

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	5
1.3. Use case	9
1.4. Proceso pjūvis	9
1.5. Fizinis pjūvis	9
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

Ivadas

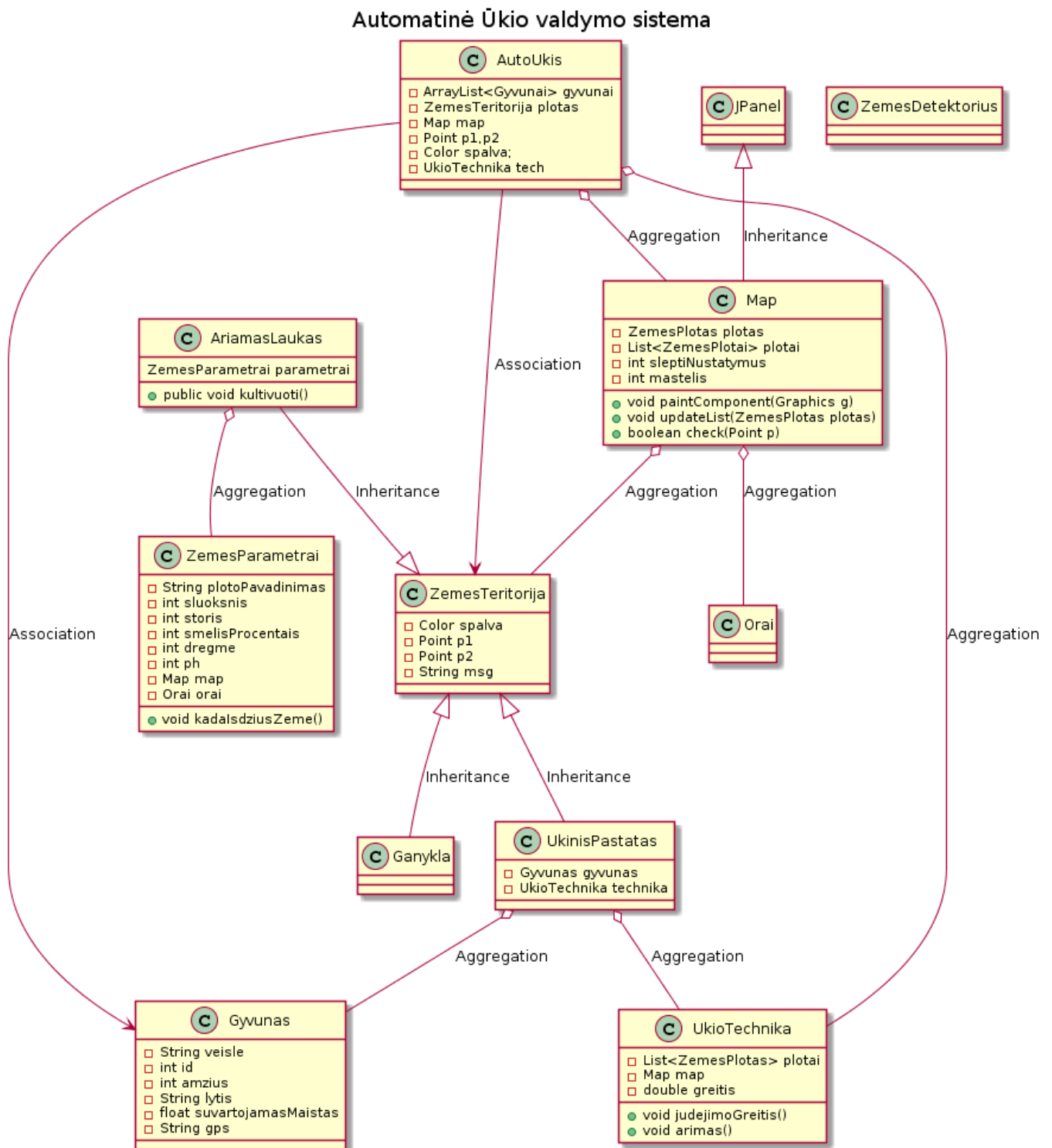
Pasirinkome programuoti Automatinę ūkio valdymo sistemą (toliau - Auto ūkis) JAVA programavimo kalba, nes visi komandos nariai turėjo pakankamas žinias programuoti šia kalba ir kalbos funkcionalumas, mūsų manymu, atitiko poreikius. Programuodami pirmą sistemos variantą stengėmės prisilaikyti bendrą objektinio programavimo principų ir JAVA kodo standarto. Apie patį sistemos projektavimą pagalvojome minimaliai dėl tuo metu neturimų žinių. UML diagramų braižymui naudojome www.planttext.com (toliau - PlantText) ir www.draw.io.

