



Kauno technologijos universitetas

Informatikos fakultetas

Sveikos gyvensenos planavimo internetinė programa

Baigiamasis bakalauro studijų projektas

Tautvydas Dikšas

Projekto autorius

prof. Eduardas Bareiša

Vadovas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

Sveikos gyvensenos planavimo internetinė programa

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

Tautvydas Dikšas

Projekto autorius

prof. Eduardas Bareiša

Vadovas

doc. Recenzentas Recenzaitis

Recenzentas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

Tautvydas Dikšas

Sveikos gyvensenos planavimo internetinė programa

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Studentės Studentaitės, baigiamasis projektas tema „Sveikos gyvensenos planavimo internetinė programa“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatyty piniginių sumų už šį darbą niekam nesu mokėjės.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę išrašyti ranka)

(parašas)

Tautvydas Dikšas. Sveikos gyvensenos planavimo internetinė programa. Bakalauro studijų baigiamasis projektas / vadovas prof. Eduardas Bareiša; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: laiko planavimas, internetinė, mobilioji programa.

Kaunas, 2021. 83 p.p.

Santrauka

Šiame darbe pristatoma sveikos gyvensenos planavimo sistema. Projektas gali padėti efektyviau planuoti savo laiką žmonėms, kurių gyvenimo tempas greitas ir įtemptas. Kuriama sistema yra aktuali dėl galimybės stebeti savo veiklos informaciją. Taigi, pagrindinis darbo tikslas - palengvinti naudotojo asmeninės veiklos stebeseną.

Analizujant konkurentų technologijas, buvo išskiriami privalumai ir trūkumai, lyginamas dabartinis funkcionalumas. Surinkta informacija iškėlė idėją, kad dabartinės populiarusios sistemos yra koncentruotos į minimalų funkcionalumą, taip išlaikant sistemą paprastą naudotojui. Darbo metu buvo parengtas gyvensenos planavimo sistemos projektas.

Sistema buvo realizuota naudojant Javascript ir Typescript programavimo kalbas, Vue.JS, Express.JS bei Nativescript karkasus. Sistemos programiniam kodui buvo parašyti unit testai, kurie leido užtikrinti kodo paleidimo sklandumą. Realizuotas detalus naudotojo vadovas, leidžiantis naujiems klientams lengviau naviguoti.

Tautvydas Dikšas. Health planner web app. Bachelor's Final Degree Project / supervisor prof. Eduardas Bareiša; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: time planning, web app, mobile app

Kaunas, 2021. 83 p.p.

Summary

This project is a healthy lifestyle planning system. It may help people effectively plan their daily activities, especially those who struggle with fast-paced lifestyle. The system is relevant as it provides many possibilities to monitor your activities. The main goal of this project is to simplify the user's activities monitoring.

By analysing competitors' technologies, advantages and disadvantages were determined. The conclusion was made, that most popular planning systems strive for minimalistic functionality, thus keeping simplistic design for the user.

The system was made using JavaScript and Typescript programming languages Vue.JS, Express.JS, and Nativescript frameworks. Unit tests were made for the system's code which ensured that the code works smoothly. Detailed user manual was made to help new clients navigate the system easily.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpū ir terminų žodynas	11
1. Analizė.....	13
1.1. Techninis pasiūlymas	13
1.1.1. Sistemos apibrėžimas	13
1.1.2. Bendras veiklos tikslas	13
1.1.3. Sistemos pagrįstumas	13
1.1.4. Konkurencija rinkoje	14
1.1.5. Prototipai ir pagalbinė informacija	15
1.1.6. Ištekliai, reikalingi sistemai sukurti	15
1.2. Galimybų analizė	15
1.2.1. Techninės galimybės	15
1.2.2. Naudotojų pasiruošimo analizė	15
2. Projektas	16
2.1. Reikalavimų specifikacija	16
2.1.1. Komercinė specifikacija	16
2.1.2. Sistemos funkcijos	17
2.1.3. Naudotojo sąsajos specifikacija	48
2.1.4. Realizacijai keliami reikalavimai	48
2.1.5. Techninė specifikacija	48
2.2. Projektavimo metodai	48
2.2.1. Projektavimo valdymas ir eiga	48
2.2.2. Projektavimo technologija	49
2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos ..	49
2.3. Sistemos projektas	49
2.3.1. Statinis sistemos vaizdas	49
2.3.2. Dinaminis sistemos vaizdas	52
3. Testavimas	71
3.1. Testavimo planas	71
3.2. Testavimo kriterijai	71
3.3. Naudotojo sąsajos testavimas	72
3.3.1. Automatiniai testai	72
3.3.2. Rankiniai testai	72
4. Dokumentacija naudotojui	74
4.1. Apibendrintas sistemos galimybų aprašymas	74
4.2. Naudotojo vadovas	74
4.2.1. Internetinė programa	74
4.2.2. Mobilė programa	77
4.3. Diegimo vadovas	82
4.4. Administravimo vadovas	82

Rezultatai ir išvados	83
------------------------------------	-----------

Lentelių sąrašas

1 lentelė. Konkurentų apžvalga	14
2 lentelė. PA „Skaičiuoti miegą” specifikacija.....	19
3 lentelė. PA „Registruotis” specifikacija	20
4 lentelė. PA „Prisijungti” specifikacija	21
5 lentelė. PA „Atsijungti” specifikacija	22
6 lentelė. PA „Redaguoti profili” specifikacija	23
7 lentelė. PA „Trinti profili” specifikacija	24
8 lentelė. PA „Peržiūrėti veiklos sąrašą” specifikacija	24
9 lentelė. PA „Kurti veiklą” specifikacija	26
10 lentelė. PA „Redaguoti veiklą” specifikacija	27
11 lentelė. PA „Trinti veiklą” specifikacija	28
12 lentelė. PA „Siųsti pranešimą” specifikacija	29
13 lentelė. PA „Peržiūrėti treniruočių sąrašą” specifikacija	30
14 lentelė. PA „Kurti treniruotę” specifikacija	33
15 lentelė. PA „Redaguoti treniruotę” specifikacija	34
16 lentelė. PA „Trinti treniruotę” specifikacija	34
17 lentelė. PA „Peržiūrėti PopoPlan pratimų sąrašą” specifikacija	35
18 lentelė. PA „Peržiūrėti wGer pratimų sąrašą” specifikacija.....	36
19 lentelė. PA „Peržiūrėti pratimų sąrašą” specifikacija	38
20 lentelė. PA „Kurti pratimą” specifikacija	39
21 lentelė. PA „Redaguoti pratimą” specifikacija	40
22 lentelė. PA „Trinti pratimą” specifikacija	41
23 lentelė. PA „Peržiūrėti PopoPlan receptų sąrašą” specifikacija	42
24 lentelė. PA „Peržiūrėti Spooncular receptų sąrašą” specifikacija	43
25 lentelė. PA „Peržiūrėti receptų sąrašą” specifikacija	44
26 lentelė. PA „Kurti receptą” specifikacija	44
27 lentelė. PA „Redaguoti receptą” specifikacija	46
28 lentelė. PA „Trinti receptą” specifikacija	47

Paveikslų sąrašas

1 pav.	Sistemos panaudojimo atvejų diagrama	17
2 pav.	Panaudojimo atvejo „Skaiciuoti miegą” veiklos diagrama	18
3 pav.	Panaudojimo atvejo „Registruotis” veiklos diagrama	19
4 pav.	Panaudojimo atvejo „Prisijungti” veiklos diagrama	20
5 pav.	Panaudojimo atvejo „Atsijungti” veiklos diagrama	21
6 pav.	Panaudojimo atvejo „Redaguoti profili” veiklos diagrama	22
7 pav.	Panaudojimo atvejo „Trinti profili” veiklos diagrama	23
8 pav.	Panaudojimo atvejo „Perziūrėti veiklos sąrašą” veiklos diagrama	25
9 pav.	Panaudojimo atvejo „Kurti veiklą” veiklos diagrama	26
10 pav.	Panaudojimo atvejo „Redaguoti veiklą” veiklos diagrama	27
11 pav.	Panaudojimo atvejo „Trinti veiklą” veiklos diagrama	28
12 pav.	Panaudojimo atvejo „Siusti pranešimą” veiklos diagrama	29
13 pav.	Panaudojimo atvejo „Perziūrėti treniruočių sąrašą” veiklos diagrama	31
14 pav.	Panaudojimo atvejo „Kurti treniruotę” veiklos diagrama	32
15 pav.	Panaudojimo atvejo „Redaguoti treniruotę” veiklos diagrama	33
16 pav.	Panaudojimo atvejo „Trinti treniruotę” veiklos diagrama	34
17 pav.	Panaudojimo atvejo „Perziūrėti PopoPlan pratimų sąrašą” veiklos diagrama	35
18 pav.	Panaudojimo atvejo „Perziūrėti wGer pratimų sąrašą” veiklos diagrama	36
19 pav.	Panaudojimo atvejo „Perziūrėti pratimų sąrašą” veiklos diagrama	37
20 pav.	Panaudojimo atvejo „Kurti pratimą” veiklos diagrama	38
21 pav.	Panaudojimo atvejo „Redaguoti pratimą” veiklos diagrama	39
22 pav.	Panaudojimo atvejo „Trinti pratimą” veiklos diagrama	40
23 pav.	Panaudojimo atvejo „Perziūrėti PopoPlan receptų sąrašą” veiklos diagrama	41
24 pav.	Panaudojimo atvejo „Perziūrėti Spooncular receptų sąrašą” veiklos diagrama	42
25 pav.	Panaudojimo atvejo „Perziūrėti receptų sąrašą” veiklos diagrama	43
26 pav.	Panaudojimo atvejo „Kurti receptą” veiklos diagrama	45
27 pav.	Panaudojimo atvejo „Redaguoti receptą” veiklos diagrama	46
28 pav.	Panaudojimo atvejo „Trinti receptą” veiklos diagrama	47
29 pav.	Iteracinis programų kūrimo proceso modelis	48
30 pav.	Sistemos PopoPlan diegimo diagrama	50
31 pav.	Sistemos PopoPlan paketu diagrama	51
32 pav.	Sistemos PopoPlan klasių diagrama	51
33 pav.	Panaudojimo atvejo „Registruotis” sekų diagrama	52
34 pav.	Panaudojimo atvejo „Prisijungti” sekų diagrama	53
35 pav.	Panaudojimo atvejo „Atsijungti” sekų diagrama	54
36 pav.	Panaudojimo atvejo „Redaguoti profili” sekų diagrama	55
37 pav.	Panaudojimo atvejo „Trinti profili” sekų diagrama	56
38 pav.	Panaudojimo atvejo „Perziūrėti veiklos sąrašą” sekų diagrama	57
39 pav.	Panaudojimo atvejo „Kurti veiklą” sekų diagrama	58
40 pav.	Panaudojimo atvejo „Redaguoti veiklą” sekų diagrama	59
41 pav.	Panaudojimo atvejo „Trinti veiklą” sekų diagrama	60

42 pav. Panaudojimo atvejo „Peržiūrėti treniruočių sąrašą” sekų diagrama	61
43 pav. Panaudojimo atvejo „Kurti treniruotę” sekų diagrama	62
44 pav. Panaudojimo atvejo „Redaguoti treniruotę” sekų diagrama.....	63
45 pav. Panaudojimo atvejo „Trinti treniruotę” sekų diagrama	64
46 pav. Panaudojimo atvejo „Peržiūrėti PopoPlan receptų sąrašą” sekų diagrama	65
47 pav. Panaudojimo atvejo „Peržiūrėti Spooncular receptų sąrašą” sekų diarama	66
48 pav. Panaudojimo atvejo „Peržiūrėti receptų sąrašą” sekų diagrama	67
49 pav. Panaudojimo atvejo „Kurti receptą” sekų diagrama	68
50 pav. Panaudojimo atvejo „Redaguoti receptą” sekų diagrama	69
51 pav. Panaudojimo atvejo „Trinti receptą” sekų diagrama	70
52 pav. Naudotojo sasajos automatinių testų rezultatai	72
53 pav. Veiklos langas	72
54 pav. Veiklos redagavimo langas	73
55 pav. Namų langas	74
56 pav. Focus langas	74
57 pav. PopoPlan pratimų langas	74
58 pav. PopoPlan receptų langas	75
59 pav. wGer pratimų langas	75
60 pav. Spooncular receptų langas	75
61 pav. Prisijungimo langas	75
62 pav. Prisijungusio naudotojo navigacija	75
63 pav. Miego langas	76
64 pav. Veiklos langas	76
65 pav. Veiklos redagavimo langas	76
66 pav. Veiklos langas	76
67 pav. Prisijungimo langas	77
68 pav. Registracijos langas	77
69 pav. Namų langas	78
70 pav. Šoninio meniu navigacija	78
71 pav. Veiklų langas	79
72 pav. Veiklos langas	79
73 pav. Veiklos pridėjimo langas	80
74 pav. Veiklos redagavimo langas	80
75 pav. Spooncular receptai	81
76 pav. Pomodoro laikmadis	81

Santrumpų ir terminų žodynas

Santrumpos:

PA - panaudojimo atvejis.

Terminai:

ORM (angl. *Object relational mapper*) - duomenų bazės įrašų manipuliavimo būdas naudojant objektyvaus programavimo principus.

API (angl. *Application Programming interface*) - tarpininkaujanti sistema, kuri suteikia galimybę naudotojui panaudoti jos funkcijas kitose sistemose.

NPM (angl. *Node package manager*) - Javascript paketų tvarkyklė.

Įvadas

Planavimo programos yra gančtinai populiarios ir dažnos. Jų paskirtis yra gan paprasta, todėl atitinkamai funkcionalumas minimalus. Pagrindinis bruožas yra bandymas išsiskirti su intuityvia naudotojo sąsaja. Dėl minimalistinio požiūrio sistemų, jos atrodė trūkstančios funkcijų. Todėl buvo sugalvoti sveikatingumo planavimo sistema.

Darbo tikslas - sukurti sistemą leidžiančią saugoti savo planus keliose platformose. Siekiant įvykdyti tikslą, buvo sukurti etapai:

1. Dabartinės rinkos konkurentų analizė.
2. Suprojektuoti sistemos ryšius.
3. Sudaryti sveikatingumo planuoklio projektą.
4. Ištesti sukurta sveikatingumo planuoklį..
5. Parengti projekto dokumentaciją.

I darbą įeina 4 dalys - analizavimo, projektavimo, testavimo ir dokumentavimo. Analizavimo dalyje palyginami konkurentai, aptartiamas tinkslingumas, įsigilinama į šiuolaikines technologijas. Projektavimo skyriuje išvardinami funkciniai ir nefunkciniai reikalavimai, paminimos naudojamos diagramų braižymo ir programavimo technologijos, statiniai ir dinaminiai sistemos vaizdai. Testavimo dalis pristato testavimo planą, kriterijus ir rezultatus. Dokumentacijos dalyje pateikiami naudotojo, diegimo ir administravimo vadovai. Galiausiai, atliekamas viso darbo apibendrinimas ir pateikiami rezultatai su išvadomis.

Darbui atlikti buvo įdėta 300 valandų ir parašyta apie 8000 kodo eilučių, kurias įeina serveris, internetinė ir mobilioji programos.

1. Analizė

1.1. Techninis pasiūlymas

1.1.1. Sistemos apibrėžimas

PopoPlan - tai sveikos gyvensenos planavimo sistema, kuri leidžia naudotojui planuoti savo veiklas ir stebėti apie save surinktą informaciją. Kuriamas laiko planavimo sprendimas padėtų asmeniui analizuoti dabartinę sveikatos būseną ir optimizuoti darbo krūvius.

Sistemą sudaro šios dalys:

- Klientinė dalis - grafinė naudotojo sąsaja.
- Serverio dalis - programų sąsaja (angl. *API*), atsakinga už duomenų valdymą.
- Mobilė programa - grafinė naudotojo sąsaja, pritaikyta išnaudoti mobiliojo prietaiso funkcijas

1.1.2. Bendras veiklos tikslas

Bendras veiklos tikslas - palengvinti duomenų prieinamumą naudotojo sveikatos analizei veiklos metu. Duomenys, tokie kaip pulso matavimo rezultatai, gali padėti aptikti problemas, pavyzdžiui, per didelį stresą darbo metu, pervargimą. Tokiu būdu naudotojas gali prieiti prie išvadų siekdamas geresnės fizinės savijautos.

Sėkmingai įdiegta sistema gali sudominti ir motyvuoti žmones atliglioti savo veiklos grafiko optimizavimą.

1.1.3. Sistemos pagrįstumas

Rinkoje yra daug planuoklių, kurie leidžia naudotojams susikurti veiklas, taip sudarant savo tvarkaraštį. Kadangi daugelis tokų programų susikoncentravusios į minimalistinį funkcionalumą ir fiksuoja tik veiklas, naudotojui būtų naudinga turėti sistemą, kurioje gali laikyti visą informaciją bei gauti įdėjų tokioms veikloms, kaip mityba ir sportas.

Taip pat vienas iš tikslų sukurti „popoPlan” sveikatingumo planavimo sistemą buvo noras išnaudoti fitbito suteiktas technologijas. Dabartiniai fitbito prietaisai turi daug funkcijų ir gali būti laikomi medicinoje.

1.1.4. Konkurencija rinkoje

1 lentelė. Konkurentų apžvalga

Lyginimo kriterijai	Popoplan	To Do List	Dreamie Planner
Naudotojų skaičius	Nėra	10 000 000 naudotojų	1 000 000 naudotojų
Operacinė sistema	Android ir iOS	iOS	Android
Kaina	Nemokama	Nemokama	Nemokama
Galima prieiti prie duomenų per naršyklę	+	-	-
Turi API	+	-	-
Receptų valdymas	+	-	-
Sporto pratimų valdymas	+	-	-
Treniruočių valdymas	+	-	-
Realizuotas Pomodoro laikmatis	+	-	-

- Naudotojų skaičius parodo, kokio populiarumo programa. Kadangi „To Do list” turi 10 kartų daugiau naudotojų, tai gali parodyti, kad tiesiog žmonėms užtenka paprasto veiklos sąrašo.
- Operacinė sistema - palaikoma platforma. Didesnis platformų skaičiau palaikymas, tuo daugiau žmonių gali naudotis kuriama sistema.
- Kaina - šiais laikais žmonėms svarbu programos, už kurias nereiktų mokėti, taip pritraukiamą daugiau naudotojų.
- Galima prieiti prie duomenų per naršyklę - daugiau palaikomų platformų suteikia klientui lankstumo.
- Turi API - duomenys gaunami atliekant užklausą į API, todėl programa tampa priklausoma nuo interneto ryšio.
- Receptų valdymas - suteikia klientams galimybę pasidomėti dėl valgių, ar tai pietų, vakarienės metu.
- Sporto pratimų valdymas - suteikia klientams su planuokliu galimybę pažiūrėti sporto pratimų ir turėti prie veiklos.
- Treniruočių valdymas - suteikia klientams galimybę valdyti treniruočių informaciją ir išsidėti į veiklą.
- Realizuotas Pomodoro laikmatis - tinkta prie planuoklio leidžiant naudotojui veiklos metu bandyti atlikti darbą efektyviau.

1.1.5. Prototipai ir pagalbinė informacija

Kuriant internetinę programą buvo naudojamos „wireframe” technologijos, kurios pritaikytos kurti preliminarų tinklapio įvaizdį su navigavimu.

Dirbant prie mobiliosios programos buvo naudojami esami NativeScript komponentai. Kiekvienos sistemos kūrimo iteracijos metu komponentų vietas buvo koreguojamos.

1.1.6. Ištekliai, reikalingi sistemai sukurti

„popoPlan” sistema turi 4 dalis: kliento dalis, serveris, mobiliojo telefono dalis ir laikrodis. Siekiant saugoti kodą ir iškelti į debesį, reikia keturių atskirų kodo talpyklų tokiuose įrankiuose, kaip GitHub, kurių paskirtis leisti programuotojui patogiai talpinti programas, kurias naudotojai galėtų pasiekti. Pačios sistemos kūrimui užtenka vieno programuotojo. Sukurti sistemai reikėjo apie 300 darbo valandų.

1.2. Galimybių analizė

1.2.1. Techninės galimybės

Internetinės programos grafinė sąsaja buvo realizuota naudojant Vue.JS karkasą su JavaScript programavimo kalba. Šis karkasas buvo naudojamas dėl to, kad yra greitesnis nei alternatyvos, pavyzdžiui, Angular ir React. Vue karkasas taip pat

Serverio programa buvo aprašoma naudojant Express.JS karkasą su TypeScript programavimo kalba. Express.JS yra patogi technologija siekiant sukurti API. Juo lengvai galima aprašyti API puslapių kelius, HTTP metodus su pranešimais, aiškiai dokumentuotas ir lengva išmokti. TS programavimo kalba buvo naudojama, nes serverio programeje apdorojami duomenys, todėl statiniai tipai gali praversti rašant aiškesnį kodą. Taip pat naudojamas Mongoose ORM (angl. *Object-relational mapper*) karkasas, kuris leidžia lengvai komuniuoti su MongoDB duomenų baze vykdant užklausas per objektus. MongoDB duomenų bazė pasirinkta, nes yra greita ir efektyvi mažoms programoms.

Mobiliai programai buvo naudojamas NativeScript su TypeScript programavimo kalba. NativeScript karkasas suteikia galimybę kurti programas Android ir iOS operacinėms sistemoms. NS leidžia suraupyti daugiau laiko nekuriant atskirų programų Android OS su Java/Kotlin kalba, o ant iOS - Swift. Be to, tai yra puikus pasirinkimas internetinių programų kūrėjams, nes karkasas veikia su Vue.js, React, Angular technologijomis. Kadangi NativeScript yra open-source community įrankis, buvo susidurta su dokumentacijos trūkumu, neaiškiai pavyzdžiais ir lėtai užkraunamu projektu. TypeScript kalba su Vue karkasu buvo užkrauta sukūrus projektą.

1.2.2. Naudotojų pasiruošimo analizė

Siekiant pasinaudoti surinkta PopoPlan sistema, naudotojui nereikės turėti papildomų žinių apie sistemą. Pries sistemos bus galima prisijungti per naršyklę ir per mobilujį telefoną.

2. Projektas

2.1. Reikalavimų specifikacija

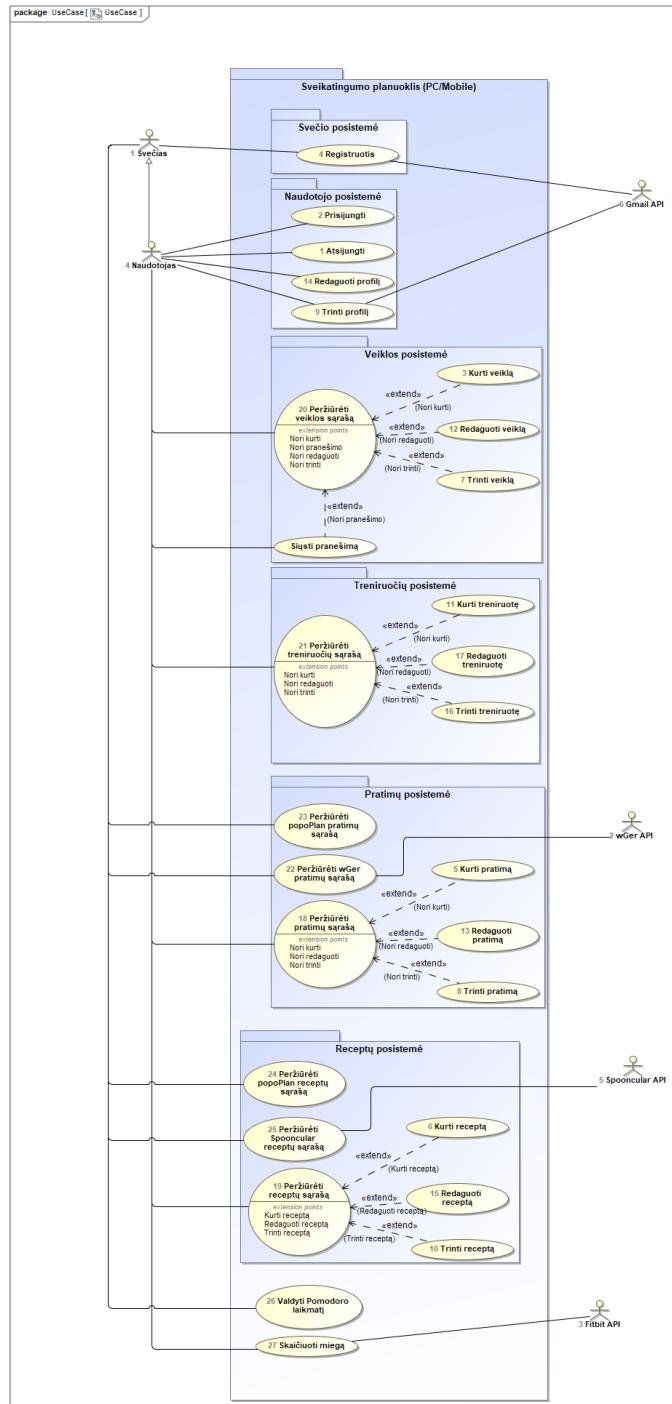
2.1.1. Komercinė specifikacija

Sukurta sistema yra baigiamasis bakalauro darbas. Projekto įdėjų sugalvojo ir realizavo universiteto studentas - Tautvydas Dikšas. Šio darbo galimi naudotojai - žmonės, norintys turėti planuoklį, turintį įvairių funkcijų, kurios suteikia įkvėpimo veikloms bei leidžia sekti duomenis apie save.

