

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 case	9
1.4. Proceso pjūvis	9
1.4.1. Veiklos diagramos.....	9
1.4.2. Būsėnų diagrama	9
1.5. Fizinis pjūvis	10
2. PERPROJEKTUOTOS SISTEMOS APRAŠYMAS(TO-BE, V2.0)	11
2.1. Loginis pjūvis	11
2.2. Kūrimo pjūvis	11
2.3. Use case	11
2.4. Proceso pjūvis	11
2.5. Fizinis pjūvis	11
REZULTATAI IR IŠVADOS	12

Išvadas

Automatinė ūkio valdymo sistema (toliau Auto ūkis) - yra programa leidžianti ūkininkui valdyti jo ūki skaitmeniniu būdu. Auto ūkis leidžia stebėti kiekvieno individualaus gyvūno bioparametrus (kraujo spaudimą, svorį, sveikatą), ūkio technikos judėjimą po žemės plotą, gyvūnų registracija. Taip pat sistema vartotojui leidžia sekti dirvos parametrus (drėgmę, pH lygį), oro prognozes ir aplinkinių teritorijų gyvūnų ligų paplitimą. Auto ūkis padeda ir su verslo valdymu, nesunkiai galima samdyti darbuotojus, atlikti buhalterinę apyskaitą, stebėti rinkos kainas ir skaičiuoti bei numatyti galimą pelną. Iškilus nelaimės per Auto ūkio sistemą galima greitai išskviesti greitąją pagalbą, policiją, gaisrinę ar saugos tarnybą. Orai prognozės yra paimtos iš www.gismeteo.lt. Pagrindinė sistemos inovacija yra tai, kad kai sistema yra pilnai įdiegta darbuotojų skaičius palaikyti ūki tampa minimalus. Ūkio technika būtų valdoma automatiškai todėl vairuotojų ir derliaus nurikėjų nereiktų. Gyvūnų sekimas yra įgyvendinamas mikrokontrolerio Arduino pagalba. Kadangi šis kontroleris ledidelis ir lengvai pritaikomas visiokio pobūdžio darbams ji kartu su WiFi modulių sistema naudoja gauti gyvūno lokaciją per Google maps. Taip pasiklydę ar pavogti gyvūnai būtų greitai gražinami surandami ir gražinami. Žemės laistymas, 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 nemokamos buhalterijos programos Wave Accounting kuri yra implementuota į Auto ūki. Auto ūkio sistema yra parašyta JAVA kalba kas leidžia programą paleisti ant bet kurios operacinės sistemos. Ateityje numatoma galimybė programą perkelti į išmaniuosius telefonus. Sistema buvo projektuojama pasitelkiant www.planttext.com ir 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 ūkinius
- * ŪkioTechnika - laiko ūkio technikos charakteristikos reikšmes. Apskaičiuoja technikos judėjimo greitį.
- * Žemės parametrai - saugo įvairius žemės parametrus(drėgmė, ph...).
- * Orai - klasė skirta pasiimti orų prognozes iš www.gismeteo.lt kurių paprašo vartotojas.
- * Žemės detektorius - klasė skirta bendrauti su žemės detektoriumi

