

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

## **Kavinės staliuko rezervavimo aplikacija**

### **Cafe table rezervation app**

Programų sistemų inžinerijos I laboratorinis darbas Nr. 1

Atliko:	2 kurso 5 grupės studentai	
	Paulius Grigaliūnas	(parašas)
	Karolis Staskevičius	(parašas)
	Modestas Dulevičius	(parašas)
	Albert Jurkoit	(parašas)
	Šarūnas Kazimieras Buteikis	(parašas)
Darbo vadovas:	dr. Vytautas Valaitis	(parašas)

# ANOTACIJA

**Darbo tikslas:** sukurti išmanų, patogų kavinių staliukų rezervavimo programėlės modelį, kuris funkcionuotų Windows ir Android sistemose. Taip pat siekiama, kad galutinė programėlė užtikrintų sklandų, spartų komunikabilumą tarp klientų ir kavinės darbuotojų, suteikiant galimybę kavinių savininkams pateikti išsamų kavinės planą, o klientams išsirinkti norimą staliuką kavinėje patieks.

## **Darbą atliko:**

Paulius Grigaliūnas  
paulius.grigaliunas.pg@gmail.com

Karolis Staskevičius  
satelistas@gmail.com

Modestas Dulevičius  
modux9@gmail.com

Albert Jurkoit  
albert.jurkoit@mif.stud.vu.lt

Šarūnas Kazimieras Buteikis  
sarunas.kazimieras.buteikis@gmail.com

## TURINYS

ANOTACIJA .....	2
ĮVADAS .....	4
1. PROGRAMŲ SISTEMOS ARCHITEKTŪRA .....	5
1.1. Struktūrinis programų sistemos modelis (angl. Logical view) .....	5
1.1.1. Klasijų diagrama .....	5
1.2. Užduotys ir jų vykdymo scenarijai (angl. Use-cases) .....	9
1.2.1. Sistemos vykdomos užduotys .....	9
1.2.2. Užduoties „Prisiregistruoti prie sistemos“ vykdymas .....	9
1.2.3. Užduoties „Pridėti naują restoraną“ vykdymas .....	10
1.2.4. Užduoties „Ieškoti restoranų“ vykdymas .....	11
1.2.5. Užduoties „Pakeisti restorano informaciją“ vykdymo scenarijus .....	12
1.3. Dinaminis programų sistemos modelis (angl. Process view) .....	14
1.3.1. Veiklos diagramos .....	14
1.4. Programų sistemos išskirstymas tinkle(angl. Deployment view) .....	16
1.4.1. Komponentų ryšių su artefaktais diagrama .....	16
1.4.2. Mazgų diagrama .....	17
1.5. Programų sistemos kūrimo pjūvis (angl. Developement view) .....	18
1.5.1. Konteksto diagrama .....	19
1.5.2. Subsistemų dekompozicija .....	20
1.5.2.1. Kliento aplikacija .....	21
1.5.2.2. Kavinės valdymas .....	21
1.5.2.3. Duomenų bazė .....	22
IŠVADOS .....	23
TERMINŲ ŽODYNAS .....	24

# **Įvadas**

## **”Book a Table” kavinės rezervavimo aplikacija**

### **Dalykinė sritis**

Kavinės ir jų rezervacija

### **Probleminė sritis**

Lietuvoje staliuko rezervavimo galimybės yra mažai išplėtos

### **Naudotojai**

Žmonės, norintys skaniai pavalgyti, bei iš anksto pasirūpinti vietą restorane.

Kavinių savininkai, suteikiantys žmonėms galimybę rezervuoti staliuką jų restorane.

### **Darbo pagrindas**

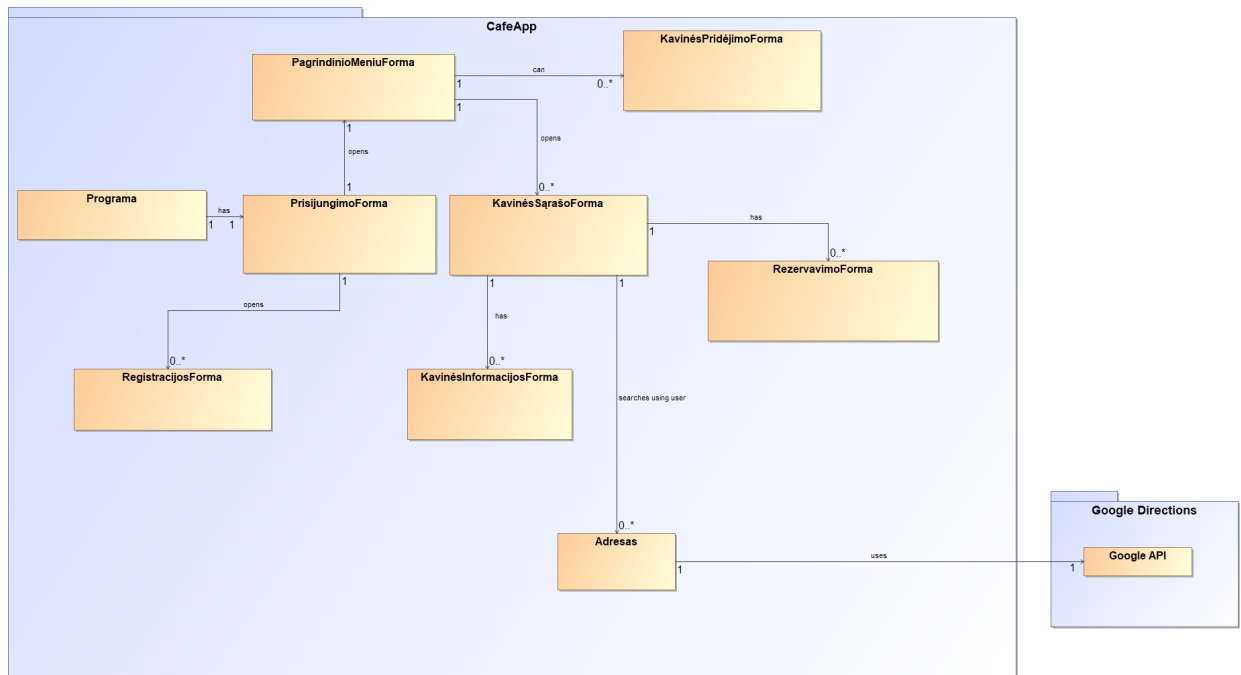
Dokumentas parengtas kaip programų sistemų inžinerijos dalyko laboratorinis darbas Nr. 1, kuriame pateikiamas suprojektuotos sistemos aprašymas.

# 1. Programų sistemos architektūra

## 1.1. Struktūrinis programų sistemos modelis (angl. Logical view)

### 1.1.1. Klasių diagrama

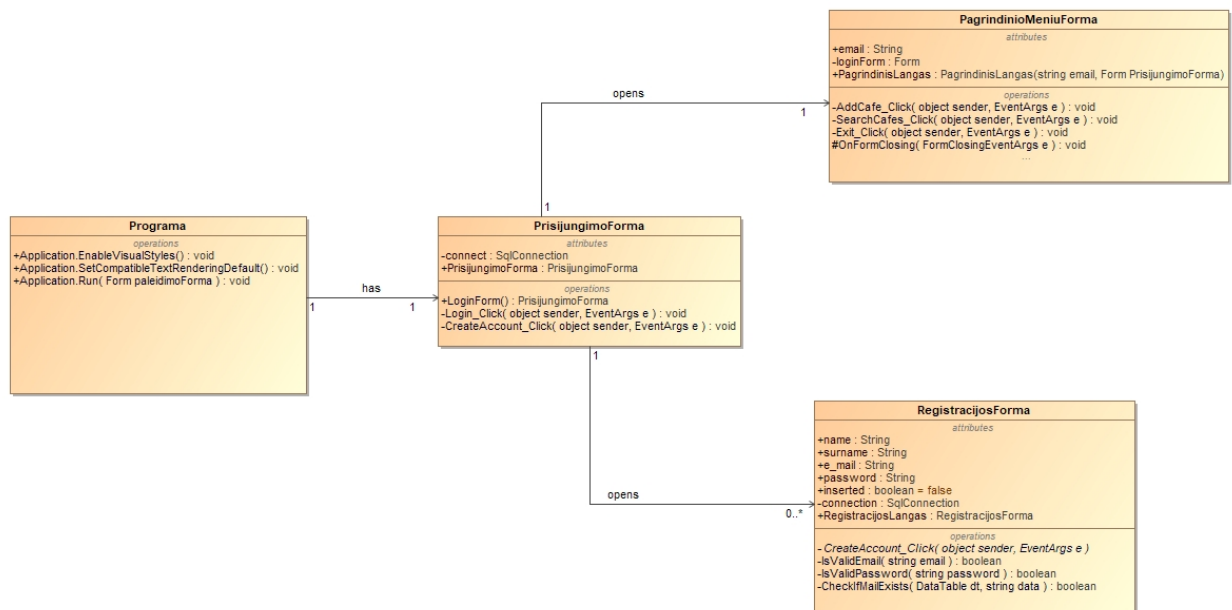
Žemiau pateiktoje klasių diagramoje (1 pav.) yra išskirtos pagrindinės esybės, kurios yra naudojamos sistemoje. Klases siejantys ryšiai pasižymi kardinalumu, t.y. nustatytas konkretus ryšių skaičius, kuriuos turės klasės egzempliorius su kitais klasės egzemplioriais.



