



**Kauno technologijos universitetas**

Informatikos fakultetas

**Moderni grafinė naudotojo sąsaja daiktų interneto  
maršrutizatoriui**

Baigiamasis magistro darbas

---

**Eimantas Žvirblys**

Projekto autorius

**Lekt. Linas Ablonskis**

Vadovas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Moderni grafinė naudotojo sąsaja daiktų interneto maršrutizatoriui**

Baigiamasis magistro projektas

Informacinių sistemų inžinerija (6211BX009)

---

**Eimantas Žvirblys**

Projekto autorius

**Lekt. Linas Ablonskis**

Vadovas

**Prof. Audrius Lopata**

Recenzentas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Informatikos Fakultetas

Eimantas Žvirblys

## **Moderni grafinė naudotojo sąsaja daiktų interneto maršrutizatoriui**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Eimantas Žvirblys

*Patvirtinta elektroniniu būdu*

Eimantas Žvirblys. Moderni grafinė naudotojo sąsaja daiktų interneto maršrutizatoriui. Magistro baigiamasis projektas / vadovas lekt. Linas Ablonskis; Kauno technologijos universitetas, Informatikos fakultetas.

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

Reikšminiai žodžiai: grafinė sąsaja, vieno puslapio programos modelis, internetinių puslapių užkrovimo greitimeika.

Kaunas, 2022. 119 p.

### **Santrauka**

Šio darbo tikslas – sumažinti daiktų interneto maršrutizatoriaus informacinės sistemos puslapių užkrovimo laiką ir naudojamų resursų kiekį.

Tiriamajame darbe atliktos internetinių puslapių kūrimo modelių ir technologijų alternatyvų analizės. Atliktas palyginimas tarp tradicinio internetinių puslapių modelio, kuriuo remiantis yra sukurta esama daiktų interneto maršrutizatoriaus grafinė naudotojo sąsaja ir vieno puslapio programos modelio. Atliktas palyginimas tarp vieno puslapio programos modelio technologijų: Angular, Vue, React. Aprašytas formalizuotas sprendimo ir dalykinės srities aprašas. Iškelti funkciniai ir nefunkciniai reikalavimai daiktų interneto grafinė naudotojo sąsajai. Realizacijos metu aprašyta grafinės naudotojo sąsajos architektūra, sekų diagramos ir diegimo diagrama. Eksperimento atlikimui, buvo iškelti funkciniai reikalavimai automatizuotai grafinės naudotojo sąsajos testavimo programai, kuri testuoja puslapių užkrovimo greitimeiką ir naudojamų resursų kiekį. Realizuotos dvi grafinės naudotojo sąsajos panaudojant Vue ir React technologijas ir automatizuota testavimo programa pagal iškeltus reikalavimus. Eksperimento metu gauti greitimeikos ir resursų panaudojimo rezultatai išanalizuoti ir pateiktos tiriamojo darbo išvados.

Eimantas Žvirblys. Modern graphical user interface for internet of things router. Master's Final Degree Project, supervisor lect. Linas Ablonskis; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information Systems, Computing.

Keywords: graphical user interface, single page application model, web page load time analysis.

Kaunas, 2022. 119 p.

### **Summary**

The aim of this work is to reduce the loading time of the IoT router information system pages and the number of resources used.

This paper analyzes website development models and technology alternatives. A comparison between the traditional web page model, which is built on the existing graphical user interface of the IoT router, and the single page application model. A comparison between single page application model technologies: Angular, Vue, React. A formalized description and subject area of the solution. Functional and non-functional requirements for the graphical user interface of the IoT router. The graphical user interface architecture, sequence diagrams, and deployment diagram for implementation. Functional requirements for an automated graphical user interface testing program that tests the speed of page loading and the number of resources used. Implementation of two graphical user interfaces using Vue and React technologies and an automated testing program according to the requirements. An experiment where the graphical user interface was tested. Conclusions according to experiment results.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>8</b>
<b>Paveikslų sąrašas .....</b>	<b>10</b>
<b>Santrumpų ir terminų sąrašas .....</b>	<b>15</b>
<b>Įvadas.....</b>	<b>16</b>
<b>1. Probleminės srities analizė.....</b>	<b>18</b>
1.1. Analizės tikslas.....	18
1.2. Internetinių puslapių kūrimo modelių analizė .....	18
1.2.1. Tradicinių internetinių puslapių modelio taikymas.....	18
1.2.2. Tradicinių internetinių puslapių modelio taikymas IoT maršrutizatoriuose .....	18
1.2.3. OpenWrt įterptinių sistemų įrenginiuose .....	20
1.2.4. Vieno puslapio programų modelio taikymas .....	21
1.2.5. Vieno puslapio programų pranašumai ir trūkumai.....	23
1.2.6. Vieno puslapių programos karkasas „Angular“ .....	25
1.2.7. Vieno puslapio programų biblioteka „React“ .....	25
1.2.8. Vieno puslapio programų biblioteka „Vue“ .....	26
1.2.9. Internetinių puslapių užkrovimo pagreitinimo gerosios praktikos.....	26
1.2.10. Internetinių puslapių technologijų kūrimo palyginimas.....	27
1.3. Analizės išvados.....	29
<b>2. Grafinės sąsajos optimizavimo metodų daiktų interneto įrenginiuose sprendimo reikalavimų specifikacija ir projektas, formalus aprašas.....</b>	<b>30</b>
2.1. Funkcinių reikalavimų specifikacija .....	30
2.2. Nefuncinių reikalavimų specifikacija .....	50
2.3. Dalykinės srities modelis .....	51
2.4. Naudotojų sąsajos modelis.....	52
2.5. Formalizuotas sprendimo aprašas .....	56
2.6. Reikalavimų apibendrinimas ir išvados .....	58
<b>3. Grafinės sąsajos optimizavimo metodų daiktų interneto įrenginiuose sprendimo realizacijos projektas.....</b>	<b>59</b>
3.1. Daiktų interneto maršrutizatoriaus sistemos loginė architektūra .....	59
3.2. Sistemos elgsenos modelis.....	59
3.3. Realizacijos modelis.....	73
3.4. Konfigūracijos failų modelis.....	73
<b>4. Sprendimo realizacija ir testavimas .....</b>	<b>75</b>
4.1. Sprendimo realizacijos ir veikimo aprašas.....	75
4.1.1. Vieno puslapio programos modelio sprendimo realizacija ir veikimo aprašas.....	75
4.1.2. Automatizuotos testavimo programos sprendimo realizacijos ir veikimo aprašas .....	76
4.2. Testavimo modelis, duomenys, rezultatai .....	76
4.2.1. Vieno puslapio programos modelio funkcinių reikalavimų testavimo duomenys .....	76
4.2.2. Vieno puslapio programos modelio nefuncinių reikalavimų testavimo duomenys .....	80
4.2.3. Automatizuotos testavimo programos funkcinių reikalavimų testavimo duomenys .....	81
<b>5. Eksperimentinis grafinių sąsajų greیتaveikos tyrimas .....</b>	<b>83</b>
5.1. Eksperimento planas .....	83
5.2. Eksperimento rezultatai.....	86
5.2.1. Puslapių užkrovimo ir duomenų išsaugojimo laikai .....	86

5.2.2. Procesoriaus apkrova ir atminties poreikis.....	114
5.3. Eksperimento apibendrinimai ir išvados.....	116
<b>Išvados .....</b>	<b>117</b>
<b>Literatūros sąrašas .....</b>	<b>118</b>

## Lentelių sąrašas

<b>1 lentelė.</b> M. Levlin testavimo greitaveikos palyginimas .....	27
<b>2 lentelė.</b> Vieno puslapio programų karkasų ir bibliotekų palyginimas.....	28
<b>3 lentelė.</b> PA 1 „Prisijungti“ specifikacija.....	31
<b>4 lentelė.</b> PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ specifikacija .....	32
<b>5 lentelė.</b> PA 3 „Peržiūrėti vidinio tinklo informaciją“ specifikacija.....	33
<b>6 lentelė.</b> PA 4 „Valdyti vidinio tinklo nustatymus“ specifikacija .....	34
<b>7 lentelė.</b> PA 5 „Peržiūrėti belaidžio ryšio informaciją“ specifikacija.....	35
<b>8 lentelė.</b> PA 6 „Valdyti belaidžio ryšio nustatymus“ specifikacija .....	36
<b>9 lentelė.</b> PA 7 „Peržiūrėti mobilaus ryšio informaciją“ specifikacija.....	37
<b>10 lentelė.</b> PA 8 „Valdyti mobilaus ryšio nustatymus“ specifikacija .....	38
<b>11 lentelė.</b> PA 9 „Peržiūrėti sistemos laiką“ specifikacija .....	39
<b>12 lentelė.</b> PA 10 „Valdyti sistemos laiko nustatymus“ specifikacija .....	40
<b>13 lentelė.</b> PA 11 „Valdyti ugniasienės nustatymus“ specifikacija .....	41
<b>14 lentelė.</b> PA 12 „Valdyti IPSec tunelio nustatymus“ specifikacija.....	42
<b>15 lentelė.</b> PA 13 „Valdyti įvesties/išvesties“ specifikacija.....	43
<b>16 lentelė.</b> PA 14 „Keisti slaptažodį“ specifikacija .....	44
<b>17 lentelė.</b> PA 15 „Valdyti MODBUS nustatymus“ specifikacija.....	45
<b>18 lentelė.</b> PA 16 „Atsijungti“ specifikacija .....	46
<b>19 lentelė.</b> PA 17 „Įvykdyti testą“ specifikacija .....	47
<b>20 lentelė.</b> PA 18 „Parinkti testuojamą technologiją“ specifikacija .....	48
<b>21 lentelė.</b> PA 19 „Generuoti testavimo ataskaitą“ specifikacija.....	49
<b>22 lentelė.</b> PA 1 „Prisijungti“ testavimo duomenys.....	76
<b>23 lentelė.</b> PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ testavimo duomenys .....	77
<b>24 lentelė.</b> PA 3 „Peržiūrėti vidinio tinklo būseną“ testavimo duomenys .....	77
<b>25 lentelė.</b> PA 4 „Valdyti vidinio tinklo nustatymus“ testavimo duomenys.....	77
<b>26 lentelė.</b> PA 5 „Peržiūrėti belaidžio ryšio būseną“ testavimo duomenys .....	77
<b>27 lentelė.</b> PA 6 „Valdyti belaidžio ryšio nustatymus“ testavimo duomenys.....	78
<b>28 lentelė.</b> PA 7 „Peržiūrėti mobilaus ryšio būseną“ testavimo duomenys .....	78
<b>29 lentelė.</b> PA 8 „Valdyti mobilaus ryšio nustatymus“ testavimo duomenys.....	78
<b>30 lentelė.</b> PA 9 „Peržiūrėti sistemos laiką“ testavimo duomenys .....	78
<b>31 lentelė.</b> PA 10 „Valdyti sistemos laiko nustatymus“ testavimo duomenys .....	79
<b>32 lentelė.</b> PA 11 „Valdyti ugniasienės nustatymus“ testavimo duomenys.....	79
<b>33 lentelė.</b> PA 12 „Valdyti IPSec tunelio nustatymus“ testavimo duomenys.....	79
<b>34 lentelė.</b> PA 13 „Valdyti įvesties/išvesties nustatymus“ testavimo duomenys.....	79
<b>35 lentelė.</b> PA 14 „Keisti slaptažodį“ testavimo duomenys.....	80
<b>36 lentelė.</b> PA 15 „Valdyti MODBUS nustatymus“ testavimo duomenys .....	80
<b>37 lentelė.</b> PA 16 „Atsijungti“ testavimo duomenys .....	80



<b>38 lentelė.</b> Vieno puslapio programos modelio nefunkcinių reikalavimų testavimo duomenys .....	80
<b>39 lentelė.</b> PA 17 „Įvykdyti testą“ testavimo duomenys.....	81
<b>40 lentelė.</b> PA 18 Parinkti testuojamą technologiją .....	81
<b>41 lentelė.</b> PA 19 „Generuoti testavimo ataskaitą“ testavimo duomenys.....	82
<b>44 lentelė.</b> Grafinės sąsajos testavimo scenarijai .....	84
<b>45 lentelė.</b> Eksperimento metu gautų rezultatų santrauka.....	116

## Paveikslų sąrašas

<b>1 pav.</b> Tradicinių internetinių puslapių sąsajos modelis .....	18
<b>2 pav.</b> AJAX technologijos veikimas internetinių puslapių programose .....	19
<b>3 pav.</b> Vieno puslapio programos veikimo modelis .....	21
<b>4 pav.</b> Dokumento objekto modelis .....	22
<b>5 pav.</b> Pagrindinė JSON struktūra .....	23
<b>7 pav.</b> IoT maršrutizatoriaus informacinės sistemos panaudojimo atvejų diagrama .....	30
<b>8 pav.</b> Automatizuota puslapių užkrovimo programos panaudos atvejų diagrama .....	31
<b>9 pav.</b> PA 1 „Prisijungti“ veiklos diagrama .....	32
<b>10 pav.</b> PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ veiklos diagrama .....	33
<b>11 pav.</b> PA 3 „Peržiūrėti vidinio tinklo informaciją“ veiklos diagrama .....	34
<b>12 pav.</b> PA 4 „Valdyti vidinio tinklo nustatymus“ veiklos diagrama .....	35
<b>13 pav.</b> PA 5 „Peržiūrėti belaidžio ryšio nustatymus“ veiklos diagrama .....	36
<b>14 pav.</b> PA 6 „Valdyti belaidžio ryšio nustatymus“ veiklos diagrama .....	37
<b>15 pav.</b> PA 7 „Peržiūrėti mobilaus ryšio būseną“ veiklos diagrama .....	38
<b>16 pav.</b> PA 8 „Valdyti mobilaus ryšio nustatymus“ veiklos diagrama .....	39
<b>17 pav.</b> PA 9 „Peržiūrėti sistemos laiką“ veiklos diagrama .....	40
<b>18 pav.</b> PA 10 „Valdyti sistemos laiko nustatymus“ veiklos diagrama .....	41
<b>19 pav.</b> PA 11 „Valdyti ugniasienės nustatymus“ veiklos diagrama .....	42
<b>20 pav.</b> PA 12 „Valdyti IPsec tunelio nustatymus“ veiklos diagrama .....	43
<b>21 pav.</b> PA 13 „Peržiūrėti įvesties/išvesties būsenas“ veiklos diagrama .....	44
<b>22 pav.</b> PA 14 „Keisti slaptažodį“ veiklos diagrama .....	45
<b>23 pav.</b> PA 15 „Valdyti MODBUS nustatymus“ veiklos diagrama .....	46
<b>24 pav.</b> PA 16 „Atsijungti“ veiklos diagrama .....	47
<b>24 pav.</b> PA 17 „Įvykdyti testą“ veiklos diagrama .....	48
<b>25 pav.</b> PA 18 „Parinkti testuojamą technologiją“ veiklos diagrama .....	49
<b>26 pav.</b> PA 19 „Generuoti testavimo ataskaitą“ veiklos diagrama .....	50
<b>27 pav.</b> Daiktų interneto maršrutizatoriaus informacinės sistemos esybių ryšių modelis .....	51
<b>28 pav.</b> Automatizuotos puslapių užkrovimo testavimo programos esybių ryšiu modelis .....	51
<b>29 pav.</b> Daiktų interneto maršrutizatoriaus informacinės sistemos pagrindinis puslapis .....	52
<b>30 pav.</b> Daiktų interneto maršrutizatoriaus prisijungimo puslapis .....	52
<b>31 pav.</b> Informacinės sistemos ugniasienės nustatymų puslapis „Firewall“ .....	53
<b>32 pav.</b> Informacinės sistemos belaidžio ryšio nustatymų puslapis „Wireless“ .....	53
<b>33 pav.</b> IoT maršrutizatoriaus informacinės sistemos vidinio tinklo nustatymų puslapis „LAN“ ...	54
<b>34 pav.</b> IoT maršrutizatoriaus informacinės sistemos laiko nustatymų puslapis „NTP“ .....	54
<b>35 pav.</b> IoT maršrutizatoriaus informacinės sistemos virtualaus tinklo nustatymų puslapis „IPsec“ .....	55
<b>36 pav.</b> IoT maršrutizatoriaus informacinės įvesties/išvesties nustatymų puslapis „Input/Output“ ..	55

<b>37 pav.</b> IoT maršrutizatoriaus informacinės sistemos nustatymų puslapis „MODBUS“ .....	56
<b>38 pav.</b> IoT maršrutizatoriaus pagrindinių nustatymų puslapis „Administration“ .....	56
<b>39 pav.</b> Tradicinio ir vieno puslapio programos modelių veikimo skirtumai .....	57
<b>40 pav.</b> Aritmetinio vidurkio matematinė formulė .....	58
<b>41 pav.</b> Medianos matematinė formulė .....	58
<b>42 pav.</b> Daiktų interneto maršrutizatoriaus loginės architektūros diagrama.....	59
<b>43 pav.</b> PA 1 „Prisijungti“ loginės architektūros diagrama .....	60
<b>44 pav.</b> PA 1 „Prisijungti“ sekų diagrama .....	60
<b>45 pav.</b> PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ loginės architektūros diagrama .....	61
<b>46 pav.</b> PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ .....	61
<b>47 pav.</b> PA 3 „Peržiūrėti vidinio tinklo būseną“ loginės architektūros diagrama.....	61
<b>48 pav.</b> PA 3 „Peržiūrėti vidinio tinklo būseną“ sekų diagrama.....	62
<b>49 pav.</b> PA 4 „Valdyti vidinio tinklo nustatymus“ loginės architektūros diagrama .....	62
<b>50 pav.</b> PA 4 „Valdyti vidinio tinklo nustatymus“ sekų diagrama .....	63
<b>51 pav.</b> PA 5 „Peržiūrėti belaidžio ryšio būseną“ loginės realizacijos klasių diagrama.....	63
<b>52 pav.</b> PA 5 „Peržiūrėti belaidžio ryšio informaciją“ loginės realizacijos sekų diagrama.....	64
<b>53 pav.</b> PA 6 „Valdyti belaidžio ryšio informaciją“ loginė architektūros diagrama .....	64
<b>54 pav.</b> PA 6 „Valdyti belaidžio ryšio nustatymus“ sekų diagrama .....	65
<b>55 pav.</b> PA 7 „Peržiūrėti mobilaus ryšio būseną“ loginė architektūros diagrama .....	65
<b>56 pav.</b> PA 7 „Peržiūrėti mobilaus ryšio būseną“ sekų diagrama.....	66
<b>57 pav.</b> PA 8 „Valdyti mobilaus ryšio nustatymus“ loginė architektūros diagrama.....	66
<b>58 pav.</b> PA 8 „Valdyti mobilaus ryšio nustatymus“ sekų diagrama .....	67
<b>59 pav.</b> PA 9 „Peržiūrėti sistemos laiką“ loginė architektūros diagrama .....	67
<b>60 pav.</b> PA 9 „Peržiūrėti sistemos laiką“ sekų diagrama .....	67
<b>61 pav.</b> PA 10 „Valdyti sistemos laiką“ loginė architektūros diagrama .....	68
<b>62 pav.</b> PA 10 „Valdyti sistemos laiką“ sekų diagrama .....	68
<b>63 pav.</b> PA 11 „Valdyti ugniasienės nustatymus“ loginė architektūra .....	68
<b>64 pav.</b> PA 11 „Valdyti ugniasienės nustatymus“ sekų diagrama .....	69
<b>65 pav.</b> PA 12 „Valdyti IPSec tunelio nustatymus“ loginė architektūros diagrama.....	69
<b>66 pav.</b> PA 12 „Valdyti IPSec tunelio nustatymus“ sekų diagrama .....	70
<b>67 pav.</b> PA 13 „Valdyti Įvesties/Išvesties nustatymus“ loginė architektūros diagrama .....	70
<b>68 pav.</b> PA 14 „Valdyti Įvesties/Išvesties nustatymus“ sekų diagrama.....	71
<b>69 pav.</b> PA 14 „Keisti slaptažodį“ loginė architektūros diagrama.....	71
<b>70 pav.</b> PA 14 „Keisti slaptažodį“ sekų diagrama .....	71
<b>71 pav.</b> PA 15 „Valdyti MODBUS nustatymus“ loginė architektūros diagrama .....	72
<b>72 pav.</b> PA 15 „Valdyti MODBUS nustatymus“ sekų diagrama.....	72
<b>73 pav.</b> PA 16 „Atsijungti“ loginė architektūros diagrama.....	72
<b>74 pav.</b> PA 16 „Atsijungti“ sekų diagrama .....	73

<b>75 pav.</b> Daiktų interneto maršrutizatoriaus informacinės sistemos realizacijos modelis.....	73
<b>76 pav.</b> Daiktų interneto maršrutizatoriaus informacinės sistemos konfigūracijos failų modelis ....	74
<b>77 pav.</b> Prisijungimo prie informacinės sistemos testavimo vidurkių diagrama .....	87
<b>78 pav.</b> „Prisijungimo“ testavimo scenarijaus įverčių pasiskirstymo diagrama .....	87
<b>79 pav.</b> Vieno puslapio programų „Prisijungimo“ testavimo scenarijaus įverčių pasiskirstymo diagrama .....	88
<b>80 pav.</b> „Mobile“ puslapio užkrovimo testavimo vidurkių diagrama .....	88
<b>81 pav.</b> „Mobile“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	89
<b>82 pav.</b> Vieno puslapio programų „Mobile“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	89
<b>83 pav.</b> „Mobile“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus vidurkių diagrama ..	90
<b>84 pav.</b> „Mobile“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	91
<b>85 pav.</b> Vieno puslapio programų „Mobile“ konfigūracijos testavimo scenarijaus įverčių pasiskirstymo diagrama .....	91
<b>86 pav.</b> „LAN“ puslapio užkrovimo testavimo vidurkio diagrama .....	92
<b>87 pav.</b> „LAN“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama.....	92
<b>88 pav.</b> Vieno puslapio programų „LAN“ užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	93
<b>89 pav.</b> „LAN“ puslapio konfigūracijos išsaugojimo testavimo vidurkio diagrama .....	93
<b>90 pav.</b> „LAN“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	94
<b>91 pav.</b> „Wireless“ puslapio užrovimo testavimo rezultatai .....	94
<b>92 pav.</b> „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	95
<b>93 pav.</b> Vieno puslapio programų „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	95
<b>94 pav.</b> „Wireless“ puslapio konfigūracijos išsaugojimo testavimo rezultatų diagrama .....	96
<b>95 pav.</b> „Wireless“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	96
<b>96 pav.</b> „Firewall“ puslapio užkrovimo testavimo rezultatai.....	97
<b>97 pav.</b> „Firewall“ puslapio užkrovimo testavimo įverčių pasiskirstymo rezultatų diagrama .....	97
<b>98 pav.</b> Vieno puslapio programos modelių „Firewall“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	98
<b>99 pav.</b> „Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus vidurkio diagrama	98
<b>100 pav.</b> „Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	99
<b>101 pav.</b> „NTP“ puslapio užkrovimo testavimo rezultatai.....	99
<b>102 pav.</b> „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	100
<b>103 pav.</b> Vieno puslapio programų „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	100

<b>104 pav.</b> „NTP“ puslapio konfigūracijos išsaugojimo testavimo rezultatai .....	101
<b>105 pav.</b> „NTP“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	101
<b>106 pav.</b> „IPSec“ puslapio užkrovimo testavimo rezultatai.....	102
<b>107 pav.</b> „IPSec“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama.....	102
<b>108 pav.</b> Vieno puslapio programų „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	103
<b>109 pav.</b> „IPSec“ puslapio konfigūracijos išsaugojimo testavimo rezultatai.....	103
<b>110 pav.</b> „IPSec“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	104
<b>111 pav.</b> „Modbus“ puslapio užkrovimo testavimo scenarijaus rezultatų diagrama .....	104
<b>112 pav.</b> „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama ..	105
<b>113 pav.</b> Vieno puslapio programos „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	105
<b>114 pav.</b> „Modbus“ puslapio nustatymų išsaugojimo testavimo diagrama .....	106
<b>115 pav.</b> „Modbus“ konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	106
<b>116 pav.</b> „Input/Output“ puslapio užkrovimo testavimo rezultatai .....	107
<b>117 pav.</b> „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	107
<b>118 pav.</b> Vieno puslapio programos modelio „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	108
<b>119 pav.</b> „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo rezultatai .....	108
<b>120 pav.</b> „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	109
<b>121 pav.</b> „Administration“ puslapio užkrovimo testavimo duomenys .....	109
<b>122 pav.</b> „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	110
<b>123 pav.</b> Vieno puslapio programų „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama.....	110
<b>124 pav.</b> „Administration“ puslapio konfigūracijos išsaugojimo testavimo rezultatai.....	111
<b>125 pav.</b> „Administration“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama .....	111
<b>126 pav.</b> Visų testavimo scenarijų bendro laiko vidurkio testavimo rezultatai .....	112
<b>127 pav.</b> Viso testavimo bendro laiko įverčių pasiskirstymo diagrama.....	112
<b>128 pav.</b> Visų puslapių užkrovimo greیتaveikos vidurkio rezultatų diagrama.....	113
<b>129 pav.</b> Konfigūracijos išsaugojimo greیتaveikos vidurkio testavimo rezultatai .....	113
<b>130 pav.</b> Vidutinė procesoriaus apkrovos testavimo metu diagrama.....	114
<b>131 pav.</b> Procesoriaus apkrova testavimo metu .....	114
<b>132 pav.</b> Operatyviosios atminties panaudojimo vidurkio diagrama.....	115

**133 pav.** Technologijų pastoviosios atminties diagrama ..... 115

## Santrumpų ir terminų sąrašas

### Santrumpos:

**IoT** – daiktų internetas (angl. Internet of Things);

**M2M** – mašina tarp mašinų (angl. Machine to machine);

**HTML** – kompiuterinė žymėjimo kalba (angl. Hyper text Markup Language“), naudojama pateikti turinį internete.

**CSS** – kalba skirta aprašyti internetinių puslapių elementų stilių struktūrizuotomis taisyklėmis. (angl. Cascading Style Sheets).

**DOM** – dokumento objekto modelis (angl. Document Object Model).

**MVC** – programinės įrangos projektavimo modelis (angl. Model – View – Controller).

**JSON** – duomenų perdavimo formatas (angl. JavaScript Object Notation).

**XML** – bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba (angl. Extensible Markup Language).

### Terminai:

**Web** – pasaulinis tinklas (angl. World Wide Web arba WWW). Interneto dalis, ištekliai, kuriuos internete galima pasiekti.

**Vue**– vieno puslapio programos modeliu paremtas karkasas skirtas kurti internetinius puslapius.

**React**– vieno puslapio programos modeliu paremta biblioteka skirta kurti internetinius puslapius.

**MODBUS** – duomenų perdavimo protokolas.

**MQTT** – atviras OASIS ir ISO standartų duomenų perdavimo protokolas.

**AJAX** – asinchroninė JavaScript ir XML svetainių programavimo technologija.

**JavaScript** – objektiškai orientuota skriptų programavimo kalba.

**TypeScript** – griežtas sintaksinis „JavaScript“ viršutinis rinkinys ir prie kalbos pridamas pasirenkamas statinis spausdinimas.

**OpenWrt** - viešojo kodo projektas skirtas įterptinių sistemų veikimui.

**Lua** – aukšto lygio programavimo kalba.

**LuCI** – OpenWrt projekte naudojamas internetinių puslapių kūrimo karkasas.

## Įvadas

Šis darbas priklauso Eimantui Žvirbliui, informacinių sistemų inžinerijos studentui.

### Darbo problematika ir aktualumas

Kasdieniniame mūsų gyvenime vis dažniau ir dažniau susiduriame su daiktų interneto (angl. Internet of things, toliau tekste - IoT) įrenginiais, kurie patenkina mūsų kasdieninius poreikius. IoT įrenginių tinklo funkcionavimui reikalingi specializuoti maršrutizatoriai, kuriuos mes vadinsime IoT maršrutizatoriais.

IoT maršrutizatoriai taip pat turi grafinę sąsają, kurios pagalba galime greitai ir patogiai sukongūruoti įrenginį pagal savo poreikius. Vienas didžiausių iššūkių daiktų interneto įrenginių gamintojams yra sukurti greitą ir patogią grafinę sąsają, leidžiančią vartotojams lengvai naudotis sistema. Deja, daiktų interneto įrenginiai dažniausiai nėra tokie galingi kaip šiuolaikiniai kompiuteriai, todėl kiekvienas daiktų interneto įrenginio funkcionalumas turi būti gerai apgalvotas ir optimizuotas. Maršrutizatorių grafinės sąsajos sistemai keliami tokie patys reikalavimai kaip ir šiuolaikinių kompiuterių internetinių puslapių sistemų. Todėl tradiciniai internetinių puslapių sprendimai, kuomet didžioji dalis loginės informacinės sistemos sprendimų priimami daiktų interneto serveryje, netinka, nes duomenų apdorojimas serveryje reikalauja daug resursų ir dėl to kenčia grafinės sąsajos puslapių užkrovimo laikas.

**Šiame darbe sprendžiama problema yra** – per ilgas maršrutizatoriaus valdymo informacinės sistemos užkrovimo laikas ir palyginti didelis resursų poreikis lyginant su maršrutizatoriaus pajėgumais.

**Tyrimo objektas** – IoT maršrutizatoriaus valdymo sistemos realizavimo technologijos ir jų galimybės.

**Tyrimo sritis** – IoT maršrutizatoriai, mažų resursų įrenginiai, interneto technologijomis grįstos sistemos ir interneto technologijomis grįstų sistemų realizavimo priemonės ir metodai.

### Darbo tikslas ir uždaviniai

**Tikslas** – sumažinti daiktų interneto maršrutizatoriaus informacinės sistemos puslapių užkrovimo laiką ir naudojamų resursų kiekį.

Uždaviniai:

1. Išanalizuoti principines internetinių puslapių technologijas;
2. Išanalizuoti vieno puslapio internetinių programų kūrimo procesą;
3. Išanalizuoti šiuo metu maršrutizatoriuje esančios valdymo PĮ veikimo modelį;
4. Suprojektuoti, realizuoti ir ištestuoti alternatyvius maršrutizatoriaus valdymo PĮ prototipus paremtus vieno puslapio internetinės programos veikimo modeliu;
5. Išmatuoti maršrutizatoriaus valdymo PĮ vartotojo sąsajos užkrovimo laikus ir resursų sueikvojimą buvusiam ir naujuose PĮ variantuose;
6. Atlikti palyginimą tarp maršrutizatoriaus valdymo PĮ versijų grindžiamų tradiciniu internetinių programų veikimo modeliu ir vieno puslapio internetinių programų veikimo modeliu;
7. Padaryti darbo išvadas ir apibendrinimus.



## **Darbo rezultatai ir jų svarba**

Darbo rezultatai yra aktualūs grafinės sąsajos kokybės ir našumo pagerinimui, o jų svarba aktuali IoT maršrutizatoriaus galutiniam naudotojui pagerinant sistemos naudojimo kokybę.

## **Darbo struktūra**

Dokumentą sudaro šešios pagrindinės dalys:

1. Pirmoji dalis – analizės dalis, skirta susipažinti su informacinių sistemų kūrimo modeliais, jų privalumais ir skirtumais;
2. Antroji dalis – kuriamo sprendimo specifikacija – aprašomas kuriamos vieno puslapio programos reikalavimų sąrašas;
3. Trečioji dalis – realizacijos projektas – aptariamas kuriamos sistemos realizacijos projektas;
4. Ketvirtoji dalis realizacija ir testavimas – aprašoma detali sukurtos grafinės sąsajos prototipo realizacija ir testavimas;
5. Penktoji dalis – eksperimentas – aprašomi atlikti eksperimentai, jų eiga ir rezultatai;
6. Šeštoji dalis – darbo išvados.

## 1. Probleminės srities analizė

### 1.1. Analizės tikslas

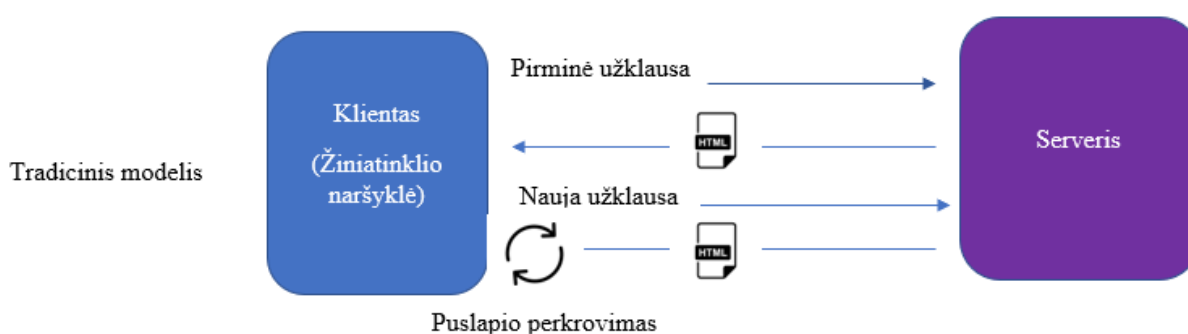
Analizės tikslas – supažindinti su informacinės sistemos kūrimo modeliais, technologijomis naudojančiomis analizuojamus modelius ir technologijų privalumais ir skirtumais.

### 1.2. Internetinių puslapių kūrimo modelių analizė

#### 1.2.1. Tradicinių internetinių puslapių modelio taikymas

Tradiciniai internetiniai puslapiai – klasikinio tipo internetinių puslapių programos, kurios paremtos kelių puslapių sąsajos modeliu [1]. Tai modelis, kuris su kiekvienu sistemos naudotojo paspaudimu siunčia užklausą į informacinės sistemos serverį iš naujo perkraudamas esantį puslapį gautiems duomenis iš serverio atvaizduoti [1].

Pirmieji tradicinių internetinių puslapių modeliu grindžiami internetiniai puslapiai buvo apibūdinti kaip „Web 1.0“. Tai statinio tipo puslapiai apibūdinami kaip „tik skaitymui“ skirti žiniatinklio puslapiai [2]. Šiuo modeliu sukurti puslapiai skirti atvaizduoti statinę informaciją, todėl informacijos atnaujinimas realiu laiku yra negalimas. Informacija atnaujinama tik po vartotojo veiksmų informacinėje sistemoje. Tradicinių internetinių puslapių modelis susideda iš kliento programos (žiniatinklio naršyklės) ir serverio, kuriame yra įdiegta informacinė sistema. Kliento programa kreipdamasi į serverį siunčia užklausą į serverį, iš kurio yra parsisiunčiamas statinis HTML puslapis duomenims atvaizduoti. Kiekvienas informacinės sistemos naudotojo įvykdytas veiksmas reikalaujantis duomenų atnaujinimo yra naujo failo atsiuntimas ir nukreipimas į naujai atsiųstą failą, todėl su kiekvienu naujai atsiųstu failu matomas žiniatinklio naršyklės puslapio persikrovimas. Tradicinių internetinių puslapių sąsajos modeliu grindžiamas internetinių puslapių programų veikimas pavaizduotas 1 paveiksle.



1. pav. Tradicinių internetinių puslapių sąsajos modelis

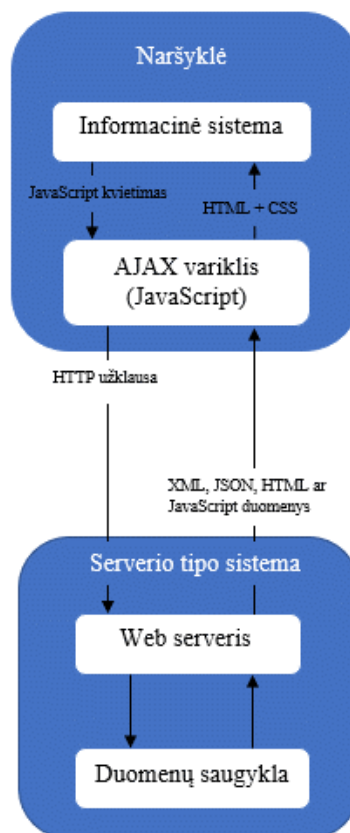
#### 1.2.2. Tradicinių internetinių puslapių modelio taikymas IoT maršrutizatoriuose

Daiktų interneto maršrutizatoriai – tai specializuoti daiktų interneto įrenginiai skirti industrijoje naudojamiems sprendimams pasiekti. Įrenginiai palaiko industrijoje naudojamus MODBUS, MQTT, SNMP protokolus, turi integruotą modemą, įvesties/išvesties sąsajas jungimui su kitais įrenginiais. Daiktų maršrutizatoriaus tikslas sujungti M2M bendru naudojamu protokolu ir perduoti reikiama informaciją įrenginio naudotojui [3]. Ne išimtis, daiktų interneto maršrutizatoriai taip pat turi grafinę sąsają, kurios pagalba yra palengvinamas sistemos konfigūravimas.

Pirmieji IoT maršrutizatoriai taip pat buvo paremti tradicinių internetinių puslapių modeliu, kadangi konfigūracija buvo paprasta ir nereikalaujanti duomenų apsikeitimo realiu laiku. Internetinių puslapių technologijoms tobulėjant kartu tobulėjo ir daiktų interneto maršrutizatoriuose naudojamos internetinių puslapių technologijos. Viena iš tokiu technologijų rinkinių – AJAX. Tai terminas, apibrėžiantis svetainių kūrimo technologiją, maksimaliam interaktyvumui pasiekti. Panaudojant AJAX technologijų rinkinį internetiniai puslapiai gali dalintis duomenimis su serveriu asinchroniškai nereikalaujant kisti atvaizduojamam internetinio puslapio turiniui. Ši technologijų rinkinys sukūrė sąlygas informacinėms sistemoms atnaujinti duomenis dinamiškai nereikalaujant puslapio perkrovimo. AJAX nėra nauja technologija, tai yra esamų technologijų panaudojimas nauju būdu. AJAX apjungia:

- Dokumento objekto modelį (angl. Document object model, toliau tekste - DOM) – daugiaplatformė ir nuo kalbos nepriklausanti sąsaja, vaizduojanti HTML dokumentą kaip medžio struktūrą;
- JSON arba XML duomenų perdavimo formatai skirti kliento programai apsikeisti duomenimis su serveriu;
- XMLHttpRequest (XHR) – metodas skirtas apsikeisti duomenimis tarp kliento programos ir serverio;
- HTML ir CSS panaudojimas atvaizduoti duomenis;
- JavaScript programavimo kalba apjungianti visas AJAX technologijas į vieną bendrą sprendimą.