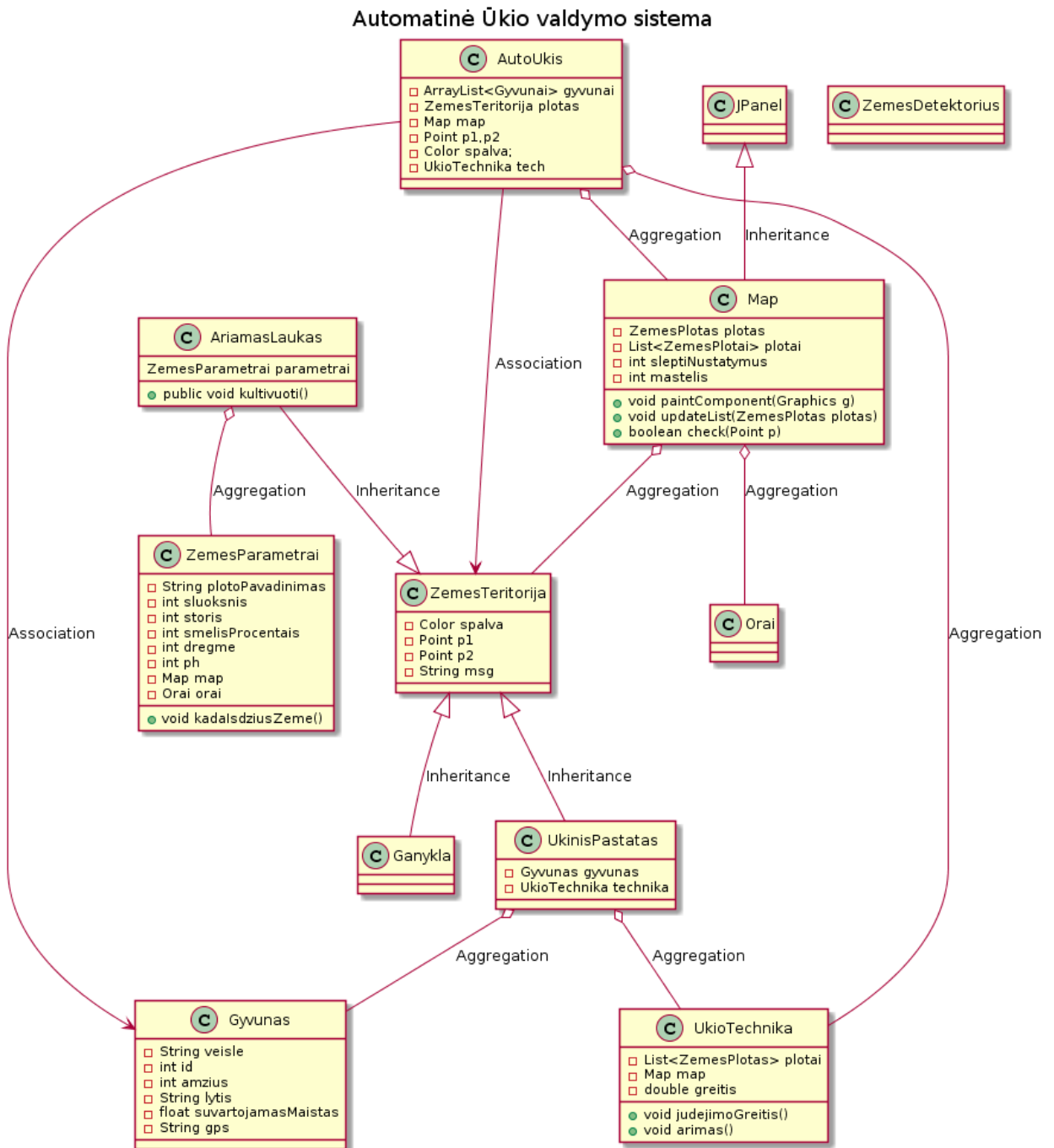
- Bendri terminai:

- * Žemės plotas - vieta kurią valdo ir gali stebėti vartotojas(ūkininkas)
- * Detektorius - Arduino mikro kontroleris
- * Ūkininkas - žmogus kurio valdomoje teritorijoje įdiegtas Auto ūkis
- * Arduino - mikrokontroleris skirtas ūkio sekimui
- * Automašiskai valdoma - 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 minėta PlnatText programa.



1 pav

- Dizainas:

- Pagrindinė klasė yra AutoUkis.form. Joje sukurtas ir aprašytas Graphical User Interface (GUI), 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.

