

VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ KATEDRA

**Transporto priemonių skelbimų aplikacija**

**Application for Vehicle Advertisement**

Programų sistemų inžinerijos I laboratorinis darbas Nr. 1

Atliko:	2 kurso 5 grupės studentai
	Toma Burneikaitė
	Žygimantas Stongvilas
	Mantas Jurčius
	Rimvydas Meškauskas
Darbo vadovas:	asist., dr., Vytautas Valaitis

## Anotacija

Šiame projektiniame darbe pristatomas automobilių skelbimų programėlės „AutoInf“ įgyvendinimas. Projektas „AutoInf“ skirta palengvinti automobilio paieškas internete. Programėlė, skirta mobiliems įrenginiams, suteikia galimybę vartotojui lengviau rasti jį dominančią transporto priemonę, sujungdama visus skelbimus, esančius Europos automobilių pardavimo skelbimų puslapiuose. Programų sistemos architektūra apibrėžiama naudojantis 4+1 architektūros pjūvių modelį. Toliau pristatomi programos reikalavimai, bei pateikiama dalykinės srities analizė.

# Turinys

<b>1</b>	<b>Įvadas</b>	<b>4</b>
<b>2</b>	<b>Projektavimas</b>	<b>5</b>
2.1	Užduotys ir jų vykdymo scenarijai . . . . .	5
2.1.1	Sistemos vykdomos užduotys . . . . .	5
2.1.2	Užduoties „Prisiregistruoti“ įgyvendinimas . . . . .	8
2.1.3	Užduoties „Prisijungti“ įgyvendinimas . . . . .	9
2.1.4	Užduoties „Rasti skelbimą“ įgyvendinimas . . . . .	10
2.1.5	Užduoties „Išsaugoti mėgstamiausią“ įgyvendinimas . . . . .	11
2.1.6	Užduoties „Pašalinti mėgstamiausią“ įgyvendinimas . . . . .	12
2.1.7	Užduoties „Peržiūrėti mėgstamiausius“ įgyvendinimas . . . . .	13
2.1.8	Užduoties „Tvarkyti skelbimų šaltinius“ įgyvendinimas . . . . .	14
2.2	Struktūrinis programų sistemos modelis . . . . .	15
2.2.1	Klasių diagrama . . . . .	15
2.2.2	Objektų diagrama . . . . .	16
2.3	Programų sistemos komponentai . . . . .	17
2.3.1	Konteksto diagrama . . . . .	17
2.3.2	Skelbimų rinkimo sistemos dekompozicija . . . . .	18
2.3.3	Sistemos branduolio dekompozicija . . . . .	19
2.4	Dinaminis programų sistemos modelis . . . . .	20
2.4.1	Veiklos diagrama . . . . .	20
2.5	Komponentų išskirstymas tinkle . . . . .	23
2.5.1	Komponentų ryšių su artifaktais diagrama . . . . .	23
2.5.2	Mazgų diagrama . . . . .	24
<b>3</b>	<b>Išvada</b>	<b>25</b>
<b>4</b>	<b>Terminų žodynas</b>	<b>26</b>
<b>5</b>	<b>Literatūros sąrašas</b>	<b>27</b>

# 1 Įvadas

## **Programų sistemos pavadinimas**

Pilnas programų sistemos pavadinimas – automobilių skelbimų aplikacija „AutoInf“

## **Dalykinė sritis**

Automobilių skelbimų iš visos Europos pateiktis vartotojams

## **Probleminė sritis**

Užsienyje galima rasti pigesnių automobilių

## **Naudotojai**

Žmonės, ieškantys automobilių

## **Darbo pagrindas**

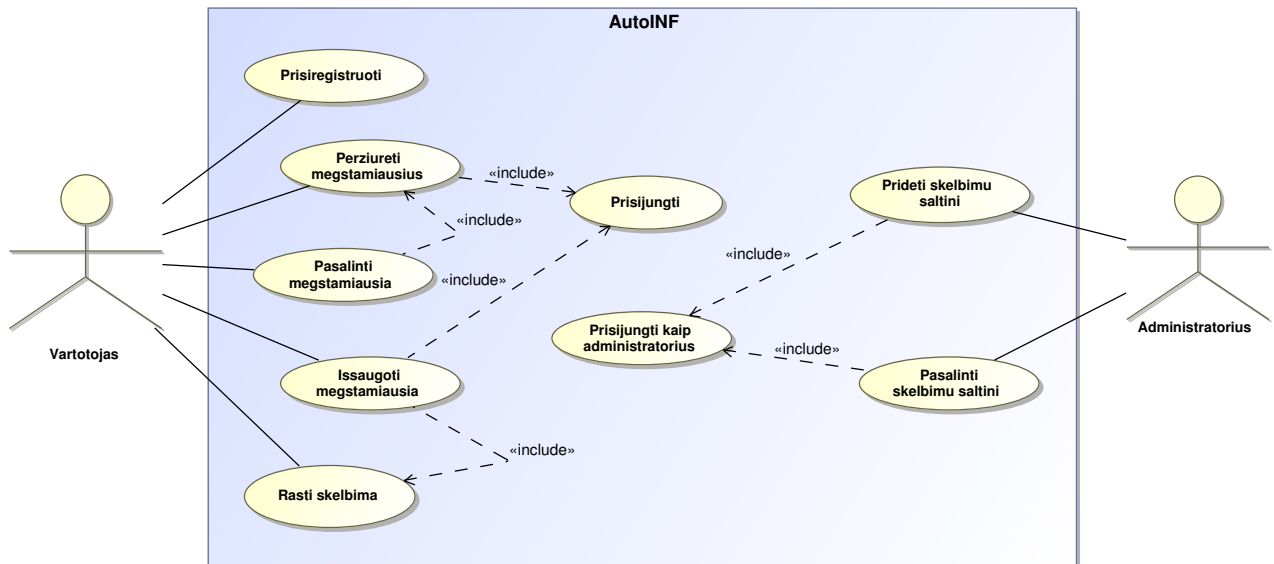
Dokumentas parengtas kaip programų sistemos inžinerijos dalyko laboratorinis darbas Nr. 1, kuriame pateikiamas suprojektuotos sistemos aprašymas.

## 2 Projektavimas

### 2.1 Užduotys ir jų vykdymo scenarijai

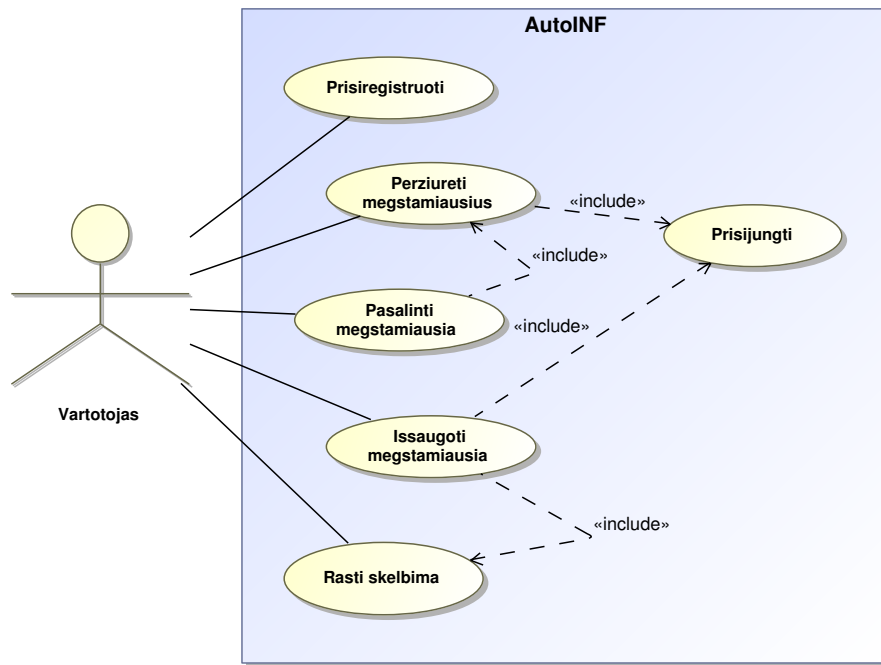
#### 2.1.1 Sistemos vykdomos užduotys

Žemiau pavaizduotoje diagramoje pavaizduoti tikslai, kurių siekia sistema besinaudojantys agentai - automobilių ieškantys vartotojai bei sistemos administratoriai. Į sistemą žiūrima kaip į vieną visumą, neatskleidžiant jos implementacijos detalių.