Projekto realizavimui skirtas laikas nuo 2021 m. vasario 1 d. iki 2021 m. gegužės 5 d. Kadangi baigiamasis darbas yra asmeninis, skirto biudžeto nėra.

2.1.2. Sistemos funkcijos

Visi funkciniai reikalavimai buvo atvaizduoti remiantis UML standartais sudarant panaudojimo atvejų diaramą. (1 pav.)



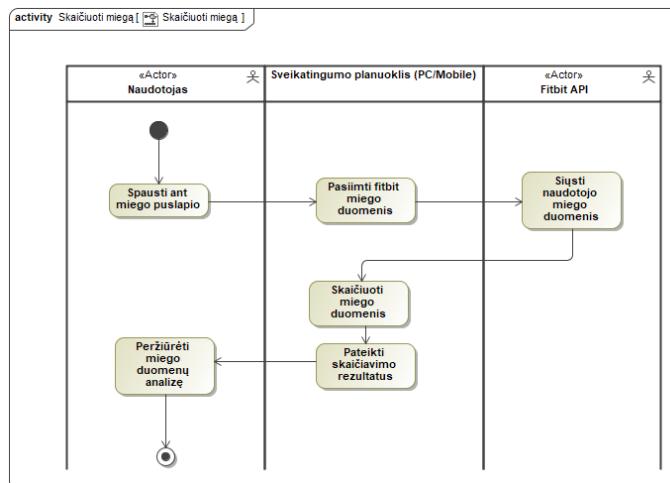
1 pav. Sistemos panaudojimo atvejų diagrama

PopoPlan sistemos funkcijos atliktos keliose platformose: internetinėje, mobilioje. Kadangi mobiliosios ir internetinės programų funkcijos atskirtoja, jos kategorizuojamos vienoje posistemėje. Taip pat PC/Mobile savyje turi specifines posistemes, kurios skirtos konkrečių duomenų manipuliavimui. PC/Mobile posistemėje yra visos pagrindinės funkcijos, kurias naudotojas gali valdyti. Šioje sistemoje yra išskiriami aktoriai:

- Naudotojas - žmogus, kuris nori pasinaudoti sistema.
- Svečias - žmogus, kuris atėjo į sistemą, bet nėra užsiregistravęs ir prisijungęs.
- GMail API - sistema, į kurią kreipiamasi siekiant išsiųsti naudotojams laiškus.
- Fitbit API - sistema, naudojama komunikujant su laikrodžiu ir gaunant duomenis apie naudotoją.
- Spooncular API - sistema, iš kurios gaunami maisto receptai.
- wGer - open source sistema, iš kurios gaunami sportui pratimai.

Kiekviena funkcija panaudojimo atvejų diagramoje aprašoma UML veiklų diagrama. Veiklų diagramose galima matyti, kaip sistema ir aktorių reaguoja į vienas kito veiksmus. PC/Mobile sveikatingumo planuoklio dalyje galima matyti svečio, naudotojo, veiklos, treniruočių, pratimų ir receptų posistemes. Pagal funkcionalumą, dabartinis sudarytas projektas yra informacinė sistema, todėl unikaliu funkcionalumu nepasižymi.

Panaudojimo atvejo „Skaičiuoti miegą” veiklos diagramoje (2 pav.) parodyta, kad naudotojas atsidarės Miego puslapį, spausdus miego puslapyje ant mygtuko, jo miego duomenys bus paimti iš fitbit sistemos ir bus suskaičiuojamas vidurkis, dažniausiai pasikartojantis rezultatas, galiausiai tai atvaizduojamas ekrane. Veiklos diagramos specifikacija aprašoma 2 lent.

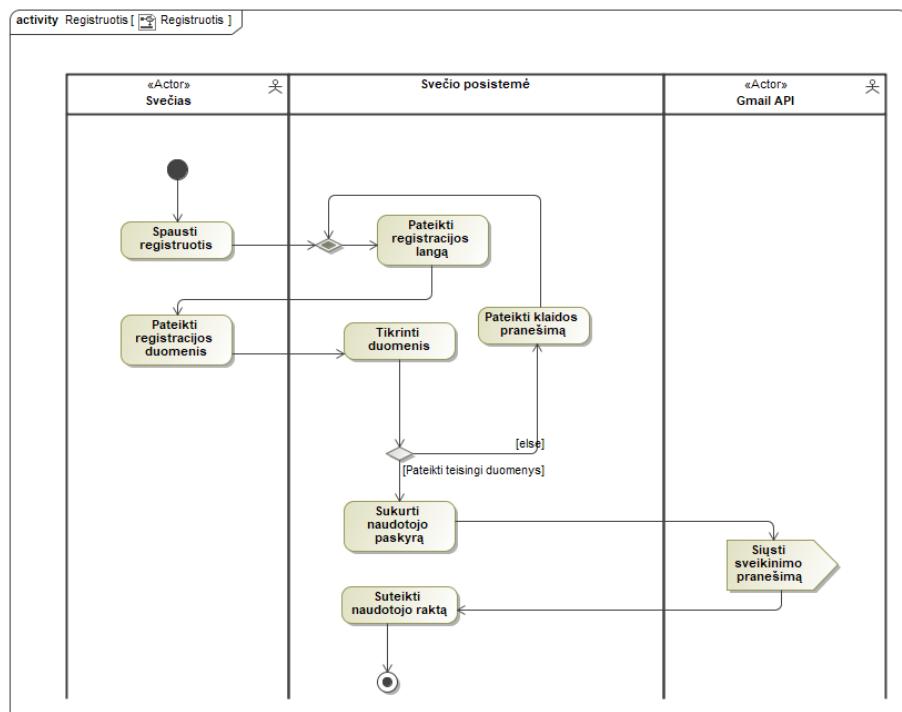


2 pav. Panaudojimo atvejo „Skaičiuoti miegą” veiklos diagrama

2 lentelė. PA „Skaičiuoti miegą” specifikacija

PA. Skaičiuoti miegą	
Tikslas. Suteikti galimybę naudotojui matyti suskaičiuotus miego duomenis	
Apašymas. PA suteikia galimybę naudotojams sužinoti apie miegą analizę.	
Prieš sąlyga	Naudotojas turi būti prisijungęs.
Aktorai	Fitbit API, Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas mato suskaičiuotą miego duomenų analizę.

Panaudojimo atvejo „Registruotis” veiklos diagramoje (3 pav.) parodyta, kad registracija prasideda paspaudus ant registracijos mygtuko, atidaromas registracijos langas, tada svečias išrašo savo duomenis į laukus ir juos pateikia. Atliekamas duomenų patikrinimas, jei duomenys blogi, pateikiamas klaidos pranešimas, jei geri, tada kreipiamasi į GMail API, kuris išsiunčia į svečio paštą - laišką. Galiausiai svečias prijungiamas prie sistemos, nes buvo suteiktas raktas. Papildomos informacijos 3 lent.

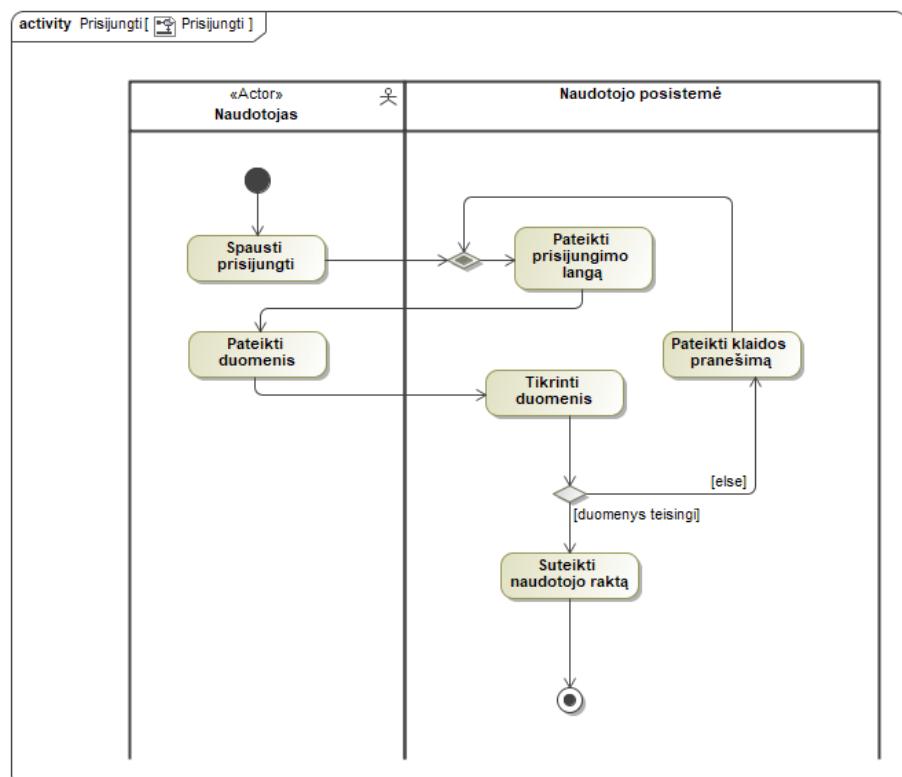


3 pav. Panaudojimo atvejo „Registruotis” veiklos diagrama

3 lentelė. PA „Registruotis“ specifikacija

PA. Registruotis	
Tikslas. Suteikti galimybę svečiui užsiregistrnuoti sistemoje.	
Aprašymas. PA suteikia galimybę svečiui užsiregistrnuoti sistemoje.	
Prieš sąlyga	Svečias turi būti įsijungęs sistemą
Aktorai	Gmail API, Svečias
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Sukurta svečio paskyra ir prijungtas prie sistemos.

Panaudojimo atvejo „Prisijungti“ veiklos diagramoje (4 pav.) parodyta, kad veikla prasideda paspaudus prisijungti mygtuką, tada pateikiamas prisijungimo langas. Tame lange surašomi naudotojo paskyros duomenys. Po to atliekami paskyros duomenų tikrinimai, jei blogi duomenys, pateikiamas klaidos pranešimas, jei geri - suteikiamas raktas ir prijungiamas naudotojas prie sistemos. Veiklos diagramos specifikacija 4 lent.

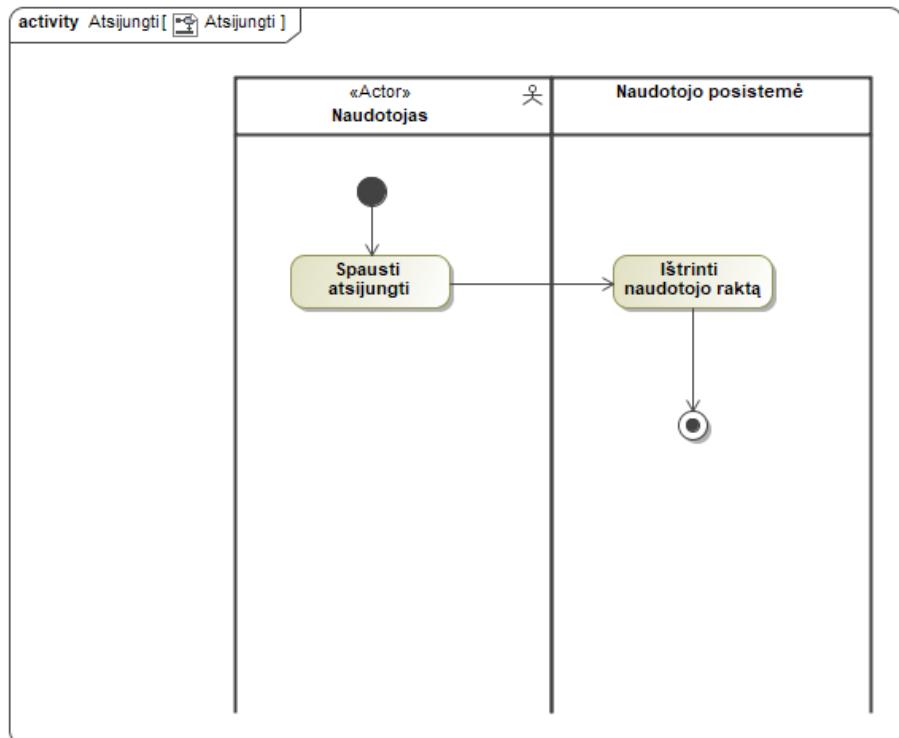


4 pav. Panaudojimo atvejo „Prisijungti“ veiklos diagrama

4 lentelė. PA „Prisijungti” specifikacija

PA. Prisijungti							
Tikslas. Suteikti galimybę svečiui prisijungti prie sistemos.							
Aprašymas. PA suteikia galimybę svečiui prisijungti prie sistemos.							
Prieš sąlyga	Svečias turi būti įsijungęs sistemą.						
Aktorai	Svečias						
Susiję panaudojimo atvejai	<table border="1"> <tr> <td>Apimantys PA</td><td>-</td></tr> <tr> <td>Išplečiantys PA</td><td>-</td></tr> <tr> <td>Specializuojantys PA</td><td>-</td></tr> </table>	Apimantys PA	-	Išplečiantys PA	-	Specializuojantys PA	-
Apimantys PA	-						
Išplečiantys PA	-						
Specializuojantys PA	-						
Po sąlyga	Svečias prijungtas prie sistemos.						

Panaudojimo atvejo „Atsijungti” veiklos diagramoje (5 pav.) parodyta, kad atsijungimas prasideda paspaudus atsijungti mygtuką ir veikla baigiasi, kai Naudotojo prisijungimo raktas ištrinamas ir jis atjungiamas nuo sistemos. Veiklos diagramos specifikacija 5 lent.

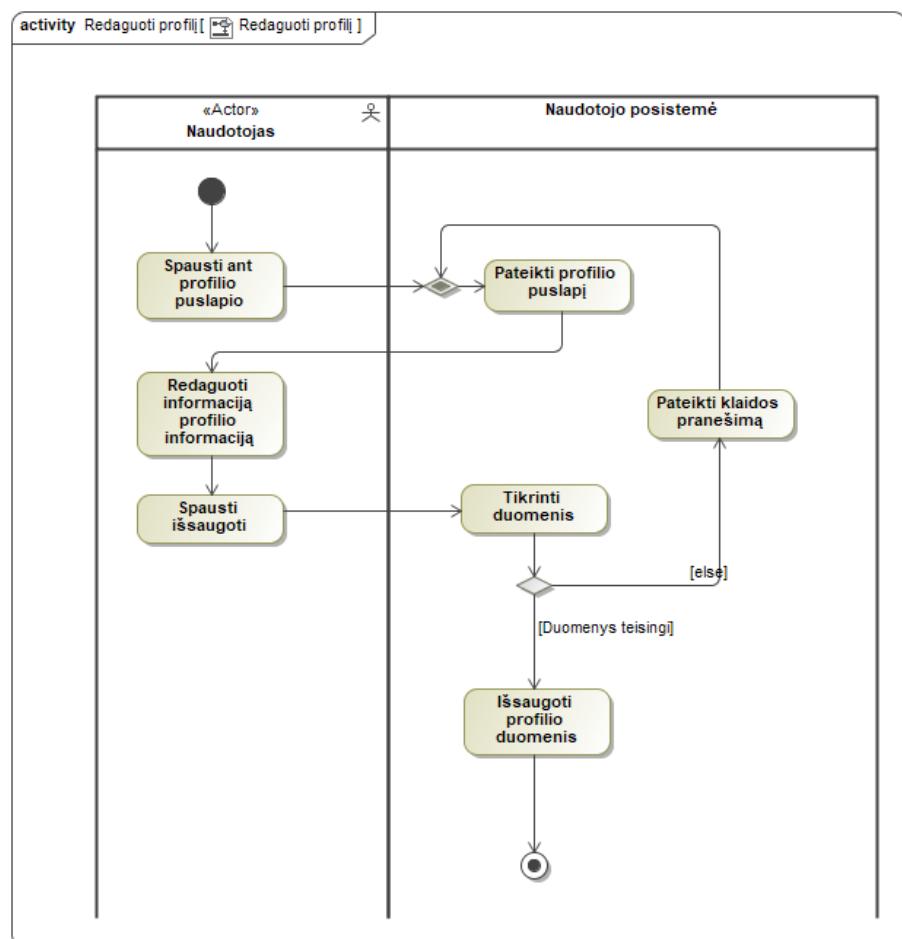


5 pav. Panaudojimo atvejo „Atsijungti” veiklos diagrama

5 lentelė. PA „Atsijungti” specifikacija

PA. Atsijungti							
Tikslas. Suteikti galimybę naudotojui atsijungti nuo sistemos.							
Aprašymas. PA suteikia galimybę naudotojui atsijungti nuo sistemos.							
Prieš sąlyga	Naudotojas turi būti prisijungęs prie sistemos.						
Aktorai	Naudotojas						
Susiję panaudojimo atvejai	<table border="1"> <tr> <td>Apimantys PA</td><td>-</td></tr> <tr> <td>Išplečiantys PA</td><td>-</td></tr> <tr> <td>Specializuojantys PA</td><td>-</td></tr> </table>	Apimantys PA	-	Išplečiantys PA	-	Specializuojantys PA	-
Apimantys PA	-						
Išplečiantys PA	-						
Specializuojantys PA	-						
Po sąlyga	Naudotojas yra atjungtas nuo sistemos.						

Panaudojimo atvejo „Redaguoti profilių“ veiklos diagramoje (6 pav.) parodyta, kad veiksmas prasideda paspaudus ant profilio nustatyto puslapio. Paskui pateikiamas profilio puslapis su visais naudotojo duomenimis. Naudojamas redaguojama informacija, spaudžia saugoti ir laukai yra tikrinami, jei blogi - metama klaida, jei teisingi - viskas išsaugoma. Veiklos diagramos specifikacija 6 lent.

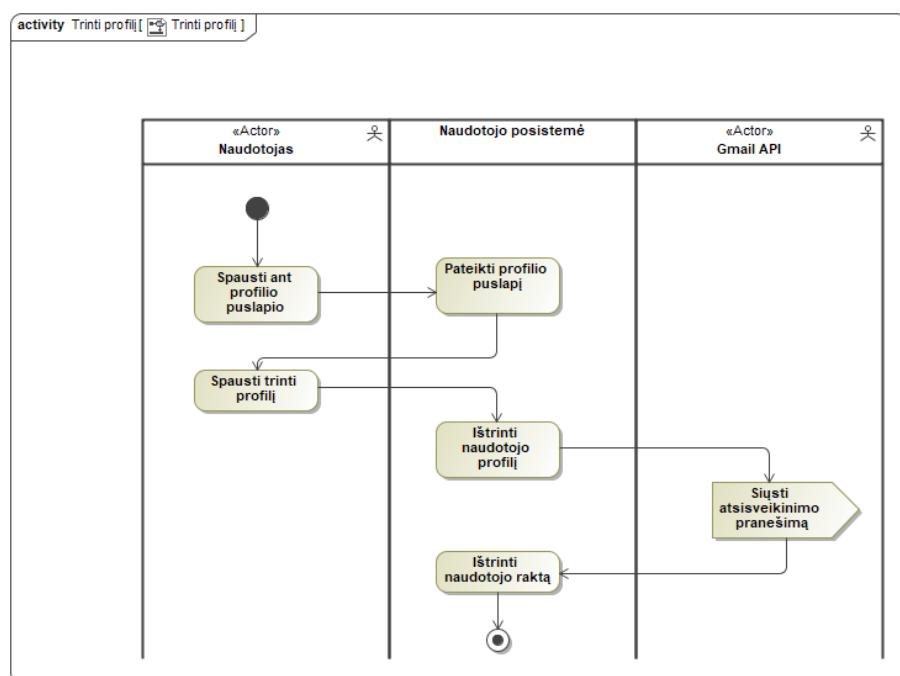


6 pav. Panaudojimo atvejo „Redaguoti profilių“ veiklos diagrama

6 lentelė. PA „Redaguoti profili“ specifikacija

PA. Redaguoti profili	
Tikslas. Suteikti galimybę naudotojui pasikeisti profilio duomenis.	
Aprašymas. PA suteikia galimybę naudotojui pasikeisti profilio duomenis.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Pakeisti naudotojo profilio duomenys.

Panaudojimo atvejo „Trinti profili“ veiklos diagramoje (7 pav.) parodyta, kad profilio trinimas gali prasidėti paspaudus ant profilio puslapio. Tada pateikiamas profilio puslapis. Paspaudžiama trinti profilį ir yra ištrinamas profilis. Po to, išsiunčiamas atsisveikinimo laiškas į naudotojo paštą. Galiausiai iš naudotojo naršyklys ištrinamas raktas ir yra atjungiamas nuo sistemos. Veiklos diagramos specifikacija aprašoma 7 lent.



7 pav. Panaudojimo atvejo „Trinti profili“ veiklos diagrama

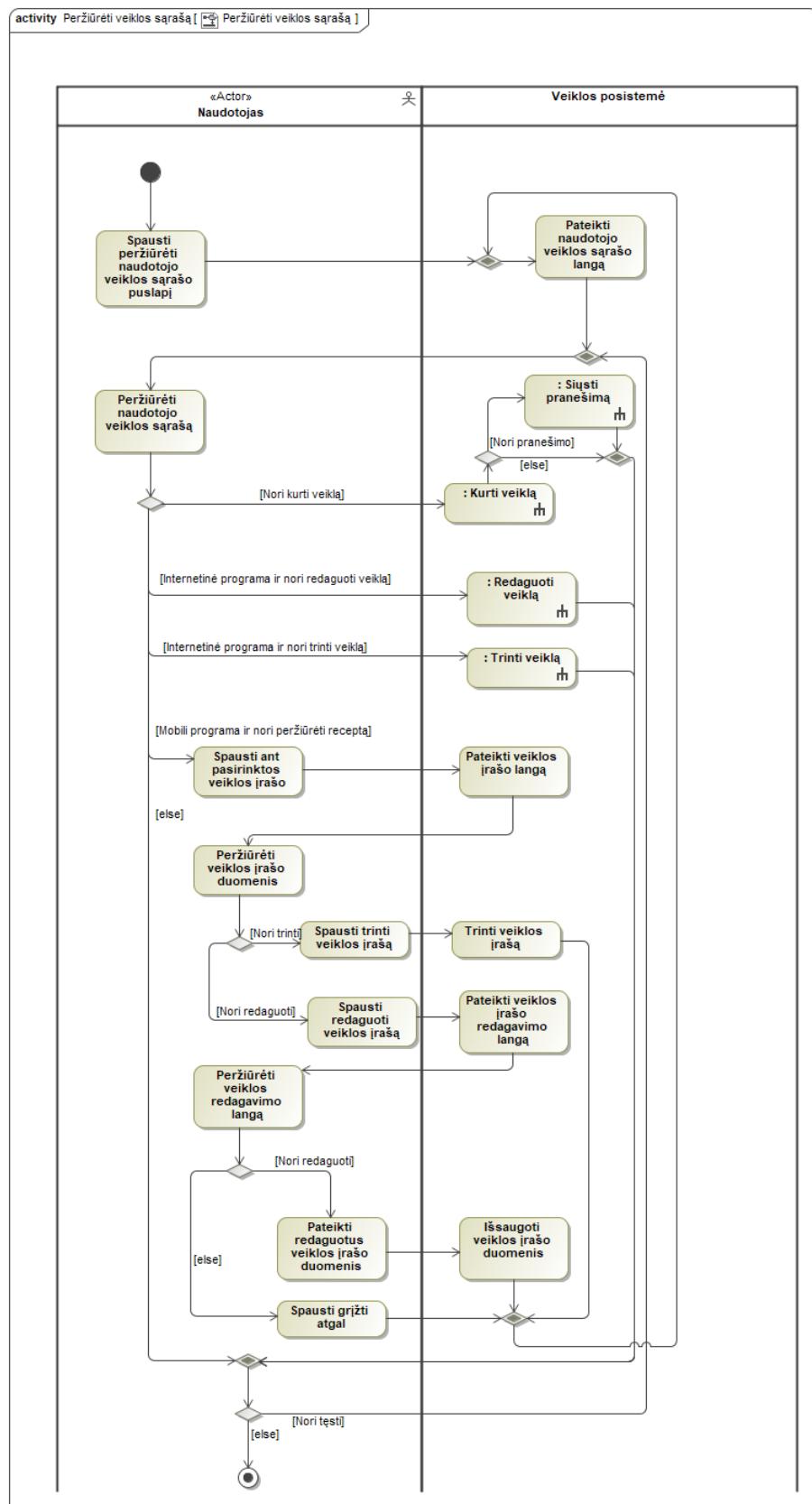
7 lentelė. PA „Trinti profilį” specifikacija

PA. Trinti profilį		
Tikslas. Suteikti galimybę naudotojui ištrinti savo profilį.		
Aprašymas. PA suteikia galimybę naudotojui ištrinti savo profilį.		
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.	
Aktorai	Naudotojas, GMail API	
Susiję panaudojimo atvejai	Apimantys PA	-
	Išplečiantys PA	-
	Specializuojantys PA	-
Po sąlyga	Naudotojas atjungtas nuo profilio ir profilis ištrintas.	

