

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

**Automatinė ūkio valdymo sistema**

**Automatic farm management system**

Laboratorinis darbas I

Atliko:	2 kurso 3 grupės studentai	
	Matas Savickis	(parašas)
	Justas Tvarijonas	(parašas)
	Greta Pyrantaitė	(parašas)
	Rytautas Kvasinskas	(parašas)
Darbo vadovas:	Karolis Petrauskas, Doc., Dr.	(parašas)

## TURINYS

ĮVADAS .....	2
ŽODYNAS .....	3
1. SUKURTOS SISTEMOS APRAŠYMAS(V1.0) .....	4
1.1. Loginis pjūvis .....	4
1.2. Kūrimo pjūvis .....	6
1.3. Use cases .....	10
1.4. Proceso pjūvis .....	11
1.4.1. Veiklos diagramos.....	11
1.4.2. Būsėnų diagrama .....	11
1.5. Fizinis pjūvis .....	13
1.6. Pirmos dalies išvada .....	14
2. PERPROJEKTUOTOS SISTEMOS APRAŠYMAS(TO-BE, V2.0) .....	15
2.1. Loginis pjūvis .....	15
2.2. Kūrimo pjūvis .....	16
2.3. Use case .....	16
2.4. Proceso pjūvis .....	16
2.4.1. Sekų diagramos .....	17
2.4.2. Būsėnų diagramos.....	18
2.4.3. Veiklos diagramos.....	18
2.5. Fizinis pjūvis .....	21
REZULTATAI IR IŠVADOS .....	22

## Įvadas

Automatinė ūkio valdymo sistema (toliau - Auto ūkis) yra programa, leidžianti ūkininkui valdyti jo ūkį skaitmeniniu būdu. Auto ūkis leidžia registruoti gyvūnus ir stebėti kiekvieno jų bioparametrus (kraujo spaudimą, svorį, sveikatą) bei matyti ūkio technikos judėjimą po žemės plotą. Taip pat sistema vartotojui leidžia sekti dirvos parametrus (drėgmę, pH lygį), oro prognozes ir gyvūnų ligų paplitimą aplinkinėse teritorijose. Auto ūkis padeda ir su verslo valdymu: nesunkiai galima samdyti darbuotojus, atlikti buhalterinę apyskaitą, stebėti rinkos kainas ir apskaičiuoti bei numatyti galimą pelną. Iškilus nelaimei per Auto ūkio sistemą galima greitai iškviešti greitąją pagalbą, policiją, gaisrinę ar saugos tarnybą. Orų prognozės yra paimtos iš [www.gismeteo.lt](http://www.gismeteo.lt). Pagrindinė sistemos inovacija yra tai, kad, kai sistema yra pilnai įdiegta, darbuotojų skaičius, reikalingas palaikyti ūkį, tampa minimalus. Kadangi ūkio technika būtų valdoma automatiškai, vairuotojų ir derliaus nurinkėjų nereiktų. Gyvūnų sekimas yra įgyvendinamas mikro kontrolerio su Arduino pagalba. Šis kontroleris nedidelis ir lengvai pritaikomas visokio pobūdžio darbams. Jį, kartu su WiFi modulių, sistema naudoja gauti gyvūno lokaciją per Google Maps,- taip pasiklydę ar pavogti gyvūnai būtų greitai surandami ir grąžinami. Žemės laistymas ir tręšimas taip pat būtų automatizuotas: parametrai gaunami per Arduino detektorius, kurie pagal pasikeitusią dirvos kompoziciją nusprendžia, ko trūksta žemei, ir aktyvuoja laistymo ir tręšimo sistemas. Darbuotojų samdymas yra įgyvendintas per darbo biržos puslapį, kur greitai ir nesunkiai galima įdėti skelbimą arba surasti darbuotoją. Buhalterija yra tvarkoma naudojantis nemokama buhalterijos programa Wave Accounting, kuri yra implementuota į Auto ūkį. Auto ūkio sistema yra parašyta JAVA kalba - tai leidžia programą paleisti ant bet kurios operacinės sistemos. Ateityje numatoma galimybė programą perkelti į išmaniuosius telefonus. Sistema buvo projektuojama pasitelkiant [www.planttext.com](http://www.planttext.com) ir [www.draw.io](http://www.draw.io) funkcionalumą.

# Žodynas

- Klasės:

- \* AutoŪkis - pagrindinė (main) programos klasė. Ši klasė piešia grafinę vartotojo sąsają ir laiko savyje kitų klasių objektus, kurių informacija reikalinga piešimui.
- \* Map - teritorijos piešimui skirta klasė.
- \* ŽemėsTeritorija - apskaičiuoja tam tikros teritorijos plotą.
- \* Gyvūnas - klasė, skirta gyvūno rodmenims ir metodams saugoti.
- \* AriamasLaukas - laiko savyje reikšmes, apibūdinančias unikalų lauką, ir metodus, susijusius su lauko darbu.
- \* Ganykla - laiko parametrus ir metodus darbui su ganyklomis, kurios yra žemės plote.
- \* ŪkinisPastatas - saugo ūkinę techniką arba gyvūnus.
- \* ŪkioTechnika - laiko ūkio technikos duomenis ir apskaičiuoja technikos judėjimo greitį.
- \* Žemės parametrai - saugo įvairius žemės parametrus (drėgmė, pH...).
- \* Orai - klasė, skirta gauti vartotojui reikalingas orų prognozes iš [www.gismeteo.lt](http://www.gismeteo.lt).
- \* Žemės detektorius - klasė, skirta bendrauti su žemės detektoriumi.

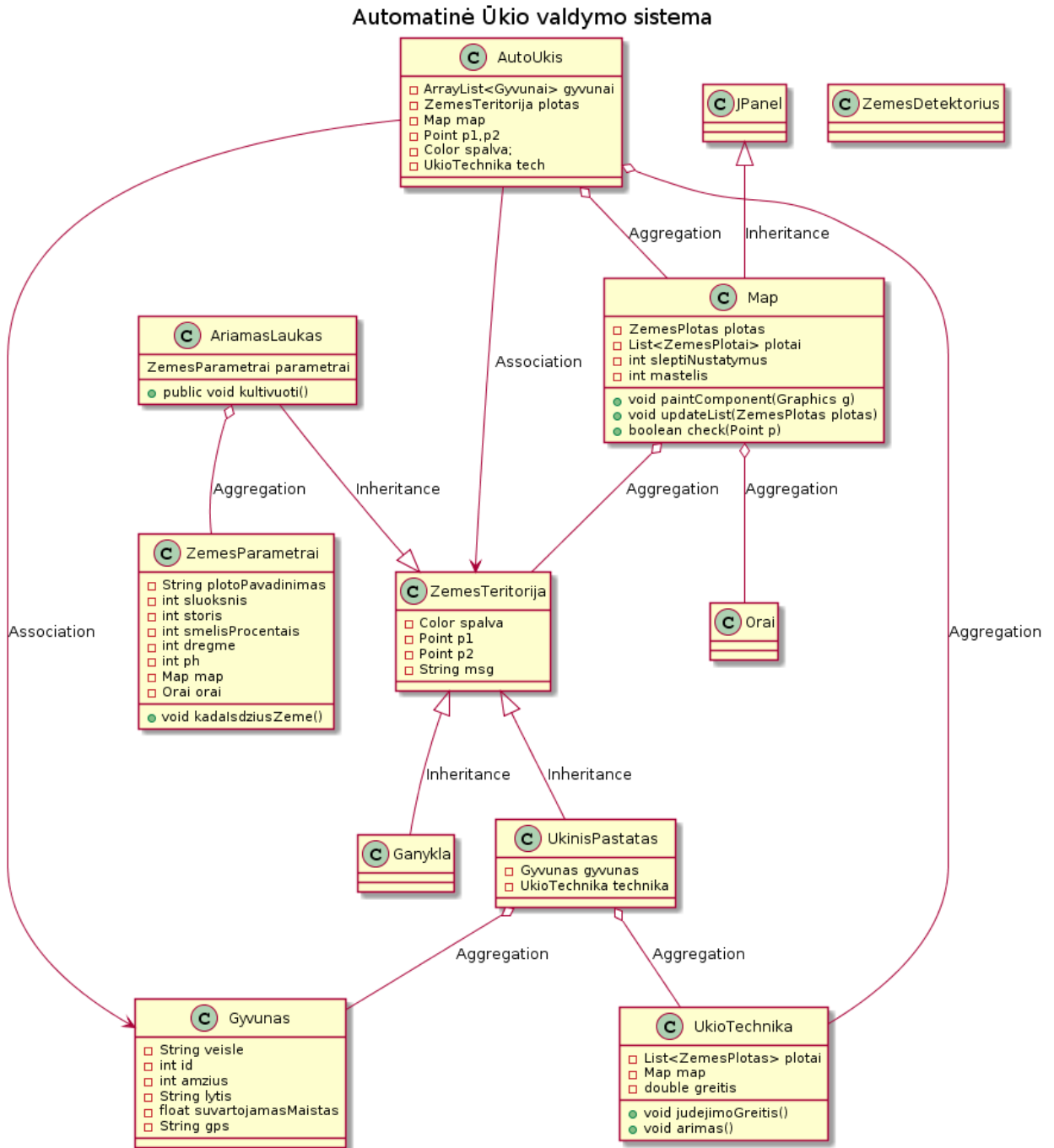
- Bendri terminai:

- \* Žemės plotas - vieta, kurią valdo ir gali stebėti vartotojas(ūkininkas).
- \* Detektorius - Arduino mikro kontrolieris.
- \* Ūkininkas - žmogus, kurio valdomoje teritorijoje įdiegtas Auto ūkis.
- \* Arduino - mikro kontrolieris, skirtas ūkio sekimui.
- \* Automatiškai valdoma - valdymui nereikalinga žmogaus pagalba.
- \* Darbuotojas - žmogus, dirbantis ūkininko versle.
- \* Gyvūnas - visi gyvūnai, kurie priklauso ūkininkui, ir yra registruoti Auto ūkis sistemoje.

# 1. Sukurtos sistemos aprašymas(v1.0)

## 1.1. Loginis pjūvis

Pagal suprogramuotą šabloninį programos karkasą nubraižėme *UML diagramą(1 pav)* minėta PlnatText programa.



1 pav

- Dizainas:

- Pagrindinė klasė yra AutoUkis.form. Joje sukurtas ir aprašytas Graphical User Interface (GUI) ir visas vartotojo bendravimas su programa vyksta per ją, nes per ją pasiekiami visi duomenys iš kitų klasių, pavyzdžiui, duomenys, esantys klasėje Gyvūnas, kurioje įrašoma vartotojo įvesta informacija apie gyvūną (veislė, amžius, t.t.). Taip pat AutoUkis klasėje kuriama dauguma objektų ir jie ten laikomi, sudedami į sąrašus. Visos kitos klasės turi savo atskiras paskirtis, tokias kaip žemėlapių braižymas, oro prognozių sekimas ir įvairių parametrų laikymas. Kai kurios klasės (pvz., ZemesDetektorius) buvo sukurtos vėlesniam panaudojimui, bet šiuo metu nėra niekur panaudotos. Dėl to, ką būtų buvę galima daryti kitaip, - GUI perkėlimas į atskirą klasę padarytų programą skaitomesnę ir tvarkingesnę, būtų lengviau rasti atskirą kodą. Dar viena alternatyva būtų įgyvendinti front-end dalį web aplinkoje, bet šiuo metu nematome tam būtinybės.

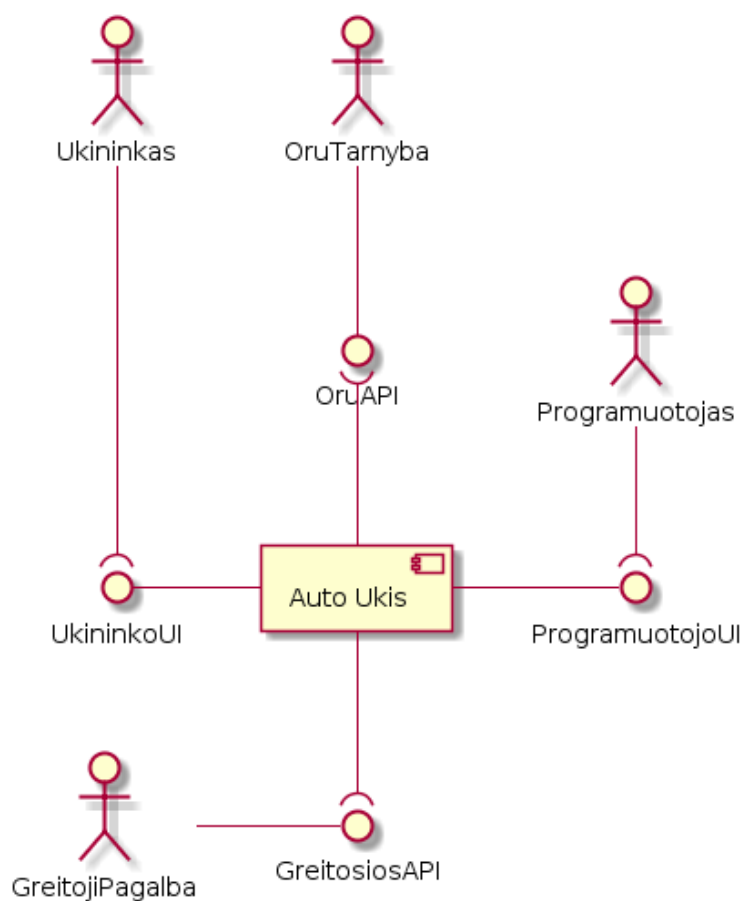
- Funkcionalumas:

- Viso užsibrėžto programos funkcionalumo įgyvendinti nepavyko. Kai kurios klasės buvo sukurtos ateityje planuojamoms funkcijoms, kurios dar nėra implementuotos. Programa kol kas veikia tik ant kompiuterio ir vienintelis jos bendravimas su internetu yra per Orai klasę, kuri skirta vartotojo pasirinkto miesto orų prognozėms gauti iš gismeteo.lt svetainės. Klasėse Gyvūnas, Map, ZemesTeritorija ir iš jos išeinančiose klasėse saugomi atitinkami duomenys apie sukurtus objektus bei aprašyti dar neišplėtoti metodai, tokie kaip žemės teritorijos žymėjimas. Planuojama, kad klasė ZemesDetektorius generuos atsitiktinius parametrus, kurie bus perduoti ZemesParametrai klasei. Nepilnai įgyvendintas funkcionalumas ir neišbaigtos klasės sukelia nepatogumų aprašant programą, nes sunku braižyti diagramas, suvokti aiškius ryšius tarp komponentų ir vykdomas funkcijas.

## 1.2. Kūrimo pjūvis

- Dizainas:

