

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

Vilnius - 2018

# Anotacija

Dokumentas parengtas kaip programų sistemos inžinerijos dalyko laboratorinis darbas Nr. 1, kuriame pateikiamas suprojektuotos sistemos aprašymas. Programų sistemos architektūra apibrėžiama naudojantis 4+1 architektūros pjūvių modelį.

# Turinys

<b>Anotacija</b>	<b>2</b>
<b>Įvadas</b>	<b>4</b>
<b>1 Užduotys ir jų vykdymo scenarijai</b>	<b>5</b>
1.1 Sistemos vykdomos užduotys . . . . .	5
1.2 Užduoties „Prisiregistruoti“ įgyvendinimas . . . . .	8
1.3 Užduoties „Prisijungti“ įgyvendinimas . . . . .	9
1.4 Užduoties „Rasti skelbimą“ įgyvendinimas . . . . .	10
1.5 Užduoties „Išsaugoti mėgstamiausią“ įgyvendinimas . . . . .	11
1.6 Užduoties „Pašalinti mėgstamiausią“ įgyvendinimas . . . . .	12
1.7 Užduoties „Peržiūrėti mėgstamiausius“ įgyvendinimas . . . . .	13
1.8 Užduoties „Tvarkyti skelbimų šaltinius“ įgyvendinimas . . . . .	14
<b>2 Struktūrinis programų sistemos modelis</b>	<b>15</b>
2.1 Klasų diagrama . . . . .	15
2.2 Objektų diagrama . . . . .	16
<b>3 Programų sistemos komponentai</b>	<b>17</b>
3.1 Konteksto diagrama . . . . .	17
3.2 Skelbimų rinkimo sistemos dekompozicija . . . . .	18
3.3 Sistemos branduolio dekompozicija . . . . .	19
<b>4 Dinaminis programų sistemos modelis</b>	<b>20</b>
4.1 Vartotojo registracijos veiklos diagrama . . . . .	20
4.2 Skelbimų paieškos veiklos diagrama . . . . .	21
4.3 Naujo mėgstamo skelbimo veiklos diagrama . . . . .	22
<b>5 Komponentų išskirstymas tinkle</b>	<b>23</b>
5.1 Komponentų ryšių su artifaktais diagrama . . . . .	23
5.2 Diegimo diagrama . . . . .	24
<b>Rezultatai</b>	<b>25</b>
<b>Išvados</b>	<b>26</b>
<b>Terminų žodynas</b>	<b>27</b>
<b>Literatūros sąrašas</b>	<b>28</b>

## Ivadas

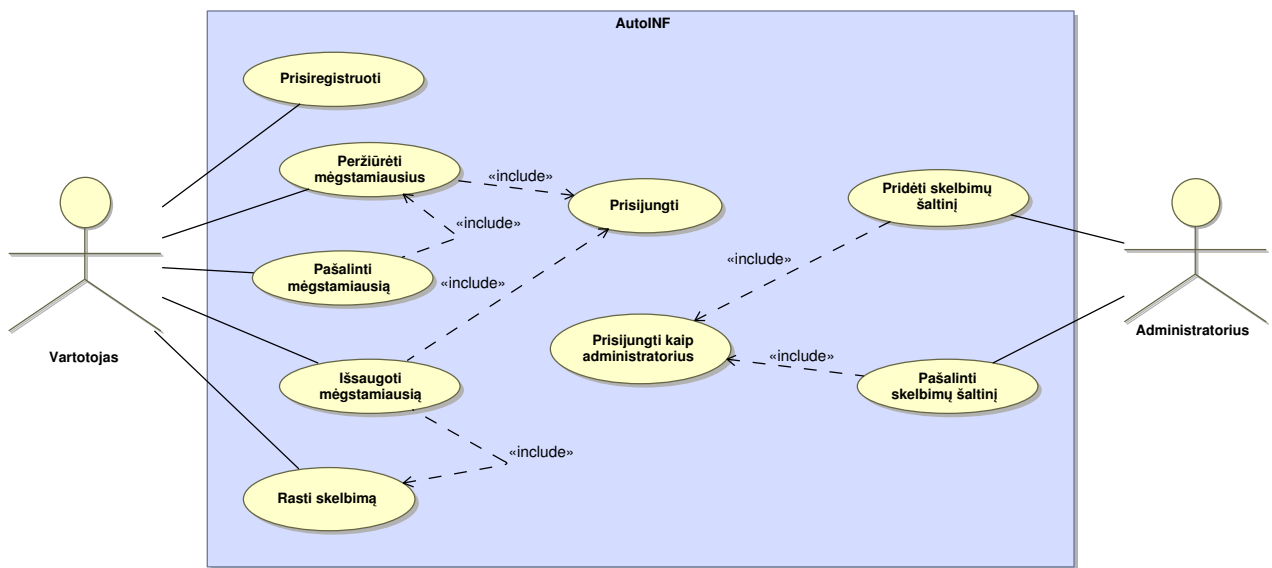
Šiame projektiniame darbe pristatomas automobilių skelbimų programėlės „AutoINF“ įgyvendinimas. Šio projekto vizija yra sukurti mobiliams įrenginiams skirtą programėlę, kuri leistų pasiekti automobilių skelbimus iš visos Europos. Idėja kilo iš to, kad dabar įvairių automobilių skelbimai yra daugybėje skirtingų puslapių (šaltinių) ir norint susirasti sau naudingiausią pasiūlymą dažniausiai reikia pereiti per daugiau nei vieną puslapį. „AutoINF“ projekto tikslas yra sukurti programėlę, kuri supaprastintų automobilių ieškojimo procesą ir padėtų vartotojui sutaupyti laiko tą darant, surinkdama visus skelbimus iš populiariausių Europos automobilių skelbimų puslapių pagal vartotojo nurodytus kriterijus. Taip pat norima, kad programėle būtų paprasta naudotis ir kad šita programėlė sugebėtų pateikti kuo daugiau vartotojui aktualios informacijos apie tai, ko jis ieško.

# 1 Užduotys ir jų vykdymo scenarijai

Šiame skyriuje pavaizduotos ir aprašytos sistemos vartotojų užduotys bei jų vykdymo scenarijai.

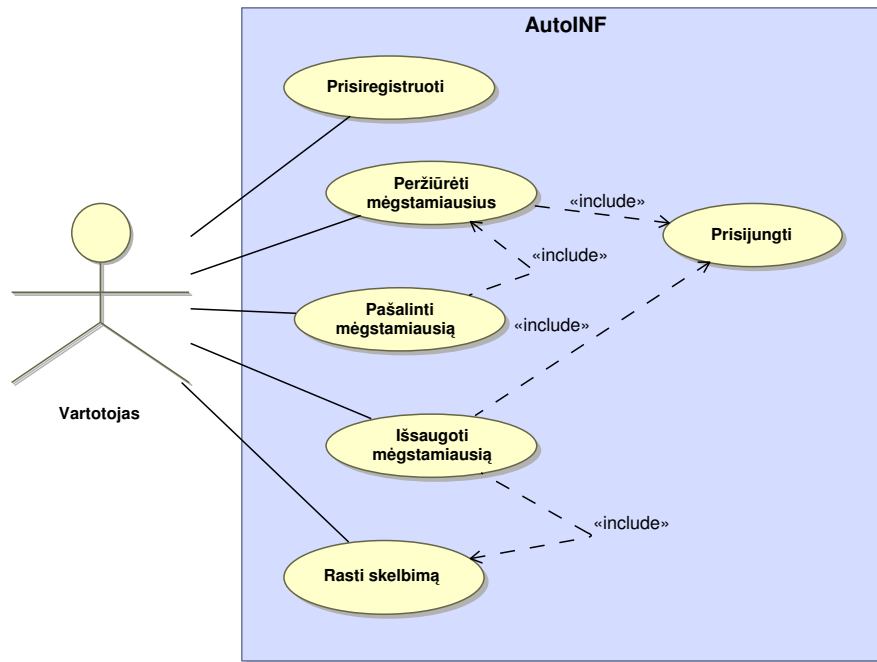
## 1.1 Sistemos vykdomos užduotys

Žemiau esančiame 1 paveikslėlyje pavaizduoti tikslai, kurių siekia sistema besinaudojantys agentai - automobilių ieškantys vartotojai bei sistemos administratoriai. Į sistemą žiūrima kaip į vieną visumą, neatskleidžiant jos implementacijos detalių.