Panaudojimo atvejo „Peržiūrėti veiklos sąrašą” veiklos diagramoje (8 pav.) parodyta, kad veiklos sąrašo peržiūréjimas prasideda spaudžiant ant naudotojo veiklos puslapio. Tada pateikiamas naudotojo veiklos puslapis. Peržiūrėjės veiklų įrašus, naudotojas gali kurti veiklą, jei veikla sukurta ant mobilios programos, nustatytu laiku bus išsiųstas pranešimas. Jei sistema naudojama per naršykłę - galima redaguoti veiklą ir trinti veiklą, jei naudojama mobili programa, tada reikia spausti ant veiklos įrašo norint atlikti trinimą arba įrašo redagavimą. Veiklos diagramos specifikacija aprašoma 8 lent.

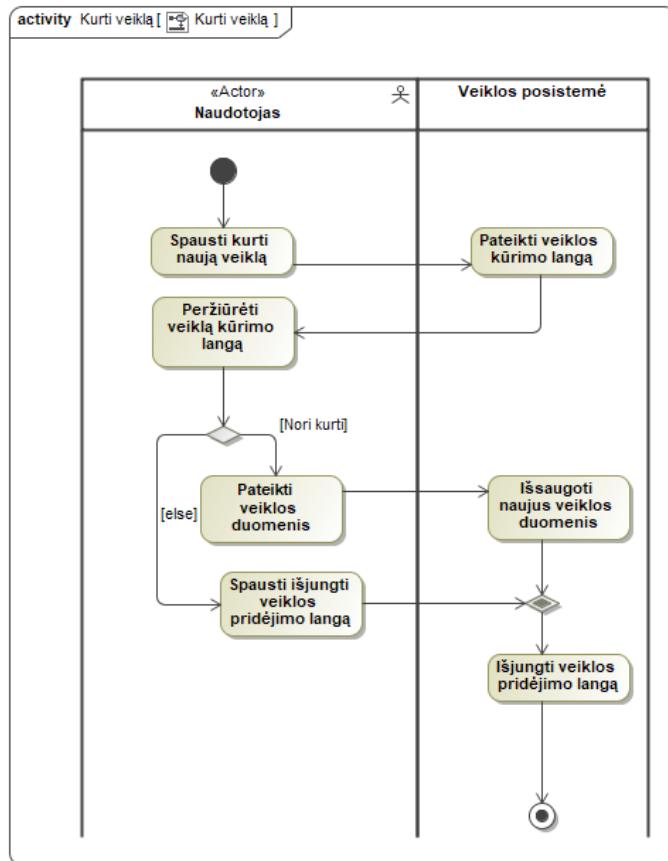
8 lentelė. PA „Peržiūrėti veiklos sąrašą” specifikacija

PA. Peržiūrėti veiklos sąrašą		
Tikslas. Suteikti galimybę naudotojui peržiūrėti savo veiklos sąrašą.		
Aprašymas. PA suteikia galimybę naudotojui peržiūrėti savo veiklos sąrašą.		
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.	
Aktorai	Naudotojas	
Susiję panaudojimo atvejai	Apimantys PA	-
	Išplečiantys PA	3 “Kurti veiklą”, 12 “Redaguoti veiklą”, “Siųsti pranešimą”, 7 “Trinti veiklą”
	Specializuojantys PA	-
Po sąlyga	Naudotojas peržiūri savo veiklos sąrašą.	



8 pav. Panaudojimo atvejo „Peržiūrėti veiklos sąrašą“ veiklos diagrama

Panaudojimo atvejo „Kurti veiklą” veiklos diagramoje (9 pav.) parodyta, kad veiklos kūrimas prasideda paspaudus kurti naują veiklą. Tada atidaromas veiklos kūrimo langas. Paskui naudotojas gali apsispręsti arba iš jungti veiklos pridėjimo langą, arba užpildyti laukus ir pateikti veiklos duomenis, taip išsaugojant veiklos įrašą. Galiausiai veiklos pridėjimo langas išjungiamas. Veiklos diagramos specifikacija aprašoma 9 lent.

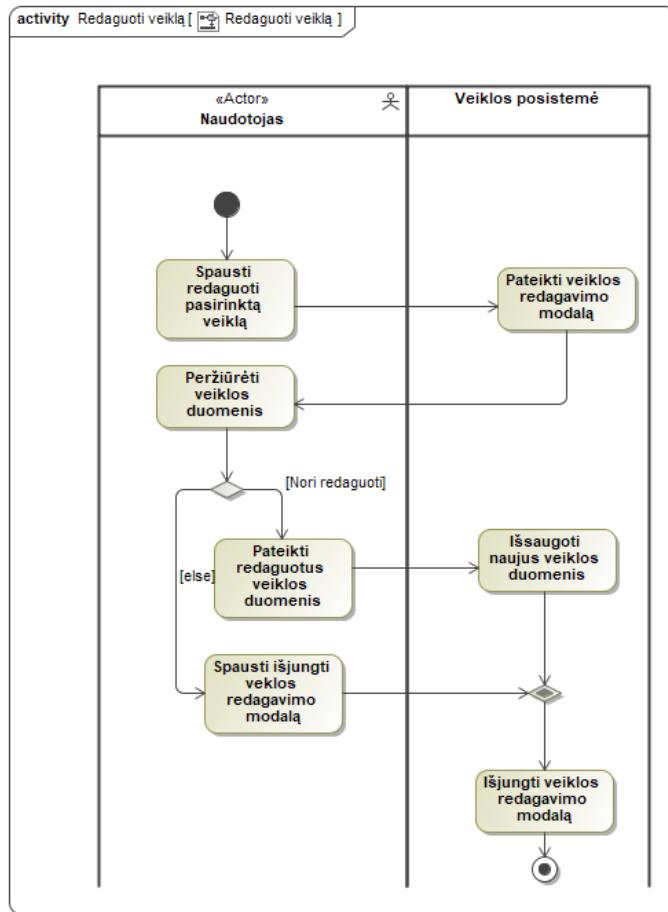


9 pav. Panaudojimo atvejo „Kurti veiklą” veiklos diagrama

9 lentelė. PA „Kurti veiklą” specifikacija

PA. Kurti veiklą	
Tikslas. Suteikti galimybę naudotojui kurti savo veiklos įrašą.	
Aprašymas. PA suteikia galimybę naudotojui kurti savo veiklos įrašą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktoriai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas sukuria savo naują veiklos įrašą.

Panaudojimo atvejo „Redaguoti veiklą“ veiklos diagramoje (10 pav.) parodyta, kad veiklos redagavimas prasideda paspaudus ant įrašo redaguoti, tada pateikiamas veiklos redagavimo langas. Peržiūrėjus duomenis, naudotojas gali pasirinkti, arba iš jungti veiklos redagavimo langą, arba redaguoti duomenis ir išsaugoti juos. Galiausiai veiklos redagavimo langas yra uždaromos. Veiklos diagramos specifikacija aprašomos 10 lent.

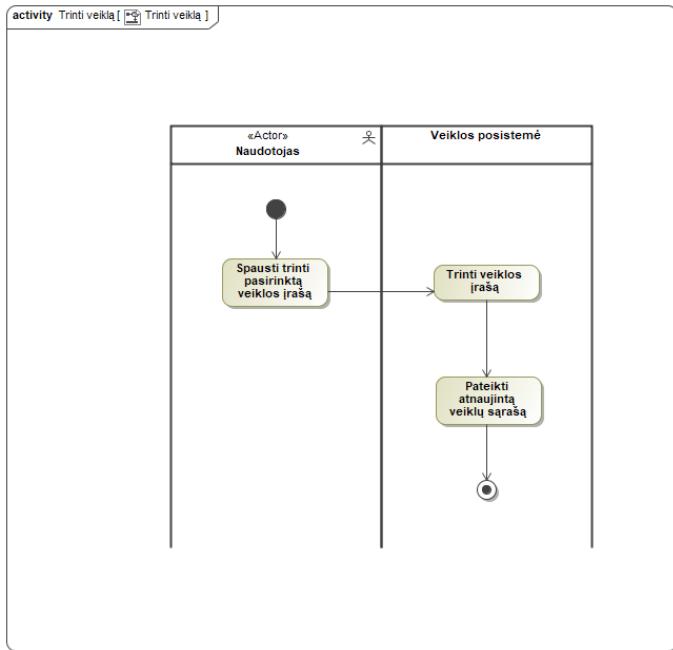


10 pav. Panaudojimo atvejo „Redaguoti veiklą“ veiklos diagrama

10 lentelė. PA „Redaguoti veiklą“ specifikacija

PA. Redaguoti veiklą	
Tikslas. Suteikti galimybę naudotojui redaguoti savo veiklos įrašo informaciją.	
Aprašymas. PA suteikia galimybę naudotojui redaguoti savo veiklos įrašo informaciją.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojo veiklos įrašo informacija redaguota.

Panaudojimo atvejo „Trinti veiklą” veiklos diagramoje (11 pav.) parodyta, kad veiklos trinimas prasideda paspaudus ant trinimo mygtuko, tada ištrinamas pats įrašas ir atnaujinama duomenų lentelė. Veiklos specifikacija aprašoma 11 lent.

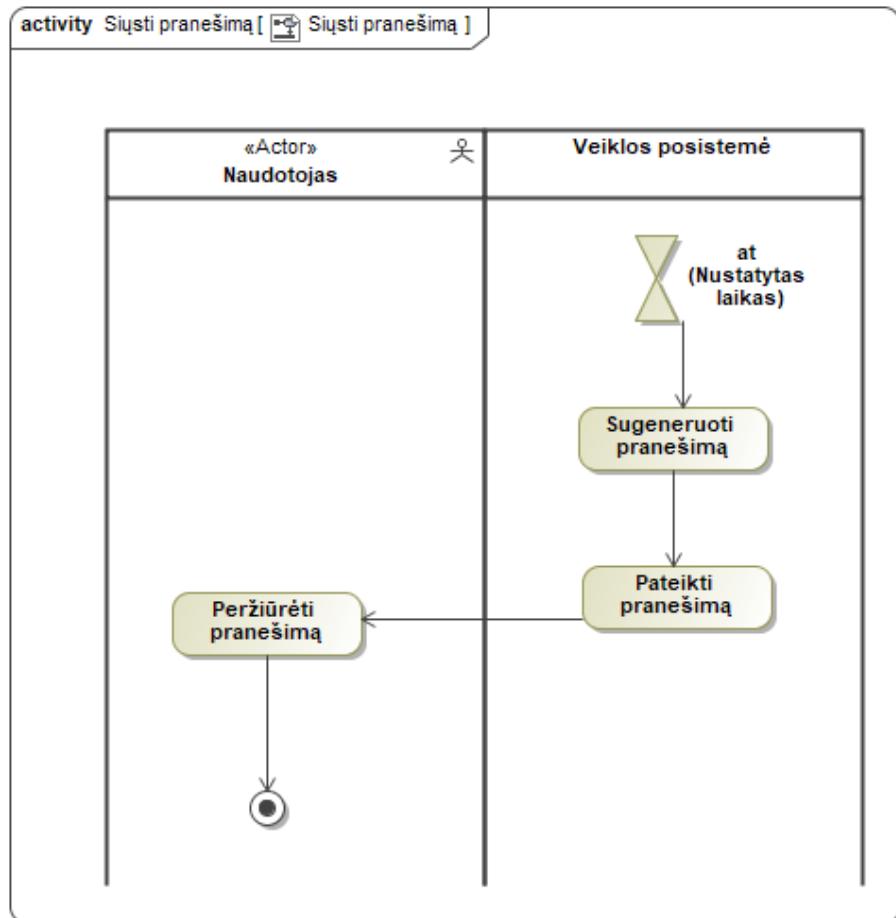


11 pav. Panaudojimo atvejo „Trinti veiklą” veiklos diagrama

11 lentelė. PA „Trinti veiklą” specifikacija

PA. Trinti veiklą	
Tikslas. Suteikti galimybę naudotojui ištrinti savo veiklos sąrašo įrašą.	
Aprašymas. PA suteikia galimybę naudotojui ištrinti savo veiklos sąrašo įrašą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojo veiklos sąrašo įrašas ištrinamas.

Panaudojimo atvejo „Siųsti pranešimą“ veiklos diagramoje (pav.) parodyta, jei naudotojas ant mobilios programos, kad nustatytu laiku bus sugeneruota žinutė ir naudotojui grįš pranešimas. Veiklos diagramos specifikacija aprašoma 12 lent.



12 pav. Panaudojimo atvejo „Siųsti pranešimą“ veiklos diagrama

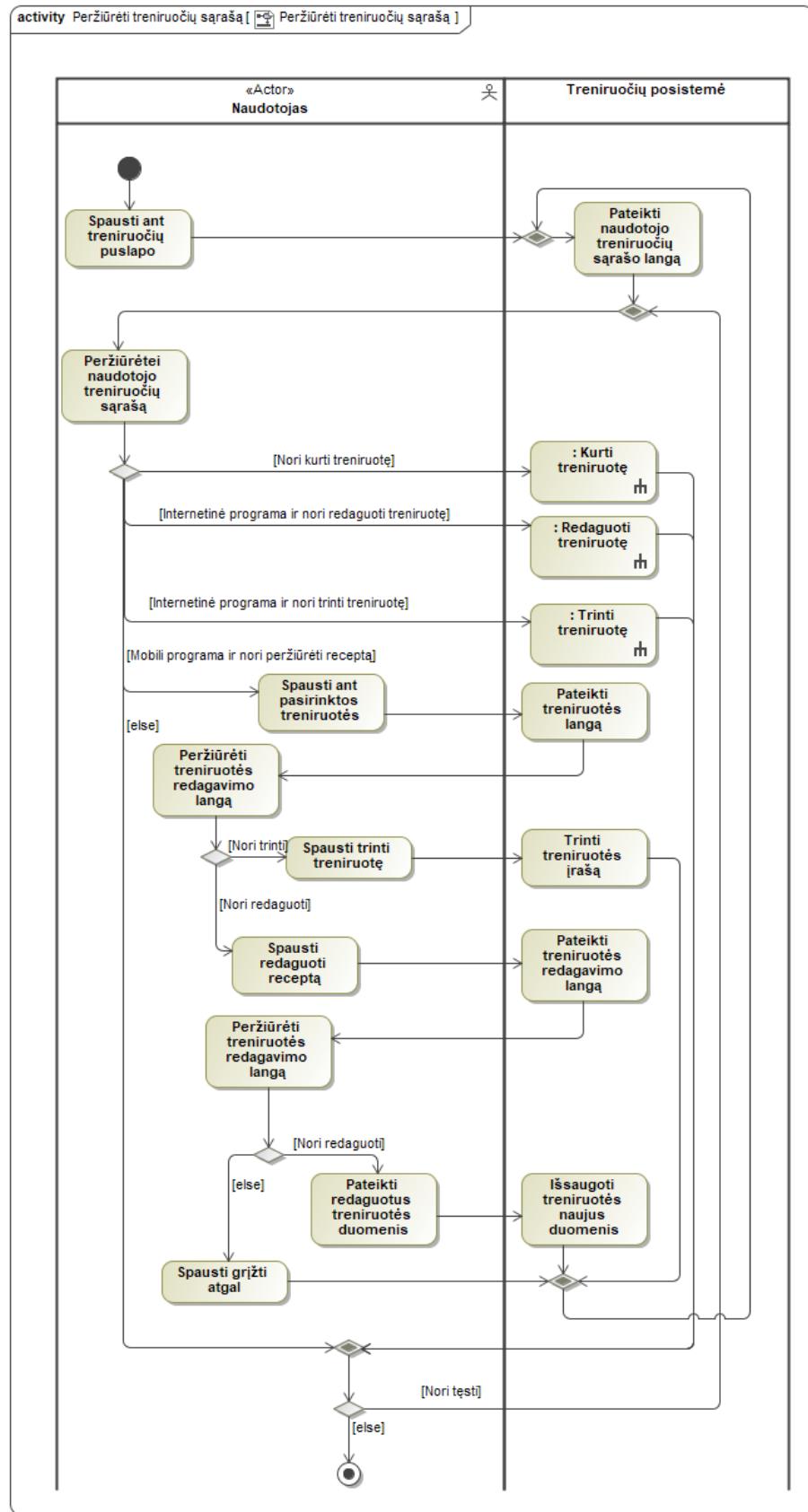
12 lentelė. PA „Siųsti pranešimą“ specifikacija

PA. Siųsti pranešimą	
Tikslas. Suteikti galimybę naudotojui siųsti pranešimą.	
Aprašymas. PA suteikia galimybę naudotojui siųsti pranešimą į telefoną.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos. Tik mobilioje programe
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Išsiųstas naudotojui pranešimas.

Panaudojimo atvejo „Peržiūrėti treniruočių sąrašą” veiklos diagramoje (13 pav.) parodyta, kad treniruočių sąrašo peržiūrėjimas prasideda spaudžiant ant naudotojo treniruočių puslapio. Tada pateikiamas naudotojo treniruočių puslapis. Peržiūrėjės treniruočių įrašus, naudotojas gali kurti treniruotę. Jei sistema naudojama per naršyklę - galima redaguoti treniruotę ir trinti treniruotę, jei naudojama mobili programa, tada reikia spausti ant treniruotės įrašo norint atlikti trinimą arba įrašo redagavimą. Veiklos diagramos specifikacija aprašoma 13 lent.

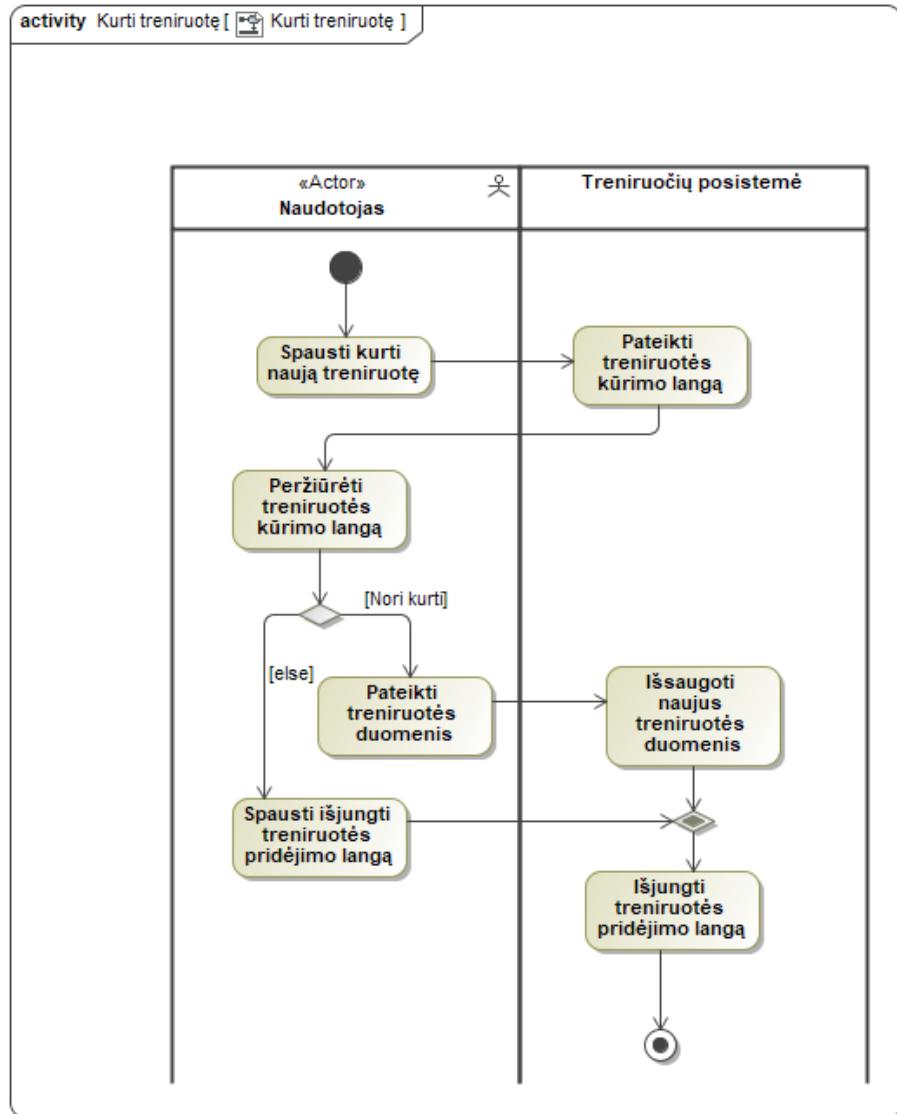
13 lentelė. PA „Peržiūrėti treniruočių sąrašą” specifikacija

PA. Peržiūrėti treniruočių sąrašą	
Tikslas. Suteikti galimybę naudotojui peržiūrėti savo treniruočių sąrašą.	
Aprašymas. PA suteikia galimybę naudotojui peržiūrėti savo treniruočių sąrašą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas mato savo treniruočių sąrašą.



13 pav. Panaudojimo atvejo „Peržiūrėti treniruočių sąrašą“ veiklos diagrama

Panaudojimo atvejo „Kurti treniruotę“ veiklos diagramoje (14 pav.) parodyta, kad treniruotės kūrimas prasideda paspaudus kurti naują treniruotę. Tada atidaromas treniruotės kūrimo langas. Paskui naudotojas gali apsispręsti arba iš jungti treniruotės kūrimo langą, arba užpildyti laukus ir pateikti treniruotės duomenis, taip išsaugojant treniruotės įrašą. Galiausiai treniruotės pridėjimo langas išjungiamas. Veiklos diogramos specifikacija aprašoma 14 lent.

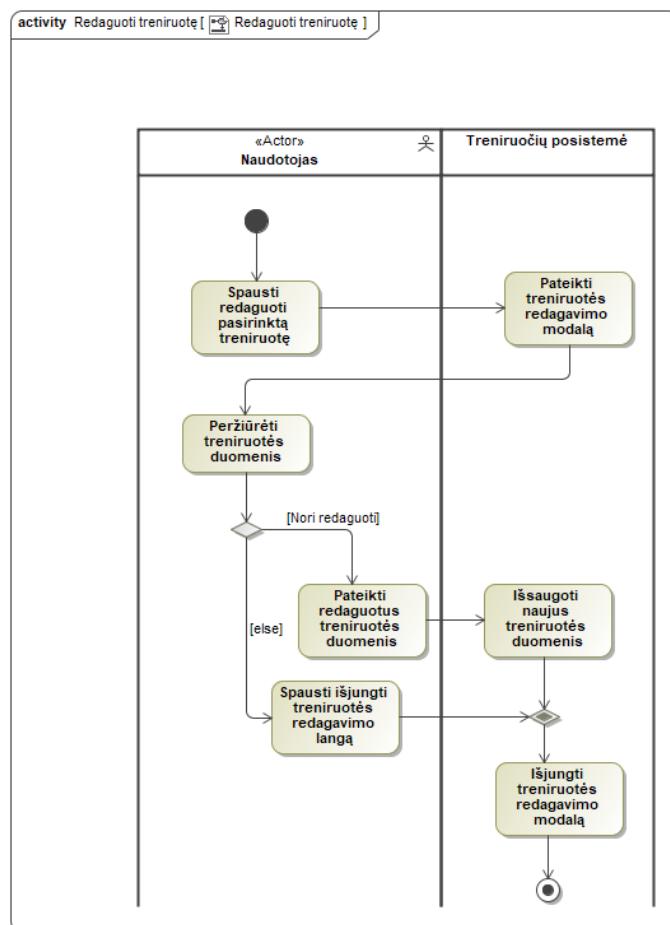


14 pav. Panaudojimo atvejo „Kurti treniruotę“ veiklos diagrama

14 lentelė. PA „Kurti treniruotę“ specifikacija

PA. Kurti treniruotę							
Tikslas. Suteikti galimybę naudotojui sukurti savo naują treniruotę.							
Aprašymas. PA suteikia galimybę naudotojui sukurti savo naują treniruotę.							
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.						
Aktorai	Naudotojas						
Susiję panaudojimo atvejai	<table border="1"> <tr> <td>Apimantys PA</td><td>-</td></tr> <tr> <td>Išplečiantys PA</td><td>-</td></tr> <tr> <td>Specializuojantys PA</td><td>-</td></tr> </table>	Apimantys PA	-	Išplečiantys PA	-	Specializuojantys PA	-
Apimantys PA	-						
Išplečiantys PA	-						
Specializuojantys PA	-						
Po sąlyga	Naudotojas sukuria savo naują treniruotę.						