1 pav. Klasių diagrama

Pateiktoje diagramoje yra visos programos sistemos modelis. Programa galima suskirstyti į 3 dalis: vartotojo prisijungimas/registracija prie aplikacijos, kavinių registravimas bei registruotų kavinių sąrašas ir kavinių paieška, rezervavimas ir kavinės informacijos modifikavimas. Apie jas bus plačiau aprašoma kitose klasių diagramose.

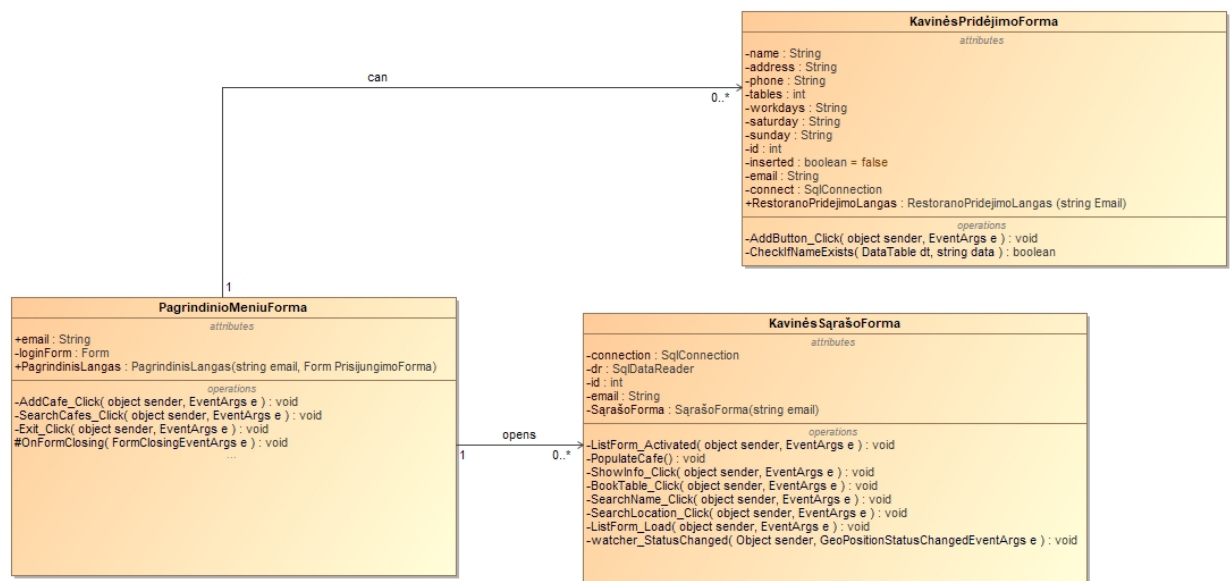
Paleidus programą (2 pav.) vartotojas gali prisijungti arba sukurti naują paskyrą. Mes nusprendėme, kad vartotojui, nuėjus į naujos paskyros langelį nedingtų pradinis langelis. Tokiu būdu vartotojui užsiregistravus bus galima iš karto prisijungti ir atsidurti mūsų programos pagrindiniame langelyje arba sukurti naują paskyrą, jeigu jis būtų nepatenkintas esama paskyra.



2 pav. Programos paleidimo ir vartotojo prisijungimo ir registracijos klasių diagrama

Kuriant naują paskyrą, privaloma įvesti paštą ir slaptažodį. Bus patikrinama ar įvesti duomenys yra korektiški, taip pat bus patikrinama ar jau nėra tokios sukurtos paskyros su įvestais duomenimis. Prisijungimo metu tikrinama ar yra tokia sukurta paskyra.

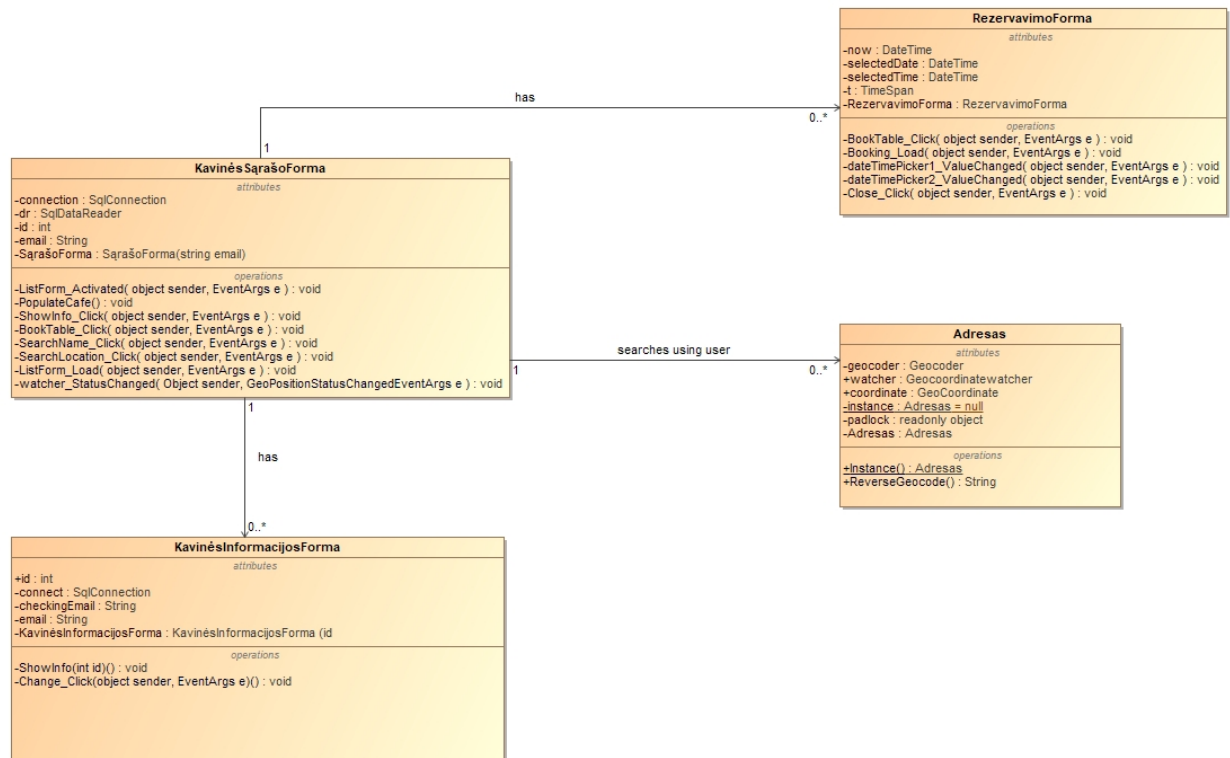
Žemiau pateiktoje klasių diagramoje (3 pav.) pavaizduotos klasės, susijusios su pagrindiniu programos langeliu. Šiame langelyje galima pridėti kavinę į kavinių sąrašą arba atsiverti kavinių sąrašą. Norint pridėti kavinę privaloma nurodyti kavinės pavadinimą, adresą, staliukų skaičių, tvarkaraštį (nuo kada iki kada dirba darbo dienomis, savaitgaliais) ir vartotojo telefono numerį.



3 pav. kavinių registravimo ir registruotų kavinių sąrašo klasių diagrama

Kavinės registravimo metu yra patikrinama ar yra kavinė su tokia pačia informacija, kad būtų išvengta dubliavimo.

4 pav. klasių diagramoje parodomas kavinių paieška ir kavinės rezervavimas. Mes nusprendėme leisti vartotojui ieškoti norimos kavinės pagal kavinės vardą arba pagal vartotojo esamą vietovę. Taip palengvinama kavinės paieška, jeigu vartotojas žino, jog yra šalia kavinės, bet nežino jos pavadinimo, arba žino kavinės pavadinimą, bet nežino kur ji randasi. Jeigu vartotojas yra ir registruotas kavinės savininkas, jis gali pakeisti jos vardą, adresą, staliukų skaičių, telefono numerį. Tokiu būdu pataisoma klaidinga registruotos kavinės informacija.