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)

1. Sukurtos sistemos aprašymas(v1.0)

1.1. Loginis pjūvis



Suprogramavę šabloninį programos karkasą nubraižėme UML diagramą pasinaudoja minėta PlantText programą.

- Dizainas:
 - Programos pagrindas yra AutoUkis.form klasė kuri pasiima duomenis iš kitų klasių. O duomenis įveda patys vartotojai. Rašant kodą stengėmės prisilaikyti objektinio programavimo principų. GUI piešimas ir pasirinkimas vyko AutoUkis klasėje kurį pasiima duomenis iš kitų klasių. Daugumos objektų kūrimas ir laikymas taip pat yra šioje klasėje. Visos kitos klasės turėjo savo

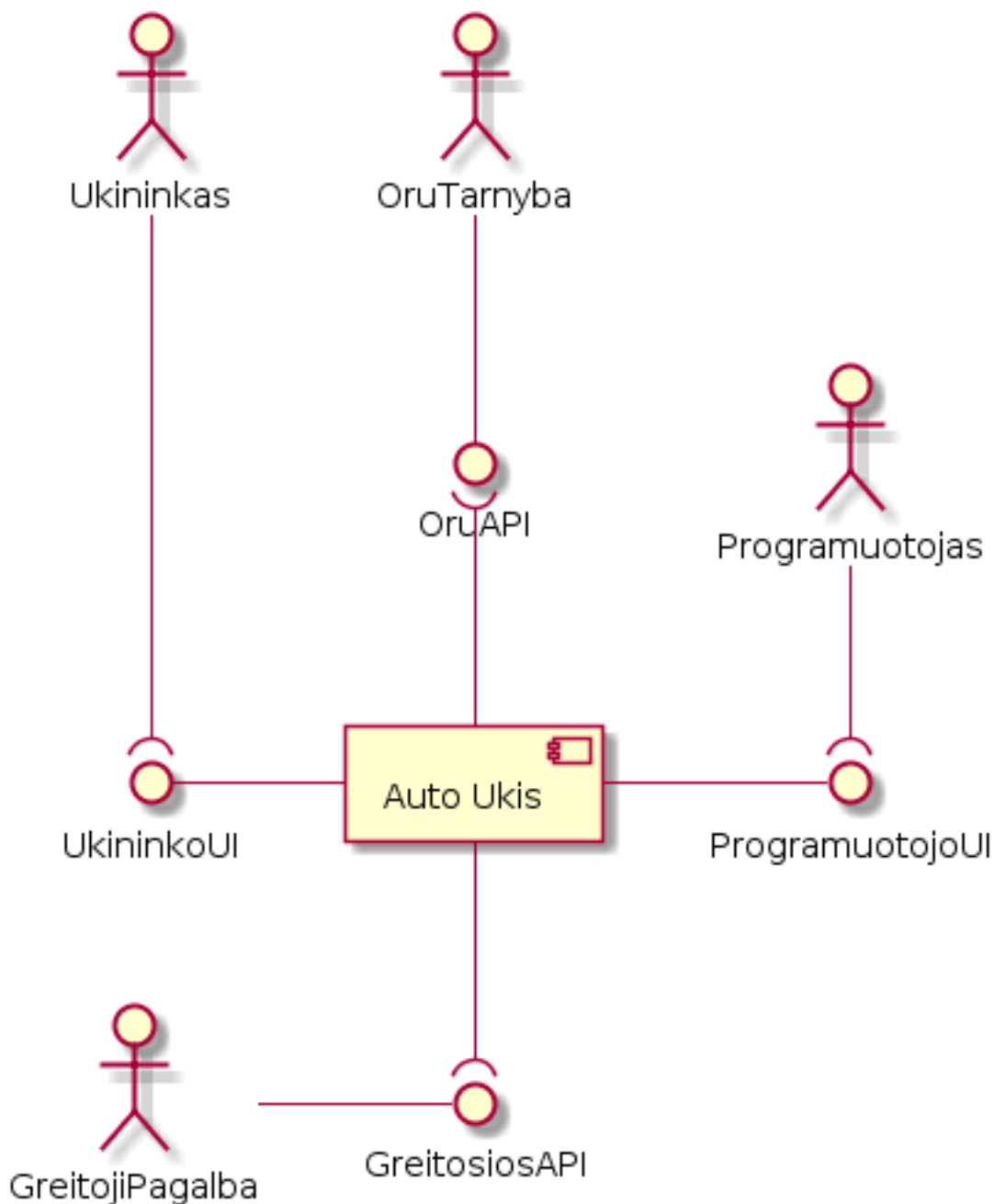
atskiras paskirtis(Gyvūnas - saugojo duomenis, Map - piešė žemėlapi). Kai kurios klasės(pvz. : ZemesDetektorius) buvo sukurtos dėl dizaino tolimesnio jų panaudojimo, bet nebuvo niekur pritaikytos.