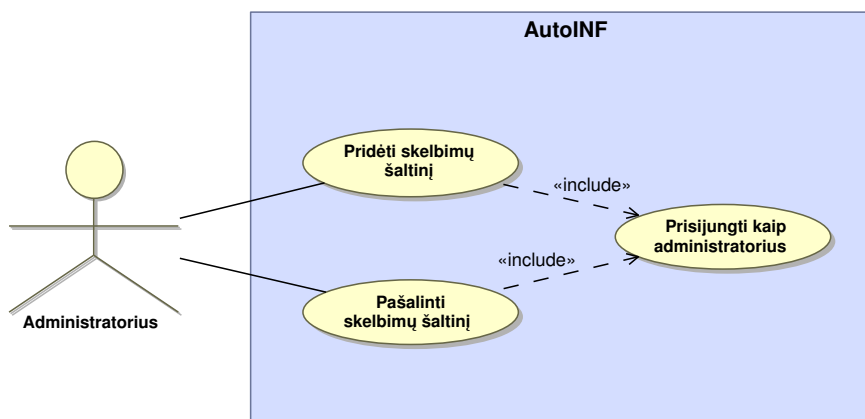
1 pav.: Sistemos užduočių diagrama

2 paveikslėlyje pateiktos vartotojui prieinamos „AutoINF“ funkcijos: registracija, prisijungimas, peržiūrėti mėgstamiausius, pašalinti mėgstamiausią, išsaugoti mėgstamiausią, rasti skelbimą. Daugiau informacijos apie šias funkcijas yra tolimesnėse (4 - 9 pav.) sekų diagramose.



2 pav.: Sistemos užduočių diagrama iš vartotojo perspektyvos

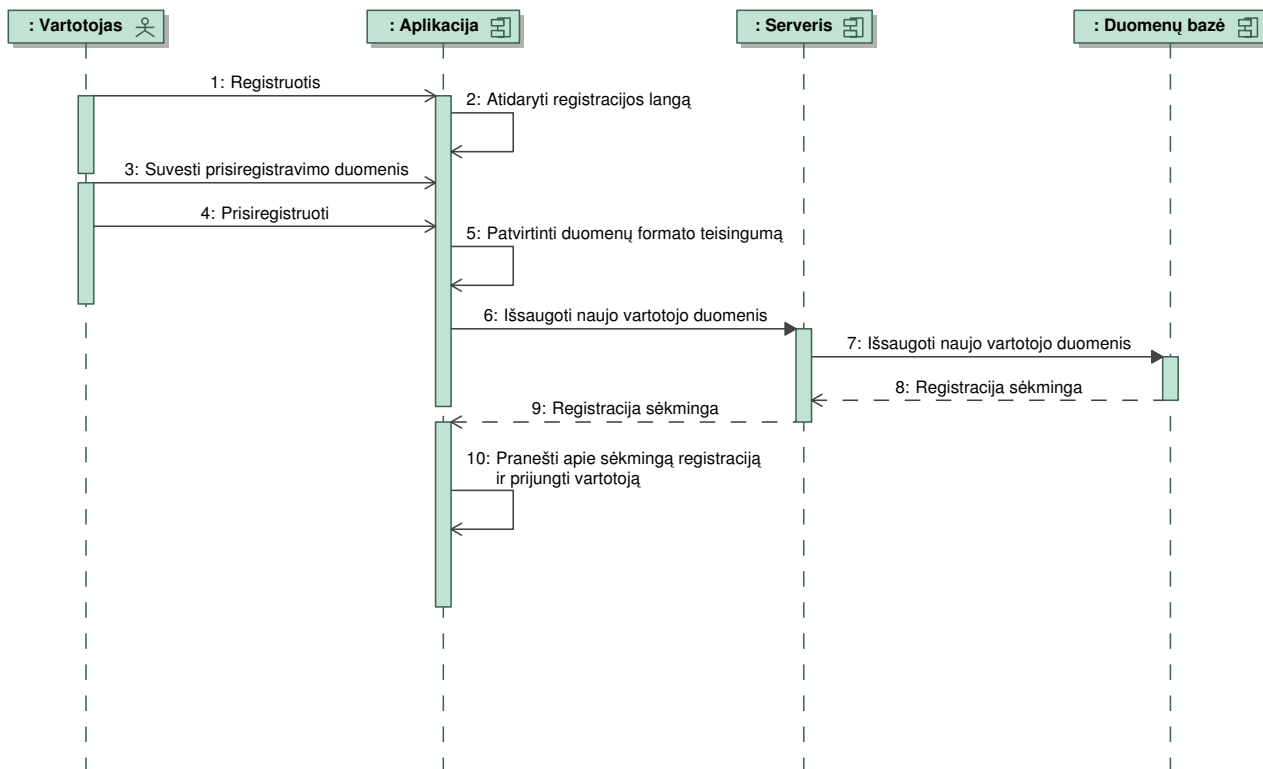
3 paveikslėlyje pateiktos administratoriui prieinamos „AutoINF“ funkcijos: pridėti skelbimų šaltinį, pašalinti skelbimų šaltinį, pašalinti vartotojo paskyrą. Šios funkcijos labiau išplėtojamos tolimesnėje (10 pav.) sekų diagramoje. Administratorius prisijungia taip pat, kaip ir vartotojas, tik jam suteiktos administratoriaus teisės.



3 pav.: Sistemos užduočių diagrama iš administratoriaus perspektyvos

## 1.2 Užduoties „Prisiregistruoti“ įgyvendinimas

Vartotojo registracijos sistemoje užduoties vykdymas pavaizduotas 4 paveikslėlyje. Vartotojas, atidarys 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.

