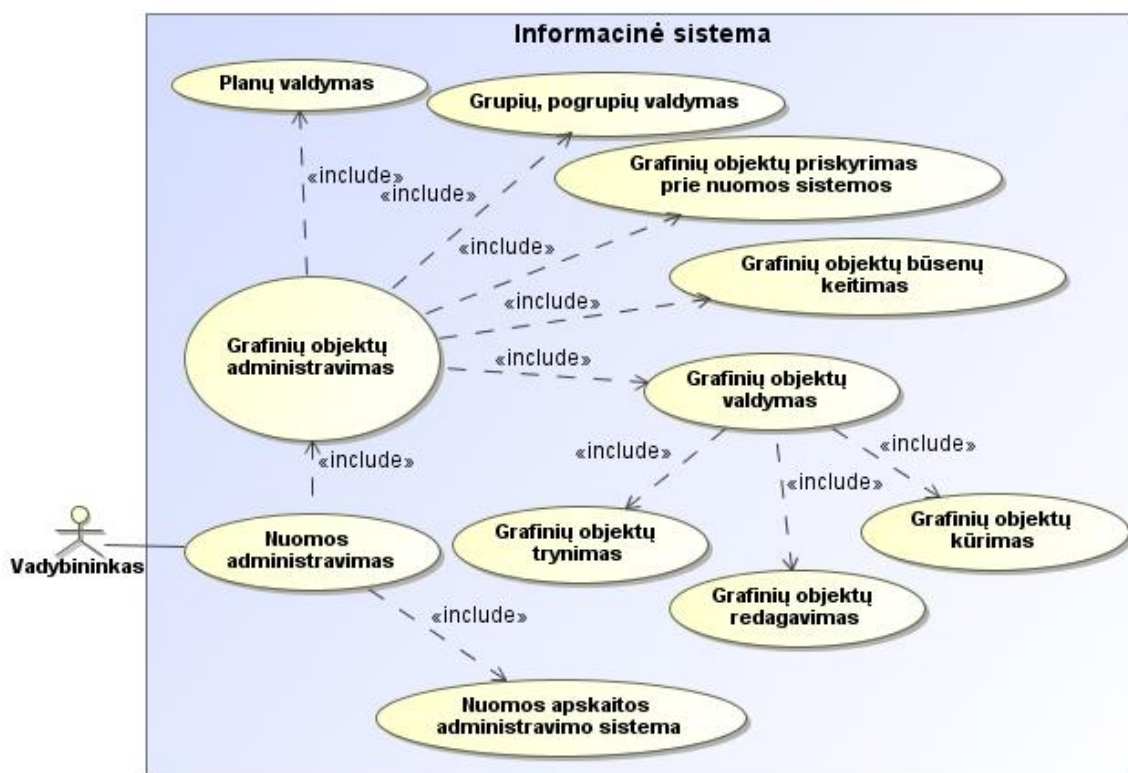
## 2 Sistemos reikalavimai

Šiame skyriuje sudaroma Nekilnojamojo turto administravimo sistemos reikalavimų specifikacija bei suformuluojami pagrindiniai sistemos funkciniai ir nefunkciniai reikalavimai.

### 2.1 Reikalavimų specifikacija

Funkciniai reikalavimai:

Sistemos panaudos atvejų diagrama vaizduoja galimus vartotojų veiksmus, valdant grafinius objektus ir sąsają tarp nuomos objektų (žr. 2.1.1 pav.). Panaudos atvejų diagramoje gali būti ir kiti vartotojai, prieinantis prie sistemos. Šiuo vienetiniu vartotoju galima nustatyti, kad prie kiekvieno objekto prieitų tik tam tikri vartotojai, tačiau, atsižvelgiant į užsakovo reikalavimus, numatomoje sistemoje nereikia tokio funkcionalumo.

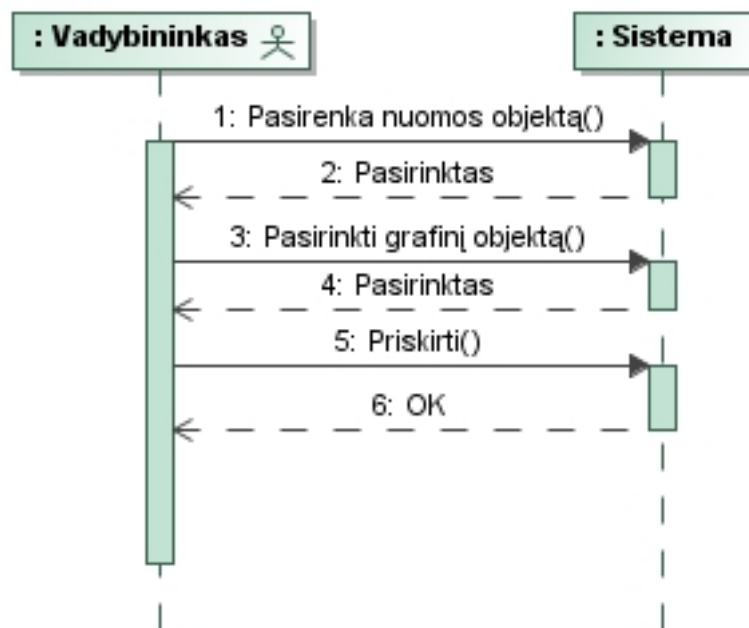


2.1.1 pav. Sistemos panaudos atvejų diagrama

Vienas iš panaudos atvejų – „Grafinių objektų priskyrimas prie nuomos sistemos“. Iš pateikto nuomos objektų sąrašo ir grafinių objektų sąrašo vykdomas vienos sistemos ir kitos sistemos sujungimas (detalesnė specifikacija pateikta 2.1.2 pav. ir 2.1.2 lentelėje).

2.1.2 lentelė Panaudos atvejo „Grafinių objektų priskyrimas prie nuomos sistemos” specifkacija

<b>Panaudojimo atvejis:</b> Grafinių objektų priskyrimas prie nuomos sistemos		
<b>Aktorius</b>	Vadybininkas	
<b>Sužadinimo sąlyga</b>	Nori priskirti prie nuomos sistemos grafinį objektą	
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs; Turi egzistuoti grafinis objektas; Turi egzistuoti nuomos sistemos objektas;	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	Grafinių objektų administravimas
	<b>Apima PA</b>	
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1. Vadybininkas pasirenka nuomos objektą		Sistema grąžina Pasirinkta
2. Vadybininkas pasirenka grafinį objektą		Sistema grąžina Pasirinkta
3. Vadybininkas pasirenka „Išsaugoti“		Sistema grąžina OK
<b>Po sąlyga</b>		Priskirtas grafinis objektas prie nuomos objekto
<b>Alternatyvūs scenarijai</b>		



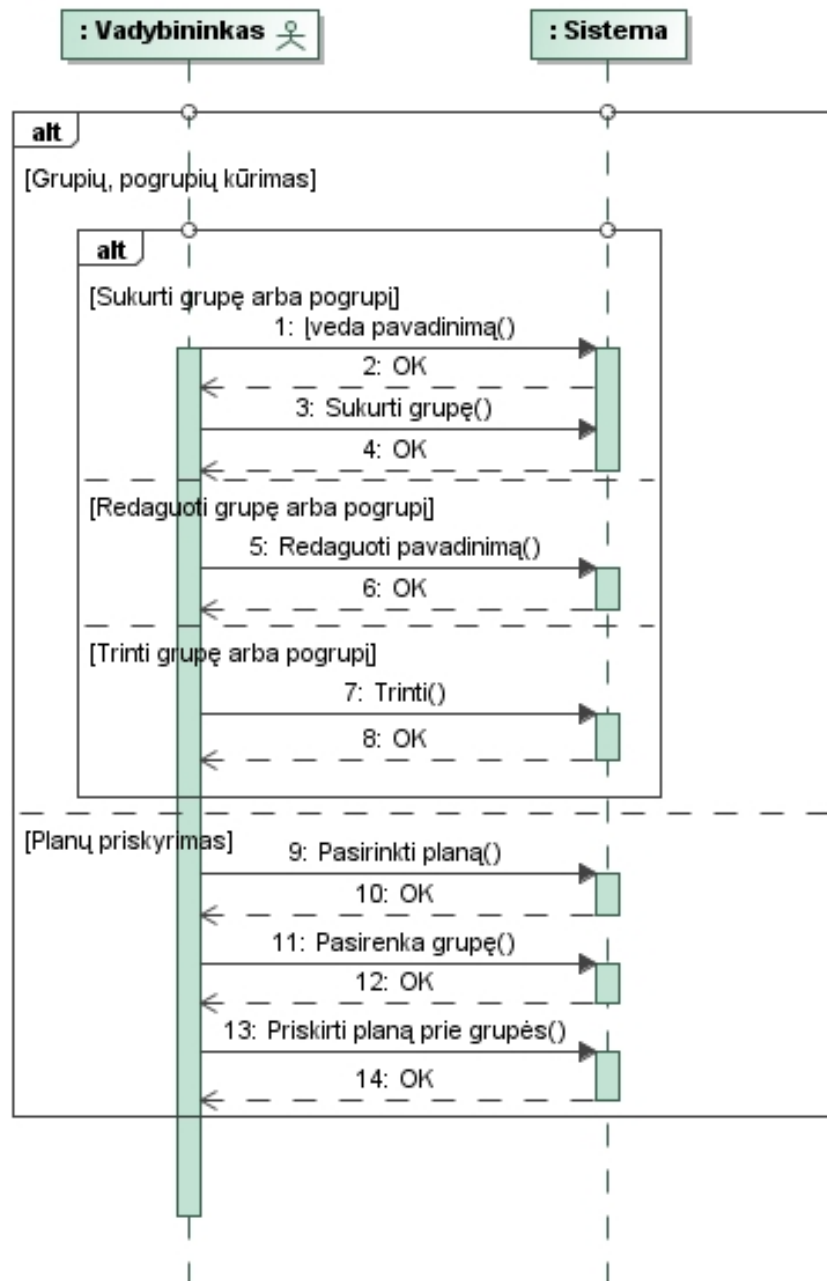
2.1.2 pav. Panaudos atvejo „Grafinių objektų priskyrimas prie nuomos sistemos” sekų diagrama



Panaudos atveju „Grupių, pogrupių valdymas“ pateikiamas esamų grupių ir pogrupių sąrašas, galimybė kiekvieną grupę ar pogrupį valdyti ir atlikti sekančius veiksmus: trinti, redaguoti, sukurti. Grupei galima sukurti pogrupį, kuris būtų kaip detalesnė grupės specifikacija; ištrynus grupę ar pogrupį panaikinami ir kiti elementai, susiję su grupe ar pogrupiu. Grupių, pogrupių valdyme taip pat leidžiama priskirti pogrupiui ar pačiai grupei planą. Vienai grupei ar pogrupiui leidžiamas tik vienas planas (detalesnė specifikacija pateikta. 2.1.3 pav. ir 2.1.3 lentelėje).

**2.1.3 lentelė** Panaudos atvejo „Grupių, pogrupių valdymas“ specifikacija

<b>Panaudojimo atvejis: Grupių valdymas</b>		
<b>Aktorius</b>		Vadybininkas
<b>Sužadinimo sąlyga</b>		Vartotojas nori valdyti grupes
<b>Prieš sąlyga</b>		Vartotojas turi būti prisijungęs;
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	Grafinių objektų administravimas
	<b>Apima PA</b>	
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1.1. Vadybininkas pasirenka sukurti grupę		
1.1.1 Vadybininkas įveda pavadinimą		Sistema grąžina OK
1.1.2 Vadybininkas pasirenka „Sukurti“		Sistema grąžina OK
1.2. Vadybininkas pasirenka „Redaguoti“ ir įveda pavadinimą		Sistema grąžina OK
1.3. Vadybininkas pasirenka „Trinti“		Sistema grąžina OK
2.1. Vadybininkas pasirenka planą		Sistema grąžina OK
2.2. Vadybininkas pasirenka grupę		Sistema grąžina OK
2.3. Vadybininkas pasirenka ir priskiria planą prie grupės		Sistema grąžina OK
<b>Po sąlyga</b>		Sukurtos grupės ir pogrupiai.
<b>Alternatyvūs scenarijai</b>		
1.1. pereina į 2.1 veiksmą		
2.1. pereina į 1.1 veiksmą		

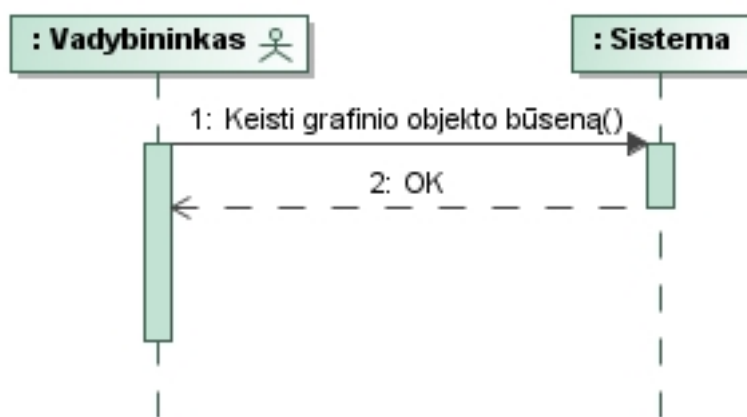


2.1.3 pav. Panaudos atvejo „Grupių, pogrupių valdymas“ sekų diagrama

Panaudos atveju „Grafinių objektų būsenų keitimas“ pateiktas grafinių planų elementų būsenos keitimas (detalesnė specifikacija pateikiama 2.1.4 pav. ir 2.1.4 lentelėje). Vaizduosime grupių bei pogrupių sąrašą su grafinių elementų pavadinimais, taip pat jų plotais.

2.1.4 lentelė Panaudos atvejo „Grafinių objektų būsenų keitimas“ specifikacija

Panaudojimo atvejis: Grafinių objektų būsenų keitimas		
<b>Aktorius</b>	Vadybininkas	
<b>Sužadinimo sąlyga</b>	Nori pakeisti grafinio objekto būseną	
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs; Turi egzistuoti grafinis objektas	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	Grafinių objektų administravimas
	<b>Apima PA</b>	
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1. Vadybininkas pasirenka grafinį objektą		Sistema grąžina OK
2. Keisti būseną.		Sistema grąžina OK
<b>Po sąlyga</b>		Pakeista grafinio objekto būseną.
<b>Alternatyvūs scenarijai</b>		

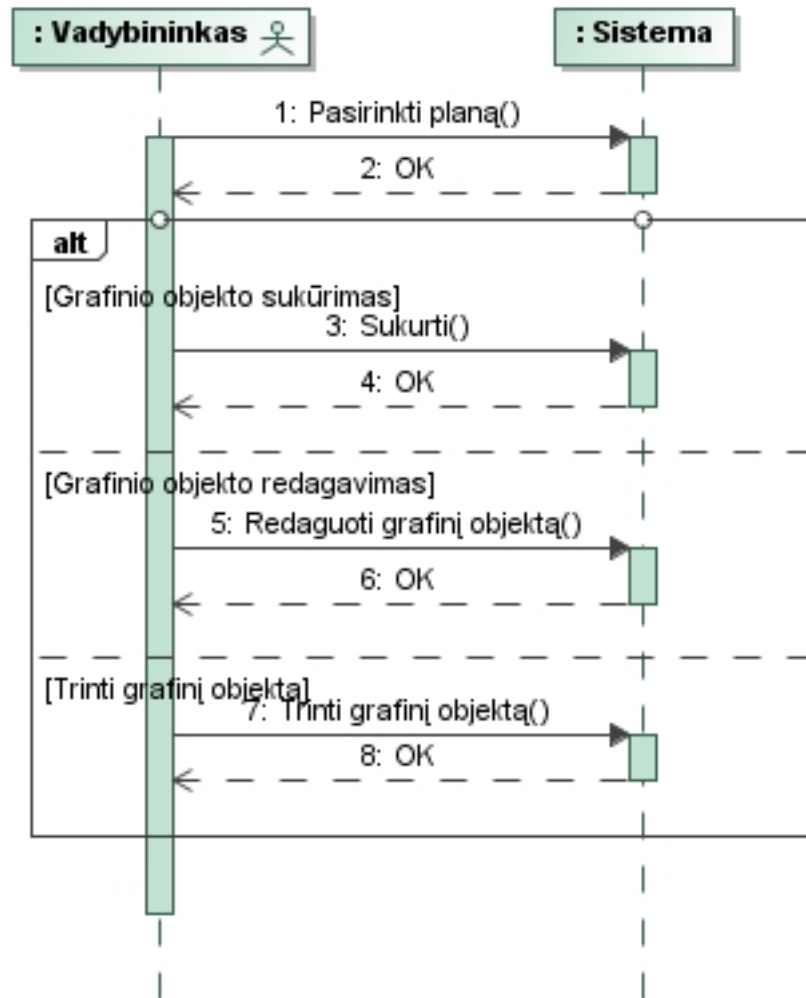


2.1.4 pav. Panaudos atvejo „Grafinių objektų būsenų keitimas“ sekų diagrama

Grafinių objektų valdyme galima pasirinkti vieną ar kelis grafinius objektus, juos vienu metu redaguoti, trinti, keisti jų pozicijas bei keisti pavadinimą (detalesnė specifikacija pateikiama. 2.1.5 pav. ir 2.1.5 lentelėje).