- Funkcionalumas:

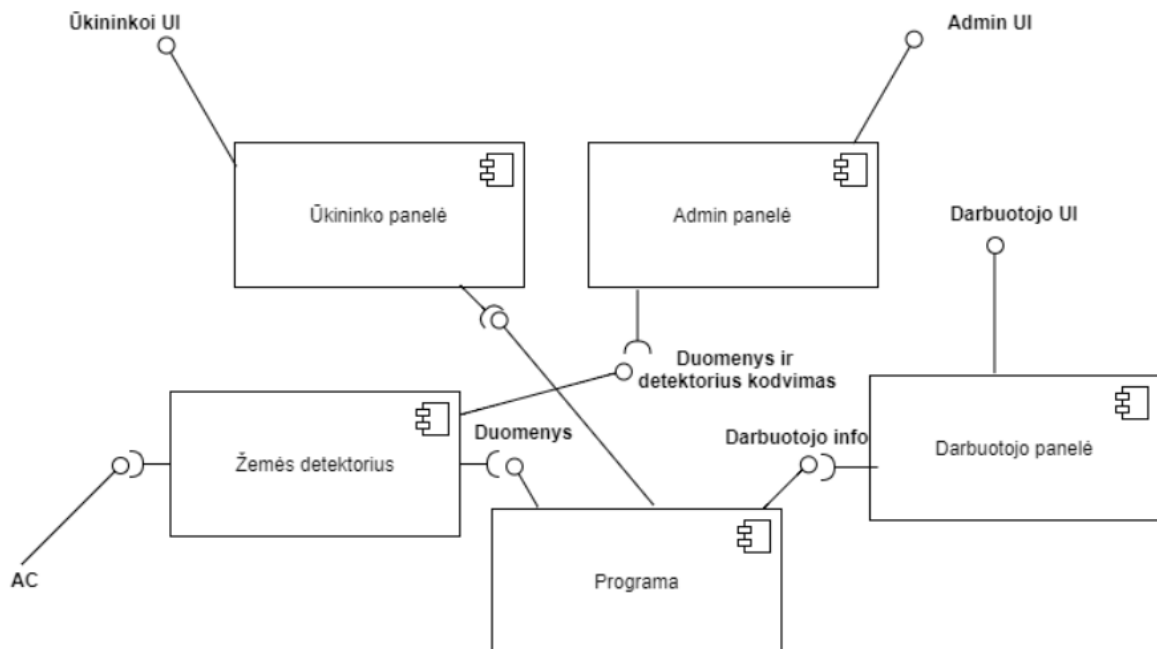
- Viso užsibrėžto programos funkcionalumo įgyvendinti nepavyko. Kai kurios klasės buvo sukurtos ateityje planuojamoms funkcijoms. Programa kol kas veikia tik ant kompiuterio ir vienintelis jos bendravimas su internetu yra per Orai klasę. Yra funkcija surašyti gyvūnus ir duomenis apie juos ir žymėti bei atlikti primityvią žemės kategorizaciją.