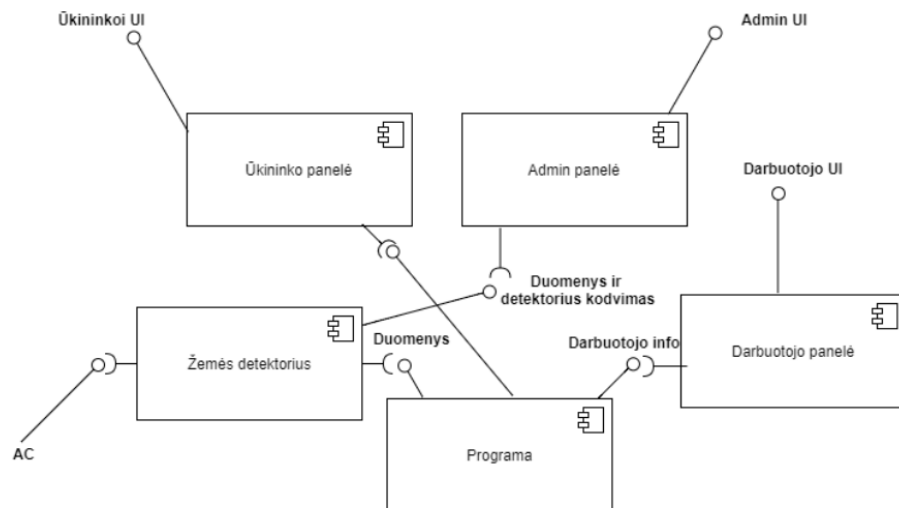
- Pradėjus rašyti programą nepagalvojome apie kūrimo pjūvį ir kaip teisingiau būtų galima pradėti viską. Žiūrint dabar, visa programa buvo pradėta kurti pagal Bottom -> Up principą. Iš pradžių apsirašėme daugybę mažų klasių ir paskui jas bandėme apjungti į didesnę sistemą. Išskyrėme tokius komponentus kaip ūkininkas, programa, orų tarnyba, žemės detektorius ir administratorius. Kiekvienas komponentas turi skirtingas prieigas prie informacijos ir skirtingas funkcijas, reikalingas ūkio visapusiškam funkcionavimui. Kai kurios klasės liko nepanaudotos, nes šiuo metu jos neatrodo pakankamai svarbios pradiniam projekto variantui. Ūkiniko, Admin ir darbuotojo panelės įtrauktos į dokumentaciją norint pavaizduoti skirtingas prieigas prie sistemos.



2 pav

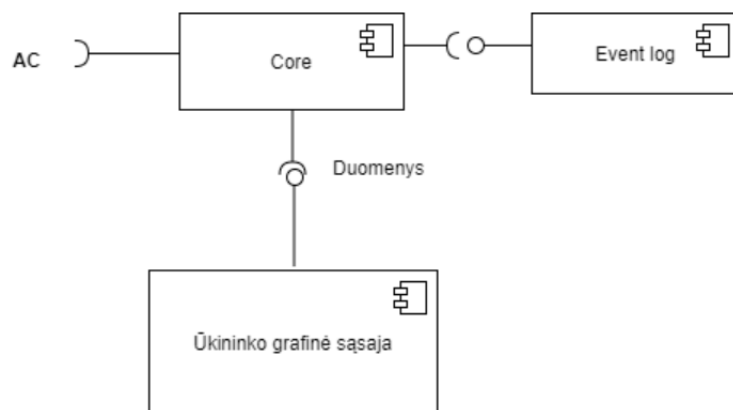
- LO:

- Šioje diagramoje (2 pav) pavaizdavome sistemos bendravimą su išoriniais agentais, tokiomis kaip Greitoji pagalba, Ūkininkas ir t.t. . Ši diagrama aiškiai ir paprastai parodo kuriuos ir įgyvendinamus interfeisus. Galbūt būtų galima Greitosios Pagalbos interfeisą išskaidyti į kelis detalesnius interfeisus, bet apskritai didelių problemų nepastebime.



3 pav

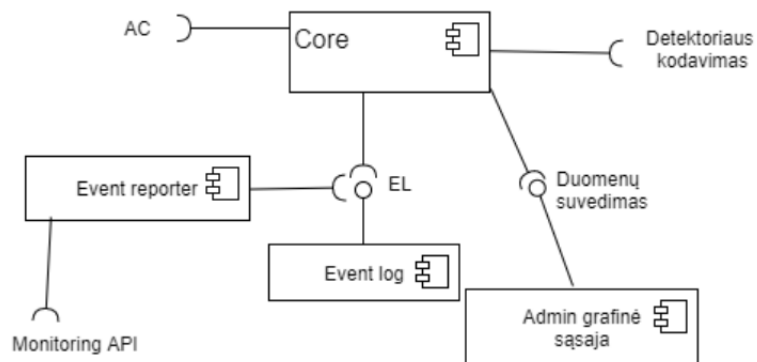
- L1: Sudėjus komandos idėjas apie tai, kaip turėtų atrodyti *L1 diagrama*(3 pav), supratome, kad mūsų sistema neturi normalios struktūros ir gerai nebuvo pagalvoję kaip visi komponentai sietis vieni su kitais, todėl ir diagrama atrodo chaotiška. Trūksta konkretumo, kaip turi Admin sietis su kitais komponentais. Programa atsiranda kaip komponentas, o tai greičiausiai yra nekorektiška.



4 pav

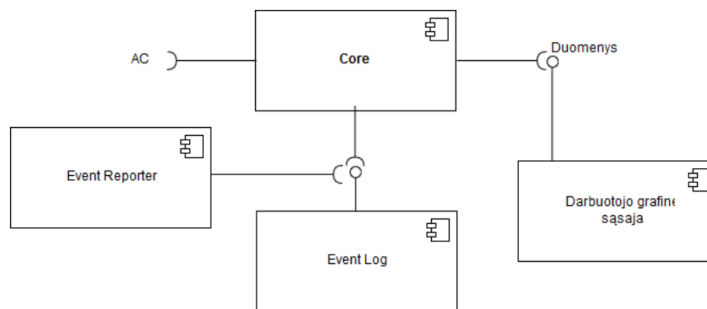
- L2(Ūkininkas): Šioje diagramoje (4 pav) parodyta, kad programos pagrindas (Core) kuria interface'ą, kuriuo perduoda duomenis grafinei vartotojo sąsajai. Duomenys yra perduodami Event Log komponentui.





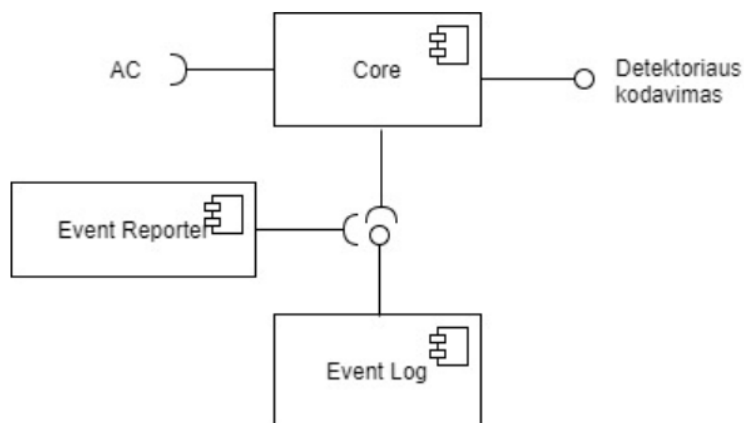
5 pav

- L2(Admin) Šioje diagramoje (pav 5) parodyta, kad programos pagrindas naudoja duomenų suvedimo interface, kurį suteikia admin grafinė sąsaja, bei naudoja Detektoriaus kodavimo interface. Visus įvykius įrašo į Event Log.



6 pav

- L2(Darbuotojas): Šioje diagramoje (pav 6) parodyta, kad programos pagrindas naudoja grafinę sąsają ir perduoda duomenis į Event Log'ą.



7 pav

- L2(Detektorius): Ši diagrama (pav 7) vaizduoja detektoriaus išvedamus duomenis. Įvykiai įrašomi Event Log'e.