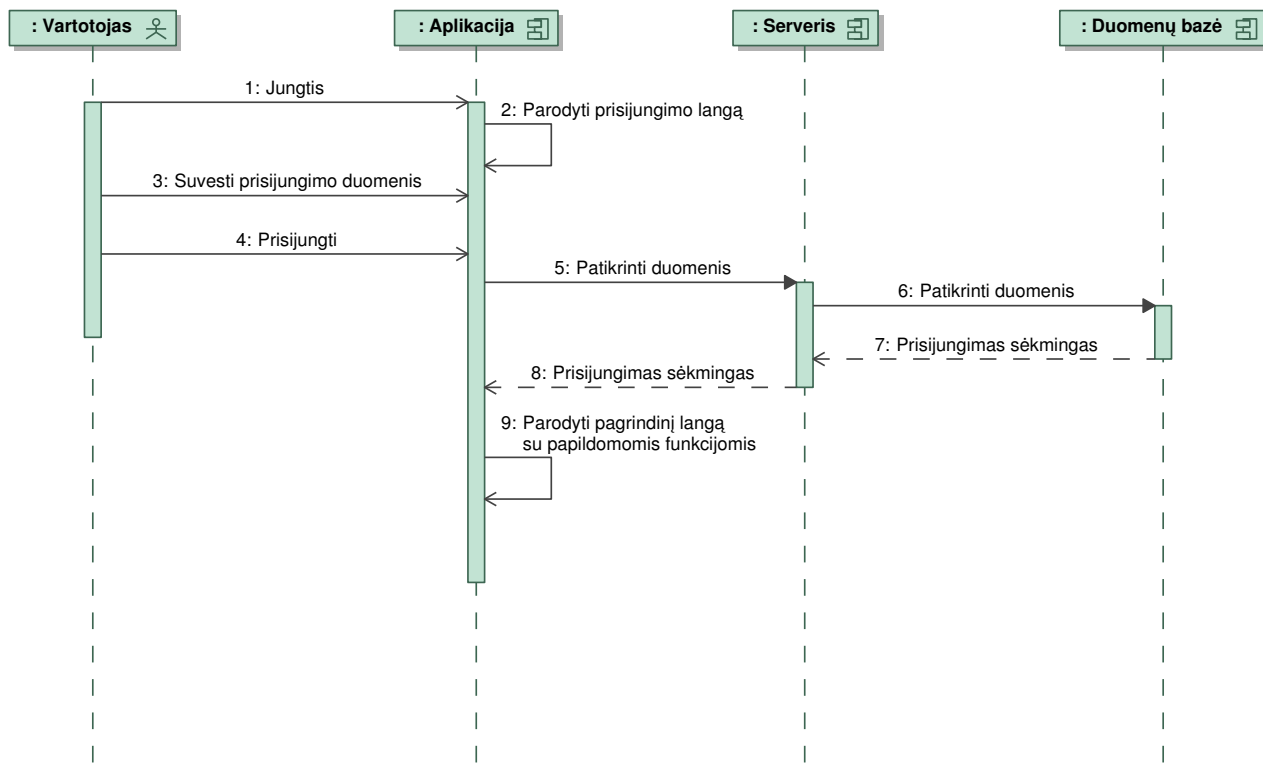


4 pav.: Užduoties „Prisiregistruoti“ sekų diagrama



### 1.3 Užduoties „Prisijungti“ įgyvendinimas

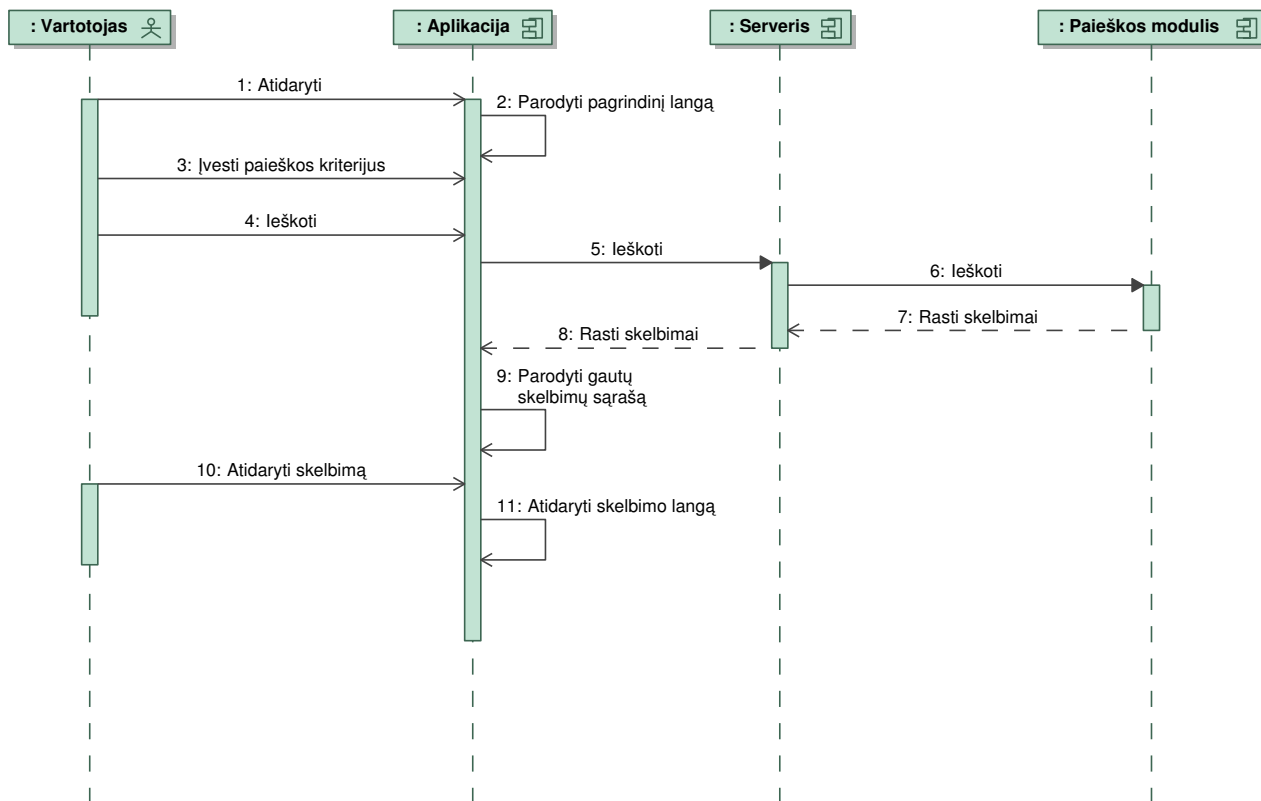
Vartotojo prisijungimo sistemoje užduoties vykdymas pavaizduotas 5 paveikslėlyje. Vartotojas, atidaręs prisijungimo langą, turi suvesti prisijungimo duomenis ir tada paspaudus prisijungimo mygtuką, duomenų bazėje yra patikrinama, ar toks vartotojas egzistuoja. Vartotojui prisijungus atsidaro pagrindinis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



5 pav.: Užduoties „Prisijungti“ sekų diagrama

## 1.4 Užduoties „Rasti skelbimą“ įgyvendinimas

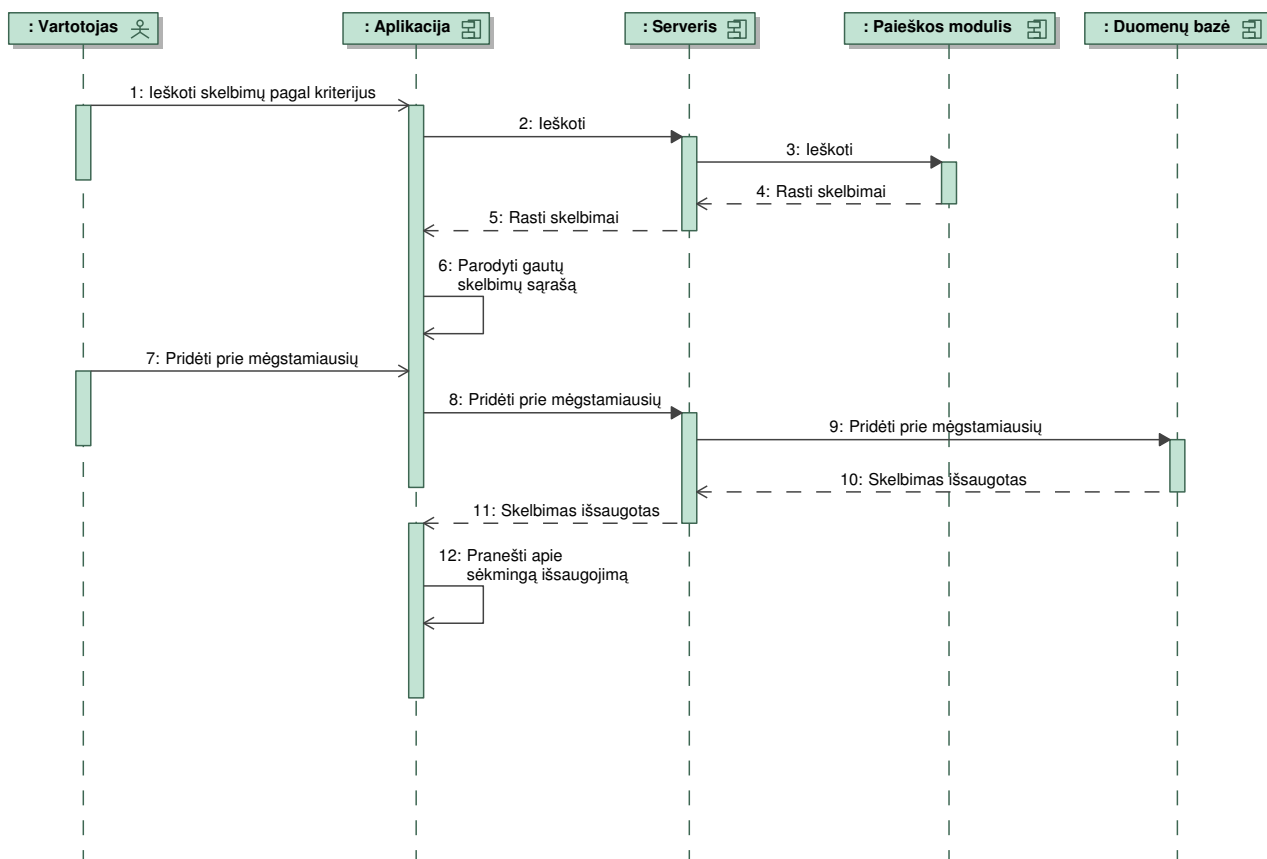
Skelbimų ieškojimo sistemoje užduoties vykdymas pavaizduotas 6 paveikslėlyje. Vartotojas nebūtinai turi būti prisijungęs, kad galėtų pasinaudoti šia sistemos funkcija. Pirmiausia vartotojas turi pagrindiniame lange įvesti paieškos kriterijus (kai kurie filtrai yra privalomi). Paspaudus paieškos mygtuką,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