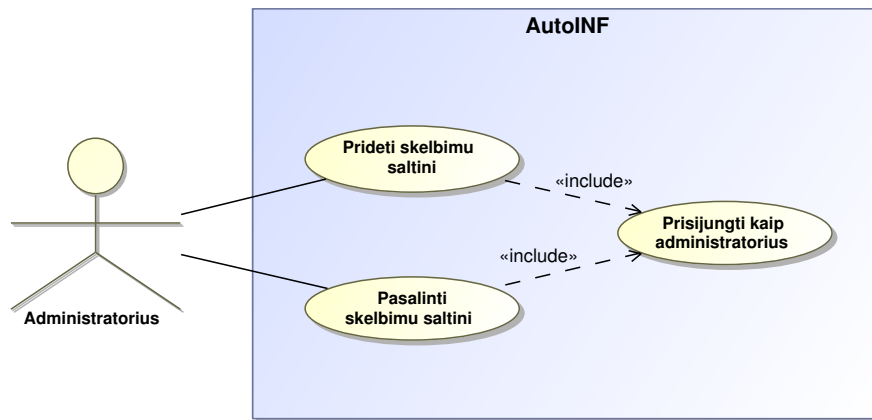
1 pav.: Sistemos užduočių diagrama

Diagramoje 1 yra pavaizduoti tikslai, kurių siekia sistema besinaudojantys agentai - vartotojas ir administratorius



2 pav.: Sistemos užduočių diagrama iš vartotojo perspektyvos

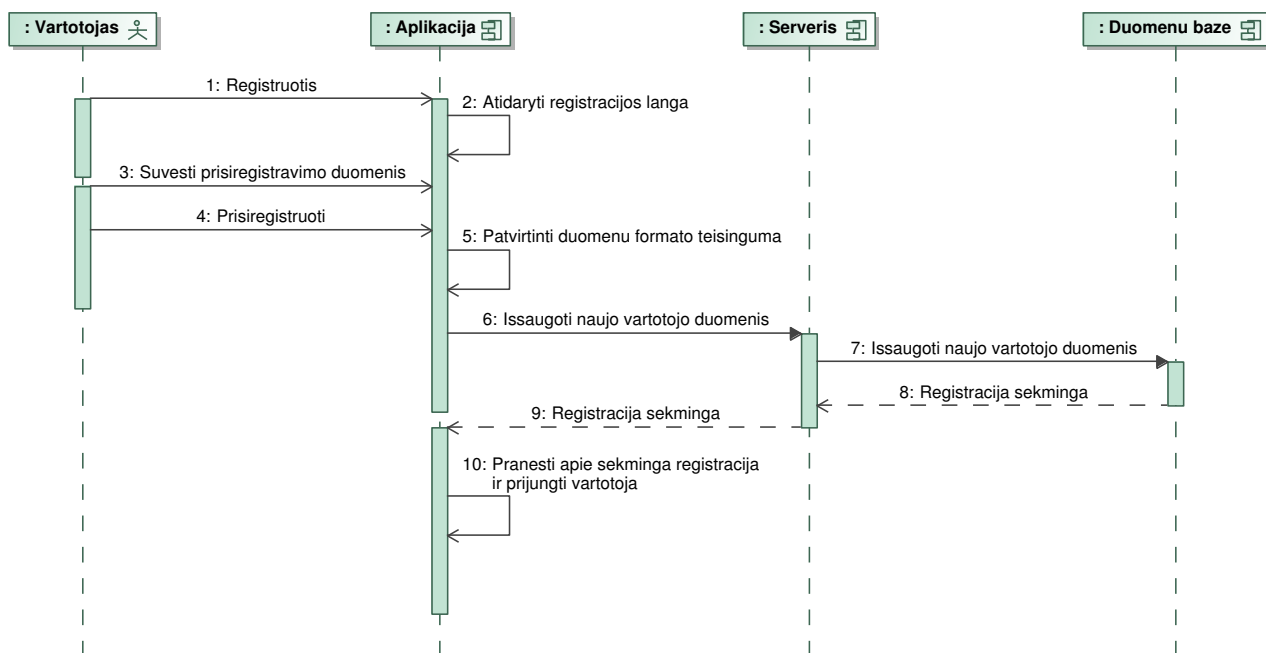
Diagramoje 2 pateiktos vartotojui prieinamos „AutoINF“ funkcijos: registracija, prisijungimas, peržiūrėti mėgstamiausius, pašalinti mėgstamiausius, išsaugoti mėgstamiausius, rasti skelbimą. Plačiau apie šias funkcijas rašoma tolimesnėse sekų diagramose.



3 pav.: Sistemos užduočių diagrama iš administratoriaus perspektyvos

Diagramoje 3 pateiktos administratoriui prieinamos „AutoINF“ funkcijos: pridėti skelbimų šaltinį, pašalinti skelbimų šaltinį, pašalinti vartotojo paskyrą. Plačiau apie šias funkcijas rašoma tolimesnėse sekų diagramose. Administratorius prisijungia taip pat kaip ir vartotojas, tik jam suteiktos administratoriaus teisės.

### 2.1.2 Užduoties „Prisiregistruoti“ įgyvendinimas

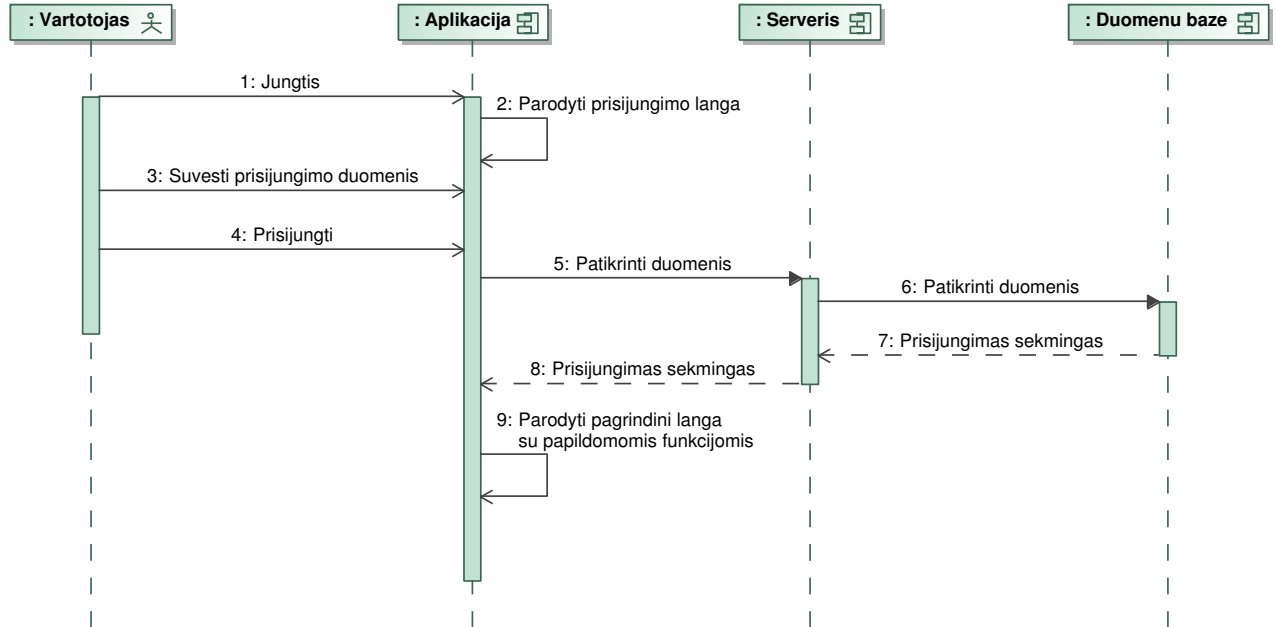


4 pav.: Užduoties „Prisiregistruoti“ sekų diagrama

4 diagramoje pavaizduotas vartotojo registracijos sistemoje užduoties vykdymas. Vartotojas, atidaręs registracijos langą, turi suvesti prisiregistravimo duomenis. Paspaudus registracijos mygtuką, aplikacija patikrina, ar duomenys suvesti reikiamu formatu. Jei formatas teisingas, tai duomenys išsaugomi duomenų bazėje. Tada vartotojas yra informuojamas apie sėkmingą registraciją ir yra prijungiamas prie sistemos.