1.2. Kūrimo pjūvis

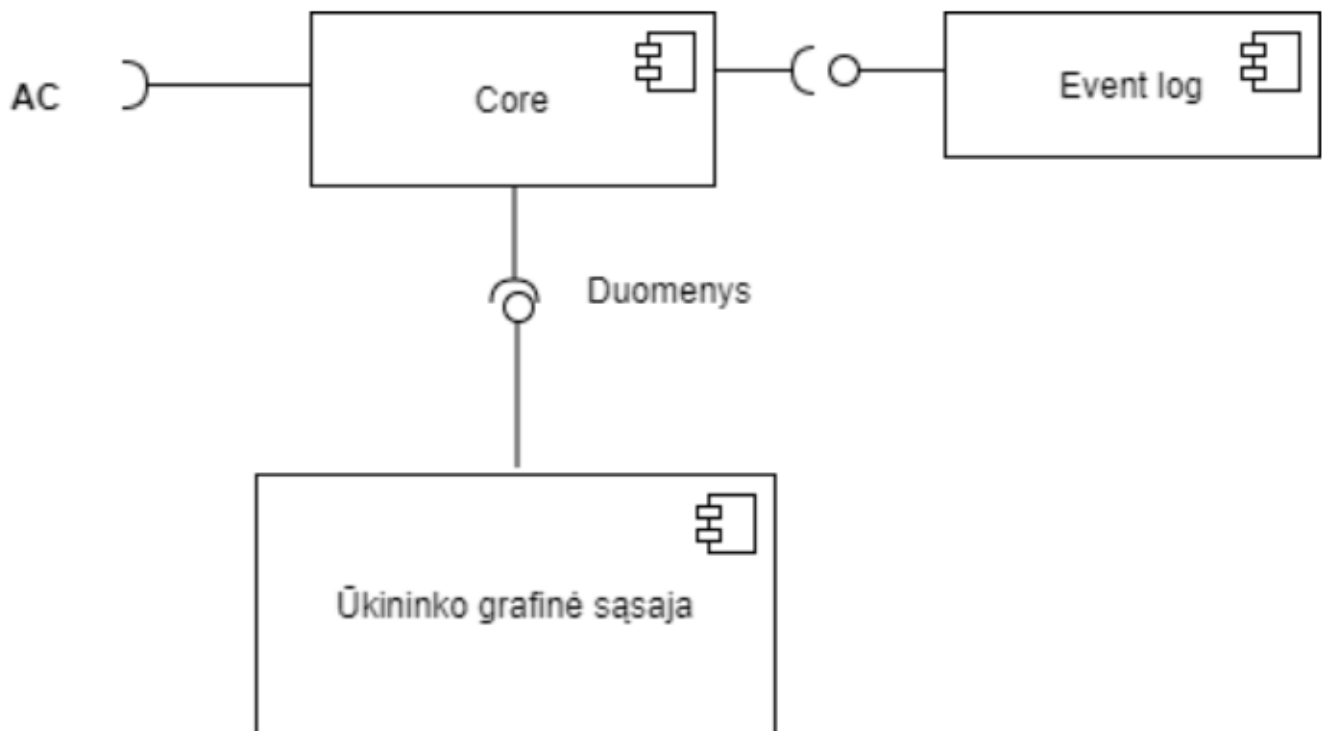
Packages - Component Diagram



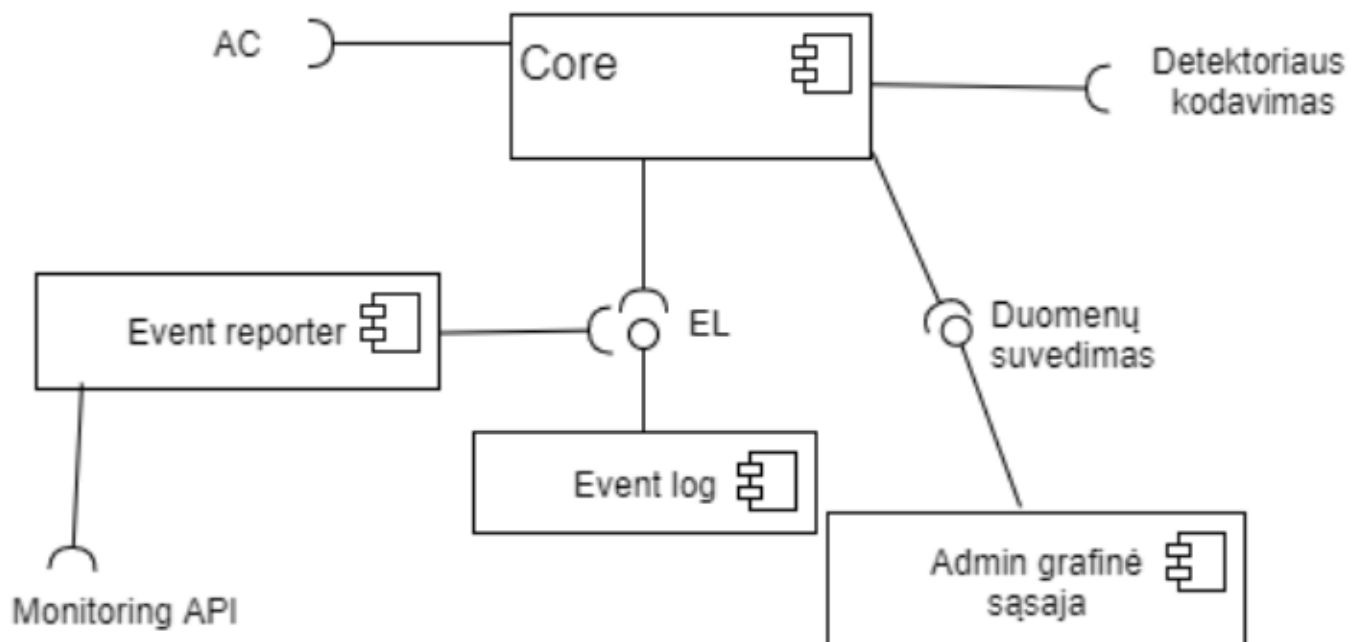
- Dizainas:
 - Pradėjus rašyti programą nepagalvojome apie kūrimo pjūvį ir kaip teisingiau būtų pradėti viską. Žiūrint dabar visa programa buvo pradėta kurti pagal Bottom -> Up principą. Iš pradžių apsirašėme daugybę mažų klasių ir paskui jas bandėme apjungti į didesnę sistemą. Kai kurios klasės liko nepanaudotos.
- L0:
 - Šioje diagramoje pavaizdavome sistemos bendravimą su išoriniais agentais tokiais kaip Greitoji pagalba, Ūkininkas ir t.t. . Ši diagrama aiškiai ir paprastai parodo kuriamus ir įgyvendinamus interfeisus. Galbūt būtų galima Greitosios Pagalbos interfeisą išskaidyti į kelis detalesnius interfeisus, bet apskritai didelių problemų nepastebime.