AJAX technologijų rinkinys leido internetiniams puslapiams tapti interaktyviems ir atnaujinti duomenis realiu laiku [4]. Šios technologijos veikimo principas pavaizduotas 2 paveiksle.



2 pav. AJAX technologijos veikimas internetinių puslapių programose

Pritaikius AJAX technologiją daiktų interneto maršrutizatoriuose atsirado interaktyvumas ir duomenų atnaujinimas realiu laiku, kas padėjo atvaizduoti tokią svarbią informaciją apie gaminį, kaip sistemos procesoriaus darbo užimtumas, vidinės ir darbinės atminties užimtumas, informacija apie prisijungusius klientus prie sistemos, mobiliosios būsenos informacija, įvesčių/išvesčių būsenas ir kt. Daiktų interneto maršrutizatoriai nepasižymi techninėmis savybėmis kaip šiuolaikiniai kompiuteriai, įrenginiai dažniausiai turi nuo vieno iki kelių branduolių procesorių dirbanti nuo 600 MHz iki 1.4 GHz dažniu, 64 MB darbinės atminties ir 16 MB vidinės atminties. Atsiradus AJAX, informacinės sistemos puslapiai prisipildė JavaScript programinio kodo, leidžiančio duomenis atnaujinti realiu laiku. Puslapiai tapo interaktyvūs ir draugiški naudotojui, tačiau nukentėjo grafinės sąsajos našumas. Failai, kuriuos įrenginys siunčia kliento programai pasipildė netik HTML ir CSS, bet ir JavaScript programiniu kodu. Naudojantis tradicinio internetinių puslapių grindžiamu modeliu failo turinys yra pirmiausia sugeneruojamas serverio pusėje ir tik tada siunčiamas į kliento programą, kurioje yra vykdomas JavaScript. AJAX išsprendė duomenų atnaujinimo problemą realiu laiku, tačiau tradicinio modelio veikimas išliko toks pat ir norint atvaizduoti naują puslapį, šis turi būti sugeneruojamas iš naujo serverio dalyje. Kuo daugiau puslapio turinys turi programinio kodo tuo ilgiau serveris užtrunka sugeneruoti failą puslapio atvaizdavimui. Generuojami failai dažnu atveju turi vienodas antraštes, poraštes, meniu punktus ar kitą programinio kodo dalį tarp skirtingų failų. Tačiau kiekvieno failo generavimo metu šios sritys yra generuojamos iš naujo ir taip yra neefektyviai išnaudojami serverio resursai atliekant pasikartojančius veiksmus.

### **1.2.3. OpenWrt įterptinių sistemų įrenginiuose**

OpenWrt yra viešojo kodo projektas, skirtas įterptinių sistemų veikimui, remiantis Linux branduolio operacinės sistemos veikimu. Projektas labiausiai orientuotas į įterptinių sistemų įrenginius, skirtus nukreipti tinklo srautus. Pagrindiniai OpenWrt viešojo kodo projekto įrenginiai – maršrutizatoriai [12], tačiau OpenWrt taip pat galima panaudoti ir išmaniuosiuose telefonuose, kišeniniuose kompiuteriuose, nešiojamuose kompiuteriuose ir kituose kompiuteriuose, kurie sukurti remiantis x86 ar ARM procesoriaus architektūra. OpenWrt projekto bazę sudaro Linux branduolys, standartinė C programavimo kalbos biblioteka „musl“, naudojama kartu su Linux branduoliu ir BusyBox pagrindinių įterptinių sistemų programų rinkinys, skirtas atlikti pagrindines sistemos funkcijas. Visus šiuos įrankius apjungus į vieną projektą buvo sukurta OpenWrt viešojo kodo operacinė sistema skirta įterptinių sistemų įrenginiams. OpenWrt projektas siūlo netik operacinę sistemą įterptinių sistemų įrenginiams palaikyti, bet ir apie 3500 įvairių programų. Programos apima nuo bendravimo protokolų tarp sistemų įvairiais protokolais iki tinklo skanavimui ir statistikos rinkimui skirtų programų. OpenWrt operacinė sistema yra konfigūruojama ne tik komandinės eilutės pagalba, bet ir turi grafinę sąsają. OpenWrt grafinė sąsaja naudoja internetinių puslapių karkasą pavadinimu – LuCI. Tai unikalus internetinių puslapių kūrimo karkasas parašytas Lua programavimo kalba, kuris buvo sukurtas OpenWrt bendruomenės. LuCI grafinės sąsajos karkasas veikia MVC programinės įrangos projektavimo modeliu. Karkase atskirai yra aprašomos valdiklio, modulio ir atvaizdavimo dalys. OpenWrt grafinės sąsajos veikimas yra paremtas tradicinių internetinių puslapių kūrimo modeliu, kuomet kiekvienas puslapis su kiekviena naudotojo užklausa yra užkraunamas iš naujo. Internetinių puslapių duomenų atnaujinimui realizuoti yra pritaikytas AJAX technologijų rinkinys, tačiau pasikartojantis kiekvieno puslapio parsisiuntimas iš serverio kaip ir visuose tradiciniuose modeliuose paremtuose internetiniuose puslapiuose yra problema, kadangi įterptinių sistemų įrenginiai nepasižymi resursų gausa ir galingomis aparatinės įrangos specifikacijomis.

#### 1.2.4. Vieno puslapio programų modelio taikymas

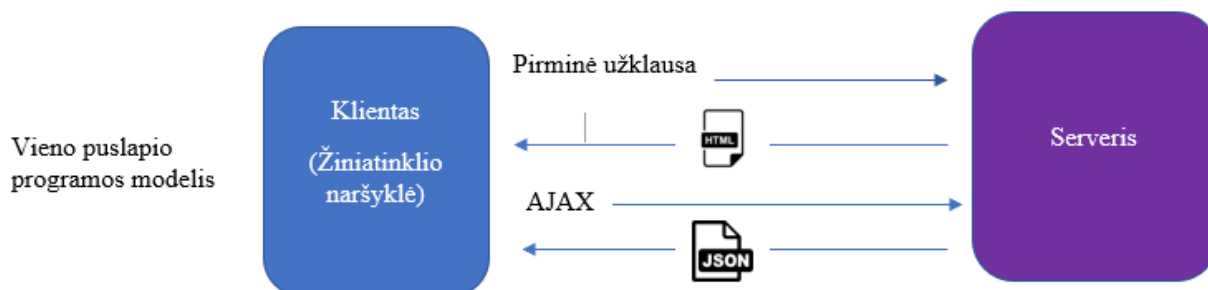
Vieno puslapio programa (angl. „SPA“) – žiniatinklio programa, kuri visą pilnam svetainės funkcionavimui reikalingą programinį kodą parsisiunčia į kliento programą pirmojo susijungimo metu [5]. Modelis susideda iš atskirų komponentų, kuriuos galima pakeisti ar atnaujinti nepriklausomai vienas nuo kito. Kiekvienas svetainės naudotojo veiksmas iššaukia komponentų atnaujinimą ar pakeitimą, kuris vyksta dinamiškai nereikalaujant puslapio perkrovimo. Svetainės puslapis niekada nėra perkraunamas, nes kiekvienas naujo turinio pridėjimas vyksta dinamiškai pridėdant ar atnaujinant naują informaciją. Puslapio naujo turinio atnaujinimas ir atvaizdavimas vyksta akimirksniu, kadangi visa funkcionalumo logika jau yra kliento programoje. Po pirmojo susijungimo su serveriu vieno puslapio programa iš serverio daugiau nebesisiunčia puslapio atvaizdavimo ir loginio programinio kodo, o tik užsiprašo konkrečių duomenų atnaujinimo. Duomenys tarp vieno puslapio programos ir serverio siunčiami JSON arba XML formatu. Informacija keliaujanti JSON arba XML formatu yra serverio dalyje sugeneruota informacija, kuri gali būti sudaryta iš duomenų bazės įrašų, programų išvesties duomenų ar kitų duomenų esančių serverio dalyje.

Pagrindinės technologijos, kurios leidžia vieno puslapio programai būti interaktyviai ir pasižyminčiai dideliu greičiu yra JavaScript programavimo kalba ir AJAX technologijų rinkinys.

JavaScript programavimo kalba buvo sukurta 1995 „Netscape“ kompanijoje, kuri nuo jos sukūrimo iki šių dienų yra viena pagrindinių tinklapių kūrimo dalių. JavaScript - tai programavimo kalba, skirta žiniatinklio programoms ir yra esminė technologija kuriant šiuolaikinius internetinius puslapius [6]. Šios kalbos programinis kodas yra parsisiunčiamas į kliento programą kartu su HTML ir CSS. Kliento programoje esantis interpretatorius, aptikęs JavaScript programinį kodą, jį suvykdo ir puslapio turinys atnaujinamas dinamiškai. JavaScript naudojama svetainės turinio šalinimui, kūrimui ar atnaujinimui.

AJAX – yra technologija, kuri leidžia internetinių puslapių programoms asinchroniškai apsikeisti duomenimis su serveriu nepriverčiant svetainės puslapio persikrauti, kas užtikrina interaktyvias ir greitai reaguojančias žiniatinklio programas [7].

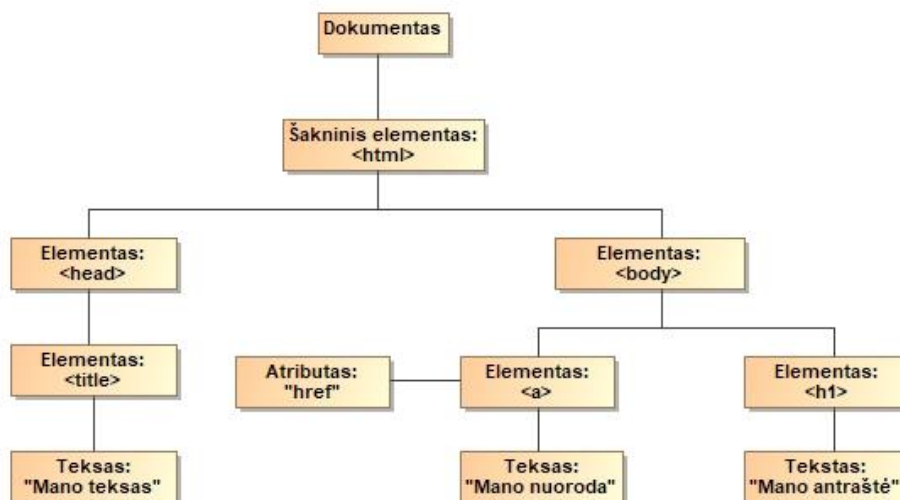
JSON – yra JavaScript objektų notacijos formatas. Tai formatas, kuris turi aprašytą struktūrą, pagal kurią AJAX technologijų pagalba kliento programa bendrauja su serveriu [8]. Žemiau 3 paveiksle pateiktas vieno puslapio programos veikimo modelis.



3 pav. Vieno puslapio programos veikimo modelis

Vieno puslapio programos interaktyvumo pagrindinis veikimo principas yra DOM manipuliavimas. Dokumento objekto modelis yra HTML arba XML dokumentų programavimo sąsaja. Kai žiniatinklio

naršyklėje yra užkraunamas puslapis, tuomet žiniatinklio programa sukuria puslapio medį, kuris atvaizduoja puslapio struktūrą. Dokumento objekto modelio pavyzdys pateiktas 5 paveiksle.



4 pav. Dokumento objekto modelis

Žiniatinklio puslapis yra dokumentas, kuris gali būti atvaizduojamas kaip turinys žiniatinklio puslapyje arba kaip HTML programinio kodo dokumentas. DOM yra tas pats dokumentas, kuris apima visus žiniatinklio puslapio HTML elementus. Visos ypatybės, metodai ir įvykiai, kuriais galima manipuluoti kuriant žiniatinklio puslapius yra suskirstyti į objektus. DOM yra objektinis tinklapio atvaizdavimas, kurį galima modifikuoti naudojant JavaScript programavimo kalbą. Vieno puslapio programos puslapiai nėra HTML puslapiai, tai yra dokumento objekto modelio dalys, sudarančios matomas žiniatinklio puslapio sritis. Po pirminio žiniatinklio programos ir serverio susijungimo visi reikalingi puslapiui atvaizduoti įrankiai ir rodiniai yra atsiunčiami ir paruošiami atvaizdavimui. Jei reikia atvaizduoti naują puslapį ar tam tikrą puslapio dalį, informacija atnaujinama naudotojo žiniatinklio naršyklėje dinamiškai prijungiant pasikeitusius duomenis prie dokumento objekto modelio naudojant JavaScript programavimo kalbą. Dėl šio dinamiško duomenų atnaujinimo vieno puslapio programoje niekada nereikia perkrauti puslapio.

Vieno puslapio programoje atvaizdavimo logikos perkėlimas į žiniatinklio programą yra labai svarbus aspektas sumažinant serverio resursų panaudojimą. Tačiau vieno puslapio programoje serverio dalis taip pat atlieką svarbų darbą visam vieno puslapio programos sprendimui egzistuoti. Žiniatinklio puslapiui atvaizduoti neužtenka perkelti duomenis į kliento programą, tačiau duomenis taip pat reikia ir atnaujinti. Duomenų atnaujinimui žiniatinklio programa su serveriu turi bendrauti tuo pačiu protokolu ir abejoms pusėms suprantamais metodais. Iš serverio į žiniatinklio programą vieno puslapio programoje yra siunčiami tik duomenys, kuriuos reikia atnaujinti dokumento objekto modelyje. Šie duomenys yra siunčiami JSON duomenų perdavimo formatu. JSON formatas yra lengvai suprantamas, todėl apdoroti ir panaudoti duomenis yra patogiu ir paprastu. Pagrindinė JSON struktūra yra raktų ir juos atitinkančių reikšmių rinkinys. Ši struktūra pateikta 5 paveiksle.

```

{
  "raktas1": "reikšmė",
  "raktas2": 0,
  "raktas3": [1, 2, 3, false, "ABC"],
  "raktas4": {
    "raktas4_5": true,
    "raktas4_6": 17
  }
}

```

5 pav. Pagrindinė JSON struktūra

JSON formatu duomenys yra perduodami tarp žiniatinklio programos ir serverio dalies. Serveris gavęs užklausą iš žiniatinklio programos, apdoroja užklausą, surenka reikiamus duomenis ir siunčia naujai suformuotus duomenis JSON formatu žiniatinklio programai. Žiniatinklio programa gavus atsakymą iš serverio apdoroja gautus duomenis JSON formatu ir atnaujina dokumento objekto modelį. Pagrindiniai aspektai dėl kurių vieno puslapio programos yra naudojamos šiuolaikinių internetinių puslapių kūrėjų, tai dėl serverio dalies resursų sumažinimo ir duomenų apdorojimo naudotojo žiniatinklio programoje.

### 1.2.5. Vieno puslapio programų pranašumai ir trūkumai

Per pastaruosius 30 metų nuo pirmųjų internetinių puslapių technologijų iki šiuolaikinių internetinių puslapių sprendimų praėjo daug laiko ir per šį laikotarpį kuriant žiniatinklio programas atsižvelgiama ne tik į turinio atvaizdavimą, bet ir kokiame įrenginyje tas turinys bus atvaizduojamas. Šiuolaikinės technologijos taip pasistūmėjo, kad žiniatinklio programos turi įvairūs įrenginiai turintys įvairias technines savybes, į kurias reikia atkreipti dėmesį. Vieno puslapio programų karkasų ir bibliotekų yra daugybė. Tai palengvina turinio atvaizdavimą, padeda kurti interaktyvų, greitą ir prisitaikantį puslapį nepriklausomai nuo įrenginio techninių savybių. Lyginant tradiciniu modeliu paremtus internetinius puslapius su vieno puslapio programomis galima išskirti esminius skirtumus:

- Puslapių užkrovimo greitis;
- Puslapių interaktyvumas;
- Žiniatinklio naršyklės spartinančiosios atminties panaudojimas;
- Optimizavimas paieškos sistemoms;
- Puslapių saugumo pažeidžiamumai.

Internetinių puslapių užkrovimo laikas yra vienas svarbiausių faktorių naudojantis internetiniu puslapiu, kadangi niekas nenori laukti, kol bus pateikiama reikalinga informacija. Remiantis Fiona Fui-Hoon Nah atliktu tyrimu, dėl naudotojų pasitenkinimo naudojantis interneto puslapiais, buvo nustatyta, kad turinio atvaizdavimas negali trukti ilgiau nei dvi sekundės norint išlaikyti naudotojo pasitenkinimą internetiniu puslapiu [13]. Vieno puslapio programos pagrindinis privalumas yra informacijos atnaujinimas akimirksniu lyginant su tradiciniu modelio internetiniais puslapiais, kurių kiekviena užklausa gali trukti skirtingą laiko tarpą priklausomai nuo internetinio puslapio dydžio. Vieno puslapio programos trūkumas dėl puslapių užkrovimo laiko gali pasižymėti pirmojo internetinio puslapio užkrovimo metu. Kadangi tuo metu yra parsiumčiami visi reikalingi internetinio puslapio failai, kurių parsiumtimas gali trukti ilgesnį laiko tarpą priklausomai nuo interneto greičio.

Naudojant vieno puslapio programų modelį internetiniuose puslapiuose galima realizuoti daug dinamiškesnį ir interaktyvesnį puslapį norint pagerinti bendrą puslapio naudojimosi kokybę. Naujo mygtuko, pranešimo, sąrašo ar kito elemento atsiradimas vykdomas iškart pasitelkiant interaktyvią animaciją. Tradiciniu modeliu taikomuose internetiniuose puslapiuose naujo elemento atsiradimas būtinas tik su puslapio perkrovimu, kadangi naują turinį serveris turi sugeneruoti iš naujo. Vieno puslapio programos internetinių puslapių interaktyvumo galimybės leidžia naudotis internetiniu puslapiu lyg tai būtų kompiuterio ar mobiliojo telefono programa. Dėl šio aspekto vieno puslapio programos puslapis naudotojui atrodo patrauklesnis, nei tradicinio modelio internetinis puslapis, nes yra suteikiamas kompiuterinės programos efektas.

Žiniatinklio naršyklės spartinančiosios atminties (angl. *Cache*) išnaudojimas vieno puslapio programose gali padaryti teigiamą įtaką naudotojo patirčiai naudojantis internetiniu puslapiu. Vieno puslapio programa gali kaupti reikiamus duomenis žiniatinklio naršyklės spartinančiojoje atmintyje, taip suteikdama galimybę internetiniu puslapiu naudotis esant prastam interneto ryšiui ar net tam tikrą laiką neturint interneto ryšio. Internetinio puslapio turinys yra parsijučiamas į žiniatinklio programos atminį, todėl, kol nereikia atnaujinti dinaminį duomenų iš serverio, tol internetiniu puslapiu galima pilnai naudotis. Tradicinių modelių paremti internetiniai puslapiai taip pat naudoja spartinančiąją atmintį, tačiau ši atmintis iš dalies veikia tik tame puslapyje, kuris yra tuo metu atidarytas.

Internetinių puslapių optimizavimas paieškos sistemoms ypač svarbus aspektas norint internetinį puslapį padaryti populiariu ir naudojamu. Vieno puslapio programos trūkumas yra internetinių puslapių paieškos sistemoms pritaikymas, nes internetinių puslapių optimizavimas paieškos sistemoms priklauso nuo HTML turinio. Paieškos sistemos reitinguoja internetinius puslapius pagal unikalias antraštes arba raktinius žodžius tikrindamos kiekvieną internetinės svetainės puslapį. Tradicinio modelio internetiniai puslapiai turi pranašumą lyginant su vieno puslapio programomis, nes jų struktūra sudaryta iš kelių puslapių, kuomet vieno puslapio programos turinį atvaizduoja viename puslapyje.

Internetinių puslapių saugumas yra vienas pagrindinių faktorių šiuolaikinėje internetinių puslapių visuomenėje. Tobulėjant internetinių puslapių technologijoms kartu su jomis atsirado ir daugiau pažeidžiamumų. Dažniausiai aptinkama internetiniuose tinklapiuose yra „Cross site scripting“ pažeidžiamumo ataka. Ši ataka yra naudojama įterpiant papildomą programinį kodą į naudotojo peržiūrimą puslapį. „Cross site scripting“ atakos tikslas išgauti konfidencialią informaciją apie naudotoją ir panaudoti ją blogiems tikslams. Internetiniai puslapiai, kuriems užkrauti reikalingas JavaScript palaikymas yra šios atakos taikiny. Vieno puslapio programose didžiąją dalį programinio kodo sudaro JavaScript programavimo kalba, todėl kuriant šio modelio internetinius puslapius reikia ypač didelį dėmesį skirti saugumui.

Apibendrinant vieno puslapio programos ir tradicinių internetinių modelių skirtumus galima išskirti:

- Vieno puslapio programos greitaveiką naršant tarp svetainės puslapių;
- Vieno puslapio programos lanksčias interaktyvumo galimybes suteikiančias internetiniams puslapiams dinaminį informacijos atnaujinimą.

Vieno puslapio programos greitaveikos ir dinaminės informacijos atnaujinimas pranoksta tradicinių internetinių puslapių modelių grindžiamus puslapius ir suteikia galimybes karkasų ir bibliotekų pagalba kurti modernius internetinius puslapius.



### 1.2.6. Vieno puslapių programos karkasas „Angular“

Angular yra atviro kodo JavaScript karkasas palaikomas Google įmonės ir bendruomenės. Šio karkaso tikslas padėti kurti internetinių puslapių programas MVC projektavimo pagrindu. MVC – tai programinės įrangos projektavimo modelis, paprastai naudojamas kuriant vartotojo sąsajas, suskaidantis susijusios programos logiką į tris tarpusavyje sujungtus elementus [5]. Angular MVC modelio elementai:

- Valdiklis (angl. Controller) – elementas atsakingas už duomenų perdavimą į atvaizdavimo elementą;
- Modelis (angl. Model) – elementas atsakingas už duomenų apdorojimą;
- Atvaizdavimas (angl. View) – elementas atsakingas už atvaizdavimą.

Pagrindinė Angular MVC modelio idėja yra turėti aiškų programinio kodo atskyrimą tarp duomenų apdorojimo (Modelis), kuriamos programos loginės srities (Valdiklis) ir informacijos atvaizdavimo naudotojui (Atvaizdavimas). Šie trys elementai pasirūpina paprastu ir struktūrizuotu internetinių puslapių programų kūrimu išskaidant funkcionalumus pagal jų panaudojimo paskirtį. Angular MVC modelio veikimas: pirmiausia atvaizdavimo valdikliui perduodami duomenys iš modelio valdiklio duomenų atvaizdavimui. Naudotojui atliekant internetiniame puslapyje rašymo, mygtukų paspaudimo ar kitus veiksmus, valdiklio elementas užtikrina duomenų atnaujinimą modelio elemente. Galiausiai modelio elementas informuoja atvaizdavimo elementą apie pasikeitusius duomenis, kuriuos atvaizdavimo elementas atnaujina internetiniame puslapyje [14].

Angular su MVC modeliu taip pat turi ir realizuotą duomenų įpareigojimą (angl. Data Binding). Duomenų įpareigojimas yra procesas, kuomet pasikeitus duomenims, šių duomenų atnaujinimas internetiniame puslapyje įvyksta automatiškai. Tradicinių modelių paremtuose internetiniuose puslapiuose reikia papildomo programinio kodo internetinio puslapio būsenų sekimui ir logikos duomenų atnaujinimui po būsenos pasikeitimo. Angular karkasas šį procesą išsprendė sukurdami papildomą mechanizmą, kuris periodiškai tikrina ar duomenys nepasikeitė. Pasikeitus duomenims, duomenų įpareigojimo mechanizmas sutikrina, kurie duomenys pasikeitė ir atitinkamai atnaujina susijusius elementus internetiniame puslapyje [14]. Angular karkasas naudoja dokumento objekto modelio formavimo mechanizmą, kuris leidžia papildyti HTML sintaksę. HTML puslapiuose karkasas leidžia pridėti aprašytus Angular karkasui priklausančius kintamuosius, kurių pagalba yra naudojamas duomenų įpareigojimas, atvaizdavimo elementų dinaminis atnaujinimas ir kitas kuriamo internetinio puslapio veikimas.

### 1.2.7. Vieno puslapio programų biblioteka „React“

React – tai vieno puslapio programų biblioteka, kuri sprendžia puslapių kūrimo problemas iš MVC modelio atvaizdavimo perspektyvos [9]. Ši biblioteka suskaido puslapio atvaizdavimo dalis į komponentus. Šie komponentai apima ir pačią programinio kodo logiką ir patį atvaizdavimą. Pagrindinis React bibliotekos išskirtinumas lyginant su kitomis vieno puslapio programomis yra virtualus dokumento objekto modelis. Norint atvaizduoti atsinaujinančią internetinio puslapio informaciją reikalingas dokumento objekto modelio atnaujinimas. Dokumento objekto modelio atnaujinimas reikalingas dvejais įvykių tipais:

- Po internetinio puslapio naudotojo įvykių (rašymo, paspaudimo, slinkimo ar kt.);
- Po serverio duomenų atsiuntimo vieno puslapio programai.

Daugumoje modernių karkasų ar bibliotekų dokumento objekto modelio atnaujinimas vykdomas duomenų įsipareigojimo metodu. Tačiau React remiasi idėja, kad dokumento objekto modelio atnaujinimas duomenų įsipareigojimo metodu yra brangi operacija. Todėl React siūlomas sprendimas informacijos atnaujinimui yra virtualus dokumento objekto modelis. React virtualaus dokumento objekto modelio veikimas:

1. Pirmą kartą prisijungus prie internetinio puslapio React biblioteka sugeneruoja virtualų dokumento objekto modelį;
2. Pasikeitus duomenims internetiniame puslapyje React biblioteka sugeneruoja naują virtualų dokumento objekto modelį;
3. React biblioteka sulygina virtualius dokumento objekto modelius ir surenka jų skirtumus;
4. Atrinkti skirtumai yra perkeltami į originalų dokumento objekto modelį.

Šis React duomenų atnaujinimo procesas naudojant virtualų dokumento objekto modelį yra daug greitesnis ir efektyvesnis už kitų karkasų naudojamus metodus, nes daugelis vieno puslapio programų pergeneruoja visą originalų dokumento objekto modelį kliento programoje, kas daro įtaką nereikalingų komponentų pergeneravimui.

### **1.2.8. Vieno puslapio programų biblioteka „Vue“**

Vue – tai vieno puslapio programų karkasas, kuris, kaip ir React, sprendžia puslapio problemas iš MVC modelio atvaizdavimo perspektyvos [10]. Vue buvo sukurtas panaudojant alternatyvių karkasų gerąsias savybes, todėl kaip ir React naudoja virtualų dokumento objekto modelio atnaujinimą internetinio puslapio duomenims atnaujinti ir taip pat kaip ir Angular turi supaprastintą HTML elementų papildymo funkcionalumą vadinamą – direktyvomis. Vue naudoja HTML tipo šablonus internetiniams puslapiams kurti, kuriuose aprašomos direktyvos virtualiam dokumento objekto modeliui atnaujinti. Vue vienas iš svarbesnių funkcijų yra komponentai. Šie komponentai padeda išplėsti pagrindinius HTML elementus pasikartojančiam funkcionalumui sukurti. Pagrindiniai karkaso principai:

- Paprastumas – Vue sintaksė yra paprasta ir lengvai įsisavinama;
- Kompaktiškumas – pagrindinė suspausta skriptų biblioteka sveria apie 33 KB ir tai yra viena iš mažiausiai sveriančių bibliotekų lyginant su jų alternatyvomis;
- Greitis – atvaizdavimo greičių analizės rezultatai lenkia daugelį karkasų;
- Universalumas – karkaso panaudojimas tinkamas nuo mažų internetinių puslapių projektų iki didelių vieno puslapių programų sprendimų.

### **1.2.9. Internetinių puslapių užkrovimo pagreitinimo gerosios praktikos**

Visais internetinių puslapių gyvavimo laikais buvo svarbus tinklapių reakcijos laikas ir kaip greitai puslapis reaguoja į naudotojo veiksmus. Tradicinių internetinių puslapių programų metodais už puslapio greitį yra atsakingas serveris ir aparatinė įranga, kurioje veikia serveris. Vieno puslapio programų metoduose už puslapio greitį yra atsakinga naudotojo žiniatinklio naršyklė ir įranga, kurioje veikia žiniatinklio naršyklė. Žinoma, pagrindinis faktorius yra programinis kodas, kuris turi būti kokybiškas ir optimizuotas, nes net ir labai galinga įranga gali veikti lėtai ir neefektyviai. Visgi, puslapio užkrovimo optimizavimo būdų yra ne vienas ir svarbu atkreipti dėmesį į kiekvieną iš jų siekiant pagerinti puslapių užkrovimo laiką. Optimizavimo metodai [11]:

- Organizuotas HTML, CSS ir JavaScript failų medis;
- Minimalus HTTP užklausų kiekis tarp kliento ir serverio;

- Programinio kodo suspaudimas, kad siunčiamos užklausos užimtų kuo mažiau dydžio;
- Strategiškai ir efektyviai panaudoti laikinosios atminties (angl. „Cache“) resursai;
- Suspaustos nuotraukos siunčiamos tarp kliento ir serverio.

### 1.2.10. Internetinių puslapių technologijų kūrimo palyginimas

Šiuolaikinių technologijų pagalba yra sukurta daugybė bibliotekų ir karkasų, kurie palengvina internetinių puslapių kūrimą ir jų palaikymą. Tačiau kiekvienas kūrimo metodas turi savąsias stiprybes ir silpnybes, toliau pateikiamos tradicinio ir vieno puslapio programų karkasų ir bibliotekų svarbiausios savybės.

Tradicinių internetinių puslapių modelis – seniausiai naudojamas modelis, tinkamas labai paprastiems puslapiams, kur nėra reikalingas interaktyvumas ir duomenų atvaizdavimas realiu laiku. Šis internetinių puslapių kūrimo modelis sutinkamas vis rečiau, dėl savo riboto interaktyvumo galimybių ir didesniu resursų išnaudojimu serverio dalyje.

Angular – seniausias iš aptartų vieno puslapio programų karkasų. Karkaso architektūra paremta MVC modeliu, turi gerai aprašytą dokumentaciją. Didžiausias trūkumas šio karkaso lyginant su alternatyvomis yra dokumento objekto modelio atnaujinimas. Karkaso dokumento objekto modelio atnaujinimo mechanizmas nusileidžia kitų karkasų virtualaus dokumento objekto modelio atnaujinimo mechanizmui. Karkasas labiausiai tinkamas projektams apimantiems dideles komandas.

React – galinga ir lanksti JavaScript biblioteka, lengvai suprantama ir išmokstama, tačiau biblioteka pagal MVC modelį sprendžia tik V dalį – atvaizdavimą. Norint realizuoti pilną sprendimą su React biblioteka reikia naudoti papildomas bibliotekas, kurios gali apsinkinti puslapio kūrimą palyginus su Angular. Didžiausias React bibliotekos privalumas yra virtualus dokumento objekto modelio panaudojimas, kuris leidžia bibliotekai pasiekti aukštus greičio rezultatus lyginant su kitais karkasais.

Vue – naujausias JavaScript karkasas iš aukščiau aptartų technologijų. Didžiausia karkaso stiprybė yra mažas dydis, puslapių atvaizdavimo greitis ir gera dokumentacija. Vue buvo sukurtas panaudojant alternatyvių karkasų gerąsias savybes apjungiant į vieną internetinių puslapių kūrimo karkasą.

2020 metais Mattias Levlin atlikto tyrimą, kuriame ištyrė Vue, React ir Angular technologijų greitaveiką. Tyrimo metu buvo testuojama kaip greitai elementai yra pridunami, redaguojami ir trinami internetiniame puslapyje [15]. M. Levlin testavimo rezultatai pateikiami 1 lentelėje.

1 lentelė. M. Levlin testavimo greitaveikos palyginimas

Testavimo scenarijus	Angular v8.2.14	React v16.12.0	Vue 2.6.11
10000 elementų pridėjimas (ms)	58,75	30,96	25,36
Vieno elemento redagavimas	6,08	16,58	22,23
10000 elementų redagavimas	896,76	17,86	20,64
Vieno elemento ištrynimasis	0,09	16,54	24,51
10000 elementų ištrynimasis	23,83	7,39	33,33

Remiantis M. Levlin rezultatų duomenimis, matoma, kad React ir Vue geriau susidoroja su daugiau elementų turinčiais testavimo scenarijais. Angular geriau pasirodė vieno elemento testavimo scenarijuose pralenkiant React ir Vue technologijas.

Lyginant vieno puslapio programos technologijas svarbų vaidmenį atlieka ne tik technologijos greitaveika, bet ir technologijos dydis, kuris ypač aktualus IoT maršrutizatoriui. Kuriant internetinius puslapius taip pat svarbu, kad technologija turėtų gerą dokumentaciją, gerus atsiliepimus ir bendruomenės palaikymą, kadangi pagal šiuos parametrus galima susidaryti geriausią nuomonę apie technologiją. Vieno puslapio programos modelių palyginimas pateikiamas 2 lentelėje.

**2 lentelė.** Vieno puslapio programų karkasų ir bibliotekų palyginimas

Nr.	Kriterijus	Angular v8.2.14	React v16.12.0	Vue.js v2.6.11
1	Technologijos programinio kodo dydis (kb)	54,03	5,36	29,95
2	DOM atnaujinimas	Periodinis DOM tikrinimas	Virtualus DOM	Virtualus DOM
3	Dokumentacijos skirtingų kalbų palaikymas	6	16	9
4	Teigiamas pasitenkinimas technologija „StateOfJs“ 2021 m. duomenimis (%)	45	84	80
5	Technologijų populiarumas „Github“ internetinėje svetainėje 2022 m. duomenimis	80 tūkst. naudotojų	187 tūkst. naudotojų	195 tūkst. naudotojų
6	Technologijos išleidimo data	2016	2013	2014

Vieno puslapio programų karkasų ir bibliotekų palyginimo lentelėje pateikti duomenys yra remiantis:

1. Technologijos programinio kodo dydis (kb) – suspaustos technologijos programinis kodas, kuris yra parsisiunčiamas prieš naudojantis technologija projekte.
2. DOM atnaujinimas – dokumento objekto modelio atnaujinimo metodas apibūdinantis technologijos metodą atnaujinti internetinio puslapio elementų atvaizdavimą.
3. Dokumentacijos skirtingų kalbų palaikymas – kiekvienos technologijos internetiniame puslapyje oficialiai skelbiamos kalbos palaikymas.
4. Teigiamas pasitenkinimas technologija „StateOfJs“ 2021 m. duomenimis (%) - remiantis 2021 metų internetinės svetainės „StateOfJs.com“ duomenimis buvo surinktas vieno puslapio programų sąrašas, kuriame atlikto tyrimo metu nustatytas bendruomenės pasitenkinimas technologijomis [16].
5. Technologijų populiarumas „Github“ internetinėje svetainėje 2022 m. duomenimis – vienas populiariausių programuotojų bendruomenės internetinis puslapis „Github.com“ talpinantis įvairių pasaulio kompanijų ir individualių asmenų viešąjį kodą, taip plečiant pateikiamo sprendimo populiarumą ir auginant bendruomenę.
6. Technologijos išleidimo data – technologijos oficiali išleidimo data.

Atsižvelgus į vieno puslapio programos modelio technologijų greitaveikas, jų dydžius, populiarumą ir bendruomenės palaikymą galima teigti, kad iš šių trijų analizuotų technologijų Angular rodo prasčiausius rezultatus. Vue ir React technologijų rezultatai nerodo didelio pranašumo viena prieš kitą, tačiau lyginant su Angular matomas pranašumas puslapių greitaveikos ir užimamos vietos rezultatuose. Įvertinus ir apibendrinus vieno puslapio programų modelio technologijas matoma, kad vienos aiškios technologijos nėra, kuri pirmautų visuose scenarijuose, todėl siekiamam tikslui pasiekti bus panaudotos Vue ir React technologijos.