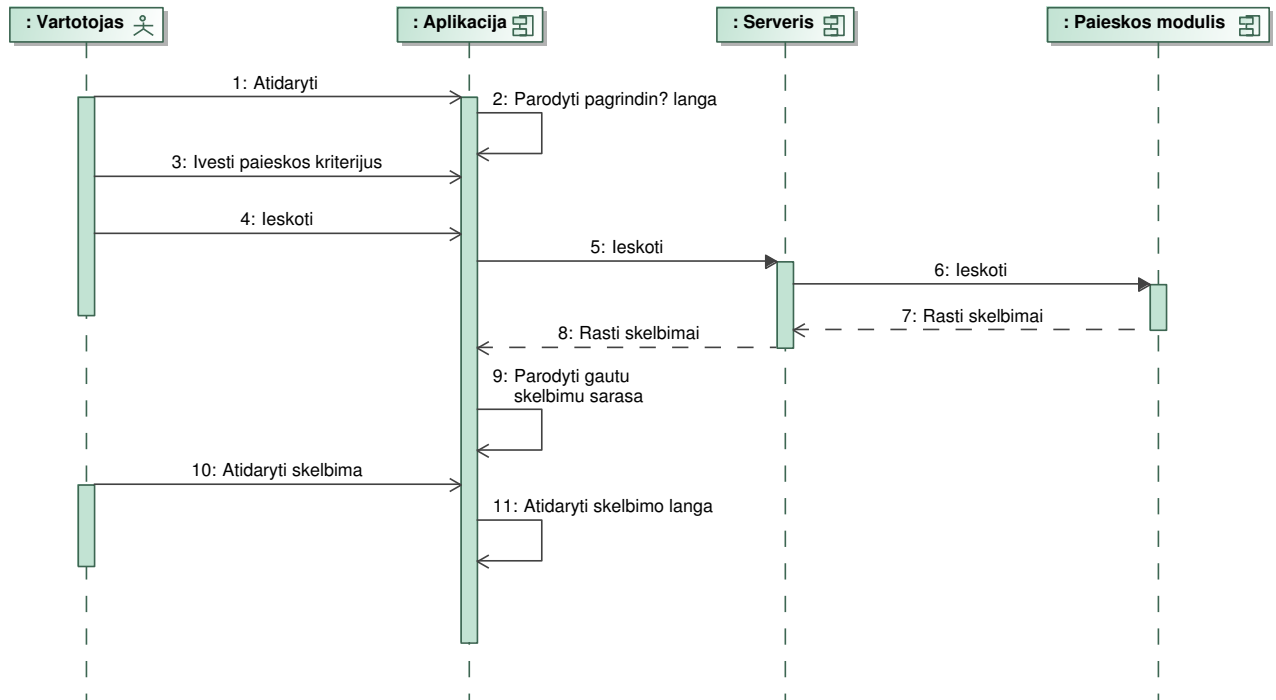
### 2.1.3 Užduoties „Prisijungti“ įgyvendinimas



5 pav.: Užduoties „Prisijungti“ sekų diagrama

5 diagramoje pavaizduotas vartotojo prisijungimo sistemoje užduoties vykdymas. Vartotojas, atidaręs prisijungimo langą, turi suvesti prisijungimo duomenis ir tada paspaudus prisijungimo mygtuką, duomenų bazėje yra patikrinama, ar toks vartotojas egzistuoja. Vartotojui prisijungus atsidaro pagrindinis langas, kuriame atsiranda funkcijos, prieinamos tik registruotiems vartotojams.

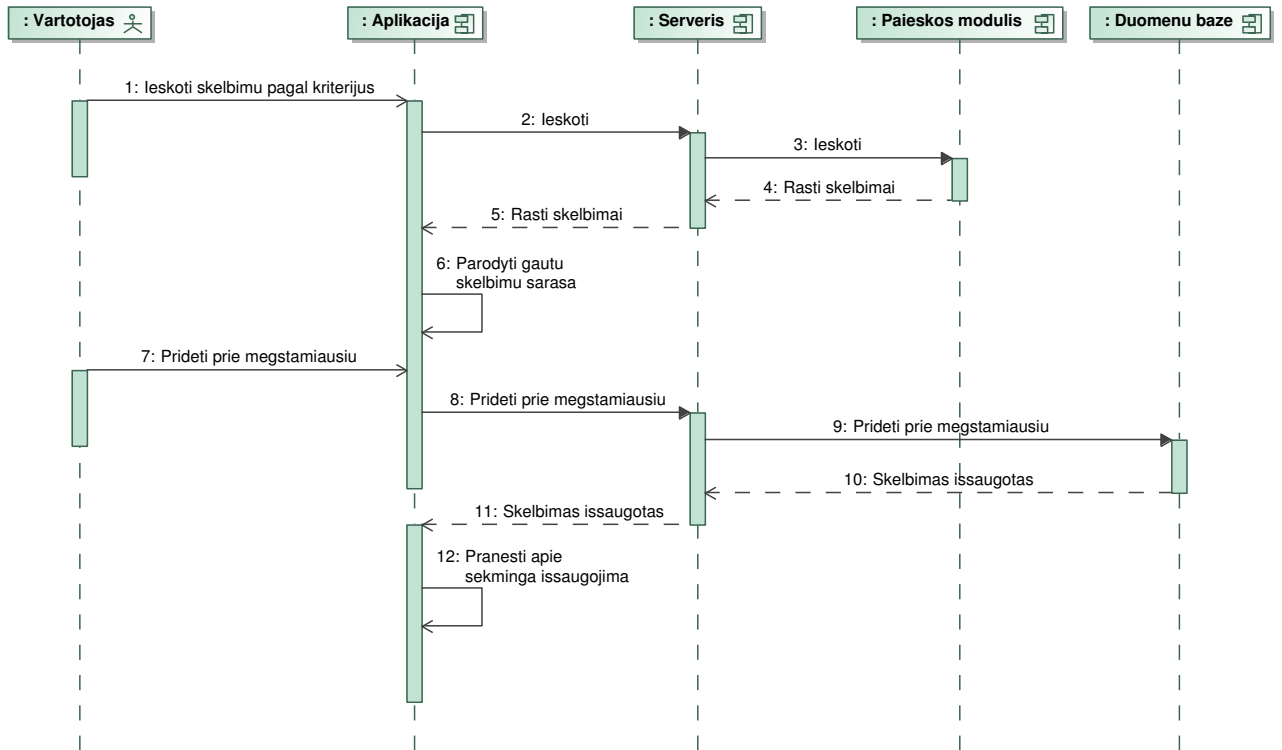
### 2.1.4 Užduoties „Rasti skelbimą“ įgyvendinimas



6 pav.: Užduoties „Rasti skelbimą“ sekų diagrama

6 diagramoje pavaizduotas skelbimų ieškojimo sistemoje užduoties vykdymas. Vartotojas nebūtinai turi būti prisijungęs, norint naudotis šia sistemos funkcija. Pirmiausia vartotojas turi pagrindiniame lange įvesti paieškos kriterijus (kai kurie filtrai yra privalomi). Paspaudus paieškos mygtuką, sistema suranda filtrą atitinkančius skelbimus ir parodo juos. Vartotojui paspaudus ant skelbimo atsidaro skelbimo langas.

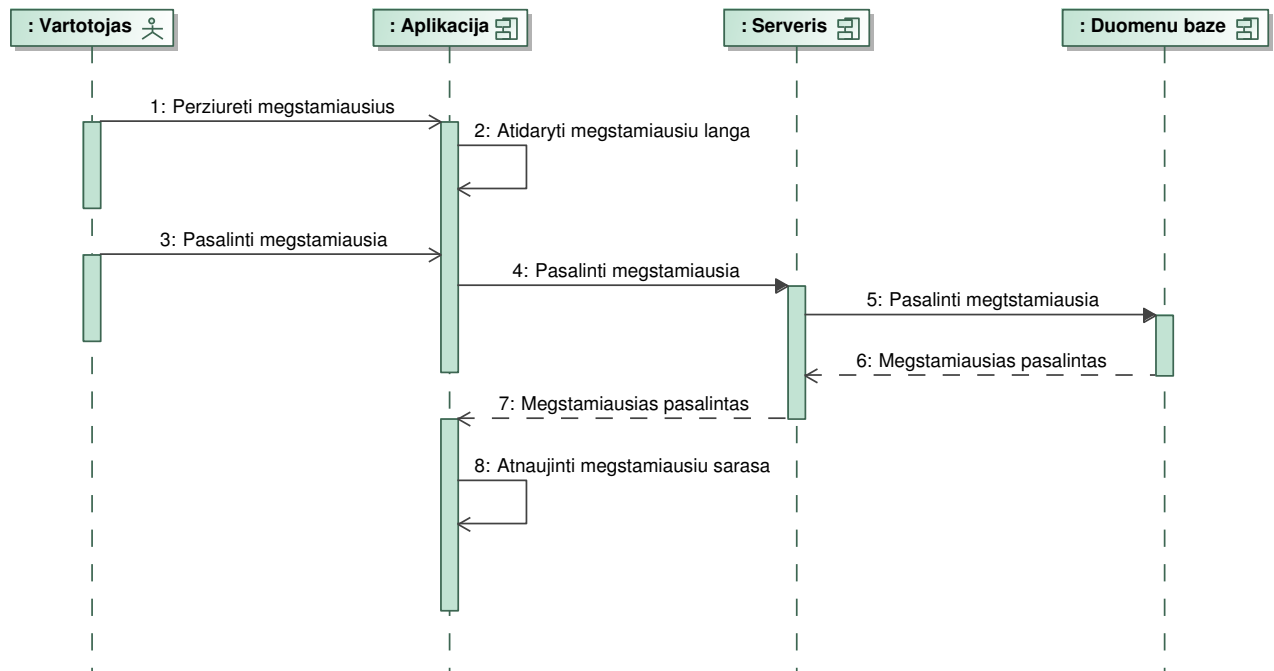
### 2.1.5 Užduoties „Išsaugoti mėgstamiausią“ įgyvendinimas



7 pav.: Užduoties „Išsaugoti mėgstamiausią“ sekų diagrama

7 diagramoje pavaizduotas mėgstamiausio skelbimo išsaugojimo užduoties vykdymas. Ši funkcija yra pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos. Norint išsaugoti skelbimą, vartotojas pirmiausiai turi susirasti skelbimą, ir tada prie jo paspausti skelbimo išsaugojimo mygtuką. Apie sėkmingą skelbimo išsaugojimą vartotojas yra informuojamas.