Panaudojimo atvejo „Redaguoti treniruotę“ veiklos diagramoje (15 pav.) parodyta, kad treniruotės redagavimas prasideda paspaudus ant įrašo redaguoti, tada pateikiamas treniruotės redagavimo langas. Peržiūrėjus duomenis, naudotojas gali pasirinkti, arba išjungti treniruotės redagavimo langą, arba redaguoti duomenis ir išsaugoti juos. Galiausiai treniruotės redagavimo langas yra uždaromas. Veiklos diogramos specifikacija aprašomas 15 lent.

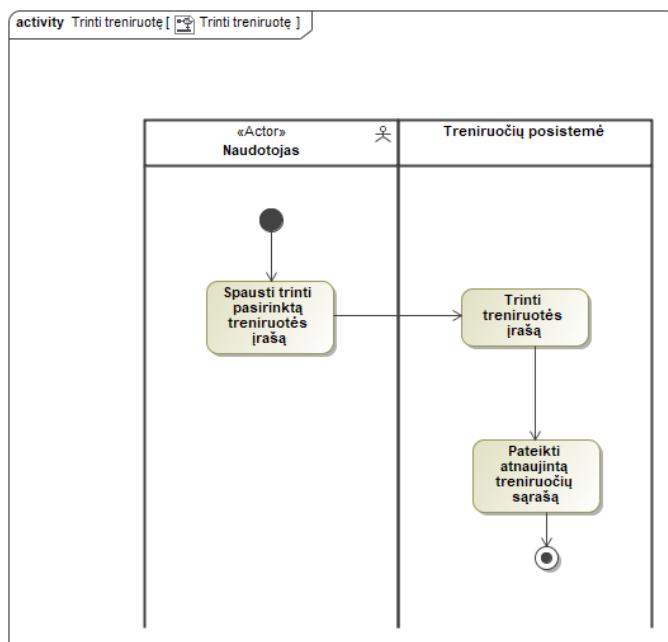


15 pav. Panaudojimo atvejo „Redaguoti treniruotę“ veiklos diagrama

15 lentelė. PA „Redaguoti treniruotę” specifikacija

PA. Redaguoti treniruotę	
Tikslas. Suteikti galimybę naudotojui redaguoti savo treniruotę.	
Aprašymas. PA suteikia galimybę naudotojui redaguoti savo treniruotę.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas redaguoja treniruotės informaciją.

Panaudojimo atvejo „Trinti treniruotę” veiklos diagramoje (16 pav.) parodyta, kad treniruotės trinimas prasideda paspaudus trinti mygtuką ant treniruotės įrašo. Tada ištrinamas treniruotės įrašas ir atnaujinama treniruočių lentelė. Veiklos diagramos specifikacija aprašoma 16 lent.

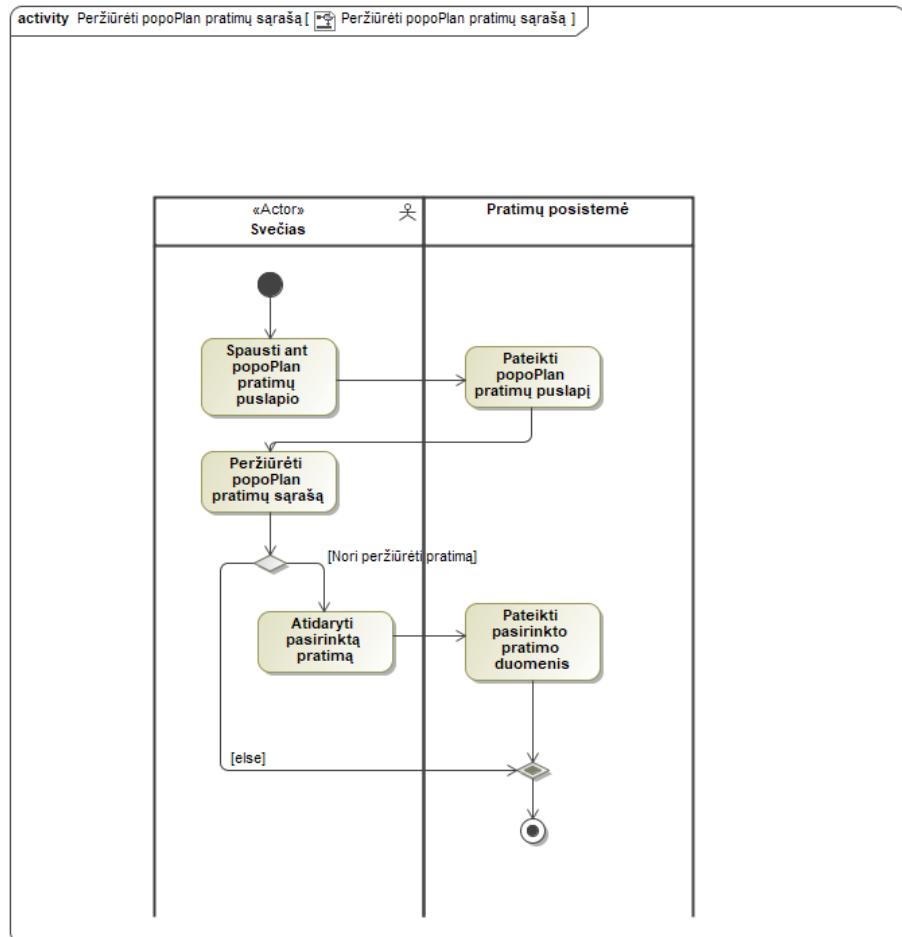


16 pav. Panaudojimo atvejo „Trinti treniruotę” veiklos diagrama

16 lentelė. PA „Trinti treniruotę” specifikacija

PA. Trinti treniruotę	
Tikslas. Suteikti galimybę naudotojui trinti savo treniruotę.	
Aprašymas. PA suteikia galimybę naudotojui trinti savo treniruotę.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas ištrina savo treniruotę.

Panaudojimo atvejo „Peržiūrėti PopoPlan pratimų sąrašą“ veiklos diagramoje (17 pav.) parodyta, kad popoPlan pratimų peržiūrėjimas prasideda paspaudus ant popoPlan pratimų lango. Tada nuvedama į popoPlan pratimų puslapį ir pateikiami popoPlan vieši pratimai. Jei žmogus yra prie sistemos ant mobilios aplikacijos, tada gali spausti ant pratimo ir bus pateikti pratimo duomenys, kuriuos naudotojas galės peržiūrėti. Veiklos diagramos specifikacija aprašoma 17 lent.

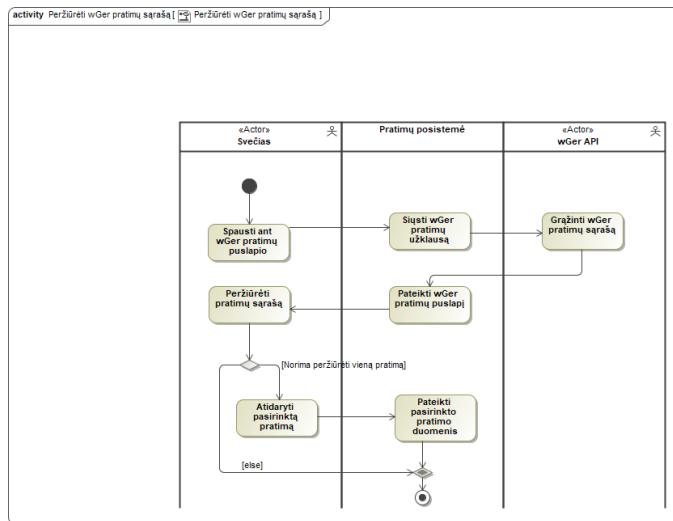


17 pav. Panaudojimo atvejo „Peržiūrėti PopoPlan pratimų sąrašą“ veiklos diagrama

17 lentelė. PA „Peržiūrėti PopoPlan pratimų sąrašą“ specifikacija

PA. Peržiūrėti PopoPlan pratimų sąrašą	
Tikslas. Suteikti galimybę svečiui peržiūrėti sistemos pratimų sąrašą.	
Aprašymas. PA suteikia galimybę svečiui peržiūrėti sistemos pratimų sąrašą.	
Prieš sąlyga	Svečias turi būti atsidaręs sistemą.
Aktorai	Svečias
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Svečias mato sistemos pratimų sąrašą.

Panaudojimo atvejo „Peržiūrėti wGer pratimų sąrašą“ veiklos diagramoje (18 pav.) parodyta, kad wGer pratimų peržiūrėjimas prasideda spaustus ant wGer pratimų puslapio. Tada vykdoma užklausa, kuri pasiima iš wGer API pratimų duomenis ir tada pateikia pratimų sąrašą. Naudotojas peržiūrėjęs pratimus, jei naudoja mobilią programą, tada gali spausti ant pratimo ir peržiūrėti jo duomenis. Veiklos diagramos specifikacija aprašoma 18 lent.

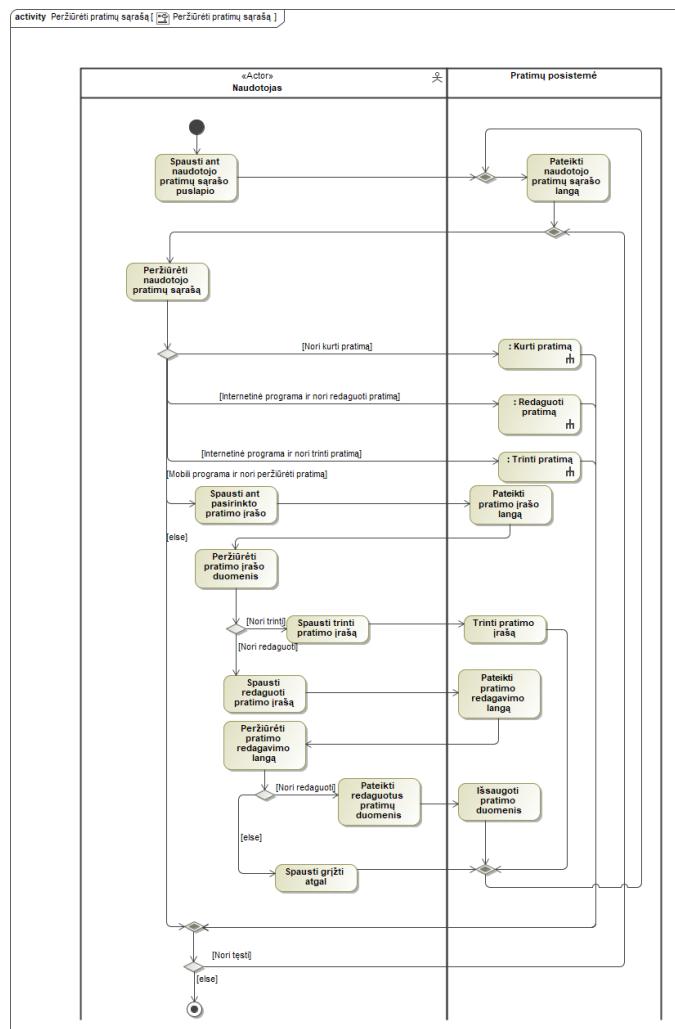


18 pav. Panaudojimo atvejo „Peržiūrėti wGer pratimų sąrašą“ veiklos diagrama

18 lentelė. PA „Peržiūrėti wGer pratimų sąrašą“ specifikacija

PA. Peržiūrėti wGer pratimų sąrašą	
Tikslas. Suteikti svečiui galimybę peržiūrėti wGet pratimų sąrašą.	
Aprašymas. PA suteikia galimybę peržiūrėti wGet pratimų sąrašą.	
Prieš sąlyga	Svečias turi būti įsijungęs sistemą.
Aktorai	Svečias, wGer API
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Svečias mato wGer pratimų sąrašą.

Panaudojimo atvejo „Peržiūrėti pratimų sąrašą“ veiklos diagramoje (19 pav.) parodyta, kad pratimų sąrašo peržiūrėjimas prasideda spaudžiant ant naudotojo pratimų puslapio. Tada pateikiamas naudotojo pratimų puslapis. Peržiūrėjęs pratimų įrašus, naudotojas gali kurti pratimą. Jei sistema naudojama per naršykį - galima redaguoti pratimą ir trinti pratimą, jei naudojama mobili programa, tada reikia spausti ant pratimo įrašo norint atlikti trinimą arba įrašo redagavimą. Veiklos diogramos specifikacija aprašoma 19 lent.

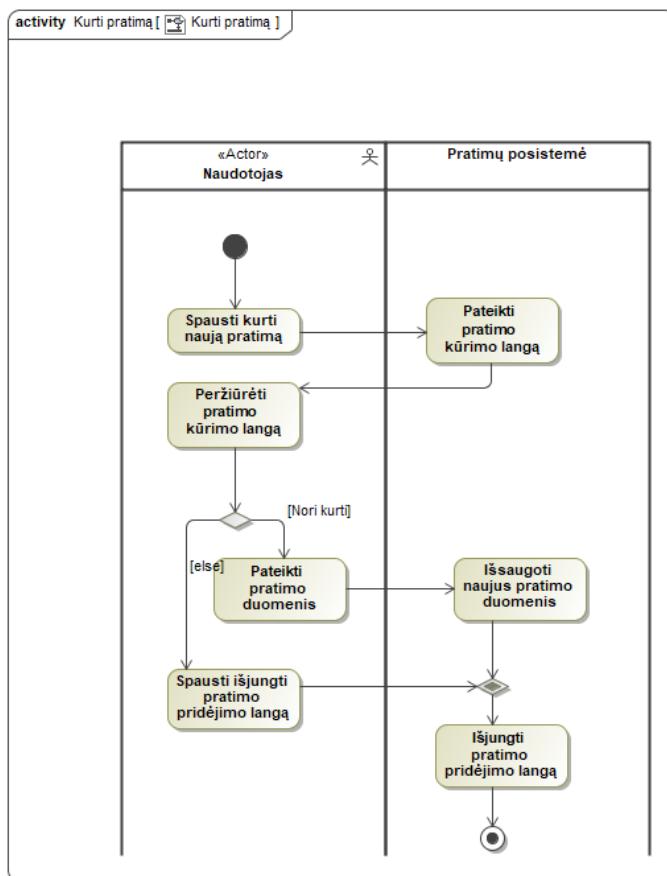


19 pav. Panaudojimo atvejo „Peržiūrėti pratimų sąrašą“ veiklos diagrama

19 lentelė. PA „Peržiūrēti pratimų sąrašą” specifikacija

PA. Peržiūrēti pratimų sąrašą	
Tikslas. Suteikti galimybę naudotojui peržiūrēti savo pratimų sąrašą.	
Apaščymas. PA suteikia galimybę naudotojui peržiūrēti savo pratimų sąrašą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas mato savo pratimų sąrašą.

Panaudojimo atvejo „Kurti pratimą” veiklos diagramoje (20 pav.) parodyta, kad pratimo kūrimas prasideda paspaudus kurti naują pratimą. Tada atidaromas pratimo kūrimo langas. Paskui naudotojas gali apsispręsti arba iš jungti pratimo pridėjimo langą, arba užpildyti laukus ir pateikti pratimo duomenis, taip išsaugojant pratimo įrašą. Galiausiai pratimo pridėjimo langas iš jungiamas. Veiklos diagramos specifikacija aprašoma 20 lent.

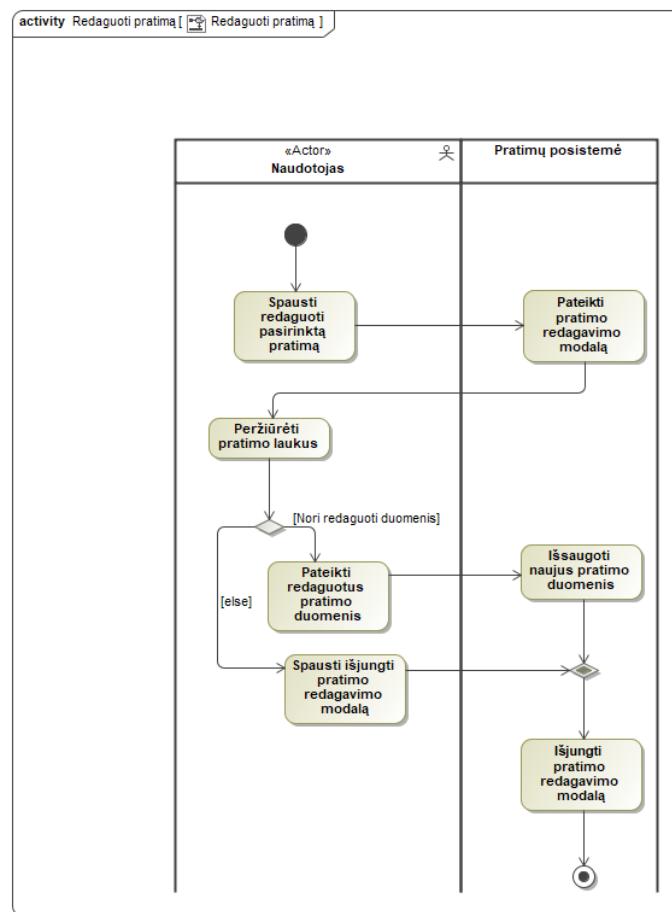


20 pav. Panaudojimo atvejo „Kurti pratimą” veiklos diagrama

20 lentelė. PA „Kurti pratimą” specifikacija

PA. Kurti pratimą	
Tikslas. Suteikti galimybę naudotojui kurti pratimą.	
Aprašymas. PA suteikia galimybę naudotojui kurti pratimą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas sukuria naują pratimą.

Panaudojimo atvejo „Redaguoti pratimą” veiklos diagramoje (21 pav.) parodyta, kad pratimo redagavimas prasideda paspaudus ant įrašo redaguoti, tada pateikiamas pratimo redagavimo langas. Peržiūrėjus duomenis, naudotojas gali pasirinkti, arba išjungti pratimo redagavimo langą, arba redaguoti duomenis ir išsaugoti juos. Galiausiai pratimo redagavimo langas yra uždaromas. Veiklos diagramos specifikacija aprašomos 21 lent.

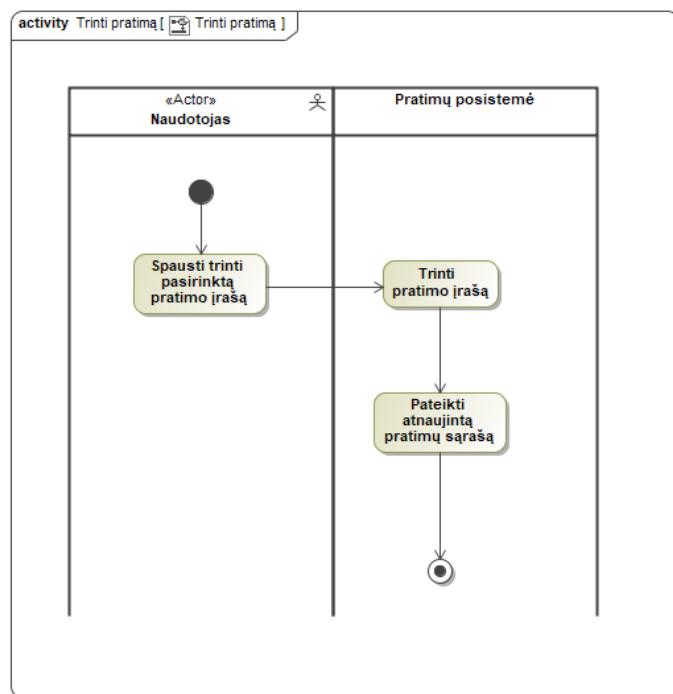


21 pav. Panaudojimo atvejo „Redaguoti pratimą” veiklos diagrama

21 lentelė. PA „Redaguoti pratimą” specifikacija

PA. Redaguoti pratimą	
Tikslas. Suteikti galimybę naudotojui redaguoti savo pratimą.	
Aprašymas. PA suteikia galimybę naudotojui redaguoti savo pratimą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas redaguoja savo pratimo duomenis.

Panaudojimo atvejo „Trinti pratimą” veiklos diagramoje (22 pav.) parodyta, kad pratimo trinimas prasideda paspaudus pratimo trinimo mygtuką. Tada ištrinamas pratimo įrašas ir atnaujinama pratimų lentelė. Veiklos diagramos specifikacija aprašoma 22 lent.

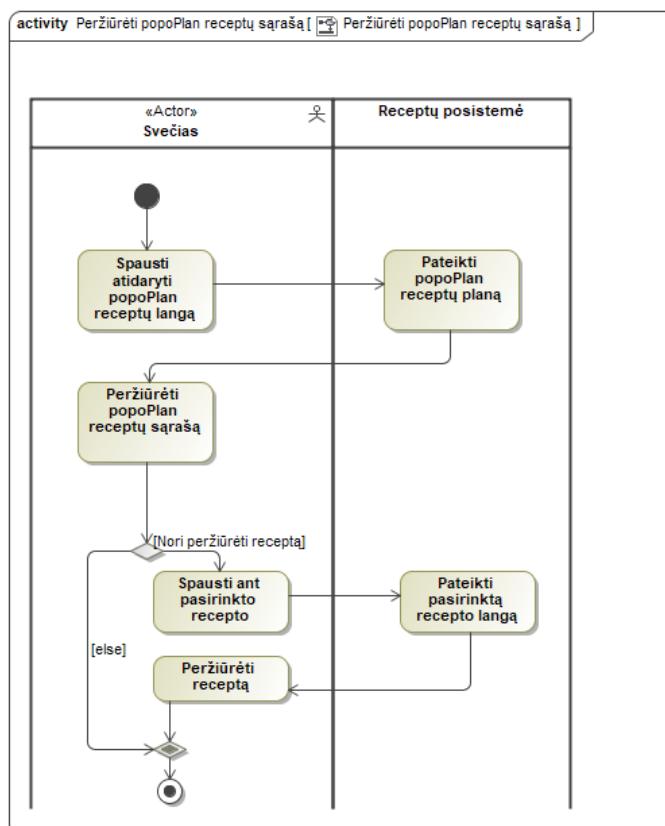


22 pav. Panaudojimo atvejo „Trinti pratimą” veiklos diagrama

22 lentelė. PA „Trinti pratimą” specifikacija

PA. Trinti pratimą	
Tikslas. Suteikti galimybę naudotojui ištrinti savo pratimą.	
Aprašymas. PA suteikia galimybę naudotojui ištrinti savo pratimą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas ištrina savo pratimą.

Panaudojimo atvejo „Peržiūrėti PopoPlan receptų sąrašą” veiklos diagramoje (23 pav.) parodyta, kad popoPlan receptų peržiūrėjimas prasideda paspaudus ant popoPlan receptų lango. Tada nuvedama į popoPlan receptų puslapį ir pateikiami popoPlan vieši receptai. Jei žmogus yra prie sistemos ant mobilios aplikacijos, tada gali spausti ant recepto ir bus pateikti recepto duomenys, kuriuos naudotojas galės peržiūrėti. Veiklos diagramos specifikacija aprašoma 23 lent.

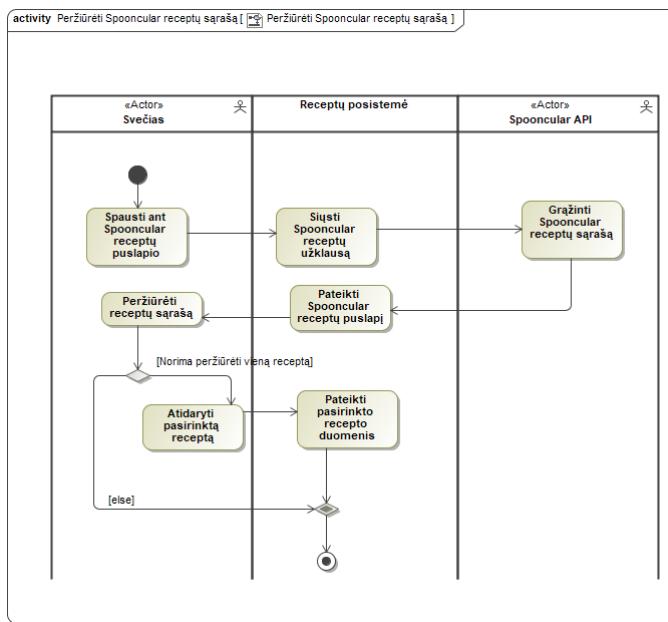


23 pav. Panaudojimo atvejo „Peržiūrėti PopoPlan receptų sąrašą” veiklos diagrama

23 lentelė. PA „Peržiūrėti PopoPlan receptų sąrašą” specifikacija

PA. Peržiūrėti PopoPlan receptų sąrašą	
Tikslas. Suteikti galimybę svečiui peržiūrėti sistemos receptų sąrašą.	
Aprašymas. PA suteikia galimybę svečiui peržiūrėti sistemos receptų sąrašą.	
Prieš sąlyga	Svečias turi būti įsijungęs sistemą.
Aktorai	Svečias
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Svečias mato sistemos receptų sąrašą.