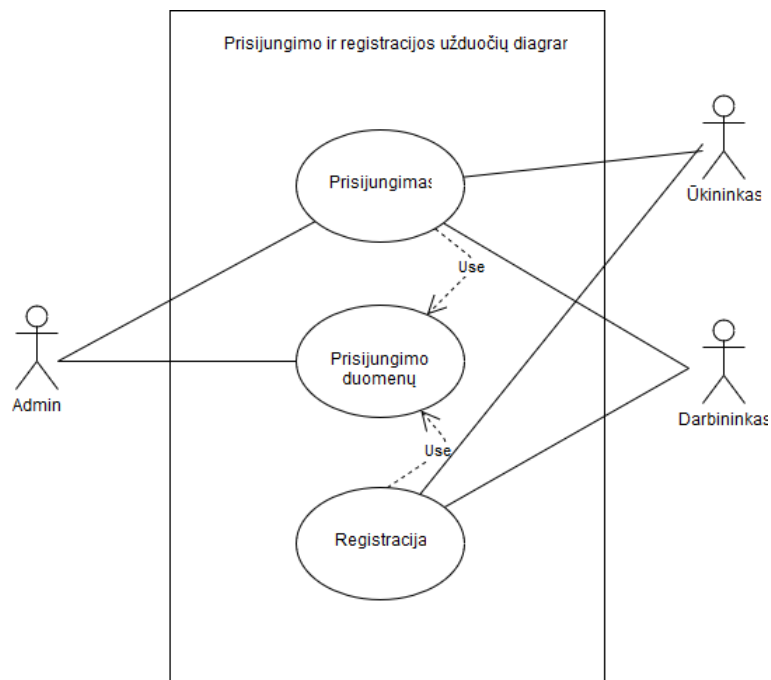
- Elementų ir užduočių ryšių matrica:

Komponentai	Detektorius	Orų tarnyba	Programa	Ūkininko UI	Admin UI	Greitoji pagalba
Užduotys						
Registruoti gyvūnus			+	+		
Orų prognozė		+				
Žemės parametrai	+					
Duomenų suvedimas			+	+	+	
Skaityti ir rašyti event log'ą			+		+	
Samdyti darbuotojus				+		
Kviesti pagalbą			+	+		+
Duomenų skaitymas			+	+	+	
Verslo reikalų tvarkymas			+	+		
Išteklių palaikymas				+		

8 pav

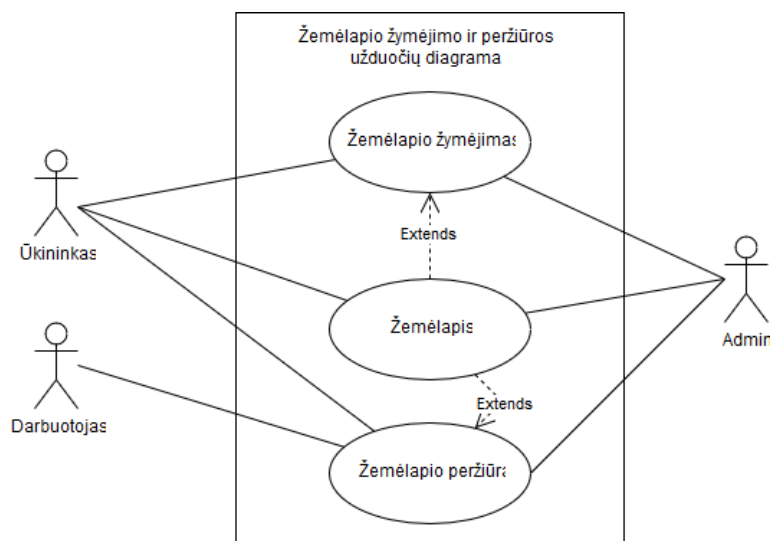
### 1.3. Use cases

- Prisijungimo ir registracijos Use Case:



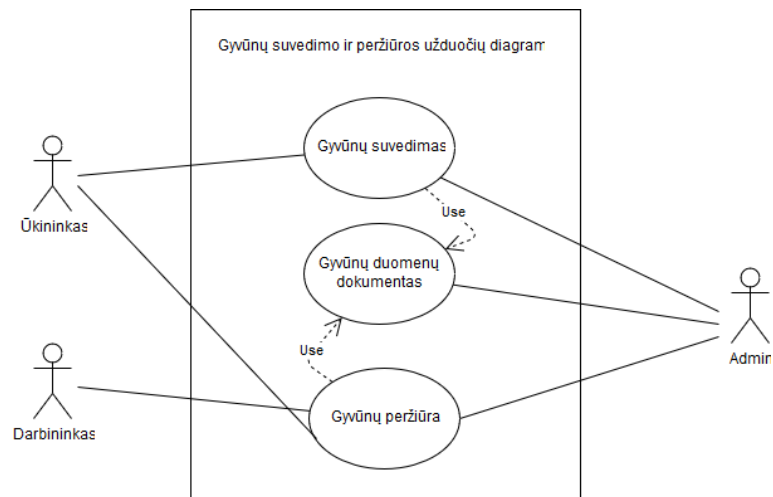
9 pav

- Žemėlapio žymėjimo ir peržiūros Use Case:



10 pav

- Gyvūnų suvedimo ir peržiūros Use Case:



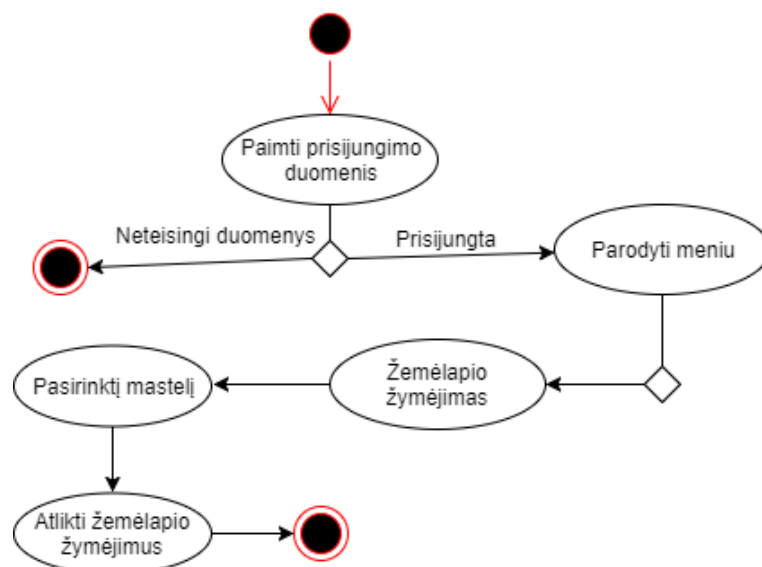
11 pav

## 1.4. Proceso pjūvis

Šiame skyriuje parodoma programos elgsena jos vykdymo metu.

### 1.4.1. Veiklos diagramos

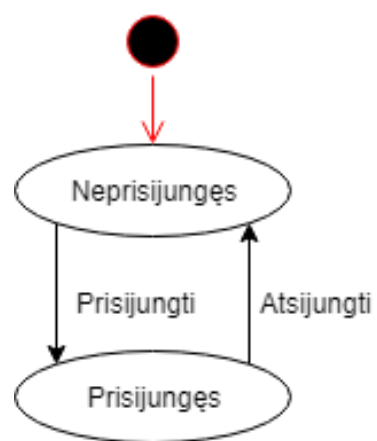
Vaizduojama *žemėlapio žymėjimo veiklos diagrama* (pav 8). Nurodomi pagrindiniai žingsniai braižant žemėlapi.



12 pav

### 1.4.2. Būsėnų diagrama

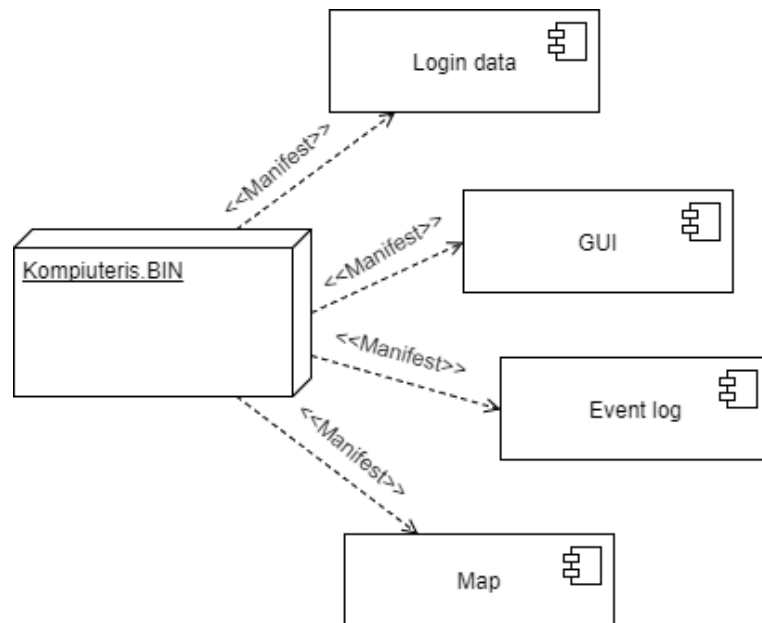
Šioje diagramoje (pav 9) parodomos galimos vartotojo būsenos ir keliai jom pasiekti. Šiuo metu programoje tėra dvi būsenos, taigi diagrama yra labai paprasta.