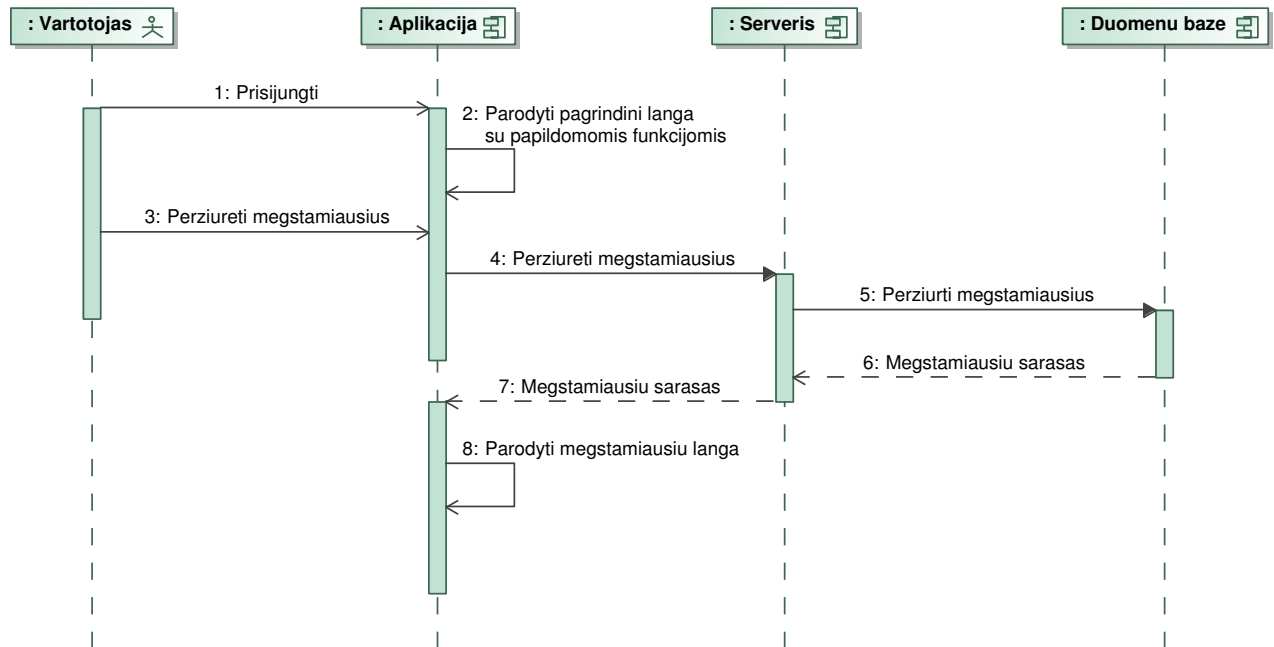
### 2.1.6 Užduoties „Pašalinti mėgstamiausią“ įgyvendinimas



8 pav.: Užduoties „Pašalinti mėgstamiausią“ sekų diagrama

8 diagramoje pavaizduotas mėgstamiausio skelbimo pašalinimo užduoties vykdymas. Ši funkcija yra pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos. Pirmiausiai vartotojas atsidaro mėgstamiausių skelbimų sąrašą. Pasirenką kurį skelbimą jis nori pašalinti ir paspaudus pašalinimo mygtuką. Sėkmingai pašalinus skelbimą iš sąrašo, šis sąrašas yra atnaujinamas.

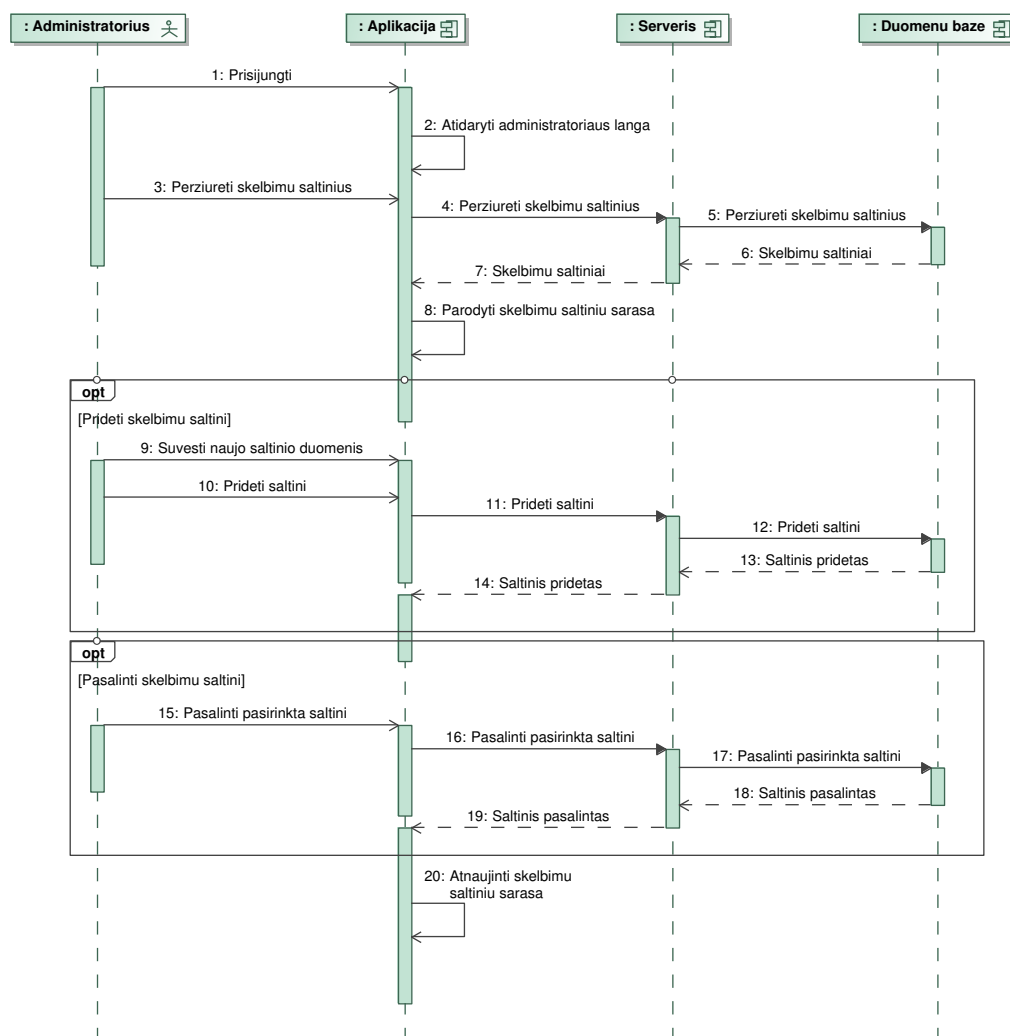
### 2.1.7 Užduoties „Peržiūrėti mėgstamiausius“ įgyvendinimas



9 pav.: Užduoties „Peržiūrėti mėgstamiausius“ sekų diagrama

9 diagramoje pavaizduotas peržiūrėti mėgstamiausius skelbimus užduoties vykdymas. Ši funkcija pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos. Vartotojui pasirinkus mėgstamiausių skelbimų langą, sistema parodo šio vartotojo mėgstamiausių skelbimų sąrašą.

## 2.1.8 Užduoties „Tvarkyti skelbimų šaltinius“ įgyvendinimas



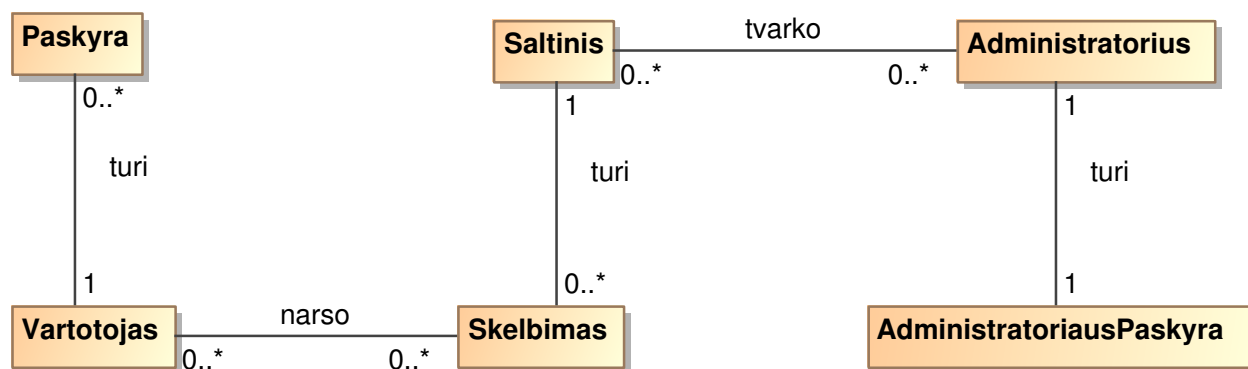
10 pav.: Užduoties „Tvarkyti skelbimų šaltinius“ sekų diagrama

10 diagramoje yra pavaizduotas skelbimų šaltinių tvarkymo užduoties vykdymas. Šia funkcija gali pasinaudoti tik administratorius. Administratorius atsidaręs skelbimų šaltinių langą gali pasirinkti, ką jis nori su šaltiniais daryti: pridėti naują skelbimų šaltinį arba pašalinti šaltinį. Norint pridėti šaltinį, administratorius spaudžia mygtuką skirtą naujam šaltiniui pridėti, o norint pašalinti skelbimą, administratorius turi pasirinkti šaltinį iš šaltinių sąrašo, ir pasirinkti šaltinio pašalinimą. Tiek po šaltinio pridėjimo, tiek po šaltinio pašalinimo, skelbimų šaltinių sąrašas yra atnaujinamas.