### **1.3. Analizės išvados**

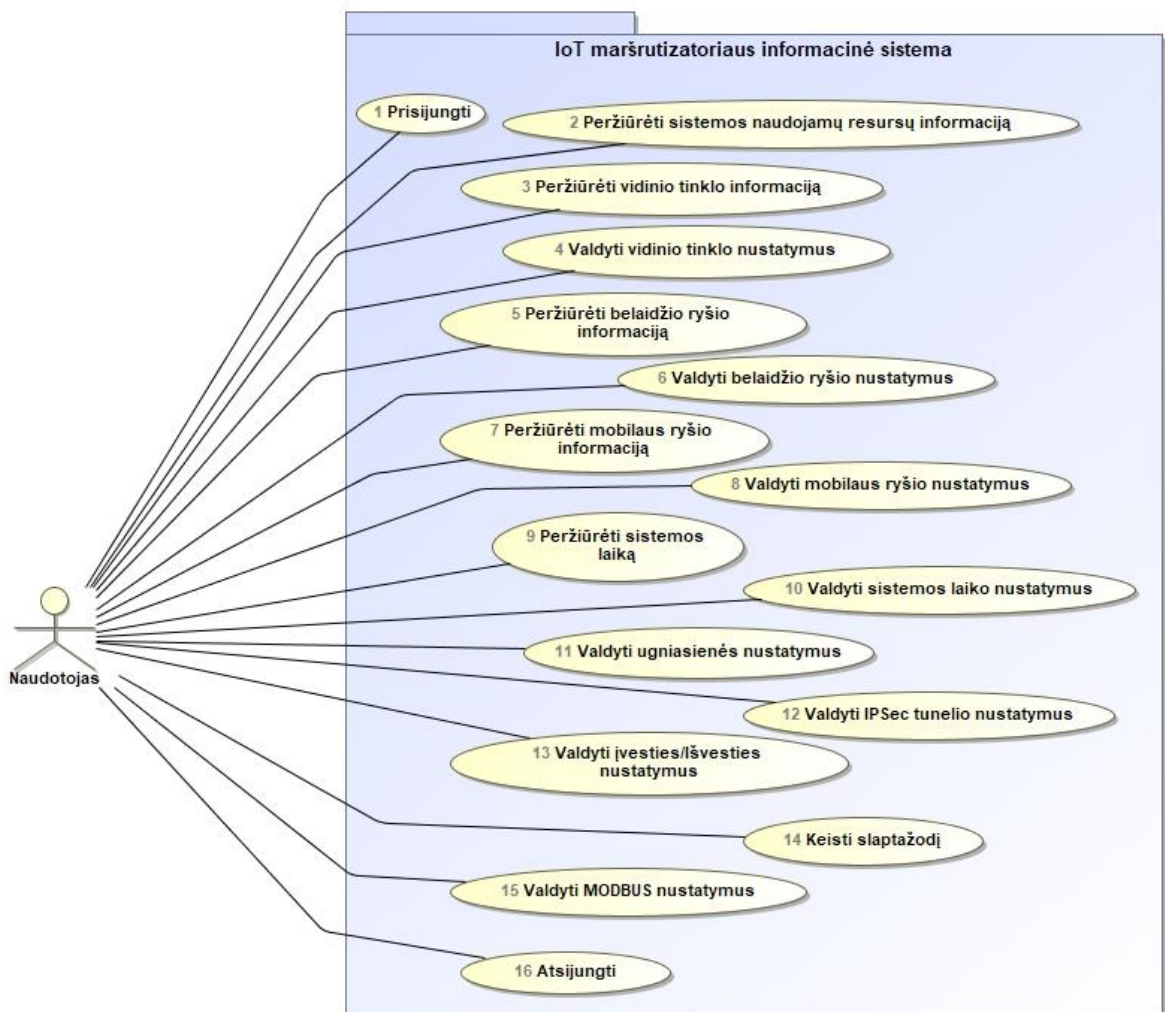
1. Internetinių puslapių per lėtas veikimas yra dėl taikomo tradicinių internetinių puslapio modelio, kuomet didžioji puslapių generavimo logika atliekama serverio dalyje.
2. Internetinių puslapių užkrovimą pagreitina vieno puslapio programų modelio taikymas, kuomet didžioji puslapių generavimo logika atliekama kliento dalyje.
3. IoT maršrutizatoriaus internetinių puslapių pagreitinimui pasirinktos Vue ir React vieno puslapio programos technologijos, pasižyminčios virtualaus DOM technologija leidžiančią pasiekti geriausius rezultatus.
4. Efektyviausiam internetinių puslapių pagreitinimui užtikrinti svarbu panaudoti gerąsias internetinių puslapių pagreitinimo praktikas, t.y., programinio kodo suspaudimas ir minimalus HTTP užklausų kiekis tarp serverio ir žiniatinklio naršyklės.

## 2. Grafinės sąsajos optimizavimo metodų daiktų interneto įrenginiuose sprendimo reikalavimų specifikacija ir projektas, formalus aprašas

Šiame skyriuje aprašomi sprendimui keliami reikalavimai ir jo veikimo principai. Sprendimui pasiekti bus suprojektuotos ir realizuotos dvi informacinės sistemos panaudojant Vue ir React vieno puslapio programos modelio technologijas identiškai atkartojant originalią IoT maršrutizatoriaus informacinę sistemą. Realizuotoms informacinėms sistemoms bus suprojektuota ir realizuota automatizuota internetinių puslapių užkrovimo greಿತaveikos ir resursų panaudojimo testavimo programa.

### 2.1. Funkcinių reikalavimų specifikacija

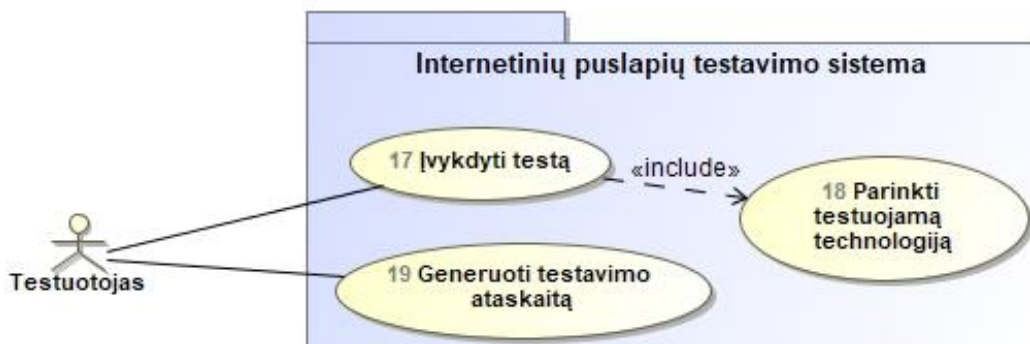
Remiantis analizės dalimi, internetinių puslapių užkrovimo laiko pagreitinimui, nuspręsta atnaujinti naudojamą internetinių puslapių kūrimo modelį. Šiam modeliui atnaujinti nuspręsta panaudoti vieno puslapio programos modelį. Atnaujinimas susidės iš informacinės sistemos sukūrimo, paremto vieno puslapio programos modeliu ir automatizuoto informacinės sistemos puslapių užkrovimo testavimu, taip pateikiant esamos ir naujai sukurtos informacinės sistemos skirtumus. Informacinės sistemos sukūrimo panaudojimo atvejų diagrama pavaizduota 7 pav.



7 pav. IoT maršrutizatoriaus informacinės sistemos panaudojimo atvejų diagrama

Pavaizduotoje diagramoje, naudotojas yra pagrindinis asmuo, kuris naudojasi informacine sistema. Ši sistema suteikia galimybę valdyti daiktų interneto įrenginį. Informacinės sistemos valdymas apima mobilaus ryšio, belaidžio ryšio, sistemos laiko, ugniasienės, IPSec tunelio, įvesties/išvesties, MODBUS komunikacijos protokolo ir sistemos slaptažodžio konfigūracijos keitimo galimybes ir būsenų peržiūrą.

Automatizuotas puslapių užkrovimo laiko testavimas turi apimti išorinę programą, kuri automatizuotai prisijungtų prie informacinės sistemos ir atliktų puslapių užkrovimo laiko testavimą. Automatizuotos puslapių užkrovimo testavimo panaudos atvejų diagrama pavaizduota 8 pav.



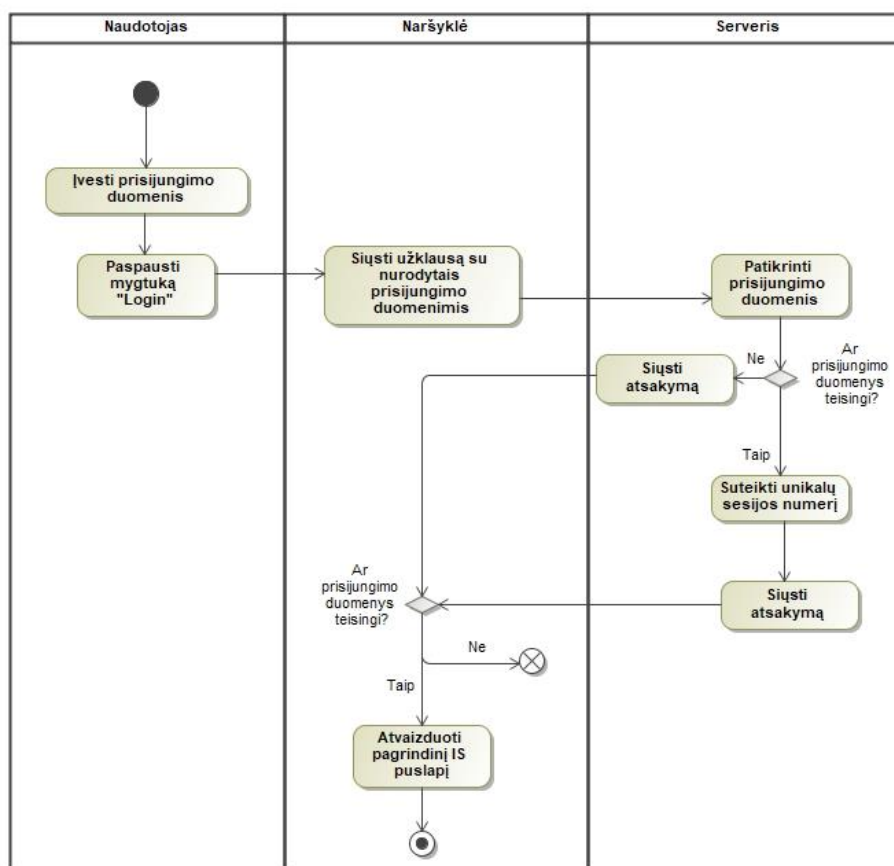
8 pav. Automatizuota puslapių užkrovimo programos panaudos atvejų diagrama

Kaip pavaizduota diagramoje testuotojas yra pagrindinis sistemos naudotojas, kuris naudojasi testavimo programa. Automatizuotos puslapių užkrovimo programos funkciniai reikalavimai yra įvykdyti testą, parinkti technologiją, kurios pagrindu bus vykdomas testavimas ir generuoti testavimo ataskaitą.

3 lentelėje pateikiama IoT maršrutizatoriaus informacinės sistemos panaudojimo atvejo „Prisijungti“ specifikacija, o 9 pav. pateikiama PA 1 „Prisijungti“ panaudojimo atvejo veiklos diagrama.

3 lentelė. PA 1 „Prisijungti“ specifikacija

PA 1 Prisijungti	
<b>Tikslas/uždavinys.</b> Prisijungti prie informacinės sistemos.	
<b>Aprašymas.</b> Naudotojas žinantis prisijungimo duomenis prisijungia prie informacinės sistemos.	
<b>Prieš-sąlyga</b>	Matomas prisijungimo puslapis.
<b>Aktorius</b>	Naudotojas.
<b>Sužadavimo sąlyga</b>	Naudotojas prisijungimo puslapyje įveda prisijungimo vardą ir prisijungimo slaptažodį ir spaudžia mygtuką [Login].
<b>Po-sąlyga</b>	Matomas pagrindinis informacinės sistemos puslapis.



9 pav. PA 1 „Prisijungti“ veiklos diagrama

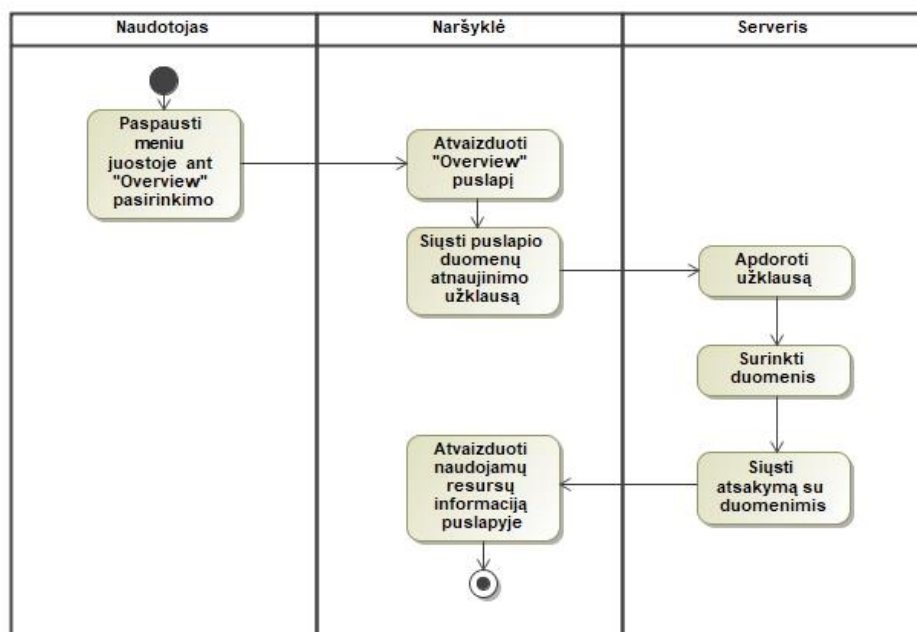
9 pav. pateiktas prisijungimo prie informacinės sistemos procesas, kuomet naudotojas įveda prisijungimo duomenis ir paspaudžia mygtuką „Login“. Naršyklė aptikus įvestus duomenis perduoda juos į serverį, kuriame jie yra patikrinami ir teisingų duomenų atvejų naudotojas yra peradresuojamas į pagrindinį sistemos puslapį.

4 lentelėje pateikiama panaudojimo atvejo sistemos naudojamų resursų peržiūros specifikacija, o 10 pav. pateikiama PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ veiklos diagrama, kurioje pateikiama sistemos naudojamų resursų informacija. Pagrindinį „Overview“ puslapį galima pasiekti meniu juostoje pasirinkus įrašą „Overview“. Šiame puslapyje yra atvaizduojama naudojamų resursų informaciją, kuri yra gaunama išsiunčiant užklausą į serverį.

4 lentelė. PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ specifikacija

PA 2 Peržiūrėti sistemos naudojamų resursų informaciją	
<b>Tikslas/uždavinys.</b> Peržiūrėti sistemos naudojamų resursų informaciją realiu laiku.	
<b>Aprašymas.</b> Informacinės sistemos pagrindiniame puslapyje „Overview“ atvaizduojama sistemos naudojamų resursų informacija.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Overview].
<b>Po-sąlyga</b>	Pagrindiniame IS „Overview“ puslapyje atvaizduojama naudojamų sistemos resursų informacija lentelėje.



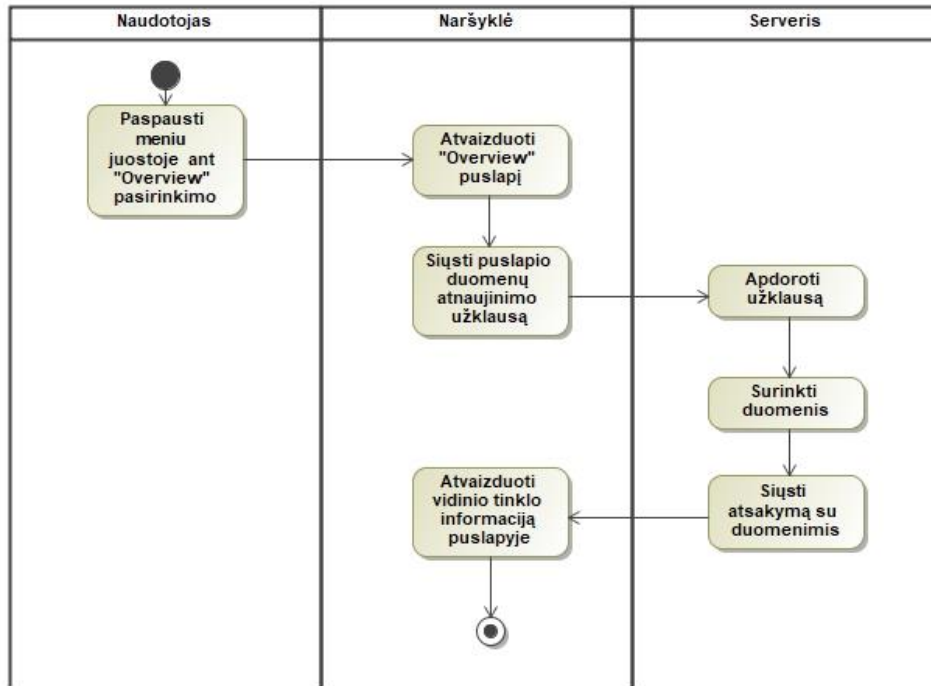


10 pav. PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ veiklos diagrama

5 lentelėje pateikiama „Peržiūrėti vidinio tinklo informaciją“ panaudojimo atvejo specifikacija, o 11 pav. pateikiama PA 3 „Peržiūrėti vidinio tinklo būseną“ veiklos diagrama, kurioje pateikiama vidinio tinklo informacijos peržiūra. Pagrindiniame IS „Overview“ puslapyje atvaizduojama vidinio tinklo informaciją lentelėje, kurioje matomas vidinio tinklo IP adresai, kaimė ir vidinio tinklo klientų skaičius. Duomenys gaunami iš serverio.

5 lentelė. PA 3 „Peržiūrėti vidinio tinklo informaciją“ specifikacija

PA 3 Peržiūrėti vidinio tinklo informaciją	
Tikslas/uždavinys. Peržiūrėti vidinio tinklo informaciją.	
Aprašymas. Informacinės sistemos pagrindiniame puslapyje „Overview“ atvaizduojama vidinio tinklo informacija.	
Prieš-sąlyga	Baigiama veikla „Prisijungti“.
Aktorius	Naudotojas.
Sužadinimo sąlyga	Menu juostoje paspaudžiamas pasirinkimas [Overview].
Po-sąlyga	Pagrindiniame IS „Overview“ puslapyje atvaizduojama vidinio tinklo informacijos lentelė.

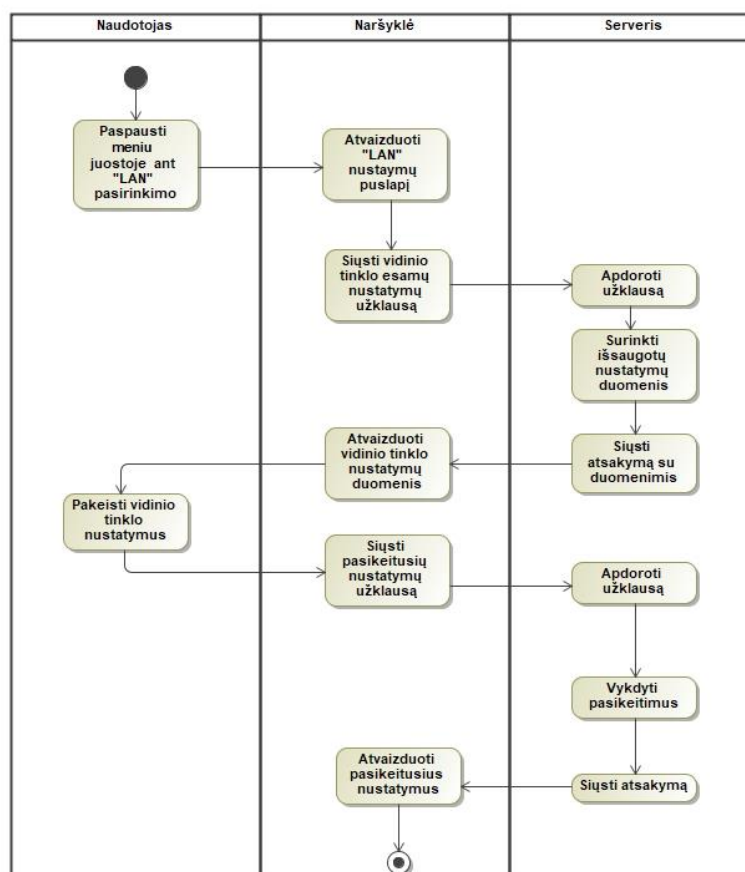


11 pav. PA 3 „Peržiūrėti vidinio tinklo informaciją“ veiklos diagrama

6 lentelėje pateikiama panaudojimo atvejų PA 4 „Valdyti vidinio tinklo nustatymus“ specifikacija, o 12 pav. pateikiama „Valdyti vidinio tinklo nustatymus“ veiklos diagrama, kurioje atvaizduojamas vidinio tinklo nustatymų keitimas. Perėjus į vidinio tinklo nustatymų keitimo puslapį serveris atvaizduoja esamą konfigūraciją. Naudotojas pakeitęs konfigūracijos nustatymus spaudžia mygtuką „Save“ ir naršyklė išsiunčia užklausą su konfigūracijos pakeitimais į serverį. Serveris apdoroja užklausą ir atsienčia atsakymą pagal kurį naršyklėje yra atvaizduojami nauji nustatymai.

6 lentelė. PA 4 „Valdyti vidinio tinklo nustatymus“ specifikacija

PA 4 Valdyti vidinio tinklo nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti vidinio tinklo nustatymus.	
<b>Aprašymas.</b> Vidinio tinklo nustatymų puslapyje keičiami vidinio tinklo nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [LAN].
<b>Po-sąlyga</b>	Vidinio tinklo nustatymų puslapyje „LAN“ atvaizduojami vidinio tinklo konfigūracijos laukeliai su išsaugotomis reikšmėmis.

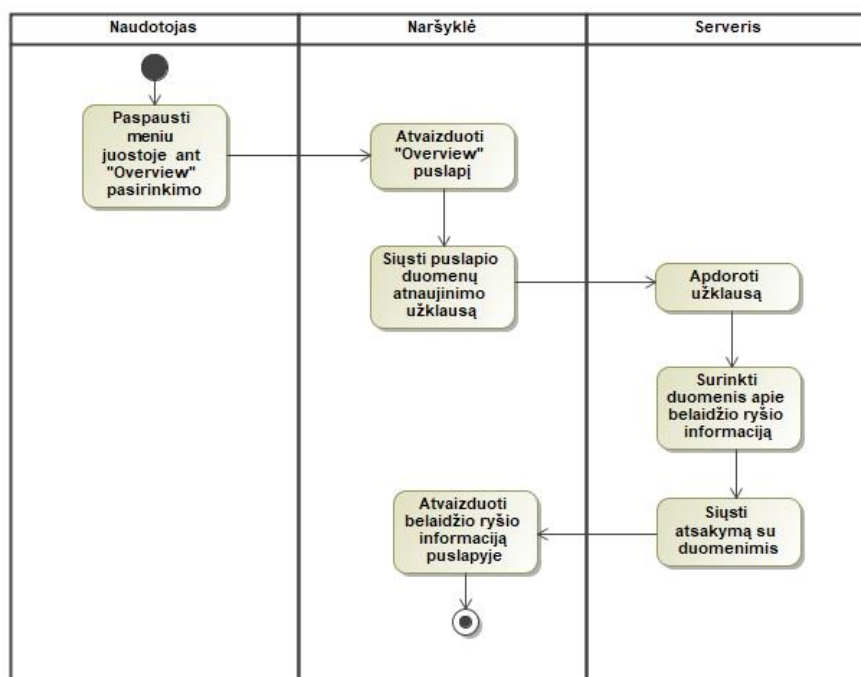


12 pav. PA 4 „Valdyti vidinio tinklo nustatymus“ veiklos diagrama

7 lentelėje pateikiama PA 5 „Peržiūrėti belaidžio ryšio informaciją“ specifikacija, o 13 pav. pateikiama PA 5 „Peržiūrėti belaidžio ryšio informaciją“ veiklos diagrama, kurioje pateikiama belaidžio ryšio nustatymų peržiūra. Pagrindiniame IS „Overview“ puslapyje atvaizduojama lentelėje, kuri yra užpildoma iš serverio gauta informacija apie belaidžio ryšio nustatymus. Lentelėje pateikiamas IoT maršrutizatoriaus belaidžio ryšio pavadinimas, tipas ir naudojamas kanalas.

7 lentelė. PA 5 „Peržiūrėti belaidžio ryšio informaciją“ specifikacija

PA 5 Peržiūrėti belaidžio ryšio informaciją	
<b>Tikslas/uždavinys.</b> Atvaizduoti belaidžio ryšio būseną.	
<b>Aprašymas.</b> Informacinės sistemos pagrindiniame puslapyje atvaizduojama belaidžio ryšio informacija	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Overview].
<b>Po-sąlyga</b>	Pagrindiniame IS „Overview“ puslapyje atvaizduojama belaidžio ryšio informacijos lentelė.

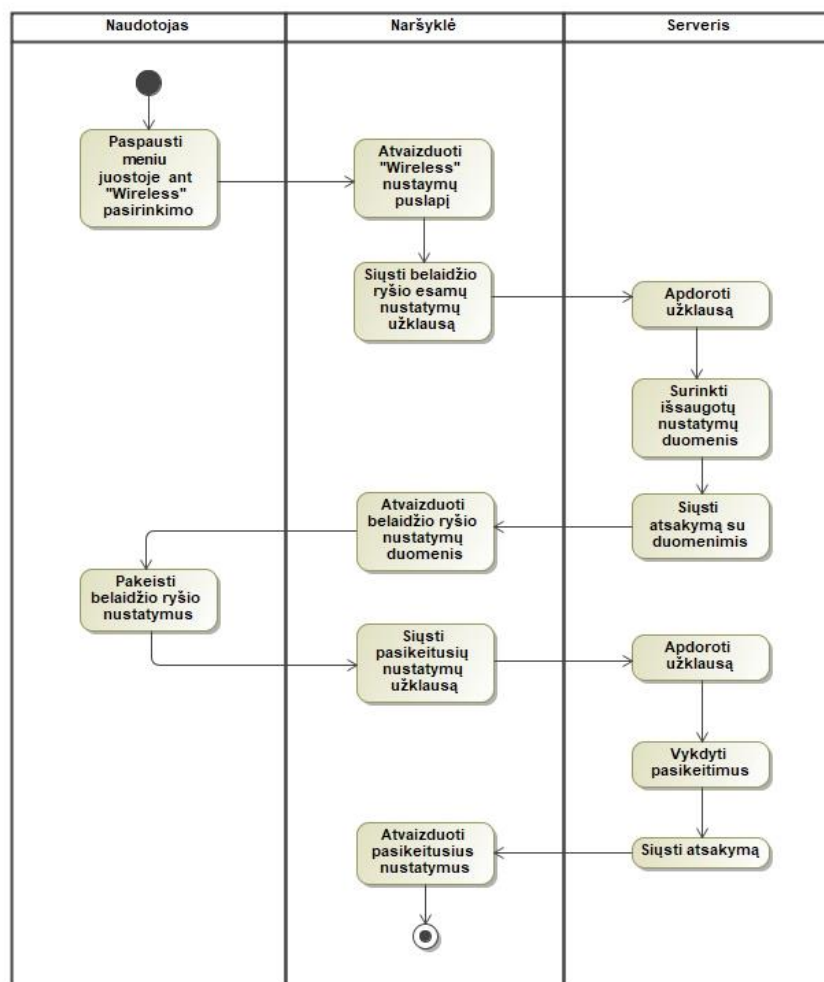


13 pav. PA 5 „Peržiūrėti belaidžio ryšio nustatymus“ veiklos diagrama

8 lentelėje pateikiama PA 6 „Valdyti belaidžio ryšio nustatymus“ specifikacija, o 14 pav. pateikiama „Valdyti belaidžio ryšio nustatymus“ veiklos diagrama, kurioje atvaizduojamas belaidžio ryšio nustatymų keitimo procesas. Naudotojas perėjęs į belaidžio ryšio nustatymų puslapį peržiūri esamus nustatymus ir pakeitęs nustatymus juos išsaugo. Naršyklė išsiunčia užklausą į serverį apie nustatymų išsaugojimą ir serveris atlikęs išsaugojimus atsako naršyklei, po kurio naršyklė atvaizduoja pasikeitusius nustatymus.

8 lentelė. PA 6 „Valdyti belaidžio ryšio nustatymus“ specifikacija

PA 6 Valdyti belaidžio ryšio nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti belaidžio ryšio nustatymus.	
<b>Aprašymas.</b> Belaidžio ryšio nustatymų puslapyje keičiami belaidžio ryšio nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Meniu juostoje paspaudžiamas pasirinkimas [Wireless].
<b>Po-sąlyga</b>	Belaidžio ryšio nustatymų puslapyje „Wireless“ atvaizduojami belaidžio ryšio konfigūracijos laukeliai su išsaugotomis reikšmėmis.

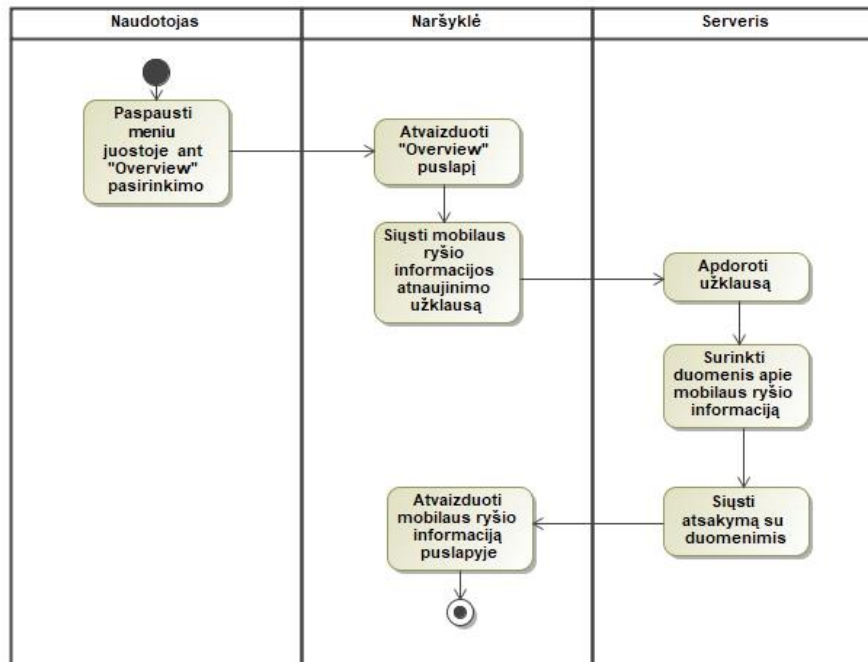


14 pav. PA 6 „Valdyti belaidžio ryšio nustatymus“ veiklos diagrama

9 lentelėje pateikiama PA 7 „Peržiūrėti mobilaus ryšio informaciją“ specifikacija, o 15 pav. pateikiama „Peržiūrėti mobilaus ryšio informaciją“ veiklos diagrama. Pagrindiniame IS „Overview“ puslapyje atvaizduojama mobilaus ryšio informacijos lentelė, kurioje pateikiamas mobilaus ryšio prisijungimo laikas, mobilaus ryšio būseną, SIM kortelės būseną, išsiųstų ir parsųstų duomenų kiekis.

9 lentelė. PA 7 „Peržiūrėti mobilaus ryšio informaciją“ specifikacija

PA 7 Peržiūrėti mobilaus ryšio informaciją	
<b>Tikslas/uždavinys.</b> Atvaizduoti mobilaus ryšio informaciją.	
<b>Aprašymas.</b> Informacinės sistemos pagrindiniame puslapyje atvaizduojama mobilaus ryšio informacija	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Overview].
<b>Po-sąlyga</b>	Pagrindiniame IS „Overview“ puslapyje atvaizduojama mobilaus ryšio informacijos lentelė.

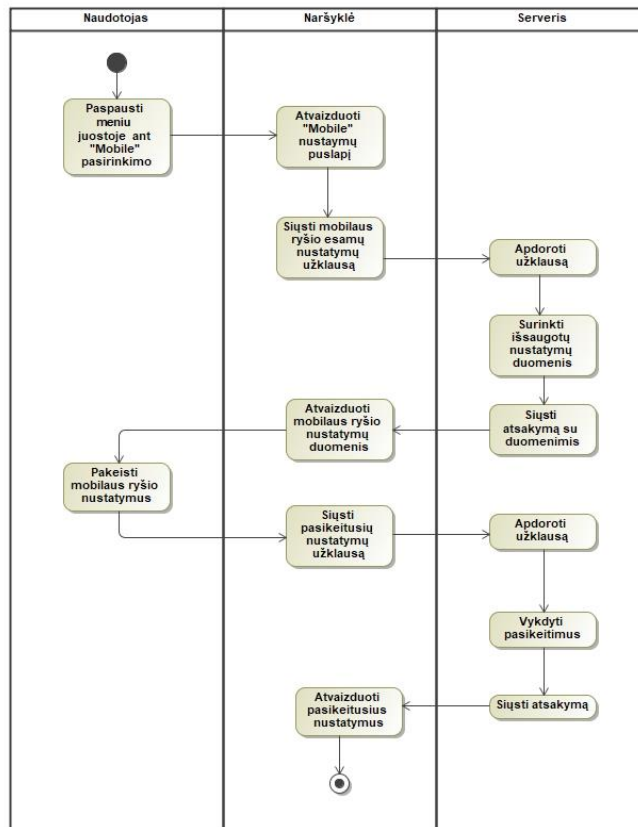


15 pav. PA 7 „Peržiūrėti mobilus ryšio būseną“ veiklos diagrama

10 lentelėje pateikia PA 8 „Valdyti mobilus ryšio nustatymus“ specifikacija, o 16 pav. pateikiama „Valdyti mobilus ryšio nustatymus“ veiklos diagrama, kurioje atvaizduojamas esamų mobilus ryšio nustatymų keitimas.

10 lentelė. PA 8 „Valdyti mobilus ryšio nustatymus“ specifikacija

PA 8 Valdyti mobilus ryšio nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti mobilus ryšio nustatymus.	
<b>Aprašymas.</b> Mobilus ryšio nustatymų puslapyje keičiami mobilus ryšio nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Mobile].
<b>Po-sąlyga</b>	Mobilus ryšio nustatymų puslapyje „Mobile“ atvaizduojami mobilus ryšio konfigūracijos laukeliai su išsaugotomis reikšmėmis.

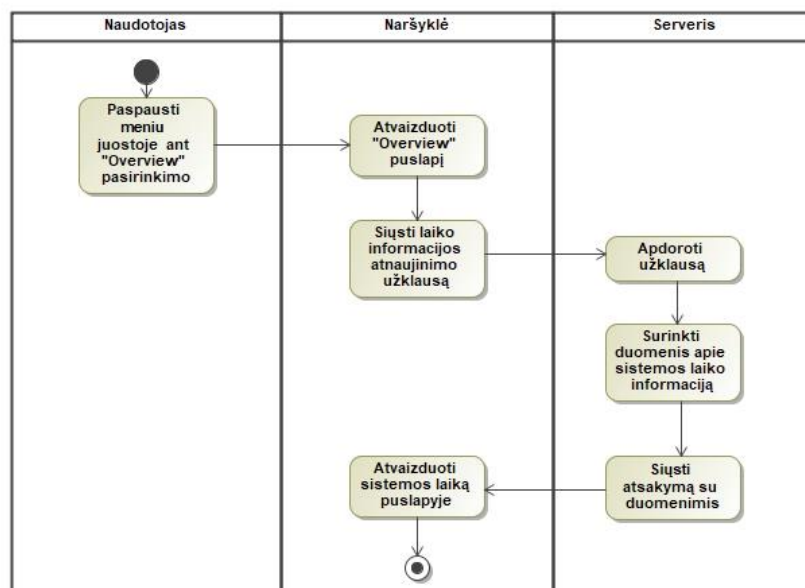


16 pav. PA 8 „Valdyti mobilaus ryšio nustatymus“ veiklos diagrama

11 lentelėje pateikiama PA 9 „Peržiūrėti sistemos laiką“ specifikacija, o 17 pav. pateikiama „Peržiūrėti sistemos laiką“ veiklos diagrama. Pagrindiniame IS „Overview“ puslapyje atvaizduojamas sistemos data ir laikas lentelėje.

11 lentelė. PA 9 „Peržiūrėti sistemos laiką“ specifikacija

PA 9 Peržiūrėti sistemos laiką	
<b>Tikslas/uždavinys.</b> Atvaizduoti sistemos laiką ir datą.	
<b>Aprašymas.</b> Informacinės sistemos pagrindiniame puslapyje atvaizduojama sistemos laiko informacija	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Overview].
<b>Po-sąlyga</b>	Pagrindiniame IS „Overview“ puslapyje atvaizduojama sistemos laiko informacijos lentelė.



17 pav. PA 9 „Peržiūrėti sistemos laiką“ veiklos diagrama

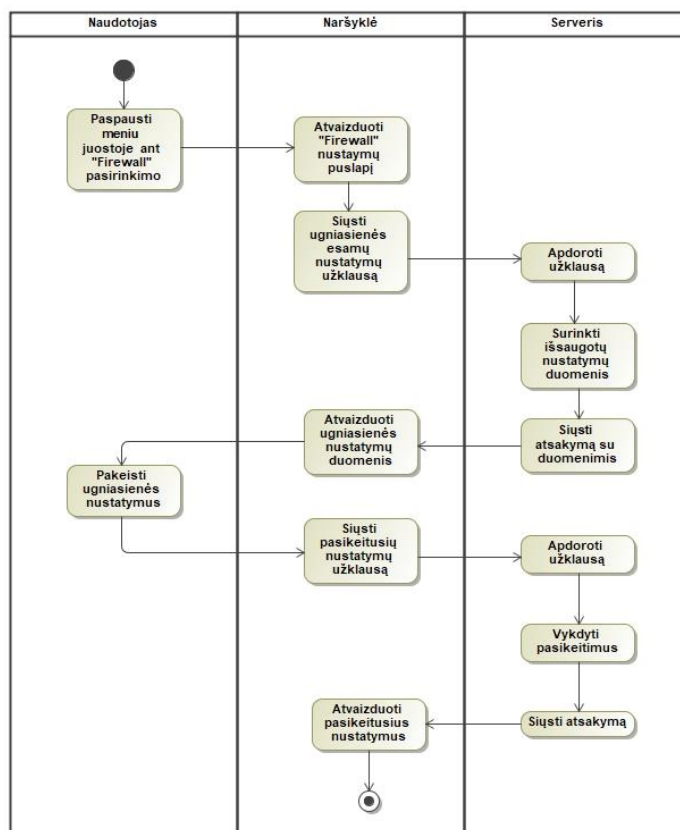
12 lentelėje pateikiama, PA 10 „Valdyti sistemos laiko nustatymus“ specifikacija, o 18 pav. pateikiama „Valdyti sistemos laiko nustatymus“ veiklos diagrama, kurioje pateikiamas sistemos laiko esamų nustatymų keitimas. Naršyklė bendraudama su serveriu atnaujiną ir atvaizduoja pasikeitusių nustatymus.