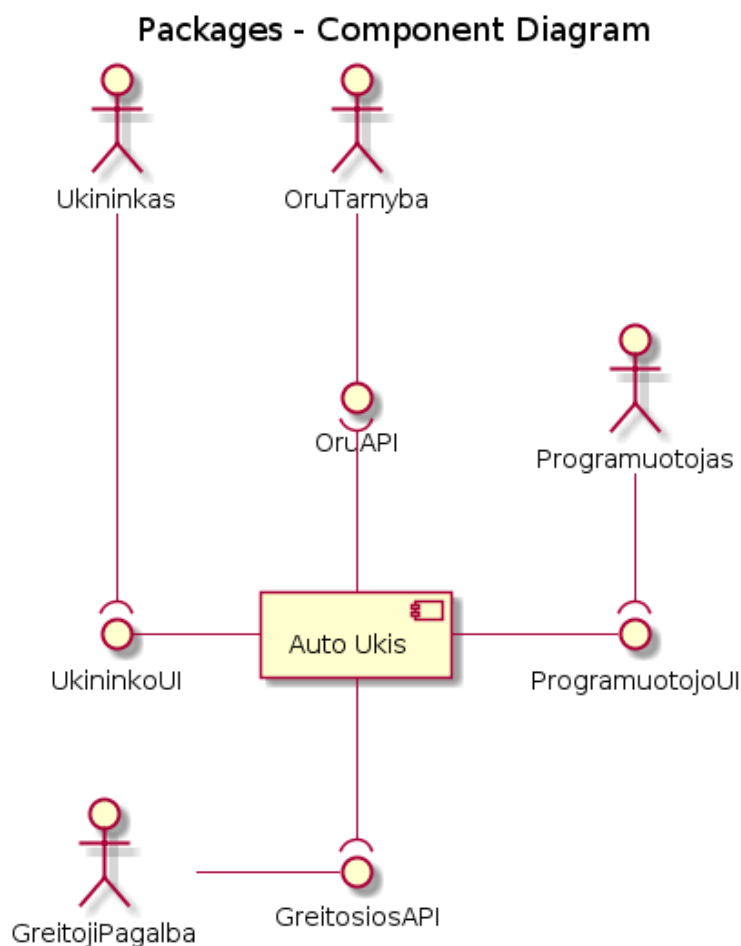
- Funkcionalumas:

- 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:

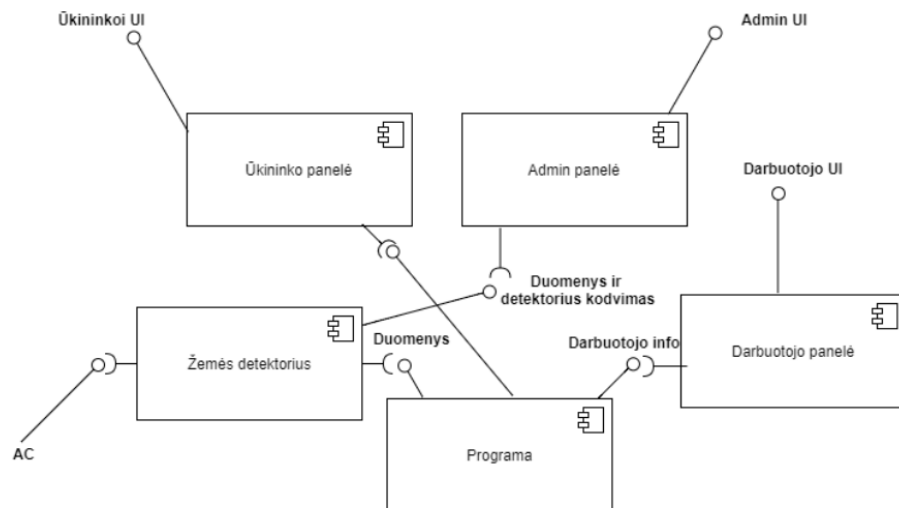
- Pradėjus rašyti programą nepagalvojome apie kūrimo pjūvį ir kaip teisingiau būtų pradėti viską. Žiūrint dabar visa programa buvo pradėta kurti pagal Bottom -> Up principą. Iš pradžių apsirašėme daugybę mažų klasių ir paskui jas bandėme apjungti į didesnę sistemą. Išskyrėme tokius komponentus kaip ūkininkas, 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