13 pav

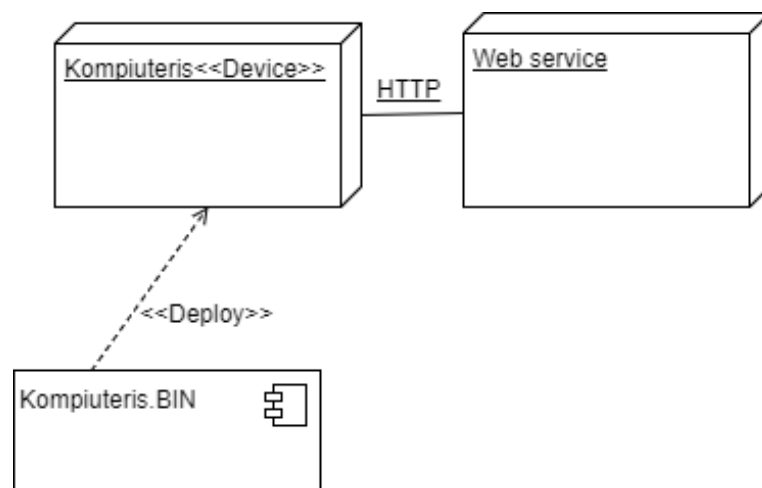
## 1.5. Fizinis pjūvis

Šiame skyriuje parodoma (10 pav.) šios sistemos techninė įranga, komunikacija, tačiau kadangi šioje šabloninėje versijoje nenaudojame kitų techninių resursų be kompiuterio, fizinis pjūvis parodo tik nedidelį kiekį informacijos.



14 pav

- D0: Šioje diagramoje(11 pav) parodyta, kas saugoma device kompiuteris. Iš diagramos matome, kad šiame įrenginyje saugomi prisijungimo duomenys, vartotojų grafinės sąsajos, teritorijos žemėlapis bei programoje atliktų veiksmų išrašas. Alternatyva buvo saugoti šiuos duomenis išnuomuotame web service, tačiau daug negalvoję nusprendėme duomenis saugoti kompiuteryje.



15 pav

- D1: Tai *bendresnė D0 diagrama*, joje matome, kad kompiuteris, norėdamas gauti pranešimus apie orus, bendrauja su web HTTP ryšiu.

## 1.6. Pirmos dalies išvada

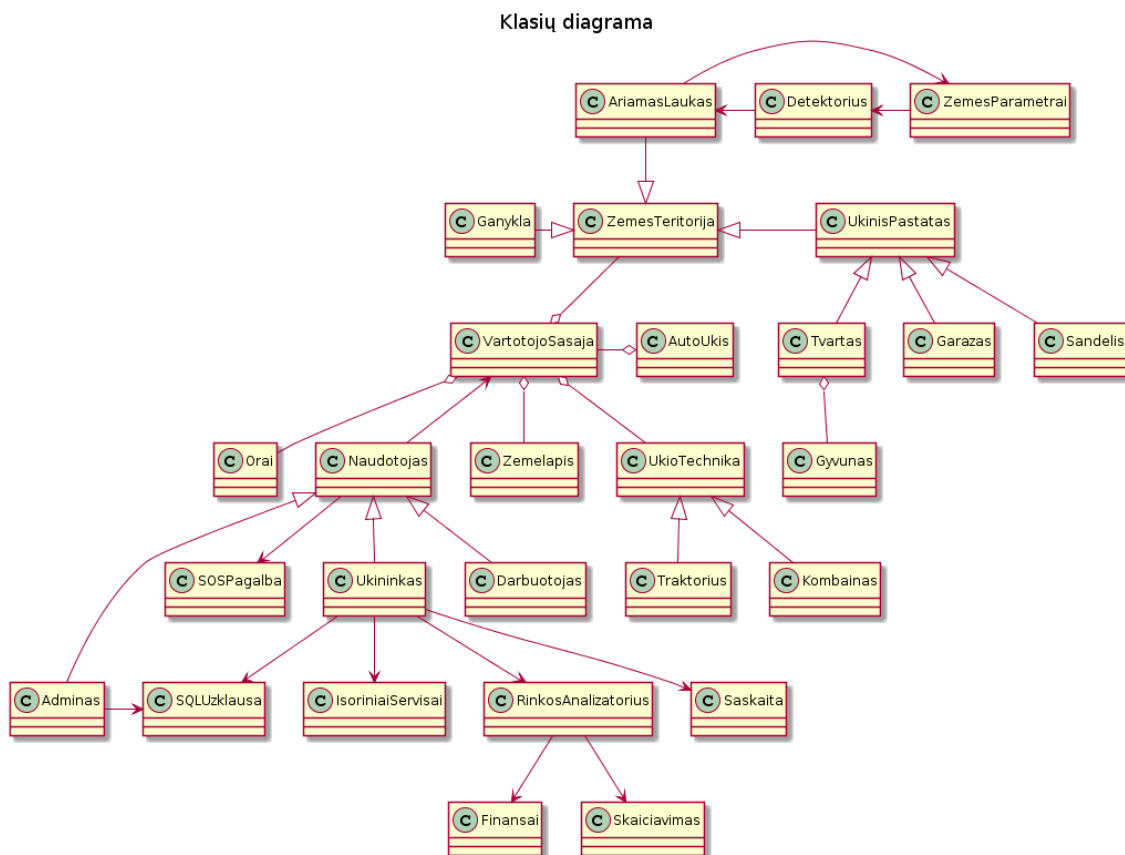
Pirmoji programos versija buvo suprogramuota ir suprojektuota per daug negalvojant apie sistemos plėtimą ateityje. Nors klasėse ir išlaikėme enkapsuliavimo principą, bendras objektinio programavimo principas nebuvo išlaikytas, klasės yra per daug viena nuo kitos priklausomos.

Rimtesniai sistemai kurti reiktų naudotis Top to Bottom principu lengvesniam naujų funkcionalumų pridėjimui. Stipri ir paprasta sistemos dalis pasimato kūrimo pjūvyje, kuriame yra aiškūs ryšiai tarp kuriamų ir įgyvendinamų interface'ų. Aprašinėdami L1 ir L2 supratome, kad nepagalvojome apie tai, kaip sistemos vartotojai bendrauja su sistema. Šią dalį reiks pergalvoti antroje programos versijoje, kad viskas taptų aiškiau ir paprasčiau, reikia labiau pasidomėti, kaip tokios sistemos veikia realiaame pasaulyje. Proceso pjūvį pavyko aprašyti pakankamai paprastai ir aiškiai, tačiau jam dar trūksta detalumo ir kitų scenarijų numatymo, pavyzdžiui, prisijungti prie sistemos. Fiziniame pjūvyje aprašyta techninė sistema yra ganėtinai paprasta ir primityvi.

Pagrindinis trūkumas - nepagalvota, kas nutiktų kompiuterio išsijungimo atveju, ar kas nutiktų atsiradus poreikiui plėsti sistemą. Žiūrint bendrais idėjos bruožais, brangiausia sistemos dalis yra automatinės mašinos, kurios dar yra sąlyginai nauja technologija, ir mikro kontrolieriai kiekvienam gyvūnui. Jeigu ferma turi jų daug, visas instaliavimas kainuotų gana brangiai ir turbūt neatneštų didelės naudos. Tai labiau inovacija dėl inovacijos, ne dėl funkcionalumo. Tačiau kitos dalys visai sėkmingai pritaikomos. Tokią sistemą, kokią padarėme dabar, įgyvendinti būtų įmanoma, tačiau praplėsti ir palaikyti ją būtų nepatogu, ir sistema turbūt nedirbtų taip greitai, kaip norėtūsi. Trūksta funkcionalumo su išmaniuoju telefonu, darbo birža, rinkos tendencijomis ir kita.