2.1.5 lentelė Panaudos atvejo „Grafinių objektų valdymas“ specifikacija

<b>Panaudojimo atvejis: Grafinių objektų valdymas</b>		
<b>Aktorius</b>		Vadybininkas
<b>Sužadinimo sąlyga</b>		Vartotojas nori sukurti grafinį objektą
<b>Prieš sąlyga</b>		Vartotojas turi būti prisijungęs; Turi būti sukurtos grupės bei pogrupiai.
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	Grafinių objektų administravimas
	<b>Apima PA</b>	Grafinių objektų kūrimas Grafinių objektų redagavimas Grafinių objektų trynimasis
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1. Vadybininkas pasirenka planą		Sistema grąžina OK
2. Vadybininkas atlieka pasirinktą veiksmą		Sistema grąžina OK
<b>Po sąlyga</b>		Sukurtas grafinis objektas.
<b>Alternatyvūs scenarijai</b>		

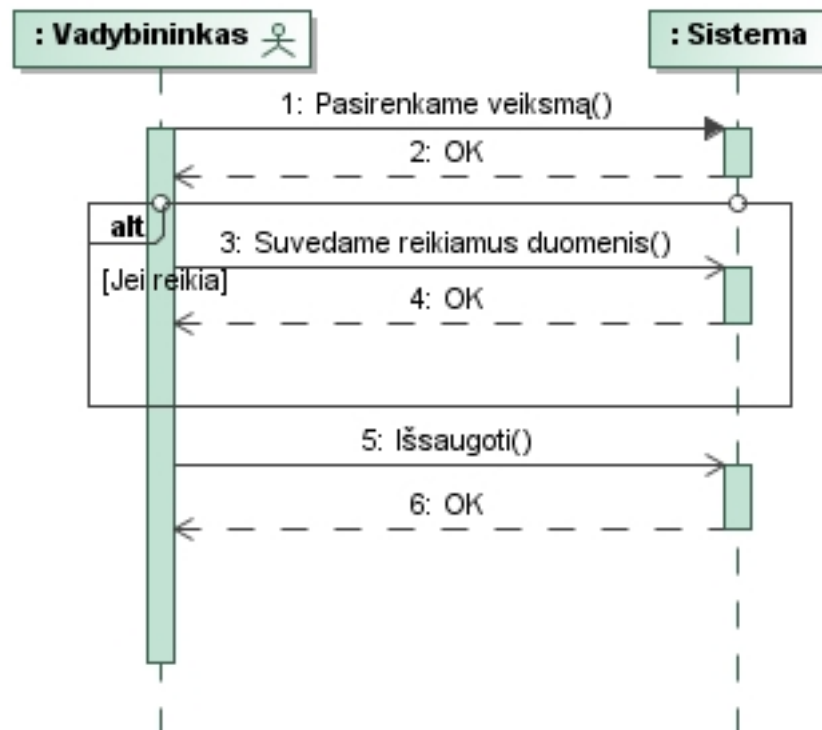


2.1.5 pav. Panaudos atvejo „Grafinių objektų valdymas“ sekų diagrama

Vieną iš panaudojimo atvejų „Planų valdymas“ detalizuoja 2.1.6 lentelė ir 2.1.6 pav. Planų valdyme galima: matyti planų sąrašą, trinti, redaguoti ir kurti planus. Vartotojas (vadybininkas) suvedęs planus gali atlikti pasirinktą veiksmą iš prieš tai išvardintų veiksmų.

2.1.6 lentelė Panaudos atvejo „Planų valdymas“ specifikacija

<b>Panaudojimo atvejis:</b> Grafinių objektų valdymas		
<b>Aktorius</b>	Vadybininkas	
<b>Sužadinimo sąlyga</b>	Vartotojas nori sukurti planą	
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs;	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečia PA</b>	Grafinių objektų administravimas
	<b>Apima PA</b>	
	<b>Specializuoja PA</b>	
<b>Pagrindinis įvykių srautas</b>		<b>Sistemos reakcija ir sprendimai</b>
1. Vadybininkas pasirenka veiksmą		Sistema grąžina OK
2. Vadybininkas išsaugoja		Sistema grąžina OK
<b>Po sąlyga</b>		Sukurtas planas
<b>Alternatyvūs scenarijai</b>		
1. jei reikia vadybininkas dar suveda reikiamus formos duomenis		



2.1.6 pav. Panaudos atvejo „Planų valdymas“ sekų diagrama

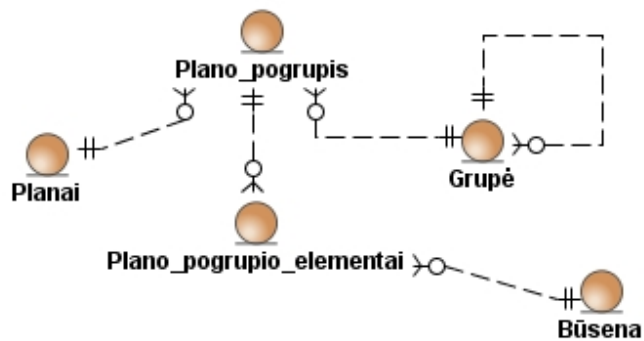
### Nefunkciniai reikalavimai:

- Išvaizdos ir patrauklumo (Look and Feel) reikalavimai: produkto išvaizdos apibūdinimas. Visas produkto vaizdas turi atitikti numatytojo dizaino vaizdą;
- Naudojimo (Usability) reikalavimai – produkto naudojimo patogumas;
- Produktyvumo didėjimas pradėjus naudoti produktą - sumažės laiko sąnaudos ruošiant naujienlaiškių siuntimo šabloną;
- Klaidų galimumas: atsiradus klaidoms ir apie tai pranešus produkto kūrėjui, kūrėjas turi priimti atsakomybę ir ištaisyti klaidas;
- Produkto naudojimas, kai vartotojai yra kitakalbiai: vartotojams suteikiama galimybė pasikeisti kalbinę versiją;
- Produkto naudojimo galimybės, kai vartotojai neturi patirties: pravedami apmokymo kursai ir pateikiama dokumentacija kaip naudotis produktu;
- Našumo (Performance) reikalavimai – kiek spartus, funkcionalus, tikslus turi būti produktas:
  - Atvaizduojamų grafinių objektų tikslumas priklauso tik nuo administruojančio vartotojo tikslumo. Sistemos kūrėjai neprisiima atsakomybės dėl grafinių objektų netikslumo, jei tai nėra programinė klaida;
  - Produkto sparta funkciniam lygyje turi būti didesnė, nei prieš tai buvusio produkto;
- Veikimo (Operational) reikalavimai – kokioje aplinkoje turi veikti produktas, kokie papildomi reikalavimai keliami aplinkai:
  - Stebėti, ar serveryje neįvyko pakeitimų, serverio administratoriams nepranešus - sistemą naudojančiams klientams;
  - Perkeliant produktą iš vieno serverio į kitą reikia nepamiršti esminių elementų:
    - Kodavimo;
    - Techninių reikalavimų:
      - Apache 1.3x arba 2.x;
      - Php 4.x arba 5.x;
      - Mysql 4.x arba 5.x;
    - Saugumo (Security) reikalavimų, t. y. - produkto saugumo ir konfidencialumo:
      - Prie sistemos gali prisijungti tik tie vartotojai, kuriems buvo sukurtos teisės prieiti prie produkto;
      - Redaguoti ir modifikuoti gali tik tie vartotojai, kuriems tai leidžiama daryti;

- Kultūriniai ir politiniai (Cultural and Political) reikalavimai: specialūs reikalavimai susiję su žmonėmis, kurie naudosis šiuo produktu. Produktas turi palaikyti kelias kalbines versijas, t.y. kad vartotojai galėtų produktą naudoti ir lietuvių, ir anglų kalbomis;
- Teisiniai (Legal) reikalavimai: kokius įstatymus ir standartus turi atitikti produktas. Administruojantys vartotojai yra patys atsakingi už teisinių reikalavimų pažeidimus, jei tai nėra susiję su programine dalimi, t. y. jie atsakingi už publikuojamos informacijos konfidencialumą.

## 2.2 Dalykinės srities modelis

Dalykinės srities modelis vaizduoja duomenų bazės sąsajas tarp pagrindinių objektų: grupės, pogrupio, planų ir būsenų. Visa tai atvaizduota 2.2 pav. detalizuotas planas pavaizduotas sistemos projekte.



2.2 pav. Dalykinės srities esybių modelis



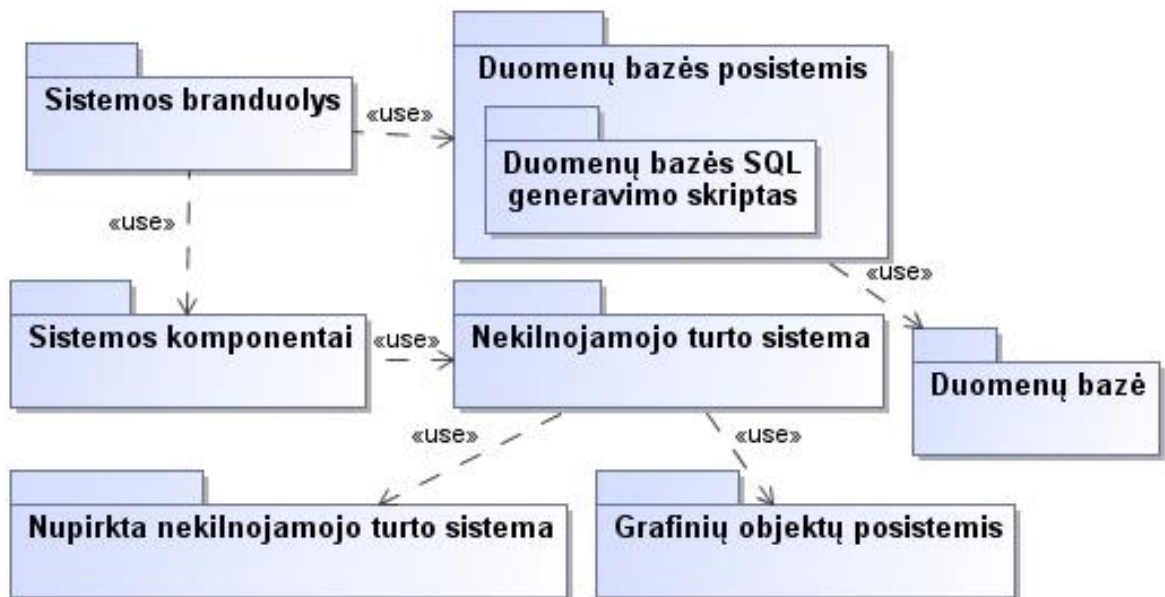
### 3 Sistemos projektas

Pagal suformuotus funkcinis ir nefunkcinis sistemos reikalavimus, sudaroma sistemos architektūra.

#### 3.1 Sistemos architektūros projektas

##### 3.1.1 Sistemos loginė architektūra

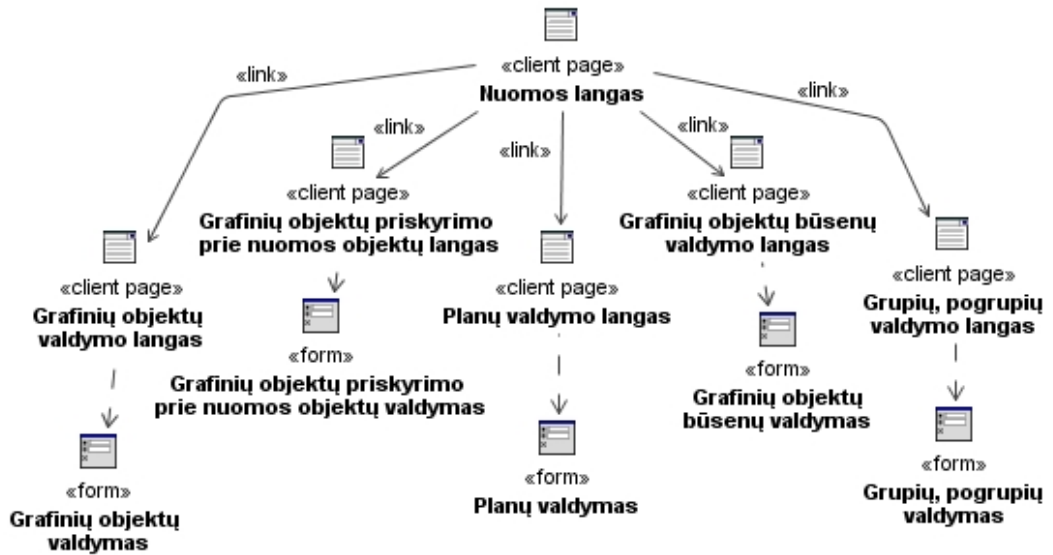
Sistemos loginė architektūra 3.1.1 pav. parodo jos pagrindinius architektūrinius komponentus taip pat grafinių objektų posistemio sąsaja su nekilnojamo turto sistema ir pačia sistema.



3.1.1 pav. Sistemos loginė architektūra

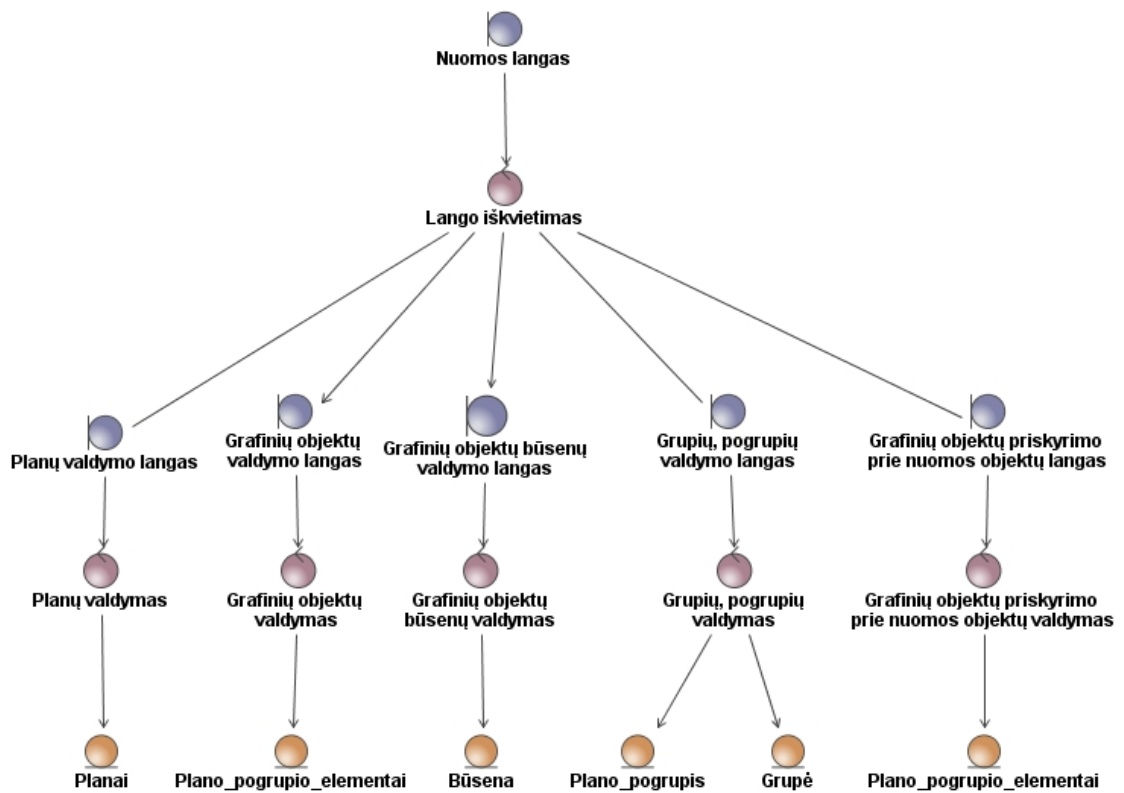
##### 3.1.2 Vartotojo paslaugos (sąsaja, sąsajos navigavimo planas)

Vartotojo sąsajos modelis vaizduoja kaip galima patekti bendrai į kiekvieną reikiama elementą (žr. 3.1.2.1 pav.). Vartotojo sąsajos modelis taip pat vaizduoja pagrindinius objektus, kurie - naudojami atskiruose languose.



3.1.2.1 pav. Vartotojo sąsajos navigacijos planas

Robastiškumo diagramos padeda orientuotis kokie objektai, kada iškviečiami ir kaip vienas su kitu objektu siejasi (žr. 3.1.2.2 pav.). Robastiškumo diagrama sudaryta iš vartotojo navigacijos plano, naudojant sąsaja su esybėmis.