## 2.2 Struktūrinis programų sistemos modelis

### 2.2.1 Klasių diagrama

Žemiau pateiktoje klasių diagramoje yra išskirtos pagrindinės esybės, kurios yra naudojamos sistemoje. Klases siejantys ryšiai pasižymi kardinalumu. Kitaip sakant, nustatytas konkretus ryšių skaičius ar skaičių aibė, kurios turės klasės egzempliorius su tam tikros kitos klasės egzemplioriais.

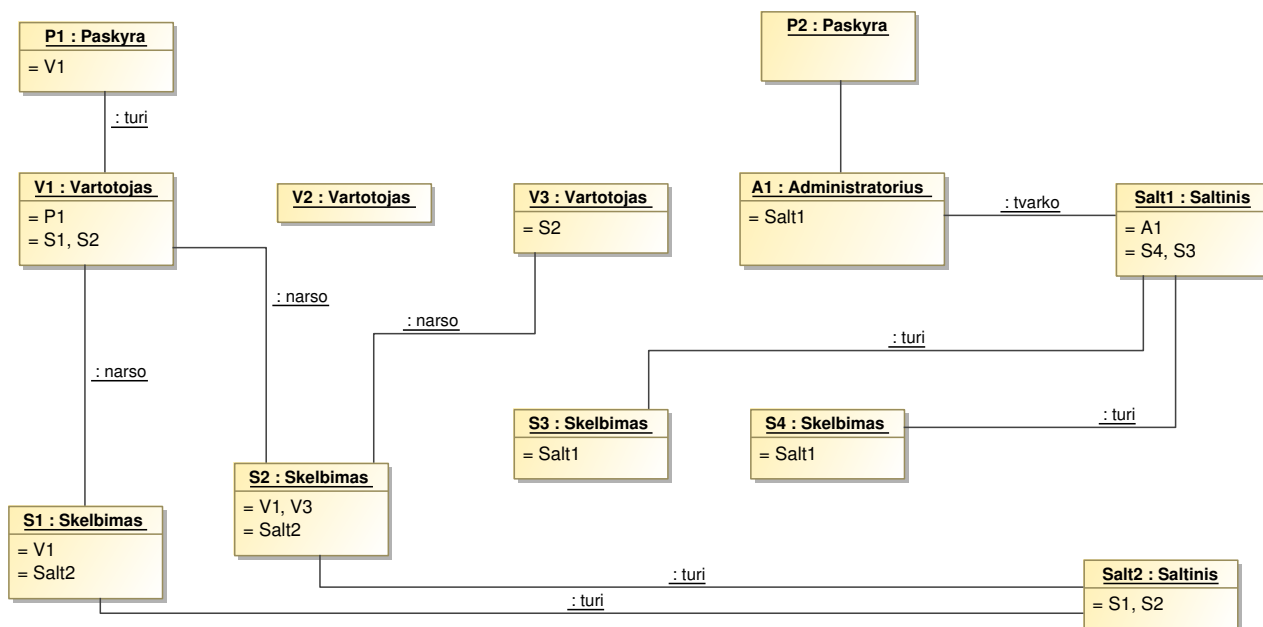


11 pav.: Užduoties „Klasių diagrama“ sekų diagrama

Šioje sistemoje egzistuoja dviejų rūšių paskyros: paprasta paskyra, ir administratoriaus paskyra. Paprastas vartotojas gali turėti daug paskyrų (pvz.: pamiršta visus duomenis apie savo paskyrą, tai jis gali susikurti naują), o paskyra priklauso tik vienam vartotojui. Administratorius gali turėti tik vieną paskyrą, vartotojui išduodamas vienkartinis kodas, kurį įvedus sukuriamą specialią administratoriaus paskyrą. Administratorius tvarko šaltinius (t.y. prideda arba pašalina šaltinius). Kiekvienas šaltinis gali būti tvarkomas kelių administratorių, ir taip pat kiekvienas administratorius gali tvarkyti kelis šaltinius. Vartotojai gali naršyti daug skelbimų ir visi skelbimai gali būti naršomi kelių vartotojų. Visi skelbimai yra paimti iš kokių nors šaltinių. Kiekvienas skelbimas gali būti paimtas tik iš vieno šaltinio, tačiau visi šaltiniai gali turėti daug skelbimų.

### 2.2.2 Objektų diagrama

Šiame skyrelyje esančios objektų diagramos iš esmės patvirtina anksčiau pateiktą klasių diagramą. Taip pat pateikia tipinių pavyzdžių, kaip klasės bus naudojamos sistemoje.



12 pav.: Užduoties „Objektų diagrama“ sekų diagrama

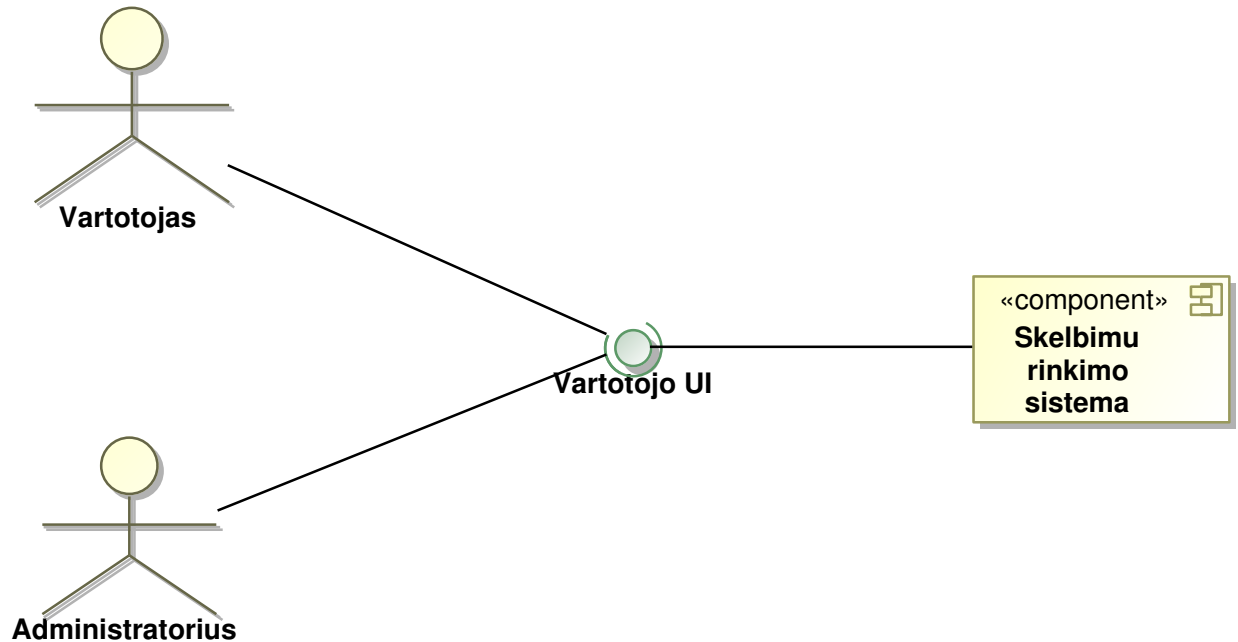
12 diagramoje parodyta tipinė sistemos situacija, kai yra sukurtos dvi paskyros, viena yra vartotojo, kita administratoriaus. Taip pat yra dar du vartotojai, vienas iš jų tuo momentu nieko nedaro sistemoje, kol kitas, kartu su registruotu vartotoju, naršo skelbimus (keli vartotojai gali naršyti tą patį skelbimą vienu metu). Taip pat yra keturi skelbimai, ir visi priklauso kažkuriam vienam šaltiniui, kol šaltiniai turi daugiau nei po vieną skelbimą (šaltiniai gali ir neturėti skelbimų). Taip pat iš diagramos galima matyti, kad administratorius gali tvarkyti šaltinius.



## 2.3 Programų sistemos komponentai

Sistemos komponentų dekompozicija įgyvendinta "top-down" būdu.