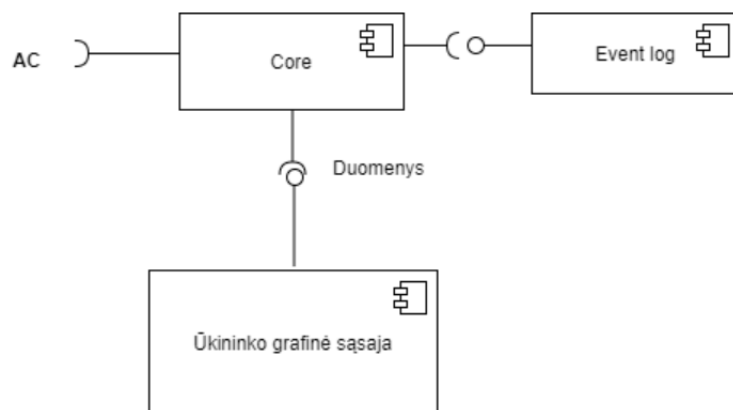
- L0:

- Šioje diagramoje pavaizdavome sistemos bendravimą su išoriniais agentais tokiais 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.



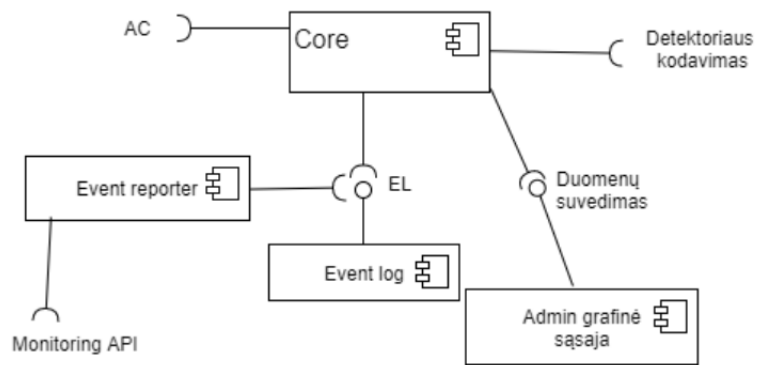
3 pav

- L1: Sudėjus komandos idėjas apie tai, kaip turėtų atrodyti *L1 diagrama*, 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 kas greičiausiai yra nekorektiška.



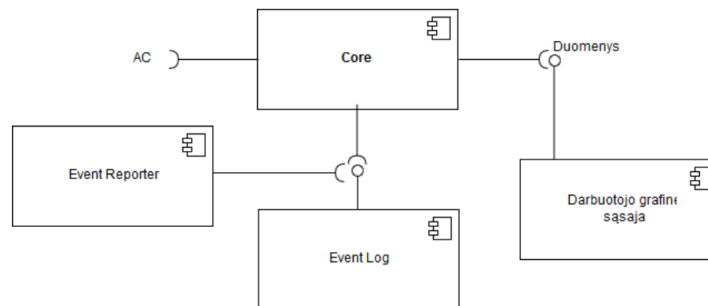
4 pav

- L2(Ūkininkas): Šioje diagramoje parodyta, kad programos pagrindas kuria suteikia interfačą grafinei vartotojo sąsajai. Paduoti duomenys yra užregistruojami Event Log'e.



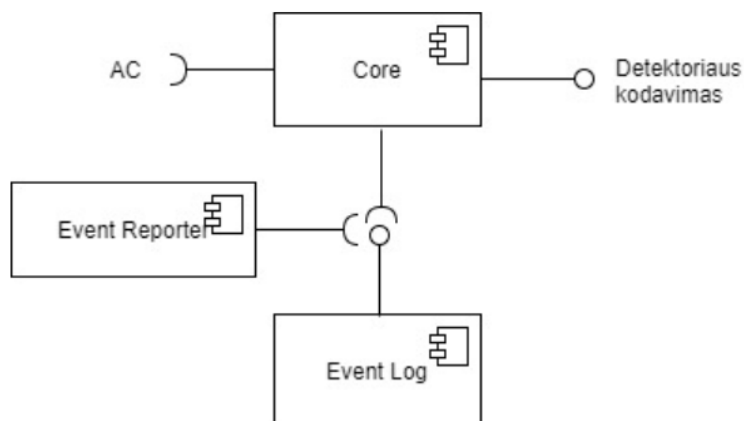
5 pav

- L2(Admin) Šioje diagramoje 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 parodyta, kad programos pagrindas grafinę sąsają ir perduoda duomenis į event log'ą.