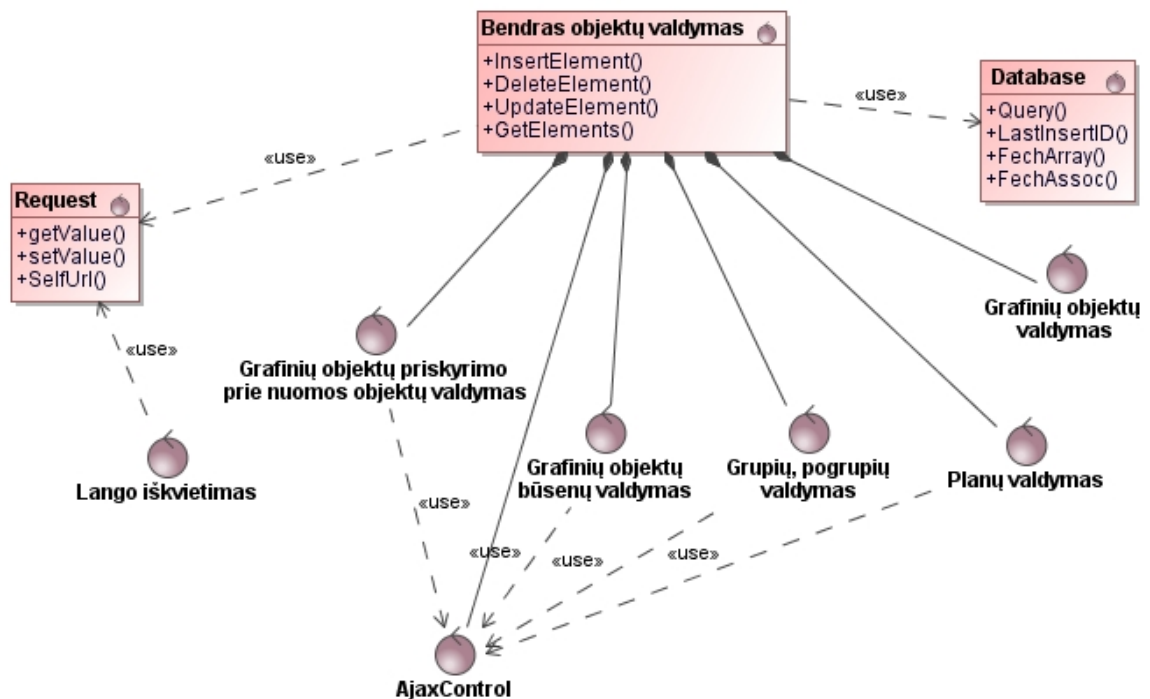


3.1.2.2 pav. Bendra robastiškumo diagrama

### 3.1.3 Veiklos paslaugos

Sistemos veiklos logika yra realizuojama valdymo klasėmis (žr. 3.1.3.1 pav.) ir nusako, kaip valdymo klasės yra tarpusavyje susijusios. Sudarant valdymo klases pridedamos papildomos klasės, kurios nėra vaizduojamos robastiškumo diagramoje, tačiau be jų negalima įsivaizduoti projekto tolesnio tobulinimo arba papildymo.

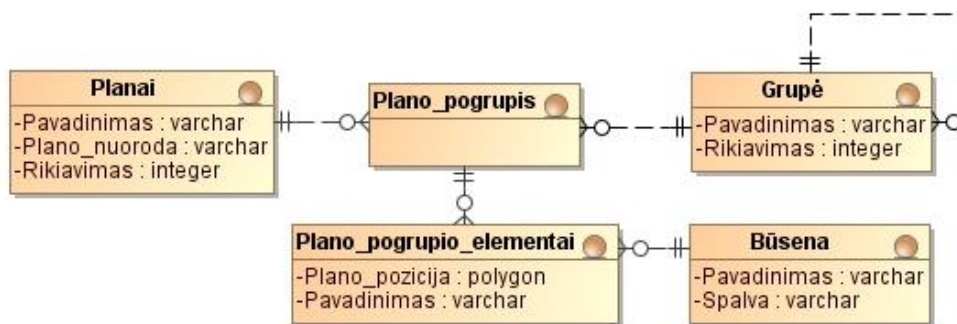
Visos valdymo klasės yra labai glaudžiai susijusios, nes vienos jų iškviečia kitas. Dauguma iš objektų turi vieną bendrą objektą, kurį kitos klasės paveldi. „Bendras objektų valdymas“ – abstrakti klasė, nurodanti, kad vaikinės klasės turės tas funkcijas, kurias turi ji pati.



3.1.3.1 pav. Sistemos valdymo klasės

### 3.1.4 Duomenų paslaugos

Dalykinės srities esybių klasių modelis pateiktas 3.1.4 paveikslėlyje. Šios klasės yra skirtos duomenims saugoti, jos neturi metodų.



3.1.4 pav. Dalykinės srities esybių klasių modelis

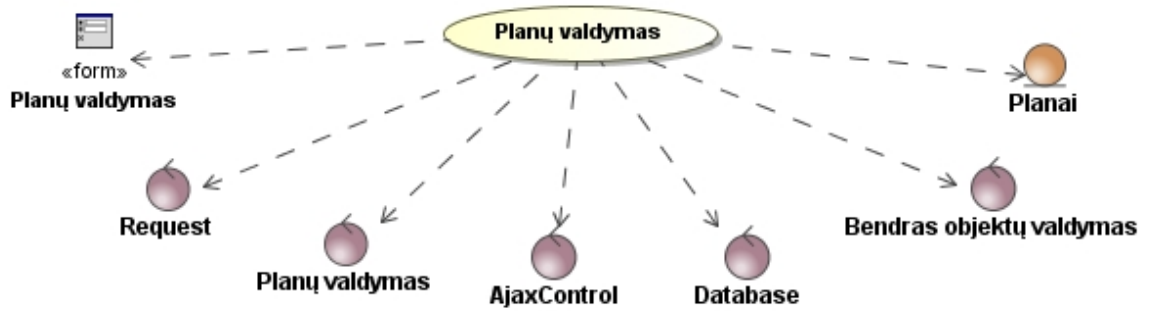
Kadangi esybių klasių modelis yra transformuojamas į duomenų bazės schemą, čia kiekvienos klasės atributas nėra paaiškintas, nes detalūs lentelių aprašymai pateikti 3.4 punkte. Kiekvienos esybės paskirtis, t. y. kokią informaciją ji saugo, pateikta 3.1.4 lentelėje.

3.1.4 lentelė Dalykinės srities esybių paskirtis

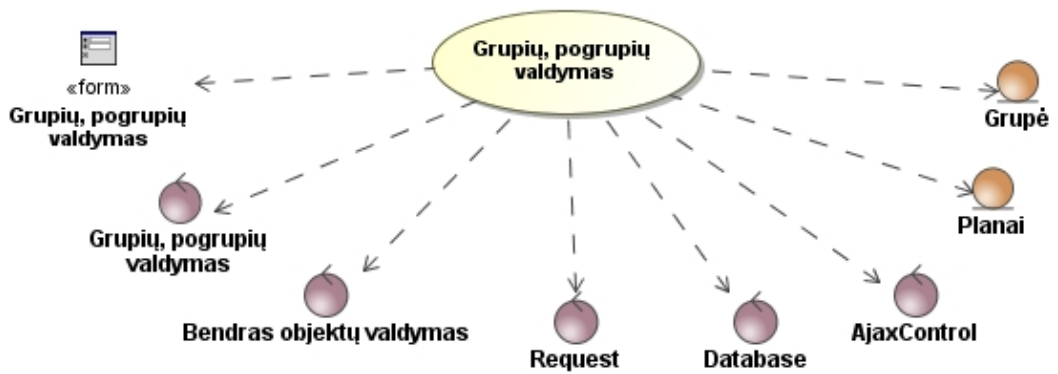
ESYBĖ	SAUGOMA INFORMACIJA
PLANAI	Duomenys apie planus: pavadinimas, planas, rikiavimas
PLANO POGRUPIS	Saugoma sąsaja tarp plano ir grupės
GRUPĖ	Grupių ir pogrupių informacija: pavadinimas ir rikiavimas
PLANO POGRUPIO ELEMENTAI	Saugoma visa informacija apie grafinį objektą: pavadinimas, būsena, taškų visuma
BŪSENA	Galimos plano pogrupio būsenos pavadinimai ir spalva

### 3.2 Detalus projektas

Kiekvienam panaudos atvejui („Planų valdymas“, „Grupių, pogrupių valdymas“, „Grafinių objektų priskyrimas prie nuomos sistemos“, „Grafinių objektų būsenų keitimas“, „Grafinių objektų valdymas“ atitinkamai vaizduojami paveikslėliuose 3.2.1 – 3.2.5) sudaroma realizacijos diagrama. Diagramose vaizduojama tai, kas sudaro panaudos atvejį: esybės, kontrolieriai ir formos. Tačiau realizacijos diagrama neatspindi pačios sistemos architektūros, metodų efektyvumo bei pačių objektų reikalingumo. Realizacijoje šios diagramos gali tik iš dalies atitikti patį projektavimą. Taip pat ir darant realizaciją yra atsižvelgiama į projektavimą.



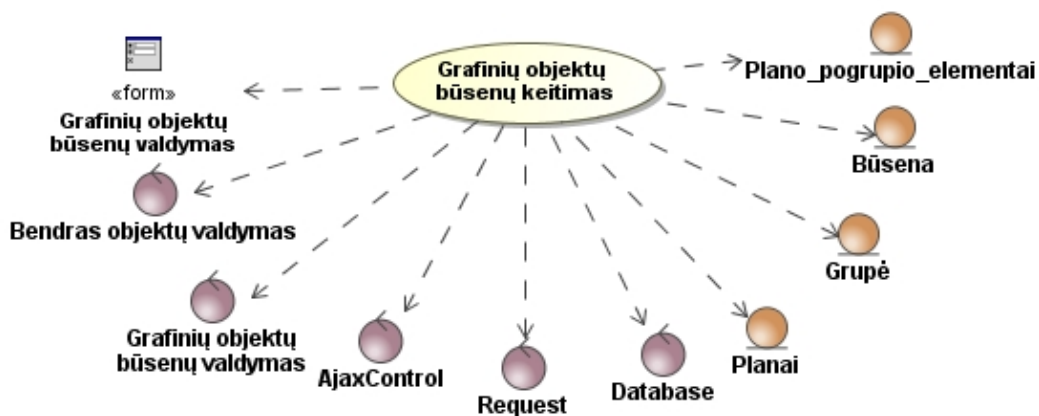
3.2.1 pav. Panaudos atvejo „Planų valdymo“ realizacijos diagrama



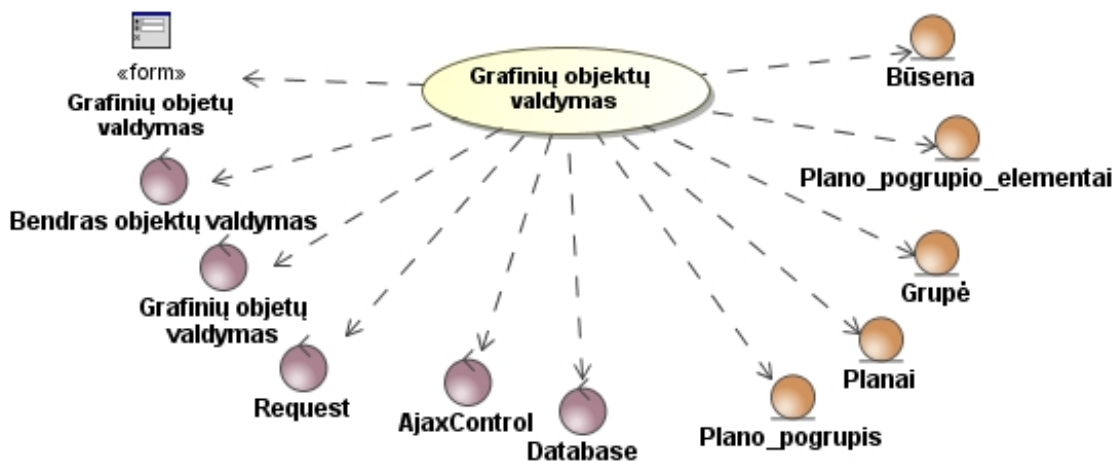
3.2.2 pav. Panaudos atvejo „Grupių, pogrupių valdymas“ realizacijos diagrama



3.2.3 pav. Panaudos atvejo „Grafinių objektų priskyrimas prie nuomos sistemos“ realizacijos diagrama

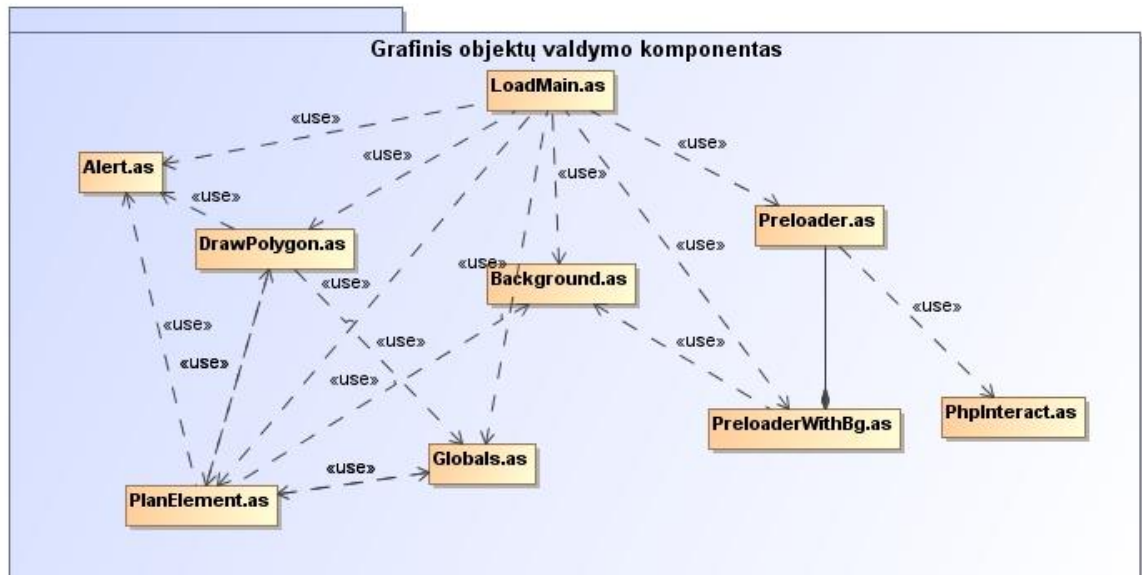


3.2.4 pav. Panaudos atvejo „Grafinių objektų būsenų keitimas“ realizacijos diagrama



3.2.5 pav. Panaudos atvejo „Grafinių objektų valdymas“ realizacijos diagrama

Kuriamas grafinių objektų valdymo komponentas yra specifinis, jame naudojama architektūra pavaizduota 3.2.6 paveikslėlyje. Pagrindinė klasė „LoadMain“ prieš užkranant FLASH identifikuojamas objektas „Alert“. Tik užkrovus langą yra vykdoma „PreloaderWithBg“ t. y. vykdomas FLASH aplikacijos krovimasis. Užsikrovus FLASH sukuriama „Background“ objektas su tekstu, vėliau daroma sąsaja su HTML, tai leidžia vykdyti FLASH funkcijas iš HTML, tos funkcijos sukuria „PlanElement“ objektą, vėliau sukuriama „DrawPolygon“ objektas ir priskiriamas kaip vaikas elementui „PlanElement“.

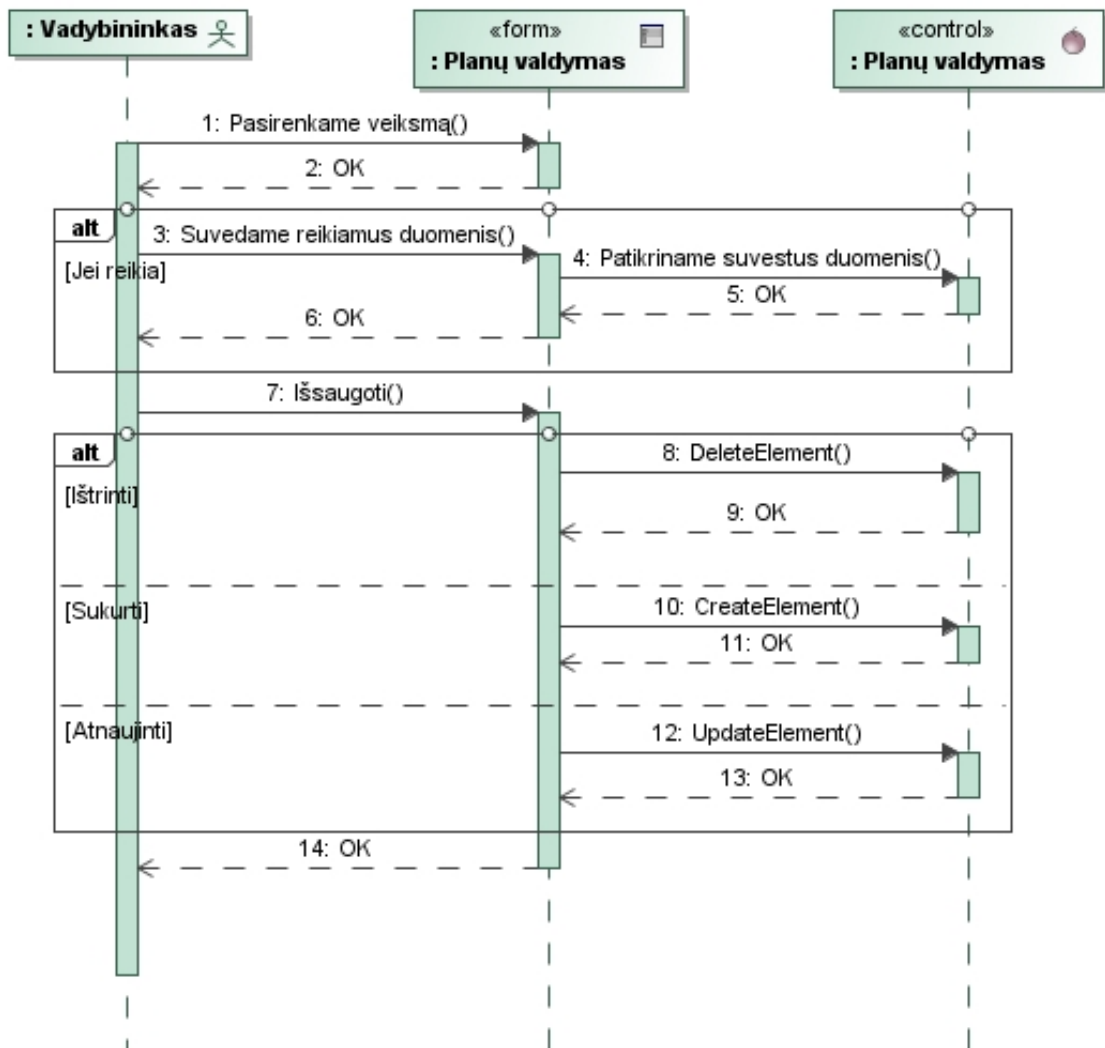


3.2.6 pav. Grafinių objektų valdymo klasių sąryšiai kuriant komponentą

### 3.3 Sistemos elgsenos modelis

Sudarytos panaudos atvejų sekos diagramos, kurios rodo bendravimą tarp klasių, realizuojančių panaudos atvejus.