## 2. Perprojektuotos sistemos aprašymas(To-Be, v2.0)

### 2.1. Loginis pjūvis



16 pav

- Dizainas:

- Visas programos dizainas paremtas Top to bottom principu. Pradėjome galvoti dideliais objektais ir juos išskirstėm į mažesnius. Pagrindinė klasė yra AutoUkis kuris iškviečia VartotojoSasaja klasę kuri ir piešia visa prieinamumą vartotojams. Galima nesunkiai pridėti kitokią vartotojo sąsają ar betkokią kita UI piešimo galimybę kaip pavyzdžiui piešti UI išnamiajame telefone. Programos modalumas leidžia nesunkiai pridėti pridėti funkcionalumo, nes programos dalys yra atskiros viena nuo kitos. Pavyzdžiui ištrynus klasę ŪkinisPastatas visos programos veikimas nesutiriktų. Tuo pačiu metu norint pridėti naują naują ūkini pastatą galima nesunkiai tai padaryti įvykdžius paveldėjimą iš tėvinės UkinisPastatas klasės taip, kaip tai daro klasės Garažas, Tvirtas ir Sandelis.

- Funkcionalumas:

- Programos funkcionalumas susideda iš trijų pagrindinių dalių, vartotojų prieiga prie sistemos, finansų bei išteklių(žmogiškųjų ir natūraliųjų) valdymas bei išteklių sekimas. Prieigos dalyje funkcionalumas pasižymi prieigos išskaidymą. Skirtindi sistemos

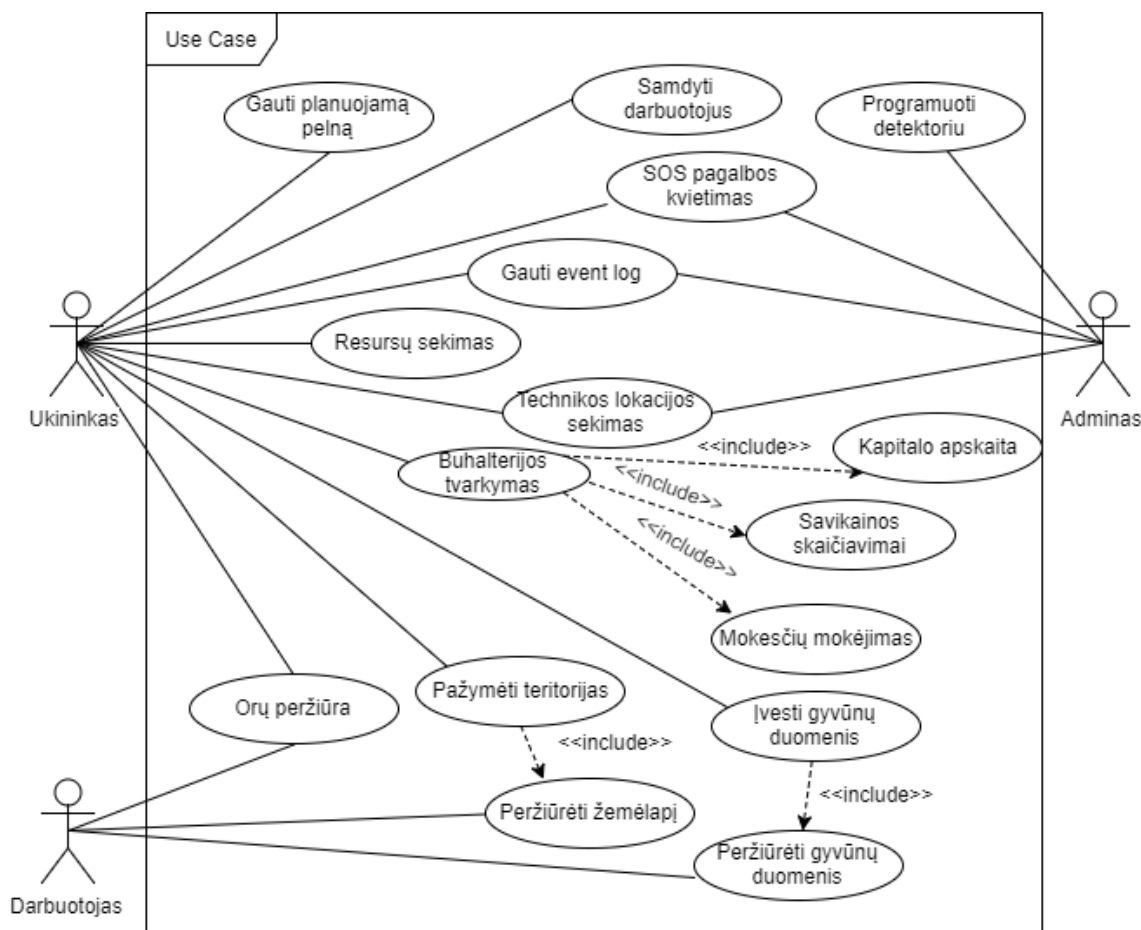


vartotojai gauna skirtingą prieigą ir galimybes naudotis sistema. Pavyzdžiui darbuotojas negali tvarkyti įmonės finansų, tačiau visi sistemos vartotojai gali kviesti greitąją pagalbą. Nesunku implementuoti naują vartotojų klasę (pvz. Svečias). Finansų bei išteklių dalyje ūkininkas gali tvarkyti tvarkyti savo išteklius. Pavyzdžiui samdyti arba atleisti darbuotojus, stebėti rinkos kainą ir nuspresti kada jam palankiausia parduoti, sudarinėti sąskaitas ir tvarkyti kitą buhalteriją. Trečiojoje sistemos dalyje yra įgyvendinamas išteklių sekimas. Vartotojai turintys prieigą gali stebėti žemės parametrus, ūkio technikos sąrašą, ūkinių pastatų sąrašą bei gyvūnų priklausančių sistemai sąrašą.

## 2.2. Kūrimo pjūvis

### 2.3. Use case

- Diagramoje (12 pav) pavaizduoti visi galimi veiksmai, kuriuos gali atlikti kiekvienas programos vartotojas.