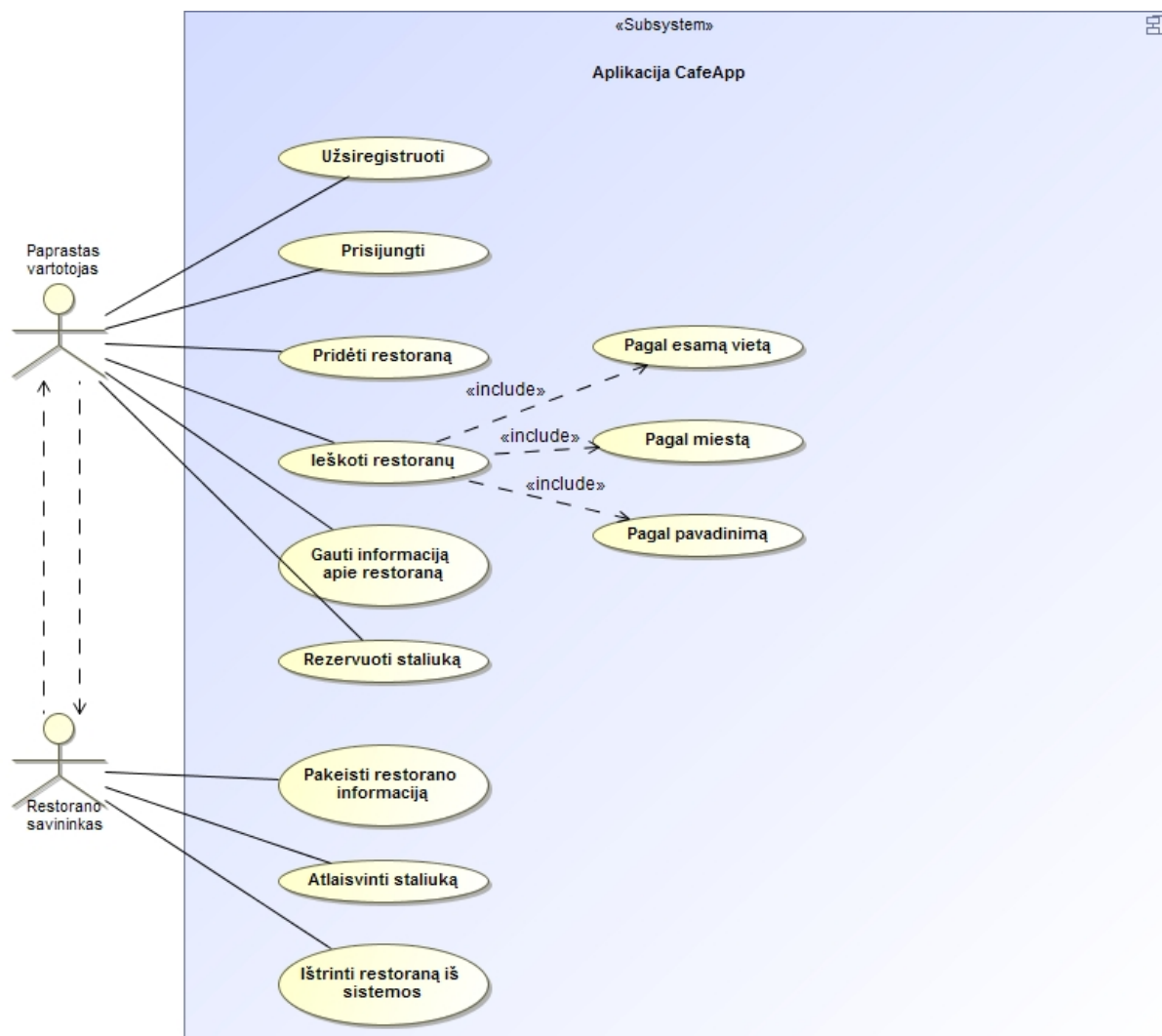
4 pav. kavinių paieškos ir kavinės rezervavimo klasių diagrama



## 1.2. Užduotys ir jų vykdymo scenarijai (angl. Use-cases)

### 1.2.1. Sistemos vykdomos užduotys

Sistema besinaudojantis vartotojas gali atlikti žemiau pateiktas užduotis. Nusprendėme užduotis išskirstyti į „Paprasto vartotojo“ ir „Restorano savininko“, nes šie agentai turi galimybę atlikti skirtingas užduotis. Restorano savininkas taip pat yra paprastas vartotojas, tačiau savininko statusas leidžia jam tvarkyti tik savo pridėtus restoranus (pvz.: pakeisti restorano informaciją). Paprastas vartotojas taip pat gali tapti restorano savininku, jei prideda savo restoraną. Tai atsispindi žemiau pateiktoje sistemos užduočių diagramoje.

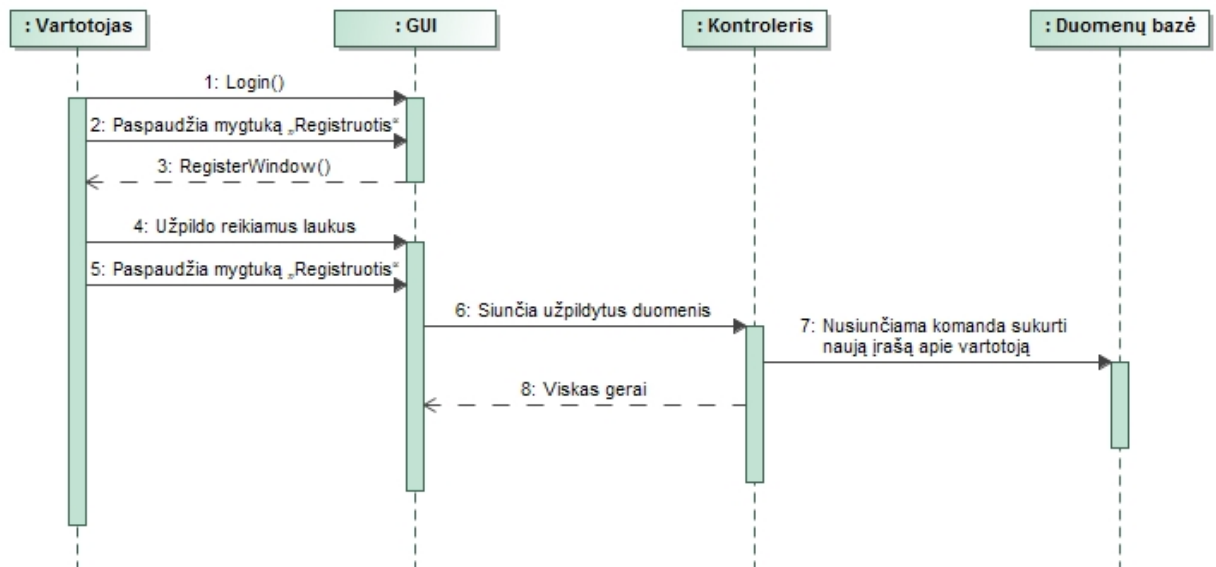


5 pav. Sistemos užduočių diagrama

### 1.2.2. Užduoties „Prisiregistruoti prie sistemos“ vykdymas

Užduoties „Prisiregistruoti prie sistemos“ sekų diagrama. Vartotojas, naudodamasis GUI, paspaudžia mygtuką „Registruotis“ ir juo iškviečia registracijos formą, kurioje užpildo reikalingus laukus. Duomenys siunčiami kontrolieriui, kuris patikrina ar jie tvarkingi (pvz.: ar egzistuoja toks elektroninio pašto adresas), ir iš kontrolierio siunčiama komanda sukurti duomenų bazėje naują

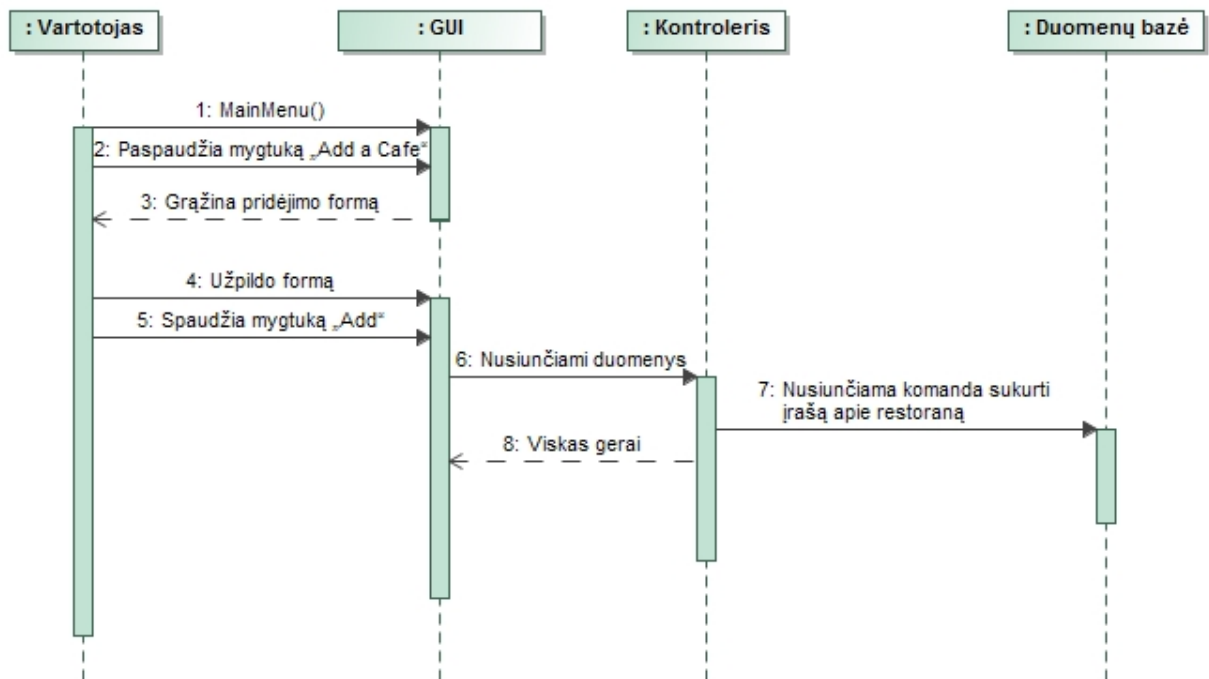
įrašą apie vartotoją.