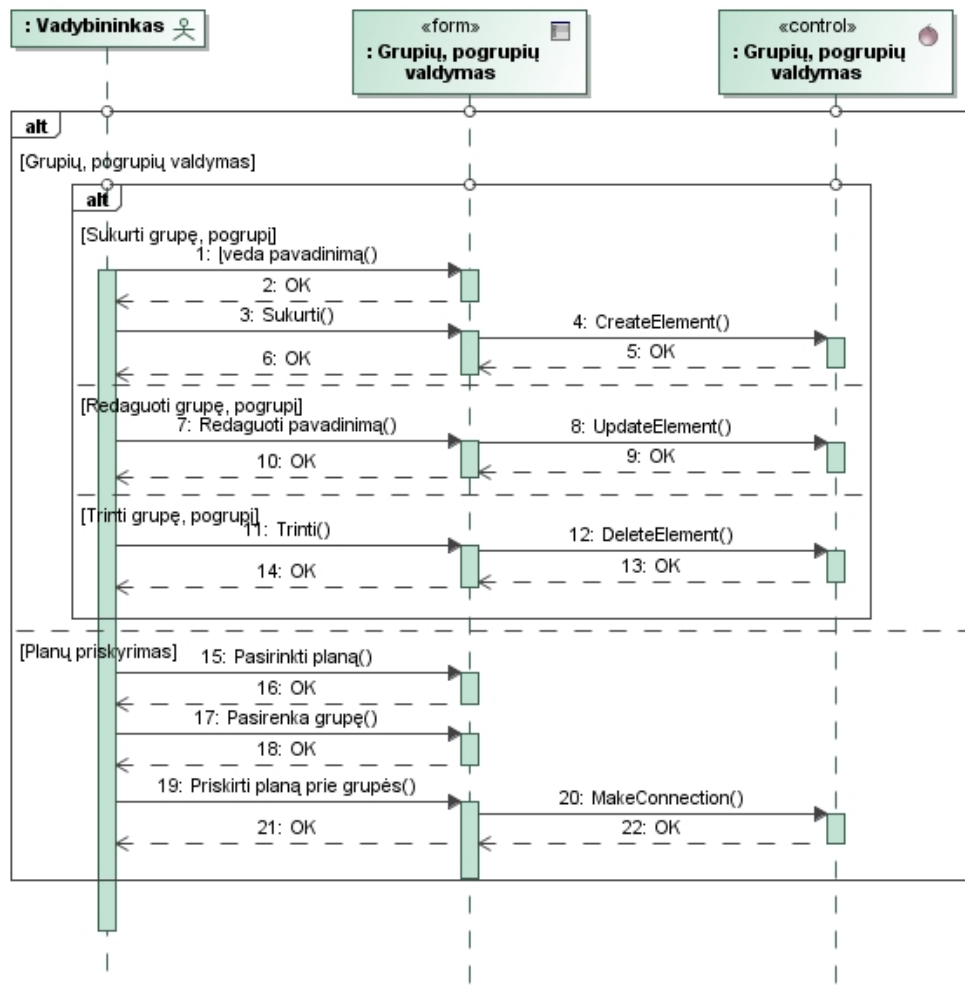
Panaudos atvejo „Planų valdymas“ realizacijoje dalyvauja planų valdymo forma ir planų valdymo klasė. Detalesnė specifikacija pateikiama 3.3.1 pav.



3.3.1 pav. Panaudos atvejo „Planų valdymas“ sekų diagrama

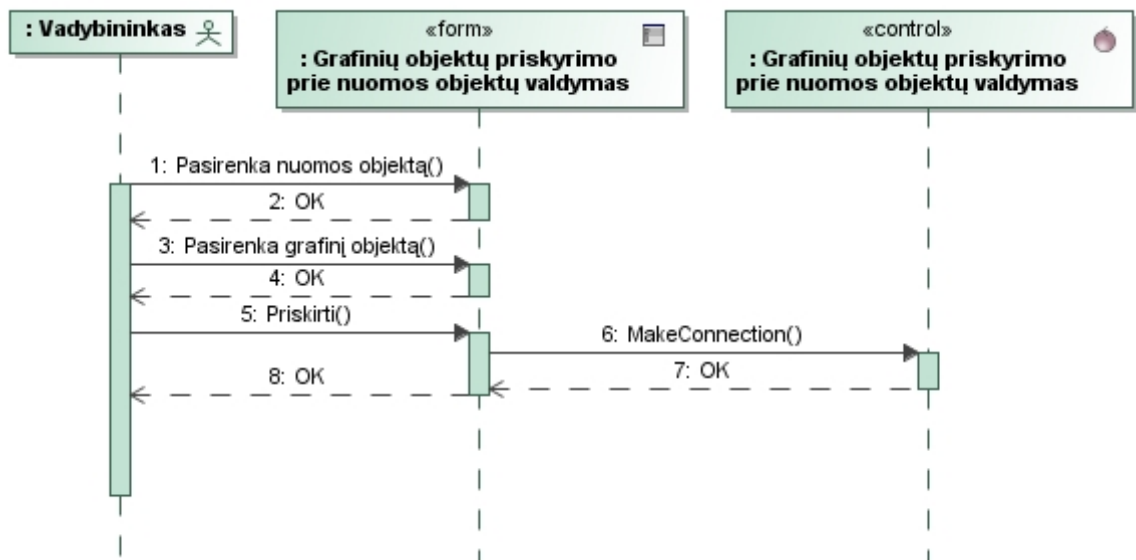
Panaudos atvejo „Grupių, pogrupių valdymas“ realizacijoje dalyvauja: grupių, pogrupių valdymo forma ir planų valdymo klasė. Detalesnė specifikacija pateikiama 3.3.2 pav.





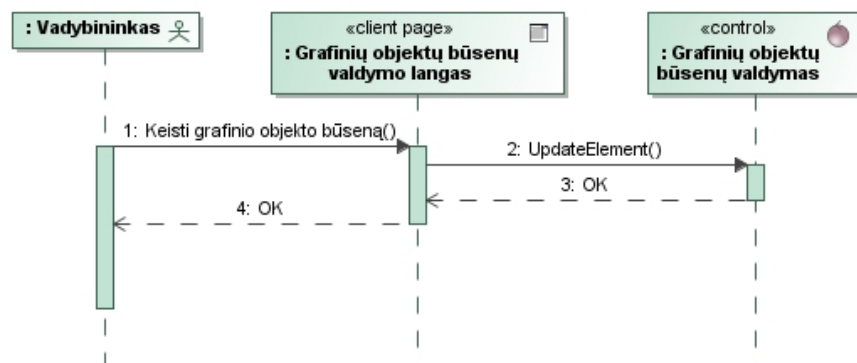
3.3.2 pav. Panaudos atvejo „Grupių, pogrupių valdymas“ sekų diagrama

Panaudos atvejo „Grafinių objektų priskyrimo prie nuomos objektų valdymas“ realizacijoje dalyvauja Grafinių objektų priskyrimo prie nuomos objektų valdymo forma ir Grafinių objektų priskyrimo prie nuomos objektų valdymo klasė. Detalesnė specifikacija pateikiama 3.3.3 pav.



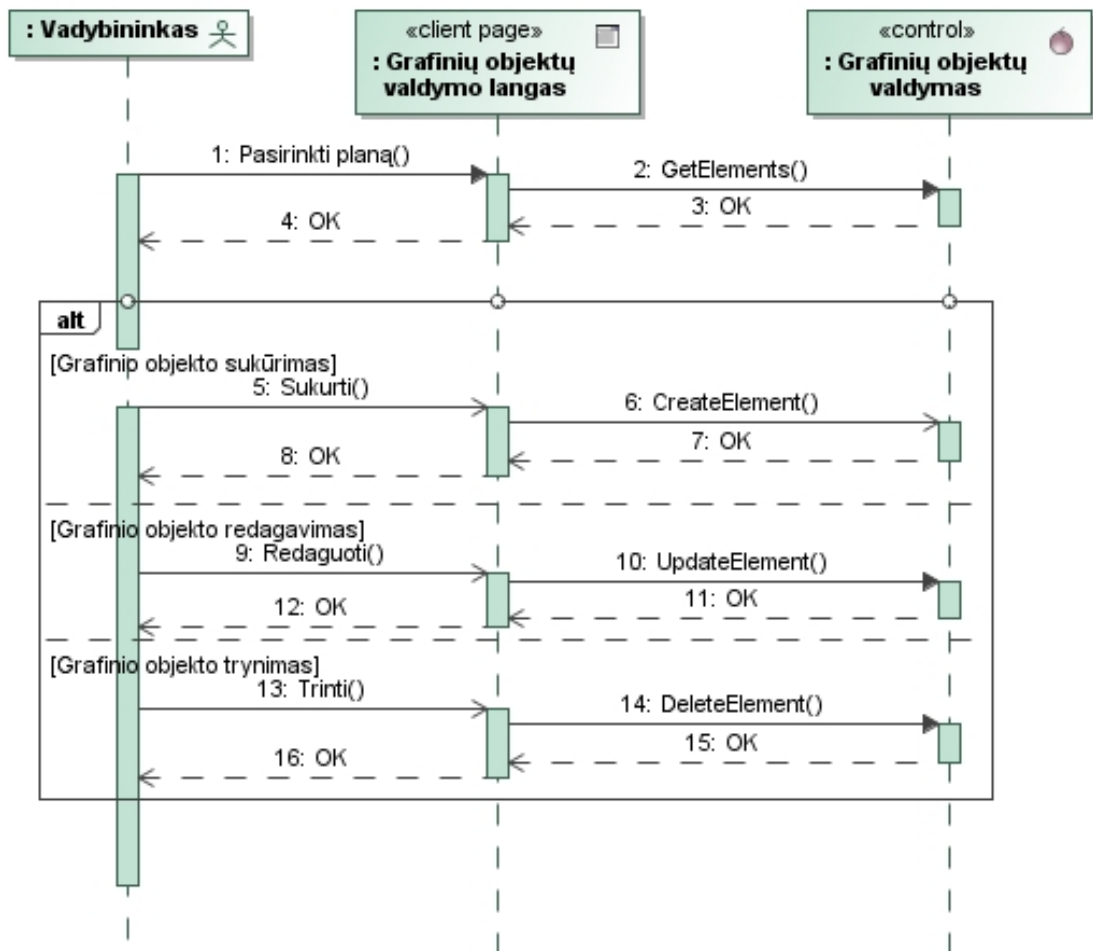
3.3.3 pav. Panaudos atvejo „Grafinių objektų priskyrimo prie nuomos objektų valdymas“ sekų diagrama

Panaudos atvejo „Grafinių objektų būsenų valdymo langas“ realizacijoje dalyvauja: Grafinių objektų būsenų valdymo forma ir Grafinių objektų būsenų valdymo klasė. Detalesnė specifikacija pateikiama 3.3.4 pav.



3.3.4 pav. Panaudos atvejo „Grafinių objektų būsenų valdymas“ sekų diagrama

Panaudos atvejo „Grafinių objektų valdymas“ realizacijoje dalyvauja: Grafinių objektų valdymo forma ir Grafinių objektų valdymo klasė. Detalesnė specifikacija pateikiama 3.3.5 pav.

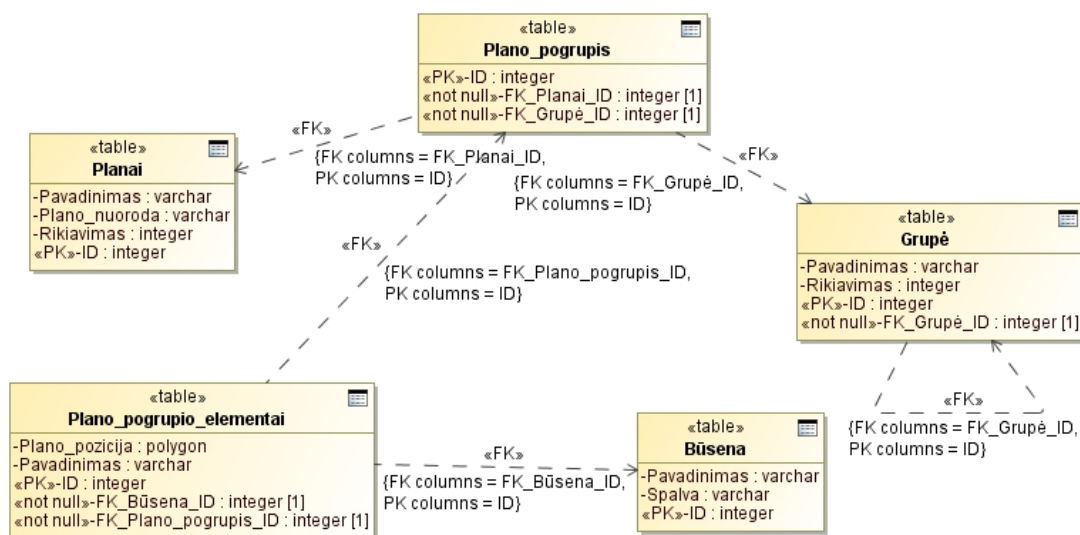


3.3.5 pav. Panaudos atvejo „Grafinių objektų valdymas” sekų diagrama

### 3.4 Duomenų bazės schema

Duomenų bazės schema transformuojama iš dalykinės srities esybių modelio, transformuotas esybių modelis atvaizduojamas 3.4 pav. Šioje duomenų bazėje naudojamas poligono elemento tipas. Naudojamas šis duomenų bazės tipo atributas suteikia labai daug galimybių, kai reikia atlikti tam tikrus veiksmus su grafiniu objektu, pavyzdžiui: galima suskaičiuoti objekto plotą, apskaičiuoti vidurio tašką (taškų centrinį objektą).

Detalesnė lentelių specifikacija apie lentelių atributus pateikta 3.4.1, 3.4.2, 3.4.3, 3.4.4 ir 3.4.5 lentelėse.



3.4 pav. Transformuotas dalykinės srities esybių modelis į duomenų bazės schema

3.4.1 lentelė Duomenų bazės lentelės „Grupė“ specifikacija

Atributas	Tipas	Paskirtis
ID	Integer	Pirminis raktas
FK_Grupės_ID	Integer	Išorinis raktas.
Pavadinimas	Varchar	Grupės pavadinimas
Rikiavimas	Integer	To paties lygio rikiavimas

3.4.2 lentelė Duomenų bazės lentelės „Plano\_pogrupis“ specifikacija

Atributas	Tipas	Paskirtis
ID	Integer	Pirminis raktas
FK_Planai_ID	Integer	Išorinis raktas.
FK_Grupės_ID	Integer	Išorinis raktas.