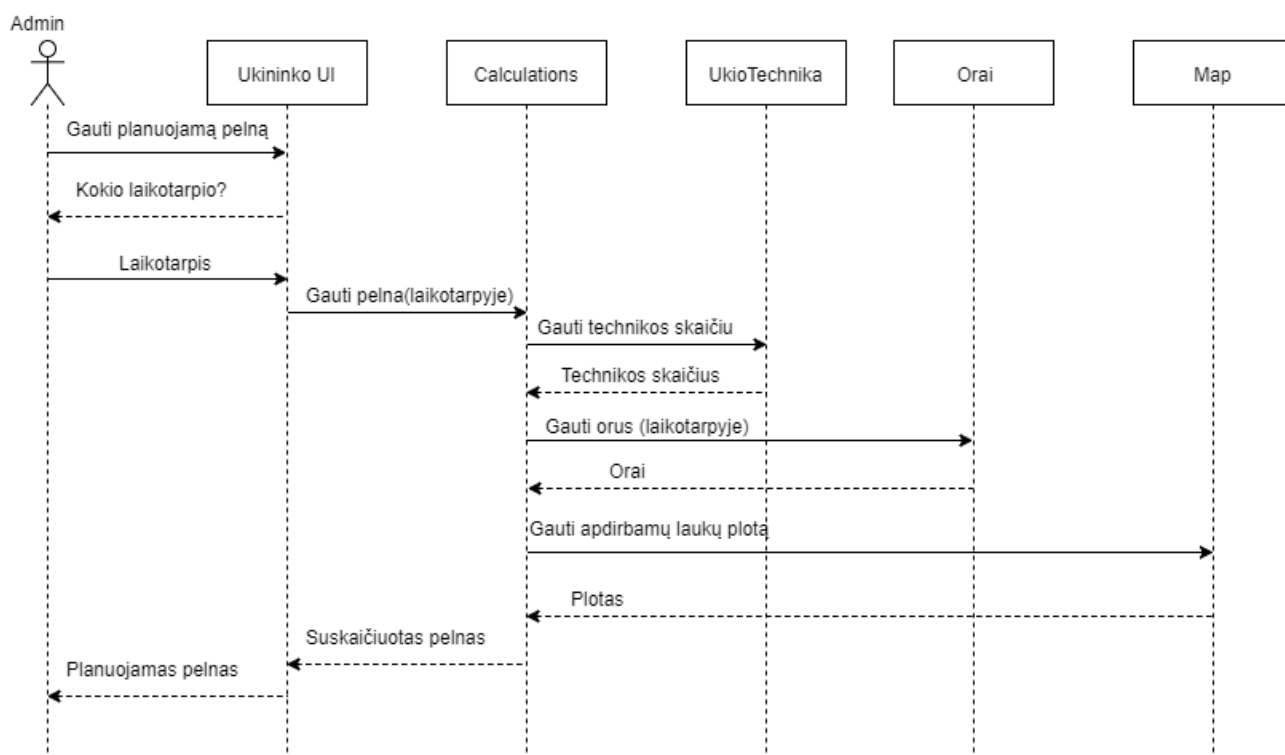
17 pav

## 2.4. Proceso pjūvis

Šiame

### 2.4.1. Sekų diagramos

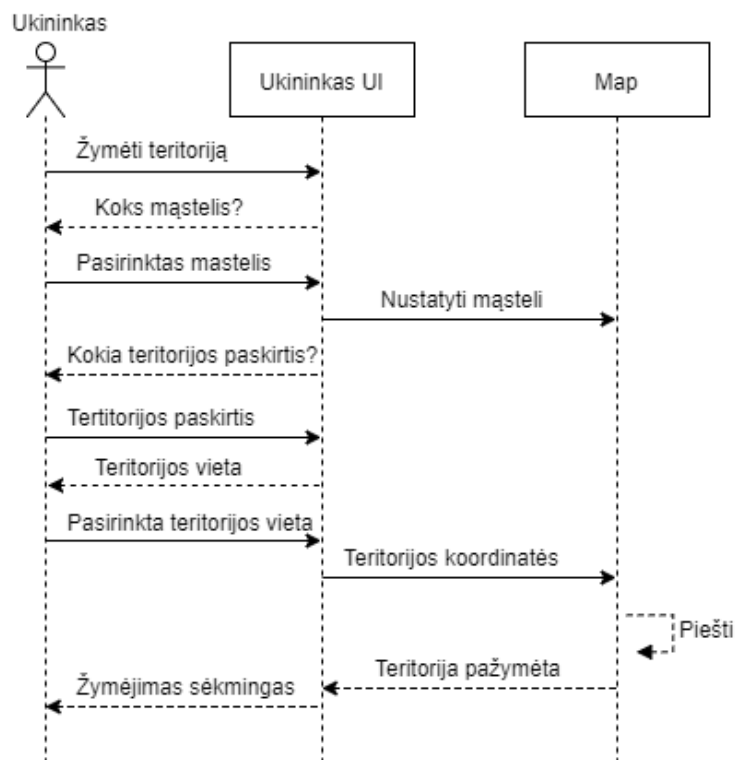
- Diagramoje (12 pav) pavaizduotos seka, kurią programa atlieka Admin norint gauti planuojamą jo pasirinkto laikotarpio pelną.



18 pav

Šioje diagramoje matome, kad ūkininkui pateikus užklausą planuojamui pelnui gauti, Ūkininkas UI kreipiasi į Calculations su prašymu jį apskaičiuoti. Šis savo ruožtu norėdama gauti reikiamus duomenis kreipiasi į klases UkioTechnika, Map ir Orai.

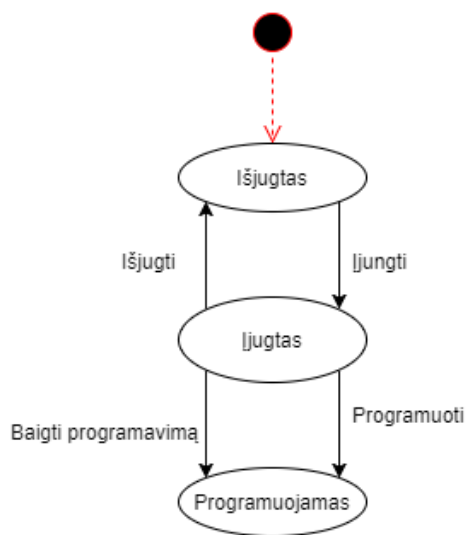
- Teritorijos žimėjimo seka:



19 pav

#### 2.4.2. Būsenų diagramos

- Detektoriaus būsenų diagrama:

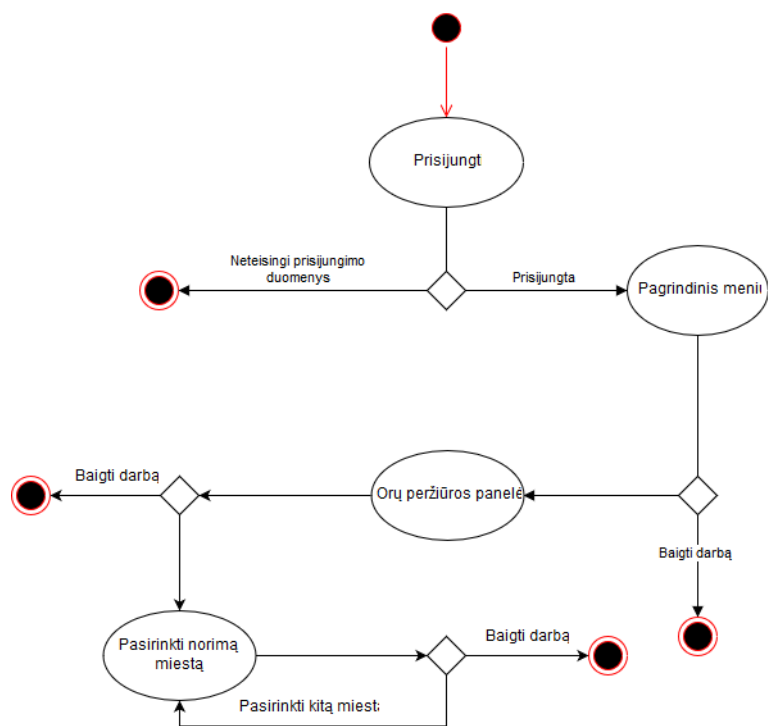


20 pav

#### 2.4.3. Veiklos diagramos

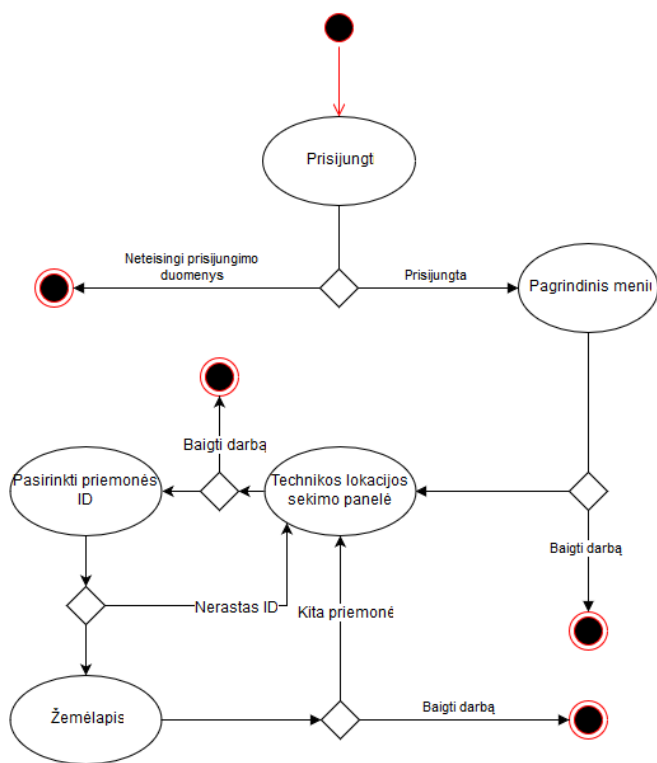
asfgh

- Orų sekimo veiklos diagrama:



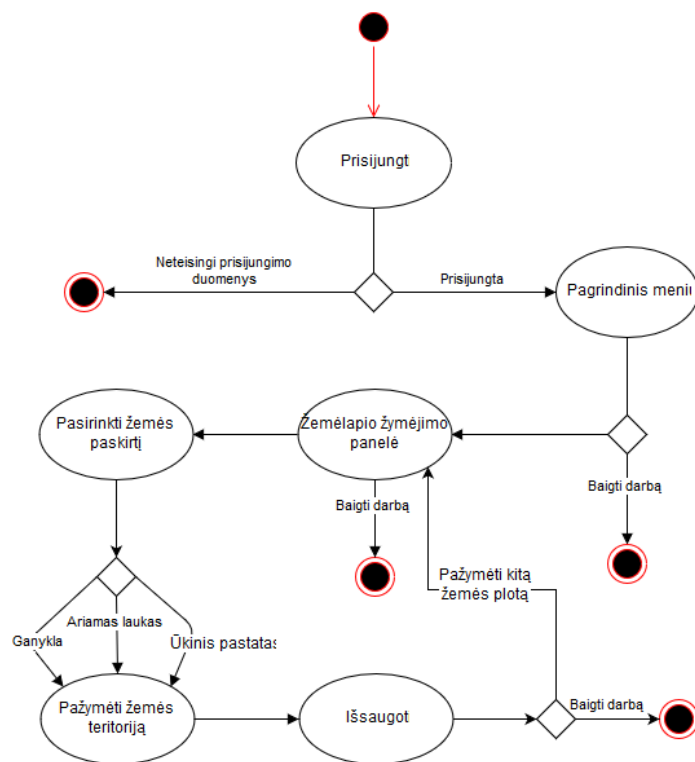
21 pav

- Ūkio technikos sekimo veiklos diagrama:



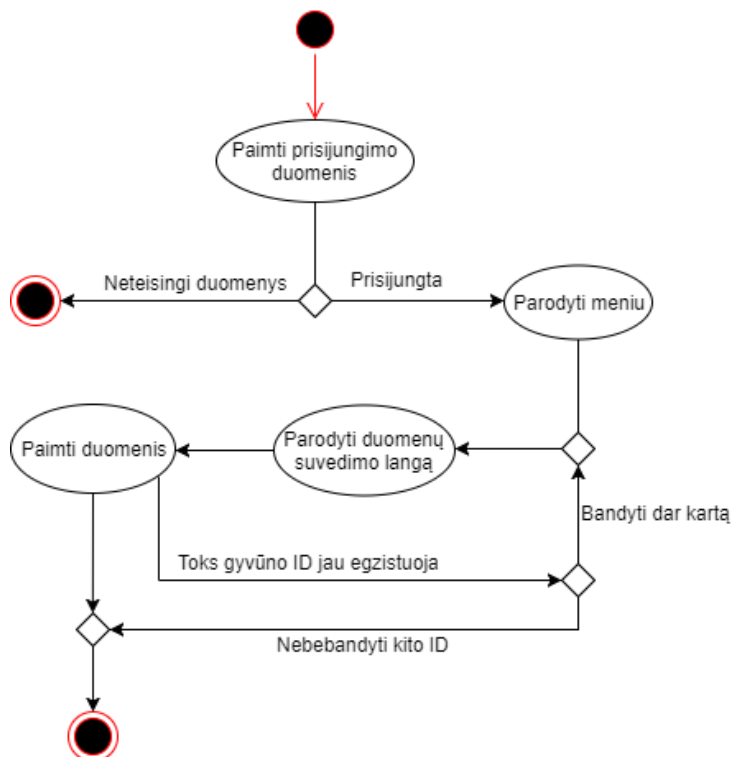
22 pav

- Teritorijos žymėjimo veiklos diagrama:



23 pav

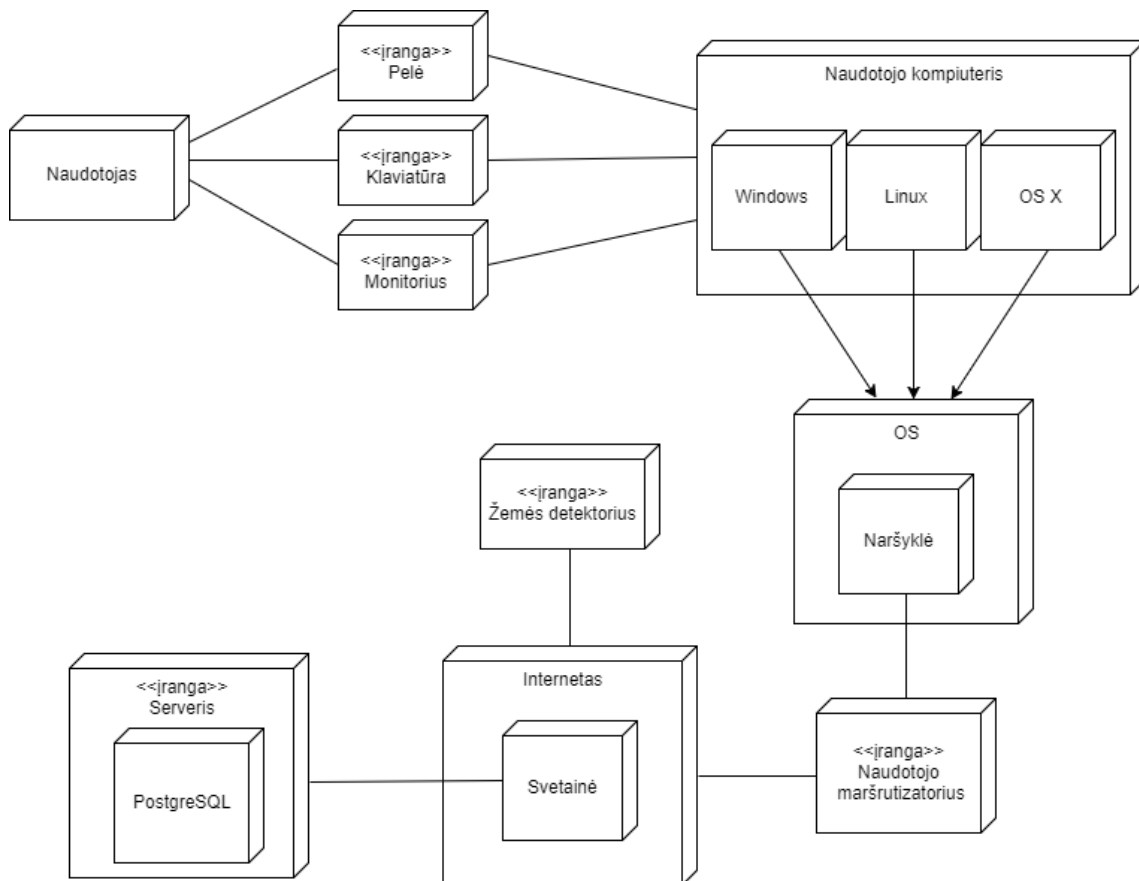
- Gyvūnų suvedimo veiklos diagrama:



24 pav

## 2.5. Fizinis pjūvis

Šiame skyriuje parodoma programos naudojama aparatinė įranga, komunikacija tarp tinklo mazgų bei programos komponentų išdėstymas juose.



25 pav

- D0: Šioje diagramoje (17 pav.) parodyta, kaip išsaugomi programos duomenys. Pirmoje projekto versijoje visi duomenys buvo saugomi kompiuteryje, vienintelis ryšys su internetu buvo per gismeteo.lt svetainę. Dabartinėje versijoje implementuotas duomenų saugojimas PostgreSQL duombazėje. Šioje diagramoje vaziduojamas kelias nuo naudotojo iki pasirinktos duombazės, naudojami techniniai įrenginiai ir jų sąryšiai su programine įranga. Taip pat pridėtas žemės detektorius, kuris siūčia programai pasirinkto žemės ploto parametrų duomenis. Informacija, gauta iš žemės detektoriaus, taip pat talpinama PostgreSQL duombazėje.

## **Rezultatai ir išvados**