6 pav.: Užduoties „Rasti skelbimą“ sekų diagrama

## 1.5 Užduoties „Išsaugoti mėgstamiausią“ įgyvendinimas

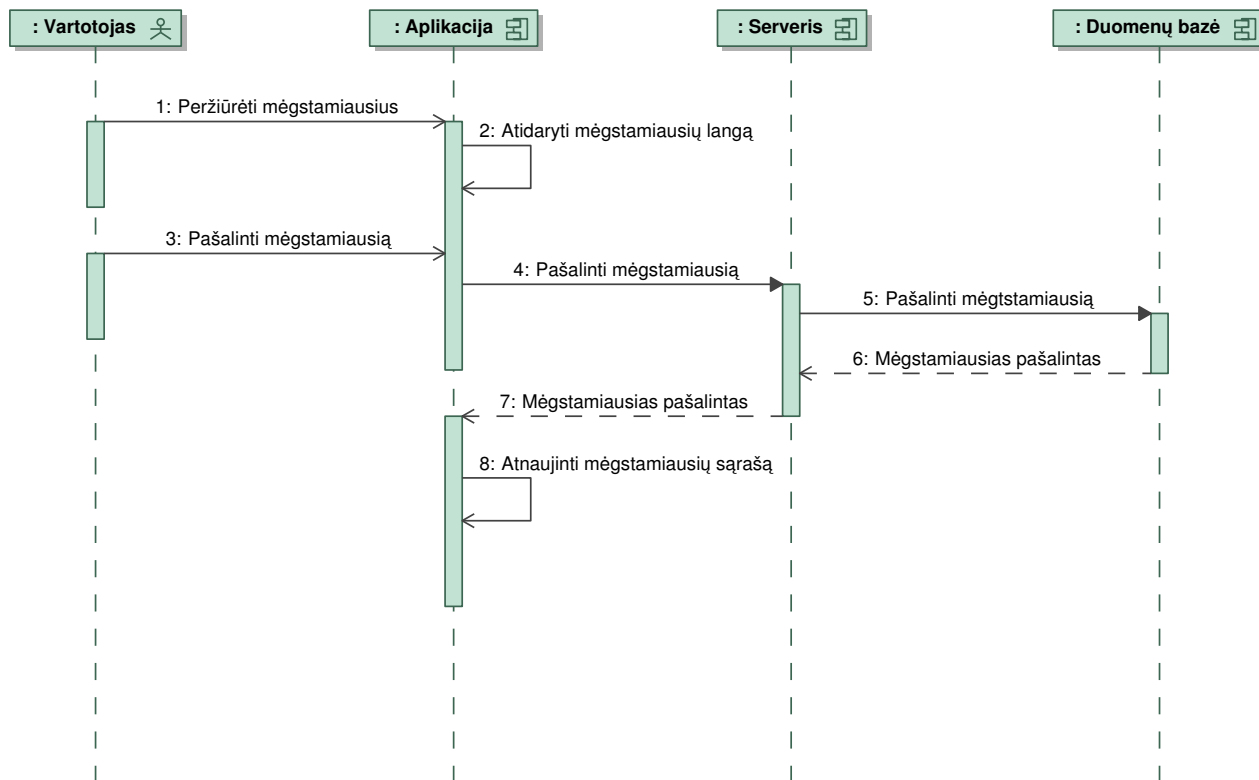
Mėgstamo skelbimo išsaugojimo užduoties vykdymas pavaizduotas 7 paveikslėlyje. Ši funkcija yra pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos. Norėdamas išsaugoti skelbimą, vartotojas pirmiausia turi susirasti skelbimą ir tada paspausti šio skelbimo išsaugojimo mygtuką. Apie sėkmingą skelbimo išsaugojimą vartotojas yra informuojamas.



7 pav.: Užduoties „Išsaugoti mėgstamiausią“ sekų diagrama

## 1.6 Užduoties „Pašalinti mėgstamiausią“ įgyvendinimas

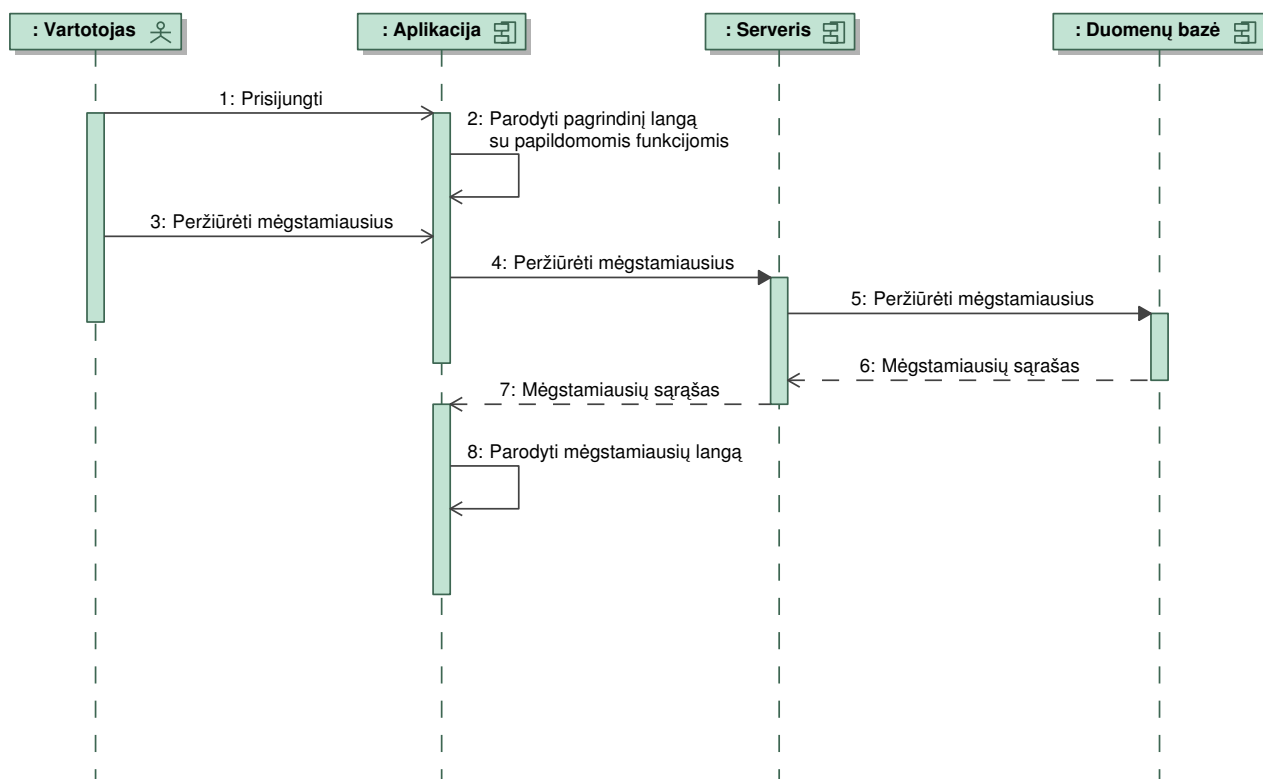
Mėgstamo skelbimo pašalinimo užduoties vykdymas pavaizduotas 8 paveikslėlyje. Ši funkcija yra pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos bei yra išsaugojęs bent vieną mėgstamą skelbimą. Pirmiausia vartotojas atsidaro mėgstamiausių skelbimų sąrašą. Tada pasirenka, kurį skelbimą jis nori pašalinti, ir paspaudžia pašalinimo mygtuką. Sėkmingai pašalinus skelbimą iš sąrašo, šis sąrašas yra atnaujinamas.



8 pav.: Užduoties „Pašalinti mėgstamiausią“ sekų diagrama

## 1.7 Užduoties „Peržiūrėti mėgstamiausius“ įgyvendinimas

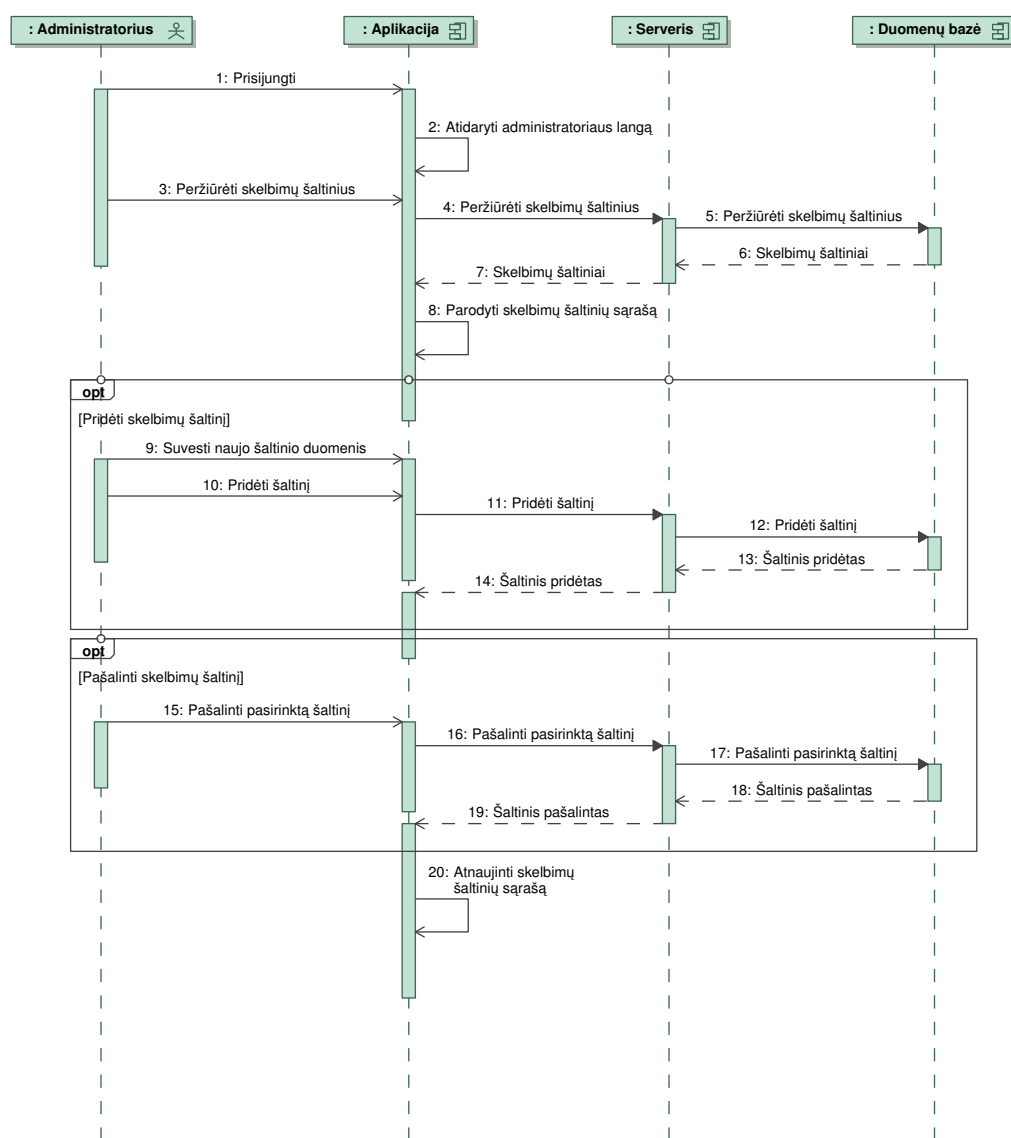
Mėgstamiausių skelbimų peržiūrėjimo užduoties vykdymas pavaizduotas 9 paveikslėlyje. Ši funkcija pasiekama tada ir tik tada, kai vartotojas yra prisijungęs prie sistemos. Vartotojui pasirinkus mėgstamiausių skelbimų langą, sistema parodo šio vartotojo mėgstamiausių skelbimų sąrašą.