12 lentelė. PA 10 „Valdyti sistemos laiko nustatymus“ specifikacija

PA 10 Valdyti sistemos laiko nustatymus	
Tikslas/uždavinys. Valdyti sistemos laiko nustatymus	
Aprašymas. Sistemos laiko nustatymų puslapyje keičiami sistemos laiko nustatymai.	
Prieš-sąlyga	Baigiama veikla „Prisijungti“.
Aktorius	Naudotojas.
Sužadinimo sąlyga	Menu juostoje paspaudžiamas pasirinkimas [NTP].
Po-sąlyga	Sistemos laiko nustatymų puslapyje „NTP“ atvaizduojami sistemos laiko konfigūracijos laukeliai su išsaugotomis reikšmėmis.





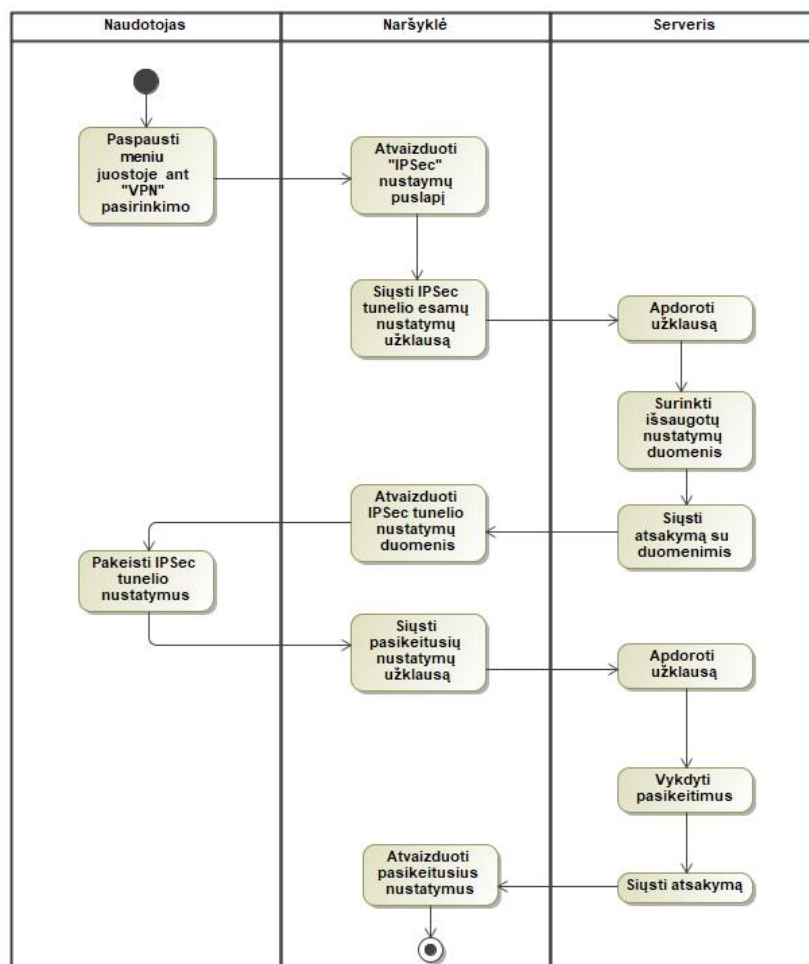


19 pav. PA 11 „Valdyti ugniasienės nustatymus“ veiklos diagrama

14 lentelėje pateikiama PA 12 „Valdyti IPSec tunelio nustatymus“ specifikacija, o 20 pav. pateikiama „Valdyti IPSec tunelio nustatymus“ veiklos diagrama, kurioje atvaizduojamas IPSec tunelio esamos konfigūracijos atvaizdavimas ir naudotojo nustatymų keitimas.

14 lentelė. PA 12 „Valdyti IPSec tunelio nustatymus“ specifikacija

PA 12 Valdyti IPSec tunelio nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti IPSec tunelio nustatymus.	
<b>Aprašymas.</b> IPSec tunelio nustatymų puslapyje keičiami IPSec tunelio nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [VPN].
<b>Po-sąlyga</b>	IPSec tunelio nustatymų puslapyje „VPN“ atvaizduojami IPSec tunelio konfigūracijos laukeliai su išsaugotomis reikšmėmis.

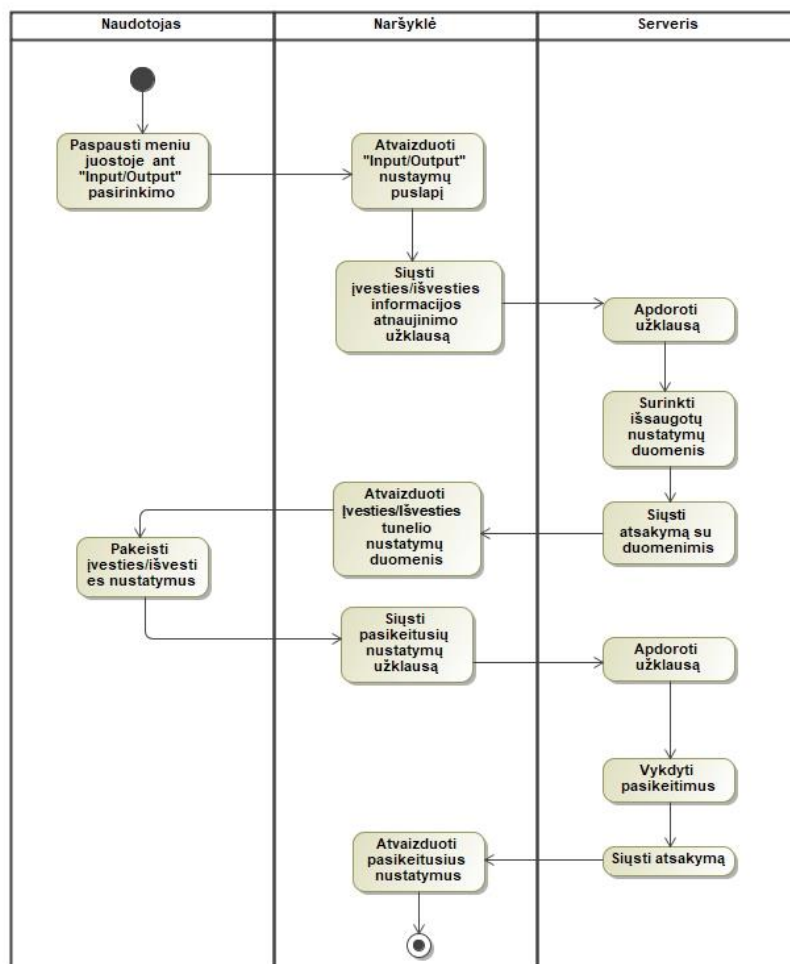


20 pav. PA 12 „Valdyti IPSec tunelio nustatymus“ veiklos diagrama

15 lentelėje pateikiama PA 13 „Valdyti įvesties/išvesties nustatymus“ specifikacija, o 21 pav. pateikiama „Valdyti įvesties/išvesties nustatymus“ veiklos diagrama, kurioje atvaizduojamas įvesties/išvesties konfigūracijos nustatymų keitimo procesas.

15 lentelė. PA 13 „Valdyti įvesties/išvesties“ specifikacija

PA 13 Valdyti įvesties/išvesties nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti įvesties/išvesties tunelio nustatymus.	
<b>Aprašymas.</b> Įvesties/išvesties nustatymų puslapyje keičiami IoT maršrutizatoriaus įvesties/išvesties nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Menu juostoje paspaudžiamas pasirinkimas [Input/Output].
<b>Po-sąlyga</b>	Įvesties/išvesties nustatymų puslapyje „Input/Output“ atvaizduojami įvesties/išvesties konfigūracijos laukeliai su išsaugotomis reikšmėmis.

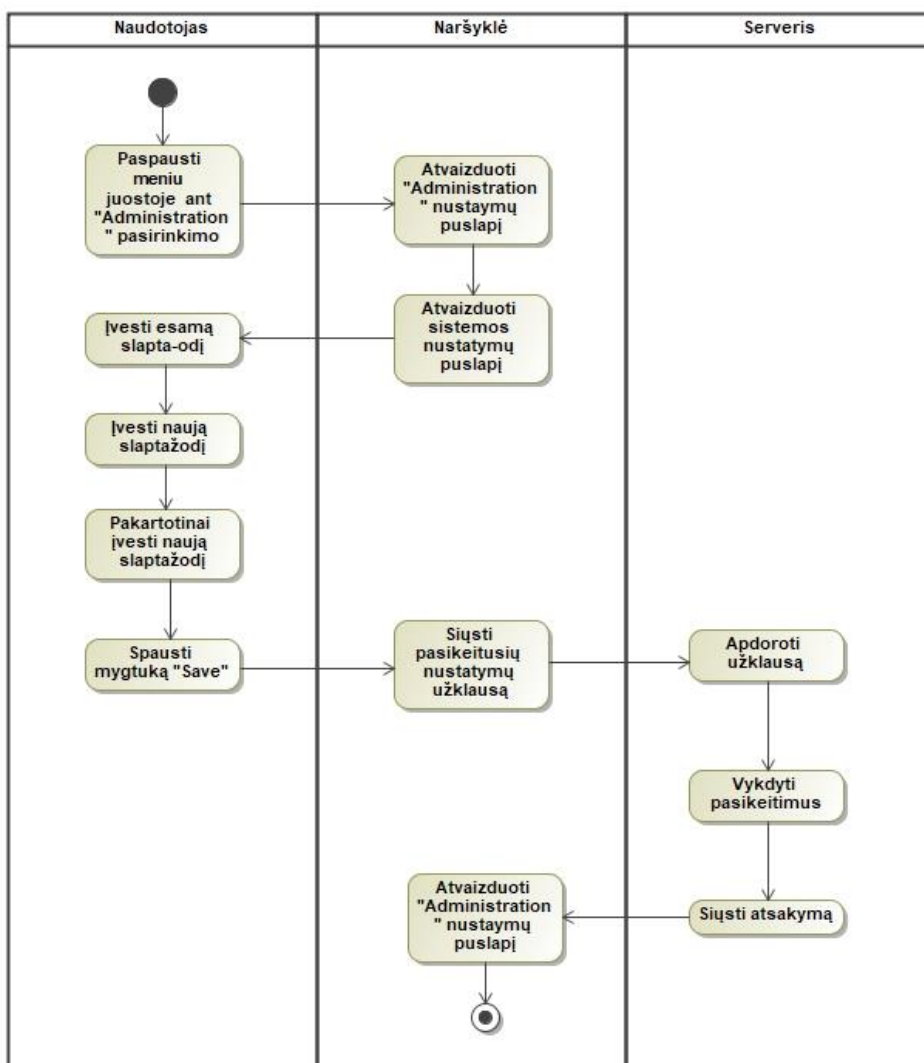


21 pav. PA 13 „Peržiūrėti įvesties/išvesties būsenas“ veiklos diagrama

16 lentelėje pateikiama PA 14 „Keisti slaptažodį“ specifikacija, o 22 pav. pateikiama „Keisti slaptažodį“ veiklos diagrama, kurioje pateikiamas sistemos slaptažodžio keitimo procesas, kuomet naudotojo įvesti duomenys yra apdorojami ir yra pakeičiamas sistemos slaptažodis.

16 lentelė. PA 14 „Keisti slaptažodį“ specifikacija

PA 14 Keisti slaptažodį	
<b>Tikslas/uždavinys.</b> Keisti sistemos prisijungimo slaptažodį.	
<b>Aprašymas.</b> Sistemos slaptažodžio keitimo puslapyje keičiamas sistemos slaptažodis.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Sistemos nustatymų puslapyje pakeičiamos nustatymų reikšmės ir paspaudžiamas mygtukas [Save].
<b>Po-sąlyga</b>	Sistemos nustatymų puslapyje „Administration“ atvaizduojami sistemos nustatymų konfigūracijos laukeliai su pakeistomis reikšmėmis.

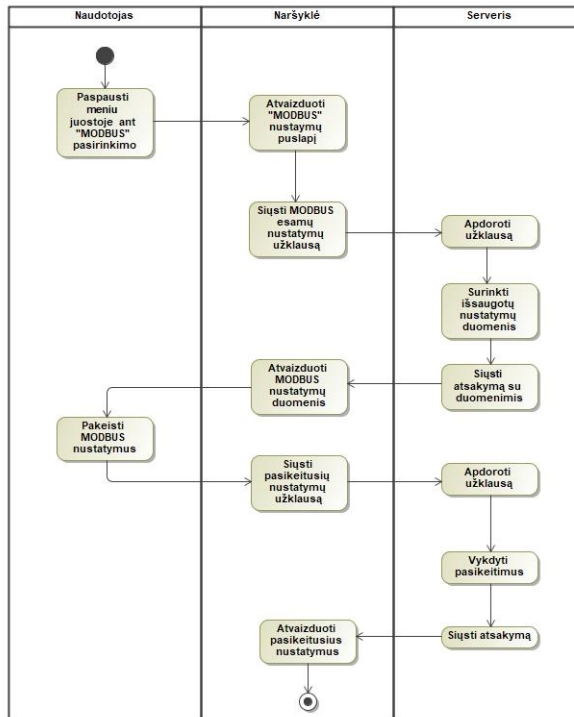


22 pav. PA 14 „Keisti slaptažodį“ veiklos diagrama

17 lentelėje pateikiama PA 15 „Valdyti MODBUS nustatymus“ specifikacija, o 23 pav. pateikiama „Valdyti MODBUS nustatymus“ veiklos diagrama, kurioje atvaizduojamas perėjimas į MODBUS nustatymų keitimo puslapį ir nustatymų keitimas.

17 lentelė. PA 15 „Valdyti MODBUS nustatymus“ specifikacija

PA 15 Valdyti MODBUS nustatymus	
<b>Tikslas/uždavinys.</b> Valdyti MODBUS nustatymus.	
<b>Aprašymas.</b> MODBUS nustatymų puslapyje keičiami MODBUS nustatymai.	
<b>Prieš-sąlyga</b>	Baigiama veikla „Prisijungti“.
<b>Aktorius</b>	Naudotojas.
<b>Sužadavimo sąlyga</b>	MODBUS puslapyje pakeičiamos nustatymų reikšmės ir paspaudžiamas mygtukas [Save].
<b>Po-sąlyga</b>	Sistemos nustatymų puslapyje „MODBUS“ atvaizduojami MODBUS nustatymų konfigūracijos laukeliai su pakeistomis reikšmėmis.

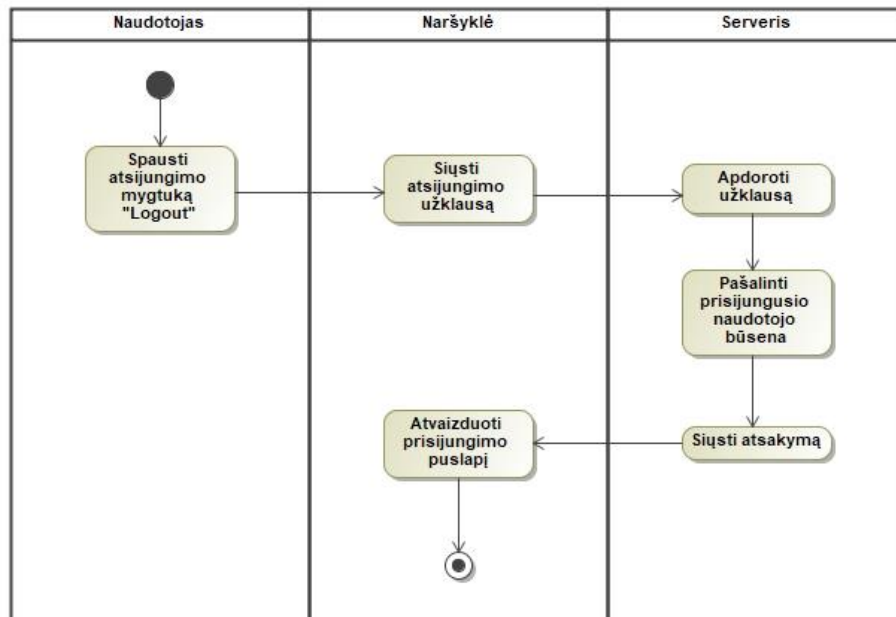


23 pav. PA 15 „Valdyti MODBUS nustatymus“ veiklos diagrama

18 lentelėje pateikiama PA 16 „Atsijungti“ specifikacija, o 24 pav. pateikiama „Atsijungti“ veiklos diagrama, kurioje atvaizduojamas atsijungimo nuo sistemos procesas.

18 lentelė. PA 16 „Atsijungti“ specifikacija

PA 16 Atsijungti	
Tikslas/uždavinys. Atsijungti nuo sistemos.	
Aprašymas. Naudotojas atsijungia nuo informacinės sistemos.	
Prieš-sąlyga	Baigiama veikla „Prisijungti“.
Aktorius	Naudotojas.
Sužadinimo sąlyga	Naudotojas pagrindiniame puslapyje spaudžia mygtuką [Logout].
Po-sąlyga	Matomas prisijungimo prie informacinės sistemos puslapis.

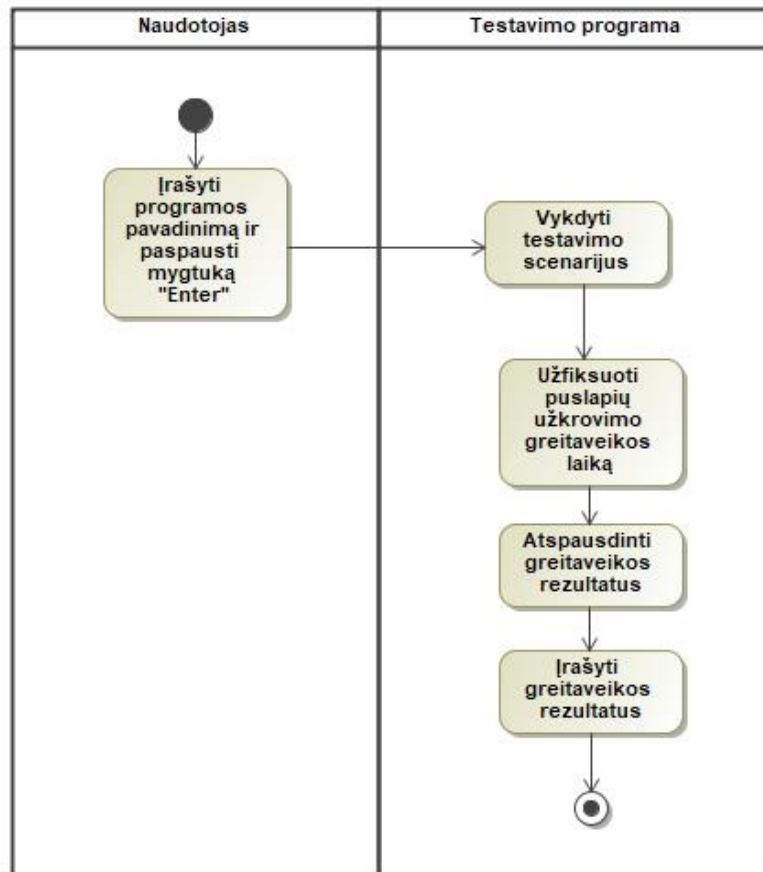


24 pav. PA 16 „Atsijungti“ veiklos diagrama

19 lentelėje pateikiama PA 17 „Įvykdyti testą“ specifikacija, o 24 pav. pateikiama „Įvykdyti testą“ veiklos diagrama, kurioje atvaizduojamas automatizuotos programos testavimo scenarijų įvykdymas.

19 lentelė. PA 17 „Įvykdyti testą“ specifikacija

PA 17 Įvykdyti testą	
<b>Tikslas/uždavinys.</b> Įvykdyti automatizuotos informacinės sistemos testavimo programos scenarijus.	
<b>Aprašymas.</b> Testuotojas paleidžia automatizuotą informacinės sistemos testavimo programą.	
<b>Prieš-sąlyga</b>	Komandinėje eilutėje įrašomas testavimo programos pavadinimas.
<b>Aktorius</b>	Testuotojas.
<b>Sužadinimo sąlyga</b>	Paspaudžiamas klaviatūros mygtukas [Enter].
<b>Po-sąlyga</b>	Automatizuota programa baigią darbą.



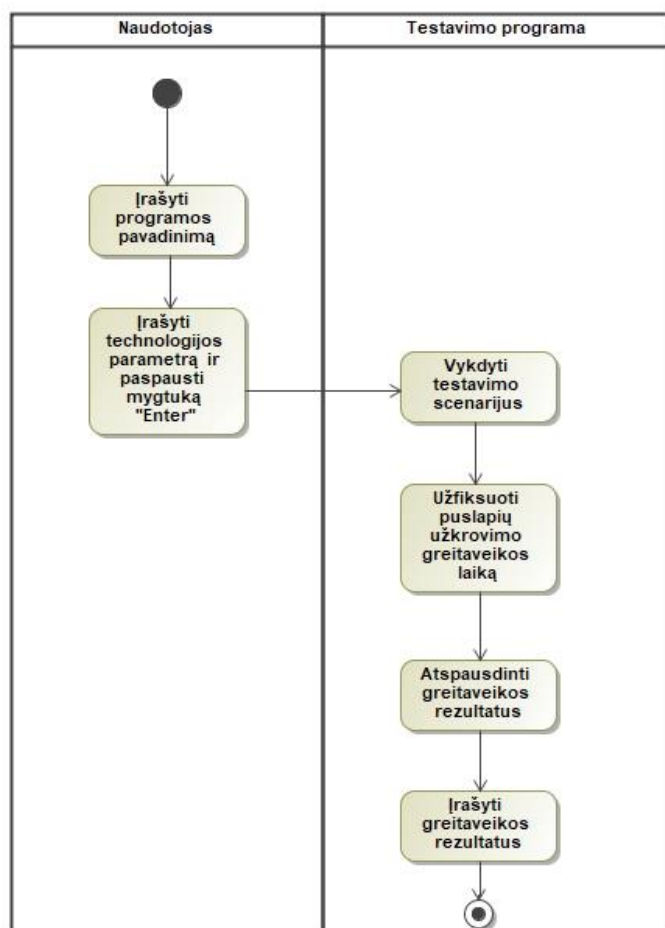
24 pav. PA 17 „Įvykdyti testą“ veiklos diagrama

20 lentelėje pateikiama PA 18 „Parinkti testuojamą technologiją“ specifikacija, o 25 pav. pateikiama „Parinkti testuojamą technologiją“ veiklos diagrama, kurioje atvaizduojamas testo paleidimo procesas, kuomet testas kreipiasi į naršyklę, bando pasiekti maršrutizatoriaus informacinę sistemą pagal nurodyti IP adresą, prisijungus atlieka testo programoje suprogramuotus nurodymus.

20 lentelė. PA 18 „Parinkti testuojamą technologiją“ specifikacija

PA 18 Pradėti testą	
<b>Tikslas/uždavinys.</b> Parinkti testuojamą technologiją, kuriai bus atliekamas automatizuotas informacinės sistemos scenarijų testavimas.	
<b>Aprašymas.</b> Testavimo programa priima technologijos parametą pagal kurį nustatomas IP adresas ir išsaugojamas testavimo scenarijų rezultatas duomenų faile.	
<b>Prieš-sąlyga</b>	Komandinėje eilutėje įrašomas testavimo programos pavadinimas.
<b>Aktorius</b>	Testuotojas.
<b>Sužadinimo sąlyga</b>	Paspaudžiamas klaviatūros mygtukas [Enter].
<b>Po-sąlyga</b>	Programa atspausdina testavimo rezultatus ir baigia darbą.



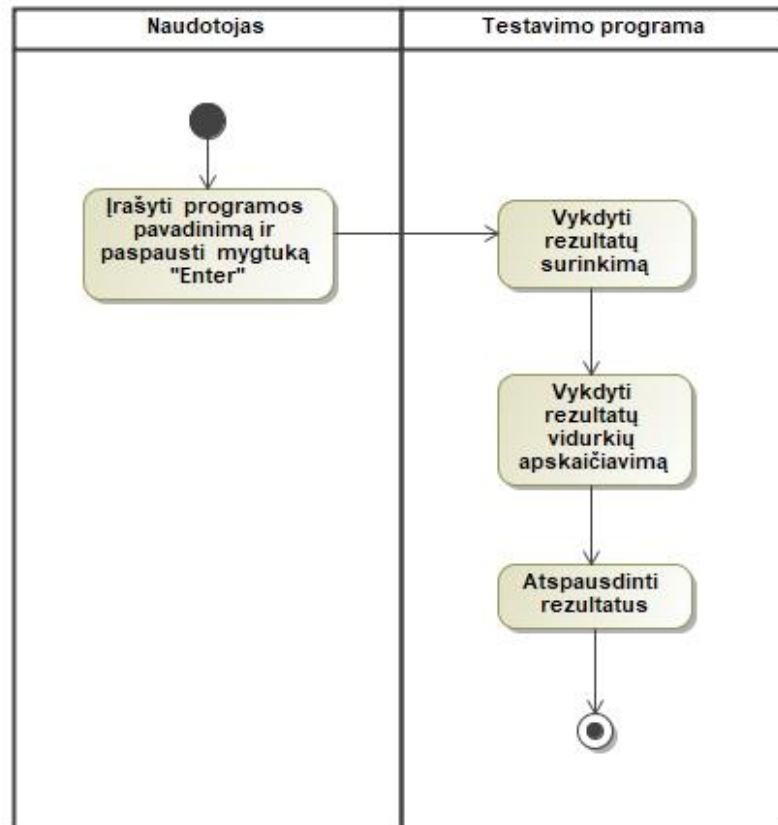


25 pav. PA 18 „Parinkti testuojamą technologiją“ veiklos diagrama

21 lentelėje pateikiama PA 19 „Generuoti testavimo ataskaitą“, specifikacija, o 26 pav. pateikiama „Generuoti testavimo ataskaitą“ veiklos diagrama, kurioje atvaizduojamas testavimo programos atliktų testavimo rezultatų vidurkių atspausdinimas komandinėje eilutėje.

21 lentelė. PA 19 „Generuoti testavimo ataskaitą“ specifikacija

PA 19 Generuoti testavimo ataskaitą	
<b>Tikslas/uždavinys.</b> Pateikti automatizuotos testavimo programos testavimo rezultatus.	
<b>Aprašymas.</b> Pateikiami automatinio testavimo rezultatai komandinėje eilutėje.	
<b>Prieš-sąlyga</b>	Komandinėje eilutėje įrašomas testavimo programos pavadinimas.
<b>Aktorius</b>	Testuotojas.
<b>Sužadavimo sąlyga</b>	Programa paleidžiama paspaudus [Enter] mygtuką.
<b>Po-sąlyga</b>	Komandinėje eilutėje atspausdinami rezultatų duomenys.



26 pav. PA 19 „Generuoti testavimo ataskaita“ veiklos diagrama

## 2.2. Nefunkcinių reikalavimų specifikacija

Bendrieji nefunkciniai reikalavimai IoT maršrutizatoriaus informacinėj sistemai:

1. Pagrindiniame IS puslapyje turi būti atvaizduojama maršrutizatoriaus resursų, mobilaus ryšio, belaidžio ryšio ir laiko informacija;
2. IS turi būti galima naudotis tik prisijungus prie sistemos;
3. IS sistemos stiliuje turi vyrauti pagrindinės baltos ir juodos spalvos;
4. IS turi būti anglų kalba;

Nefunkciniai reikalavimai tradicinių internetinių puslapių modeliui:

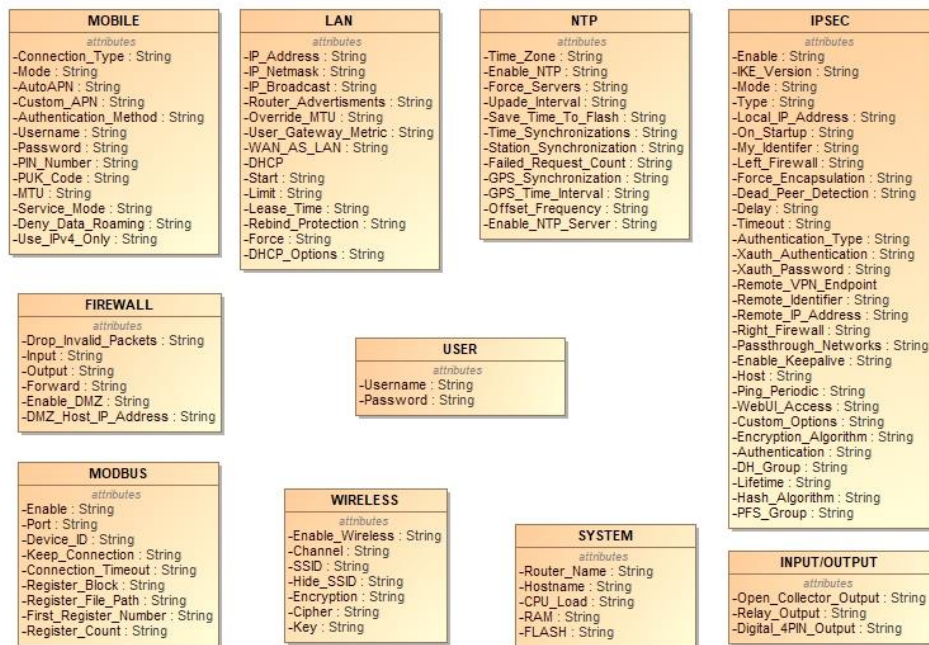
1. Kiekvieno puslapio atidarymo metu siunčiamas naujo puslapio duomenų failas iš serverio;
2. IS duomenų atnaujinimas atliekamas naudojantis AJAX technologija;
3. IS didžiąją dalį internetinių puslapių sudaro statiniai HTML failai;
4. IS turi veikti MVC programinės įrangos projektavimo modeliui;

Nefunkciniai reikalavimai vieno puslapio programos modeliui:

1. Naudoti mechanizmą paremtą vieno puslapio programos modeliui;
2. Pirmojo prisijungimo prie sistemos metu, serveris turi nusiųsti klientui visus reikiamus failus informacinės sistemos informacijai atvaizduoti;
3. Informacinės sistemos turinys turi atsinaujinti realiu laiku ir naudotojas neturi perkrauti internetinio puslapio, kad pamatytų atsinaujinusių informaciją;
4. Informacija tarp serverio ir kliento turi būti perduodama JSON formatu;
5. Internetinio puslapio elementai turi būti atnaujinami naudojantis virtualiu DOM;

### 2.3. Dalykinės srities modelis

Daiktų interneto maršrutizatoriaus informacinės sistemos esybės yra maršrutizatoriaus internetiniai puslapiai atspindintys maršrutizatoriaus funkcionalumus, todėl nei viena informacinės sistemos esybė neturi ryšio su kita esybe, nes maršrutizatoriaus funkcionalumais neturi nieko bendro vienas su kitu. Daiktų interneto maršrutizatoriaus informacinės sistemos esybių ryšių modelis pateiktas 27 pav.

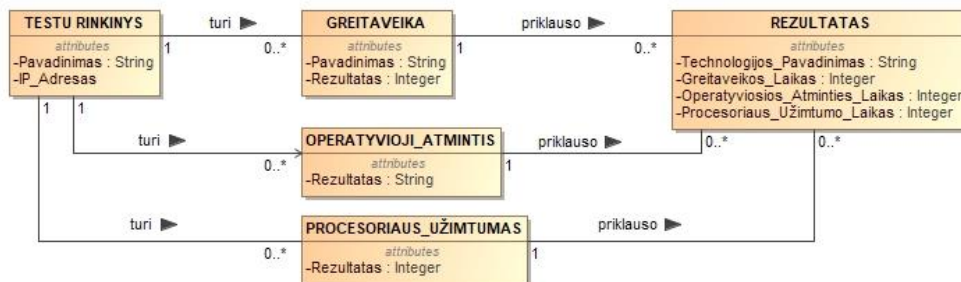


27 pav. Daiktų interneto maršrutizatoriaus informacinės sistemos esybių ryšių modelis

Kaip ir daiktų interneto informacinė sistema, taip ir automatizuota puslapių užkrovimo testavimo programa turi savo esybes. Esybes susideda iš:

- Testų rinkinio – kuriame yra nustatomas testuojamos technologijos pavadinimas ir IP adresas,
- Greitaveika – informacinės sistemos internetinių puslapių greitaveikos scenarijai;
- Operatyvioji atmintis – informacinės sistemos operatyviosios atminties rezultatas;
- Procesoriaus užimtumas – informacinės sistemos procesoriaus užimtumo rezultatas;

Automatizuota puslapių užkrovimo testavimo programa pateikta 28 pav.



28 pav. Automatizuotos puslapių užkrovimo testavimo programos esybių ryšių modelis

## 2.4. Naudotojų sąrašas modelis

Kadangi daiktų interneto maršrutizatoriaus informacinė sistema jau egzistuoja, todėl projektuojama nauja informacinė sistema turi būti identiška esamai informacinei sistemai. 29 pav. pateikiamas daiktų interneto maršrutizatoriaus informacinės sistemos pagrindinis puslapis „Overview“.

The screenshot displays the 'Overview' page of a Teltonika device. The top navigation bar includes 'Status', 'Network', 'Services', and 'System', with 'Logout' on the right. The main content is organized into several sections:

- System:** Shows 11.3% CPU load, Router uptime (13d 8h 52m 2s), Local device time (2022-04-17, 00:12:51), Memory usage (RAM: 23% used, FLASH: 22% used), and Firmware version (RUT9XX\_R\_GPL\_00.06.08.6).
- Mobile:** Shows 0 dBm signal strength, Data connection (Disconnected), State (Unregistered, N/A, NOSERVICE), SIM card slot in use (SIM 1 (not inserted)), and Bytes received/sent (0 B / 90.1 KB).
- Wireless:** Shows ON status, SSID (RUT950 (AP)), and Mode (1- AP; 4 CH (2.427 GHz)).
- WAN:** Shows Wired status, IP address (N/A), and WAN failover status (Failover link is enabled).
- Local Network:** Shows IP / netmask (192.168.1.2 / 255.255.0.0) and DHCP Leases (0).
- Remote Management System:** Shows OFF status, Status (Disabled), and Connection State (N/A).
- Recent System Events:** Lists four events related to Web UI authentication and Firewall configuration.
- Recent Network Events:** Lists four events related to Mobile data connection status.

Footnote: \* Your carrier's data usage accounting may differ. Teltonika is not liable should any accounting discrepancies occur.

Footer: Teltonika solutions | Wiki Teltonika | teltonika-networks.com

29 pav. Daiktų interneto maršrutizatoriaus informacinės sistemos pagrindinis puslapis

Prisijungimo prie informacinės sistemos puslapis atvaizduojamas 30 pav. daiktų interneto maršrutizatoriaus prisijungimo puslapis.

The screenshot shows the login page of the Teltonika device. The header is 'Authorization Required'. Below it, there is a prompt: 'Please enter your username and password.' The form contains two input fields: 'Username' with the value 'admin' and 'Password' with masked characters. A 'Login' button is positioned below the password field. The footer includes 'Teltonika solutions' and 'Wiki Teltonika | teltonika-networks.com'.

30 pav. Daiktų interneto maršrutizatoriaus prisijungimo puslapis

31 pav. pateikiamas informacinės sistemos ugniasienės nustatymų puslapis „Firewall“.

Profile in use: default FW version: RUT9XX\_R\_GPL\_00.06.08.6

**General Settings**

Drop invalid packets

Input

Output

Forward

**DMZ Configuration**

Enable

DMZ host IP address

Teltonika solutions Wiki Teltonika | teltonika-networks.com

**31 pav.** Informacinės sistemos ugniasienės nustatymų puslapis „Firewall“

32 pav. pateikiamas belaidžio ryšio puslapis „Wireless“

Profile in use: default FW version: RUT9XX\_R\_GPL\_00.06.08.6

**Wireless Access Point**

Here you can configure your wireless settings like radio frequency, mode, encryption etc...

**Device Configuration**

Enable wireless

Channel

**Interface Configuration**

General Setup **Wireless Security**

SSID

Hide SSID

Teltonika solutions Wiki Teltonika | teltonika-networks.com

**32 pav.** Informacinės sistemos belaidžio ryšio nustatymų puslapis „Wireless“

33 pav. pateikiamas IoT maršrutizatoriaus vidinio tinklo nustatymų puslapis „LAN“.

Profile in use: default FW version: RUT9XX\_R\_GPL\_00.06.08.6

### LAN

**Configuration**

General Setup **Advanced Settings**

IP address: 192.168.1.2  
 IP netmask: 255.255.0.0  
 IP broadcast:

**DHCP Server**

General Setup **Advanced Settings**

DHCP: Enable  
 Start: 100  
 Limit: 150  
 Lease time: 12 Hours

Save

Teltonika solutions Wiki Teltonika | teltonika-networks.com

33 pav. IoT maršrutizatoriaus informacinės sistemos vidinio tinklo nustatymų puslapis „LAN“

34 pav. pateikiamas IoT maršrutizatoriaus sistemos nustatymų puslapis.

Profile in use: default FW version: RUT9XX\_R\_GPL\_00.06.08.6

### Time Synchronization

**General**

Current system time: 2022-04-17 00:15:05  
 Time zone: Europe/Vilnius  
 Enable NTP:   
 Force servers:   
 Update interval (in seconds): 3660  
 Save time to flash:   
 Count of time synchronizations: 10  
 Operator Station Synchronization:   
 Count of failed NTP requests: 5

**Clock Adjustment**

Offset frequency: 1

**NTP Server**

**General**

Enable:

Save

Teltonika solutions Wiki Teltonika | teltonika-networks.com

34 pav. IoT maršrutizatoriaus informacinės sistemos laiko nustatymų puslapis „NTP“

35 pav. pateikiamas IPSec konfigūracijos nustatymų keitimo puslapis.

Teltonika solutions Wiki Teltonika | teltonika-networks.com

35 pav. IoT maršrutizatoriaus informacinės sistemos virtualaus tinklo nustatymų puslapis „IPSec“

36 pav. pateikiamas įvesties/išvesties konfigūracijos nustatymų keitimo puslapis.

Teltonika solutions Wiki Teltonika | teltonika-networks.com

36 pav. IoT maršrutizatoriaus informacinės įvesties/išvesties nustatymų puslapis „Input/Output“

37 pav. pateikiamas MODBUS konfigūracijos nustatymų keitimo puslapis.