6 pav. Užduoties „Prisiregistruoti prie sistemos“ sekų diagrama

### 1.2.3. Užduoties „Pridėti naują restoraną“ vykdymas

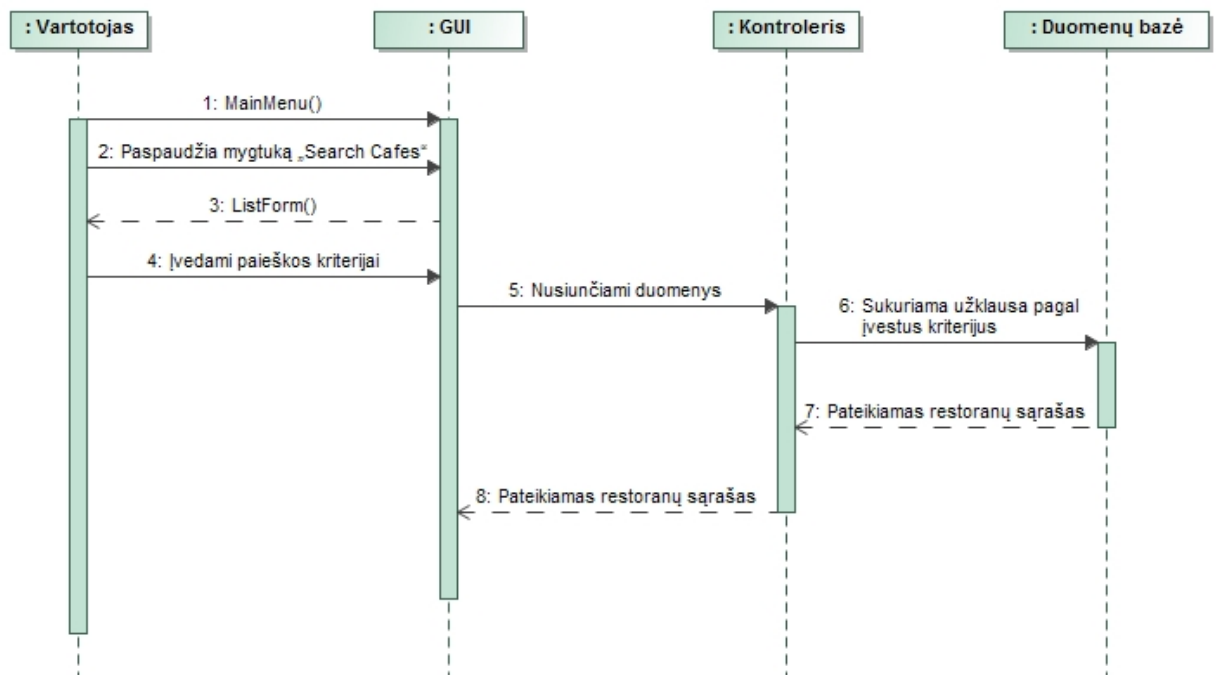
Žemiau pateiktoje sekų diagramoje pavaizduotas užduoties „Pridėti naują restoraną“ vykdymas. Vartotojas spaudžia mygtuką „Add a Cafe“ taip iškviesdamas GUI formą, kurią užpildo. Tada paspaudžia mygtuką „Add“ ir duomenys yra validuojami kontroleryje. Jeigu jie neatitinka nustatytų reikalavimų (pvz.: trūksta būtinų užpildyti laukų), vartotojui pranešama ir prašoma pataisyti duomenis. Priešingu atveju, siunčiama užklausa į duomenų bazę, kuri sukuria naują įrašą, apie pridėtą restoraną, o vartotojas automatiškai tampa restorano savininku.



7 pav. Užduoties „Pridėti naują restoraną“ sekų diagrama

#### 1.2.4. Užduoties „Ieškoti restoranų“ vykdymas

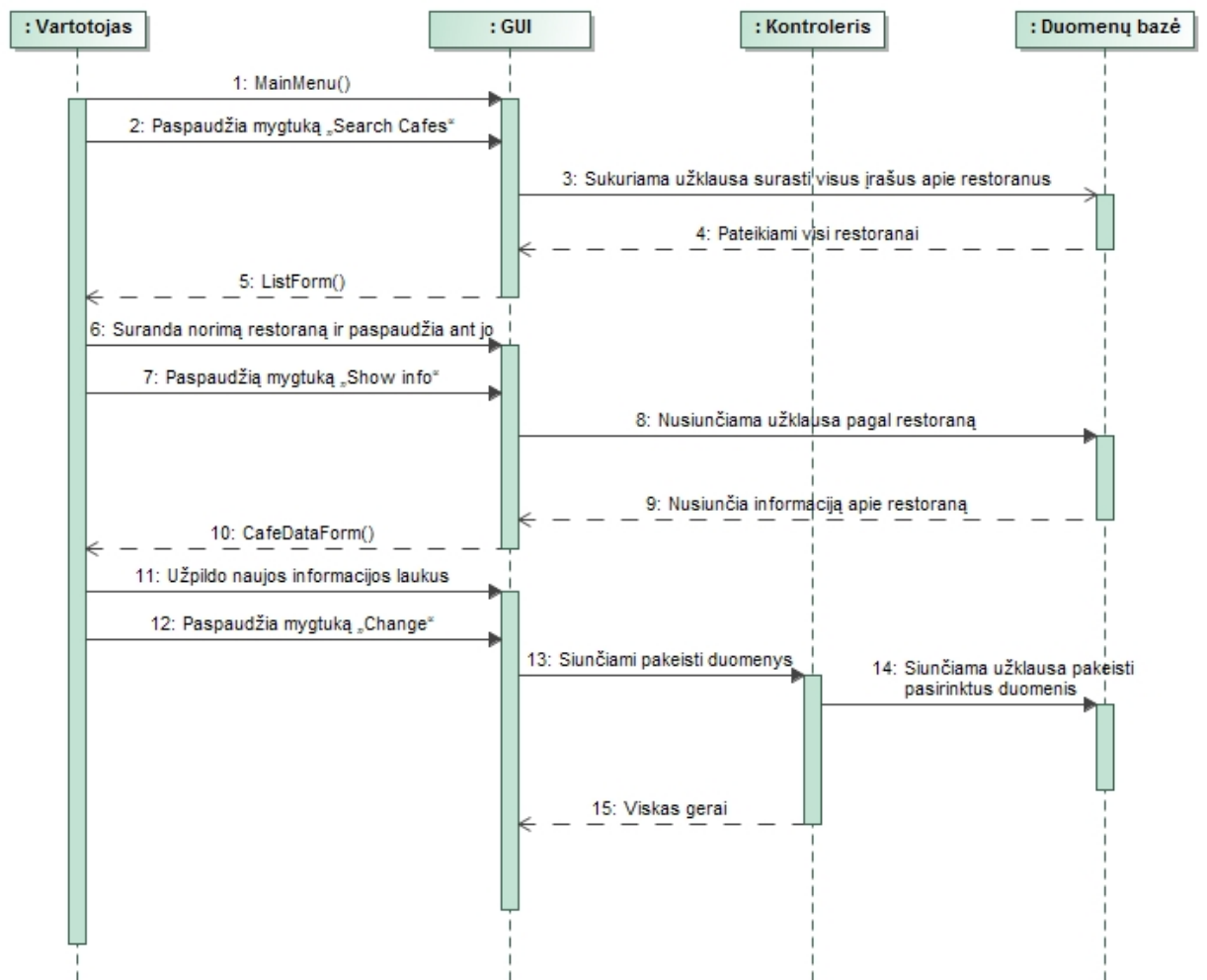
Užduoties „Ieškoti restoranų“ sekų diagrama. Vartotojas paspaudžia mygtuką „Search Cafes“ ir iškviečia naują duomenų formą, kurioje užpildo paieškos kriterijus. Yra numatyti trys kriterijai: pagal esamą vietą, pagal miestą/adresą ir pagal restorano pavadinimą. Duomenys siunčiami kontrolieriui, kuris patikrina pateiktus kriterijus, jeigu reikia, nustato esamą vietą, paprašydamas įjungti įrenginio lokaciją. Jeigu viskas yra gerai, siunčiama paieškos užklausa į duomenų bazę ir ji, naudodamasi GUI, pateikia restoranus pagal pasirinktus paieškos laukus. Tai atsispindi žemiau pavaizduotoje sekų diagramoje.