9 pav.: Užduoties „Peržiūrėti mėgstamiausius“ sekų diagrama

## 1.8 Užduoties „Tvarkyti skelbimų šaltinius“ įgyvendinimas

Skelbimų šaltinių tvarkymo užduoties vykdymas yra pavaizduotas 10 paveikslėlyje. Šia funkcija gali pasinaudoti tik administratorius. Administratorius, atsidaręs skelbimų šaltinių langą, gali pasirinkti, ką jis nori su šaltiniais daryti: pridėti naują skelbimų šaltinį ar pašalinti jau pridėtą šaltinį. Norėdamas pridėti šaltinį, administratorius spaudžia mygtuką, skirtą naujam šaltiniui pridėti, o siekdamas 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.



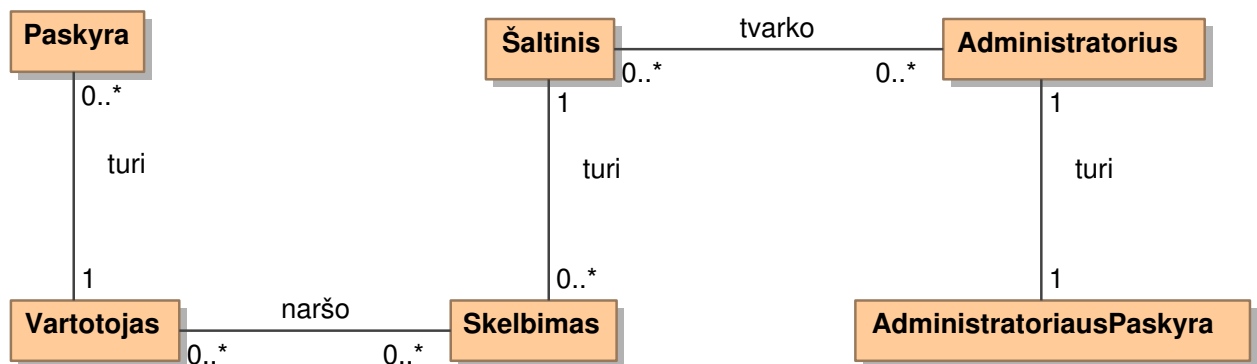
10 pav.: Užduoties „Tvarkyti skelbimų šaltinius“ sekų diagrama

## 2 Struktūrinis programų sistemos modelis

Šiame skyriuje yra pateikta sistemos klasių diagrama ir, pasitelkiant objektų diagramą, parodytas sistemos naudojimo pavyzdys.

### 2.1 Klasių diagrama

Žemiau pateiktame 11 paveikslėlyje 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.

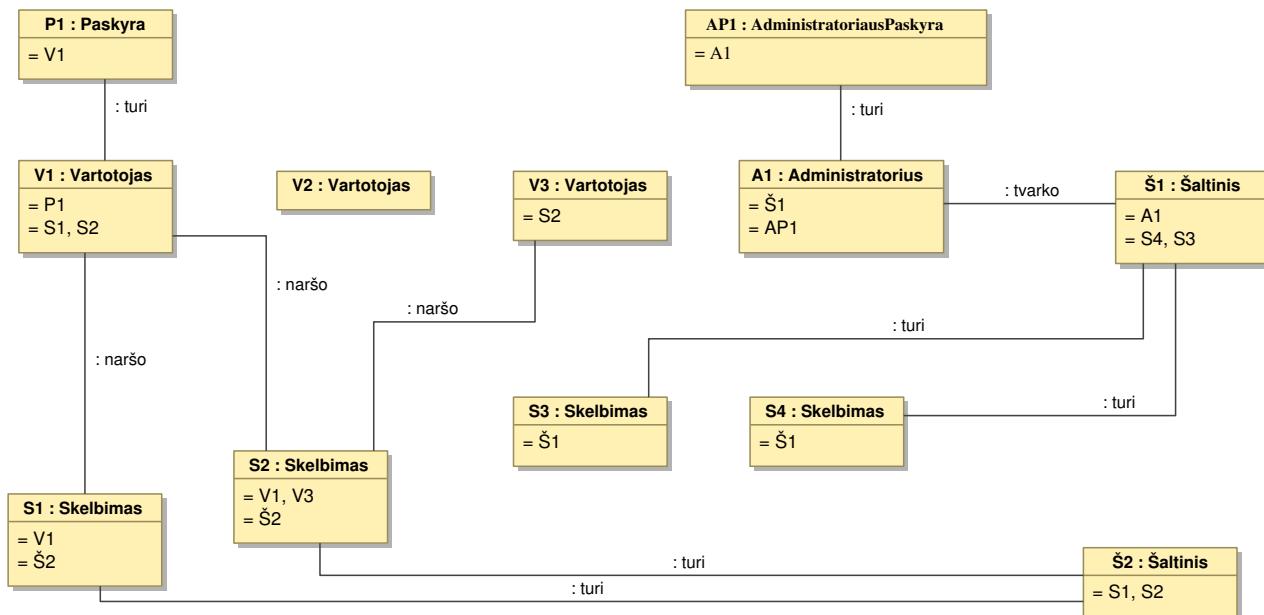


11 pav.: Klasių diagrama

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 viena paskyra priklauso tik vienam vartotojui. Administratorius gali turėti tik vieną paskyrą: vartotojui, kuriam suteikiamos administratoriaus teisės, išduodamas vienkartinis kodas, kurį įvedus sukuriamą speciali administratoriaus paskyra. 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 kiekvienas šaltiniai gali turėti daug skelbimų.

## 2.2 Objektų diagrama

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



12 pav.: Objektų diagrama

Aukščiau esančiame 12 paveikslėlyje parodyta tipinė sistemos situacija, kai yra sukurtos dvi paskyros: viena yra vartotojo, kita - administratoriaus. Taip pat yra dar du vartotojai, iš kurių vienas 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.

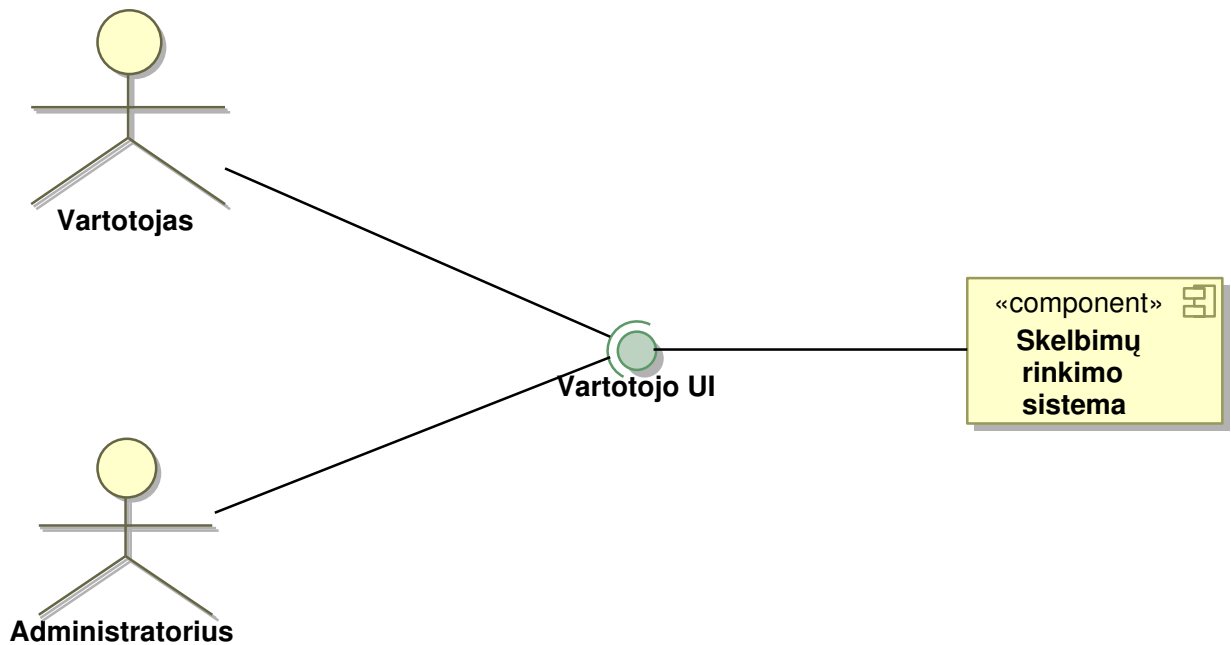


### 3 Programų sistemos komponentai

Šiame skyriuje yra pavaizduoti sistemos komponentai. Sistemos komponentų dekompozicija įgyvendinta „top-down“ būdu.