3.4.3 lentelė Duomenų bazės lentelės „Plano\_pogrupio\_elementai“ specifikacija

Atributas	Tipas	Paskirtis
ID	Integer	Pirminis raktas
FK_Plano_pogrupis_ID	Integer	Išorinis raktas.
FK_Būseną_ID	Integer	Išorinis raktas.
Plano_pozicija	Polygon	Taškų visuma
Pavadinimas	Varchar	Plano elemento pavadinimas

3.4.4 lentelė Duomenų bazės lentelės „Planai“ specifikacija

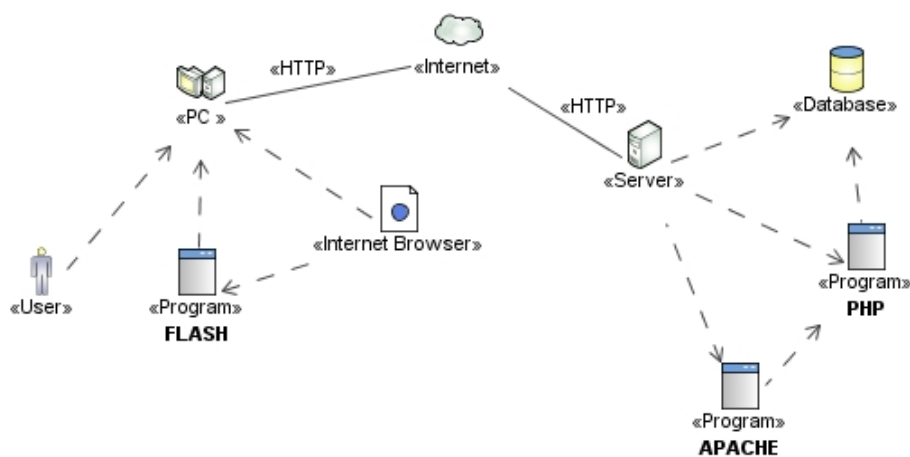
Atributas	Tipas	Paskirtis
ID	Integer	Pirminis raktas
Pavadinimas	Varchar	Plano pavadinimas
Plano_nuoroda	Varchar	Nuoroda iki paveikslėlio
Rikiavimas	Integer	To paties lygio rikiavimas

3.4.5 lentelė Duomenų bazės lentelės „Būseną“ specifikacija

Atributas	Tipas	Paskirtis
ID	Integer	Pirminis raktas
Pavadinimas	Varchar	Būsenos pavadinimas
Spalva	Varchar	Spalvos kodas HEX formatu

### 3.5 Realizacijos modelis

Techninė sistemos architektūra (žr. 3.5 pav.) vaizduoja pagrindinius sistemos elementus, kurie turi būti įdiegti serveryje ir vartotojo kompiuteryje, norit naudotis minimalia sistema, nepaisant kitų naudojamų sistemų. Serverio reikalavimai taip pat yra minimi reikalavimų specifikacijos nefunkciniuose reikalavimuose.



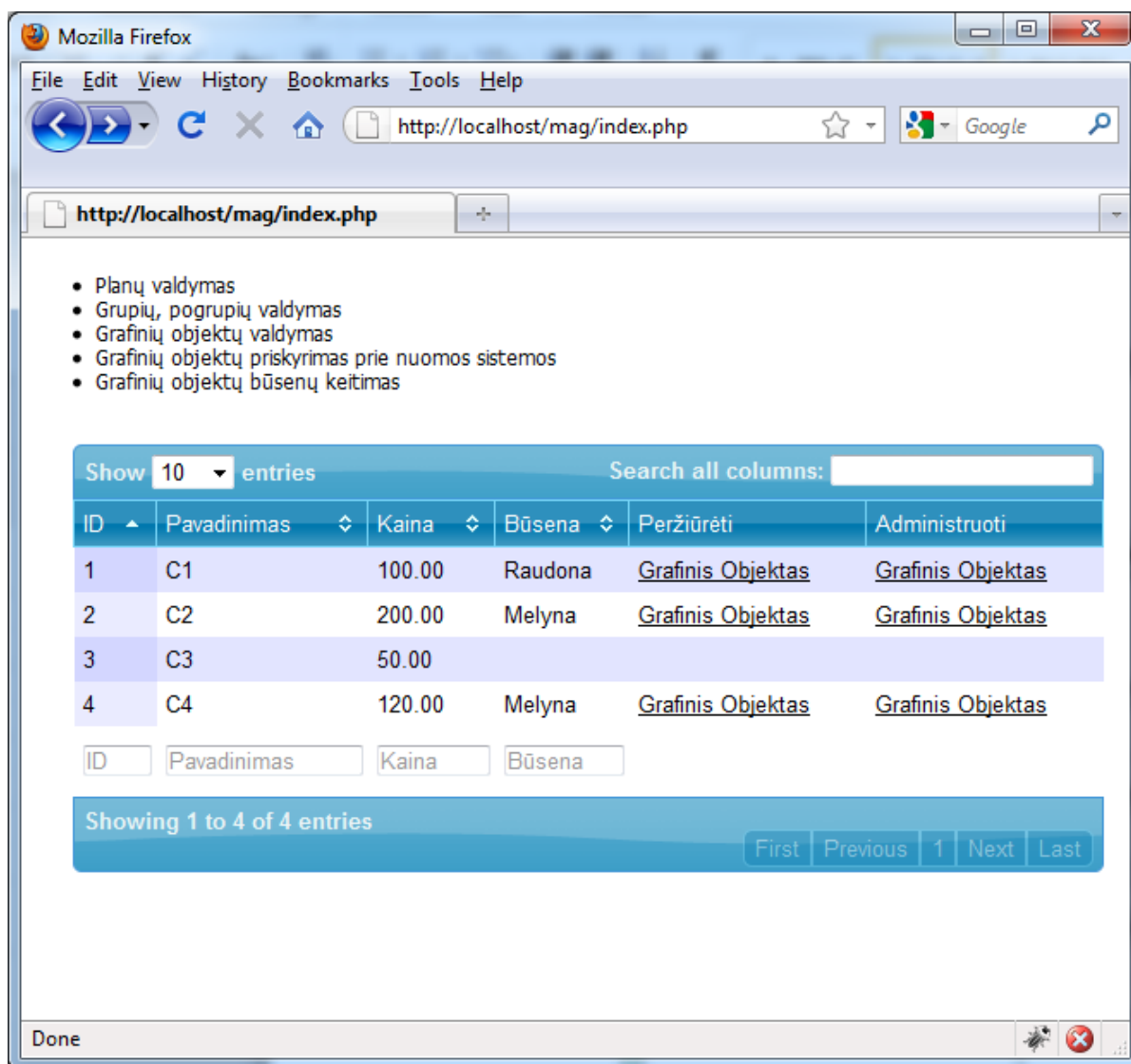
3.5 pav. Techninė sistemos architektūra

## 4 Sistemos realizacija

Ankstesniame skyriuje buvo darytas sistemos projektas, pagal projektą atlikta sistemos realizacija. Šiame skyriuje aprašyta sistemos realizacija, veikimo aprašymas ir sistemos testavimas.

### 4.1 Sistemos veikimo aprašymas

Pradedant dirbti su sistema užkraunamas pradinis sistemos langas (4.1.1 pav.), kuriame galima pasirinkti tolimesnius veiksmus. Paveikslėlio turinio viršuje pateiktas sąrašas veiksmų, kuriuos galima atlikti, žemiau – nuomos sistemos elementų sąrašas, kuriame galima atlikti filtravimą pagal visus raktinius žodžius arba pagal kiekvieną stulpelį atskirai. Šią lentelę galima redaguoti pagal užsakovo norus ir reikalavimus.

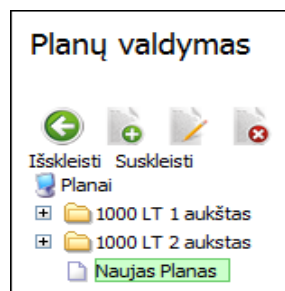


4.1.1 pav. Pradinis sistemos langas

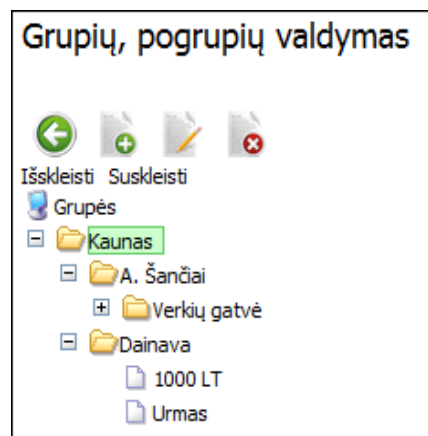
Prieš pradėdant darbą su grafinais objektais reikia:

- susikurti planus;
- sukurti grupes ir pogrupius;
- priskirti planus prie pogrupio arba grupės;
- užsikrauti grafinių objektų valdymą;
- pasirinkti grupę kurioje priskirtas planas;
- pasirinkti planą iš grupės;
- valdyti grafinius objektus.

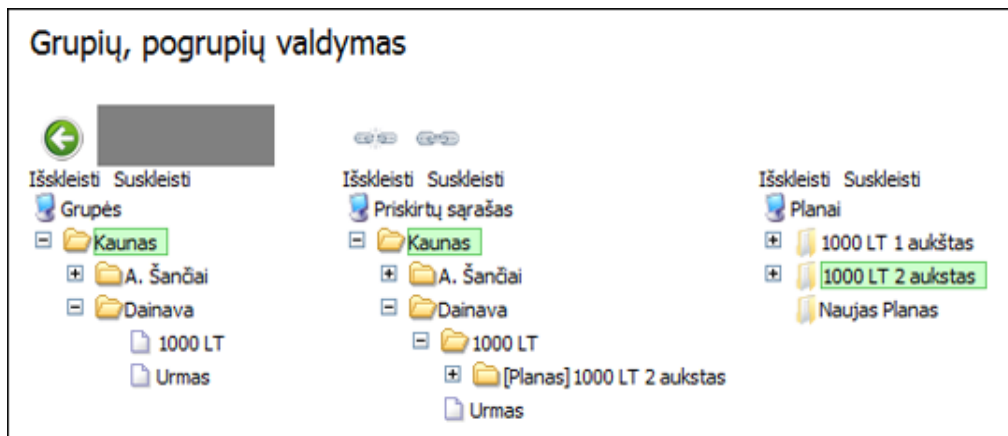
Žemiau pateikti visi etapų paveikslėliai 4.1.2 – 4.1.5.



4.1.2 pav. Planų valdymo langas

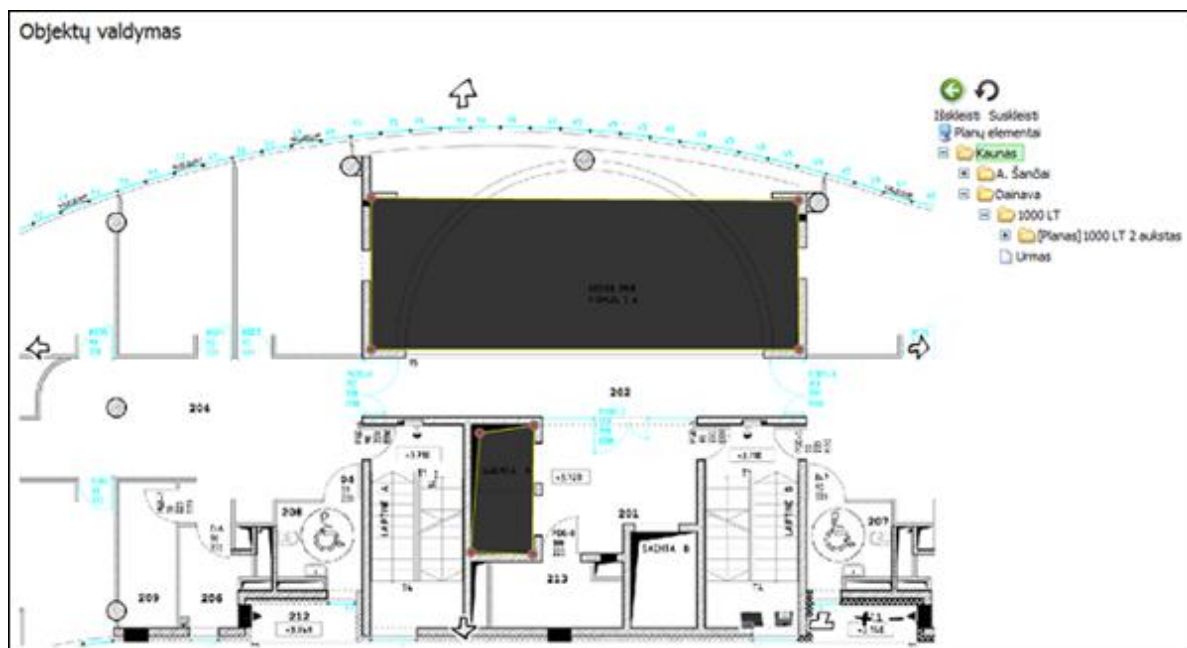


4.1.3 pav. Grupių, pogrupių valdymas



4.1.4 pav. Planų priskyrimas prie grupių

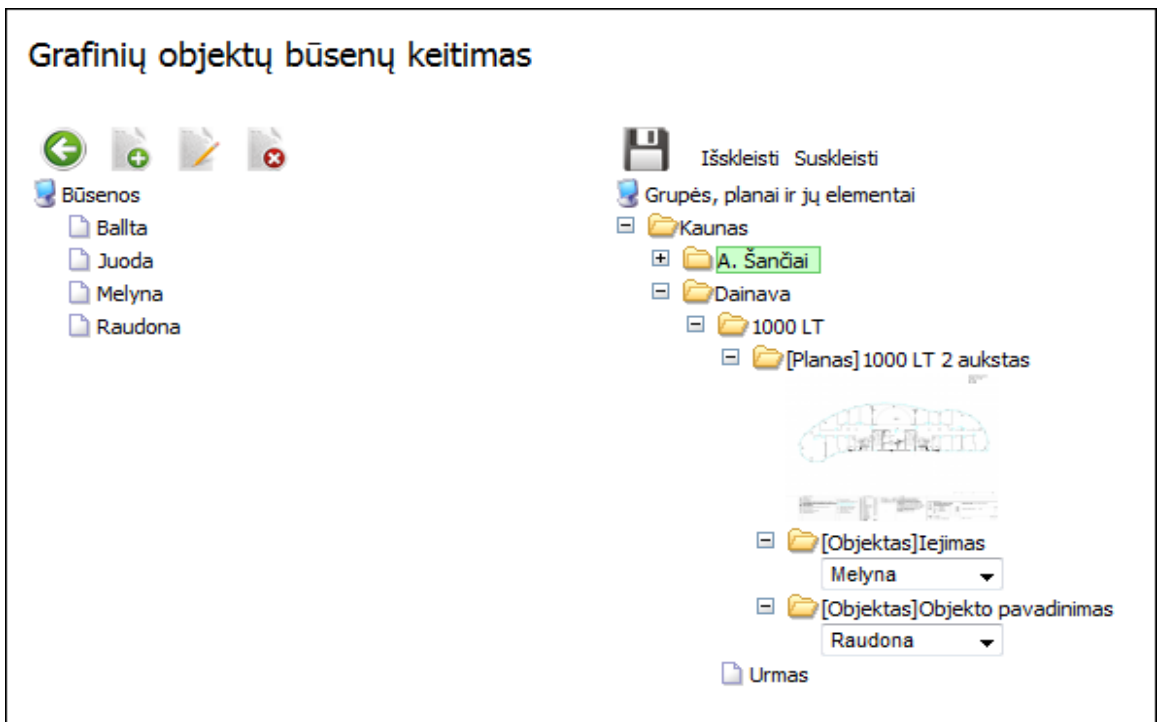
4.1.4 paveikslėlyje užtūšuota pilka vieta planų priskyrimams nenaudojama, ji naudojama grupių valdymui.



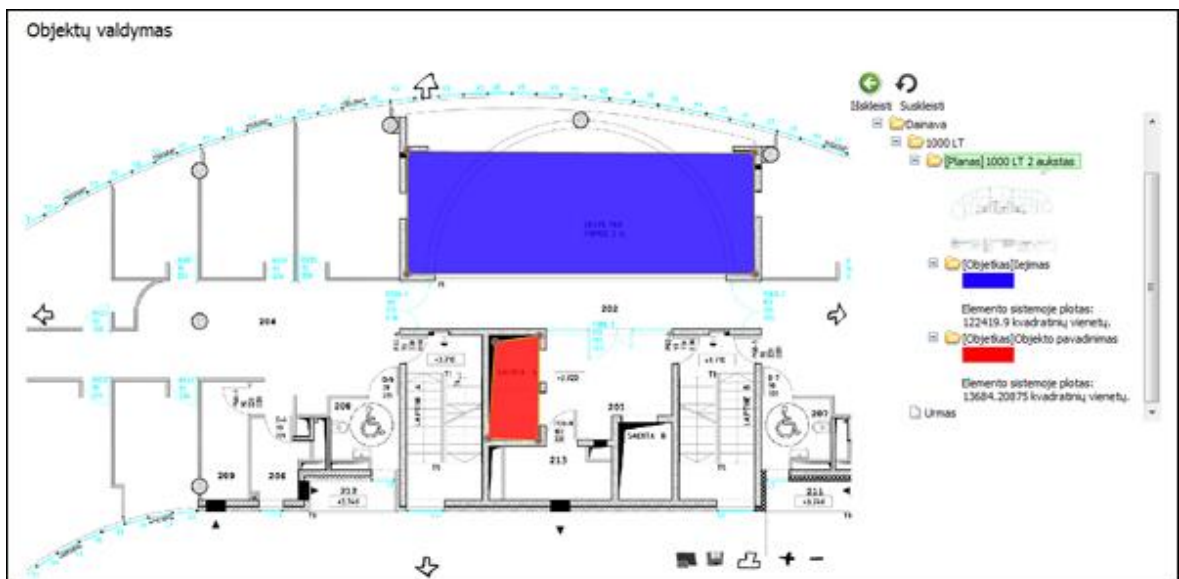
4.1.5 pav. Pasirinkti grupę kurioje priskirtas planas

4.1.5 paveikslėlyje atliekami paskutiniai 3 veiksmai: „pasirinkti grupę, kuriai priskirtas planas“, „pasirinkti planą iš grupės“, „valdyti grafinius objektus“. Norint valdyti objektų būsenas, reikia eiti į būsenų valdymo langą (4.1.6 pav). Pakeitus būsenas ir grįžus į objektų valdymo langą, matomi objektai su kitomis būsenomis (4.1.7 pav.).