Teltonika solutions | Wiki Teltonika | teltonika-networks.com

37 pav. IoT maršrutizatoriaus informacinės sistemos nustatymų puslapis „MODBUS“

38 pav. pateikiamas sistemos pagrindinių nustatymų puslapis „Administration“.

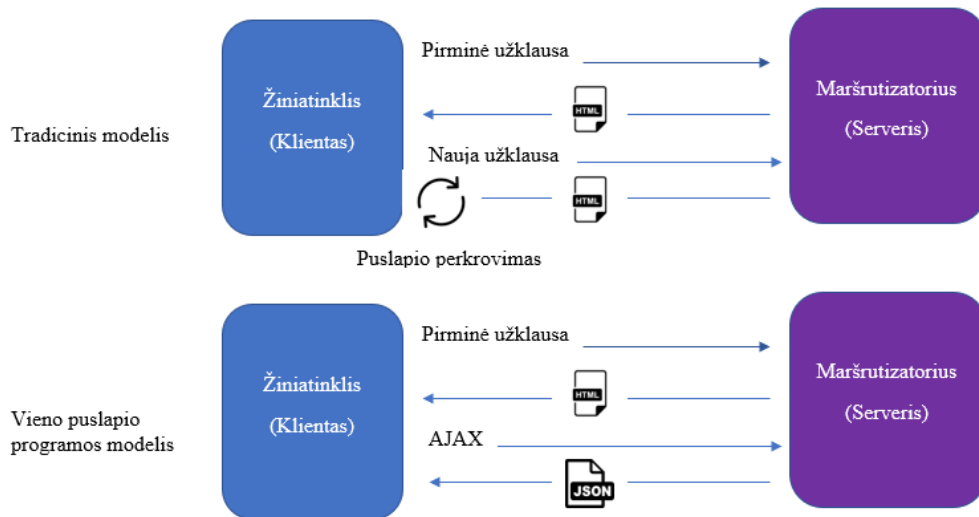
Teltonika solutions | Wiki Teltonika | teltonika-networks.com

38 pav. IoT maršrutizatoriaus pagrindinių nustatymų puslapis „Administration“

## 2.5. Formalizuotas sprendimo aprašas

Siekiamas sprendimas – daiktų interneto maršrutizatoriaus informacinės sistemos puslapių užkrovimo laiko sutrumpinimas ir naudojamų resursų sumažinimas. Sprendimui pasiekti parinktas internetinių puslapių veikimo modelis – vieno puslapio programos modelis. Šio modelio veikimo skirtumas lyginant su tradiciniu modeliu pateikiamas 39 pav.





**39 pav.** Tradicinio ir vieno puslapio programos modelių veikimo skirtumai

Pagal internetinių puslapių veikimo modelius matoma, kad pirmiausia yra vykdoma pirminė užklausa į serverį, kuris atsakydamas siunčia HTML failą internetinio puslapio atvaizdavimui naršyklėje. Tradicinio modelio atveju puslapis yra atvaizduojamas toks koks buvo sugeneruotas serverio dalyje ir norint patekti į kitą puslapį ar puslapį perkrauti yra vėl siunčiama užklausa į serverį naujo puslapio generavimui. Tradicinis modelis pasižymi veikimu, kuris reikalauja kiekvieną naują puslapio pasikeitimą atnaujinti su nauju serverio sugeneruotu HTML failu. HTML failo generavimas serveryje reikalauja resursų ir nuo serverio galingumo priklauso internetinio puslapio greitaveika. HTML failo generavimo metu yra sugeneruojami atvaizdavimo elementai su konkrečių duomenų reikšmėmis.

Vieno puslapio programos modelio atveju, serveris persiunčia HTML failą, kuriame yra visos internetinės svetainės atvaizdavimo programinis kodas. Konkrečių duomenų reikšmės yra atnaujinamos asinchroninės JavaScript (angl. AJAX) technologijos pagalba. AJAX technologijos pagalba į serverį yra išsiunčiama užklausa duomenims gauti. Serveris atsakydamas į užklausą ir surinkęs konkrečius duomenis persiunčia surinktus duomenis nustatytu formatu (pvz. JSON arba XML). Naudojant tokį veikimo principą, persiunčiamų duomenų dydis yra ženkliai sumažinamas, dėl to internetinio puslapio greitaveika ženkliai sumažėja kartu sumažindama ir serverio darbą.

Vieno puslapio programos modelis ypač atsiskleidžia žemų aparatinės įrangos parametrų serveriuose, kuriems puslapių generavimas reikalauja pakankamai daug resursų. Šio modelio informacinės sistemos atvaizdavimo generavimas vyksta naršyklėje, o šiuolaikiniai kompiuteriai yra pakankamai galingi apdoroti didelius kiekius HTML ir JavaScript programinio kodo eilučių duomenų turiniui atvaizduoti lyginant su daiktų interneto maršrutizatoriais. Vieno puslapio programos modelio panaudojimas daiktų interneto maršrutizatoriui veikiančiam 550 MHz procesoriaus greičiu, turinčiam 16 megabaitų pastoviosios ir 128 megabaitus operatyviosios atminties ypač palengvins internetinės svetainės palaikymo darbą.

Siekiamam sprendimui patvirtinti panaudojant automatizuotos testavimo programos rezultatus bus apskaičiuotas aritmetinis vidurkis ir vidurkių mediana. Tikslas bus laikomas pasiektu, jei vieno puslapio programos modelio informacinės sistemos atliks testavimo scenarijaus greičiau, nei tradicinio modelio informacinė sistema. Aritmetinio vidurkio ir medianos rezultatai leis tiksliau nustatyti skirtumus tarp informacinių sistemų ir pateikti tikslesnius rezultatus. Aritmetinis vidurkis

yra konkrečių reikšmių suma padalyta iš reikšmių skaičiaus. Aritmetinio vidurkio matematinė formulė pateikiama 40 pav.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

**40 pav.** Aritmetinio vidurkio matematinė formulė

Pagal 40 pav. pateiktą aritmetinio vidurkio formulę  $\bar{x}$  - aritmetinio vidurkio rezultatas,  $n$  – rezultatų skaičius,  $\sum_{i=1}^n x_i$  – suma visų testavimo reikšmių. Mediana – yra imties variacinės eilutės vidurinis skaičius, kai imties dydis yra nelyginis skaičius, arba dviejų vidurinių skaičių aritmetinis vidurkis, kai imties dydis yra lyginis skaičius. Medianos matematinė formulė pateikiama 41 pav.

$$\begin{cases} X_{\left(\frac{n+1}{2}\right)}, & \text{kai } n - \text{nelyginis} \\ X_{\left(\frac{n}{2}\right)} + X_{\left(\left(\frac{n}{2}\right)+1\right)}, & \text{kai } n - \text{lyginis} \end{cases}$$

**41 pav.** Medianos matematinė formulė

Atsižvelgus į aritmetinio vidurkio ir medianos apskaičiuotus rezultatus bus atliktas palyginimas tarp tradicinio modelio ir vieno puslapio programos modelio informacinių sistemų. Tikslas bus laikomas pasiektu, jei vieno puslapio programos modelio informacinių sistemų rezultatai bus mažesni nei tradicinio modelio paremtos informacinės sistemos testavimo rezultatų.

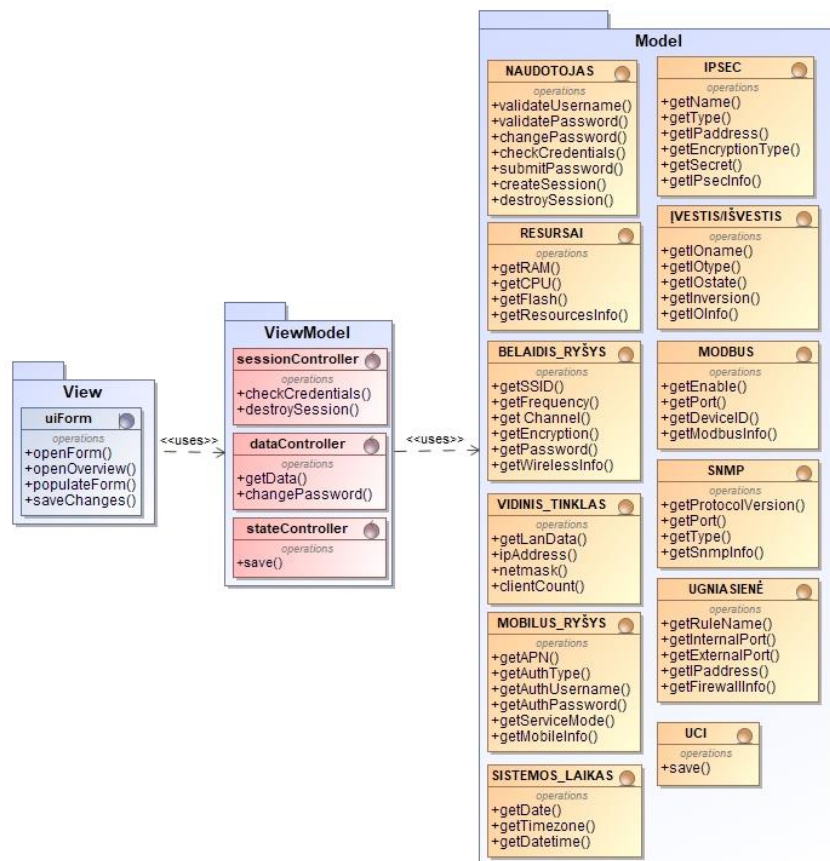
## **2.6. Reikalavimų apibendrinimas ir išvados**

1. Atsižvelgiant į atliktą analizės dalį, reikalavimai informacinėj sistemai buvo išskaidyti į dvi dalis, daiktų interneto maršrutizatoriaus informacinės sistemos reikalavimai ir automatizuotos informacinės sistemos testavimo programos reikalavimai;
2. Daiktų interneto maršrutizatoriaus informacinėj sistemai iškelta 16 funkcinių reikalavimų, kurie nurodo kokias funkcijas turi atitikti daiktų interneto maršrutizatoriaus informacinė sistema;
3. Automatizuotos informacinės sistemos testavimo programai iškelta 3 funkciniai reikalavimai, kurie apibrėžia testavimo programos veikimo funkcijas.
4. Panaudojus automatizuotos informacinės sistemos testavimo programos rezultatus apskaičiuojant aritmetinį vidurkį ir medianą, bus galima atlikti išvadas ir pateikti rekomendacijas daiktų interneto maršrutizatoriaus sistemai.

### 3. Grafinės sąsajos optimizavimo metodų daiktų interneto įrenginiuose sprendimo realizacijos projektas

#### 3.1. Daiktų interneto maršrutizatoriaus sistemos loginė architektūra

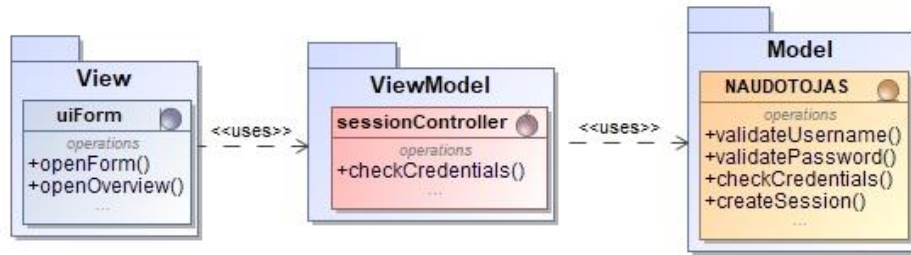
Daiktų interneto maršrutizatoriaus informacinė sistema kurta remiantis View – ViewModel - Model loginiu principu naudojamų vieno puslapio programos internetiniams puslapiams kurti. 42 pav. pateikiama sistemos loginė architektūra.



42 pav. Daiktų interneto maršrutizatoriaus loginės architektūros diagrama

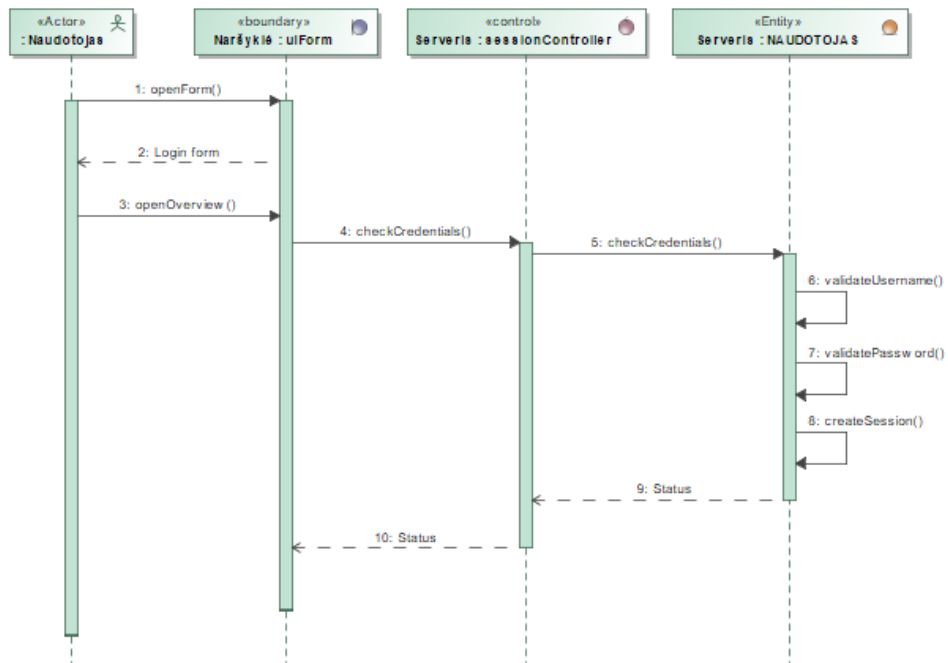
#### 3.2. Sistemos elgsenos modelis

Sekų diagramos projektuojamos kiekvienam daiktų interneto maršrutizatoriaus informacinės sistemos ir automatizuotos testavimo programos panaudojimo atvejui. Sekų diagramos projektavimas prasideda nuo panaudojimo atvejų loginių diagramų, pagal kurias toliau projektuojamos sekų diagramos. 43 pav. pateikiama PA 1 „Prisijungti“ panaudojimo atvejo loginė diagrama.



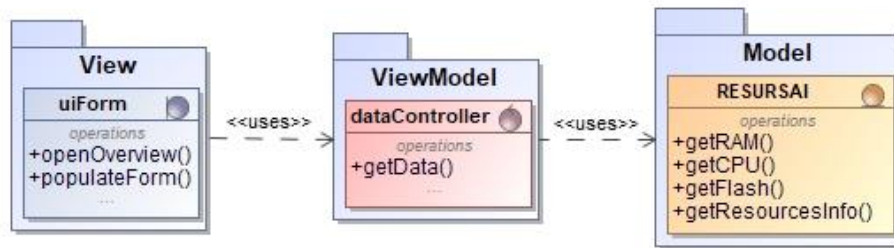
43 pav. PA 1 „Prisijungti“ loginės architektūros diagrama

44 pav. PA 1 „Prisijungti“ pateikiama sekų diagrama, kurioje pateikiamas naudojo prisijungimas prie informacinės sistemos. Pirmiausia yra atvaizduojamas prisijungimo puslapis, vėliau naudotoji suvedus prisijungimo vardą ir slaptažodį šie yra sutikrinami ir atitinkamai grąžinamas atsakymas iš serverio.



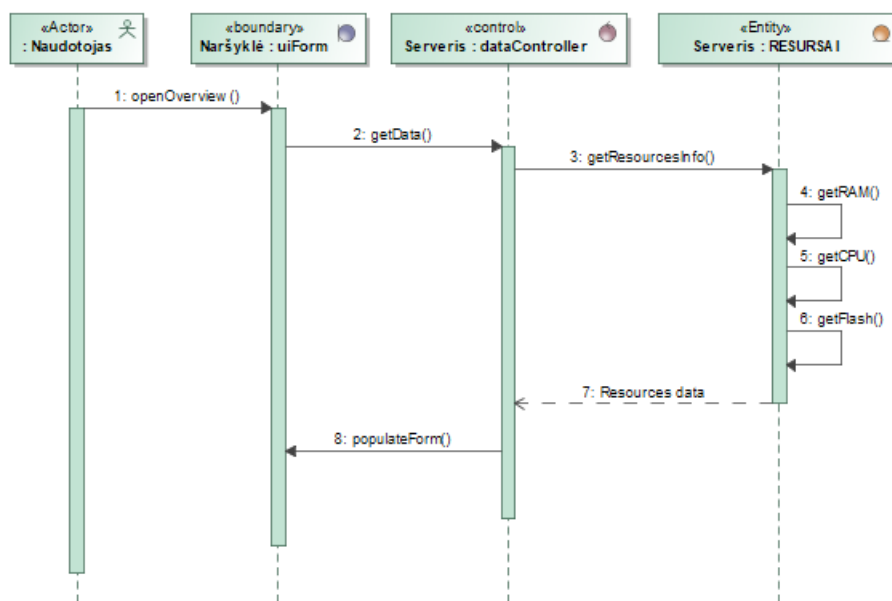
44 pav. PA 1 „Prisijungti“ sekų diagrama

45 pav. pateikiama PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ loginė diagrama, kurioje atsispindi sistemos naudojamų resursų informacijos užkrovimo ir atvaizdavimo loginė architektūra.



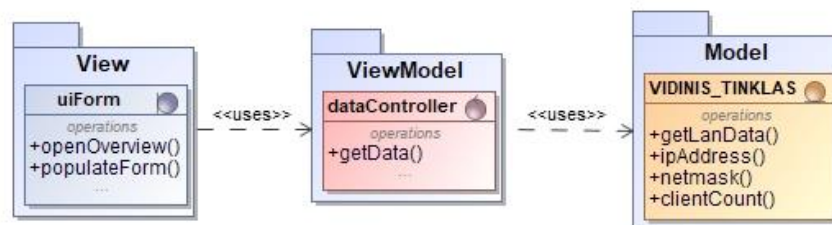
45 pav. PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ loginės architektūros diagrama

46 pav. pateikiama PA 2 „Peržiūrėti sistemos naudojamų resursų informacija“ sekų diagrama, kurioje atvaizduojamas nukreipimas į pagrindinį sistemos puslapį, po kurio seka duomenų atnaujinimo užklausa serveriui ir serverio surinktų duomenų atsakymas.



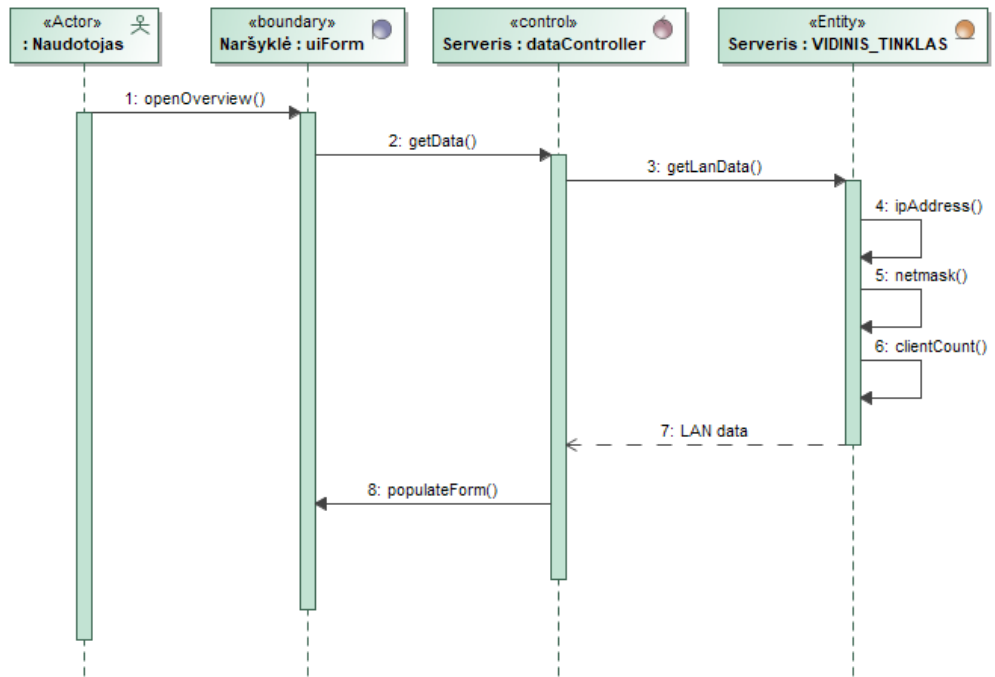
46 pav. PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“

47 pav. pateikiama PA 3 „Peržiūrėti vidinio tinklo būseną“ loginė diagrama, kurioje atsispindi vidinio tinklo informacijos loginė architektūra.



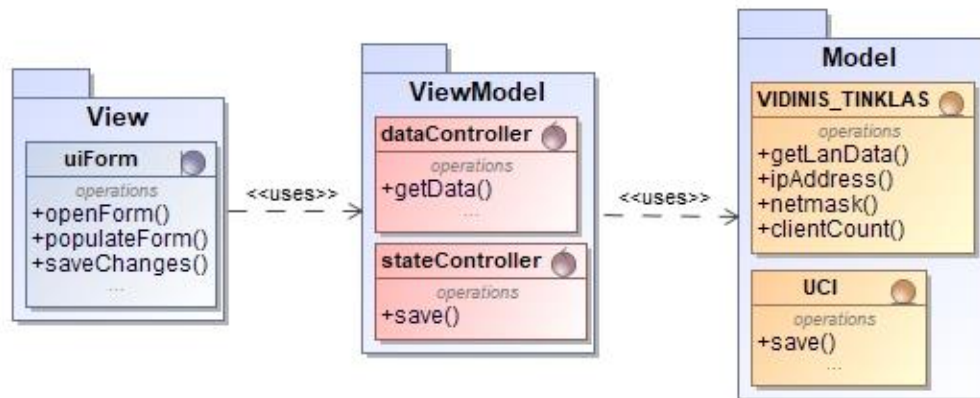
47 pav. PA 3 „ Peržiūrėti vidinio tinklo būseną“ loginės architektūros diagrama

48 pav. pateikiama PA 3 „Peržiūrėti vidinio tinklo būseną“ sekų diagrama, kurioje atvaizduojamas vidinio tinklo informacijos atvaizdavimas pagrindiniame sistemos puslapyje pagal duomenis gautus iš serverio.



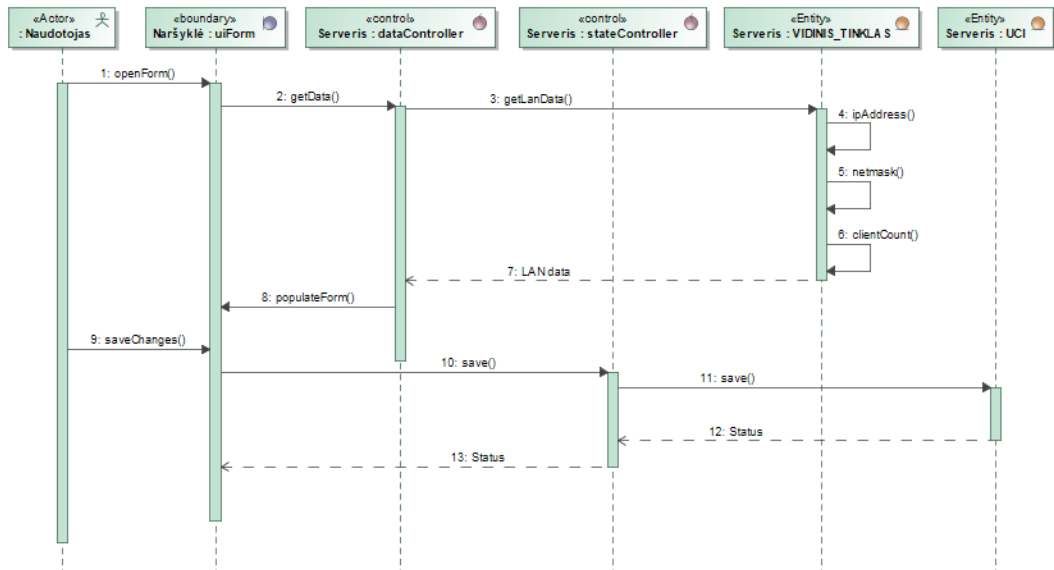
48 pav. PA 3 „Peržiūrėti vidinio tinklo būseną“ sekų diagrama

49 pav. pateikiama PA 4 „Valdyti vidinio tinklo nustatymus“ loginė diagrama, kurioje atvaizduojama vidinio tinklo nustatymų keitimo loginė architektūra.



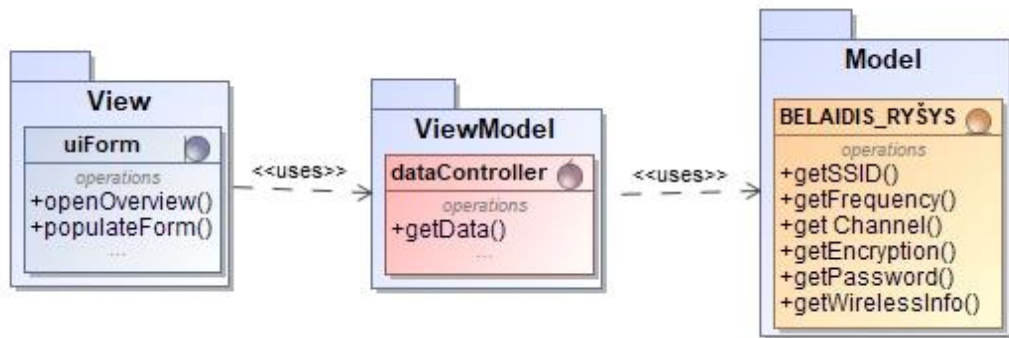
49 pav. PA 4 „Valdyti vidinio tinklo nustatymus“ loginės architektūros diagrama

50 pav. pateikiama PA 4 „Valdyti vidinio tinklo nustatymus“ sekų diagrama, kurio atvaizduojamas vidinio tinklo nustatymų puslapių užkrovimas, duomenų atnaujinimas ir konfigūracijos išsaugojimas.



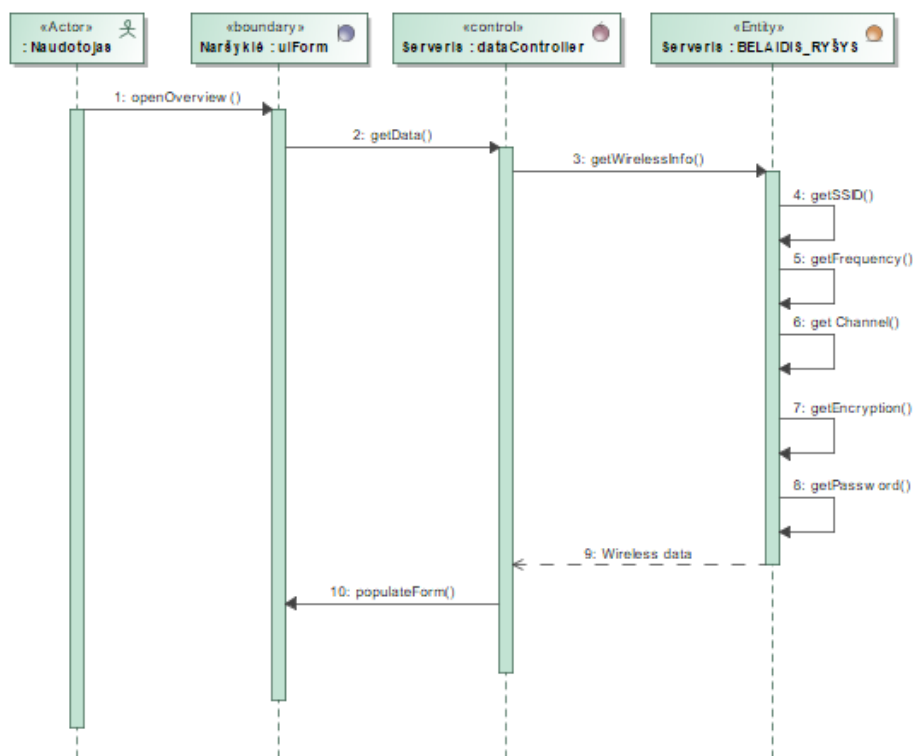
50 pav. PA 4 „Valdyti vidinio tinklo nustatymus“ sekų diagrama

51 pav. pateikiama PA 5 „Peržiūrėti belaidžio ryšio būseną“ loginė architektūros diagrama, kurioje atvaizduojamas belaidžio ryšio informacijos atvaizdavimas.



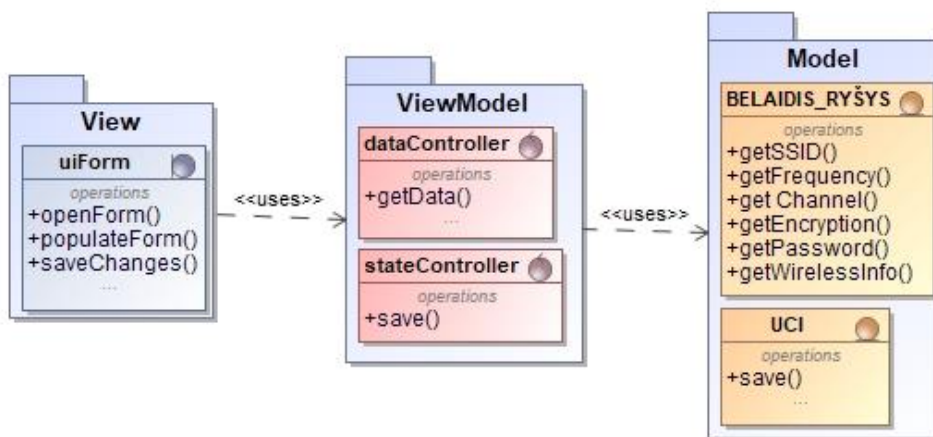
51 pav. PA 5 „Peržiūrėti belaidžio ryšio būseną“ loginės realizacijos klasių diagrama

52 pav. pateikiama PA 5 „Peržiūrėti belaidžio ryšio būseną“ sekų diagrama, kurioje atvaizduojamas belaidžio ryšio informacijos atvaizdavimas pagrindiniame puslapyje.



52 pav. PA 5 „Peržiūrėti belaidžio ryšio informaciją“ loginės realizacijos sekų diagrama

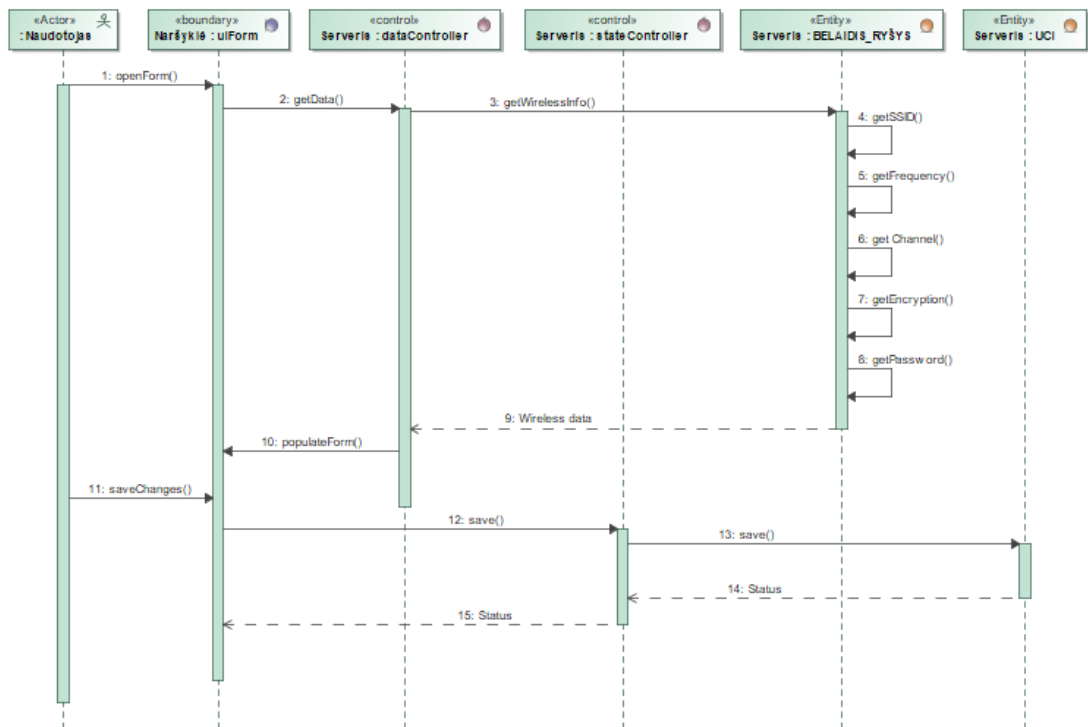
53 pav. pateikiama PA 6 „Valdyti belaidžio ryšio nustatymus“ loginė architektūros diagrama, kurioje atvaizduojamas belaidžio ryšio nustatymų keitimas.



53 pav. PA 6 „Valdyti belaidžio ryšio informaciją“ loginė architektūros diagrama

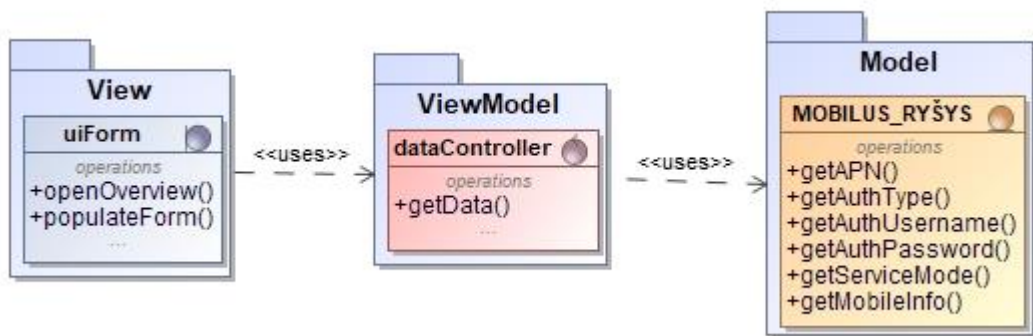
54 pav. pateikiama PA 6 „Valdyti belaidžio ryšio informaciją“ sekų diagrama, kurioje atvaizduojamas belaidžio ryšio nustatymų puslapio atvaizdavimas, esamos konfigūracijos užkrovimas ir naujų nustatymų išsaugojimas.





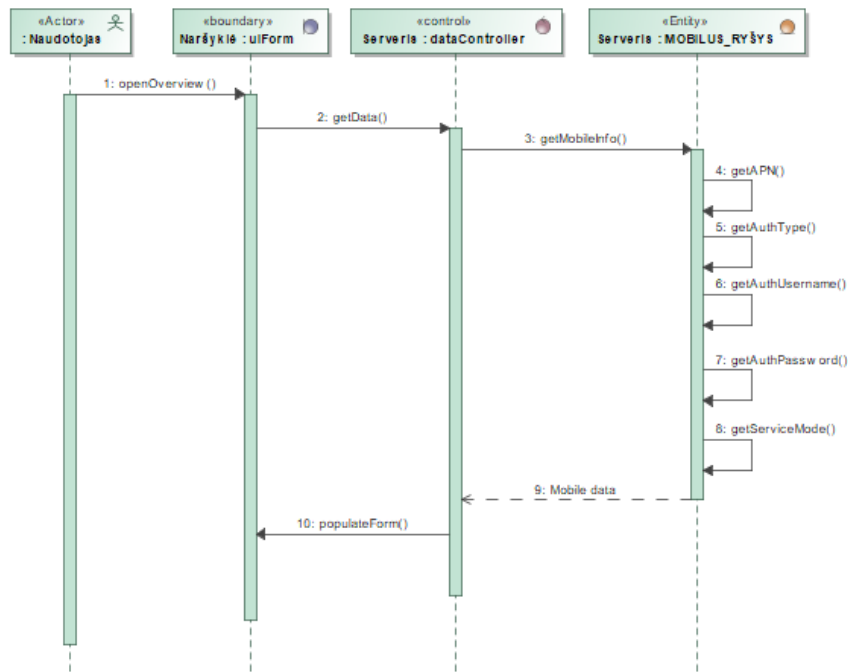
54 pav. PA 6 „Valdyti belaidžio ryšio nustatymus“ sekų diagrama

55 pav. pateikiama PA 7 „Peržiūrėti mobilaus ryšio būseną“ loginė architektūros diagrama, kurioje atvaizduojama mobilaus ryšio būsenos atvaizdavimo loginė architektūra.



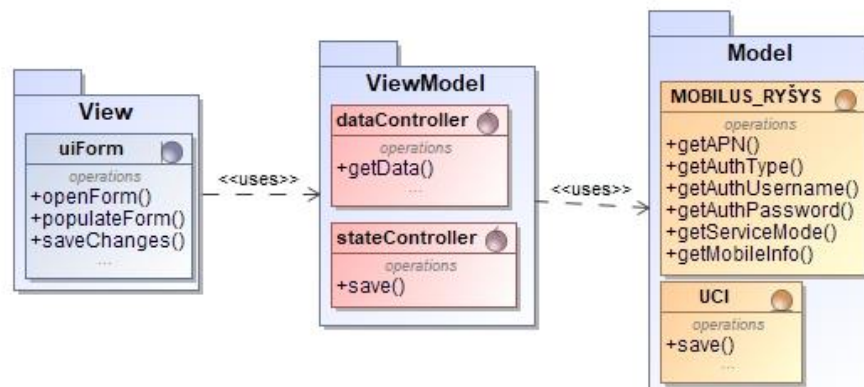
55 pav. PA 7 „Peržiūrėti mobilaus ryšio būseną“ loginė architektūros diagrama

56 pav. pateikiama PA 7 „Peržiūrėti ryšio būseną“ sekų diagrama, kurioje atvaizduojama mobilaus ryšio būsenų peržiūra pagrindiniame sistemos puslapyje.