8 pav. Užduoties „Ieškoti restoranų“ sekų diagrama

#### 1.2.5. Užduoties „Pakeisti restorano informaciją“ vykdymo scenarijus

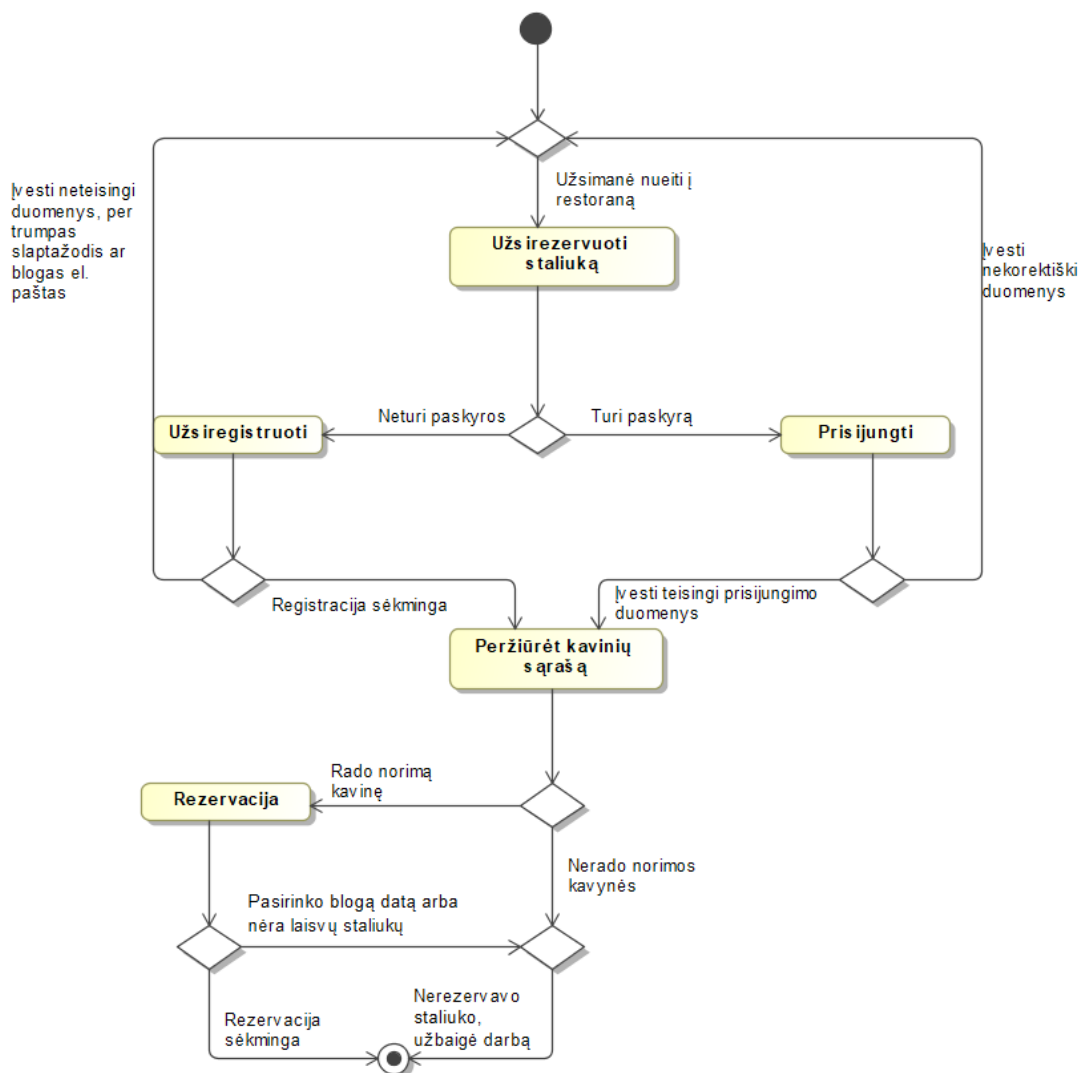
Žemiau pavaizduota užduoties „Pakeisti restorano informaciją“ sekų diagrama. Restorano savininkas paspaudžia mygtuką „Search Cafes“ ir juo GUI kreipiasi į duomenų bazę, kuri pateikia visus užregistruotus restoranus. Vykdydamas užduotį „Ieškoti restoranų“ savininkas susiranda savo restoraną, jį pažymėdamas kairiuoju pelės klavišu. Tada paspaudžia mygtuką „Show Info“ ir GUI išskviečia naują formą, kuri kreipiasi į duomenų bazę pagal restorano „ID“ ir parodo informaciją apie restoraną. Joje yra tušti laukai, kuriuos užpildydamas vartotojas gali pakeisti tam tikrą restorano informaciją. Mygtuku „Change“ vartotojas kreipiasi į kontrolerį, kuris siunčia užklausą į duomenų bazę, patikriną, ar būtent šis vartotojas sukūrė įrašą apie restoraną, ir jeigu tai tiesa, vėl kreipiamasi į duomenų bazę - atnaujinti pakeistus duomenis. Jeigu vartotojas neturi teisės keisti informacijai, jam apie tai pranešama.



9 pav. Užduoties „Pakeisti restorano informaciją” sekų diagrama

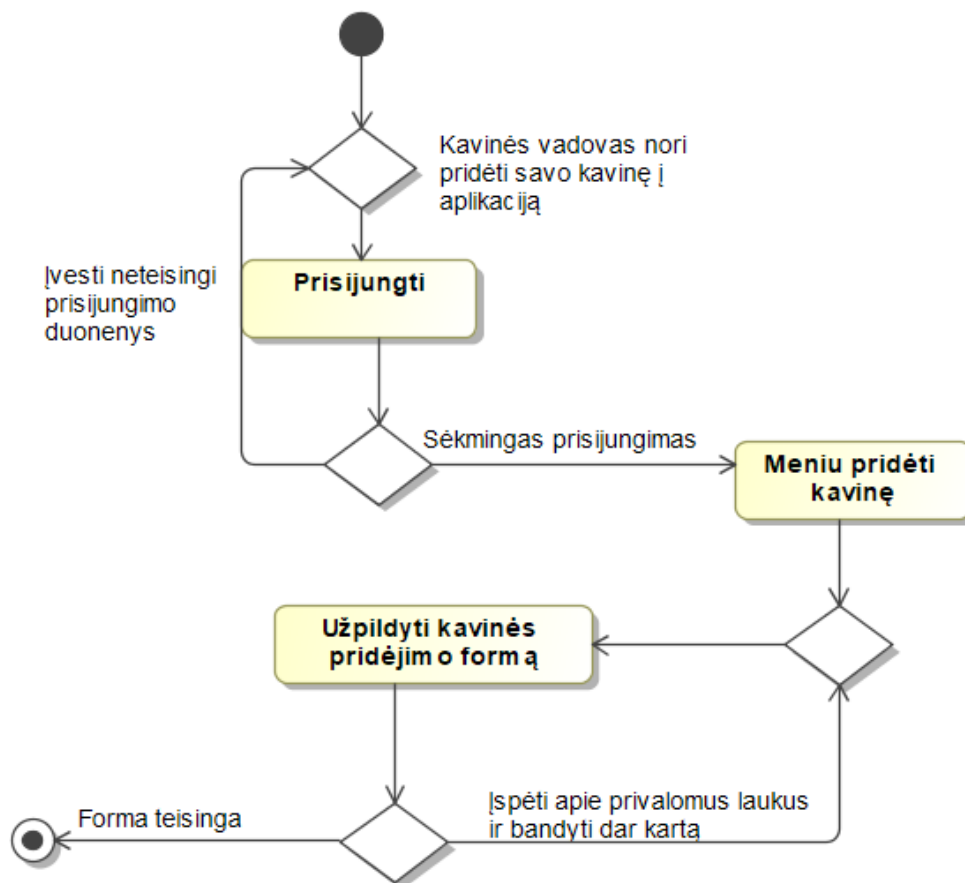
### 1.3. Dinaminis programų sistemos modelis (angl. Process view)