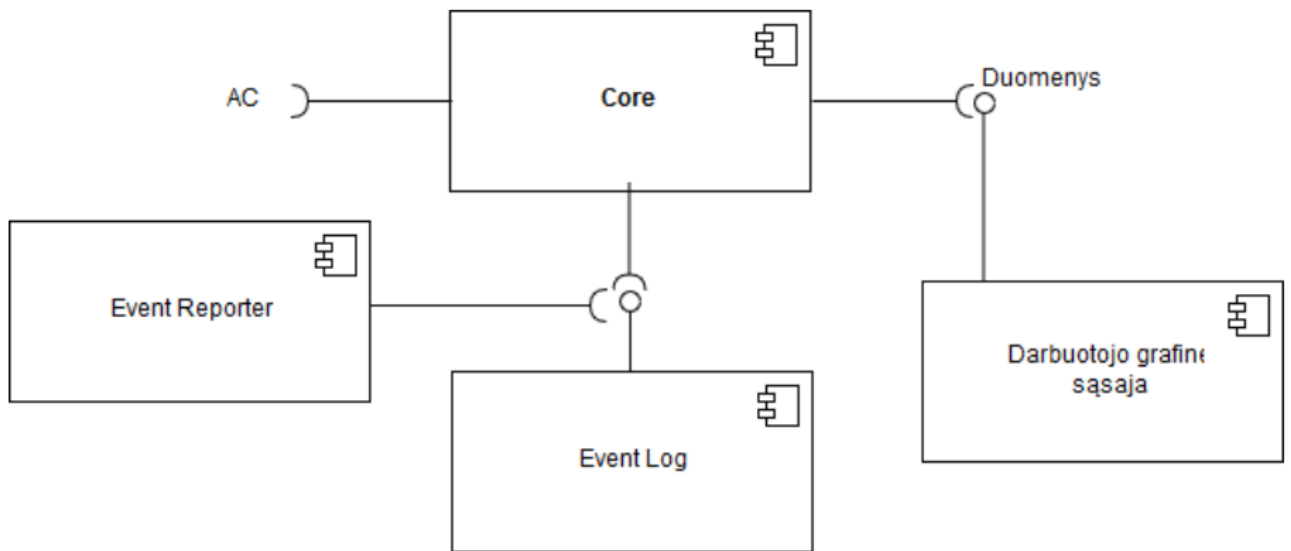
- L1: Sudėjus komandos idėjas apie tai, kaip turėtų atrodyti L1 diagrama, supratome, kad mūsų sistema neturi normalios struktūros ir gerai nebuvo pagalvoję kaip visi komponentai sietis vieni su kitais, todėl ir diagrama atrodo chaotiška. Trūksta konkretumo kaip turi Admin sietis su kitais komponentais.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



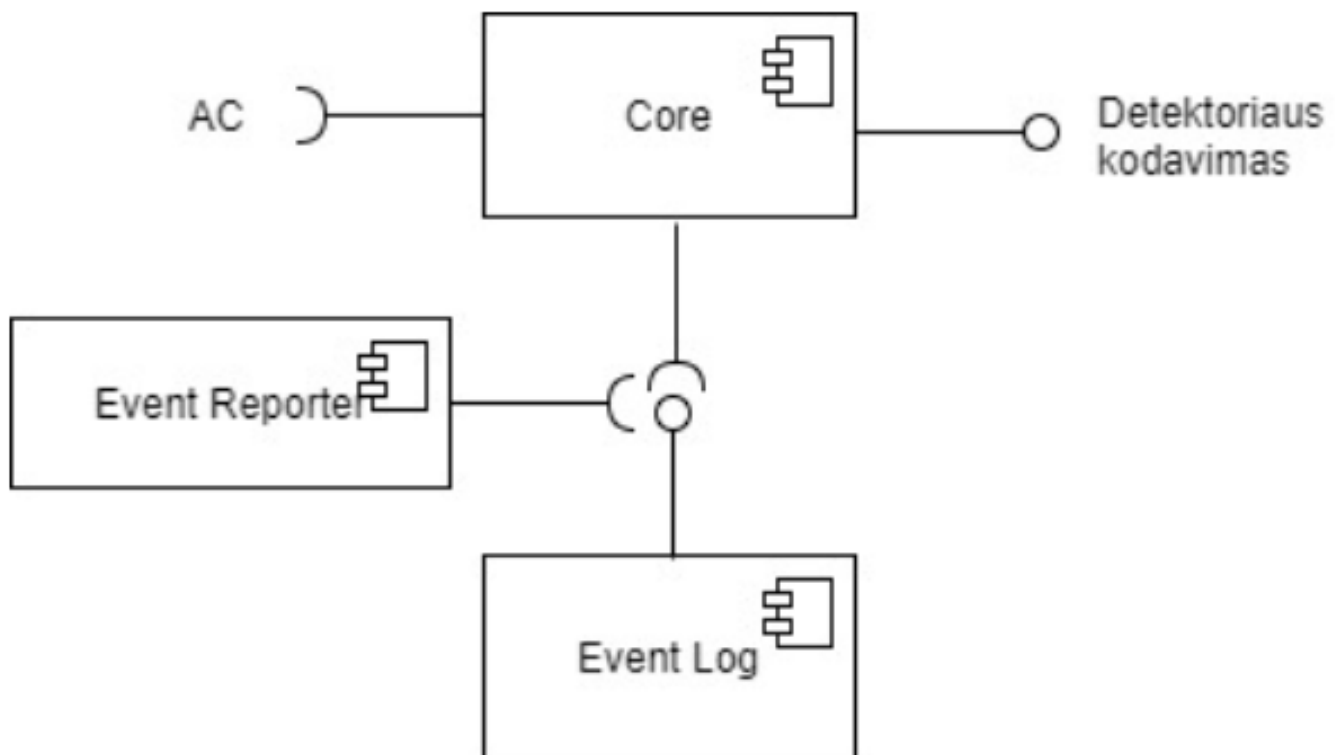
- L2(Ūkininkas): Šioje diagramoje parodyta, kad programos pagrindas kuria suteikia interface'ą grafiinei vartotojo sąsajai. Paduoti duomenys yra užregistruojami Event Log'e.