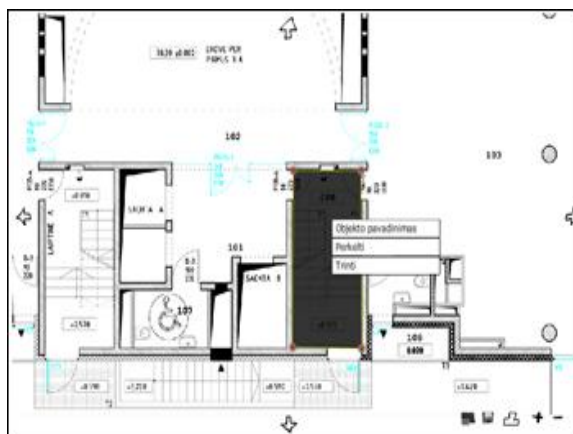
4.1.6 pav. Grafinių objektų būsenų langas



4.1.7 pav. Objektų pakeistos būsenos

Atsidarius grafinių objektų valdymo langą, galima:

- suteikti pasirinktam objektui pavadinimą;
- judinti pasirinktą tašką, t. y. keisti taško poziciją;
- trinti objektą;
- sukurti naują objektą;
- stumdyti planą į keturias puses: aukštyn, žemyn, į kairę ir į dešinę;
- priartinti arba nutolinti plano vaizdą;
- išdidinti plano vaizdą per visą ekraną, taip palengvinant darbą su elementais.



4.1.8 pav. Flash komponento vaizdas

4.1.8 paveikslėlyje pateiktas FLASH komponento vaizdas, kuris atspindi daugumą išvardintų komponento funkcijų.

Visi pakeitimai nevykdomi iškart, jie vykdomi intervalu, kas 30 sekundžių. Prieš uždarant planą arba pasirenkant naują planą, patartina spustelėti „Saugoti“ mygtuką, kuris yra diskelio formos.

Sistema labiau orientuota į grupės ir pogrupio vaidmenį. Kiekviena grupė - tai tarsi vienas geografinis objektas, pavyzdžiui, KTU centriniai rūmai; kita grupė galėtų būti 4 KTU bendrabutis ar kitas objektas. Kiekvienoje iš šių grupių, tarkime, yra įrengtos apsaugos sistemos, taip pat yra laisvos patalpos, kurios yra nuomojamos arba tiesiog turinčios būseną. Tuomet galima toliau pereiti prie sekančio etapo. Susidaromas planas „JPG“, „GIF“, „PNG“, „SWF“ formatu. dydis nėra rodiklis. Geriausia būtų tokio dydžio planas, kad galėtume įskaityti, kas jame parašyta, ir aiškiai matomi objektai, kad suprastume, kas tai yra.

## 4.2 Testavimo modelis bei duomenys, kontrolinis pavyzdys

Testavimo tikslas yra patikrinti, ar realizuotos sistemos veikimas atitinka sistemos reikalavimų specifikacijoje aprašytus funkcinis ir nefunkcinius reikalavimus. Taip pat surasti sistemos architektūros ir realizavimo defektus, kad juos būtų galimam ištaisyti ir užtikrinti, jog galutinis produktas yra kokybiškas.

Testavimui skirtas ribotas laikas: pagal projekto planą jam numatyta 20 dienų.

Testavimo ištekliai apima naudotą programinę ir aparatūrinę įrangą. Toliau pateikiamas testavimo išteklių aprašo pavyzdys.

Testuojama sistema buvo įdiegta kompiuteryje su šia programine įranga:

- Windows 7;
- Internet Explorer 8;
- Adobe FLASH player 10.0.45.2;

- EASYPHP 2.0.0.0;
- MYSQL 5.1.37;
- APACHE 2.2.13;
- PHP 5.2.10.

Atskirai ištestuoti komponentai palaipsniui integruojami į sistemą, kuri testuojama toliau. Tokiu būdu atliekamas stambinantysis (bottom-up) testavimas. Sistema integruojama tokia tvarka:

- Planų valdymas;
- Grupių, pogrupių valdymas;
- Grafinių objektų valdymas;
- Grafinių objektų būsenų keitimas;
- Grafinių objektų priskyrimas prie pasirinktos sistemos.

Toliau pateikiami konkrečių modulių testavimo atvejai ir testavimo rezultatai.

Testuojant funkcinių reikalavimų realizavimą, kiekvienam panaudojimo atvejui reikia sukurti testavimo atvejus – tiek, kiek yra sėkmingų ir nesėkmingų variantų. Visi testavimo panaudojimo atvejai pateikti lentelėse 4.2.1 – 4.2.5.

**4.2.1 lentelė** *Panaudojimo atvejo „Planų valdymas“ testavimas*

<b>Panaudojimo atvejis</b>		Planų valdymas		
<b>Testo paskirtis</b>		Patikrinti įvestų duomenų redagavimą, naujų duomenų pateikimą, esamų duomenų trynimą		
<b>Testiniai duomenys</b>		Pavadinimas = „Naujas Planas“; „Planas“; „Plano pavadinimas“; plano paveikslėlis = „test.jpg“; „test2.jpg“		
<b>Nr.</b>	<b>Veiksmas</b>	<b>Įvesti duomenys</b>	<b>Laukiami rezultatai</b>	<b>Gauti rezultatai</b>
1.	Paspausti „Sukurti“.		Sukurtas naujas elementas sąrašė pavadinimu „Naujas Planas“	Sąrašė vaizduojamas naujas planas pavadinimu „Naujas Planas“
2	Redaguoti „Naujas Planas“	Pavadinimas = „Planas“	Sąrašė pakeistas pavadinimas į „Planas“	Vaizduojamas pakeistas pavadinimas „Planas“

3	Redaguoti „Planas“	Pavadinimas = „Planas“ Paveikslėlis = „test.jpg“	Sąrašė turi vaizduoti paveikslėlį	Sąrašė vaizduojamas paveikslėlis
4.	Redaguoti „Planas“	Pavadinimas = „Plano pavadinimas“ Paveikslėlis = „test2.jpg“	Sąrašė turi vaizduoti paveikslėlį ir turi būti pakeistas pavadinimas „Plano pavadinimas“	Sąrašė vaizduojamas paveikslėlis ir pakeistas pavadinimas „Plano pavadinimas“
5	Ištrinti „Plano pavadinimas“		Turi būti ištrintas elementas „Plano pavadinimas“	Ištrintas „Plano pavadinimas“

4.2.2 lentelė Panaudojimo atvejo „Grupių, pogrupių valdymas“ testavimas

<b>Panaudojimo atvejis</b>		Grupių, pogrupių valdymas		
<b>Testo paskirtis</b>		Patikrinti įvestų duomenų redagavimą, naujų duomenų pateikimą, esamų duomenų trynimą, planų priskyrimą prie grupių		
<b>Testiniai duomenys</b>		Pavadinimas = „grupė“; „grupė su pogrupiu“; „pogrupis“; „planas“		
<b>Nr.</b>	<b>Veiksmas</b>	<b>Įvesti duomenys</b>	<b>Laukiami rezultatai</b>	<b>Gauti rezultatai</b>
1.	Paspausti „Sukurti naują grupę“		Sukurtas naujas elementas, sąrašė pavadinimu „Nauja grupė“ ir atnaujinti sąrašai	Sąrašė vaizduojama nauja grupė pavadinimu „Nauja grupė“ ir atnaujinti sąrašai
2	Redaguoti „Nauja grupė“	Pavadinimas = „grupė“	Sąrašė pakeistas pavadinimas į „grupė“ ir atnaujinti sąrašai	Vaizduojamas pakeistas pavadinimas „grupė“ ir atnaujintas sąrašas
3	Redaguoti „grupė“	Pavadinimas = „grupė su	Sąrašė pakeistas pavadinimas į „grupė su	Vaizduojamas pakeistas pavadinimas „grupė su

		pogrupiu“	pogrupiu“ ir atnaujinti sąrašai	pogrupiu“ ir atnaujintas sąrašas
4.	Grupės sąrašė paspausti ant elemento „grupė su pogrupiu“, vėliau sukurti pogrupį		Sukurtas naujas elementas sąrašė pavadinimu „Nauja grupė“ po elementu „grupė su pogrupiu“ ir atnaujinti sąrašai	Sąrašė vaizduojama nauja grupė pavadinimu „Nauja grupė“ po elementu „grupė su pogrupiu“ ir atnaujinti sąrašai.
5	Redaguoti „Nauja grupė“	Pavadinimas = „pogrupis“	Sąrašė turi būti pakeistas pavadinimas į „pogrupis“ ir atnaujinti sąrašai.	Pakeistas grupės pavadinimas į „pogrupis“ ir atnaujinti sąrašai.
6	Iš sukurtų planų pasirinktas „planas“ ir kito sąrašo „pogrupis“. Paspaustas „Sudaryti sąsają.“		Sujungtų planų ir grupių sąrašė turi būti naujas sujungimas su grupe „pogrupis“ ir planu „planas“.	Sąrašė vaizduojamas naujas sujungimas su grupe „pogrupis“ ir planu „planas“
7	Panaikinti sukurtą sujungimą tarp „planas“ ir „pogrupis“		Sujungimų sąrašė nerodyti „planas“ sąjungos su „pogrupis“.	Nevaizduoja sąjungos tarp „planas“ ir „pogrupis“
8	Atliekame testavimo 6 žingsnį. Tada iš grupių sąrašo pasirenkame „pogrupis“ ir ištriname pogrupį		Grupių sąrašė panaikinta grupė „pogrupis“ ir sujungimų sąrašė panaikinta sąjunga tarp „pogrupis“ ir „planas“	Grupių sąrašė panaikinta grupė „pogrupis“ ir sąjungų sąrašė panaikinta sąjunga tarp „pogrupis“ ir „planas“

4.2.3 lentelė Panaudojimo atvejo „Grafinių objektų valdymas“ testavimas

<b>Panaudojimo atvejis</b>		Grafinių objektų valdymas		
<b>Testo paskirtis</b>		Patikrinti įvestų duomenų redagavimą, naujų duomenų pateikimą, esamų duomenų trynimą		
<b>Testiniai duomenys</b>		Pavadinimas = „lova“; „sofa“, „ Kampų skaičius = „5“		
<b>Nr.</b>	<b>Veiksmas</b>	<b>Įvesti duomenys</b>	<b>Laukiami rezultatai</b>	<b>Gauti rezultatai</b>
1	Paspausta ant  “Sąrašo planas“		Turi užkrauti planą flash‘e	Užkrautas planas flash‘e
2	Paspausta „Sukurti grafinį objektą“	Kampų skaičius = „5“	Turi iššokti langas, kuriame įvedamas kampų skaičius; kiek įvesta, tiek poligonas privalo turėti kampų	Iššoko langas su įvesti kampų skaičiumi, Toliau spaudinėjant ant flash sudarytas poligonas su 5 kampais
3	Paspaustas ir tempiamas konkretus kampas		Turi kisti pasirinkto taško pozicija ir numatytoju laiko intervalu išsaugoti į duomenų bazę	Pakito taško pozicija ir išsaugojo taškų pozicijas į duomenų bazę
4	Paspausta „Redaguoti objekto pavadinimą“	Pavadinimas = „lova“	Turi pasikeisti objekto pavadinimas į „lova“ ir numatytoju laiko intervalu išsisaugoti į duomenų bazę	Pasikeitė pavadinimas į „lova“ ir išsaugojo pavadinimą duomenų bazėje numatytoju laiko intervalu
5	Paspausta „Redaguoti objekto pavadinimą“	Pavadinimas = „sofa“	Turi pasikeisti objekto pavadinimas į „sofa“ ir laiko numatytoju intervalu išsisaugoti į duomenų bazę	Pasikeitė pavadinimas į „sofa“ ir išsaugojo pavadinimą duomenų bazėje numatytoju laiko intervalu
6	Paspausta „Redaguoti objekto pavadinimą“	Pavadinimas = „“	Turi pasikeisti objekto pavadinimas į „“ ir laiko numatytoju laiko intervalu išsisaugoti į	Pasikeitė pavadinimas į „“ ir išsaugojo pavadinimą duomenų bazėje numatytoju

			duomenų bazę	laiko intervalu
7	Paspausta „Perkelti“		Judinant pelę flash'e turi keistis objekto pozicija pagal pelės poziciją; paspaudus dar kartą ant flash objekto pozicija turi užsifikuoti. Pakeitimai turi išsisaugoti duomenų bazėje laiko numatytoju laiko intervalu.	Pajudinus pelę, kito objekto pozicija, paspaudus ant flash'o dar kartą. užsifikuoti objekto pozicija. Pakeitimai išsisaugoją duomenų bazėje numatytoju laiko intervalu
8	Paspausta „Trinti“		Turi pasišalinti grafinis objektas iš plano. Pakeitimai turi išsisaugoti duomenų bazėje numatytoju laiko intervalu.	Panaikintas grafinis objektas iš plano. Pakeitimai išsisaugoją duomenų bazėje numatytoju laiko intervalu

4.2.4 lentelė Panaudojimo atvejo „Grafinių objektų būsenų keitimas“ testavimas

<b>Panaudojimo atvejis</b>		Grafinių objektų būsenų keitimas		
<b>Testo paskirtis</b>		Patikrinti įvestų duomenų redagavimą, naujų duomenų pateikimą, esamų duomenų trynimą		
<b>Testiniai duomenys</b>		Pavadinimas = „raudona“; „juoda“; „balta“ spalva = „ff0000“; „000000“; „ffffff“		
<b>Nr.</b>	<b>Veiksmas</b>	<b>Įvesti duomenys</b>	<b>Laukiami rezultatai</b>	<b>Gauti rezultatai</b>
1	Sukurti naują būseną		Turi atsirasti nauja būseną pavadinimu „Nauja būseną“	Atsirado nauja būseną pavadinimu „Nauja būseną“
2	Paspausti ant „Nauja būseną“ ir redaguoti	Pavadinimas = „raudona“ Spalva = „ff0000“	Turi pasikeisti būsenos pavadinimas ir būsenos spalva pagal įvestus duomenis	Pasikeitė būsenos pavadinimas į „raudona“ ir būsenos spalva į „ff0000“
3	Pasirinkti		Pasirinkto grafinio	Duomenų bazėje

	grafikos objektą iš objektų sąrašo, pasirinkti būseną „raudona“ ir paspausti „Išsaugoti“		objekto būseną turi būti pakeista į „raudona“ duomenų bazėje	pakeista pasirinkto grafinio objekto būseną į „raudona“.
4	Pasirinkti būseną „raudona“ ir paspausti „Ištrinti“		Turi būti pašalinta „raudona“ būseną iš duomenų bazės ir iš sąrašo	Pašalinta būseną iš duomenų bazės ir iš sąrašo
5	Sukurti naują būseną		Turi atsirasti nauja būseną pavadinimu „Nauja būseną“	Atsirado nauja būseną pavadinimu „Nauja būseną“
6	Paspausti ant „Nauja būseną“ ir redaguoti	Pavadinimas = „juoda“ Spalva = „000000“	Turi pasikeisti būsenos pavadinimas ir būsenos spalva pagal įvestus duomenis	Pasikeitė būsenos pavadinimas į „juoda“ ir būsenos spalva į „000000“
7	Paspausti ant „juoda“ ir redaguoti	Spalva = „ffffff“	Turi pasikeisti būsenos spalva į „ffffff“	Pasikeitė būsenos spalva į „ffffff“
8	Paspausti ant „juoda“ ir redaguoti	Pavadinimas = „balta“	Turi pasikeisti būsenos pavadinimas į „balta“	Pasikeitė būsenos pavadinimas į „balta“
9	Pasirinkti grafikos objektą iš objektų sąrašo ir pasirinkti būseną „balta“ ir paspausti „Išsaugoti“		Pasirinkto grafinio objekto būseną turi būti pakeista į „balta“ duomenų bazėje	Duomenų bazėje pakeista pasirinkto grafinio objekto būseną į „balta“.



### **4.3 Realizacijos apibendrinimas**

Pateikti testavimo rezultatai parodo sistemos teisingumą – tai ko buvo tikėtasi, tas ir gauta. Tai parodo visi testavimo rezultatai. Realizacijos dalis pilnai ištestuota, tolesniu eksperimentu parodoma kaip įdiegti sistemą. Taip pat dar vienas panaudos atvejis nebuvo ištestuotas – tai yra grafinių objektų priskyrimas prie nuomos sistemos. Kadangi ši realizacija yra komponentinė, eksperimente aprašyta, kaip reiktų įdiegti į pasirinktą sistemą šį komponentą.

Pasirinktas sprendimas atitinka visus sistemos reikalavimus, sistemą galima taikyti ne tik nekilnojamojo turto valdyje, bet ir kitose sistemose, kuriose valdomos objektų būsenos.