#### 3.1 Konteksto diagrama

Abstrakčiausias (aukščiausias) komponentų struktūros lygis parodomas 13 paveikslėlyje.



13 pav.: Konteksto komponentų diagrama

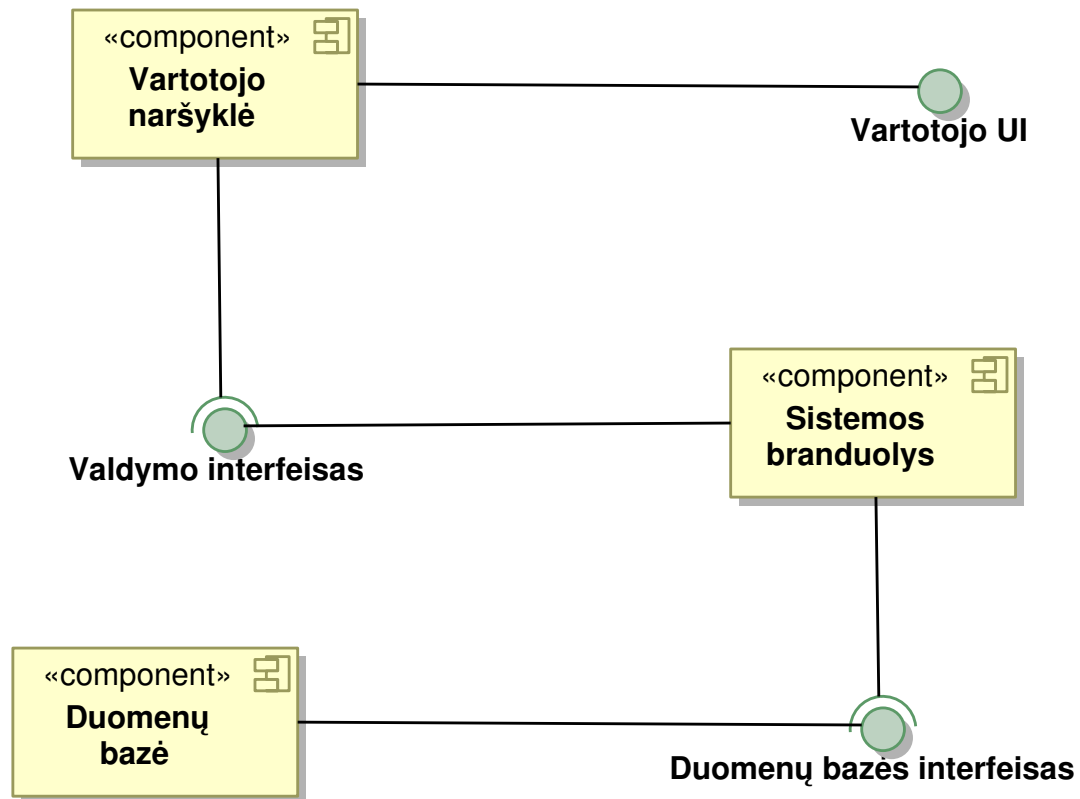
Visa sistema laikoma kaip vienas darinys ir parodoma sąsaja su išore:

- 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 ta, kad abu vartotojai naudojami ta pačia sistema, tik su šiek tiek skirtingomis galimybėmis.

## 3.2 Skelbimų rinkimo sistemos dekompozicija

Žemiau esančiame 14 paveikslėlyje detalizuojamas 13 paveikslėlyje esantis „Skelbimų rinkimo sistema“ komponentas.



14 pav.: Skelbimų rinkimo sistemos komponento dekompozicija

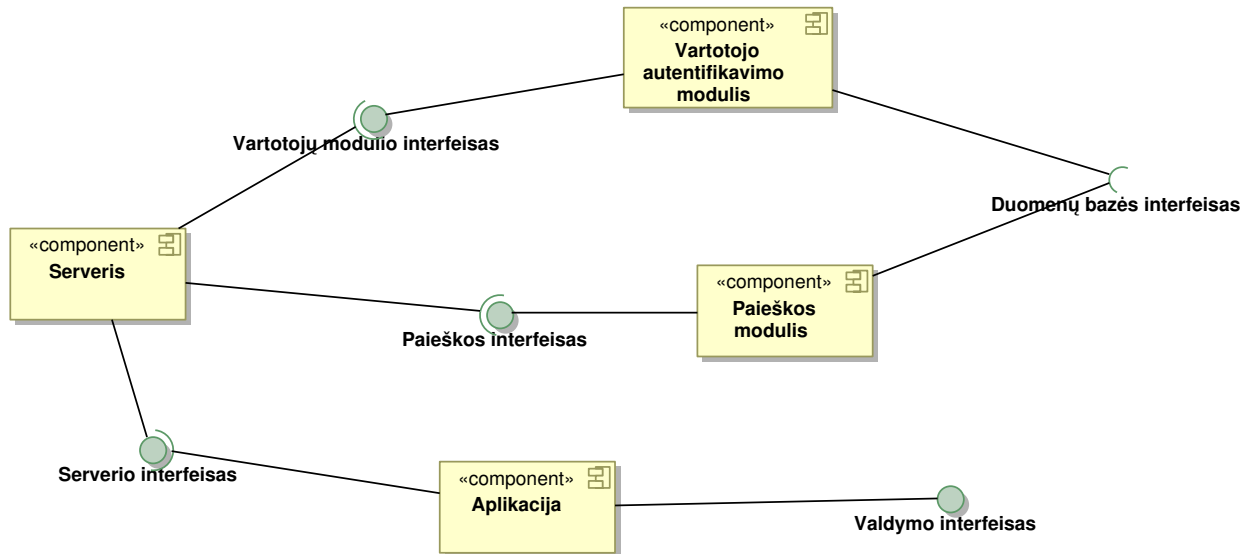
**Vartotojo naršyklė** - komponentas, sukuriantis vartotojo interfeisą bei vykdomas komunikaciją tarp vartotojo ir sistemos branduolio.

**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.

### 3.3 Sistemos branduolio dekompozicija

15 paveikslėlyje smulkinamas 14 paveikslėlyje esantis „Sistemos branduolio“ komponentas.



15 pav.: Sistemos branduolio komponento dekompozicija

Jį sudarantys komponentai:

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

**Paieškos modulis** - komponentas, pagal tam tikrus nurodytus kriterijus surandantis skelbimus.

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

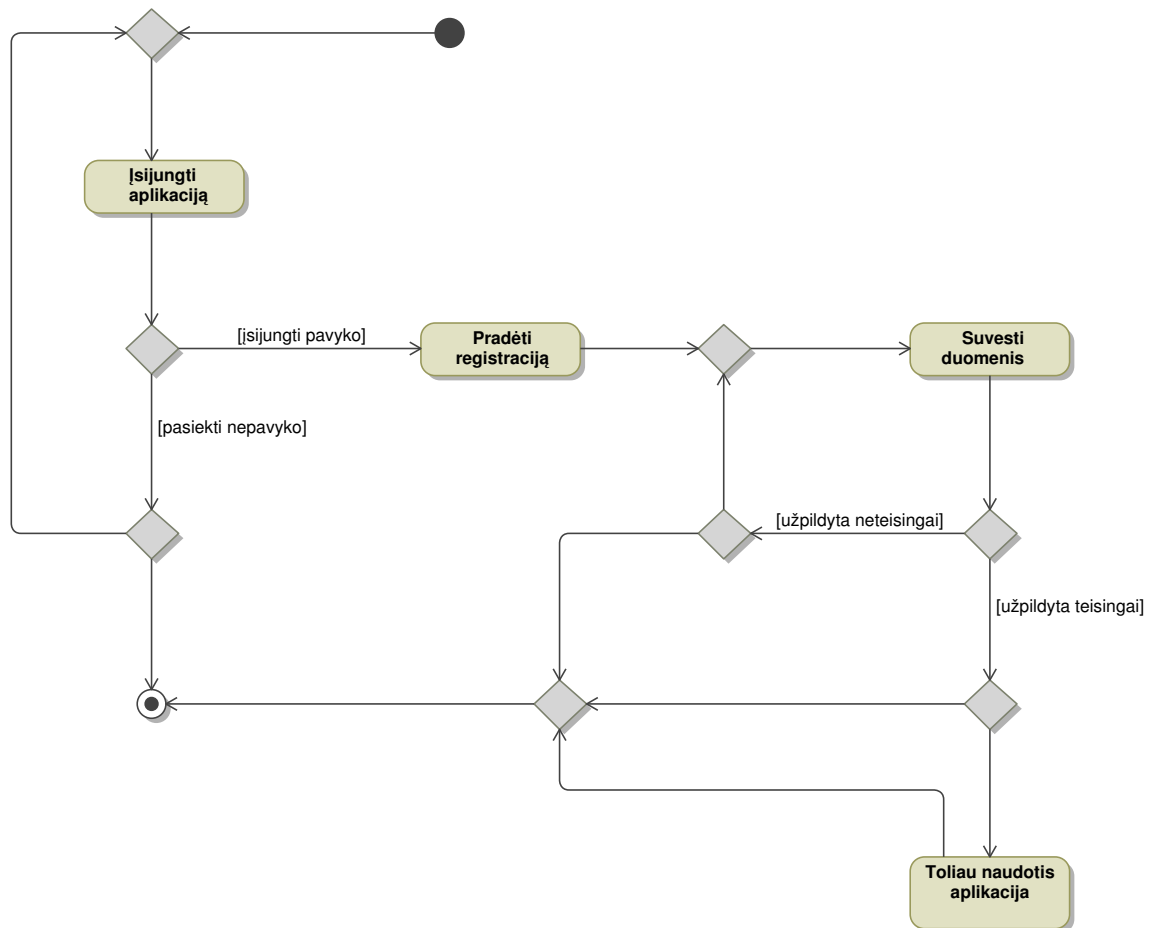
**Aplikacija** - komponentas, kuris pateikia vartotojui skelbimų sąrašą, konkrečių skelbimų detalesnę informaciją.

## 4 Dinaminis programų sistemos modelis

Šiame skyriuje parodoma, kaip vyksta šios sistemos veiksmai bei aprašytos kai kurios galimos vartotojų užduotys.

### 4.1 Vartotojo registracijos veiklos diagrama

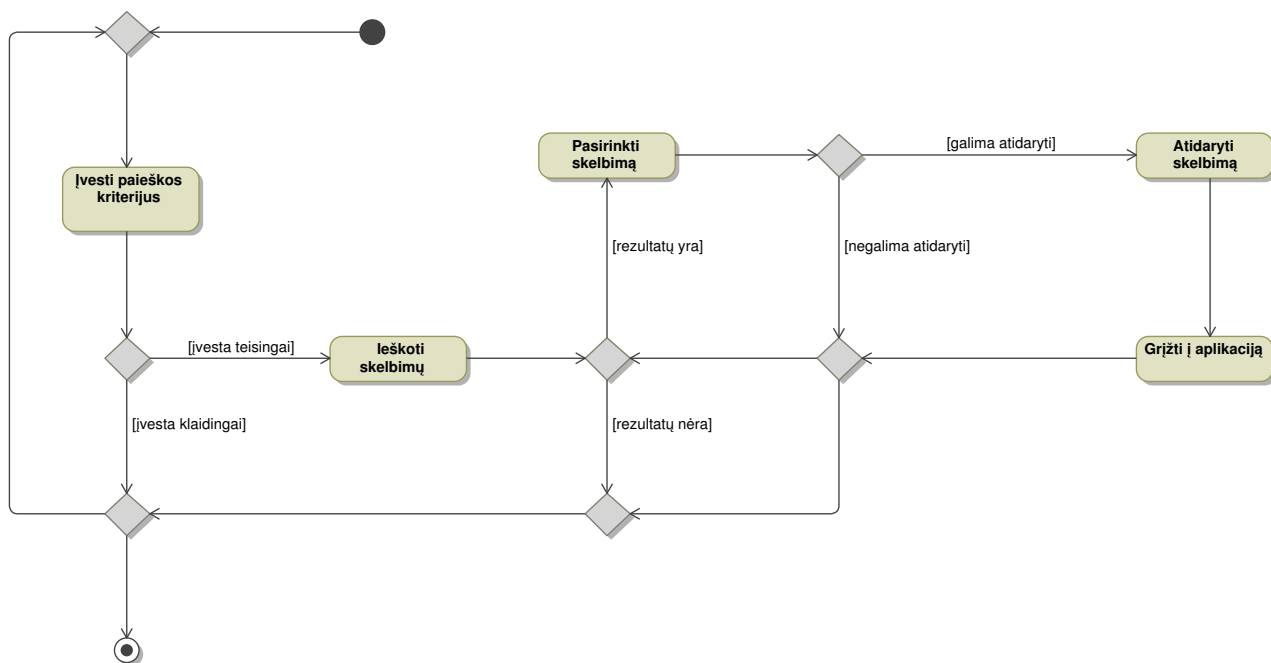
Procesai, vykstantys vartotojo registracijos metu, parodyti 16 paveikslėlyje. Registravimas vyksta tada, kai vartotojas įsijungia aplikaciją ir pasirenka „Registracija“. Tada vartotojui suvedus reikiamus duomenis sistema patikrina, ar viskas užpildyta teisingai. Jei ne, vartotojas gali bandyti vesti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



16 pav.: Vartotojo registracijos veiklos diagrama

## 4.2 Skelbimų paieškos veiklos diagrama

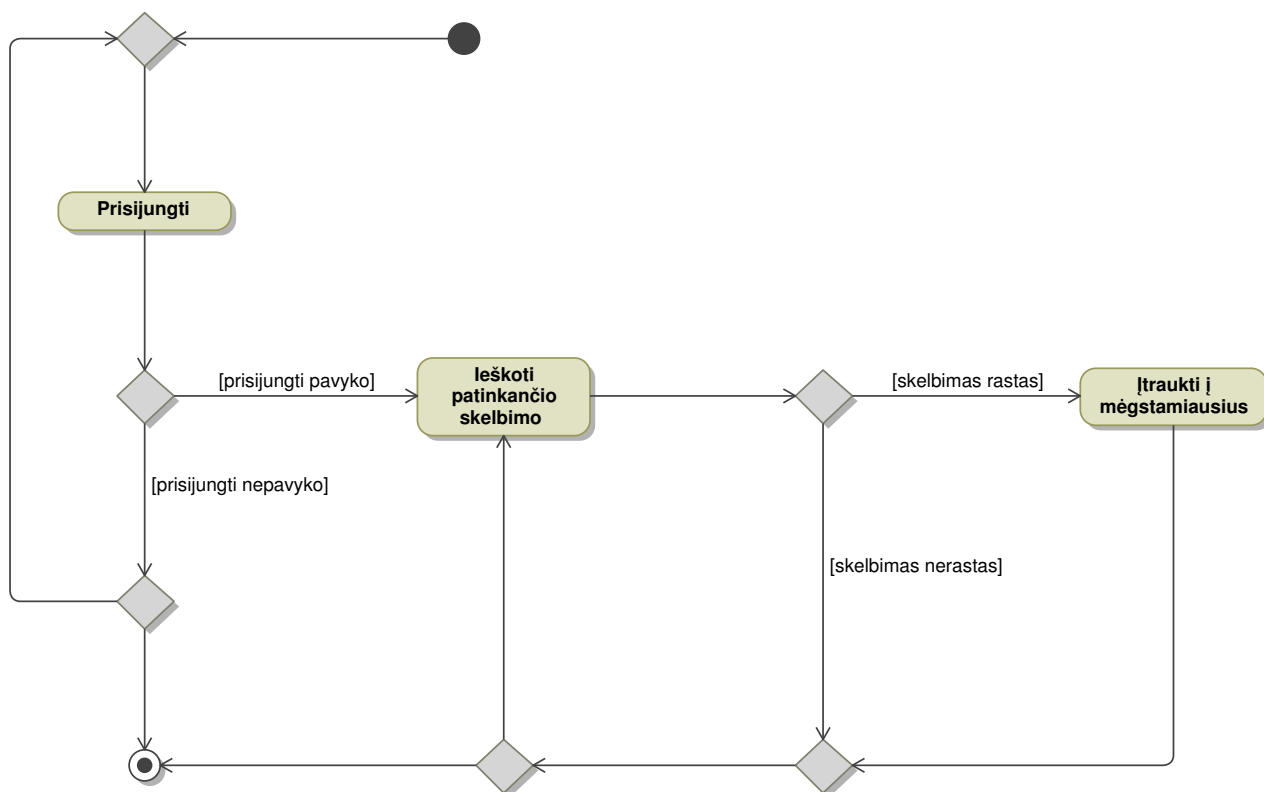
Procesai, vykstantys skelbimo ieškojimo metu, parodyti 17 paveikslėlyje. Pagrindiniame lange vartotojas turi įvesti filtrus, pagal kuriuos bus ieškomi skelbimai. Sistema patikrina, ar įvesti kriterijai yra teisingai suvesti. Jei įvesta klaidingai, vartotojai gali bandyti iš naujo suvesti paieškos kriterijus arba išeiti iš programos. Jei paieškos kriterijai įvesti teisingai, sistema ieško skelbimų. Jei rezultatų nėra, 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, vartotojas gali atidaryti skelbimą ir po to grįžti į aplikaciją.