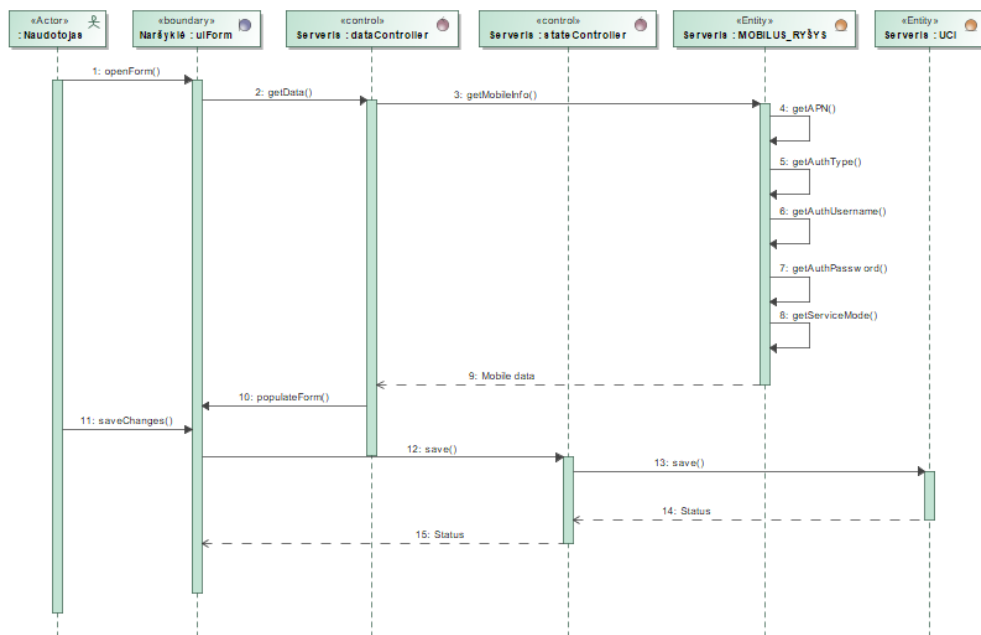
56 pav. PA 7 „Peržiūrėti mobilus ryšio būseną“ sekų diagrama

57 pav. pateikiama PA 8 „Valdyti mobilus ryšio nustatymus“ loginė architektūros diagrama, kurioje atvaizduojamas mobilus ryšio nustatymų atvaizdavimas ir keitimas.



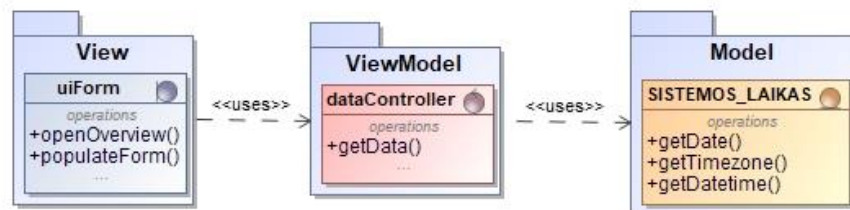
57 pav. PA 8 „Valdyti mobilus ryšio nustatymus“ loginė architektūros diagrama

58 pav. pateikiama PA 8 „Valdyti mobilus ryšio nustatymus“ sekų diagrama, kurioje atvaizduojamas mobilus ryšio esamos konfigūracijos atvaizdavimas ir naujų nustatymų išsaugojimas.



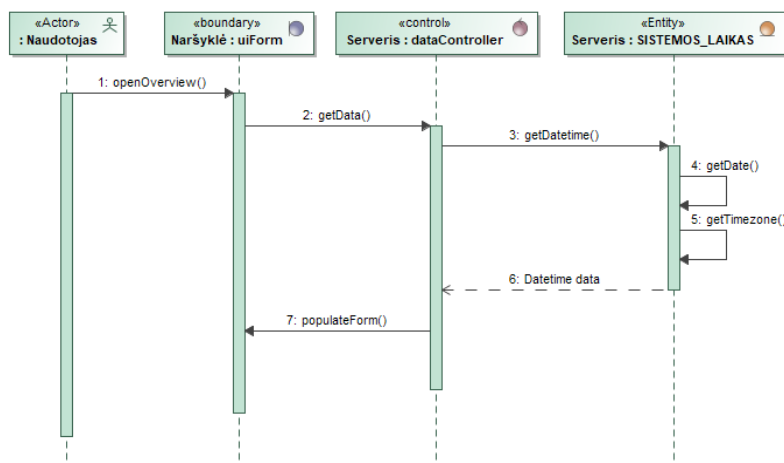
58 pav. PA 8 „Valdyti mobilaus ryšio nustatymus“ sekų diagrama

59 pav. pateikiama PA 9 „Peržiūrėti sistemos laiką“ loginė architektūros diagrama, kurioje atvaizduojamas sistemos laiko atvaizdavimas.



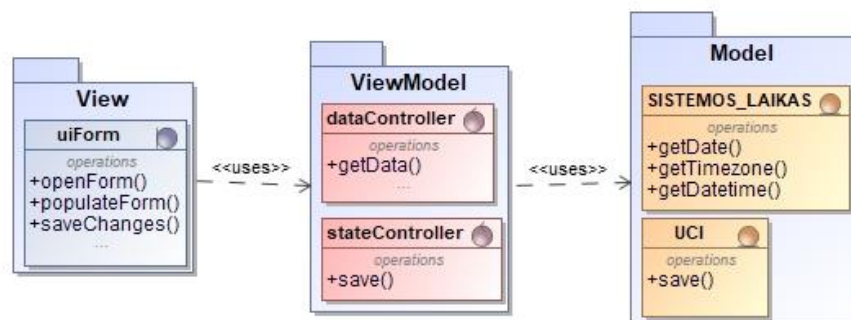
59 pav. PA 9 „Peržiūrėti sistemos laiką“ loginė architektūros diagrama

60 pav. pateikiama PA 9 „Peržiūrėti sistemos laiką“ sekų diagrama, kurioje atvaizduojamas sistemos laiko ir laiko zonos informacijos atvaizdavimas pagrindiniame puslapyje.



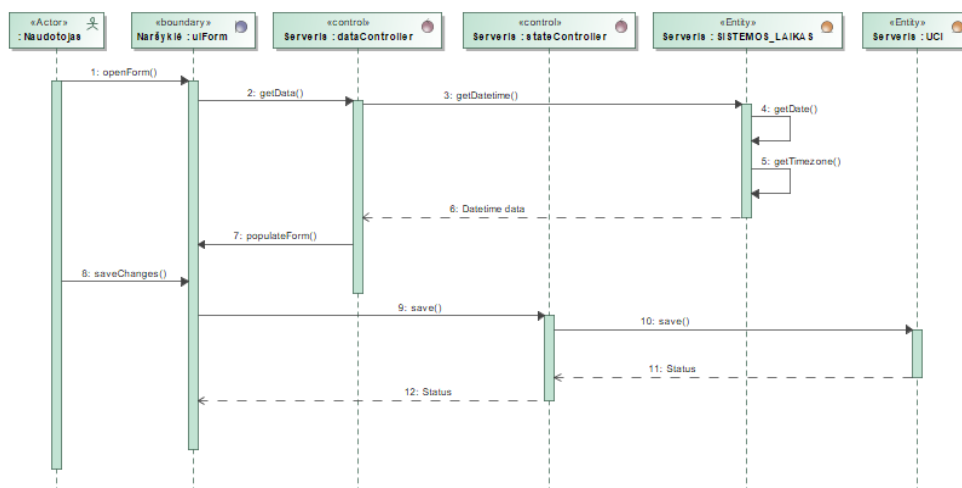
60 pav. PA 9 „Peržiūrėti sistemos laiką“ sekų diagrama

61 pav. pateikiama PA 10 „Valdyti sistemos laiką“ loginė architektūros diagrama, kurioje atvaizduojama sistemos laiko ir laiko zonos nustatymų keitimo loginė architektūra.



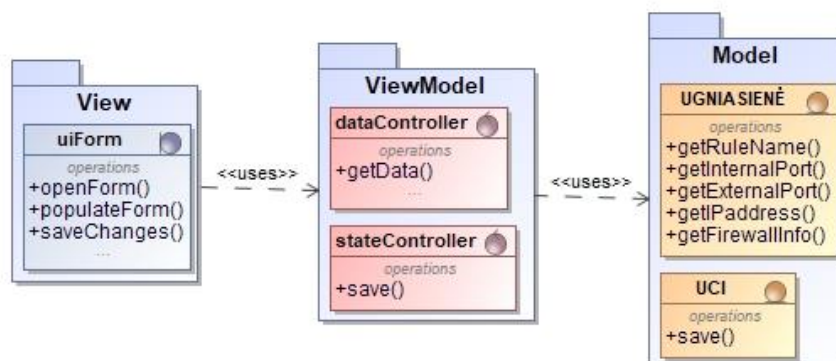
61 pav. PA 10 „Valdyti sistemos laiką“ loginė architektūros diagrama

62 pav. pateikiama PA 10 „Valdyti sistemos laiką“ sekų diagrama, kurioje atvaizduojama sistemos laiko ir laiko zonos esamos konfigūracijos atvaizdavimas ir naujų nustatymų išsaugojimas.



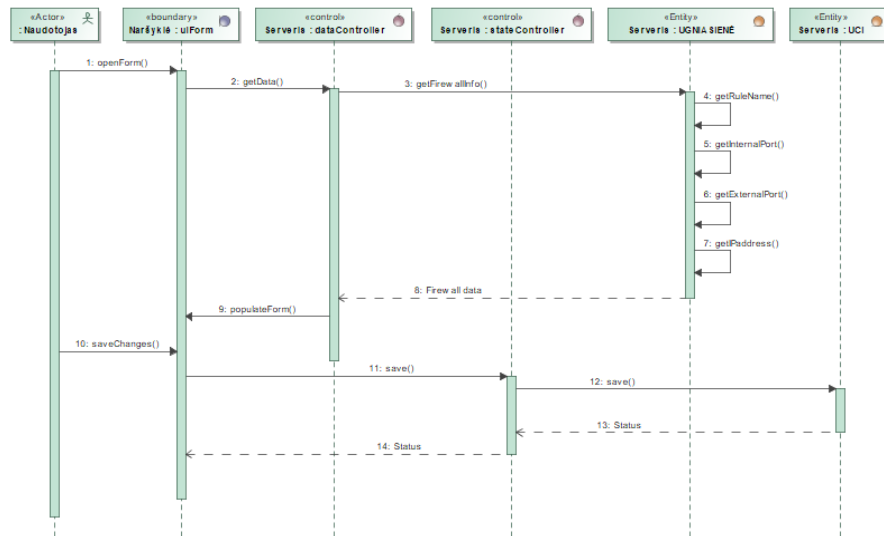
62 pav. PA 10 „Valdyti sistemos laiką“ sekų diagrama

63 pav. pateikiama PA 11 „Valdyti ugniasienės nustatymus“ loginė architektūros diagrama, kurioje atvaizduojama ugniasienės nustatymų loginė architektūra.



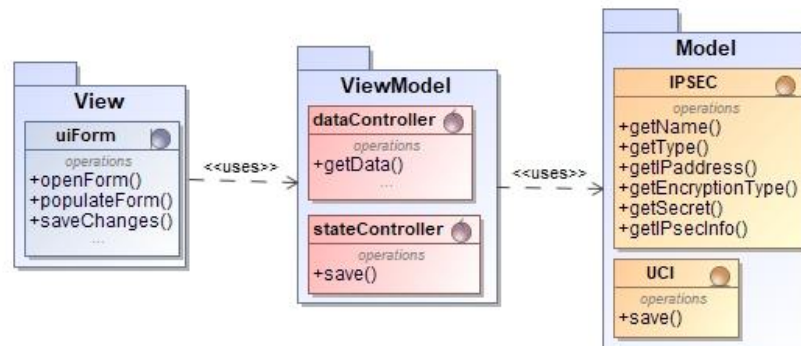
63 pav. PA 11 „Valdyti ugniasienės nustatymus“ loginė architektūra

64 pav. pateikiama PA 11 „Valdyti ugniasienės nustatymus“ sekų diagrama, kurioje atvaizduojamas ugniasienės esamų nustatymų atvaizdavimas ir naujos konfigūracijos išsaugojimas.



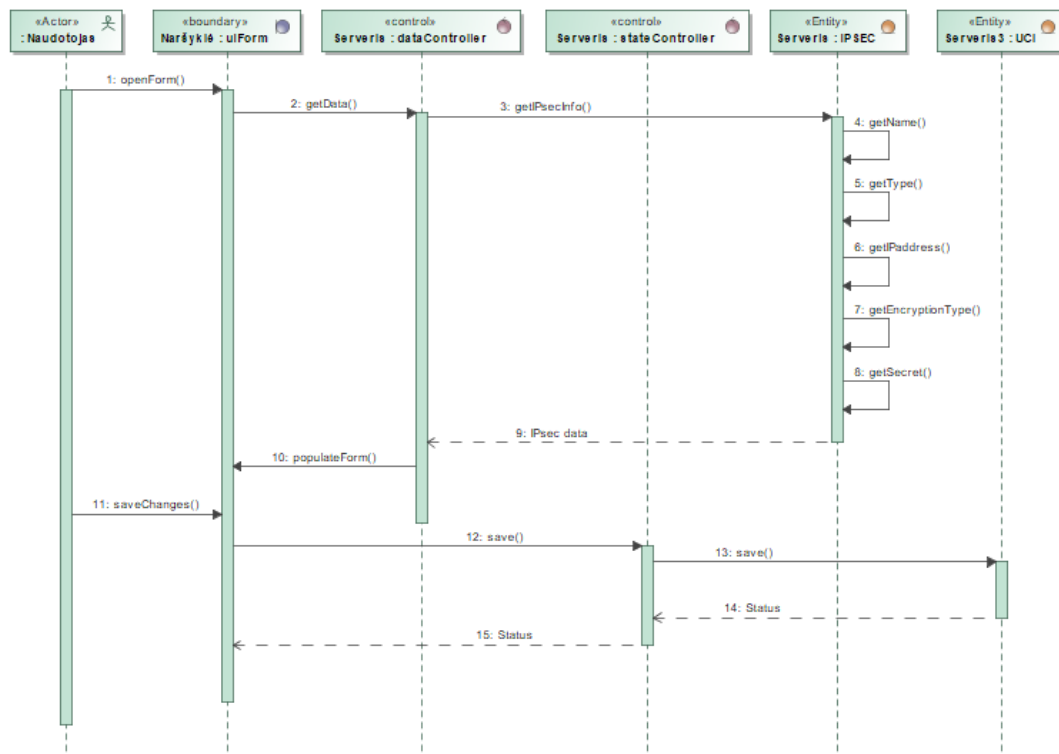
64 pav. PA 11 „Valdyti ugniasienės nustatymus“ sekų diagrama

65 pav. pateikiama PA 12 „Valdyti IPsec tunelio nustatymus“ loginė architektūros diagrama, kurioje atvaizduojama IPsec nustatymų loginė architektūra.



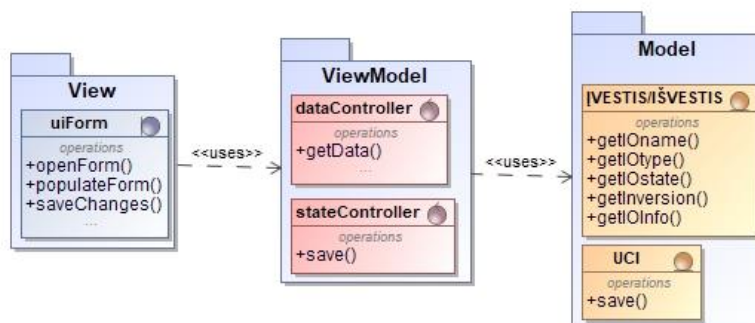
65 pav. PA 12 „Valdyti IPsec tunelio nustatymus“ loginė architektūros diagrama

66 pav. pateikiama PA 12 „Valdyti IPsec tunelio nustatymus“ sekų diagrama, kurioje atvaizduojamas IPsec esamų nustatymų atvaizdavimas ir naujos konfigūracijos išsaugojimas.



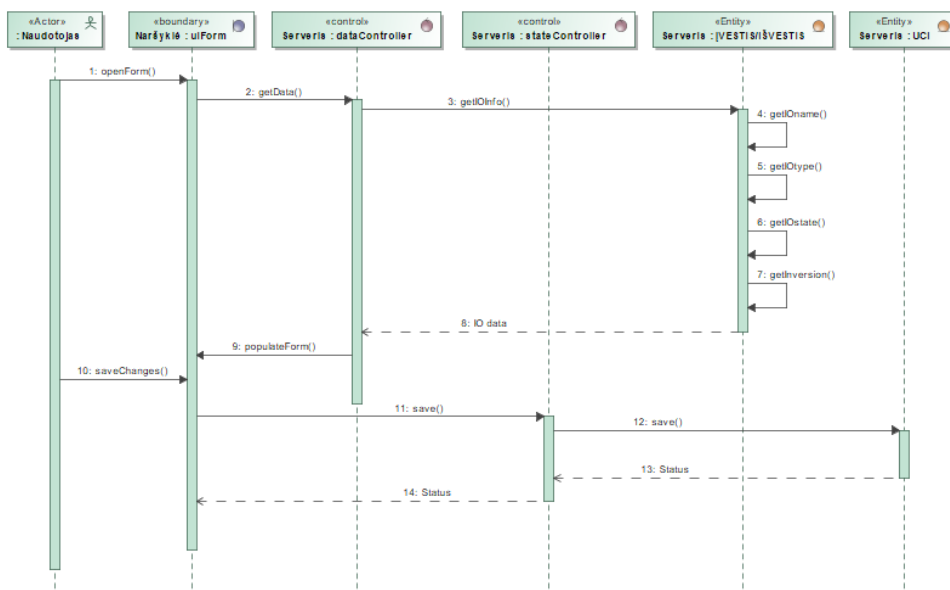
66 pav. PA 12 „Valdyti IPsec tunelio nustatymus“ sekų diagrama

67 pav. pateikiama PA 13 „Valdyti Įvesties/Išvesties nustatymus“ loginė architektūros diagrama, kurioje atvaizduojama įvesties/išvesties nustatymų loginė architektūra.



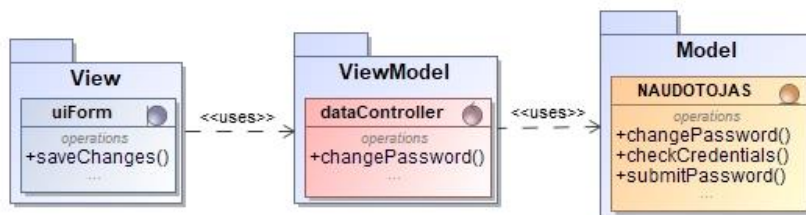
67 pav. PA 13 „Valdyti Įvesties/Išvesties nustatymus“ loginė architektūros diagrama

68 pav. pateikiama PA 13 „Valdyti Įvesties/Išvesties nustatymus“ sekų diagrama, kurioje atvaizduojamas įvesties/išvesties esamų nustatymų atvaizdavimas ir naujos konfigūracijos išsaugojimas.



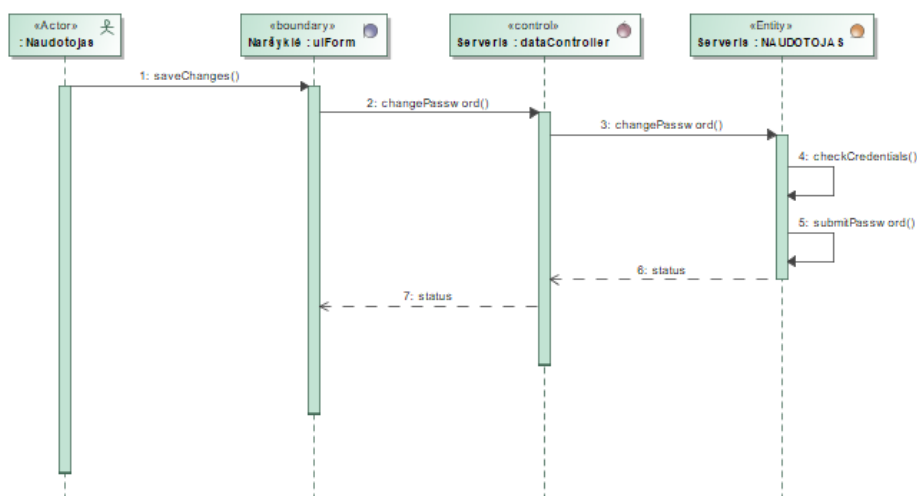
68 pav. PA 14 „Valdyti Įvesties/Išvesties nustatymus“ sekų diagrama

69 pav. pateikiama PA 14 „Keisti slaptažodį“ loginė architektūros diagrama, kurioje atvaizduojama slaptažodžio keitimo loginė architektūra.



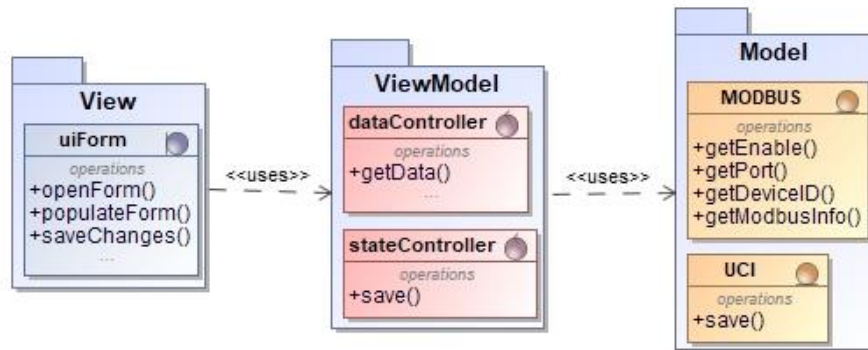
69 pav. PA 14 „Keisti slaptažodį“ loginė architektūros diagrama

70 pav. pateikiama PA 14 „Keisti slaptažodį“ sekų diagrama, kurioje atvaizduojamas sistemos naudotojo slaptažodžio keitimas.



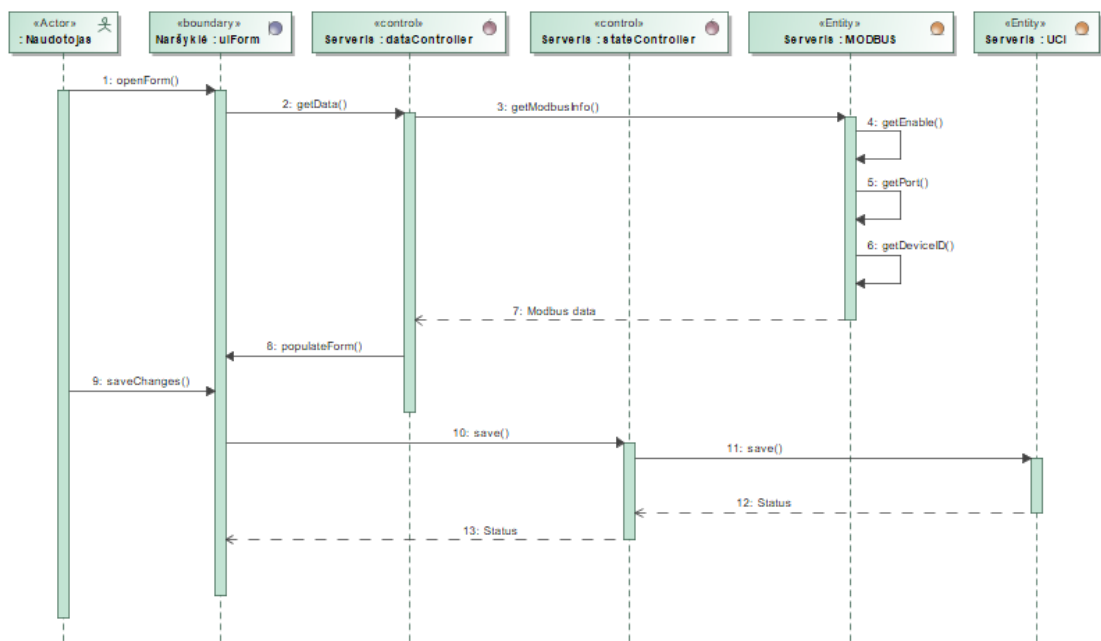
70 pav. PA 14 „Keisti slaptažodį“ sekų diagrama

71 pav. pateikiama PA 15 „Valdyti MODBUS nustatymus“ loginė architektūros diagrama, kurioje atvaizduojama MODBUS nustatymų loginė architektūra.



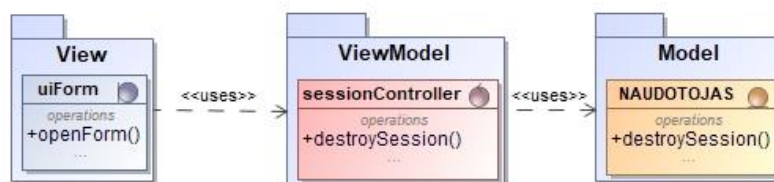
71 pav. PA 15 „Valdyti MODBUS nustatymus“ loginė architektūros diagrama

72 pav. pateikiama PA 15 „Valdyti MODBUS nustatymus“ sekų diagrama, kurioje atvaizduojamas MODBUS esamų nustatymų atvaizdavimas ir naujos konfigūracijos išsaugojimas.



72 pav. PA 15 „Valdyti MODBUS nustatymus“ sekų diagrama

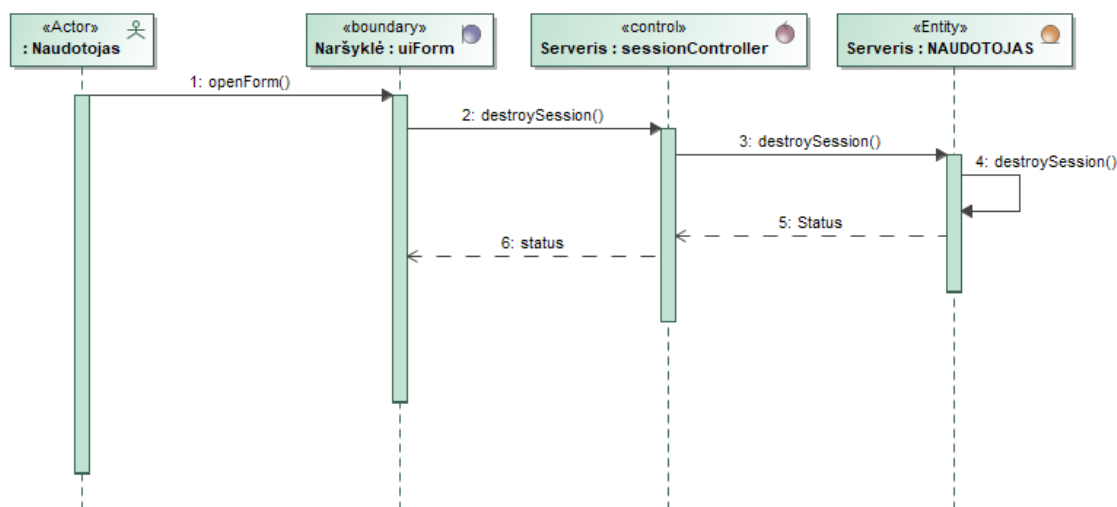
73 pav. pateikiama PA 16 „Atsijungti“ sekų diagrama, kurioje atvaizduojamas atsijungimas nuo informacinės sistemos.



73 pav. PA 16 „Atsijungti“ loginė architektūros diagrama



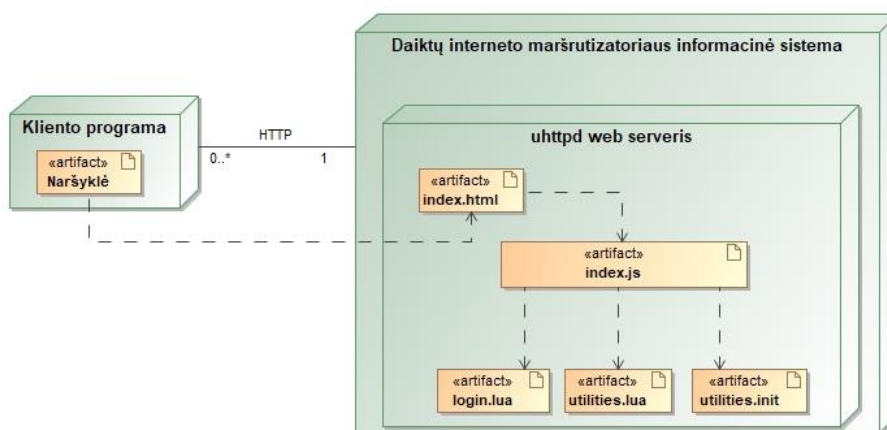
74 pav. pateikiama PA 16 „Atsijungti“ sekų diagrama, kurioje atvaizduojamas atsijungimas nuo informacinės sistemos.



74 pav. PA 16 „Atsijungti“ sekų diagrama

### 3.3. Realizacijos modelis

75 pav. pateikiamas daiktų interneto maršrutizatoriaus informacinės sistemos realizacijos modelis, kuriame atvaizduojama informacinės sistemos sąveiką su kliento programa. Informacinė sistema pasiekama per naršyklę. Daiktų interneto maršrutizatoriuje „uhttpd“ Web serveris perduoda visus esančius failus reikalingus informacinėi sistemai veikti naršyklėje. Pagrindinis informacinės sistemos artefaktas yra „index.js“ failas, kuriame yra visa pagrindinė vieno puslapio programos logika. Žemiausiame lygyje esantys failai atlieka funkcijas, kurios yra reikalingos prisijungti ir perduoti duomenis į naršyklę.

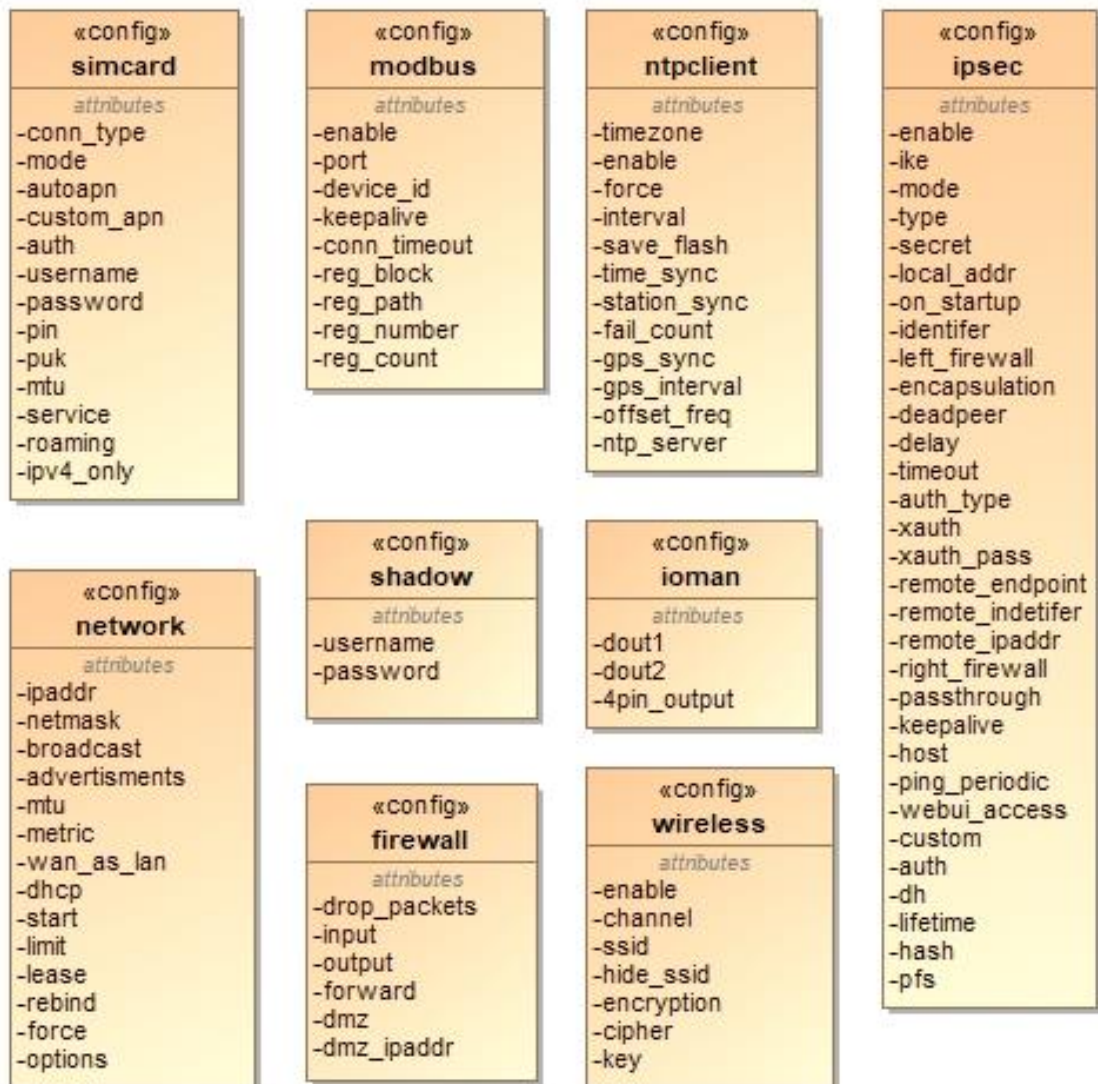


75 pav. Daiktų interneto maršrutizatoriaus informacinės sistemos realizacijos modelis

### 3.4. Konfigūracijos failų modelis

Daiktų interneto maršrutizatoriuje duomenų bazės nėra naudojamos, tačiau visa konfigūracija ir nustatymai yra saugomi konfigūracijos failuose. Kiekvienas informacinės sistemos nustatymų

valdymo puslapis turi jam priklausantį konfigūracijos failą. 76 pav. pateikiamas daiktų interneto maršrutizatoriaus informacinės sistemos konfigūracijos failų modelis.



76 pav. Daiktų interneto maršrutizatoriaus informacinės sistemos konfigūracijos failų modelis

## 4. Sprendimo realizacija ir testavimas

### 4.1. Sprendimo realizacijos ir veikimo aprašas

Sprendimo realizavimui buvo sukurtos dvi identiškos grafinės sąsajos daiktų interneto maršrutizatoriaus naudojamai operacinėj sistemai ir automatizuota testavimo programa patikrinti puslapių užkrovimo greitaveiką. Naujai sukurtos grafinės sąsajos paremtos vieno puslapio programos modeliu panaudojant Vue karkasą ir React biblioteką. Realizuotos grafinės sąsajos buvo realizuotos originalios tradicinių internetinių puslapių modeliu paremtos grafinės sąsajos pagrindu, identiška atkartojant veikimo principus. Automatizuota testavimo programa buvo realizuota „Selenium“ bibliotekos įrankiu.

#### 4.1.1. Vieno puslapio programos modelio sprendimo realizacija ir veikimo aprašas

Vieno puslapio programos modeliu sukurtos grafinės sąsajos identiška atkartoja tradicinių internetinių puslapių modeliu esamos grafinės sąsajos veikimą ir atvaizdavimą pagal iškeltus funkcinius ir nefunkcinius reikalavimus. Vue ir React technologijos naudoja tą patį internetinių puslapių kūrimo modelį, tačiau tarpusavyje turi nemažai skirtumų. Kūrimo procese Vue internetinio puslapio elementai kuriami naudojant karkaso komponentus, kurie vėliau gali būti panaudojami tolimesniame kūrimo procese pritaikant tą pačią veikimo logiką. React bibliotekos kūrimo procese pasikartojančiam panaudojimui kuriamos klasės, kurios leidžia pasiekti tą patį rezultatą kaip ir komponentai Vue karkase. Internetinių puslapių elementų stiliaus realizavimui React naudoja savitą CSS stiliaus taisyklių žymėjimą, kai Vue stiliaus taisyklės aprašomos standartiniu CSS taisyklių stiliaus žymėjimu. Panaudoti vieno puslapio programų modeliai naudoja vienodą puslapių turinio atnaujinimo metodiką, naudojant virtualų dokumento objekto modelį. Virtualus dokumento objekto modelis sulyginamas su originaliu dokumento objekto modeliu ir aptikus elementų skirtumą atnaujinami tik tie elementai, kurie skiriasi, kas leidžia turiniui greičiau atsinaujinti ir sunaudoti mažiau resursų kliento naršyklėje.

Realizuotos grafinės sąsajos susideda iš:

- Prisijungimo puslapis – puslapis skirtas identifikuoti naudotoją įvedant prisijungimo vardą ir prisijungimo slaptažodį;
- Pagrindinis sistemos puslapis „Overview“ – puslapis atvaizduojantis pagrindines sistemos būsenas ir įvykių žurnalus;
- Mobilųjų nustatymų puslapis „Mobile“ – puslapis skirtas konfigūruoti mobiliojo ryšio nustatymus;
- Lokalaus tinklo nustatymų puslapis „LAN“ – puslapis skirtas konfigūruoti lokalaus tinklo nustatymus;
- Belaidžio ryšio nustatymų puslapis „Wireless“ – puslapis skirtas konfigūruoti belaidžio ryšio nustatymus;
- Ugniasienės nustatymų puslapis „Firewall“ – puslapis skirtas konfigūruoti ugniasienės nustatymus;
- Maršrutizatoriaus laiko nustatymų puslapis „NTP“ – puslapis skirtas konfigūruoti maršrutizatoriaus laiko nustatymus;
- Virtualaus privataus tinklo nustatymų puslapis „IPSec“ – puslapis skirtas konfigūruoti virtualaus privataus tinklo „IPSec“ technologijos nustatymus;

- Duomenų perdavimo protokolo „Modbus“ nustatymų puslapis – puslapis skirtas konfigūruoti duomenų perdavimo technologijos „Modbus“ nustatymus;
- Maršrutizatoriaus įvesties ir išvesties nustatymų puslapis „Input/Output“ – puslapis skirtas valdyti maršrutizatoriaus įvestį ir išvestį;
- Maršrutizatoriaus bendrųjų nustatymų puslapis „Administration“ – puslapis skirtas keisti maršrutizatoriaus pavadinimą ir naudotojo prisijungimo slaptažodį;
- Atjungimas nuo sistemos „Logout“ – funkcija leidžianti atjungti naudotoją nuo sistemos ir nukreipti į prisijungimo puslapį;
- Konfigūracijos saugojimo funkcija – funkcija leidžianti išsaugoti naudotojo įvestus duomenis konfigūravimo puslapiuose.