### 2.3.1 Konteksto diagrama



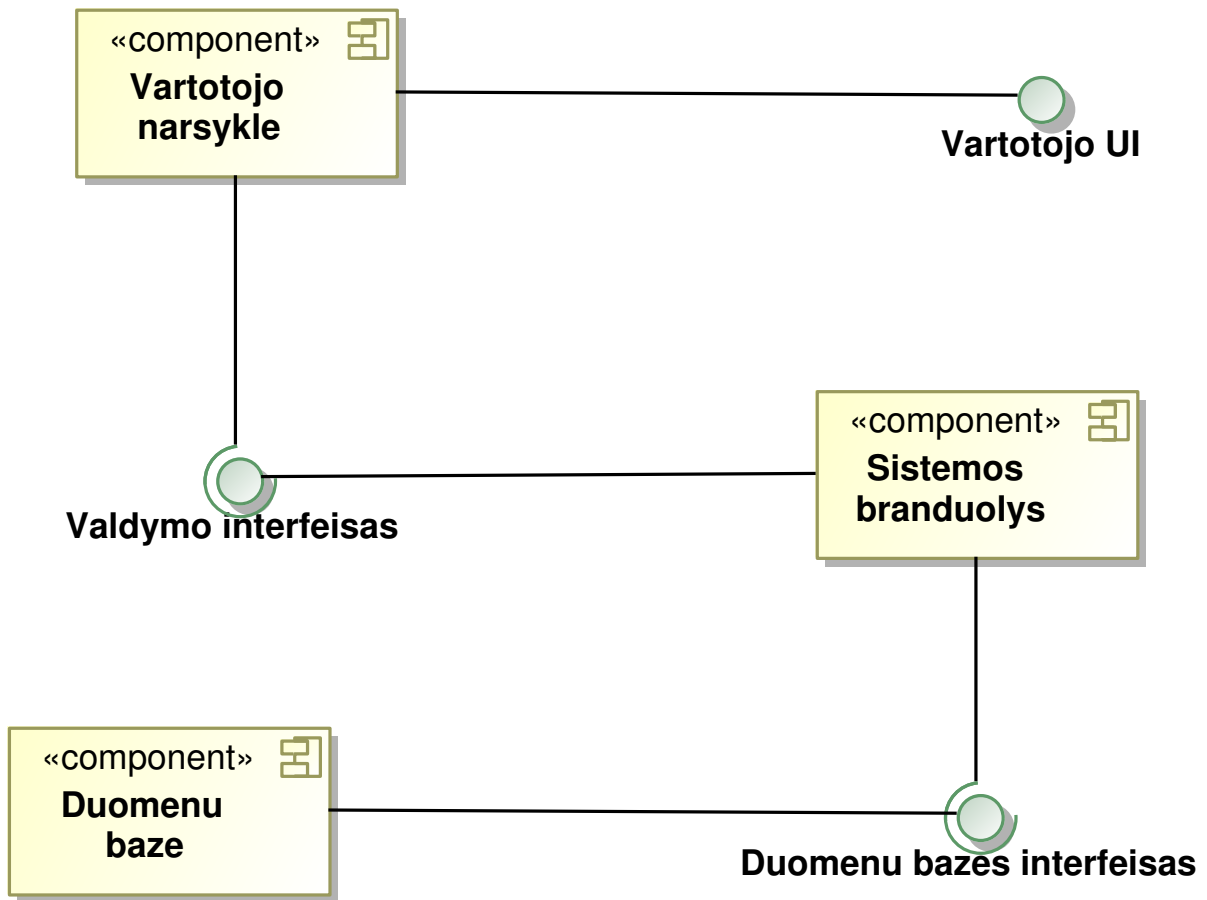
13 pav.: Konteksto komponentų diagrama

13 diagramoje prodomas abstrakčiausias (aukščiausias) komponentų struktūros lygis. Visa sistema laikoma kaip vienas darinys, ir parodoma sąsaja su išore:

- Interfeisai, kuriais išoriniai vartotojai naudojami visa sistema
- Išorinių sistemų interfeisai, kuriais naudojasi sistema

Tiek vartotojas, tiek administratorius turi naudotis tuo pačiu vartotojo UI interfeisu. To priežastis yra tai, kad abu vartotojai naudojami ta pačia sistema, tik su šiek tiek skirtingomis galimybėmis.

### 2.3.2 Skelbimų rinkimo sistemos dekompozicija



14 pav.: Skelbimų rinkimo sistemos komponento dekompozicija

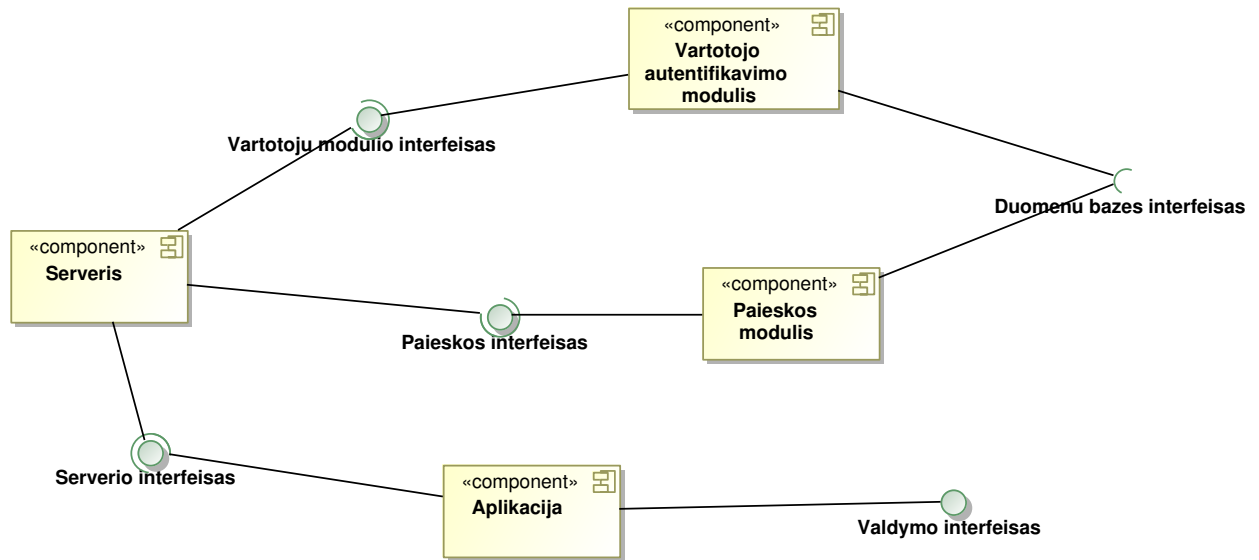
14 diagramoje detalizuojamas 13 diagramoje esantis „Skelbimų rinkimo sistema“

**Vartotojo naršyklė** - komponentas, sukuriantis vartotojo interfeisą bei vykdamas komunikaciją tarp vartotojo ir sistemos branduolio.

**Sistemos branduolys** - komponentas, priimančias prašymus iš vartotojo naršyklės, kontroliuojantis duomenų gavimą ar išsaugojimą duomenų bazėje bei siunčiantis duomenis vartotojo naršyklei.

**Duomenų bazė** - komponentas, kuriame saugoma informacija apie registruotus vartotojus ir kartu tos paskyros mėgstamiausius skelbimus.

### 2.3.3 Sistemos branduolio dekompozicija



15 pav.: Sistemos branduolio komponento dekompozicija

15 diagramoje smulkinamas 14 diagramoje esantis „Sistemos branduolio“ komponentas. Jį sudarantys komponentai:

**Vartotojo autentifikavimo modulis** - šis komponentas vykdo vartotojų autentifikaciją. Jo funkcijos: vartotojų registravimas, prisijungimas.