Panaudojimo atvejo „Peržiūrėti Spooncular receptų sąrašą” veiklos diagramoje (24 pav.) parodyta, kad Spooncular receptų peržiūrėjimas prasideda paspaudus ant Spooncular receptų puslapio. Tada vykdoma užklausa, kuri pasiima iš Spooncular API receptų duomenis ir tada pateikia receptų sąrašą. Naudotojas peržiūrėjęs receptus, jei naudoja mobilią programą, tada gali spausti ant recepto ir peržiūrėti jo duomenis. Veiklos diagramos specifikacija aprašoma 24 lent.

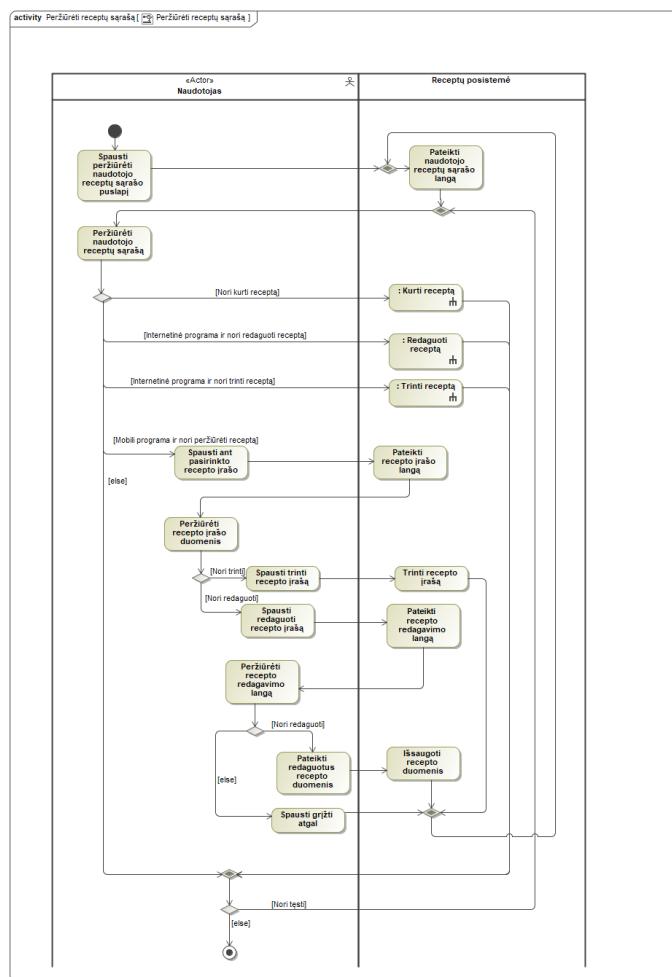


24 pav. Panaudojimo atvejo „Peržiūrėti Spooncular receptų sąrašą” veiklos diagrama

24 lentelė. PA „Peržiūrēti Spooncular receptų sąrašą“ specifikacija

PA. Peržiūrēti Spooncular receptų sąrašą	
Tikslas. Suteikti galimybę svečiui peržiūrēti spooncular receptų sąrašą.	
Aprašymas. PA suteikia galimybę svečiui peržiūrēti spooncular receptų sąrašą.	
Prieš sąlyga	Svečias turi būti įsijungęs sistemą.
Aktorai	Svečias, Spooncular API
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Svečias mato spooncular receptų sąrašą.

Panaudojimo atvejo „Peržiūrēti receptų sąrašą“ veiklos diagramoje (25 pav.) parodyta, kad receptų sąrašo peržiūrėjimas prasideda spaudžiant ant naudotojo receptų puslapio. Tada pateikiamas naudotojo receptų puslapis. Peržiūrėjės receptų įrašus, naudotojas gali kurti receptą. Jei sistema naudojama per naršykli - galima redaguoti receptą ir trinti receptą, jei naudojama mobilė programa, tada reikia spausti ant recepto įrašo norint atliliki trinimą arba įrašo redagavimą. Veiklos diagramos aprašoma 25 lent.



25 pav. Panaudojimo atvejo „Peržiūrēti receptų sąrašą“ veiklos diagrama

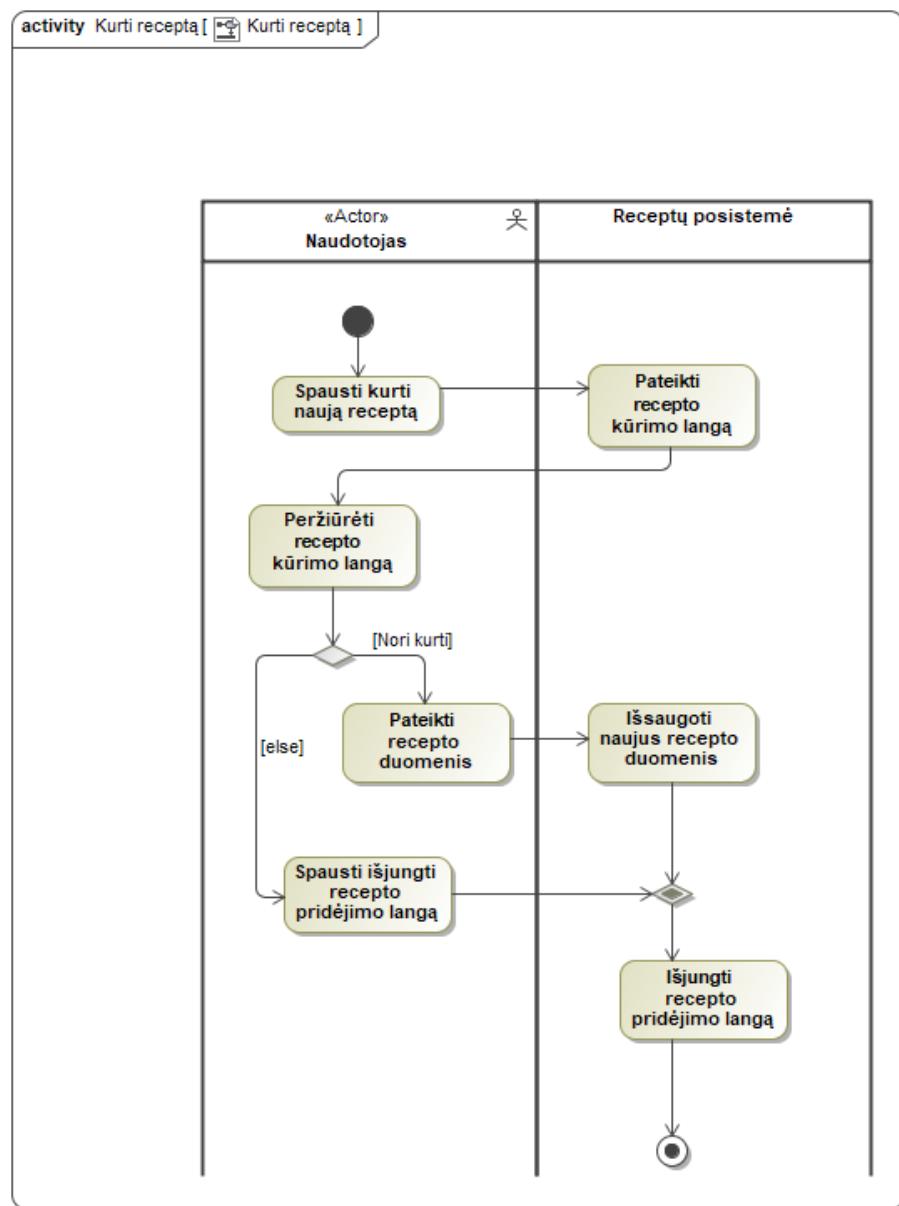
25 lentelė. PA „Peržiūrėti receptų sąrašą” specifikacija

PA. Peržiūrėti receptų sąrašą		
Tikslas. Suteikti galimybę naudotojui peržiūrėti savo receptų sąrašą.		
Aprašymas. PA suteikia galimybę naudotojui peržiūrėti savo receptų sąrašą.		
Prieš sąlyga		Naudotojas turi būti prisijungęs sistemos.
Aktorai		Naudotojas
Susiję panaudojimo atvejai	Apimantys PA	-
	Išplečiantys PA	6 “Kurti receptą”, 15 “Redaguoti receptą”, 10 “Trinti receptą”
	Specializuojantys PA	-
Po sąlyga		Naudotojas mato savo receptų sąrašą.

Panaudojimo atvejo „Kurti receptą” veiklos diagramoje (26 pav.) parodyta, kad recepto kūrimas prasideda paspaudus kurti naują receptą. Tada atidaromas recepto kūrimo langas. Paskui naudotojas gali apsispresti arba išjungti recepto pridėjimo langą, arba užpildyti laukus ir pateikti recepto duomenis, taip išsaugojant recepto įrašą. Galiausiai recepto pridėjimo langas išjungiamas. Veiklos diagramos specifikacija aprašoma 26 lent.

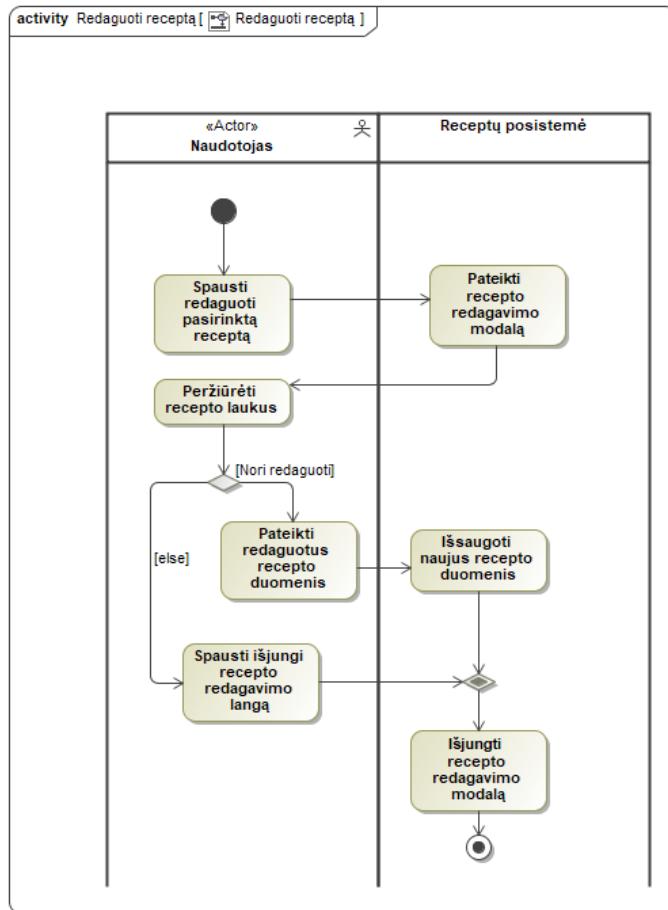
26 lentelė. PA „Kurti receptą” specifikacija

PA. Kurti receptą		
Tikslas. Suteikti galimybę naudotojui sukurti savo naują recepto įrašą.		
Aprašymas. PA suteikia galimybę naudotojui sukurti savo naują recepto įrašą.		
Prieš sąlyga		Naudotojas turi būti prisijungęs sistemos.
Aktorai		Naudotojas
Susiję panaudojimo atvejai	Apimantys PA	-
	Išplečiantys PA	-
	Specializuojantys PA	-
Po sąlyga		Naudotojas sukuria savo naują receptą.



26 pav. Panaudojimo atvejo „Kurti receptą” veiklos diagrama

Panaudojimo atvejo „Redaguoti receptą” veiklos diagramoje (27 pav.) parodyta, kad recepto redagavimas prasideda paspaudus ant įrašo redaguoti, tada pateikiamas recepto redagavimo langas. Peržiūrėjus duomenis, naudotojas gali pasirinkti, arba išjungti recepto redagavimo langą, arba redaguoti duomenis ir išsaugoti juos. Galiausiai recepto redagavimo langas yra uždaromos. Veiklos diagramos specifikacija aprašomos 27 lent.

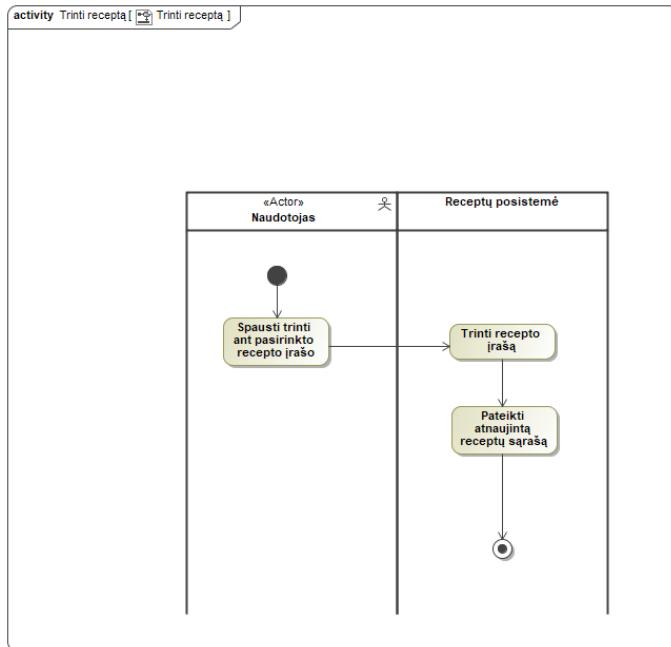


27 pav. Panaudojimo atvejo „Redaguoti receptą” veiklos diagrama

27 lentelė. PA „Redaguoti receptą” specifikacija

PA. Redaguoti receptą	
Tikslas. Suteikti galimybę naudotojui redaguoti savo receptą.	
Aprašymas. PA suteikia galimybę naudotojui redaguoti savo receptą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	Išplečiantys PA
	Specializuojantys PA
Po sąlyga	Naudotojas pakeitė savo recepto duomenis.

Panaudojimo atvejo „Trinti receptą” veiklos diagramoje (28 pav.) parodyta, kad recepto trinimo veikla prasideda paspaudus trinti receptą mygtuką. Tada receptas ištrinamas ir atnaujinama receptų lentelė. Veiklos diagramos specifikacija 28 lent.



28 pav. Panaudojimo atvejo „Trinti receptą” veiklos diagrama

28 lentelė. PA „Trinti receptą” specifikacija

PA. Trinti receptą	
Tikslas. Suteikti galimybę naudotojui trinti savo receptą.	
Aprašymas. PA suteikia galimybę naudotojui trinti savo receptą.	
Prieš sąlyga	Naudotojas turi būti prisijungęs sistemos.
Aktorai	Naudotojas
Susiję panaudojimo atvejai	Apimantys PA
	-
	Išplečiantys PA
Specializuojantys PA	-
Po sąlyga	Naudotojas ištrina savo receptą.

2.1.3. Naudotojo sasajos specifikacija

Grafinei naudotojo sasajai nebuvu iškelti jokie reikalavimai.

2.1.4. Realizacijai keliami reikalavimai

Sveikatingumo planavimo sistemai iškelti nefunkciniai reikalavimai:

1. Mobile programa veikia ant Android ir iOS operacinių sistemų.
2. Kodo talpinimo vieta - Github.
3. Kodos versijavimas kontroliuojamas naudojant git.
4. Projekto naudotojo sasaja yra anglų kalba.
5. Serverio API atitinka REST principus.

2.1.5. Techninė specifikacija

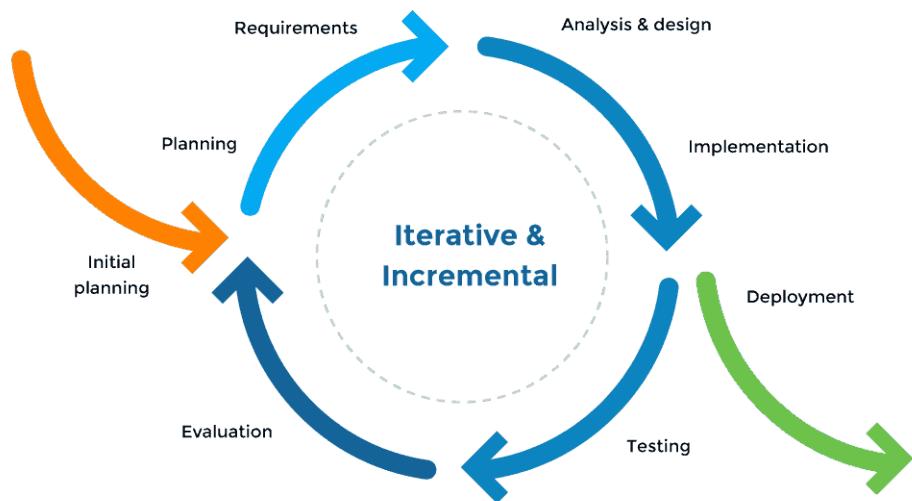
Sistemai panaudoti reikia techninės įrangos:

- Kompiuterio, turinčio „Windows”, „Linux” arba „macOS” operacinę sistemą su prieiga prie interneto.
- Mobiliojo telefono, turinčio Android arba iOS operacinę sistemą ir prijungto prie internetinio ryšio.
- Duomenų bazės MongoDB.
- Projekto paleidimui - npm arba yarn paketų tvarkyklės.

2.2. Projektavimo metodai

2.2.1. Projektavimo valdymas ir eiga

Sistemos realizacijos metu buvo taikomas iteracinis programų kūrimo proceso modelis, kurio žingsniai matomi (29 pav.)



29 pav. Iteracinis programų kūrimo proceso modelis

Pasirinkimas taikyti šį programų kūrimo proceso modelį lėmė neaiškūs funkcijų apibrėžimai projekto pradžioje. Kiekvienos iteracijos metu buvo tikslinama ir redaguojama, kol buvo gaunami norimi rezultatai. Iš viso buvo 4 iteracijos, kurioms skirtas dviejų savaičių laikas atlikimui.

2.2.2. Projektavimo technologija

Siekiant aprašyti projektą vizualiai, buvo sukurtos diagramos naudojant UML standartus. Visų funkcijų atvaizdavmui buvo pasitelkta panaudojimo atvejų diagrama, funkcijų eigai - veiklų diagrama, funkcijų veikimui iš programinės pusės - sekų ir t.t. Diagramų projektavimo įrankis - Magic Draw.

2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemas

Javascript paketų tvarkykla - npm. Kliento dalyje buvo naudojama Javascript programavimo kalba su Vue karkasu. Kartu buvo naudojamas Babel kompiliatorius, sukompiliuojantis naujausias js funkcijas, kurias palaiko senesnės naršyklės.

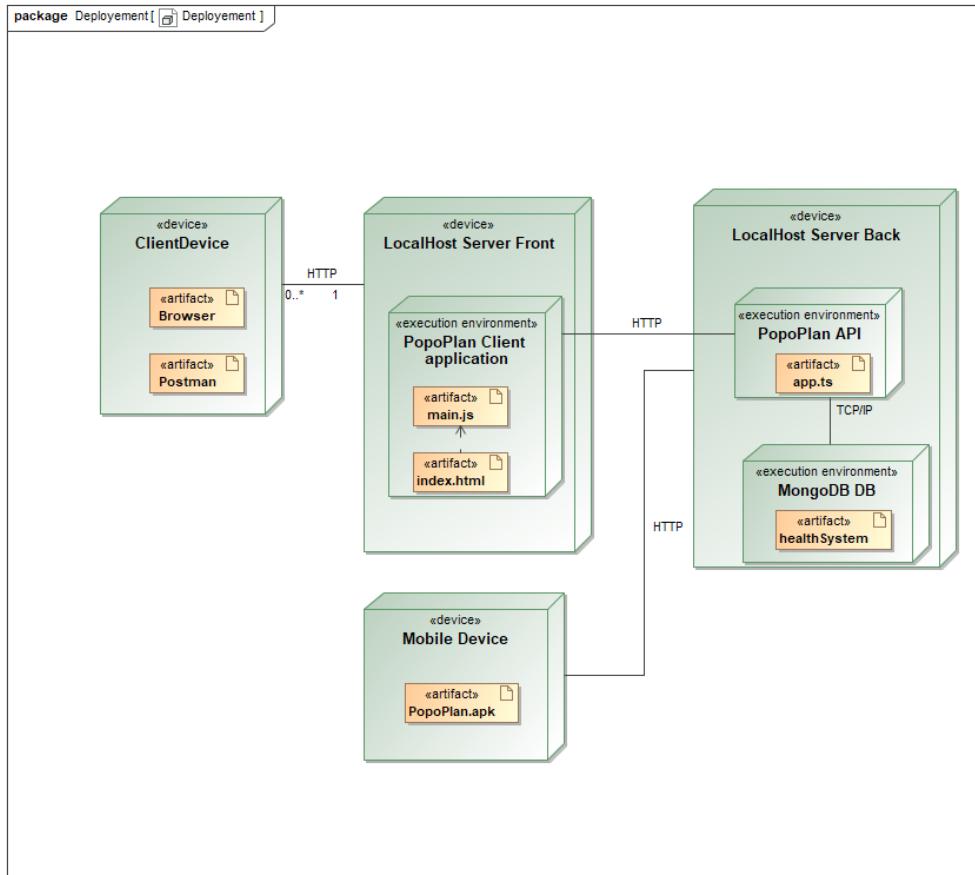
Serverio dalis naudoja Typescript programavimo kalbą. Duomenys talpinami MongoDB duomenų bazėje. Siekiant lengvai sudaryti duomenų modelius buvo naudojamas Mongoose Object-relational mapper. Puslapių keliai sukurti su Express.js karkasu. Mobilė programa sukurta naudojant NativeScript su Vue ir TypeScript programavimo kalba. Microsoft Visual Studio Code kodo redaktorius buvo naudojamas programavimui.

Programinė įranga ataskaitos kūrimui - L^AT_EX.

2.3. Sistemos projektas

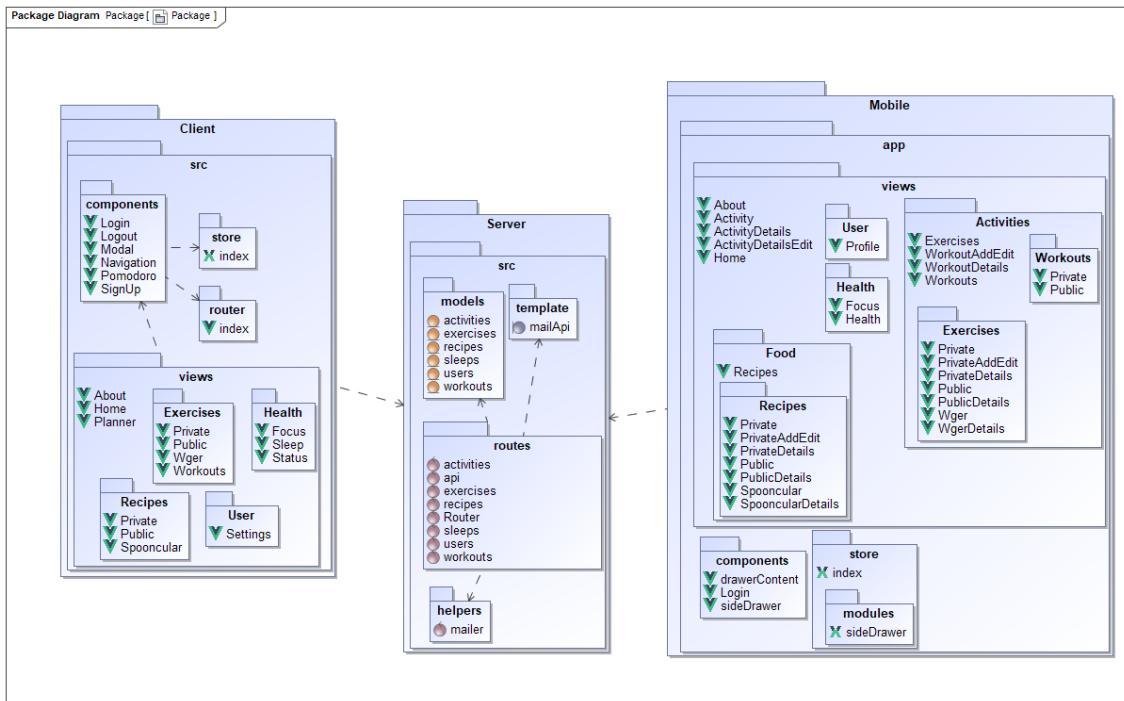
2.3.1. Statinis sistemos vaizdas

PopoPlan sistemą sudaro kelios dalys, jas galima pamatyti UML diegimo diagramoje (30 pav.). Kadangi projektas kol kas paleidžiamas lokalioje sistemoje, naudotojo prietaisui reikia prisijungti į tą patį tinklą. Tada žmogus gali prisijungti prie sistemos per naršyklę, naudojant Postman, arba su mobiliuoju prietaisu. Visos užklausos vyksta HTTP protokolu. Užklausos, einančios iš kliento dalies ir mobiliosios programos į serverį, taip pat komunikuoja HTTP protokolu, bet serveriu iš imant duomenis iš duomenų bazės - TCP/IP protokolu.