#### 4.1.2. Automatizuotos testavimo programos sprendimo realizacijos ir veikimo aprašas

Sukurtoms grafinės sąsajoms buvo sukurta automatizuota testavimo programa internetinių puslapių užkrovimo greitaveikai patikrinti pagal iškeltus funkcinius reikalavimus. Automatizuotą testavimo programą sudaro:

- Kiekvieno realizuoto puslapio užkrovimo greitaveikos patikrinimo funkcijos;
- Kiekvieno realizuoto nustatymų puslapio konfigūracijos išsaugojimo greitaveikos funkcijos;
- Procesoriaus užimtumo tikrinimas visą testavimo scenarijų laikotarpį;
- Operatyviosios atminties panaudojimo tikrinimas visą testavimo scenarijų laikotarpį;
- Užfiksuotų greitaveikos rezultatų kaupimas JSON formatu;
- Testo paleidimas – nurodomas greitaveikos testas pagal grafinę sąsają (Vue, React ar Tradicinis);
- Rezultatų išvedimas – išvedami internetinių puslapių greitaveikos testavimo rezultatai milisekundėmis.

#### 4.2. Testavimo modelis, duomenys, rezultatai

Šiame skyriuje aprašomas testavimo modelis pagal funkcinius ir nefunkcinius reikalavimus grafinėms sistemoms ir automatizuotai testavimo programai;

##### 4.2.1. Vieno puslapio programos modelio funkcinių reikalavimų testavimo duomenys

22 lentelėje pateikiama PA 1 „Prisijungti“ funkcinio reikalavimo numatyti ir atlikti žingsniai, bei rezultatas ir reikalavimo išpildymas. Sukurtos grafinės sąsajos turi prisijungimo puslapį leidžiantį prijungti naudotoją prie sistemos. Funkcinio reikalavimo išpildymas – sėkmingas.

22 lentelė. PA 1 „Prisijungti“ testavimo duomenys

PA 1 „Prisijungti“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Prisijungti prie informacinės sistemos	Realizuotas prisijungimo prie informacinės sistemos puslapis	Įvedus prisijungimo vardą ir slaptažodį naudotojas yra prijungiamas prie sistemos	Sėkmingas

23 lentelėje pateikiama PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ funkcinio reikalavimo numatyti žingsniai ir reikalavimo išpildymas. Sukurtose grafinėse sąsajose pagrindiniame puslapyje yra atvaizduojama maršrutizatoriaus procesoriaus apkrova realiu laiku. Funkcinio reikalavimo išpildymas – sėkmingas.

**23 lentelė.** PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“ testavimo duomenys

PA 2 „Peržiūrėti sistemos naudojamų resursų informaciją“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Pagrindiniame puslapyje atvaizduojama procesoriaus apkrova	Realizuotas maršrutizatoriaus procesoriaus apkrovos atvaizdavimas	Perėjus į pagrindinį informacinės sistemos puslapį lentelėje atvaizduojama procesoriaus apkrova realiu laiku	Sėkmingas

Funkcinio reikalavimo PA 3 „Peržiūrėti vidinio tinklo būseną“ testavimo duomenys pateikiami 24 lentelėje. Pagrindiniame grafinės sąsajos puslapyje „Overview“ yra atvaizduojamas vidinio tinklo IP adresai ir prisijungusių klientų skaičius. Funkcinio reikalavimo išpildymas – sėkmingas.

**24 lentelė.** PA 3 „Peržiūrėti vidinio tinklo būseną“ testavimo duomenys

PA 3 „Peržiūrėti vidinio tinklo būseną“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos pagrindiniame puslapyje „Overview“ atvaizduojama vidinio tinklo informacija	Realizuota lentelė pagrindiniame puslapyje „Overview“ atvaizduojanti vidinio tinklo informaciją	Pagrindiniame sistemos puslapyje „Overview“ matoma vidinio tinklo informacijos lentelė	Sėkmingas

25 lentelėje pateikiami funkcinio reikalavimo PA 4 „Valdyti vidinio tinklo nustatymus“ testavimo duomenys. Grafinės sąsajos „LAN“ puslapyje yra atvaizduojami vidinio tinklo nustatymų konfigūravimo laukai. Funkcinio reikalavimo išpildymas – sėkmingas.

**25 lentelė.** PA 4 „Valdyti vidinio tinklo nustatymus“ testavimo duomenys

PA 4 „Valdyti vidinio tinklo nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „LAN“ puslapyje pateikiami vidinio tinklo konfigūracijos laukai	Realizuotas vidinio tinklo nustatymų konfigūravimo puslapis	Informacinės sistemos „LAN“ puslapyje matomi vidinio tinklo konfigūravimo laukai.	Sėkmingas

Funkcinio reikalavimo PA 5 „Peržiūrėti belaidžio ryšio būseną“ testavimo duomenys pateikiami 26 lentelėje. Pagrindinės grafinės sąsajos puslapyje „Overview“ yra atvaizduojamas belaidžio ryšio tinklo pavadinimas ir naudojamas dažnis. Funkcinio reikalavimo išpildymas – sėkmingas.

**26 lentelė.** PA 5 „Peržiūrėti belaidžio ryšio būseną“ testavimo duomenys

PA 5 „Peržiūrėti vidinio tinklo būseną“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos pagrindiniame puslapyje „Overview“ atvaizduojama belaidžio ryšio informacija	Realizuota lentelė pagrindiniame puslapyje „Overview“ atvaizduojanti belaidžio ryšio informaciją	Pagrindiniame sistemos puslapyje „Overview“ matoma belaidžio ryšio informacijos lentelė	Sėkmingas

27 lentelėje pateikiami funkcinio reikalavimo PA 6 „Valdyti belaidžio ryšio nustatymus“ testavimo duomenys. „Wireless“ puslapyje atvaizduojami belaidžio ryšio konfigūravimo laukai leidžiantys konfigūruoti maršrutizatoriaus belaidžio ryšio nustatymus. Funkcinio reikalavimo išpildymas – sėkmingas.

**27 lentelė.** PA 6 „Valdyti belaidžio ryšio nustatymus“ testavimo duomenys

PA6 „Valdyti belaidžio ryšio nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „Wireless“ puslapyje atvaizduojami belaidžio ryšio nustatymų konfigūravimo laukai.	Realizuotas „Wireless“ puslapis leidžianti konfigūruoti maršrutizatoriaus belaidžio ryšio nustatymus.	Sukurtas „Wireless“ puslapis, kuriame pateikiami belaidžio ryšio konfigūravimo laukai.	Sėkmingas

Funkcinio reikalavimo PA 7 „Peržiūrėti mobilaus ryšio būseną“ testavimo duomenys pateikiami 28 lentelėje. Pagrindinės grafinės sąsajos puslapyje „Overview“ yra atvaizduojama mobiliojo ryšio būsenų lentelė, kurioje pateikiama mobilaus ryšio būseną, įdėtos SIM kortelės būseną ir išsiųstų ir gautų duomenų kiekis. Funkcinio reikalavimo išpildymas – sėkmingas.

**28 lentelė.** PA 7 „Peržiūrėti mobilaus ryšio būseną“ testavimo duomenys

PA 7 „Peržiūrėti mobilaus ryšio būseną“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos pagrindiniame puslapyje „Overview“ atvaizduojama mobilaus ryšio informacija	Realizuota lentelė pagrindiniame puslapyje „Overview“ atvaizduojanti mobilaus ryšio informaciją	Pagrindiniame sistemos puslapyje „Overview“ matoma mobilaus ryšio informacijos lentelė	Sėkmingas

Funkcinio reikalavimo PA 8 „Valdyti mobilaus ryšio nustatymus“ testavimo duomenys pateikiami 29 lentelėje. Mobilaus ryšio nustatymų puslapyje atvaizduojami APN, PIN, PUK, MTU ir kiti nustatymų laukai leidžiantis įvesti mobiliam ryšiui gauti reikiamus nustatymus. Funkcinio reikalavimo išpildymas – sėkmingas.

**29 lentelė.** PA 8 „Valdyti mobilaus ryšio nustatymus“ testavimo duomenys

PA8 „Peržiūrėti mobilaus ryšio būseną“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Mobilaus ryšio nustatymų puslapyje keičiami mobilaus ryšio nustatymai	Realizuotas mobiliųjų nustatymų puslapis leidžiantis konfigūruoti mobiliojo ryšio nustatymus	Mobilaus ryšio nustatymai keičiami įvedant norimas reikšmes į atitinkamus laukelius	Sėkmingas

Funkcinio reikalavimo PA 9 „Peržiūrėti sistemos laiką“ testavimo duomenys pateikiami 30 lentelėje. Pagrindiniame informacinės sistemos puslapyje „Overview“ pateikiamas maršrutizatoriaus gyvavimo laikas ir esamas sistemos laikas. Funkcinio reikalavimo išpildymas – sėkmingas.

**30 lentelė.** PA 9 „Peržiūrėti sistemos laiką“ testavimo duomenys

PA 9 „Peržiūrėti sistemos laiką“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos pagrindiniame puslapyje atvaizduojama sistemos laiko informacija	Realizuota lentelė pagrindiniame puslapyje „Overview“ atvaizduojanti sistemos laiką	Pagrindiniame sistemos puslapyje „Overview“ matoma sistemos laiko informacijos lentelė	Sėkmingas

31 lentelėje pateikiami funkcinio reikalavimo PA 10 „Valdyti sistemos laiko nustatymus“ testavimo duomenys. Informacinės sistemos „NTP“ puslapyje yra pateikiami sistemos laiko nustatymų konfigūravimo laukai. Funkcinio reikalavimo išpildymas – sėkmingas.

**31 lentelė.** PA 10 „Valdyti sistemos laiko nustatymus“ testavimo duomenys

PA 10 „Valdyti sistemos laiko nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „NTP“ puslapyje atvaizduojami sistemos laiko nustatymų konfigūravimo laukai.	Realizuotas „NTP“ sistemos laiko nustatymų puslapis.	Informacinės sistemos „NTP“ puslapyje atvaizduojami sistemos laiko nustatymų konfigūravimo laukai.	Sėkmingas

Funkcinio reikalavimo PA 11 „Valdyti ugniasienės nustatymus“ testavimo duomenys pateikiami 32 lentelėje. Informacinės sistemos „Firewall“ puslapyje atvaizduojami ugniasienės konfigūravimo laukai. Funkcinio reikalavimo išpildymas – sėkmingas.

**32 lentelė.** PA 11 „Valdyti ugniasienės nustatymus“ testavimo duomenys

PA 11 „Valdyti ugniasienės nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „Firewall“ puslapyje atvaizduojami ugniasienės konfigūravimo laukai.	Realizuotas „Firewall“ ugniasienės nustatymų puslapis.	Informacinės sistemos „Firewall“ puslapyje atvaizduojami ugniasienės konfigūravimo laukai.	Sėkmingas

33 lentelėje pateikiami PA 12 „Valdyti IPSec tunelio nustatymus“ testavimo duomenys. Informacinės sistemos „VPN“ puslapyje pateikiama „IPSec“ virtualaus vidinio tinklo nustatymų konfigūracija. Funkcinio reikalavimo išpildymas – sėkmingas.

**33 lentelė.** PA 12 „Valdyti IPSec tunelio nustatymus“ testavimo duomenys

PA 12 „Valdyti IPSec tunelio nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „VPN“ puslapyje atvaizduojami „IPSec“ tunelio konfigūravimo laukai.	Realizuotas „VPN“ puslapis leidžiantis konfigūruoti „IPSec“ virtualaus vidinio tinklo nustatymus.	Informacinės sistemos „VPN“ puslapyje atvaizduojami „IPSec“ virtualaus vidinio tinklo konfigūravimo laukai.	Sėkmingas

34 lentelėje pateikiami funkcinio reikalavimo PA 13 „Valdyti įvesties/išvesties nustatymus“ testavimo duomenys. „Input/Output“ informacinės sistemos puslapyje atvaizduojami laukai leidžiantis pakeisti įvesties ir išvesties aparatinės įrangos būsenas. Funkcinio reikalavimo išpildymas – sėkmingas.

**34 lentelė.** PA 13 „Valdyti įvesties/išvesties nustatymus“ testavimo duomenys

PA 13 „Valdyti įvesties/išvesties nustatymus“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „Input/Output“ atvaizduojami aparatinės įrangos įvesties ir išvesties nustatymų laukai.	Realizuotas „Input/Output“ puslapis, kuriame pateikiami aparatinės įrangos įvesties ir išvesties būsenų konfigūravimo laukai.	Informacinės sistemos „Input/Output“ puslapyje atvaizduojami maršrutizatoriaus aparatinės įrangos įvesties ir išvesties konfigūravimo laukai..	Sėkmingas

Funkcinio reikalavimo PA 14 „Keisti slaptažodį“ testavimo duomenys pateikiami 35 lentelėje. Informacinės sistemos „Administration“ puslapyje pateikiami sistemos slaptažodžio keitimo laukai. Funkcinio reikalavimo išpildymas – sėkmingas.

**35 lentelė.** PA 14 „Keisti slaptažodį“ testavimo duomenys

PA 14 „Keisti slaptažodį“ testavimo duomenys			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „Administration“ puslapyje atvaizduojami sistemos slaptažodžio keitimo laukai.	Realizuotas „Administration“ puslapis atvaizduojantis sistemos slaptažodžio keitimo laukus.	Informacinės sistemos „Administration“ puslapyje atvaizduojami slaptažodžio keitimo laukai.	Sėkmingas

36 lentelėje pateikiami funkcinio reikalavimo PA 15 „Valdyti MODBUS nustatymus“ testavimo duomenys. Informacinėje sistemoje atvaizduojami komunikacijos protokolo MODBUS konfigūravimo laukai.

**36 lentelė.** PA 15 „Valdyti MODBUS nustatymus“ testavimo duomenys

PA 15 „Valdyti MODBUS nustatymus“ testavimo duomenys			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Informacinės sistemos „Modbus“ puslapyje atvaizduojami informacijos perdavimo protokolo konfigūravimo laukai.	Realizuotas „Modbus“ puslapis atvaizduojantis informacijos perdavimo protokolo konfigūravimo laukus.	Informacinės sistemos „Modbus“ puslapyje atvaizduojami informacijos perdavimo protokolo konfigūravimo laukai.	Sėkmingas

Funkcinio reikalavimo PA 16 „Atsijungti“ testavimo duomenys pateikiami 37 lentelėje. Visoje grafinėje sąsajos viršutiniame dešiniame kampe matomas atsijungimo mygtukas leidžiantis atjungti naudotoją nuo informacinės sistemos. Funkcinio reikalavimo išpildymas – sėkmingas.

**37 lentelė.** PA 16 „Atsijungti“ testavimo duomenys

PA 16 „Peržiūrėti sistemos laiką“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Naudotojas atsijungia nuo informacinės sistemos	Realizuotas atsijungimo mygtukas matomas visoje grafinėje sąsajoje	Naudotojas nukreipiamas į prisijungimo puslapį	Sėkmingas

#### 4.2.2. Vieno puslapio programos modelio nefunkcinių reikalavimų testavimo duomenys

Nefunkcinių reikalavimų vieno puslapio programos modelio testavimo duomenys pateikiami 38 lentelėje.

**38 lentelė.** Vieno puslapio programos modelio nefunkcinių reikalavimų testavimo duomenys

Vieno puslapio programos modelio nefunkcinių reikalavimų testavimo duomenys			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Naudoti metodą paremtą vieno puslapio programos modeliu	Realizuotos grafinės sąsajos paremtos vieno puslapio modeliu	Realizuotos grafinės sistemos panaudojant Vue ir React technologijas, kurios paremtos vieno puslapio programos modeliu	Sėkmingas



Pirmojo prisijungimo prie sistemos metu serveris turi nusiųsti klientus visus reikiamus failus informacinės sistemos informacijai atvaizduoti	Realizuotos grafinės sąsajos pirmojo prisijungimo prie serverio metu gauna visus reikiamus failus grafinėi sąsajai atvaizduoti	Naršant grafinėje sąsajoje atnaujinami tik dinaminiai duomenys nereikalaujant atvaizdavimo duomenų atnaujinimo parsuntimo	Sėkmingas
Informacinės sistemos turinys turi atsinaujinti realiu laiku ir naudotojas neturi perkrauti internetinio puslapio, kad pamatytų atsinaujinusią informaciją	Realizuota grafinė sąsaja atnaujinama tik dinaminiais duomenimis kintančius realiu laiku	Grafinėje sąsajoje duomenys atnaujinami be puslapio perkrovimo	Sėkmingas
Informacija tarp serverio ir kliento turi būti perduodama JSON formatu	Realizuotos grafinės sąsajos su serveriu keičiasi dinaminiais duomenimis naudojant JSON formatą	Grafinėje sąsajoje duomenys gaunami, skaitomi ir siunčiami JSON formatu.	Sėkmingas
Internetinio puslapio elementai turi būti atnaujinami naudojantis virtualiu DOM	Realizuotos grafinės sąsajos paremtos vieno puslapio programos modeliu naudoja virtualų DOM dinaminį duomenų elementų pasikeitimui	Grafinėje sąsajoje duomenys atnaujinami neperkraudami puslapio	Sėkmingas

#### 4.2.3. Automatizuotos testavimo programos funkcinių reikalavimų testavimo duomenys

Funkcinio reikalavimo PA 17 „Įvykdyti testą“ testavimo duomenys pateikiami 39 lentelėje. Komandinėje eilutėje paleidžiamas ir sėkmingai užbaigiamas testas. Funkcinio reikalavimo išpildymas – sėkmingas.

**39 lentelė.** PA 17 „Įvykdyti testą“ testavimo duomenys

PA 17 „Įvykdyti testą“ testavimo duomenys.			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Testuotojas paleidžia programą.	Realizuoti automatizuoti testavimo scenarijai grafinėms sąsajoms	Automatizuota programa įvykdo scenarijų testavimą ir baigia darbą.	Sėkmingas

Funkcinio reikalavimo PA 18 „Parinkti testuojamą technologiją“ testavimo duomenys pateikiami 40 lentelėje. Komandinėje eilutė prieš paleidžiant testą prirašomas norimos technologijos pavadinimas. Atidaroma naršyklė, kurioje atliekami automatizuotos programos vykdomi scenarijai. Funkcinio reikalavimo išpildymas – sėkmingas.

**40 lentelė.** PA 18 Parinkti testuojamą technologiją

PA 18 Parinkti testuojamą technologiją.			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Testuotojas paleidžia automatizuotą programą prirašydamas technologijos pavadinimą.	Realizuota testavimo programa, kuri pagal nurodytą technologiją jungiasi prie informacinės sistemos.,	Automatizuota testavimo programa jungiasi prie nurodytos technologijos informacinės sistemos pagal užprogramuotą IP adresą.	Sėkmingas

Funkcinio reikalavimo PA 19 „Generuoti testavimo ataskaitą“ testavimo duomenys pateikiami 41 lentelėje. Atlikus testą yra sugeneruojami duomenys JSON formatu, kurie įrašomi į bendrą rezultatų

failą ir taip pat yra atspausdinami esamo testo rezultatai. Funkcinio reikalavimo išpildymas – sėkmingas.

**41 lentelė.** PA 19 „Generuoti testavimo ataskaitą“ testavimo duomenys

PA 19 „Generuoti testavimo ataskaitą“			
Numatyti žingsniai	Atlikti žingsniai	Rezultatas	Išpildymas
Automatizuota programa pateikia testavimo ataskaitą	Realizuotas testavimo rezultatų įrašymas į bendrą rezultatų failą ir atspausdinamas esamo testo rezultatas	Atspausdinamas įvykdyto testo rezultatas	Sėkmingas

## 5. Eksperimentinis grafinių sąsajų greitaveikos tyrimas

### 5.1. Eksperimento planas

Daiktų interneto maršrutizatoriaus sukurtoms grafinės sistemoms buvo panaudoti automatizuoti testai panaudojant „Selenium“ biblioteką, kurios pagalba yra imituojami naudotojo veiksmai įvedant nurodytą tekstą arba paspaudžiant reikiamus mygtukus. Automatizuotą programą sudaro:

1. Prisijungimas – naudotojo vardo ir naudotojo slaptažodžio įvedimas ir mygtuko paspaudimas „Login“.
2. Pagrindinio puslapio užkrovimo laukimas – po prisijungimo grafinė sąsaja nukreipia naudotoją į pagrindinį sistemos puslapį „Overview“. Automatizuota programa laukia kol užsikraus visi puslapio elementai ir skaičiuoja kiek laiko truko nuo prisijungimo iki pilno puslapio „Overview“ užkrovimo.
3. Perėjimas iš „Overview“ į „Mobile“ puslapį – meniu juostoje pasirenkamas „Mobile“ punktas, kuris nukreipia į mobiliųjų nustatymų puslapį. Kol vyksta nukreipimas automatizuota programa skaičiuoja kiek laiko trunka šis procesas.
4. Mobilųjų nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „MTU“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
5. Perėjimas iš „Mobile“ į „LAN“ puslapį – meniu juostoje parenkamas „LAN“ punktas, kuris nukreipia į vidinio tinklo nustatymų puslapį. Kol vyksta nukreipimas automatizuota programa skaičiuoja kai laiko trunka nukreipimo procesas.
6. Vidinio tinklo „LAN“ nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „Start“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
7. Perėjimas iš „LAN“ į „Wireless“ puslapį – meniu juostoje parenkamas „Wireless“ punktas, kuris nukreipia į bendrąjį belaidžio ryšio nustatymų puslapį. Toliau paspaudžiamas mygtukas „Edit“, kuris nukreipia naudotoją į belaidžio ryšio nustatymų puslapį. Kol vyksta nukreipimas į belaidžio ryšio nustatymų puslapį automatizuota programa skaičiuoja laiką per kurį įvyksta nukreipimas.
8. Belaidžio ryšio „Wireless“ nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „SSID“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
9. Perėjimas iš „Wireless“ į „Firewall“ puslapį – meniu juostoje parenkamas „Firewall“ punktas, kuris nukreipia į ugniasienės nustatymų puslapį. Nukreipimo metu matuojamas nukreipimo greitaveikos laikas.
10. Ugniasienės nustatymų „Firewall“ keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „DMZ host IP address“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
11. Perėjimas iš „Firewall“ į „NTP“ puslapį – meniu juostoje parenkamas „NTP“ punktas, kuris nukreipia naudotoją į sistemos laiko nustatymų puslapį. Nukreipimo metu matuojamas greitaveikos laikas.
12. Sistemos laiko nustatymų „NTP“ keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „Update interval (in seconds)“ konfigūravimo puslapio laukelį ir paspaudžia

- mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
13. Perėjimas iš „NTP“ į „VPN“ puslapį - meniu juostoje parenkamas „VPN“ punktas, kuris nukreipia į bendrąjį virtualaus vidinio tinklo nustatymų puslapį. Toliau paspaudžiamas mygtukas „Edit“, kuris nukreipia naudotoją į „IPSec“ nustatymų puslapį. Kol vyksta nukreipimas į „IPSec“ nustatymų puslapį automatizuota programa skaičiuoja laiką per kurį įvyksta nukreipimas.
  14. Virtualaus vidinio tinklo „IPSec“ nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „Custom“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
  15. Perėjimas iš „VPN“ į „Modbus“ puslapį - meniu juostoje parenkamas „Modbus“ punktas, kuris nukreipia naudotoją į informacijos perdavimo protokolo nustatymų puslapį. Nukreipimo metu matuojamas greitimeikos laikas.
  16. „Modbus“ nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „Device ID“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
  17. Perėjimas iš „Modbus“ į „Input/Output“ puslapį – meniu juostoje parenkamas „Input/Output“ punktas, kuris nukreipia naudotoją į įvesties ir išvesties konfigūravimo puslapį. Nukreipimo metu matuojamas laikas per kurį įvyksta nukreipimas.
  18. Įvesties ir išvesties nustatymų keitimas ir išsaugojimas – automatizuota programa parenka nurodytą reikšmę į „Relay output“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
  19. Perėjimas iš „Input/Output“ į „Administration“ puslapį - meniu juostoje parenkamas „Administration“ punktas, kuris nukreipia naudotoją į įvesties ir išvesties konfigūravimo puslapį. Nukreipimo metu matuojamas laikas per kurį įvyksta nukreipimas.
  20. Maršrutizatoriaus pavadinimo nustatymų keitimas ir išsaugojimas – automatizuota programa įveda nurodytą reikšmę į „Router name“ konfigūravimo puslapio laukelį ir paspaudžia mygtuką „Save“. Po mygtuko paspaudimo programa skaičiuoja laiką per kurį maršrutizatoriaus sistema išsaugo įrašytą reikšmę.
  21. Atsijungimas nuo sistemos – informacinės sistemos puslapyje paspaudžiamas „Logout“ mygtukas, kuris nukreipia naudotoją į prisijungimo prie sistemos puslapį.

Apibendrinti atvejai ir gaunami rezultatai pateikiami 44 lentelėje „Grafinės sąsajos testavimo scenarijai“.

**44 lentelė.** Grafinės sąsajos testavimo scenarijai

Nr.	Scenarijus	Matavimo atvejai ir žingsniai	
1.	Prisijungimas	1. Įvedamas prisijungimo vardas ir slaptažodis ir paspaudžiamas mygtukas „Login“.	2. Nukreipiama į pagrindinį „Overview“ puslapį.
2.	Pagrindinio puslapio užkrovimo laukimas	1. Užfiksavus „Login“ mygtuko paspaudimą pradedamas skaičiuoti laikas kol užsikraus „Overview“ puslapio elementai ir duomenys.	2. Pilnai užsikrovus „Overview“ puslapiui užfiksuojamas laikas per kurį buvo prisijungta ir nukreiptą „Overview“ puslapį.
3.	Perėjimas iš „Overview“ į „Mobile“	1. Pasirenkamas ir paspaudžiamas meniu juostoje „Mobile“ elementas ir	2. Užfiksuojamas laikas per kurį buvo įvykdytas nukreipimas iš „Overview“ puslapio į „Mobile“ puslapį.

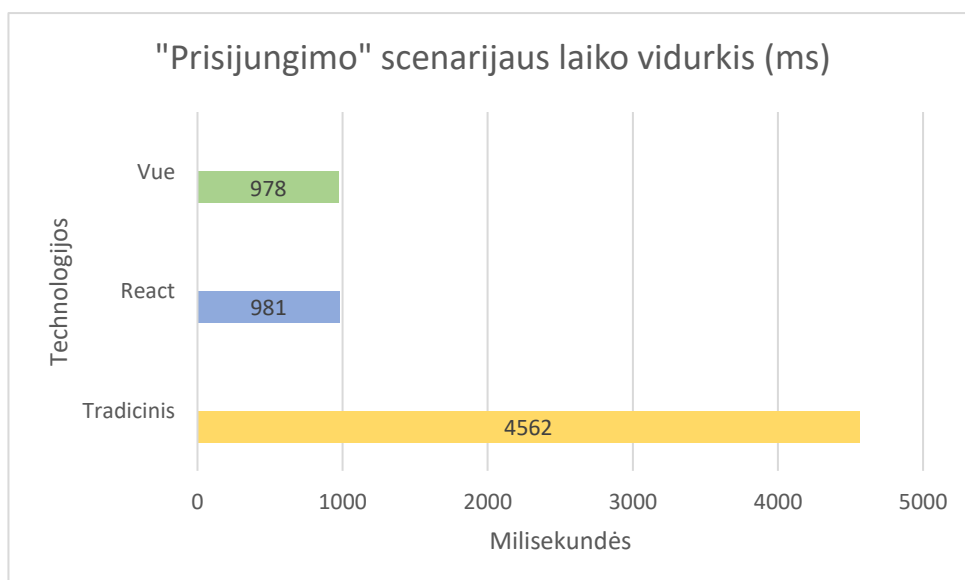
		pradedamas skaičiuoti laikas nuo mygtuko paspaudimo.	
4.	Mobiliųjų nustatymų „Mobile“ keitimas ir išsaugojimas	1. Mobilųjų nustatymų puslapyje į „MTU“ laukelį įrašoma programoje nustatyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti laikas ir laukiama kada konfigūraciją bus išsaugota.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
5.	Perėjimas iš „Mobile“ į „LAN“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „LAN“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „LAN“ puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
6.	Vidinio tinklo „LAN“ nustatymų keitimas ir išsaugojimas	1. Vidinio tinklo puslapyje į „Start“ laukelį įrašoma programoje nustatyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
7.	Perėjimas iš „LAN“ į „Wireless“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „Wireless“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „Wireless“ puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
8.	Belaidžio ryšio „Wireless“ nustatymų keitimas ir išsaugojimas	1. Belaidžio ryšio puslapyje į „SSID“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
9.	Perėjimas iš „Wireless“ į „Firewall“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „Firewall“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „Firewall“ puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
10.	Ugniasienės nustatymų „Firewall“ keitimas ir išsaugojimas	1. Ugniasienės puslapyje į „DMZ host IP address“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
11.	Perėjimas iš „Firewall“ į „NTP“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „NTP“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „NTP“ puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
12.	Sistemos laiko nustatymų „NTP“ keitimas ir išsaugojimas	1. Ugniasienės puslapyje į „Update interval (in seconds)“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
13.	Perėjimas iš „NTP“ į „VPN“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „VPN“ punktas ir pagrindiniame virtualaus vidinio tinklo nustatymų puslapyje paspaudžiamas „Edit“ mygtukas. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Informacinė sistema nukreipia į „IPSec“ nustatymų puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.

14.	Virtualaus vidinio tinklo „IPSec“ nustatymų keitimas ir išsaugojimas	1. Virtualaus vidinio tinklo puslapyje į „Custom“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
15.	Perėjimas iš „VPN“ į „Modbus“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „Modbus“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „Modbus“ nustatymų puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
16.	„Modbus“ nustatymų keitimas ir išsaugojimas	1. „Modbus“ puslapyje į „Device ID“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
17.	Perėjimas iš „Modbus“ į „Input/Output“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „Input/Output“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „Input/Output“ nustatymų puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
18.	Įvesties ir išvesties nustatymų keitimas ir išsaugojimas	1. Įvesties ir išvesties nustatymų puslapyje „Relay output“ laukeliui parenkama nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
19.	Perėjimas iš „Input/Output“ į „Administration“ puslapį	1. Meniu juostoje parenkamas ir paspaudžiamas „Administration“ punktas ir užfiksuojamas nukreipimo pradžios laikas testavimo programoje.	2. Informacinė sistema nukreipia į „Administration“ nustatymų puslapį, o testavimo programa užfiksuoja laiką per kurį buvo įvykdytas nukreipimas.
20.	Maršrutizatoriaus pavadinimo nustatymų keitimas ir išsaugojimas	1. „Administration“ puslapyje į „Router name“ laukelį įrašoma nurodyta reikšmė ir paspaudžiamas mygtukas „Save“. Paspaudus mygtuką pradedamas skaičiuoti konfigūracijos išsaugojimo laikas.	2. Pasibaigus konfigūracijos išsaugojimui yra užfiksuojamas laikas per kurį įvyko išsaugojimo procesas.
21.	Atsijungimas nuo sistemos	1. Paspaudžiamas atsijungimo „Logout“ mygtukas.	2. Informacinė sistema nukreipia į prisijungimo prie sistemos puslapį.

## 5.2. Eksperimento rezultatai

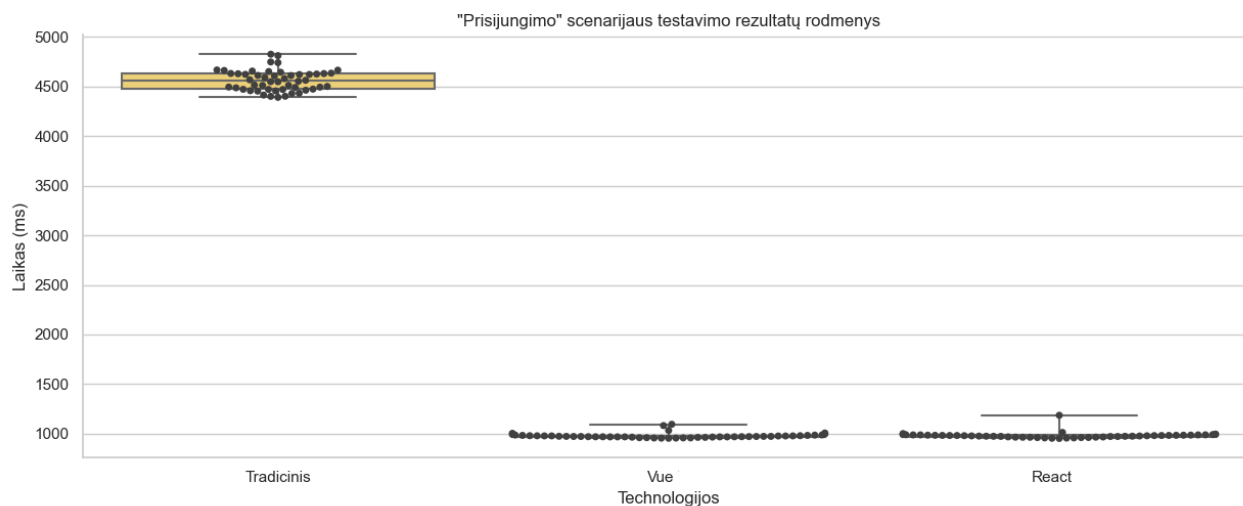
### 5.2.1. Puslapių užkrovimo ir duomenų išsaugojimo laikai

Kiekvienai grafinėj sąsajai buvo atlikta po 50 testavimo iteracijų automatizuotos testavimo programos pagalba atliekant vienodus testavimo scenarijus. Eksperimento rezultatai pateikiami diagramose pagal gautas rezultatų vidurkių reikšmes ir testavimo įverčių pasiskirstymą panaudojant „Box plot“ ir „Swarm plot“ tipo diagramas. 77 pav. pateikiami prisijungimo prie grafinės sąsajos testavimo vidurkių rezultatai.

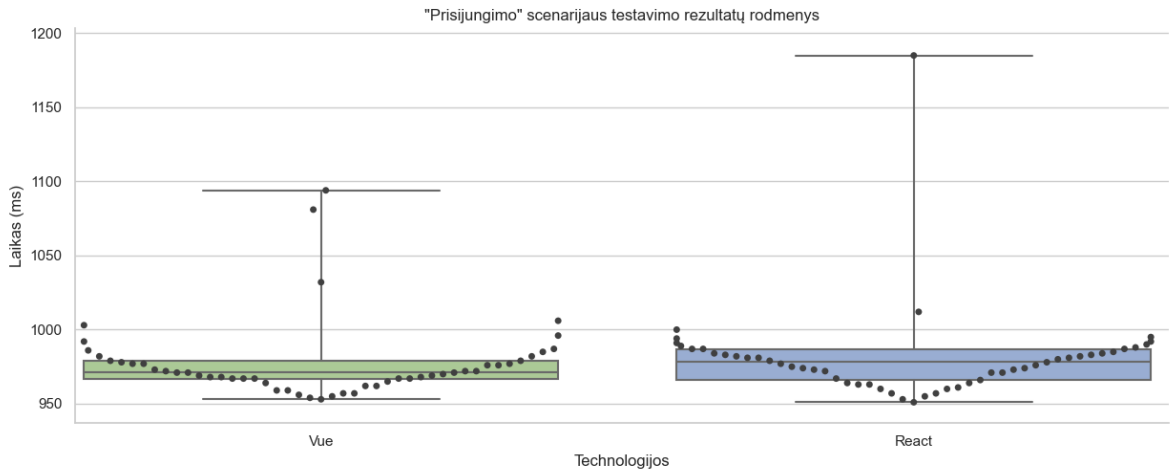


77 pav. Prisijungimo prie informacinės sistemos testavimo vidurkių diagrama

Pagal prisijungimo prie informacinės sistemos testo rezultatus matoma, kad vieno puslapio programos modelio technologijos scenarijų atliko per beveik vienodą laiką, skirtumas tarp technologijų yra tik 3 milisekundės. Vieno puslapio modeliu paremtos grafinės sąsajos prisijungimo testą atliko 4,5 karto greičiau, nei tradiciniu modeliu paremta grafinė sąsaja, kurios testas truko, šiek tiek ilgiau nei 4,5 sekundes. Vieno puslapio programos modeliu paremtos grafinės sąsajos atlikto tą patį scenarijų per mažiau, nei 1 sekundę. 78 pav. ir 79 pav. pateikiamos prisijungimo prie informacinės sistemos įverčių pasiskirstymo diagramos.



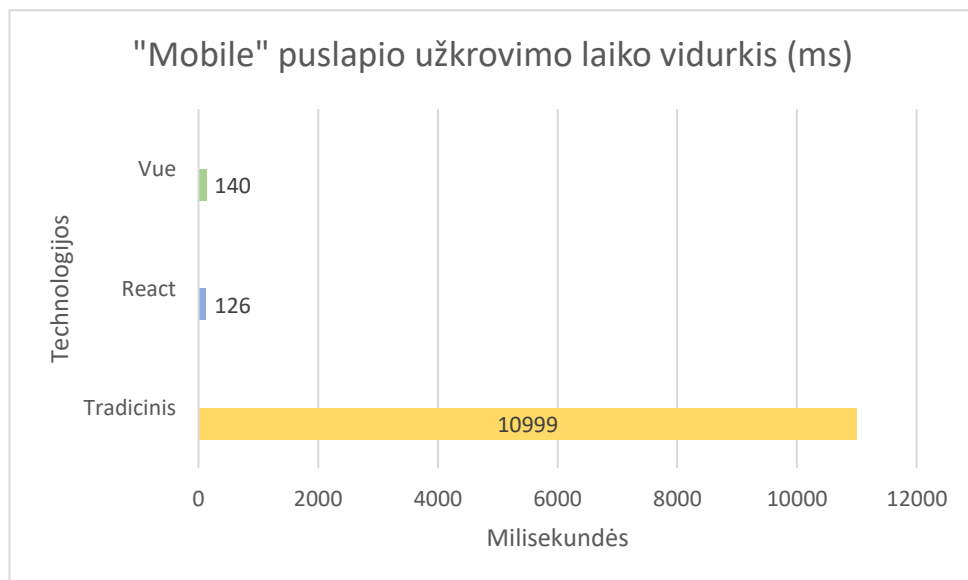
78 pav. „Prisijungimo“ testavimo scenarijaus įverčių pasiskirstymo diagrama



**79 pav.** Vieno puslapio programų „Prisijungimo“ testavimo scenarijaus įverčių pasiskirstymo diagrama

Pagal įverčių pasiskirstymo diagramas matoma, kad tradicinio modelio informacinės sistemos sklaidos taškai svyruoja per apytiksliai 500 milisekundžių. Vieno puslapio programų sklaidos taškai svyruoja 50 milisekundžių intervale, neskaitant kelių taškų viršijančių Q3 kvantilius. Taškai 100 milisekundžių viršijantys kvantilius gali būti užfiksuoti, dėl testavimo aplinkos trukdžių. Taip pat atsižvelgiant į diagramų duomenis matoma, kad Vue mediana yra ties 971 milisekundėmis, React – 979 milisekundžių, o Tradicinio modelio – 4560 milisekundžių.