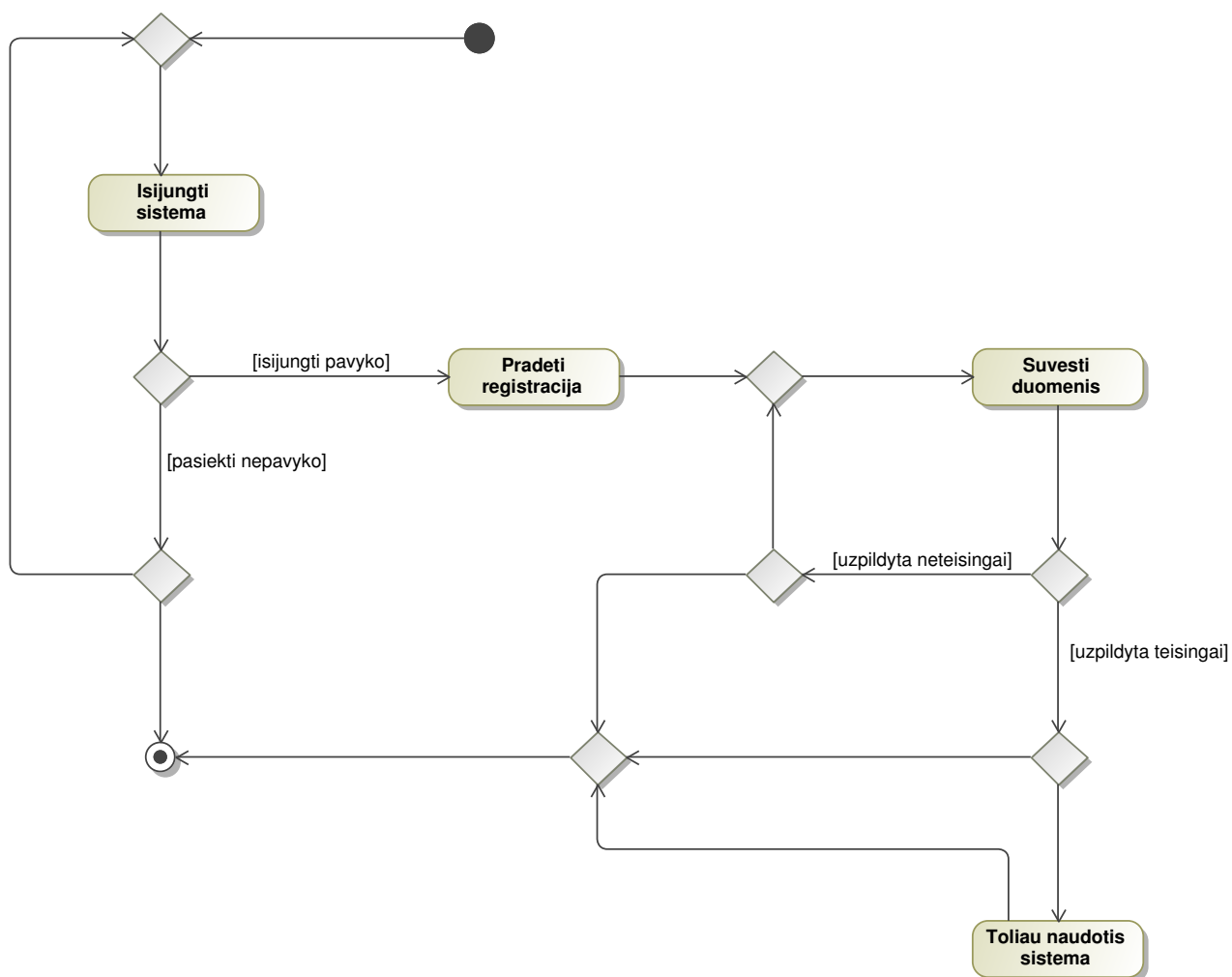
**Paieškos modulis** - pagal tam tikrus nurodytus kriterijus suranda skelbimus.

**Serveris** - komponentas, valdantis prieigą prie duomenų.

**Aplikacija** - komponentas, kuris pateikia vartotojui skelbimų sąrašą, konkrečių skelbimų detlesnę informaciją.

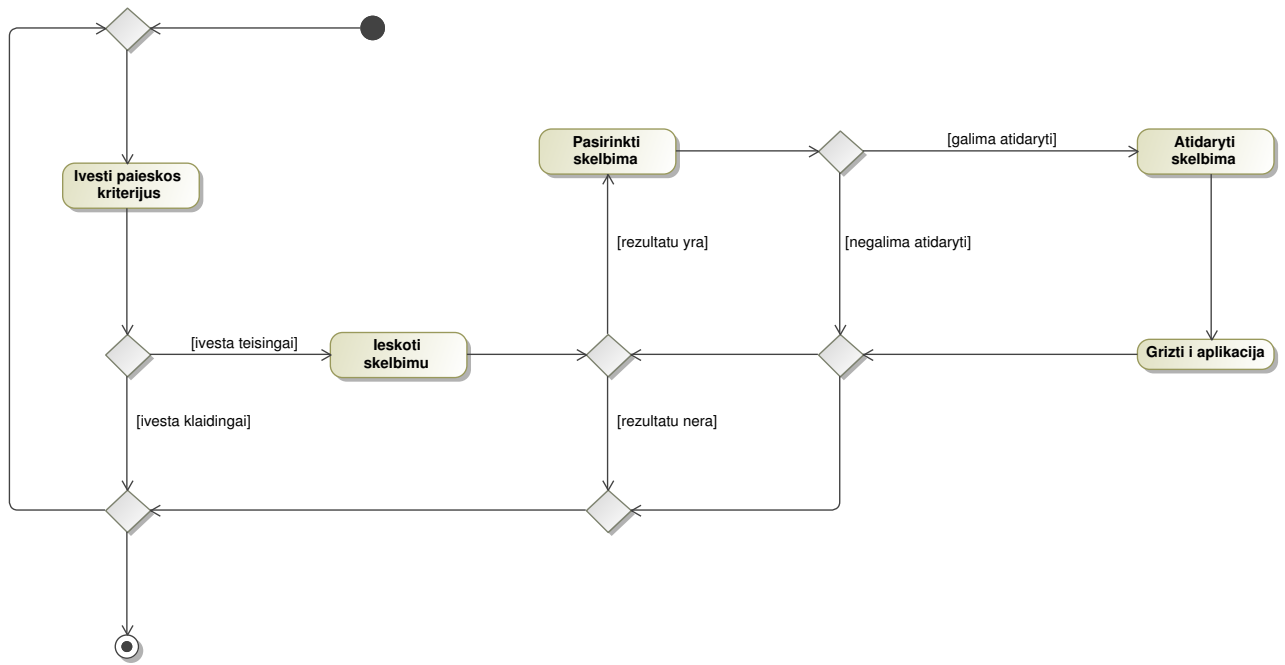
## 2.4 Dinaminis programų sistemos modelis

### 2.4.1 Veiklos diagrama



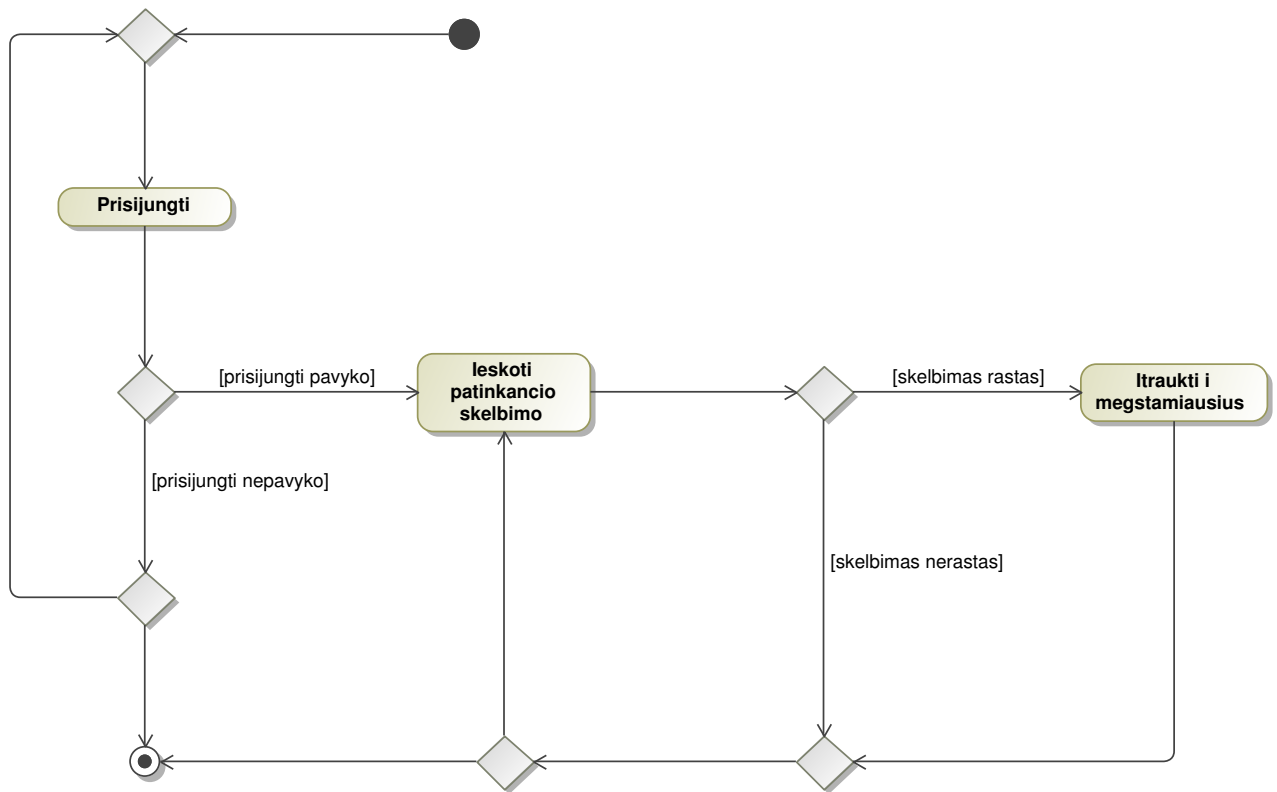
16 pav.: Vartotojo registracijos veiklos diagrama

16 diagramoje parodyti procesai vykstantys registracijos metu. Registravimas vyksta tada, kai vartotojas įsijungia aplikaciją ir pasirenka „Registraciją“. Tada, vartotojui suvedus reikiamus duomenis, sistema patikrina, ar viskas užpildyta teisingai, jei ne, tai vartotojas gali bandyti vesti informaciją iš naujo, arba išeitį iš registravimosi lango. Jei informacija užpildyta teisingai, tai vartotojas gali toliau naudotis aplikacija, arba išeiti iš jos.