#### 1.3.1. Veiklos diagramos



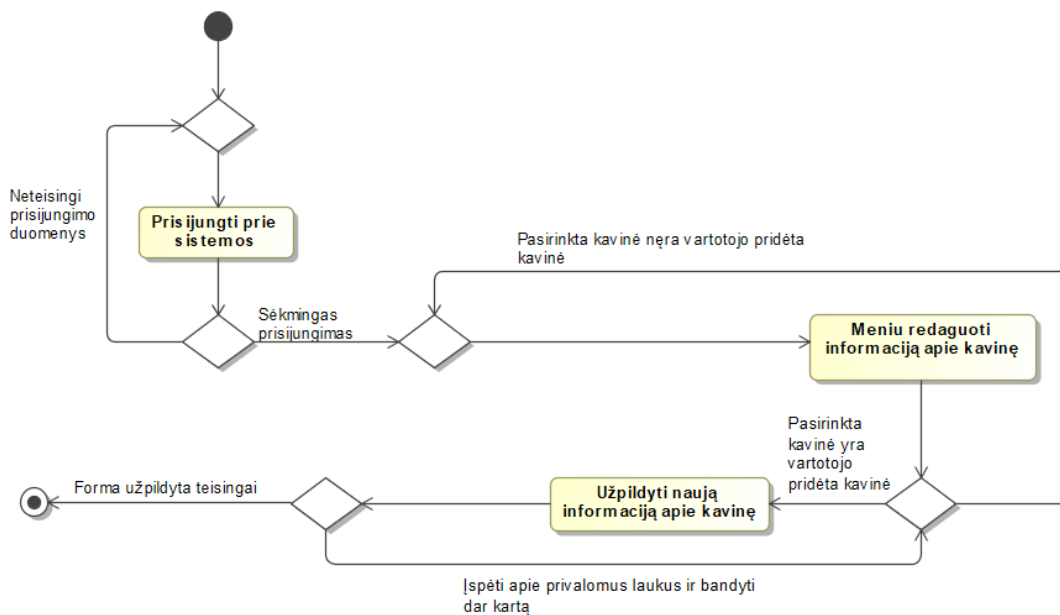
10 pav. Kavinės rezervacijos veiklos diagrama

10 pav. diagramoje nagrinėjami procesai, vykstantys tuo metu, kai vartotojas nori rezervuoti staliuką kavinėje. Rezervacija yra pasiekama tik po prisijungimo arba užsiregistravimo sistemoje. Vartotojas pamato prisijungimo ir registracijos opcijas tik paleidęs aplikaciją. Būsimas sistemos narys privalo užpildyti registracijos formą, parinkti saugų slaptažodį, bei nurodyti egzistuojantį el. paštą. Užpildžius formą neteisingai, reikia pakeisti netinkamus laukus. Sėkmingai prisijungus prie sistemos, vartotojas gali peržiūrėti aplikacijoje užregistruotų kavinių sąrašą. Jeigu vartotojas randa jam patinkančią kavinę, jis užpildo rezervavimo formą. Jeigu formoje visi laukai yra nurodyti teisingai ir restorane yra laisvų staliukų - rezervacija yra sėkminga. Darbas yra baigiamas tuo metu, kai vartotojas sėkmingai užsirezervavo staliuką, arba nusprendė nutraukt rezervaciją.



11 pav. Kavinės pridėjimo prie sistemos veiklos diagrama

11 pav. diagramoje nagrinėjami procesai, vykstantys vartotojui į sistemą pridedant kavinę. Norint pridėti kavinę į kavinių sąrašą, vartotojui būtina prisijungti (o neturint prisijungimo - prisiregistruoti) prie sistemos. Prisijungus meniu spaudžiama ant "Add cafe" mygtuko ir užpildoma kavinės pridėjimo forma. Jeigu visi laukai pažymėti "\*" (būtini) yra užpildyti - kavinė yra pridama prie sąrašo.



12 pav. Informacijos apie kavinę redagavimo veiklos diagrama

12 pav. diagramoje nagrinėjami procesai, vykstantys vartotojui norint pakeist arba atnaujint informaciją apie kavinę. Norint redaguot kavinės informaciją, vartotojui būtina prisijungti prie sistemos. Vartotojas atidaro visų kavinių sąrašą ir pasirinkus savo kavinę ir paspaudus mygtuką "Show info" jis gauna informacija apie jo kavinę bei apačioje formą, kuria teisingai užpildžius ir paspaudus mygtuką "Change" galima atnaujint/pakeist egzistuojančią informaciją apie kavinę.

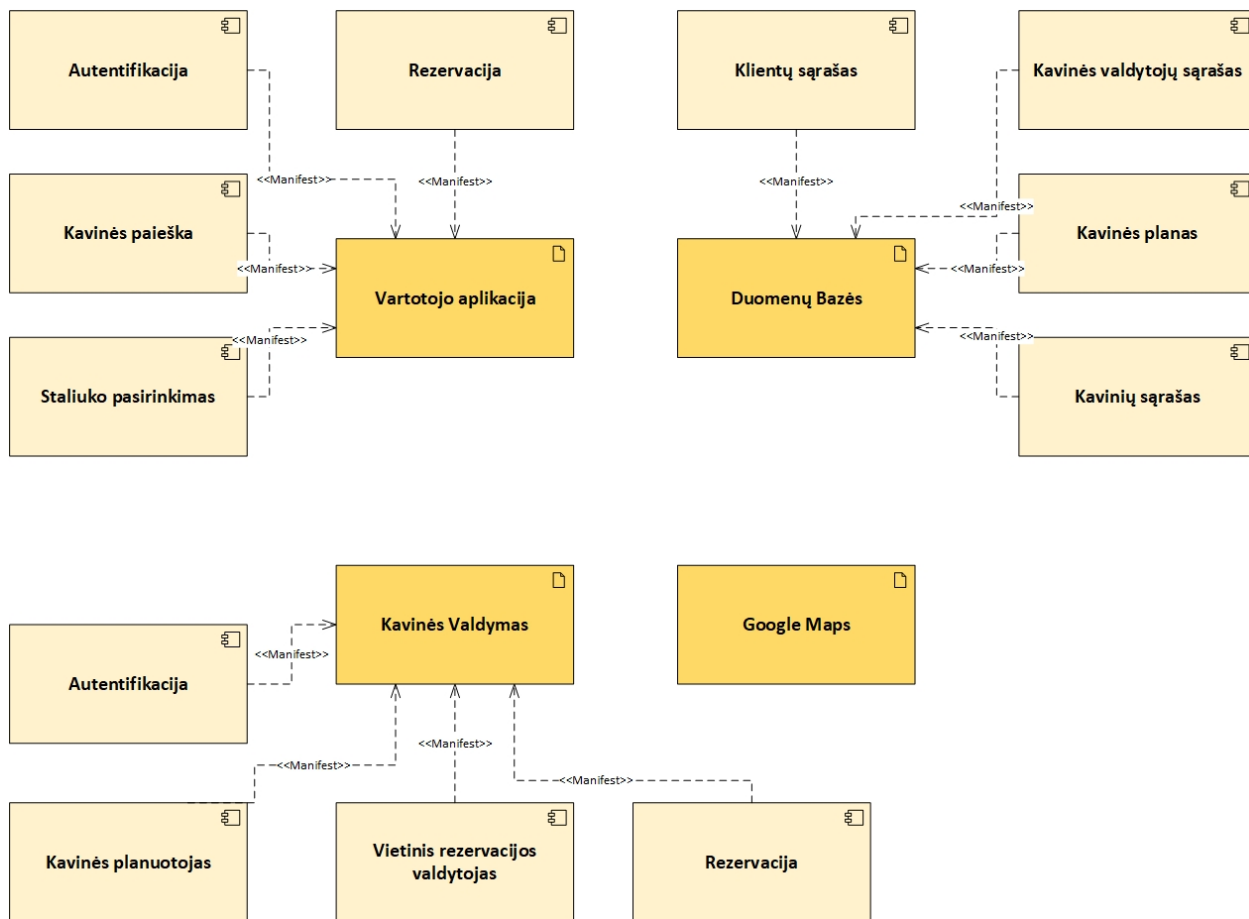
## 1.4. Programų sistemos išskirstymas tinkle(angl. Deployment view)

Fizinis pjūvis sudarytas iš Komponentų ir artefaktų ryšių ir mazgų diagramų. Šiose diagramose vaizduojamas programos komponentų išdėstymas tinkle bei komunikacijos protokolai tarp jų. Tuo pačiu diagramose parodoma, kokia programinė įranga turi būti įdiegta fiziniuose sistemos komponentuose.

### 1.4.1. Komponentų ryšių su artefaktais diagrama

Žemiau pateiktoje komponentų ryšių su artefaktais diagramoje yra išskirti pagrindiniai sistemos artefaktai. Artefaktus (angl. "artifact") ir komponentus (angl. "component") tarpusavyje sieja manifestacijos (angl. "Manifest") ryšys. Tai reiškia, kad artefakto sudaromoji dalys yra konkretus komponentai.