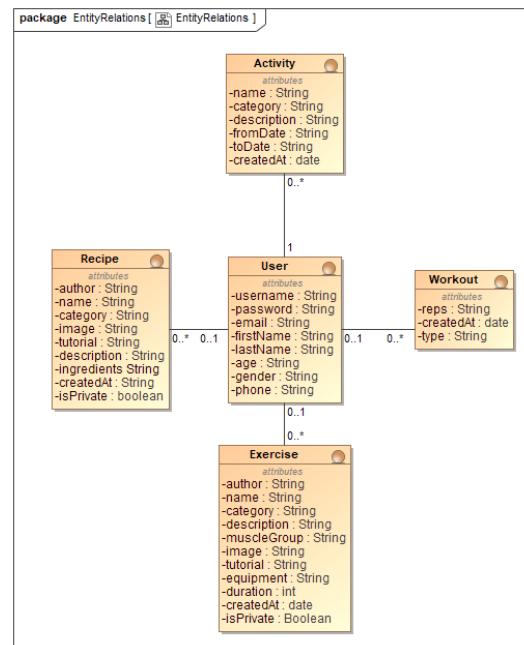
30 pav. Sistemos PopoPlan diegimo diagrama

Naudojantis UML paketų diagrama galima parodyti projekto architektūrą. Diagramoje pavaizduota sistema (31 pav.) yra kelių sluoksnių (angl. *Multitier architecture*). Tai reiškia, kad kodas padalintas į kelias dalis, šiuo atveju - tris, t. y. kliento, serverio ir duomenų bazės. Tokiu būdu galima atskirti komponentus, tvarkingesnis kodas, greitesnis veikimas, lengvesnis prieinamumas prie sistemos. Pačioje kliento dalyje yra Vue karkasas, kuris naudoja MVVM (angl. *Model-View-ViewModel*) architektūrą, suteikiančią lengvą duomenų bendrovimą tarp duomenų modelio ir atvaizdavimo šablono.



31 pav. Sistemos PopoPlan paketų diagramma

Pavaizduotoje UML klasių ryšių diagramoje (32 pav.) galima pamatyti, kokie objektai yra susiję tarpusavyje. Šiuo atveju naudotojas gali turėti daug receptų, treniruočių, veiklų ir pratimų.

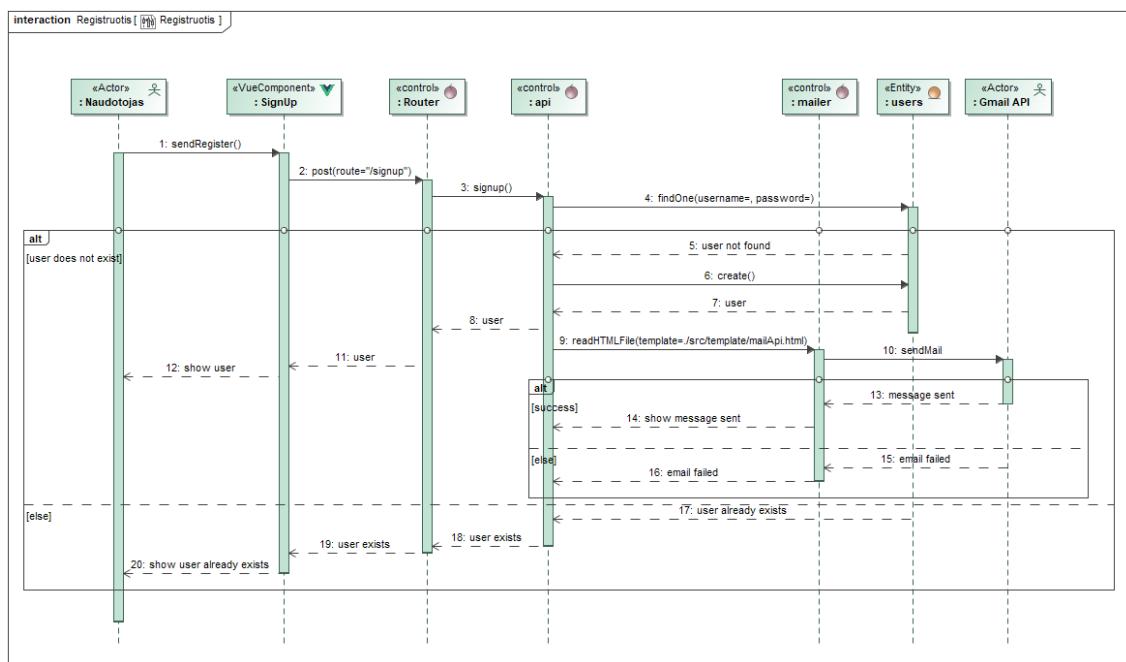


32 pav. Sistemos PopoPlan klasių diagramma

2.3.2. Dinaminis sistemos vaizdas

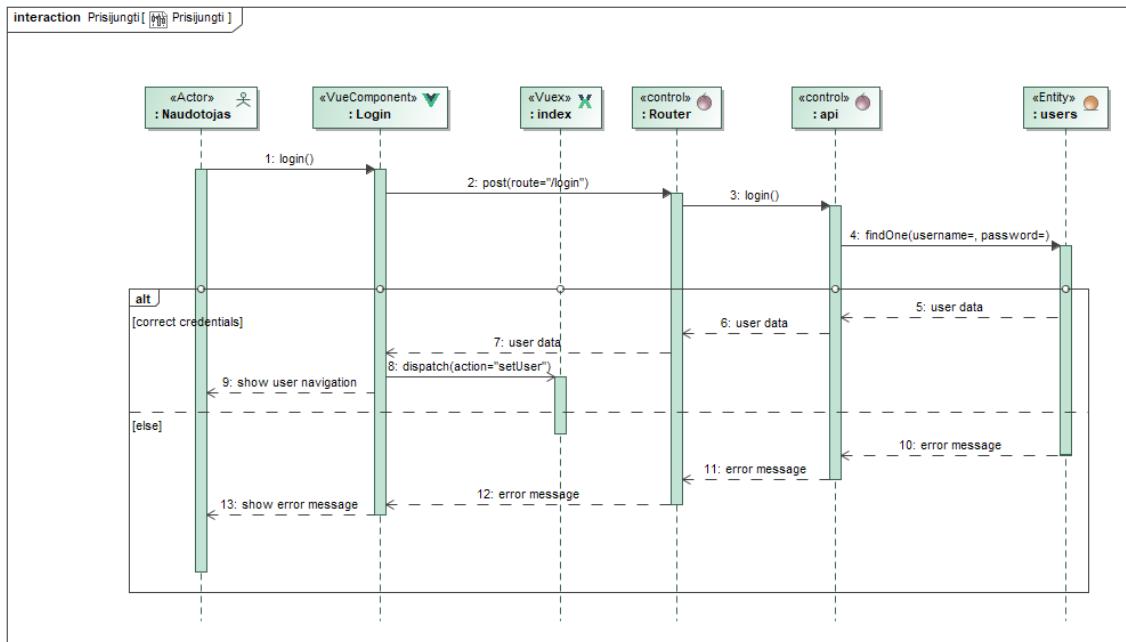
Siekiant atskleisti PopoPlan sistemos dinaminį vaizdą, reikalingos sekų diagramos. Kiekvieno panaudojimo atvejo metu komponentai sąveikauja, taip pat atskleidžiami tarpusavio kvietiniai.

Panaudojimo atvejo „Registruotis“ sekų diagramoje (33 pav.) registracijos dinaminis vaizdas prasideda suvedus duomenis ir paspaudžiant registruotis mygtuką, kuris iškviečia sendRegister funkciją. Visi duomenys nurodytu keliu atsiranda Router komponent ir perduodama į api metodą signup, kuriame yra apdorojama duota informacija. Pirmiausia yra atliekama naudotojo patikra ir imami duomenys iš users entity, jei naudotojo paskyra rasta, tada grąžinama, kad toks paskyros įrašas yra. Jei paskyros nėra, tada api kviečia create metodą users klasei ir sukuriama paskyra ir grąžinamas naudotojo įrašas į SignUp komponentą ir iki Naudotojo. Tuo tarpu po naudotojo grąžinimo, kviečiamas metodas, kuris nuskaito paštui sukurtą html šabloną ir jį paduoda į GMail API, kuris nusiunčia laišką į naudojo paskyrą.



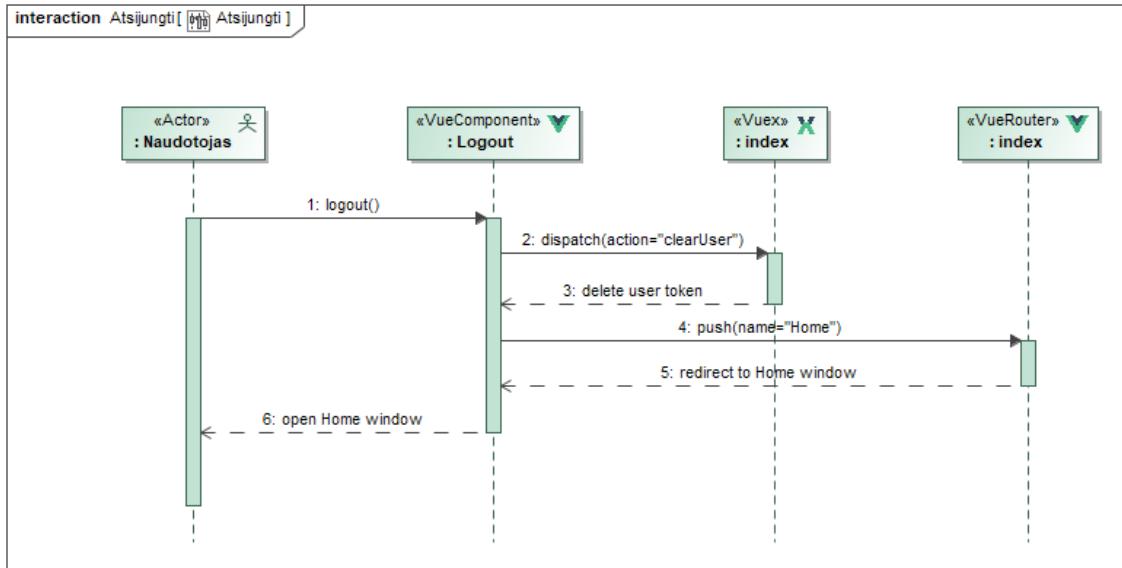
33 pav. Panaudojimo atvejo „Registruotis“ sekų diagrama

Panaudojimo atvejo „Prisijungti“ sekų diagramoje (34 pav.) prisijungimo pradžioje Naudotojas suveda į login komponentą duomenis ir spaudžia prisijungti, tada iškviečiamas login metodas. Po to duomenys paduodami į Router komponentą. Router komponentas perduodamas duomenis kreipiasi į anoniminį login metodą, tada vyksta duomenų patikrinimas. Users modelis ieško paduotos paskyros, jei visi duomenys sutampa, į login komponentą grįžta naudotojo duomenys. Paskui kviečiamas asinchroninis metodas dispatch, kuris nustato naudotojo prisijungimą, suteikiant jam daugiau privilegijų (naudotojas gali pasiekti savo paskyros informaciją) naviguojant. Jei paskyros duomenys buvo blogi, sugrįžtė naudotojui klaidos pranešimas.

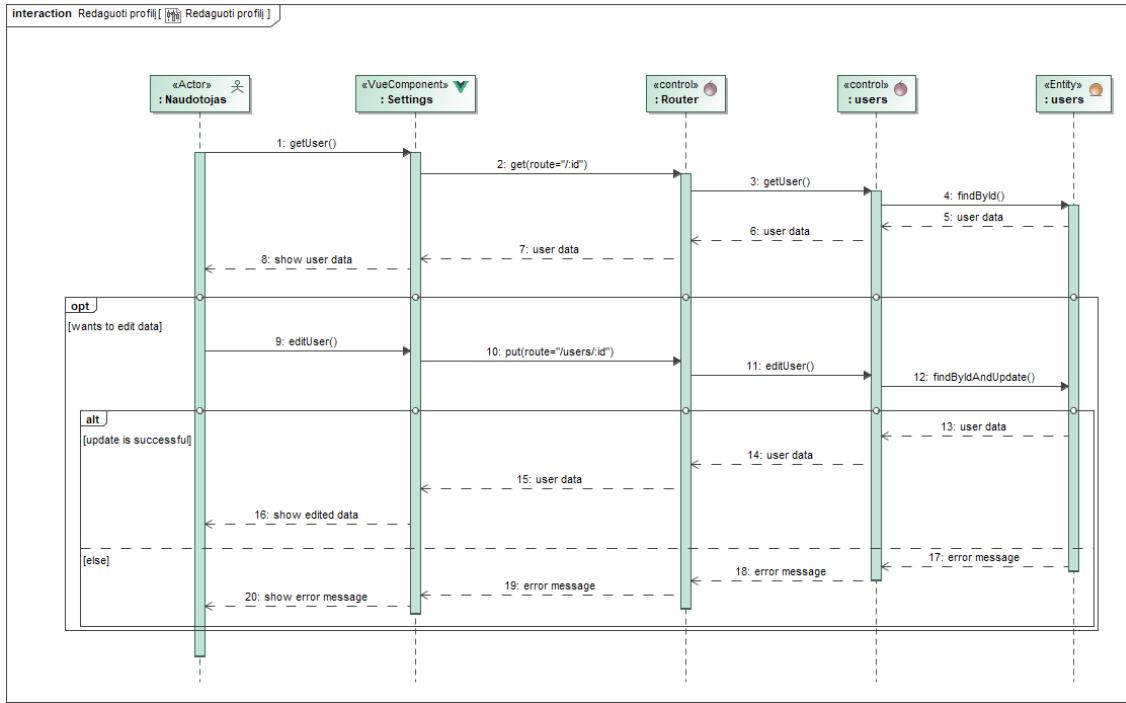


34 pav. Panaudojimo atvejo „Prisijungti“ sekų diagrama

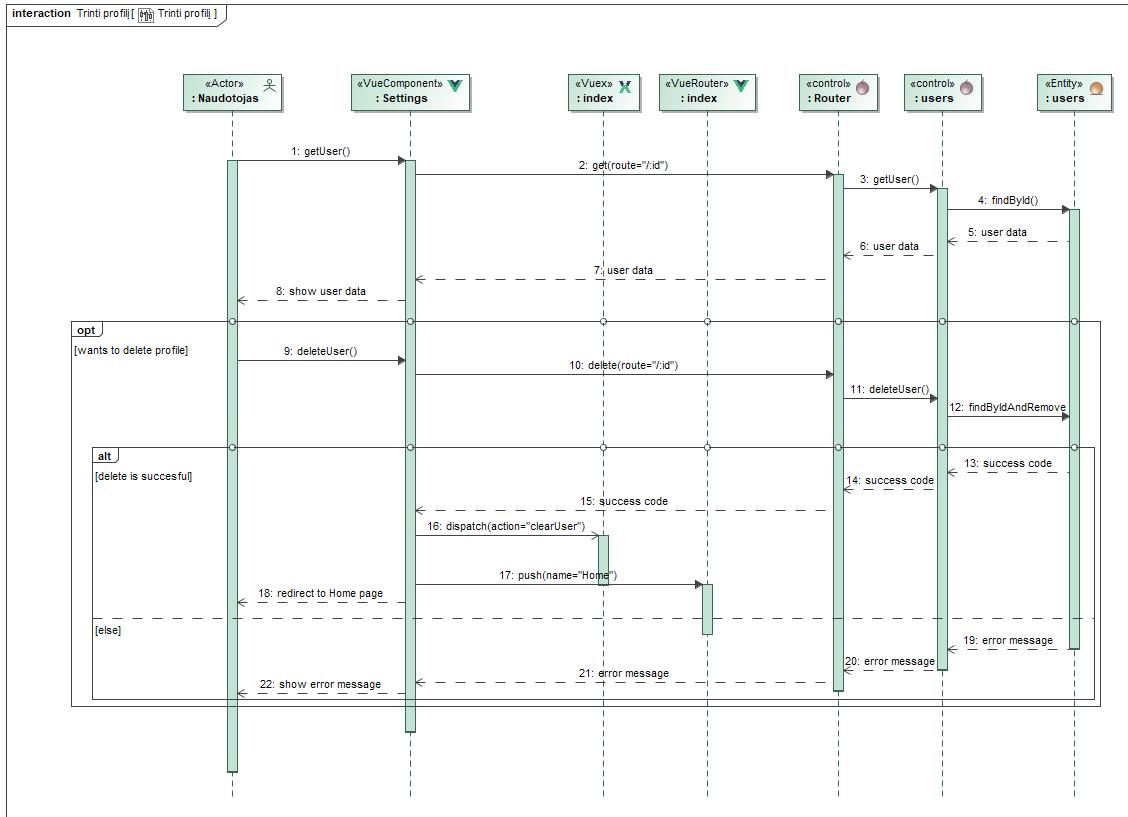
Panaudojimo atvejo „Atsijungti” sekų diagramoje (35 pav.) siekiant atsijungti nuo sistemos, kreipiamasi į logout komponentą, kuris kreipiasi į vuex ir ištrina naudotojo duomenis, tada vueRouter nukreipia naudotoją į pagrindinį puslapį.



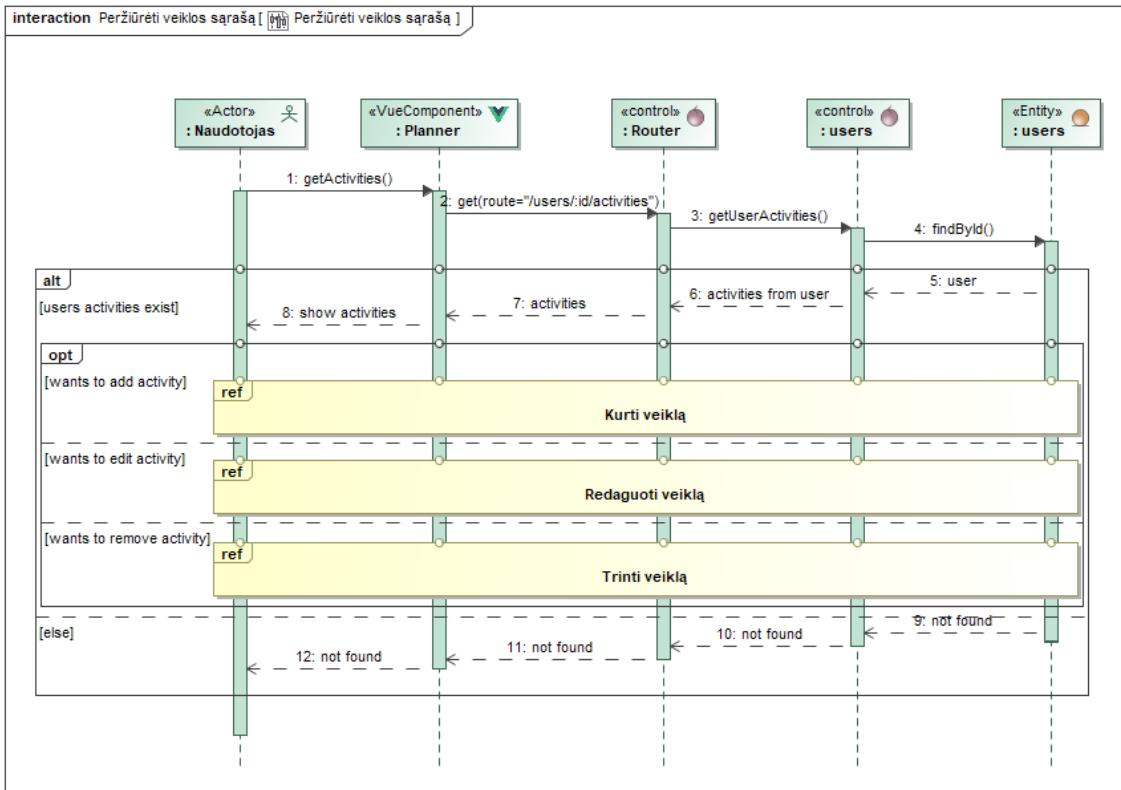
35 pav. Panaudojimo atvejo „Atsijungti” sekų diagrama



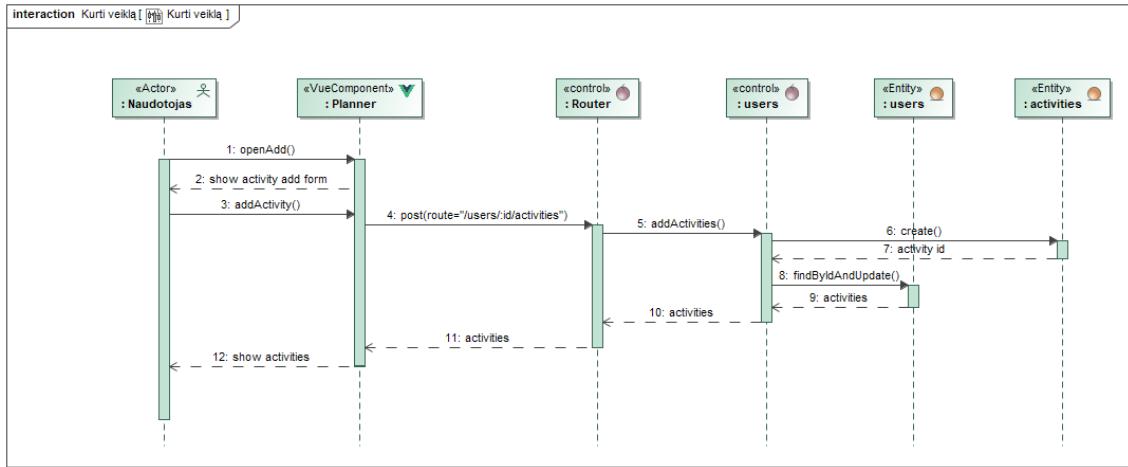
36 pav. Panaudojimo atvejo „Redaguoti profilį” sekų diagramma



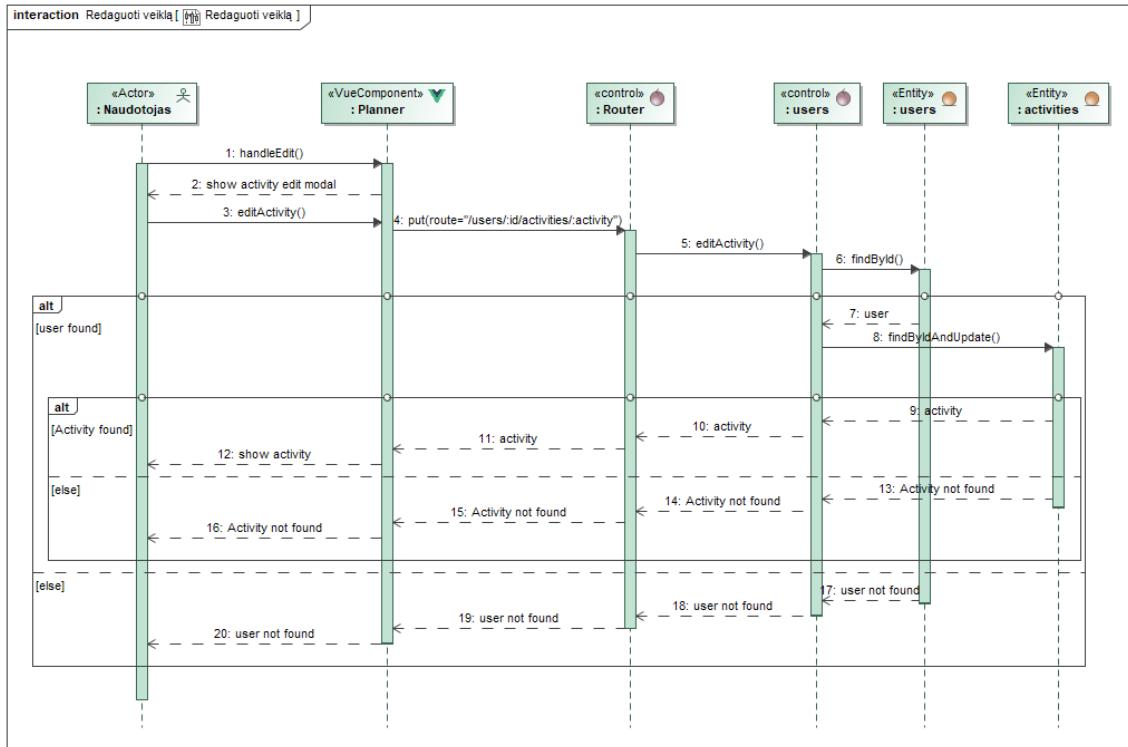
37 pav. Panaudojimo atvejo „Trinti profili“ sekų diagramma