17 pav.: Skelbimų paieškos veiklos diagrama

17 diagramoje parodyti procesai vykstantys skelbimo ieškojimo metu. Pagrindiniame lange vartotojas turi įvesti filtrus, pagal kuriuos bus ieškomi skelbimai. Sistema patikrina, ar įvesti kriterijai yra teisingai suvesti. Jei įvesta klaidingai, tai vartotojai gali bandyti iš naujo suvesti paieškos kriterijus, arba išeiti iš programos. Jei paieškos kriterijai įvesti teisingai, tai sistema ieško skelbimų, jei rezultatų nėra, tai galima pasirinkti kitokius filtrus arba išeiti iš programos. Suradus rezultatų, vartotojas gali pasirinkti skelbimą, jei jo negalima atidaryti galima bandyti paiešką atlikti iš naujo, jei galima atidaryti, tai vartotojas gali atidaryti skelbimą ir po to grįžti į aplikaciją.



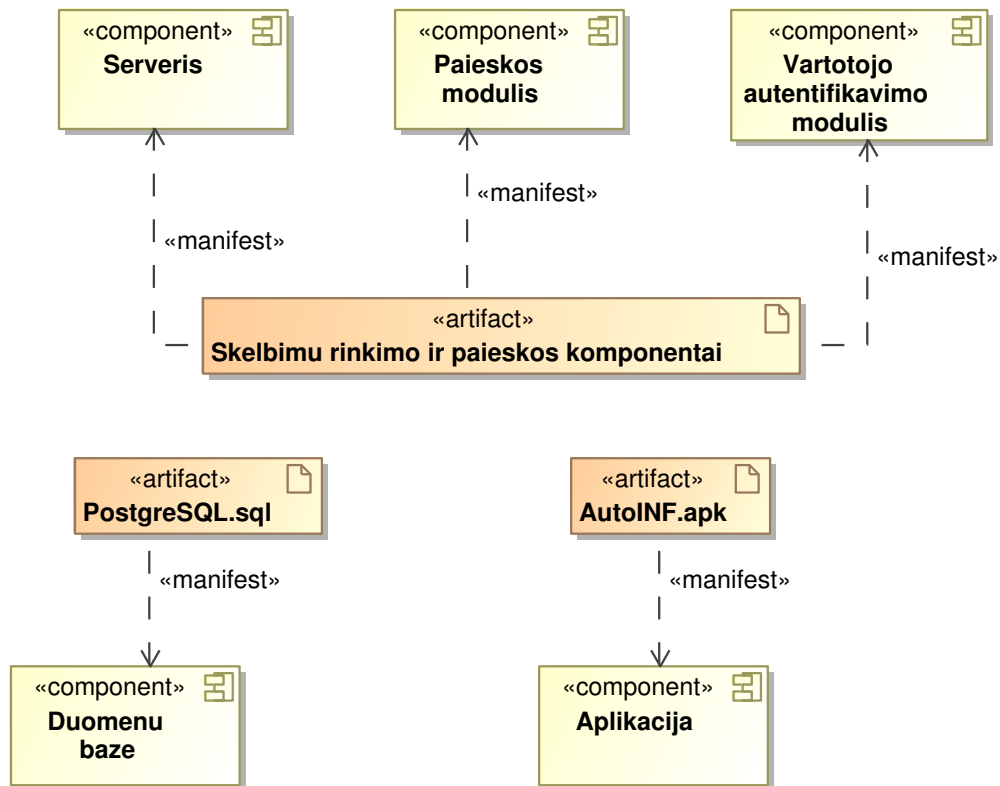
18 pav.: Naujo mėgstamiausio skelbimo veiklos diagrama

18 diagramoje parodyti procesai vykstantys vartotojui norint įtraukti skelbimą į mėgstamiausių sąrašą. Pirmiausia vartotojas turi prisijungti, jei to padaryti nepavyksta, jis gali bandyti iš naujo arba išeiti iš programos. Prisijungus prie sistemos, vartotojas ieško jam patinkantį skelbimą, suradus jį, vartotojas gali įtraukti į mėgstamiausių sąrašą, o po to ieškoti naujo skelbimo, arba išeiti iš šito lango. Jei patinkančio skelbimo vartotojas neranda, tai gali bandyti ieškoti iš naujo arba išeiti iš šito lango.

## 2.5 Komponentų išskirstymas tinkle

### 2.5.1 Komponentų ryšių su artifaktais diagrama

19 pateiktoje komponentų ryšių su artifaktais diagramoje su artifaktais yra išskirti pagrindiniai sistemos artifaktai. Artifaktus ir komponentus tarpusavyje sieja manifestacijos ryšys. Tai reiškia, kad artifakto sudaromoji dalis yra konkretus komponentas.

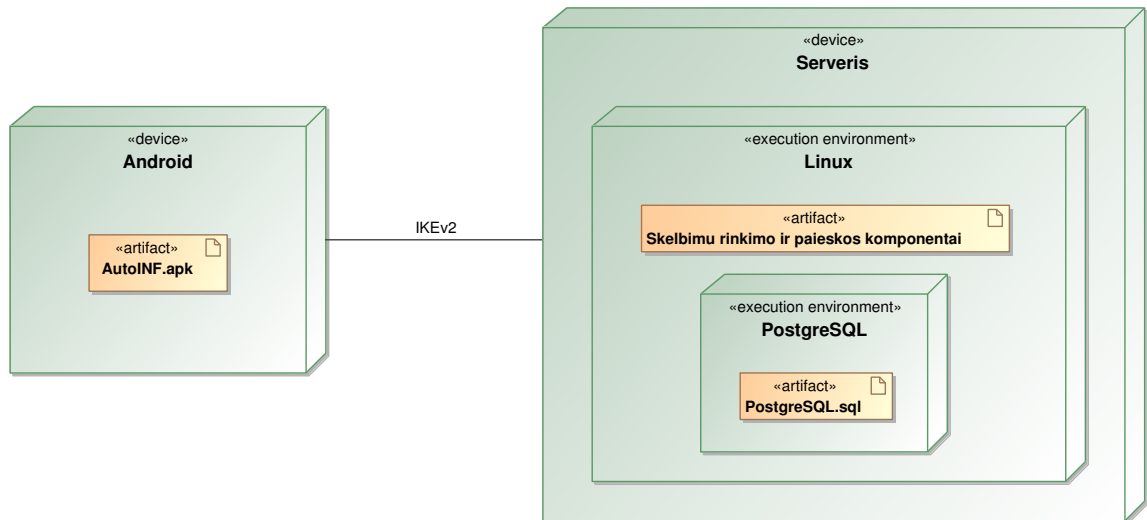


19 pav.: Komponentų ryšių su artifaktais diagrama

Sistema susidaro iš trijų pagrindinių dalių - aplikacijos, skelbimų rinkimo ir paieškos komponentų ir duomenų bazės. Informacijai saugoti apie vartotojų paskyras pasirinkta PostgreSQL duomenų bazė.

### 2.5.2 Mazgų diagrama

Žemiau pateiktoje mazgų diagramoje yra išskirti fiziniai įrenginiai, reikalingi sistemos darbo palaikymui bei artifaktų pasiskirstymas tarp jų.



20 pav.: Mazgų diagrama

Visa sistema yra laikoma serveryje. Šiame serveryje yra sudiejami skelbimų rinkimo ir paieškos komponentai. Tame pačiame serveryje yra ir PostgreSQL duomenų bazė. Vartotojui, norint naudotis sistema iš mobiliojo įrenginio, siunčiama užklausa IKEv2 protokolu siekiant gauti sistemos failus.



### 3 Išvada

Šiame laboratoriniame darbe pasitelkiant skirtingus sistemos pjūvius aprašyta knygų dalinimosi sistemos architektūra. Užduočių pjūvyje išsiaiškinti pagrindiniai agentų tikslai naudojantis sistema. Loginis pjūvis leido išskirti pagrindines esybes bei ryšius tarp jų. Kūrimo pjūvyje atlikta sistemos dekompozicija pradedant nuo bendro komponento toliau ją detalizuojant. Procesų pjūvyje išskirti procesai, jų komunikacija, objektų gyvavimo ciklai. Galiausiai fiziniame pjūvyje apibrėžtas sistemos išdėstymas tinkle. Šis skirtingų požiūrių rinkinys leido iš anksto aptikti sistemoje galimas klaidas bei sukurti tinkamą sistemos architektūrą.

## 4 Terminų žodynas

**Vartotojas** - žmogus, besinaudojantis sistema.

**Skelbimas** - iš šaltinio gautas skelbimas.

**Šaltinis** - puslapis, iš kurio gaunami skelbimai.

## 5 Literatūros sarakšas

Doug Rosenberg and Matt Stephens “Use Case Driven Object Modeling with UML“