17 pav.: Skelbimų paieškos veiklos diagrama

### 4.3 Naujo mėgstamo skelbimo veiklos diagrama

Procesai, vykstantys vartotojui norint įtraukti skelbimą į mėgstamiausių sąrašą, parodyti 18 paveikslėlyje. Pirmiausia vartotojas turi prisijungti. Jei to padaryti nepavyksta, jis gali bandyti iš naujo arba išeiti iš programos. Prisijungęs prie sistemos vartotojas ieško jam patinkančio skelbimo ir jį suradęs 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.



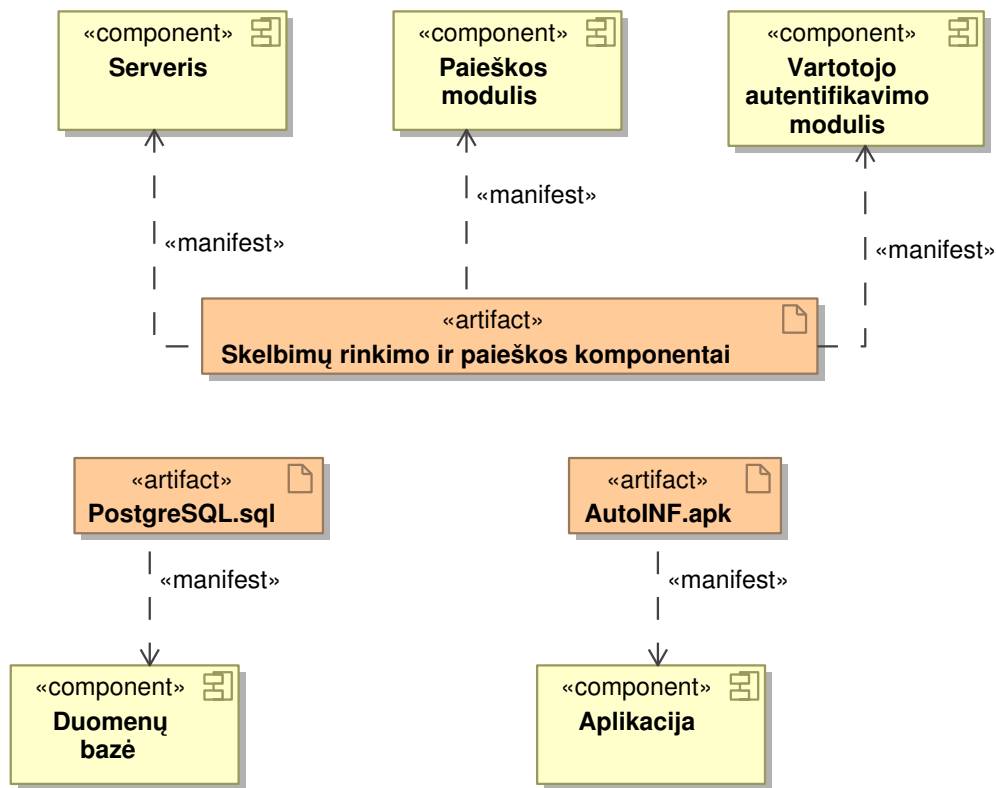
18 pav.: Naujo mėgstamo skelbimo veiklos diagrama

## 5 Komponentų išskirstymas tinkle

Šiame skyriuje parodytas sistemos komponentų išdėstymas tinkle.

### 5.1 Komponentų ryšių su artifaktais diagrama

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

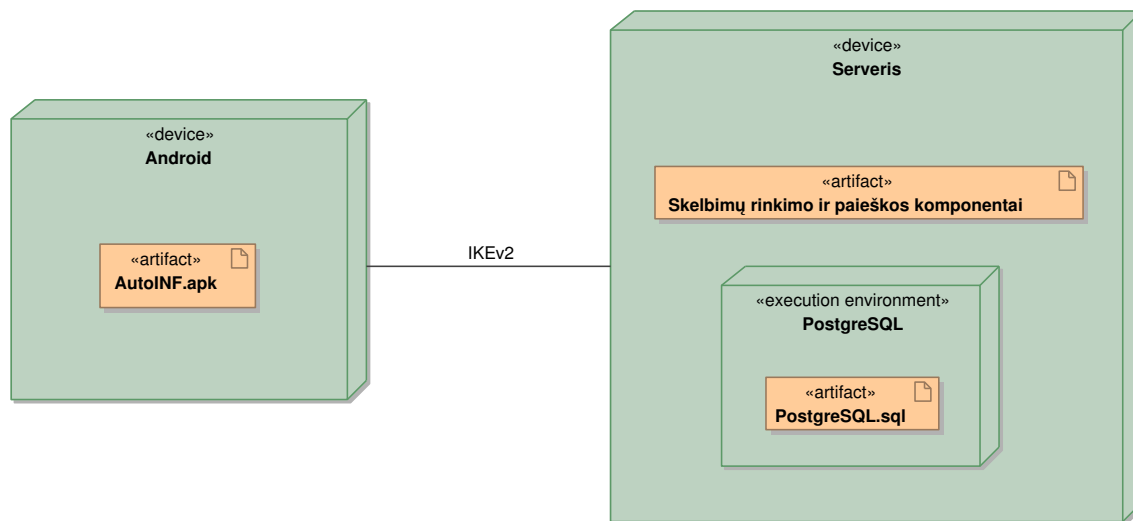


19 pav.: Komponentų ryšių su artifaktais diagrama

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

## 5.2 Diegimo diagrama

Žemiau pateiktame 20 paveikslėlyje yra išskirti fiziniai įrenginiai, reikalingi sistemos darbo palaikymui, bei artifaktų pasiskirstymas tarp jų.



20 pav.: Diegimo diagrama

Beveik visa sistema yra laikoma serveryje. Šiame serveryje yra sudiejami skelbimų rinkimo ir paieškos komponentai. Tame pačiame serveryje yra ir PostgreSQL duomenų bazė. Vartotojui, norinčiam naudotis sistema iš mobiliojo įrenginio, kuriame yra likusi sistemos dalis, siunčiama užklausa IKEv2 protokolu siekiant gauti sistemos failus.



## Rezultatai

Šiame laboratoriniame darbe pasitelkiant skirtingus sistemos pjūvius aprašyta knygų dalinimosi sistemos architektūra. Užduočių pjūvyje pasitelkiant užduočių ir sekų diagramomis išaiškinti pagrindiniai agentų (vartotojų ir administratorių) tikslai naudojantis sistema. Loginis pjūvis, pavaizduotas klasių ir objektų diagramomis, leido išskirti pagrindines esybes bei ryšius tarp jų. Kūrimo pjūvyje atlikta sistemos dekompozicija pradedant nuo bendro komponento toliau jį detalizuojant. Procesų pjūvyje, pasitelkiant veiklos diagramas, išskirti procesai, jų komunikacija, objektų gyvavimo ciklai. Galiausiai fiziniame pjūvyje apibrėžtas sistemos išdėstymas tinkle. Šis skirtingų požiūrių rinkinys leido aptikti ankčiau sukurtame sistemos prototipe buvusias klaidas bei išplėtoti tinkamą sistemos architektūrą.

## Išvados

Remiantis rezultatais bus galima sukurti mobiliems įrenginiams skirtą programėlę, kuri leis vienoje vietoje pasiekti automobilių skelbimus iš visos Europos, taip palengvindama paieškos procesą daugeliui žmonių.

## Terminų žodynas

**Vartotojas** - žmogus, besinaudojantis sistema.

**Administratorius** - vartotojas, kuriam suteikiamos teisės tvarkyti šaltinius.

**Skelbimas** - iš skelbimų šaltinio gautas sistemoje patalpintas skelbimas.

**Šaltinis** - internetinis skelbimų puslapis, iš kurio gaunami skelbimai naudojami sistemoje.

**Megstamas / mėgstamiausias** - skelbimas, kurį vartotojas išsaugo galimam lengvam ir greitam vėlesniam pasiekiamumui.

## Literatūros sąrašas

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