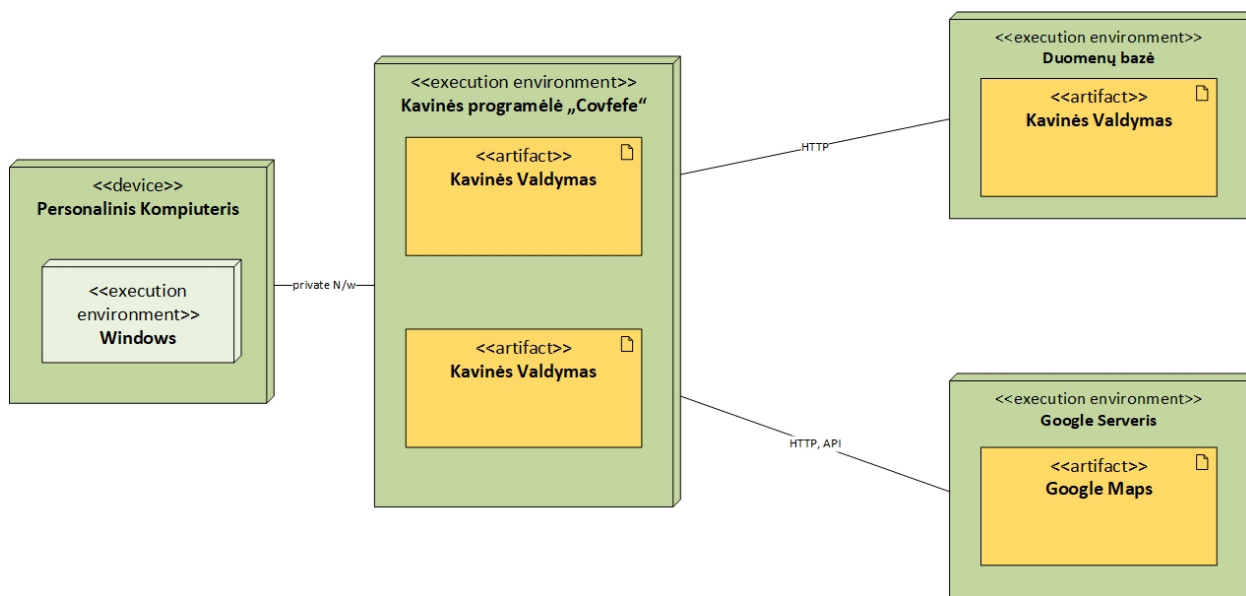


13 pav. Komponentų ryšių su artefaktais diagrama

Komponentų ryšių su artefaktais diagramoje (13 pav.) pavaizduoti keturi Kavinės programėlės artefaktai: tai Vartotojo aplikacija, Kavinės valdymas, Duomenų bazės, Google maps. Vartotojo aplikacija įgyvendina Autentifikacijos, Rezervacijos, Kavinės paieškos, Staliuko pasisirkimo komponentus. Kavinės valdymas - Autentifikacijos, Kavinės planuotojo, Vietinės rezervacijos valdymo, Rezervacijos komponentus. Duomenų Bazės - Klientų sąrašo, Kavinės valdytojų sąrašo, Kavinės plano, Kavinių sąrašo, komponentus. Google Maps turi savus komponentus, kurių detaliau neanalizuosim.

#### 1.4.2. Mazgų ir artefaktų ryšių diagrama

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



14 pav. Mazgų ir artefaktų ryšių diagrama

Sistema susideda iš dviejų pagrindinių mazgų Kavinės programėlės "Covfefe" ir "Azure" SQL duomenų bazės. Kavinės programėlės visi duomenys yra saugomi serverinėje duomenų bazėje, taip ši programėlė išvengia papildomų duomenų failų, neikvoja daug kompiuterio atminties ir paspartina procesų darbą. Programa pritaikyta veikti "Windows" platformoje. Vartotojui norint naudotis šia sistema, tereikia atsisiųsti ir įdiegti "Kavinės programėlę" ir turėti prieigą prie Interneto. Taip „Kavinės aplikacija“ pasiekia duomenis iš duomenų bazės, kurioje yra saugoma vartotojų prisijungimai, kavinės duomenys, kavinių staliukų duomenys. Tuo pačiu metu naršyklė kreipiasi į "Google" serverius, siekiant gauti "Google" žemėlapius. Toks sprendimas pasirinktas todėl, kad neužimtų serverio vietos, bei dėl to, kad "Google" riboja prieigą prie savo žemėlapių serviso.

### 1.4.3. Mazgų ryšių diagrama

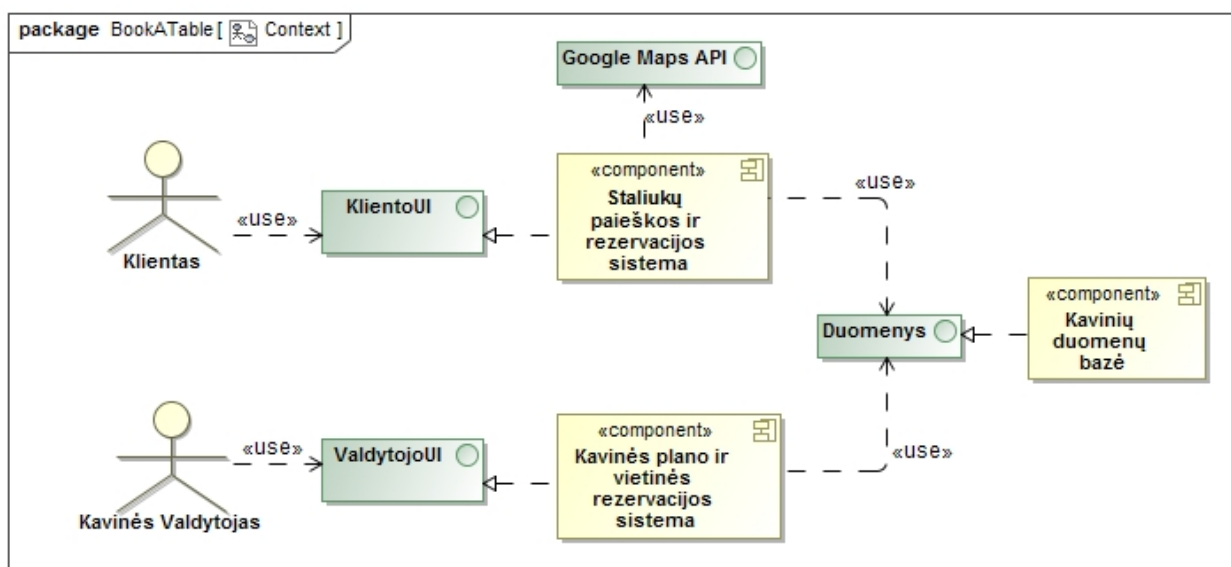
15 pav. Mazgų ryšių diagrama

Šiame paveikslėlyje pavaizduotas įrenginių (mazgų) išsidėstymas tinkle. Kavinės programėlė turi tiesioginį ryšį su duomenų baze ir Google serveriu, o vartotojai gali prisijungti prie programėlės. Tačiau klientai negali tiesiogiai pasiekti duomenų bazės bei Google serverio ir jose saugomų duomenų.

## 1.5. Programų sistemos kūrimo pjūvis (angl. Developement view)

Kūrimo pjūvis išdėstytas "top-down" būdu, t.y. nuo bendresnių diagramų pereinant iki detalesnių.