- L2(Admin): Šioje diagramoje parodyta, kad programos pagrindas naudoja duomenų suvedimo interfacę, kurią suteikia admin grafinė sąsaja, bei naudoja Detektoriaus kodavimo interfacę. Visus įvykius įrašo į event log



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

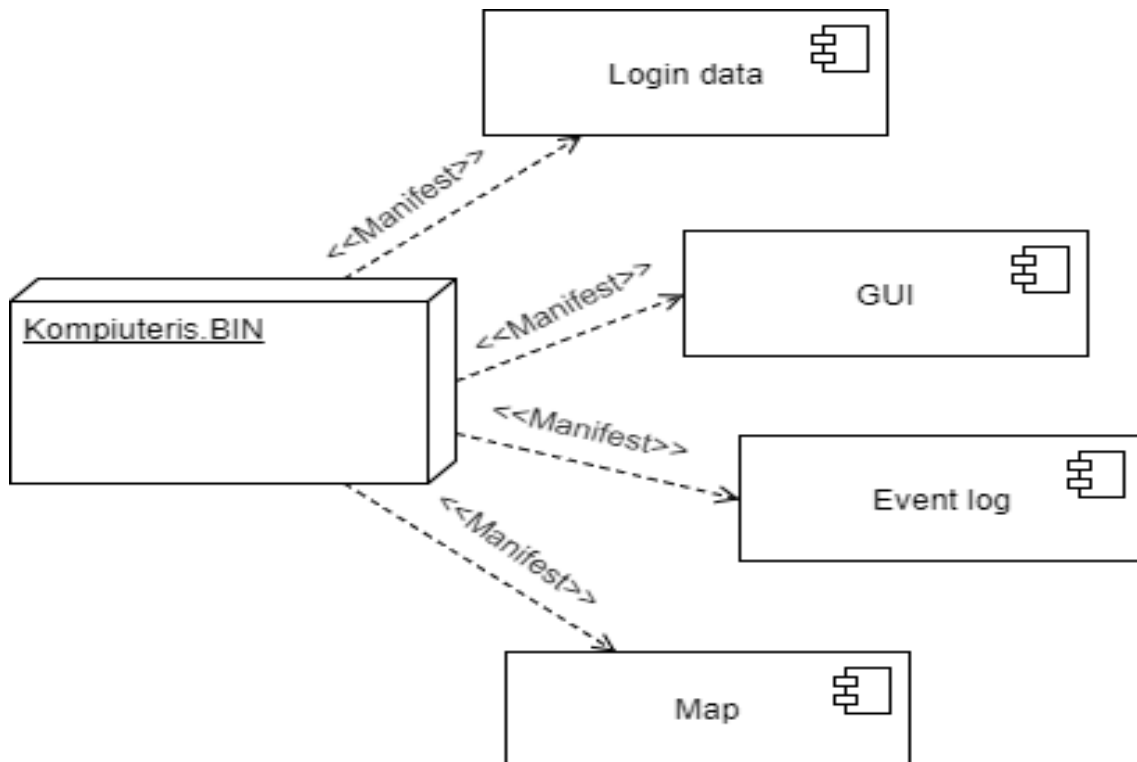