7 pav

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

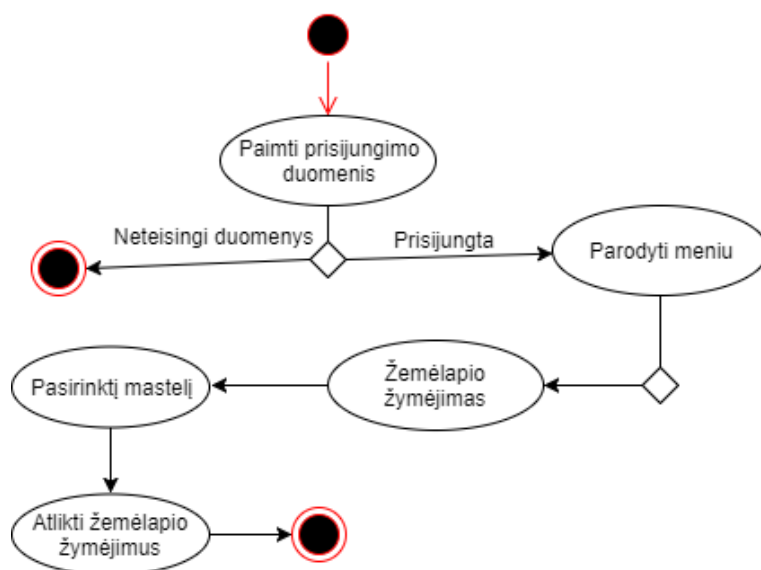
1.3. Use case

1.4. Proceso pjūvis

Šiame skyriuje parodoma programos elgsena jos vykdymo metu.

1.4.1. Veiklos diagramos

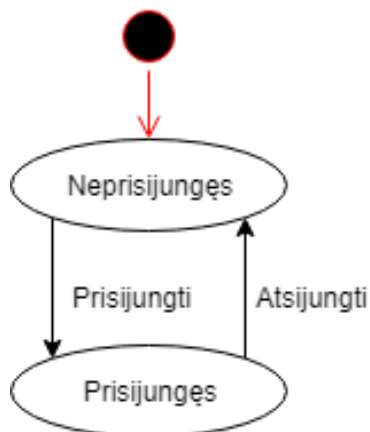
Vaizduojama žemėlapių žymėjimo veiklos diagrama. Nurodomi pagrindiniai žingsniai braižant žemėlapi.



8 pav

1.4.2. Būsėnų diagrama

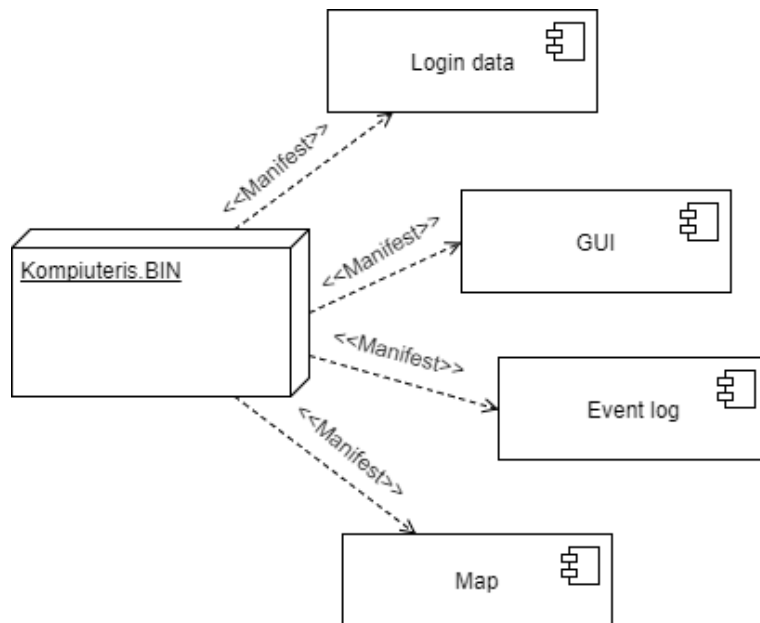
Šioje diagramoje parodomos galimos vartotojo būsenos ir keliai kaip tas būsenas pasiekti. Šiuo metu programoje tėra 2 būsenos, taigi diagrama yra labai paprasta.