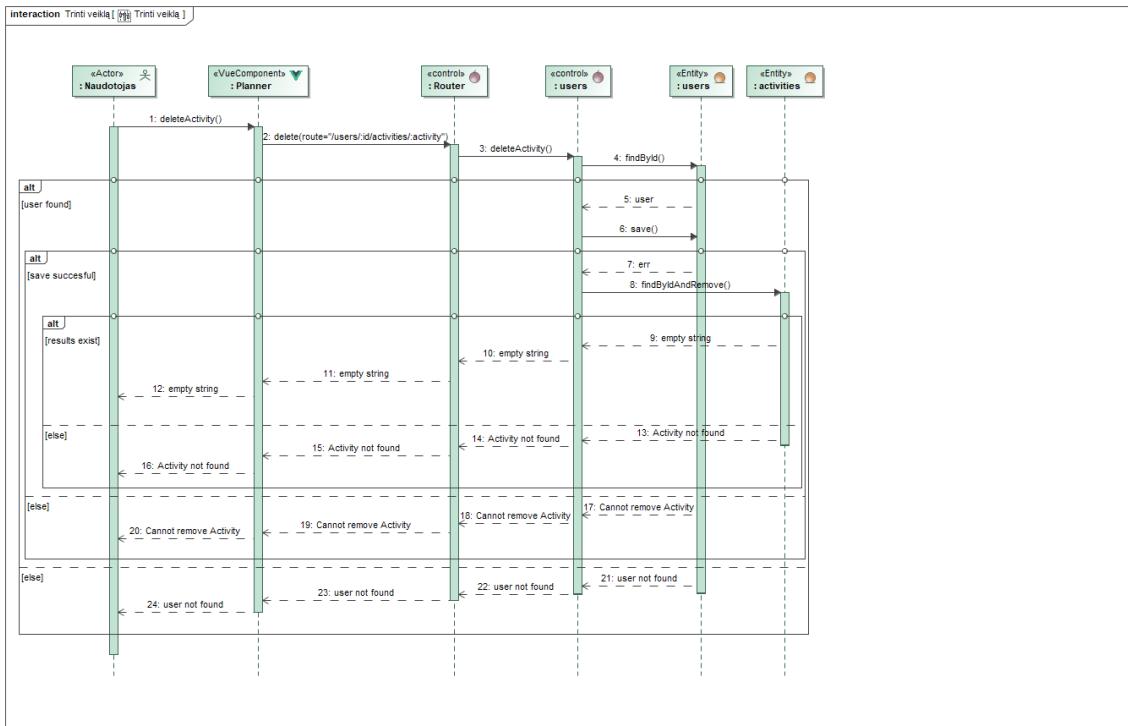
38 pav. Panaudojimo atvejo „Peržiūrėti veiklos sąrašą“ sekų diagrama



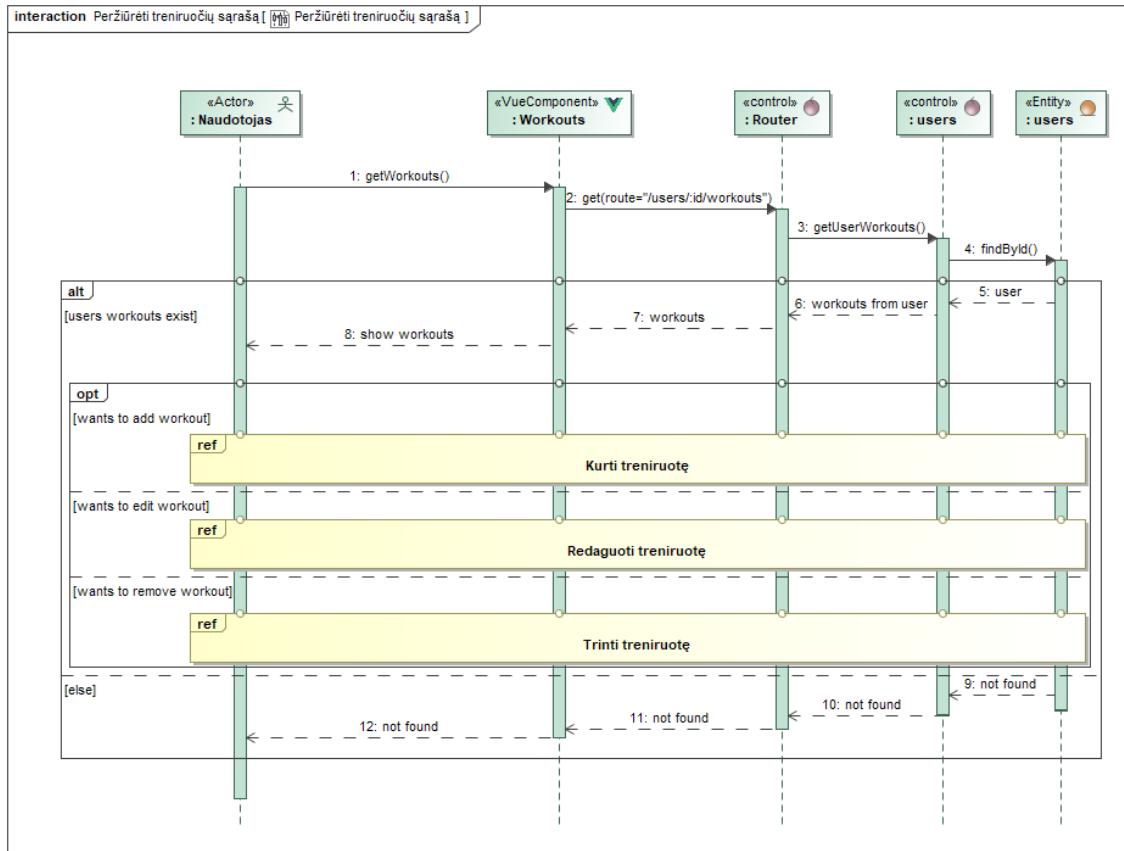
39 pav. Panaudojimo atvejo „Kurti veiklą” sekų diagrama



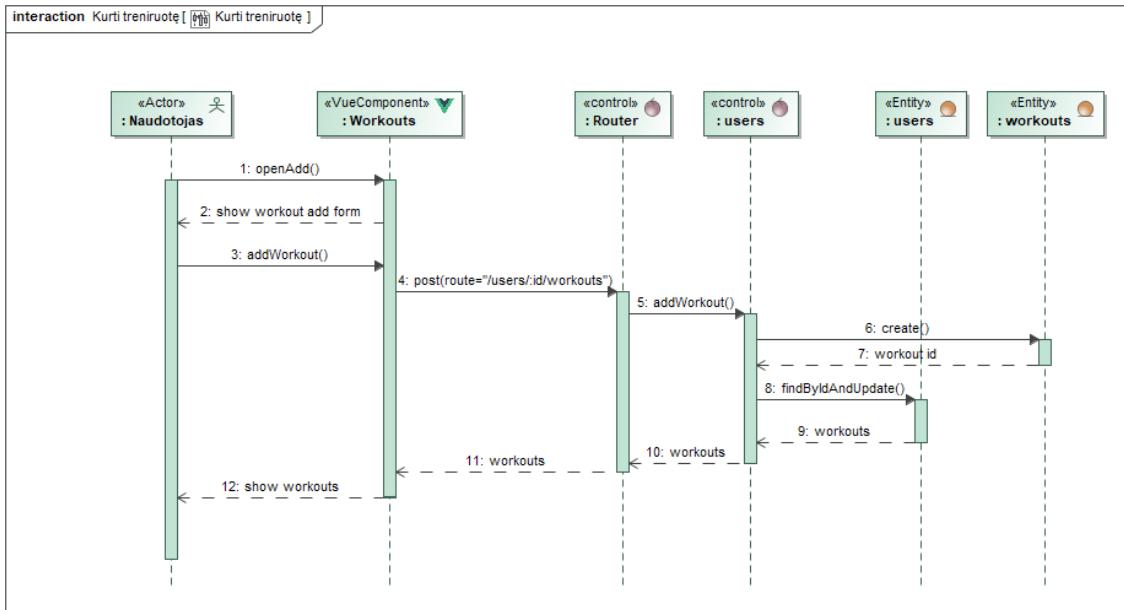
40 pav. Panaudojimo atvejo „Redaguoti veiklą“ sekų diagrama



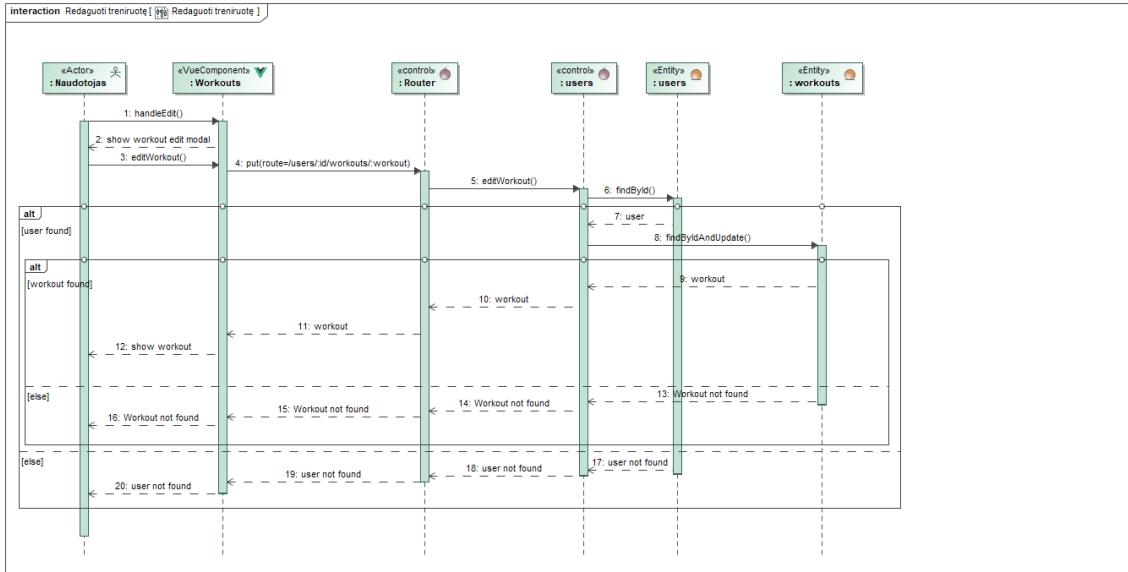
41 pav. Panaudojimo atvejo „Trinti veiklą“ sekų diagrama



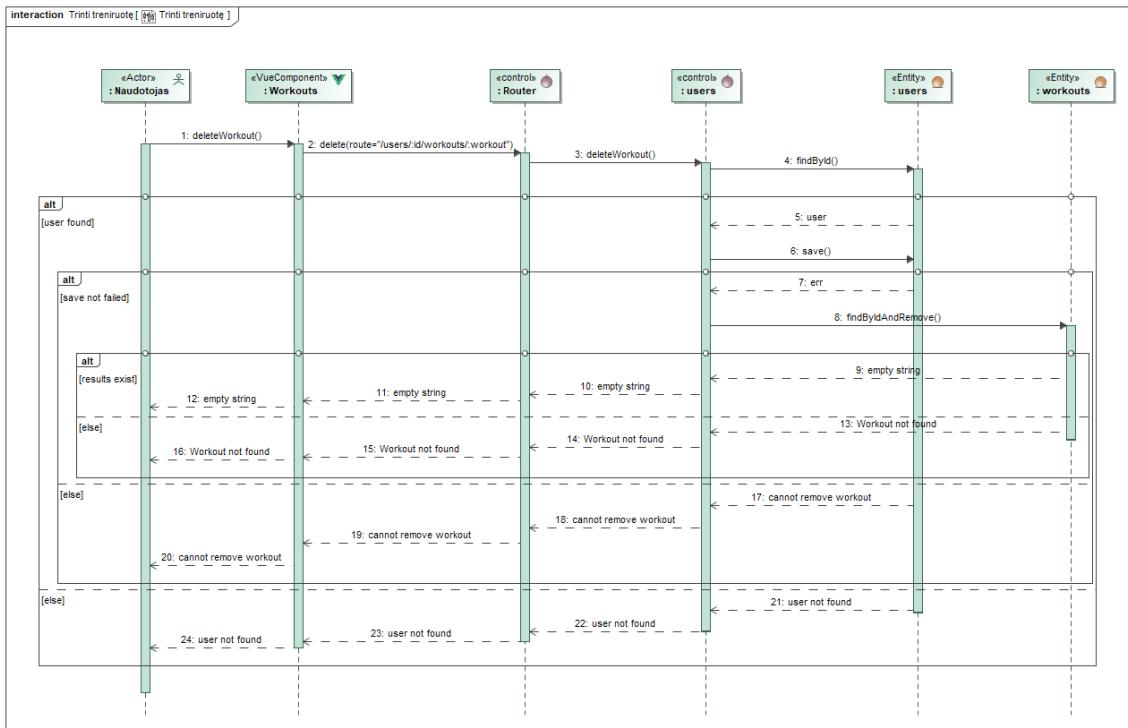
42 pav. Panaudojimo atvejo „Peržiūrėti treniruočių sąrašą“ sekų diagrama



43 pav. Panaudojimo atvejo „Kurti treniruotę“ sekų diagrama

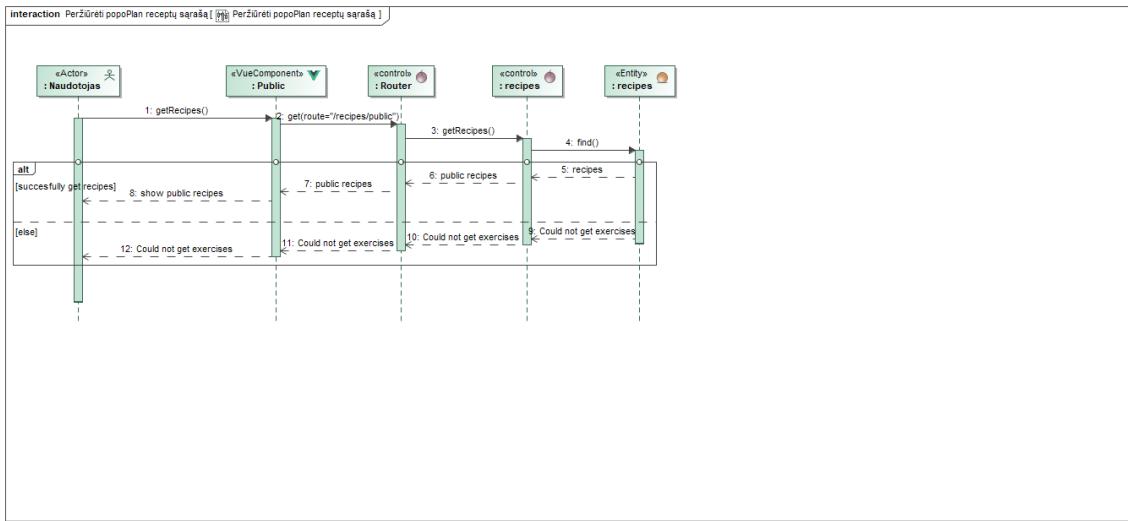


44 pav. Panaudojimo atvejo „Redaguoti treniruotę“ sekų diagrama



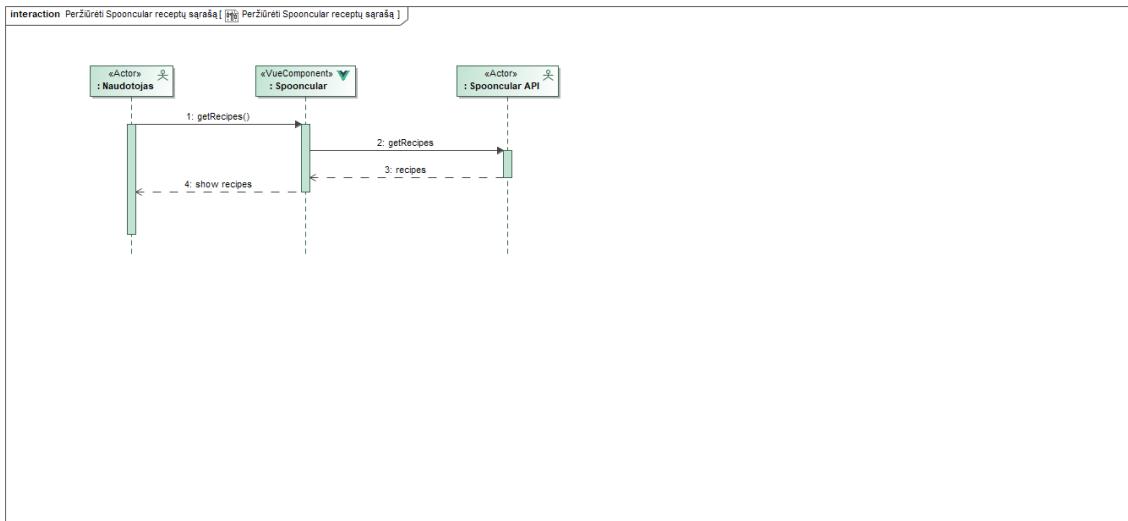
45 pav. Panaudojimo atvejo „Trinti treniruotę“ sekų diagrama

Panaudojimo atvejo „Peržiūrėti PopoPlan receptų sąrašą“ sekų diagramoje (??av.) rodoma, kad Public komponentas kreipiasi į Router komponentą, kuris kreipiasi į recipes. Recipes komponentas tada kreipiasi į users modelį, iš kurio gauna receptus ir yra grąžina naudotojui, jei nepavyksta, grąžinama klaida.

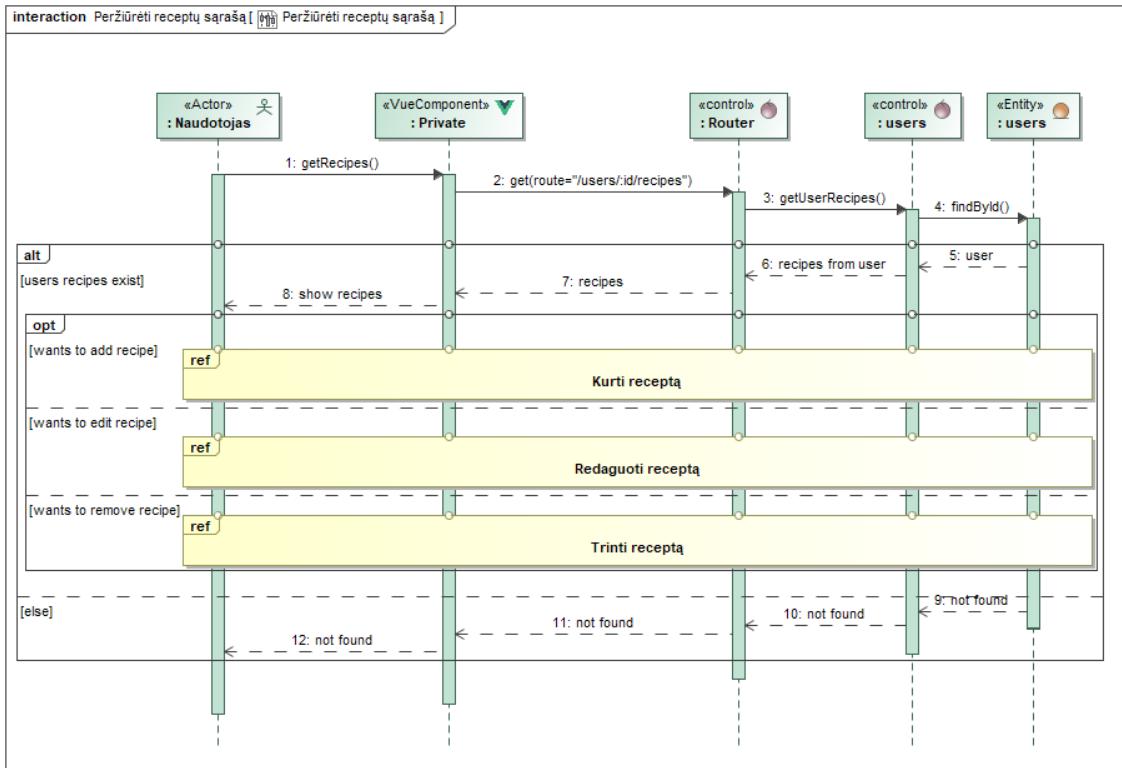


46 pav. Panaudojimo atvejo „Peržiūrėti PopoPlan receptų sąrašą“ sekų diagrama

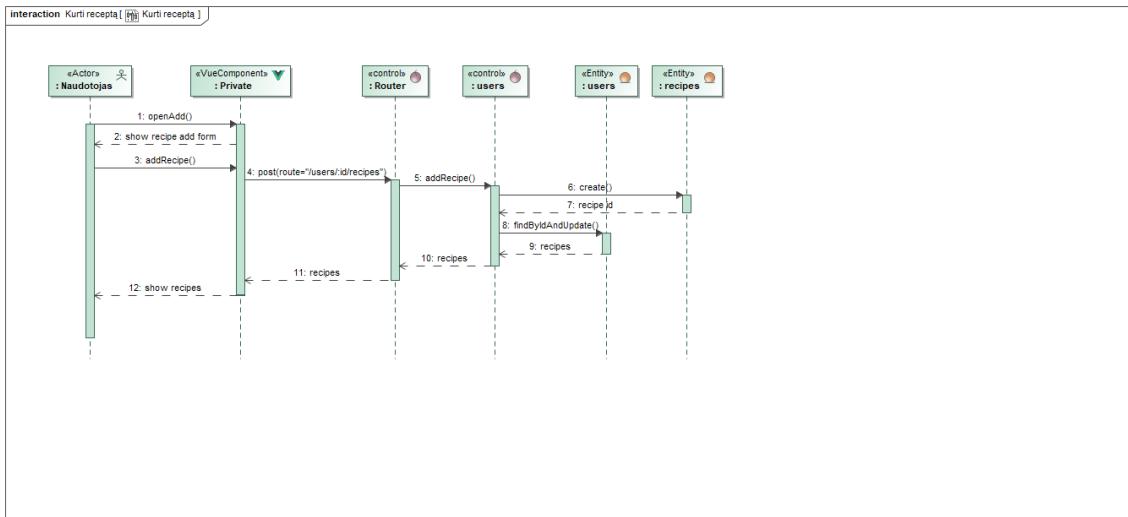
Panaudojimo atvejo „Peržiūrėti Spooncular receptų sąrašą” sekų diagramoje (47 pav.) rodoma, kad kreipiantis į Spooncular komponentą, šis kreipsis į Spooncular API. Tada iš spooncular API imami duomenys grįžta į Spooncular komponentą ir parodomai naudotojui