- L2(Detektorius): ši diagrama vaizduoja detektoriaus išvedamus duomenis. Įvykiai įrašomi Event Log'e.

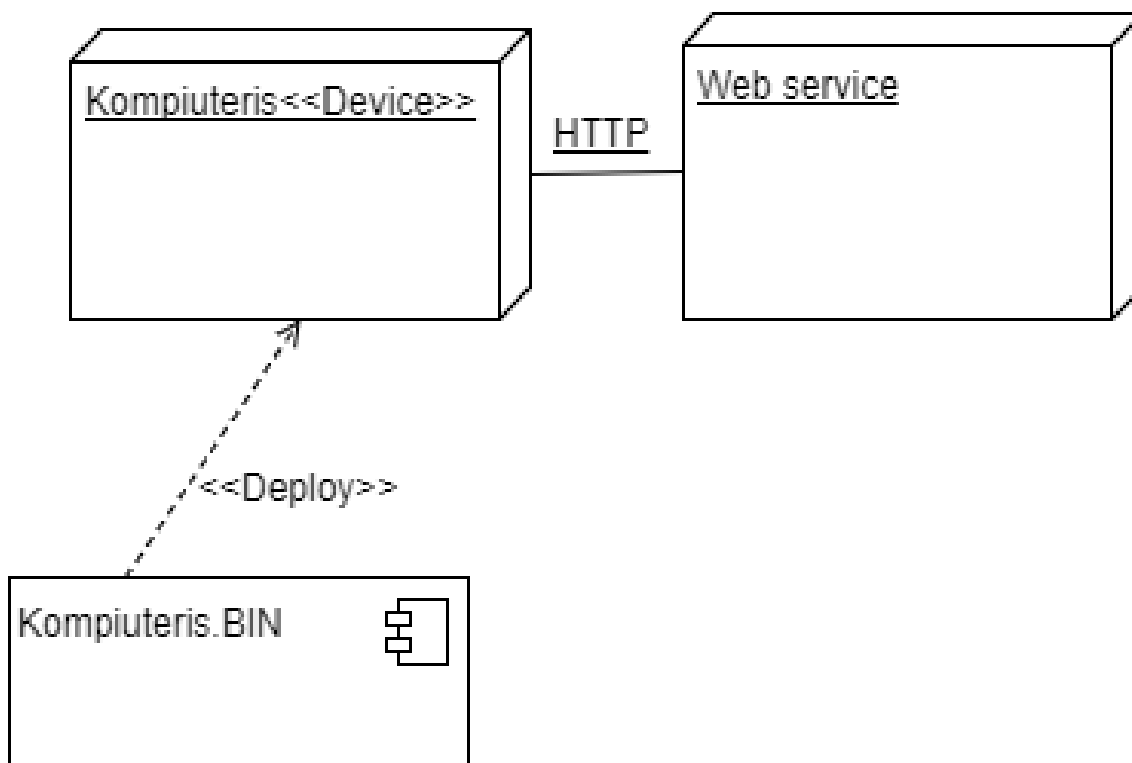
1.3. Use case

1.4. Proceso pjūvis

1.5. Fizinis pjūvis



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



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

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