## 5 Eksperimentinis sistemos tyrimas

Tyrimo tikslas – sulyginami buvusią sistemą ir sukurtą komponentą: kiek kiekvienu atveju bus sunaudota laiko, atliekant veiksmus. Sekantis eksperimentas parodo, kaip įdiegti komponentą kitoje dalykinėje srityje, keičiant konfigūracinio failo parametrus.

### 5.1 Savybių analizė

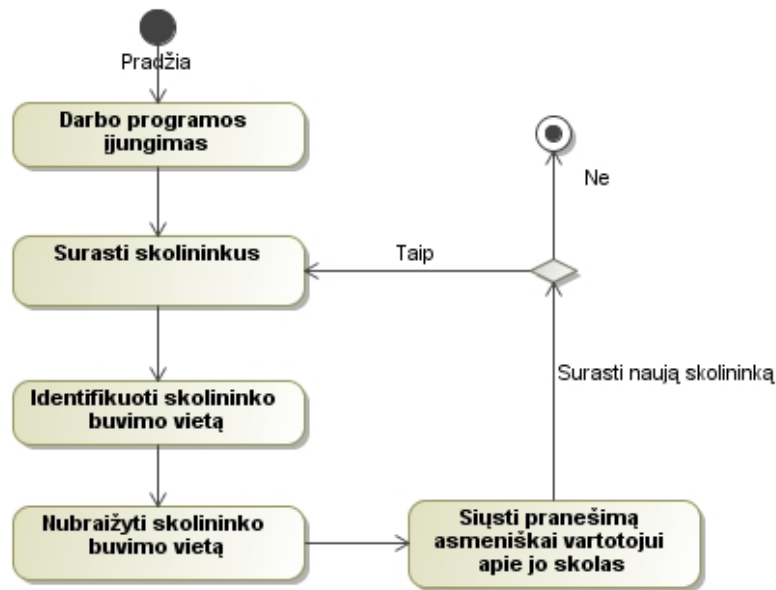
Norit atlikti kasdienes darbus, tokius kaip informuoti skolininkus apie jų skolas arba laisvus nuomos plotus, privalu atlikti tam tikrą veiksmų seką (žr. 5.1.1 pav.), kuri pateikta žemiau:

- kiekvieną kartą reikia įsijungti darbo programą;
- surasti skolininkus;
- identifikuoti skolininko buvimo vietą;
- nubraižyti jo buvimo vietą (naudojantis esama sistema);
- siųsti pranešimą asmeniškai vartotojui apie jo skolas.

Taigi atliekame laiko skaičiavimą: kiek užtrunka atlikti šią veiksmų seką įvairiems vadybininkams (5.1.1 lentelė). Visi laikai surašyti sekundėmis.

5.1.1 lentelė Buvusios sistemos laiko sąnaudos

Atliekamas darbas	Darbo programos įjungimas	Surasti skolininkus	Skolininko identifikavimas	Nubraižyti buvimo vietą vieno skolininko vietą	Siųsti pranešimą asmeniškai vienam vartotojui	Iš viso Sugaištas laikas
Ekspertas						
Ekspertas 1	10	60	60	120	90	<b>340</b>
Ekspertas 2	15	120	60	180	90	<b>465</b>
Ekspertas 3	12	90	70	160	80	<b>412</b>
Ekspertas 4	20	130	90	240	120	<b>600</b>
<b>Vidurkis</b>	<b>14,25</b>	<b>100</b>	<b>70</b>	<b>175</b>	<b>95</b>	<b>454,25</b>



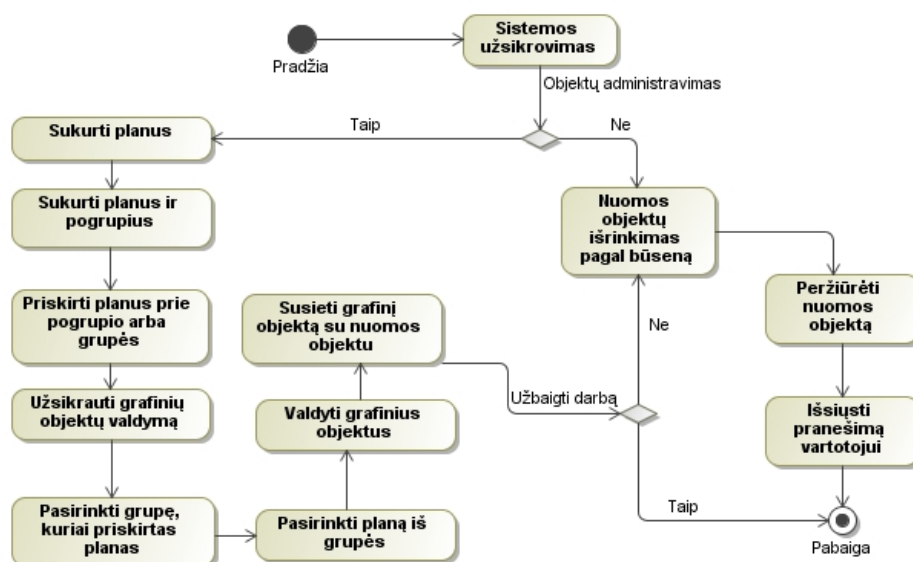
5.1.1 pav. Buvusios sistemos veiksmų seka

Iš lentelės duomenų matyti, kad išsiųsti pranešimą vartotojams vidutiniškai užtrunka apie 454 sekundžių senoje sistemoje, t. y. apie aštuonias minutes vienam vartotojui.

Išanalizuokime, kiek laiko užtruktų tai padaryti su jau realizuota sistema. Veiksmų seka, atliekant išsiuntimą pranešimą vartotojams, yra tokia:

- Sistemos užsikrovimas;
- Pirminis planų susiejimas su nuomos sistema, šis veiksmas atliekamas vieną kartą. Kai priskyrimas atliekamas daugiau kartų, šio veiksmo atlikinėti nereikia;
- Pagal būseną galima išsirinkti nuomos objektus ir vaizduoti juos viename sąrašė;
- Nuorodos paspaudimu matoma objekto vieta;
- Pagal paruoštą šabloną siunčiamas pranešimas vartotojams.

Vienas iš minėtų punktų nebus skaičiuojamas į bendrą laiko skaičiavimą, tačiau apskaičiuosime, kiek laiko bus sugaišta sudarant visus planus ir kiek laiko užtruks sudaryti grafinių objektų sąsają su nuomos objektais (žr. 5.1.2 pav.).



5.1.2 pav. Esamos sistemos veiksmų seka

Kaip buvo minėta sistemos realizacijos skyriuje, „sistemos veikimo aprašymas“, pagal tuos punktus atlikta analizė, kiek laiko užtrunka sukurti vieną grafinį objektą su vienu planu. Tai atliks keturi ekspertai, bus fiksuojamas laikas, kaip greitai jie atlieka visus veiksmus (pavaizduota 5.1.2 lentelėje). Ekspertai testuos, kiek laiko jie sugaišta, kurdami vieną grafinį objektą, t. y. vykdydami visus žingsnius.

Sekantys veiksmai atliekami dažniau, nei grafinių objektų kūrimas ir valdymas, vartotojų identifikavimas, filtravimas pagal būsenas (5.1.3 lentelė).

5.1.2 lentelė Grafinių objektų paruošimo ir susiejimo su nuomos objektu laiko sąnaudos

Atliekamas darbas									Viso
	Susikurti planus	Sukurti grupes ir pogrupius	Priskirti planus prie pogrupio arba grupės	Užsikrauti grafinių objektų valdymą	Pasirinkti grupę, kuriai priskirtas planas	Pasirinkti planą iš grupės	Valdyti grafinius objektus	Susieti grafinį objektą su nuomos objektu	
Ekspertas									
Ekspertas 1	10	15	5	5	5	5	60	10	<b>115</b>
Ekspertas 2	15	20	10	10	10	5	70	10	<b>150</b>
Ekspertas 3	12	15	5	10	5	5	90	10	<b>152</b>
Ekspertas 4	20	10	5	10	5	4	55	10	<b>119</b>
<b>Vidurkis</b>	<b>14,25</b>	<b>15</b>	<b>6,25</b>	<b>8,75</b>	<b>6,25</b>	<b>4,75</b>	<b>68,75</b>	<b>10</b>	<b>134</b>

5.1.3 lentelė Grafinių objektų paruošimo ir susiejimo su nuomos objektu laiko sąnaudos

Atliekamas darbas	Sistemos užsikrovimas	Nuomos objektų išrinkimas pagal būseną	Peržiūrėti nuomos objektą	Išsiųsti pranešimą vartotojui	Viso
Ekspertas					
Ekspertas 1	10	20	15	20	<b>65</b>
Ekspertas 2	15	25	10	15	<b>65</b>
Ekspertas 3	20	15	15	20	<b>70</b>
Ekspertas 4	15	20	10	20	<b>65</b>
<b>Vidurkis</b>	<b>15</b>	<b>20</b>	<b>12,5</b>	<b>18,75</b>	<b>66,25</b>

Taigi bendrai susumavus rezultatus laiko sąnaudos buvo  $134 + 66,25 = 200,25$ , t. y. apie keturios minutės, iš kurių apie dvi minutės buvo reikalingos sukurti planą, grupes ir grafinius objektus.

Sulyginus šiuos senos ir naujos sistemų rezultatus, skirtumas yra  $454,25 - 200,25 = 254$ , t. y. senosios sistemos atveju papildomai sunaudojamos keturios minutės. Tokiu būdu paspartinamas darbo procesas, pagerinamas vadybininkų darbas, nes jie dabar gali išsiųsti daugiau laiškų.

Su senąja sistema vienas vadybininkas išsiunčia  $\frac{60 \cdot 60 \cdot 8}{454,25} \approx 63$  tokius laiškus per dieną. O su realizuota sistema -  $\frac{60 \cdot 60 \cdot 8}{200,25} \approx 143$  laiškus per dieną. Tai du kartus daugiau, nei su senąja sistema, be to, vieną kartą priskyrus nuomos sistemą prie grafinio objekto, nereikia tai daryti antrą kartą, ir tą laiką gali išnaudoti siunčiant laiškus vartotojams. Tokiu atveju vadybininkas galėtų išsiųsti  $\frac{60 \cdot 60 \cdot 8}{66,25} \approx 434$  laiškus, o tai yra 6,8 karto daugiau, nei su senąja sistema.

## 5.2 Komponento diegimas mašinų stovėjimo aikštelėje

Norit komponentą pritaikyti kitoje srityje, reikia pakeisti pagrindinius konfigūracinio failo duomenis pagal serverio ir duomenų bazės parametrus:

- Duomenų bazės adresas
- ```
define ( 'RK_COMPONENT_DATABASE_HOST', 'localhost' );
```
- duomenų bazės vartotojo vardas
- ```
define ( 'RK_COMPONENT_DATABASE_USER', 'root' );
```

- duomenų bazės vartotojo slaptažodis

```
define ( 'RK_COMPONENT_DATABASE_PASSWORD', '' );
```

- duomenų bazės vardas

```
define ( 'RK_COMPONENT_DATABASE_NAME', 'automobiliu_stovejimo_aiskstele' );
```

Norit pakeisti duomenų bazės lentelių pavadinimus, reikia keisti:

- Duomenų bazės priešdėlį

```
define ( 'RK_COMPONENT_DATABASE_TABLES_PREFIX', 'rk_' );
```

- būsena lentelės pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_BUSENA', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'busena' );
```

- grupių lentelės pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_GRUPE', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'grupe' );
```

- planų lentelės pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_PLANAI', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'planai' );
```

- plano pogrupio elementų lentelės pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_PLANO_POGRUPIO_ELEMENTAI', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'plano_pogrupio_elementai' );
```

- plano pogrupio lentelės pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_PLANO_POGRUPIS', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'plano_pogrupis' );
```

- nuomos lentelės pavadinimą ir lentelės stulpelių pavadinimą

```
define ( 'RK_COMPONENT_DATABASE_TABLE_RENT', RK_COMPONENT_DATABASE_TABLES_PREFIX. 'nuoma' );
```

```
define ( 'RK_COMPONENT_DATABASE_TABLE_RENT_COLUMN_ID', "ID" );
```

```
define ( 'RK_COMPONENT_DATABASE_TABLE_RENT_COLUMN_NAME', "Pavadinimas" );
```

```
define ( 'RK_COMPONENT_DATABASE_TABLE_RENT_COLUMN_OBJECT_ELEMENT_ID', "ObjectElementID" );
```

Jeigu yra būtina arba naudojamos specifinės adreso eilutes, reikia keisti šiuos parametrus:

- Adresui formuoti naudojami elementai:

- „?“ – naudojamas iškart po failo pabaigos „php?“ arba „html?“, arba „htm“, jei adresas yra formuojamas kitaip, tada reikia pakeisti šį elementą

```
define ( 'RK_COMPONENT_URL_FIRST_JOIN_ELEMENT', '?' );
```

- Sekantys parametro elementai sujungiami “&” simboliu, pvz.: „...php?id=a&limit=...“

```
define ( 'RK_COMPONENT_URL_SECOND_JOINS_ELEMENT', '&' );
```

- Parametro reikšmės „=“ nurodantis elementas pvz.: „...php?id=a...“

```
define ( 'RK_COMPONENT_URL_VALUE_EQUAL', '=' );
```

- jei sistemoje keičiasi pradžios, adresas tokiu atveju keičiame

```
define ( 'RK_COMPONENT_URL_START_BASELINK', 'index.php' );
```

- identifikuojame pagrindinius parametrus „id=...“, ir veiksmus, tokius kaip „saugoti“, „trinti“ aprašomus per „action=...“. Šiuos parametrus galime pakeisti

```
define ( 'RK_COMPONENT_URL_COMPONENT_NAME', 'rk_id' );
```

```
define ( 'RK_COMPONENT_URL_ACTION_NAME', 'rk_action' );
```

- AJAX užklausų vykdymo parametrai

```
define ( 'RK_COMPONENT_URL_AJAX_NAME', 'ajax' );
```

```
define ( 'RK_COMPONENT_URL_AJAX_NAME_VALUE', '1' );
```

- Pagrindinių veiksmų adresai keičiami šiais parametrais

```
define ( 'RK_COMPONENT_NAME_PLANU_VALDYMAS', 'planai' );
```

```
define ( 'RK_COMPONENT_NAME_GRUPIU_POGRUPIU_VALDYMAS', 'grupes' );
```

```
define ( 'RK_COMPONENT_NAME_GRAFINIU_OBJEKTU_VALDYMAS', 'objektai' );
```

```
define ( 'RK_COMPONENT_NAME_GRAFINIU_OBJEKTU_PRISKYRIMAS_PRIE_NUOMOS_SISTEMOS', 'priskyrimas' );
```