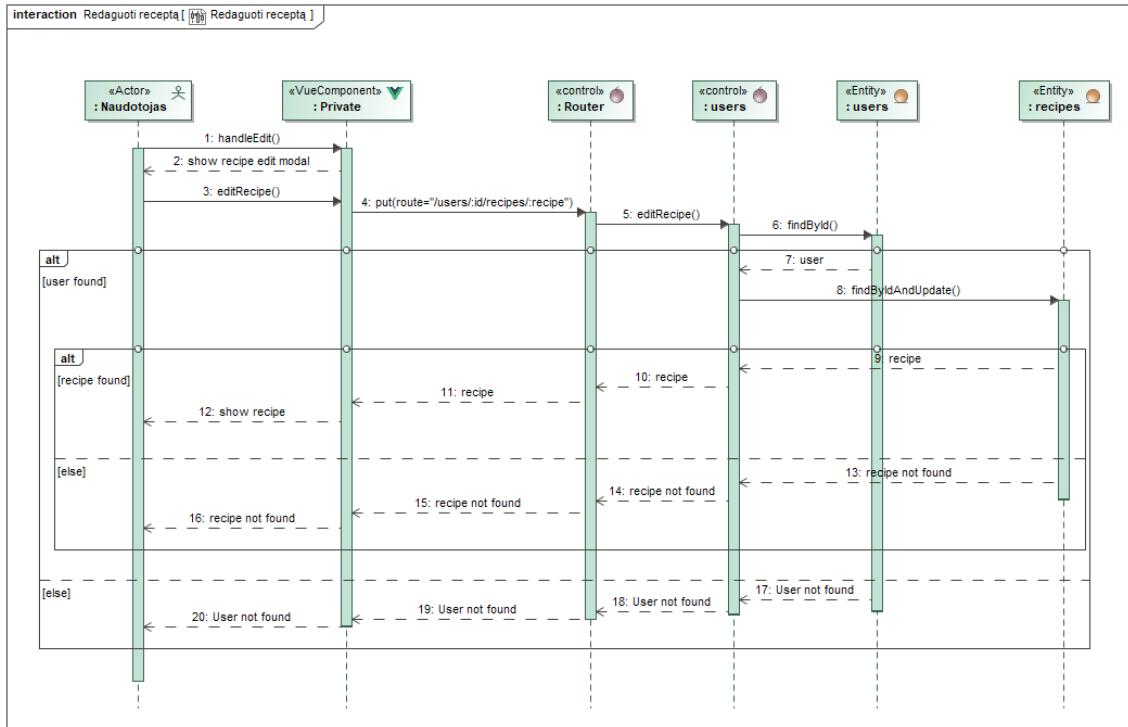
47 pav. Panaudojimo atvejo „Peržiūrėti Spooncular receptų sąrašą” sekų diagrama



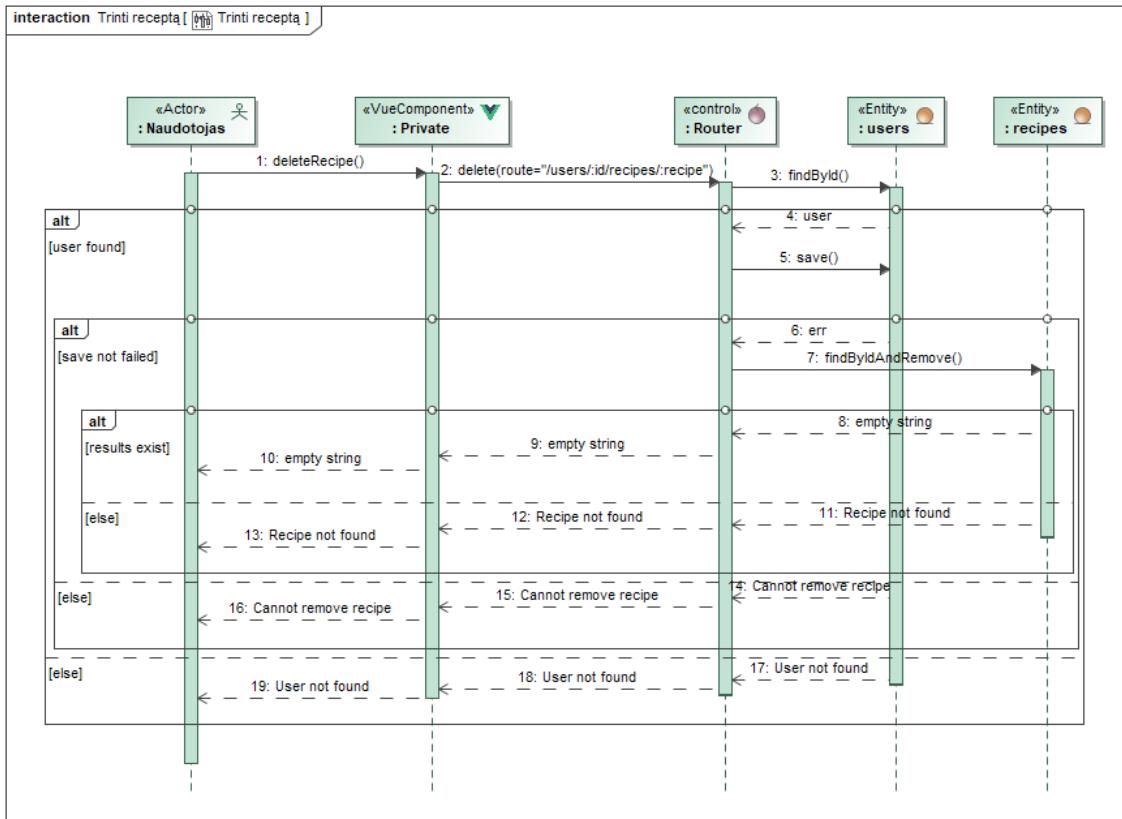
48 pav. Panaudojimo atvejo „Peržiūrėti receptų sąrašą“ sekų diagrama



49 pav. Panaudojimo atvejo „Kurti receptą“ sekų diagrama



50 pav. Panaudojimo atvejo „Redaguoti receptą“ sekų diagrama



51 pav. Panaudojimo atvejo „Trinti receptą” sekų diagrama

3. Testavimas

3.1. Testavimo planas

Testavimo planui vykdomui naudojamos priemonės:

1. Statinis kodo analizatorius
2. Automatinis ir rankinis naudotojo sasajos testavimas.

Statinis kodo analizatorius buvo naudojamas visose projekto dalyse, viso darbo metu. Jam atlikti buvo naudojama ESLint. Kliento daliai buvo iškelti reikalavimai: visi stringai rašomi viengubomis kabutėmis, nėra kabliataškių, kintamųjų ir funkcijų pavadinimai camelCase, gilėjant kodui - atitraukiama dvejais tarpais nuo kairės ir vue3 rekomenduojamos taisyklės. Serverio dalyje naudojami reikalavimai: kabutės - viengubos, kabliataškių nėra, nurodyti tipą priskiriant kintamajam nežinomą objektą - privaloma.

Automatiniu ir rankiniu būdu naudotojo sasaja buvo testuojama po sistemos užbaigimo.

3.2. Testavimo kriterijai

Kliento dalies programos kodo kriterijai:

1. Negali būti nereikalingi kintamieji;
2. Lyginant if'o sąlygoje, turi būti tikrinima ir reikšmė, ir tipas;
3. Stringai naudoja viengubas kabutes;
4. Gilėjant kodui - tabuliacija yra du tarpai nuo kairės;
5. Būtinai nenaudojami semicolon;
6. Kintamųjų ir funkcijų pavadinimai camelCase.

Serverio dalies programos kodo kriterijai:

1. Būtinai nenaudojami semicolon;
2. Kintamųjų ir funkcijų pavadinimai camelCase;
3. Lyginant if'o sąlygoje, turi būti tikrinima ir reikšmė, ir tipas;
4. Būtinai nurodomas tipas, jei jis nėra aiškus.

3.3. Naudotojo sasajos testavimas

3.3.1. Automatiniai testai

Naudotojo sasajos automatiniai testai buvo padaryti naudojant Selenium IDE. Padaryti 13 automatiniai testai (52 pav.), kurie išbando funkcionalumą.

Project: health-planner*

Tests +

Search tests...

1.Login*
2.Logout*
3.Register*
4.See Private recipe list*
5.See activity List*
6.See popoPlan exercise List
7.See popoPlan recipe list*
8.See wGer exercise List*
9.add activity*
10.add exercise*
11.add recipe*
12.see Private exercise list*
13.see Workout list*

Command Target Value

8 click css=div:nth-child(8) > input
9 click css=div:nth-child(6) > input
10 type css=div:nth-child(6) > input none
11 type css=div:nth-child(7) > input legs
12 click css=.modal-content > button:nth-child(4)
13 close

Command open
Target /
Value
Description

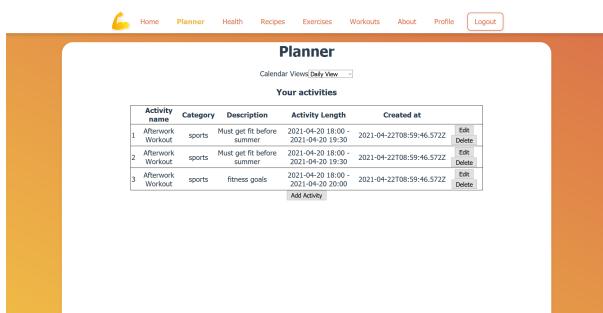
Log Reference

1. open OK 05:31:30
2. setWindowSize on 1549x933 OK 05:31:30
3. Trying to find linkText=My exercises... OK 05:31:30
4. close OK 05:31:36

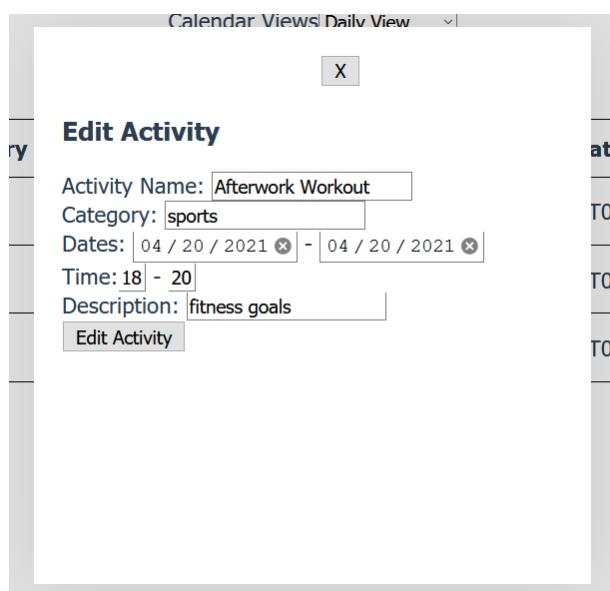
'see Private exercise list' completed successfully 05:31:36

52 pav. Naudotojo sasajos automatiniai testų rezultatai

3.3.2. Rankiniai testai



53 pav. Veiklos langas



54 pav. Veiklos redagavimo langas

4. Dokumentacija naudotojui

4.1. Apibendrintas sistemos galimybių aprašymas

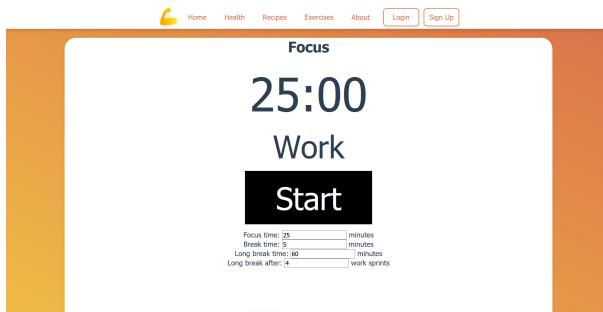
PopoPlan sistema skirstoma į kelias dalis: mobilioji programa, internetinė programa ir serveris. Paprasčiausias būdas išbandyti sistemą yra per naršykę. Visas funkcijas galima išbandyti per internetą, bet patogiau naudotojui gali būti atsiuntus mobiliają programą. Ji geriau pritaikyta, nes gali grįžti pranešimai, o ne spam'as į elektroninį paštą. Žmonės, norintys įdarbinti sistemos funkcijas savo projektui, gali pasinaudoti serveryje esančiu API.

4.2. Naudotojo vadovas

4.2.1. Internetinė programa



55 pav. Namų langas



56 pav. Focus langas



57 pav. PopoPlan pratimų langas

PopoPlan database recipes

Recipe Image	Name	Author	Creation date
	PEPPADEW® Prawn Salad	tautvis62	2021-04-20
	This recipe for easy beef enchiladas is perfect for meal planning or a quick-and-easy weeknight dinner.	tautvis62	2021-04-20
	This recipe for easy beef enchiladas is perfect for meal planning or a quick-and-easy weeknight dinner.	tautvis62	2021-04-22
	This recipe for easy beef enchiladas is perfect for meal planning or a quick-and-easy weeknight dinner.	tautvis62	2021-04-22

58 pav. PopoPlan receptų langas

Exercises

Exercise Image	Name	Author	Creation date
	Bent High Pulses	lakerbeezel	2014-09-01
	Bent High Pulses	lakerbeezel	2014-09-01
	Cable Woodchoppers	robhoyt	2013-06-23
	Cable Woodchoppers	robhoyt	2013-06-23
	Crunches	wiger.de	2013-05-05
	Crunches	wiger.de	2013-05-05
	Crunches With Cable	wiger.de	2013-05-05

59 pav. wGer pratimų langas

Recipes

Food Image	Name
	Cauliflower, Brown Rice, and Vegetable Fried Rice
	Homemade Garlic and Basil French Fries
	Berry Banana Breakfast Smoothie
	Garlicky Kale
	Chicken Tortilla Soup (Slow Cooker)
	African Chicken Peanut Stew
	Nigerian Smal Stew
	Red Kidney Bean Jambalaya

60 pav. Spooncular receptų langas

login

Username:
 Password:

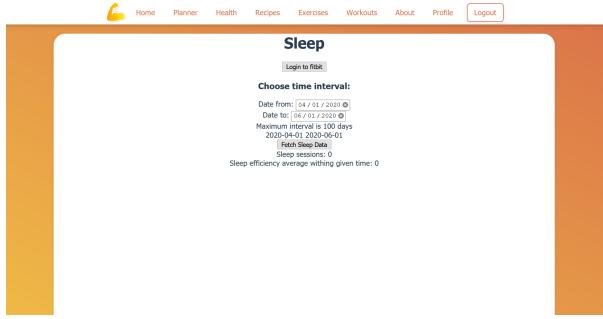
Remember me
[Forgot Password](#)

Login with

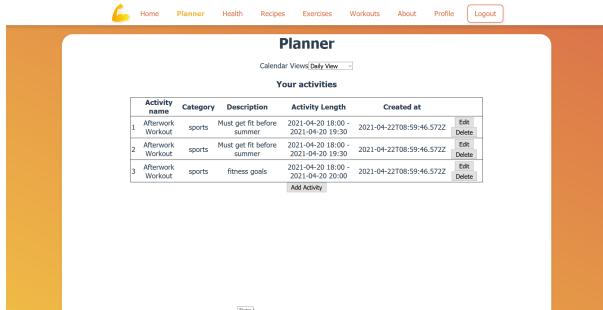
61 pav. Prisijungimo langas

Home **Planner** **Health** **Recipes** **Exercises** **Workouts** **About** **Profile**

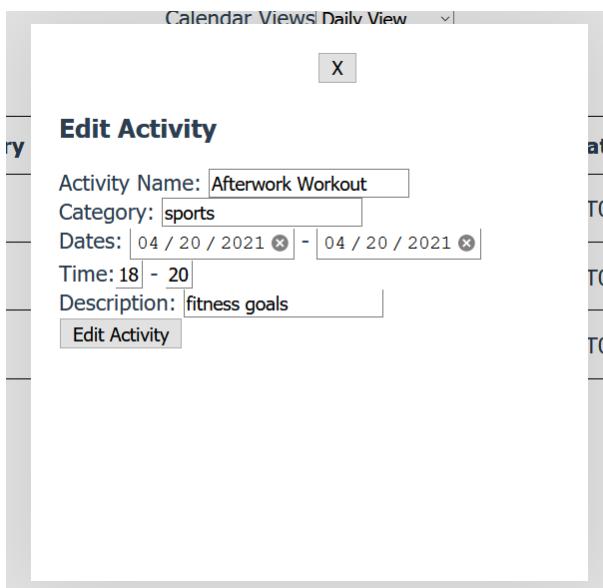
62 pav. Prisijungusio naudotojo navigacija



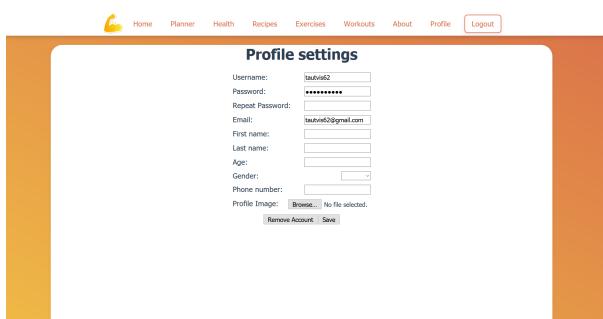
63 pav. Miego langas



64 pav. Veiklos langas

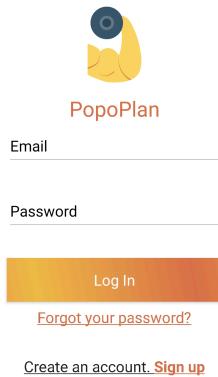


65 pav. Veiklos redagavimo langas

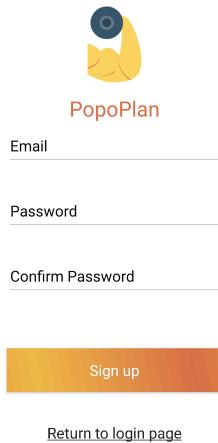


66 pav. Veiklos langas

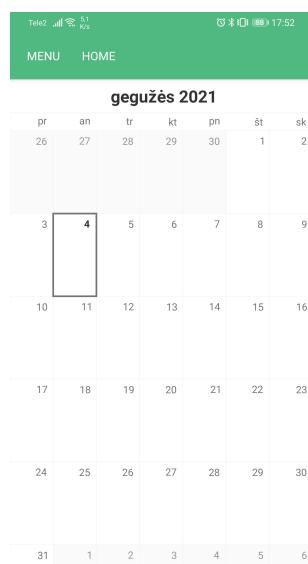
4.2.2. Mobili programa



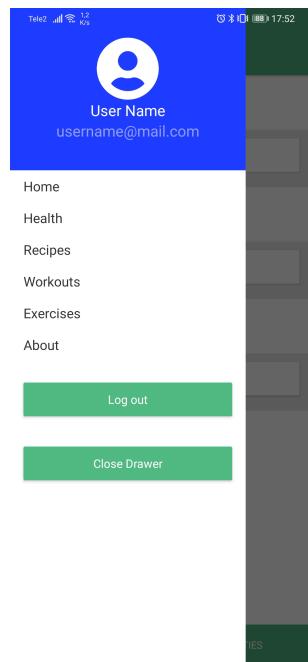
67 pav. Prisijungimo langas



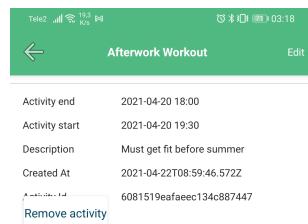
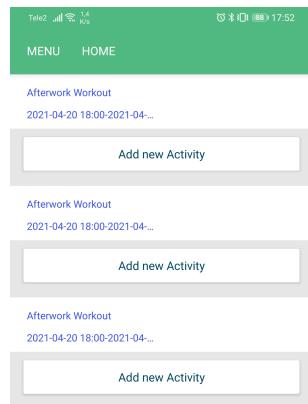
68 pav. Registracijos langas



69 pav. Namų langas



70 pav. Šoninio meniu navigacija



72 pav. Veiklos langas

The screenshot shows a mobile application interface for adding activity details. At the top, there is a green header bar with the text "Add newActivity Details" and buttons for "Cancel" and "Done". Below the header is a grey header bar with the text "Make Activity". The main area contains five input fields: "Name" (with a placeholder line), "Category" (with a placeholder line), "Image url" (with a placeholder line), "Event end" (with a placeholder line), and "description" (with a placeholder line). The background of the app shows a blurred view of a landscape.

73 pav. Veiklos pridėjimo langas

The screenshot shows a mobile application interface for editing activity details. At the top, there is a green header bar with the text "EditActivity Details" and buttons for "Cancel" and "Done". Below the header is a grey header bar with the text "Make Activity". The main area contains four input fields: "Afterwork Workout" (with a placeholder line), "sports" (with a placeholder line), "2021-04-20 18:00" (with a placeholder line), and "2021-04-20 19:30" (with a placeholder line). Below these fields is a note: "Must get fit before summer" (with a placeholder line). The background of the app shows a blurred view of a landscape.

74 pav. Veiklos redagavimo langas



Cauliflower, Brown Rice, an...



Homemade Garlic and Basi...



Berry Banana Breakfast Sm...



Garlicky Kale



POPOPLAN

SPOONULAR

MY RECIPES

75 pav. Spooncular receptai



Pomodoro timer

25:00

Button



76 pav. Pomodoro laikmadis

4.3. Diegimo vadovas

Visa PopoPlan sistema parašyta Javascript/TypeScript kalbomis. Siekiant pasileisti visas programas reikės tokios paketų tvarkyklės, kaip „NPM” arba „Yarn”. Projektas yra įkeltas į Github kodo talpyklą, todėl reikės arba atsiųsti kodą, arba daryti klonavimą su „git” versijavimo sistema. Turint kodą darbinėje aplinkoje, kiekvienam aplanke reikia paleisti „npm install” komandą. Klientinę sistemą galima pasileisti jos aplanke su komanda „npm run serve”. Kadangi klientinė ir mobilioji dalys ima duomenis iš serverio, reikėtų pirmiau pasileisti MongoDB ir serverio dalį. Serverio dalis paleidžiama su komanda „npm run dev”. Galiausiai siekiant prisijungti prie mobiliosios programos reikia prisijungti mobilujį telefoną prie kompiuterio, tada projektą paleisti su „tns run” komanda ir projektas bus įrašytas į prietaisą.

Siekiant, kad kodas veiktu, būtini šie etapai:

1. Kodo analizė su EsLint įrankiu - patikrinama, ar nėra sintaksės problemų.
2. Kodo paleidimas; mobiliosios progamos įdiegiamas (angl. *build*).
3. Kodas paleistas naudojimui.

Jei paleidus nurodytas komandas kode bus sintaksės problemų, komanda nustos veikti ir projektas nepasileis.

4.4. Administravimo vadovas

PopoPlan yra lokali sistema. Siekiant ja sėkmingai pasinaudoti pirmiausia reikia įdiegti „server” aplanke esančią programą - API. Taip pat svarbu turėti įjungtą MongoDB duomenų bazę. Po serverio paleidimo galima paleisti klietinę dalį ir duomenys bus gaunami iš serverio. Dirbant lokaliame tinkle, joks naudotojas negalės pasiekti tinklapio, nebent bus atidarytas IP adreso port’as.

Tai reiškia, jei dabartinė klientinė dalis netinka, naudotojos gali pasikurti šią projekto dalį . Mobilė programa dar nėra įkelta į Google Play Store, todėl reikia atsiųsti programą iš PopoPlan pagrindiniame puslapyje esančios nuorodos. Dirbant su API, kiekvienas pranešimas su statuso kodais yra aprašytas, todėl naudojimas turėtų būti paprastas.

Rezultatai ir išvados

1. Projekto tikslų laiku įvykdyti nepavyko, nes sistema nebuvvo išleista į Google App Store, iš kurios būsimi naudotojai būtų galėjė panaudoti programą ir pateikę darbo įvertinimą.
2. Atlikus konkurentų analizę, buvo pastebėta, kad programų funkcionalumas minimalus, bet bandoma įgauti naujų naudotojų intuityvia sasaja.
3. Sukurta sistema buvo norima padaryti įdomesnį planuoklį įtraukiant daugiau funkcijų, t.y. leisti iš tos pačios programos atlkti treniruotes, pasirinkti mitybą ir analizuoti žmogaus sveikatos rodmenis, taip suteikiant naudotojui savikontrolę, bet visiškai realizuoti nepavyko.
4. Dabartinis projektas yra informacinė sistema, kuri leidžia konfigūruoti duomenis su: veikla, treniruotėmis, receptais ir sporto pratimais.
5. Projekto atlikimo metu buvo taikomos technologijos, su kuriomis dirbant nebuvvo patirties. Kilo daug nesklandumų. Nativescript atviro kodo karkasas yra geras įrankis siekiant sukurti sistemą Android ir iOS operacinėms sistemoms, bet dėl paviršutiniško dokumentavimo, pasenusių kodo pavyzdžių ir pan. buvo sudėtinga greitai išmokti naudotis technologija.
6. Visiškai realizuoti PopoPlan sistemą nepavyko dėl per mažo duoto laiko, t.y. 300 valandų. Atliekant projektą keliose platformose, bandant realizuoti tas pačias funkcijas, darbo dydis išaugo du kartus. Neturint žinių dirbant prie mobiliosios programos, buvo sugaišta nemažai laiko mokymuisi ir klaidų taisymui.
7. Kuriama sistema atlakta su JavaScript ir TypeScript programavimo kalbomis, Vue, NativeScript Vue ir Express.js karkasais. TypeScript ir NativeScript Vue technologijos prieš baigiamajį darbą nebuvvo anksčiau naudotos.
8. Sistema galima naudotis mobiliuoju telefonu ir PC platformoje.
9. Siekiant padaryti projektą svaresnį, reikėtų įvykdyti visišką funkcionalumą su Fitbit ir kitais išmaniaisiais laikrodžiais, kurie seka naudotojo sveikatą.