### 1.5.1. Konteksto diagrama



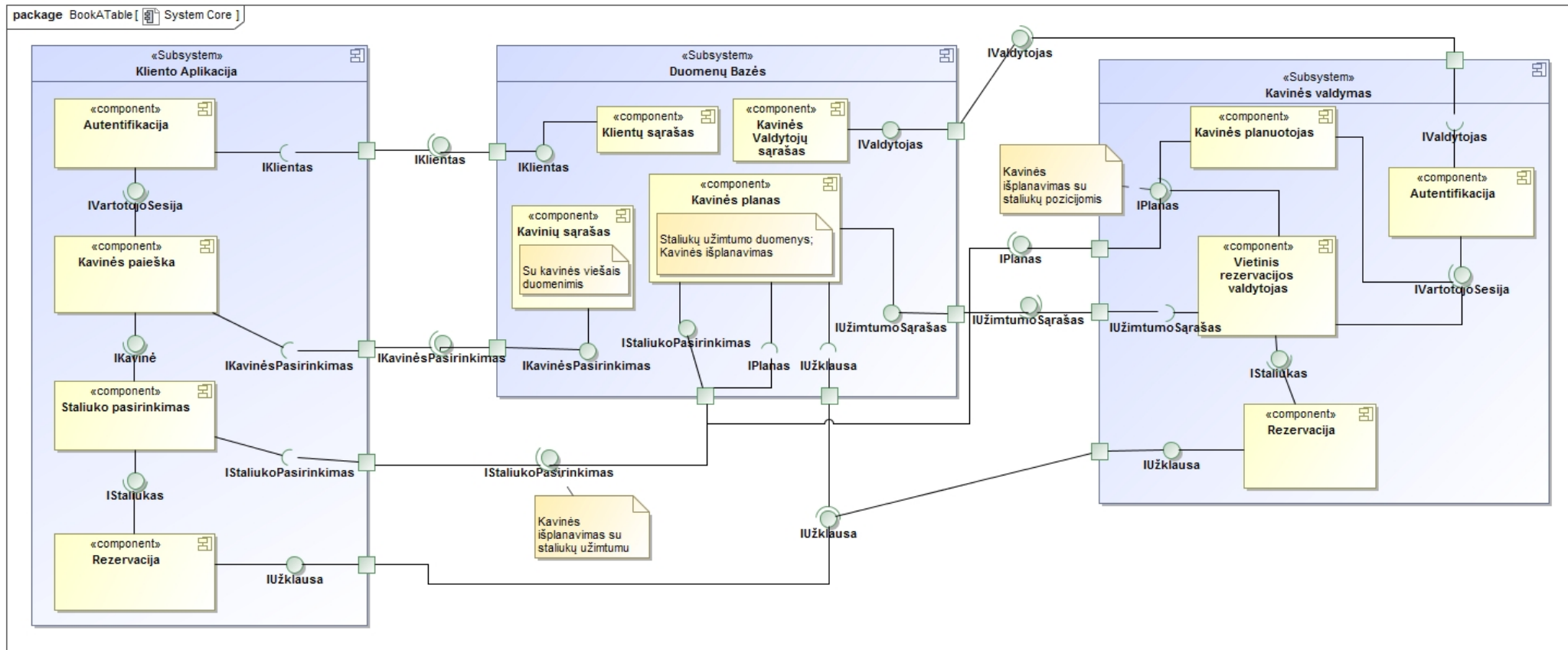
16 pav. Konteksto komponentų diagrama

15 pav. diagramoje parodomas aukščiausias komponentų struktūros lygis. Klientas naudoja Kliento Vartotojo Sąsają (diagramoje pavaizduota kaip KlientoUI) kurią gauna iš staliukų paieškos ir rezervacijos sistemos kuri pati naudoja Google Maps API, padedantį atfiltruoti netoliese esančias kavines.

Kavinės savininkas/vadovas/darbuotojas naudoja Vadovo Vartotojo Sąsają (diagramoje pavaizduota kaip ValdytojoUI) kurią gauna iš Kavinės plano ir vietinės rezervacijos sistemos.

Tiek Staliukų paieškos ir rezervacijos sistema, tiek Kavinės plano ir vietinės rezervacijos sistema naudoja duomenis, kuriuos gauna iš Kavinių duomenų bazės.

## 1.5.2. Subsystemų dekompozicija



17 pav. Subsystemų dekompozicijos diagrama

16 pav. diagrama atvaizduoja detalų sistemos komponentų struktūros lygį. 16 pav. pavaizduotos mėlynai apipavidalintos subsistemos (angl. subsystem) yra 15 pav. pavaizduoti komponentai:

- Kliento aplikacija yra Kavinų paieškos ir staliukų rezervacijos sistema;
- Kavinės valdymas yra Kavinės plano ir vietinės rezervacijos sistema;
- Duomenų bazės yra Kavinių duomenų bazės.

Toliau yra detaliau nagrinėjami šių subsistemų komponentai.

#### 1.5.2.1. Kliento aplikacija

- **Autentifikacija.** Vartotojui sėkmingai prisijungus, iš duomenų bazės klientų sąrašo gaunami įvairūs vartotojo duomenys. Pasinaudojus tais duomenimis pradedama vartotojo sesija (angl. User Session), kurios dėka klientas gali naudotis tolesniu aplikacijos funkcionalumu.
- **Kavinės paieška.** Kreipiasi į duomenų bazės kavinių sąrašo komponentą ir gauna, pagal vartotojo nurodytą filtrą, kavinių sąrašą su pagrindiniais kavinės duomenimis.
- **Staliuko pasirinkimas.** Iš kavinės paieškos vartotojui išsirinkus kavinę vykdoma kavinės duomenų (staliuko užimtumo/rezervacijos laiko, kavinės išplanavimo) užklausa į duomenų bazės kavinės plano komponentą. Gavus duomenis, vartotojas gali patogiai išsirinkti kurį nors laisvą staliuką konkrečioje kavinės vietoje.
- **Rezervacija.** Iš staliuko pasirinkimo komponento gaunamas norimas rezervuoti staliukas. Vartotojas nurodo rezervavimo laiką-datą. Turint visus rezervacijos duomenis, išsiunčiama rezervavimo užklausa (diagramoje pavaizduota kaip IUžklausa) į duomenų bazę, kurioje atsinaujinama staliuko būsena iš laisvo į rezervuotą.

#### 1.5.2.2. Kavinės valdymas

- **Autentifikacija.** Suvedus teisingus prisijungimo duomenis iš duomenų bazės gaunamas leidimas (diagramoje atvaizduotas kaip IVartotojoSesija) dirbti su konkrečios kavinės duomenimis.
- **Kavinės planuotojas.** Iš autentifikacijos komponento gavus valdomos kavinės duomenis, leidžia keisti kavinės išplanavimą (tuo pačiu ir staliukų skaičių, išsidėstymą). Išplanavimo duomenys vėliau siunčiami į duomenų bazės kavinės plano komponentą; naudojami vietinės rezervacijos valdyme.
- **Vietinės rezervacijos valdytojas.** Po sėkmingos autentifikacijos leidžia stebėti kurie staliukai yra laisvi, kurie užimti, kurie rezervuoti per aplikaciją iš kliento pusės. Taip pat suteikia galimybę rezervuoti, arba pažymėti kaip užimtą, staliuką iš kavinės pusės.
- **Rezervacija.** Iš Vietinės rezervacijos valdytojo komponento pasiima duomenis apie staliukų būseną (užimtas, laisvas, rezervuotas) ir siunčia duomenų pakeitimo užklausa į duomenų bazės kavinės plano komponentą.

### 1.5.2.3. Duomenų bazė

- **Klientų sąrašas.** Suteikia galimybę autentifikuoti vartotoją, laiko papildomus duomenis apie jį.
- **Kavinės Valdytojų sąrašas.** Suteikia galimybę autentifikuoti kavinės savininką/vadovą/darbuotoją, taip pat pateikia duomenis kokiai kavinei dirba šis asmuo (vėliau ši informacija reikalinga žinoti kurios kavinės duomenys modifikuojami).
- **Kavinių sąrašas.** Klientui ieškant kavinės, pateikia atfiltruotą kavinių sąrašą, kartu su kavinės reprezentacine informacija (užimtumas, darbo valandos, reitingas ir pan.).
- **Kavinės planas.** Laiko visa svarbiausią informaciją apie kavines:
  - Išplanavimas, staliukų išsidėstymas ir kiti tos srities duomenys ir jų pakeitimai gaunami iš Kavinės valdymo sistemos Kavinės planuotojo komponento.
  - Staliukų būseną - laisva/rezervuota - pakeičiama gavus prašymą (diagramoje pavaizduota kaip IUžklausa) iš Kliento aplikacijos sistemos. Įvairesnius pakeitimus gali atlikti užklausa iš Kavinės valdymo sistemos: atlaisvinti, užimti, rezervuoti, atšaukti rezervaciją.
  - Kavinės išplanavimas, staliukų būseną ir išdėstymo duomenys perduodami gavus užklausa iš Kliento aplikacijos sistemos Staliuko pasirinkimo komponento.
  - Konkrečios kavinės staliukų užimtumui pasikeitus, nauji duomenys yra siunčiami į tos konkrečios kavinės Kavinės valdymo sistemos Vietinės rezervacijos valdytojo komponentą, kurio dėka savininkas/vadovas/darbuotojas mato, jog klientas užsirezervavo staliuką.

## Išvados

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

## Terminų žodynas

**API**(angl. Application programming interface ) - Interfeisas, pateikiamas trečiųjų šalių, leidžiantis naudotis išorinės programos servais.

**Artefaktas** - Failas arba failų rinkinys, atsakingas už kurią nors sistemos veikimo dalį.

**HTTP** protokolas - Duomenų perdavimo protokolas, naudojamas visose naršyklėse komunikuojant su serveriu.

**Kavinės savininkas** - vartotojas, kuris yra užregistravęs jam priklausančią kavinę sistemoje.

**Kavinės savininkas** - vartotojas, kuris yra užregistravęs jam priklausančią kavinę sistemoje.

**Manifestacijos** (angl. “Manifest”) ryšys - nurodo, kad artefaktas negali egzistuoti be komponento, su kuriuo jis yra susietas šiuo ryšiu.

**Vartotojas** - prie sistemos prisijungęs žmogus, kuriam suteiktos teisės naudotis sistemos paslaugomis.