```
);
```

Pakeitus šiuos parametrus, reikia įdiegti SQL programinį kodą į duomenų bazę. SQL programinis kodas yra šakniniame sistemos kataloge. Jis pateiktas priede Nr.1.

Sistema jau paruošta darbui.

Sukuriami testiniai duomenys automobilio nuomos sistemai. Atliekami visi tie patys veiksmai, kaip nurodyta sistemos realizacijos veikimo apraše, sukuriami planai, grupės, priskiriami planai prie grupių, sukuriami grafiniai objektai, sukuriama būsenos, pakeičiamos sukurtų grafinių objektų būsenos ir sudaroma grafinių objektų sąsaja su nuomos sistema (visų veiksmų seka pavaizduota 5.2.1 – 5.2.5. pav.).

- Planų valdymas
- Grupių, pogrupių valdymas
- Grafinių objektų valdymas
- Grafinių objektų priskyrimas prie nuomos sistemos
- Grafinių objektų būsenų keitimas

ID	Pavadinimas	Kaina	Būsena	Peržiūrėti	Administruoti
1	C1	20.00	Laisva	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
2	C2	20.00	Laisva	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
3	C3	20.00	Laisva	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
4	C4	0.00	Laisva	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
5	C5	20.00	Rezervuota	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
6	C6	0.00	Užimta	<a href="#">Grafinis Objektas</a>	<a href="#">Grafinis Objektas</a>
7	C7	20.00			
8	C8	20.00			
9	C9	20.00			
10	C10	20.00			

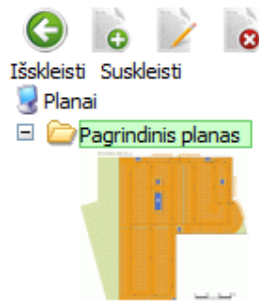
ID  Kaina  Būsena

Showing 1 to 10 of 10 entries

First Previous 1 Next Last

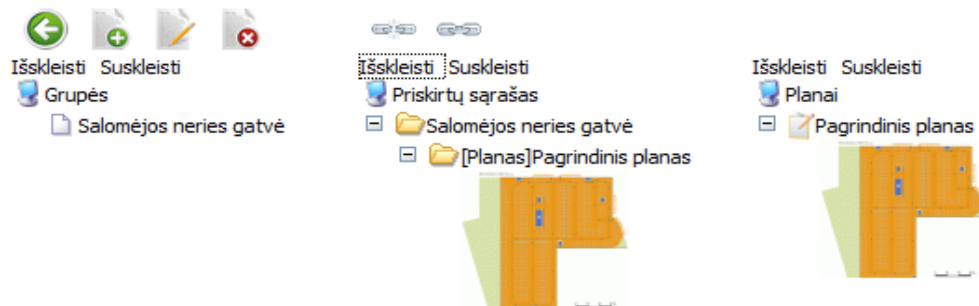
5.2.1 pav. Mašinų stovėjimo aikštelės administravimo pradinis langas

## Planų valdymas



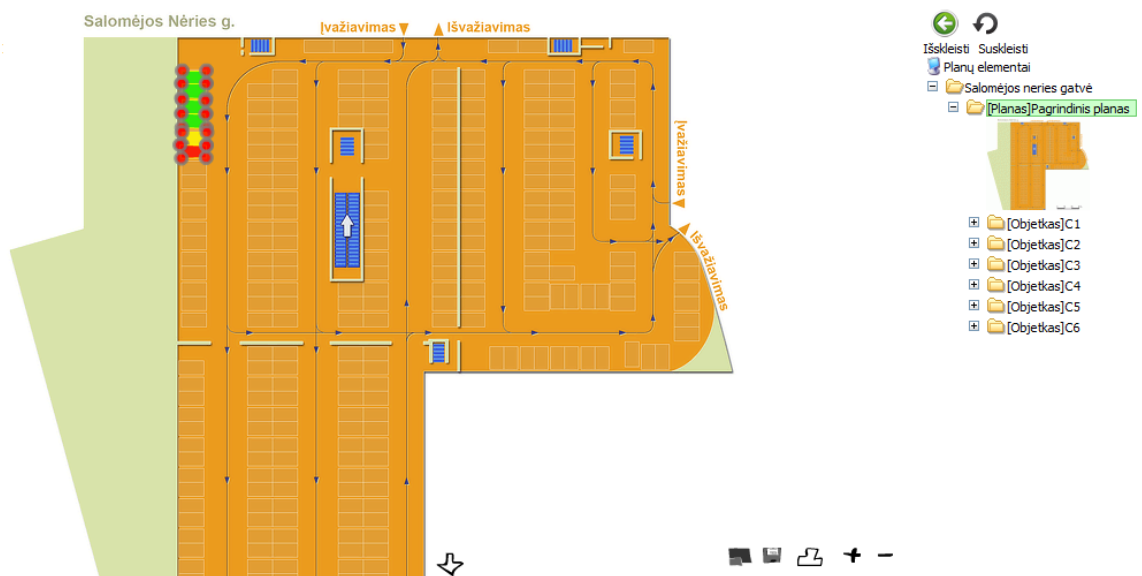
5.2.2 pav. Įkeltų planų langas

## Grupių, pogrupių valdymas



5.2.3 pav. Sukurtų grupių bei sudarytų planų su grupe vaizdas

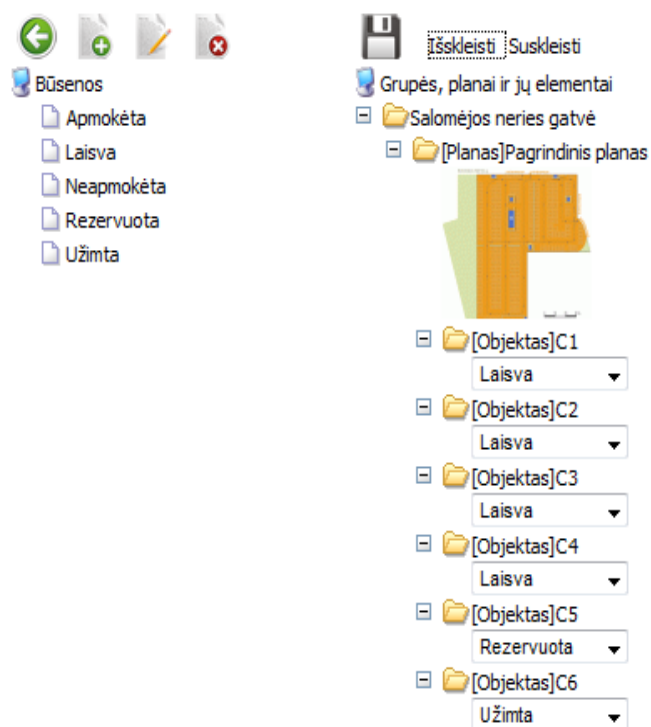
## Objektų valdymas



5.2.4 pav. Grafinių objektų valdymo langas

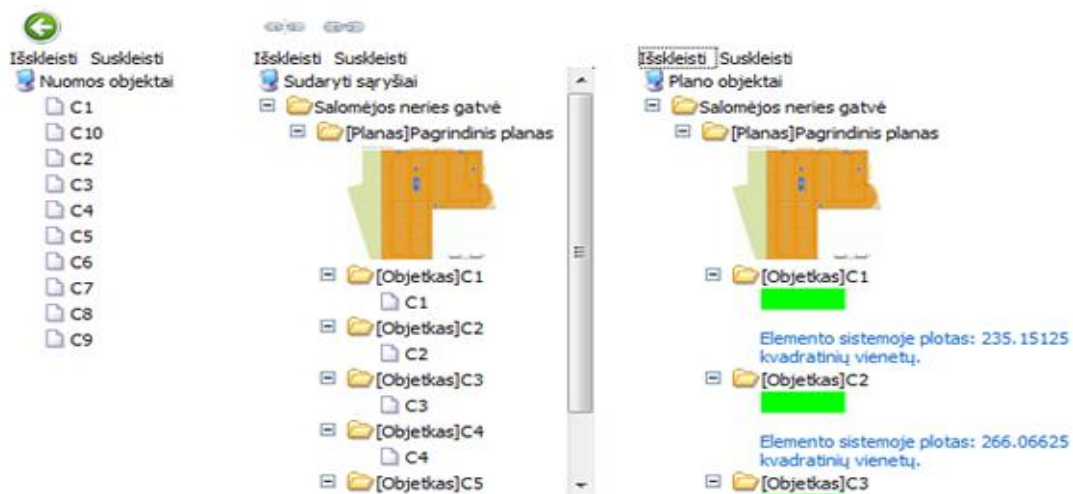


## Grafinių objektų būsenų keitimas



5.2.5 pav. Mašinų stovėjimo aikštelės būsenų keitimo langas

## Mašinų stovėjimo aikštelėje objektų priskyrimas



5.2.6 pav. Mašinų stovėjimo aikštelės grafinių objektų priskyrimo prie stovėjimo vietų langas

## 6 Išvados

1. Išanalizavus programavimo kalbas, pasirinkta Adobe Flash, nes ji užima mažiau atminties, negu JAVA applet. Adobe Flex naudojamas tokių duomenų vaizdavimui, kaip lentelės forma pateikiami duomenys.
2. Trijų sistemų analizė parodė, kad jų funkcionalumas yra visiškai pritaikytas konkrečiai dalykinei sričiai, šių sistemų ar jų dalies adaptavimas, sprendžiant nekilnojamojo turto administravimo problemas, yra neįmanomas.
3. Darbe pateiktas koncepcinis komponento projektas. Komponentas išsprendžia grafinių objektų atvaizdavimo internetinėse sistemose problemą, tuo pat metu juos susiejant su informacija saugoma duomenų bazėje.
4. Koncepcinis komponentas realizuotas kaip atskiras modulis Nekilnojamojo turto valdymo ir administravimo sistemoje.
5. Nekilnojamojo turto sistemos realizacija rodo, kad pasirinktas architektūrinis modelis užtikrina sistemos patikimumą; tai patvirtina sistemos funkcinio testavimo rezultatai.
6. Eksperimentiniu būdu atlikus senosios ir naujai sukurtos Nekilnojamojo turto sistemų lyginamąjį tyrimą, nustatyta, kad vadybininko darbo sparta gali padidėti iki 6,8 karto.
7. Sukurtą komponentą galima pritaikyti ir kitoms dalykinėms sritims; tai patvirtina atliktas eksperimentas su mašinų stovėjimo aikštele.

## 7 Literatūra

1. Awarepoint Corporation. Real-time Location Systems (RTLS) [žiūrėta 2008-12-05]. Prieiga per internetą: <<http://www.awarepoint.com/rtls.shtml>>.
2. Wikipedia, the free encyclopedia. Adobe Flash [žiūrėta 2008-12-18]. Prieiga per internetą: <[http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash)>.
3. Charles. Flex vs. AJAX | work. [žiūrėta 2008 12 01]. Prieiga per internetą: <<http://work.whoischarles.com/content/flex-vs-ajax>>.
4. Wikipedia, the free encyclopedia. Java applet [žiūrėta 2008-12-05]. Prieiga per internetą: <[http://en.wikipedia.org/wiki/Java\\_applet](http://en.wikipedia.org/wiki/Java_applet)>
5. AeroScout, Inc. Asset Management. [žiūrėta 2010-02-05]. Prieiga per internetą: <<http://www.aeroscout.com/content/asset-management>>.
6. Carlos Real. *Flash, Flex & Ajax - How do they compare?* [žiūrėta 2009 05 10]. Prieiga per internetą: <<http://www.insightdesignsblog.com/?p=350>>.
7. Pakalnickas, E.; Gudas, S. Sąsajomis grįstas IS projektavimas, iš *Informacinės technologijos konferencijos, įvykusios 2006 m. sausio 25 d.* Kaunas, 189-193.
8. Cisco Systems. *Cisco Wireless Location Appliance - Products & Services - Cisco Systems.* [žiūrėta 2008-11-29]. Prieiga per internetą: <<http://www.cisco.com/en/US/products/ps6386/index.html>>.
9. Štuikys, V.; ir Damaševičius, R. 2008. *Modelių ir programų abstrakčiosios transformacijos.* Kaunas. 38p.
10. Wikipedia, the free encyclopedia. Duomenų bazė, PNG, JPG, GIF, WYSUWYG, SVG, Adobe Flash, W3C, Lankstusis programavimas [žiūrėta 2010 03 22]. Prieiga per internetą: <<http://lt.wikipedia.org/wiki/>>.
11. Teiniker, E.; Schmoelzer, G.; Faschingbauer, J.; Kreiner, C.; Weiss, R. 2005. Hybrid Component-Based System Development Process, in *Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference.* 152-159p. ISBN: 0-7695-2431-1.
12. Pfadenhauer, K.; Dustdar, S.; Kittl, B. 2005. Challenges and solutions for model driven web service composition, in *Proceedings of the 14th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprise.* IEEE Computer Society. 126–134p. ISBN ~ ISSN:1524-4547, 0-7695-2362-5.

## 8 Terminų ir santrumpų žodynis

DB – Duomenų bazė yra organizuotas (susistemintas, metodiškai sutvarkytas) duomenų rinkinys, kuriuo galima individualiai naudotis elektroniniu ar kitu būdu. Nors terminas kilo kompiuterių pramonėje, dabartinės vartojimo prasmės ribos praplėstos ir EU Database Directive (kurianti duomenų bazių intelektinė nuosavybės teises) apima ir neelektronines rūšis [10].

PNG (angl. Portable Network Graphics) – bitų masyvo formatas, kuris suglaudinas „be nuostolių“. Sukurtas kaip GIF formato pakaitalas – formatas nereikalaujantis patento licencijos [10].

JPG (arba JPEG, angl. Joint Photographic Experts Group) – fotografinių vaizdų išsaugojimo formatas (ir jų suspaudimo algoritmas). JPG labai efektyviai suspaudžia fotografinius vaizdus. Nors dalis informacijos esant didesniai suspaudimo laipsniui prarandama, formato ypatybės tokios, jog šie praradimai yra labai sunkiai pastebimi. Prarasta informacija daug labiau matoma įvairiose schemose bei diagramose, kur daug plonų linijų bei vienspalvių plotų. Vaizdo kokybę galima žymiai padidinti parenkant mažesni suspaudimo laipsni, bet tuomet didėja failo apimtis. Todėl schemoms bei diagramoms paprastai naudojami kiti formatai (GIF ir PNG). Nuotraukoms tinka ir PNG, tačiau šiuo formatu sukuriami nuotraukų failai būna gerokai didesni [10].

GIF (Graphics Interchange Format) – grafinių vaizdų saugojimo formatas, kurį 1987 m. sukūrė interneto paslaugų tiekėja „CompuServe“ kompanija. 1989m. buvo paskelbta patikslinta specifikacija (vadinamoji GIF versija 89a). Standartinis GIF saugo tik iki 256 spalvų paletes, nors yra galimybė įsiminti ir 24 bitų (truecolor) vaizdus, tačiau tai „purvinas reikalas“ (ir paminėtinas tik aprašymo pilnumui). Leidžiama nurodyti „vaiskiā“ (angl. transparent) spalvą [10].

WYSIWYG – akronimas pasakymui *What You See Is What You Get* (angl. Tai, ką matote, atitinka tai, ką gausite), kuris naudojamas kalbant apie kompiuterines programas, norint apibūdinti kokią nors programinę įrangą, kurios turinio išvaizda rengimo metu yra labai panaši į tos programinės įrangos pateikiamo galutinio produkto išvaizdą. Dažniausiai vartojama teksto rengyklės atveju WYSIWYG sąvoka reiškia, kad redaguojamo teksto išvaizda ekrane bus identiška ant popieriaus atspausdinto teksto išvaizdai. Neretai ir kalbant apie kitas programas, kaip, pavyzdžiui, interneto puslapių rengyklės, jos apibūdinamos kaip WYSIWYG programinė įranga [10].

SVG (Scalable Vector Graphics) – dvimačių vektorinių vaizdų saugojimo formatas XML žymėjimo kalbos pagrindu. Šiuo formatu galima aprašyti tiek statinius, tiek ir animuotus vaizdus. SVG standartas yra atviras, jį sukūrė „W3C“ konsorciumas [10].

Adobe Flash (anksčiau vadintas Shockwave Flash ir Macromedia Flash) – daugialypės programinės įrangos komplektas, sukurtas „Macromedia“ kompanijos, o šiuo metu jis vystomas „Adobe“ kompanijos („Adobe“ nupirko „Macromedia“ kompaniją). Nuo jo pradžios 1996 m., Flash tapo populiariu būdu pridėti animacijos ir interaktyvumo į interneto svetaines. Flash yra dažniausiai naudojamas sukurti animaciją, reklamas ir įvairius tinklalapio komponentus, integruoti video į svetaines ir sukurti „turtingas“ interneto svetaines (vos ne tokią pat programą naršyklėje kaip veikiančią kompiuteryje). Adobe Flash programinė įranga veikia įvairiose operacinėse sistemose – Windows, Linux, Mac OS X, Solaris [10].

World Wide Web Consortium (W3C) – konsorciumas, leidžiantis programinės įrangos standartus („rekomendacijas“, kaip jie jas vadina) žiniatinkliui (angl. World Wide Web). Konsorciui vadovauja Tim Berners-Lee, sukūręs URL (Uniform Resource Locator), HTTP (HyperText Transfer Protocol) ir HTML (HyperText Markup Language), interneto technologinį pagrindą [10].

## 9 Priedai

### 9.1 Priedas Nr. 1 (SQL programinis kodas)

```
-----  
  
--  
-- Table structure for table `rk_busena`  
--  
  
DROP TABLE IF EXISTS `rk_busena`;  
CREATE TABLE IF NOT EXISTS `rk_busena` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Pavadinimas` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `Spalva` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=6 ;  
  
-----  
  
--  
-- Table structure for table `rk_grupe`  
--  
  
DROP TABLE IF EXISTS `rk_grupe`;  
CREATE TABLE IF NOT EXISTS `rk_grupe` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Pavadinimas` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `Rikiavimas` int(255) DEFAULT NULL,  
  `FK_Grupe_ID` int(11) NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `FK_Grupe_ID` (`FK_Grupe_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=2 ;  
  
-----  
  
--  
-- Table structure for table `rk_nuoma`  
--  
  
DROP TABLE IF EXISTS `rk_nuoma`;  
CREATE TABLE IF NOT EXISTS `rk_nuoma` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Pavadinimas` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `Kaina` decimal(10,2) NOT NULL,  
  `ObjectElementID` int(11) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=11 ;  
  
-----  
  
--  
-- Table structure for table `rk_planai`  
--  
  
DROP TABLE IF EXISTS `rk_planai`;  
CREATE TABLE IF NOT EXISTS `rk_planai` (  
  `Pavadinimas` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `Plano_nuoroda` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `Plano_nuoroda_th` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `Rikiavimas` int(11) DEFAULT NULL,
```

```

`ID` int(11) NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=2 ;

```

-----

```

--
-- Table structure for table `rk_plano_pogrupio_elementai`
--

```

```

DROP TABLE IF EXISTS `rk_plano_pogrupio_elementai`;
CREATE TABLE IF NOT EXISTS `rk_plano_pogrupio_elementai` (
  `Plano_pozicija` polygon DEFAULT NULL,
  `Pavadinimas` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `FK_Busena_ID` int(11) NOT NULL,
  `FK_Plano_pogrupis_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `FK_Busena_ID` (`FK_Busena_ID`),
  KEY `FK_Plano_pogrupis_ID` (`FK_Plano_pogrupis_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=7 ;

```

-----

```

--
-- Table structure for table `rk_plano_pogrupis`
--

```

```

DROP TABLE IF EXISTS `rk_plano_pogrupis`;
CREATE TABLE IF NOT EXISTS `rk_plano_pogrupis` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `FK_Planai_ID` int(11) NOT NULL,
  `FK_Grupe_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `FK_Planai_ID` (`FK_Planai_ID`),
  KEY `FK_Grupe_ID` (`FK_Grupe_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=2 ;

```