9 pav

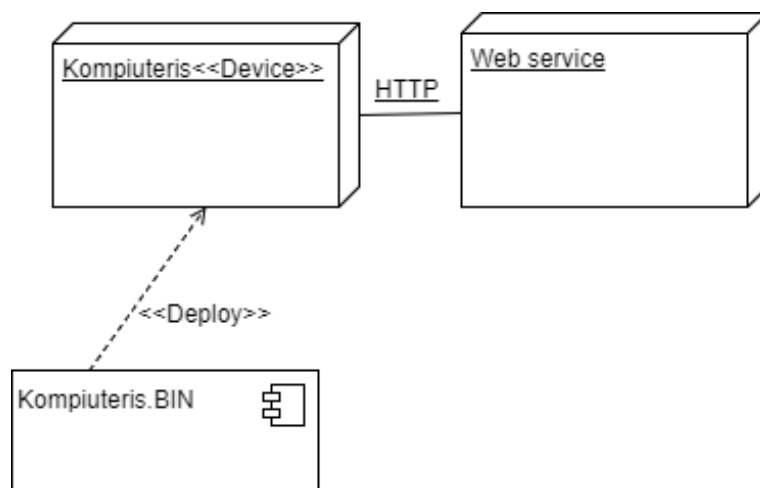
1.5. Fizinis pjūvis

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



10 pav

- D0: Šioje diagramoje 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.



11 pav

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

1.6. Pirmos dalies išvada

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

Rimtesniai sistemai kurti reiktu naudotis Top to bottom principu lengvesniam naujų funkcionalumų pridėjimui. Stipri ir paprasta sistemos dalis pasimato kūrimo pjūvyje kur aiškiai kuriami ir įgyvendinami, lengva kurti ar įgyvendinti naujus interfeisus. Aprašinėdami L1 ir L2 supratome, kad nepagalvojome apie tai kaip sistemos vartotojai bendrauja su sistema. Šią dalį reiks pergaltvoti antroje programos versijoje kad viskastaptu aiškiau ir paprasčiau, labiau pasidomėti, kaip tokios sistemos veikia realiame pasaulyje. Proceso pjūvį pavyko aprašyti pakankamai paprastai ir aiškiai, tačiau jam dar trūksta detalumo ir kitų scenarijų apmatymo, pavyzdžiui prisijungti prie sistemos. Fiziniame pjūvyje aprašyta techninė sistema yra ganėtinai paprasta ir primityvi. Pagrindinis trūkumas, nepagalvota kas nutiktu kompiuterio išsijungimo atveju ar kas nutiktu atsiradus poreikiui plėsti sistemą. Žiūrint bendrais idėjos bruožais brangiausia sistemos dalis yra automatinės mašinos, kurios dar yra salyginai nauja technologija ir mikro kontrolieriai kiekvienam gyvūnui. Jeigu ferma turi jų daug visas instaliavimas kainuotų salyginai brangiai ir turbūt neatneštų didelės naudos. Tai labiau inovacija dėl inovacijos, ne dėl funkcionalumo. Tačiau kitos dalys visai sėkmingai pritaikomos. Tokią sistemą kaip padarėme dabar įgyvendinti būtų įmanoma, tačiau praplėti ir palaikyti ja būtų nepatogu ir turbūt sistema nedirbtų taip greitai kaip norėtūsi. Trūksta funkcionalumo su išnamiuoju telefonu, darbo birža, rinkos tendencijos ir kitą.

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

2.1. Loginis pjūvis

2.2. Kūrimo pjūvis

2.3. Use case

2.4. Proceso pjūvis

2.5. Fizinis pjūvis

Rezultatai ir išvados