Pagrindinio puslapio „Overview“ perėjimas į mobiliųjų nustatymų puslapį „Mobile“ pateikiamas 80 pav. „Pagrindinio puslapio nukreipimas į mobiliųjų nustatymų puslapį“ diagramoje.

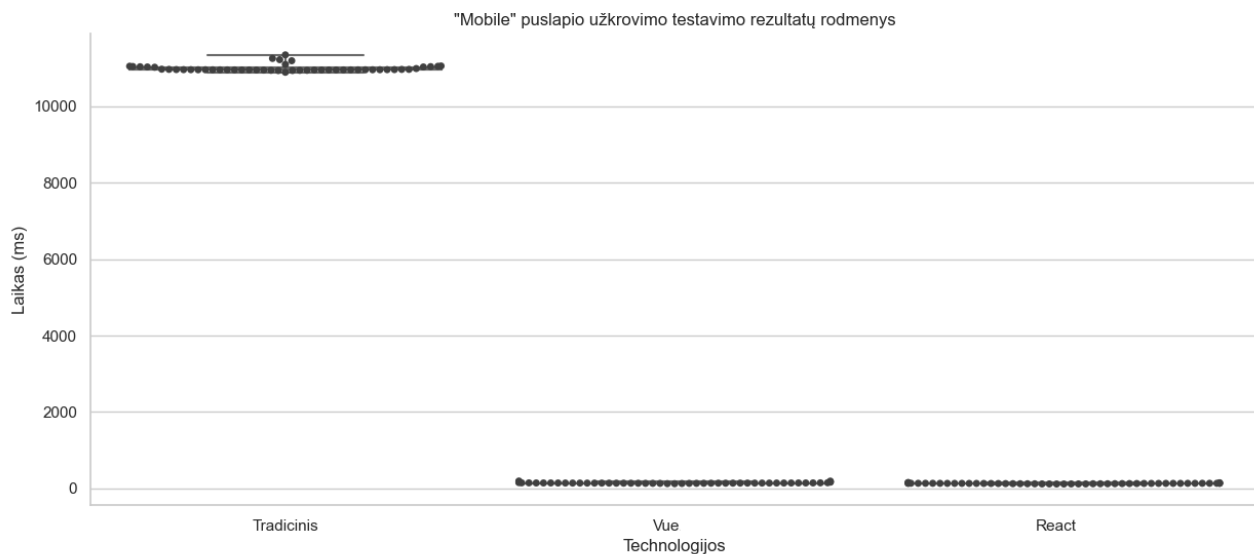


**80 pav.** „Mobile“ puslapio užkrovimo testavimo vidurkių diagrama

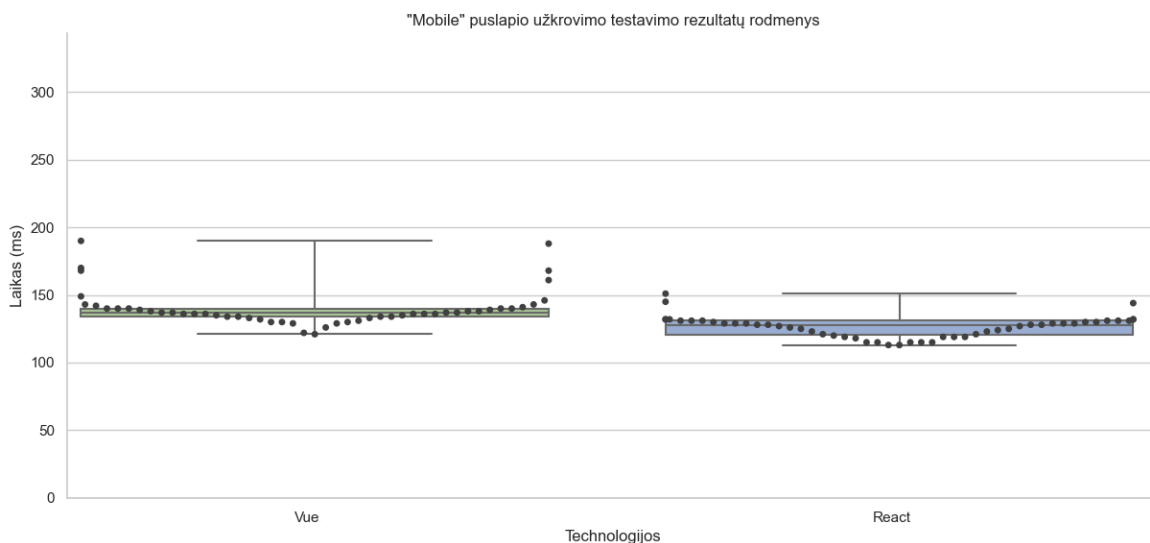
Pagal testavimo rezultatus matoma, kad vieno puslapio programos modelio grafinės sąsajos truko beveik vienodai, tačiau lyginant su tradicinio internetinių puslapių modeliu išsiskyrė viršijant daugiau nei 78 kartais greitesniu puslapių užkrovimu. Vieno puslapio programos pasižymėjo tokia didele sparta, dėl internetinių puslapių turinio, kuris jau yra laikomas naršyklėje ir reikia tik atvaizduoti turinį, kai tradicinio internetinių puslapių modeliu paremtoje grafinėje sąsajos pereinant į kitą puslapį reikia iš naujo sugeneruoti puslapį. Šis procesas ypač daug laiko kainuoja daiktų interneto



maršrutizatoriui dėl mažos spartos aparatinės įrangos procesoriaus ir procesų reikalingų naujo puslapio sugeneravimui. Tradicinio internetinių puslapių modelio grafinė sąsaja testas truko šiek tiek beveik 11 sekundžių, kai vieno puslapio programos modeliu paremtoms grafinėms sąsajomis truko mažiau nei 150 milisekundžių. 81 pav. ir 82 pav. pateikiamos „Mobile“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



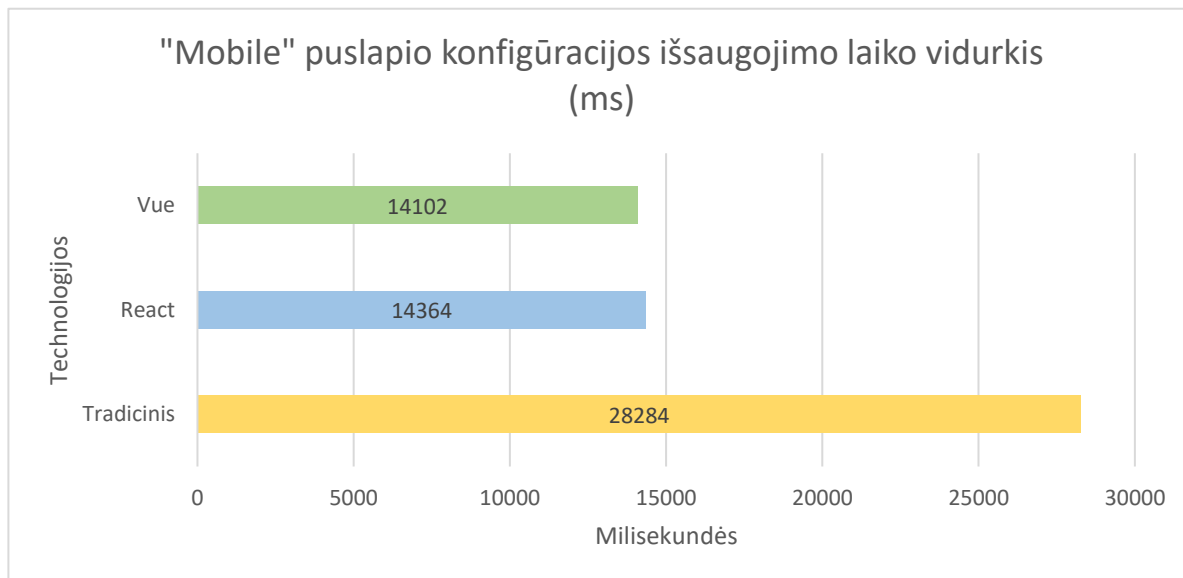
81 pav. „Mobile“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



82 pav. Vieno puslapio programų „Mobile“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

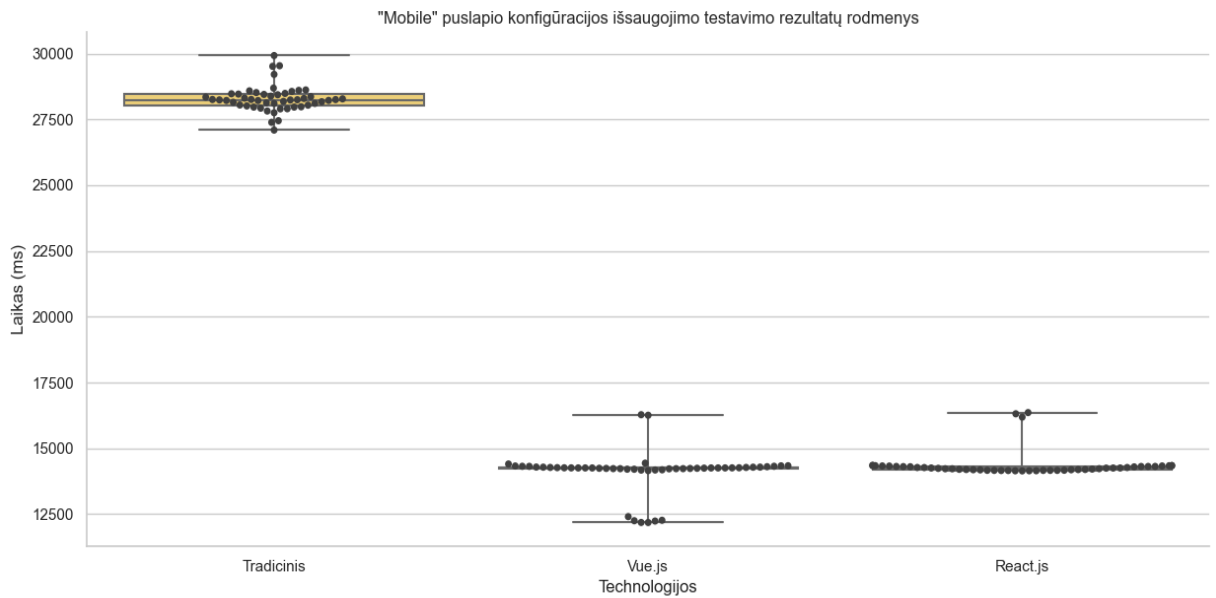
Atsižvelgiant į įverčių pasiskirstymo diagramas matoma, kad tradicinio modelio paremtos grafinės sąsajos testavimo scenarijaus įverčių mediana yra – 10961 milisekundžių. Vieno puslapio programos Vue mediana yra – 137 milisekundžių, na o React – 128 milisekundžių. Šį scenarijų React atliko greičiausiai.

Konfigūracijos saugojimo testas trunka nuo įvestos konfigūracijos išsaugojimo mygtuko „Save“ paspaudimo iki paskutinio saugojimo proceso užbaigimo, kuris susideda iš procesų vykstančių daiktų interneto maršrutizatoriaus serverio pusėje. Saugojimas serverio dalyje vykdomas priklausomybę turinčių programų perkrovimo veiksmu užtikrinant naujai išsaugotos konfigūracijos panaudojimą. 83 pav. pateikiami išsaugojimo testo rezultatai.

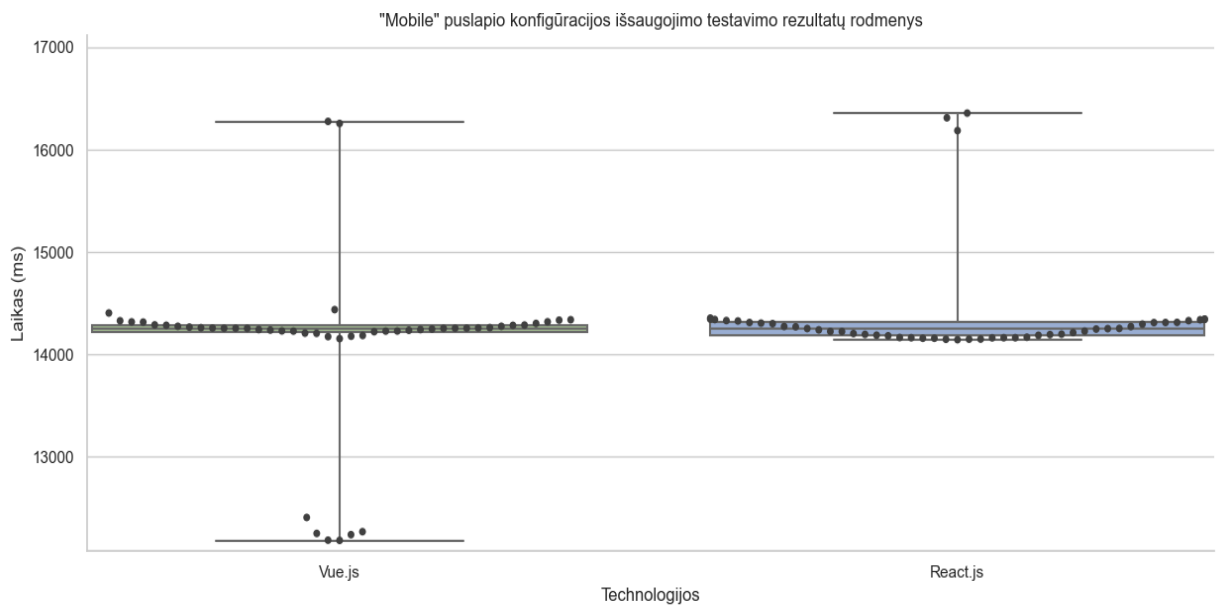


83 pav. „Mobile“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus vidurkių diagrama

Pagal pateiktus rezultatus matoma, kad Vue testą atliko per 14,1 sekundžių, React atliko per 14,3 sekundes. Vieno puslapio programos modelio informacinės sistemos scenarijų atliko 2 kartus greičiau, nei tradicinio internetinių puslapių modelio grafinė sąsaja. Tradicinio internetinių puslapių modelio grafinės sąsajos testų rezultatai atsilieka dėl modelio veikimo, kuris po mygtuko paspaudimo perkrauna puslapį reikalaujamas puslapį sugeneruoti iš naujo, kuris reikalauja daug daiktų interneto maršrutizatoriaus resursų ir laiko. Vieno puslapio programos modeliu paremtos grafinės sąsajos puslapių neperkrauna, nes puslapio pakeitimus atvaizduoja dinamiškai užsiklausiant tik pasikeitusių duomenų ir nereikalaujant serverio resursų sugeneruoti naują puslapį. 84 pav. ir 85 pav. pateikiamos „Mobile“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



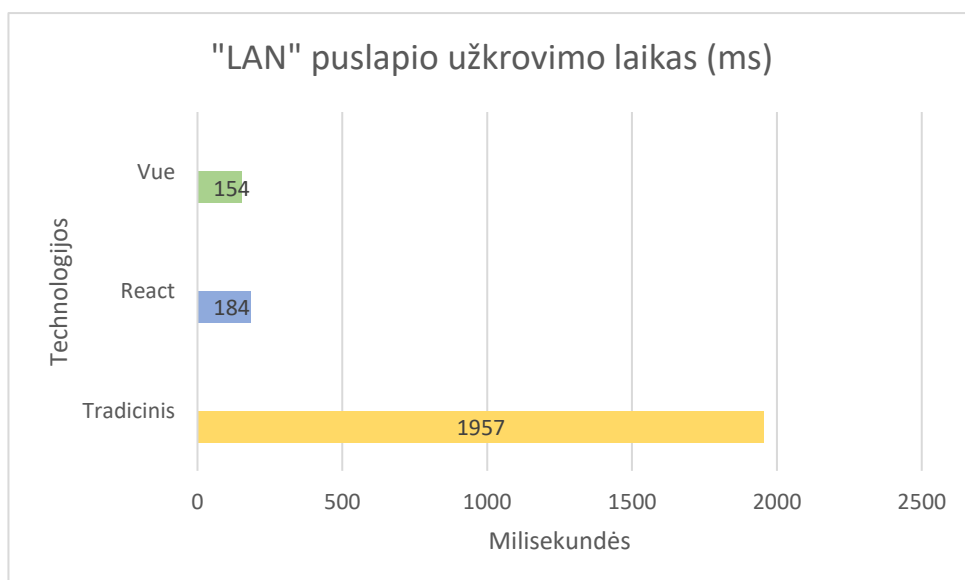
84 pav. „Mobile“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama



85 pav. Vieno puslapio programų „Mobile“ konfigūracijos testavimo scenarijaus įverčių pasiskirstymo diagrama

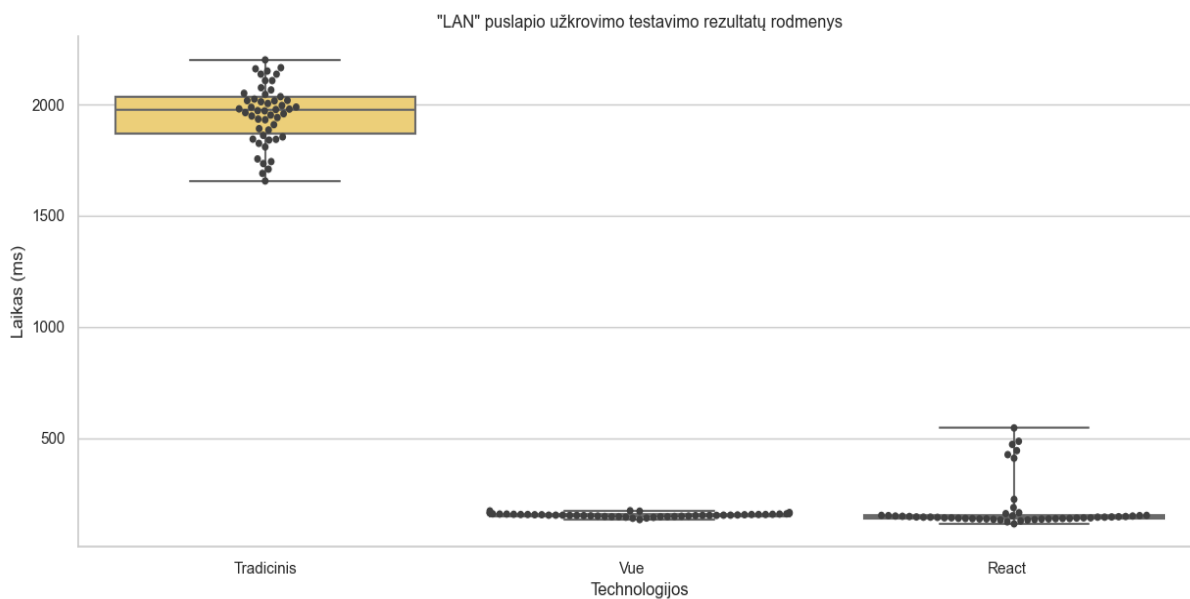
Atsižvelgiant į įverčių pasiskirstymo diagramas matoma, kad konfigūracijos išsaugojimas retkarčiais šokinėja ir rezultatai svyruoja 400 milisekundžių intervale, ypač gerai šis požymis pastebimas React ir Vue, kadangi didžioji dalis įverčių skiriasi per kelias milisekundes, o keli taškai ypač nutolę nuo kvantilių. Šio testavimo scenarijaus React mediana – 14224 milisekundžių, Vue – 14225 milisekundžių ir Tradicinio modelio informacinės sistemos mediana – 28242 milisekundžių.

Perėjimo iš mobiliųjų nustatymų puslapio „Mobile“ į vidinio tinklo „LAN“ nustatymų puslapio testavimo vidurkio rezultatai pateikiami 86 pav.

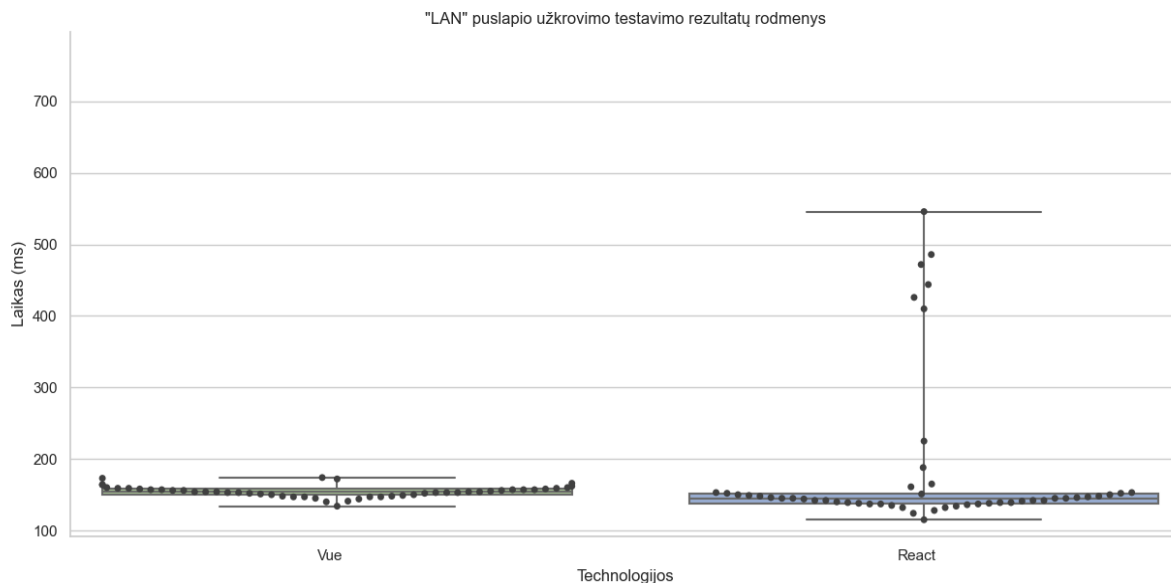


86 pav. „LAN“ puslapio užkrovimo testavimo vidurkio diagrama

Pagal pateiktus rezultatus matoma, kad vieno puslapio programos modeliu paremtos grafinės sąsajos puslapį užkrovė per mažiau nei 200 milisekundžių ir tai buvo 10 kartų greičiau, nei tradicinio modelio paremta grafinė sąsaja, kuri užtruko beveik 2 sekundes. 87 pav. ir 88 pav. pateikiamos „LAN“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



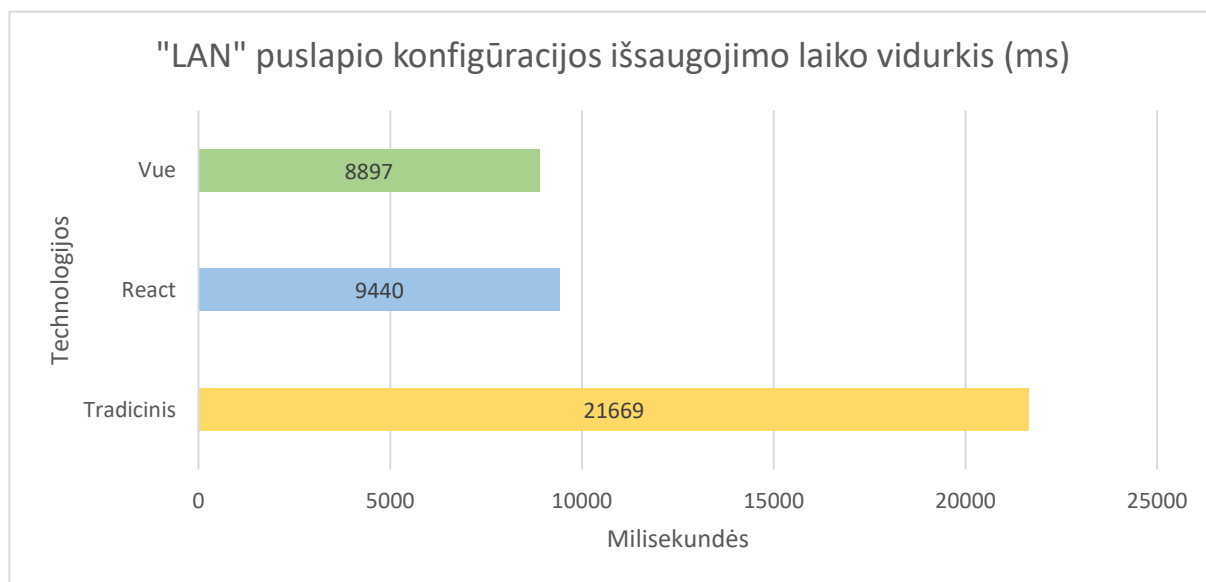
87 pav. „LAN“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



**88 pav.** Vieno puslapio programų „LAN“ užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Atsižvelgiant į įverčių pasiskirstymo diagramas matoma, kad React keli taškai yra aukščiau tiek vidurkio reikšmių tiek medianos, šie taškai galėjo atsirasti dėl testavimo aplinkos trukdžių. Tačiau React vis tiek parodė greičius „LAN“ puslapio užkrovimo rezultatus. React mediana siekė – 145 milisekundes, Vue – 154 milisekundes, o tradicinio modelio – 1974 milisekundžių.

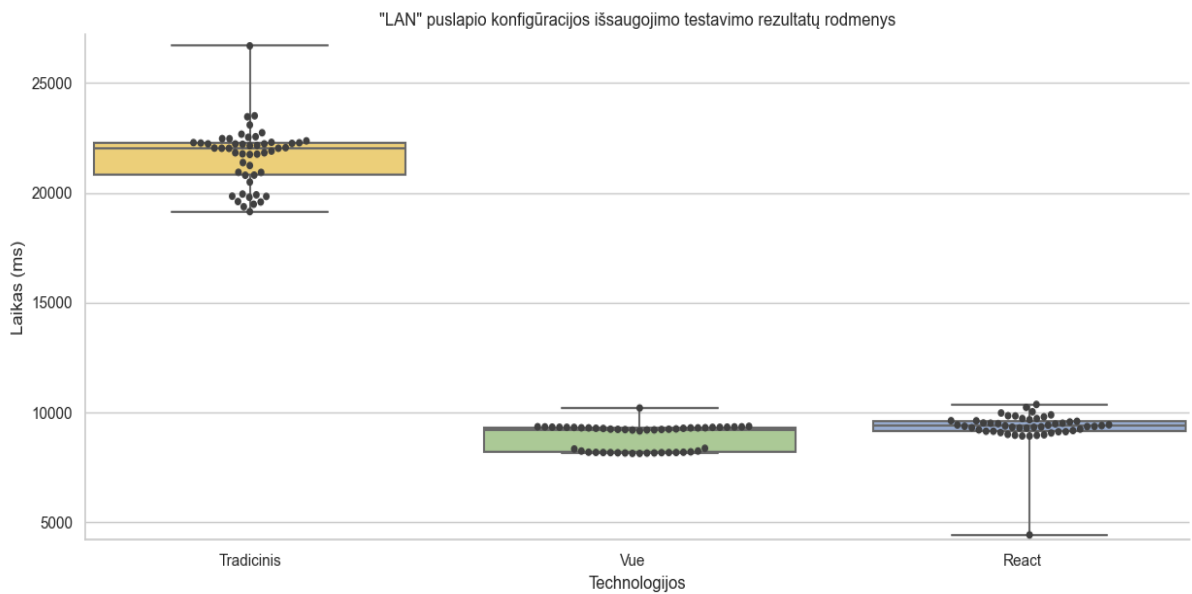
„LAN“ puslapio konfigūracijos nustatymų išsaugojimo testavimo scenarijaus rezultatai pateikiami 89 pav.



**89 pav.** „LAN“ puslapio konfigūracijos išsaugojimo testavimo vidurkio diagrama

Pagal „LAN“ puslapio konfigūracijos išsaugojimo rezultatus Vue karkaso rezultatai buvo geriausi, kurie siekė mažiau nei 9 sekundes. React bibliotekos testavimo vidurkis siekė 9,4 sekundes, o tradiciniu modeliu paremtos grafinės sąsajos konfigūracijos išsaugojimo testavimo rezultatas buvo 21,6 sekundę. Vieno puslapio programos modelio informacinės sistemos konfigūraciją išsaugojo 2

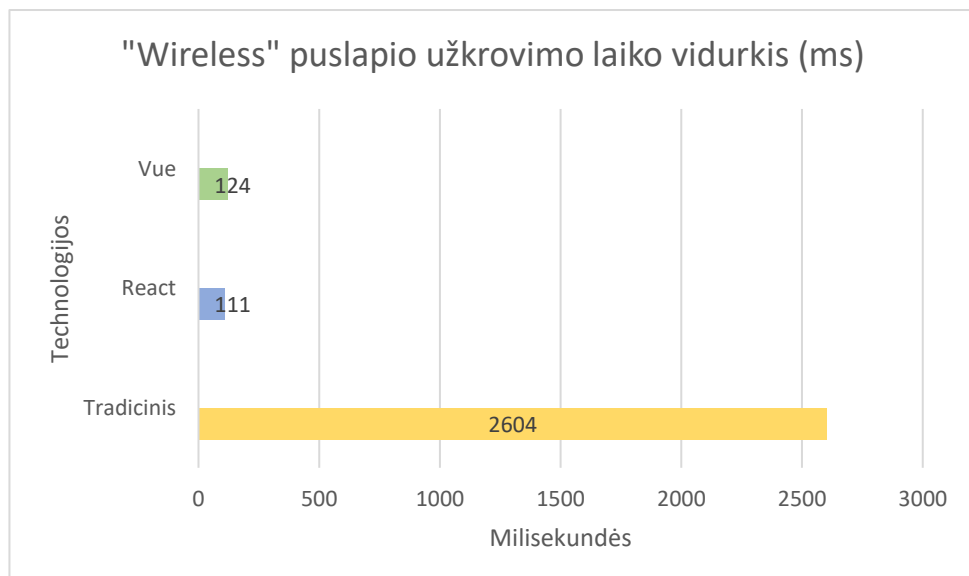
karto greičiau. 90 pav. pateikiama „LAN“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



**90 pav.** „LAN“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Pagal aukščiau esančią diagramą matoma, kad Vue testą atliko greičiausiai, šios technologijos testavimo įverčių mediana yra – 9235 milisekundžių. React mediana – 9398 milisekundžių, o tradicinio modelio mediana – 22036 milisekundžių.

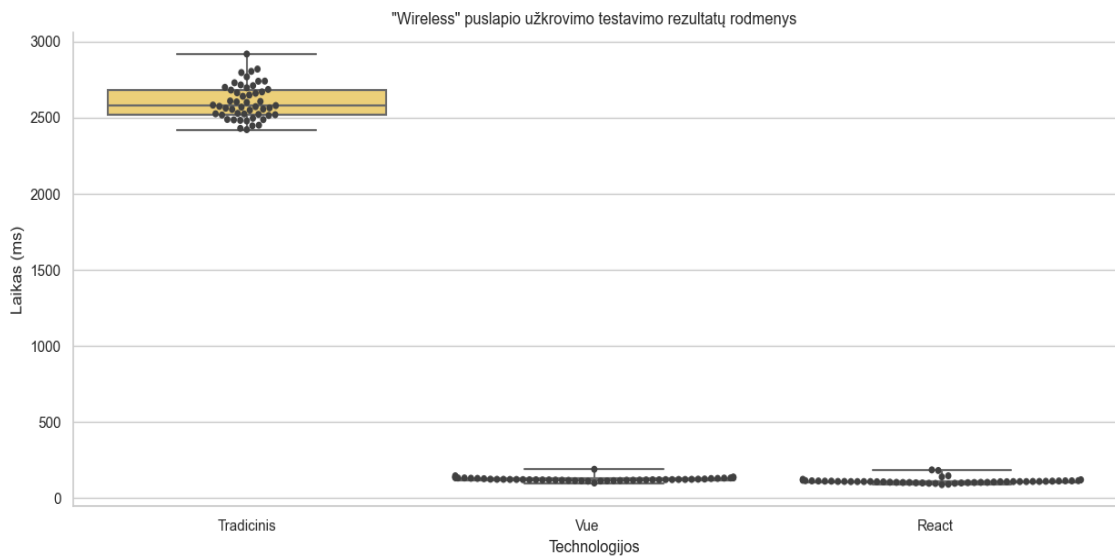
Belaidžio ryšio „Wireless“ puslapio užkrovimo testavimo scenarijaus vidurkio diagrama pateikiama 91 pav.



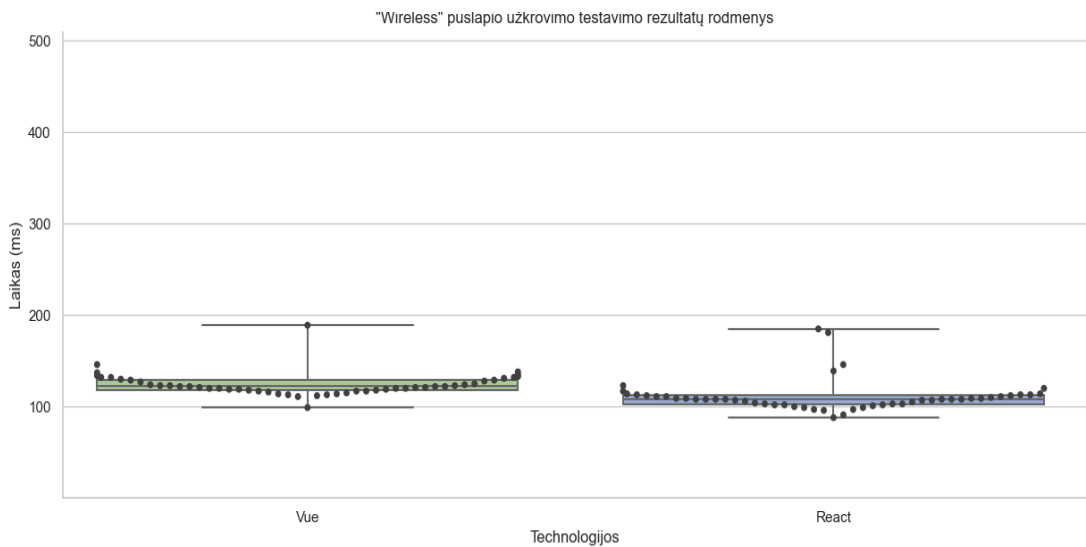
**91 pav.** „Wireless“ puslapio užkrovimo testavimo rezultatai

Vieno puslapio aplikacijos „Wireless“ puslapį užkrovė per trumpesnę laiką nei 130 milisekundžių, kai tradicinio modelio paremtai informacinėi sistemai užkrauti puslapį pririekė 2.6 sekundės. 92 pav.

ir 93 pav. pateikiamos „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



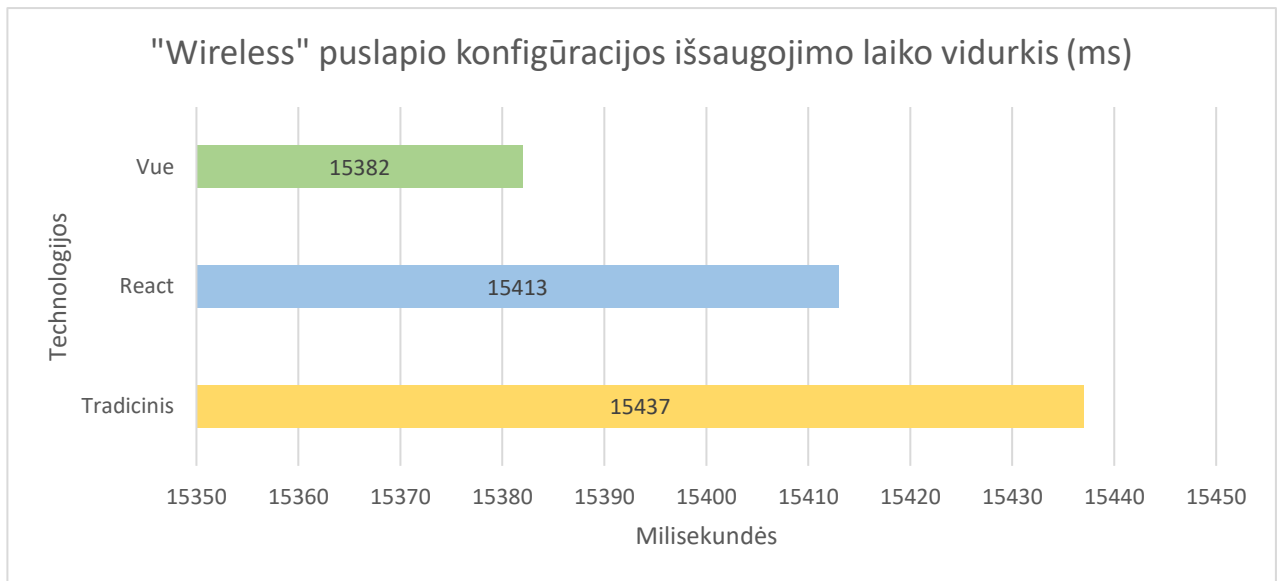
92 pav. „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



93 pav. Vieno puslapio programų „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

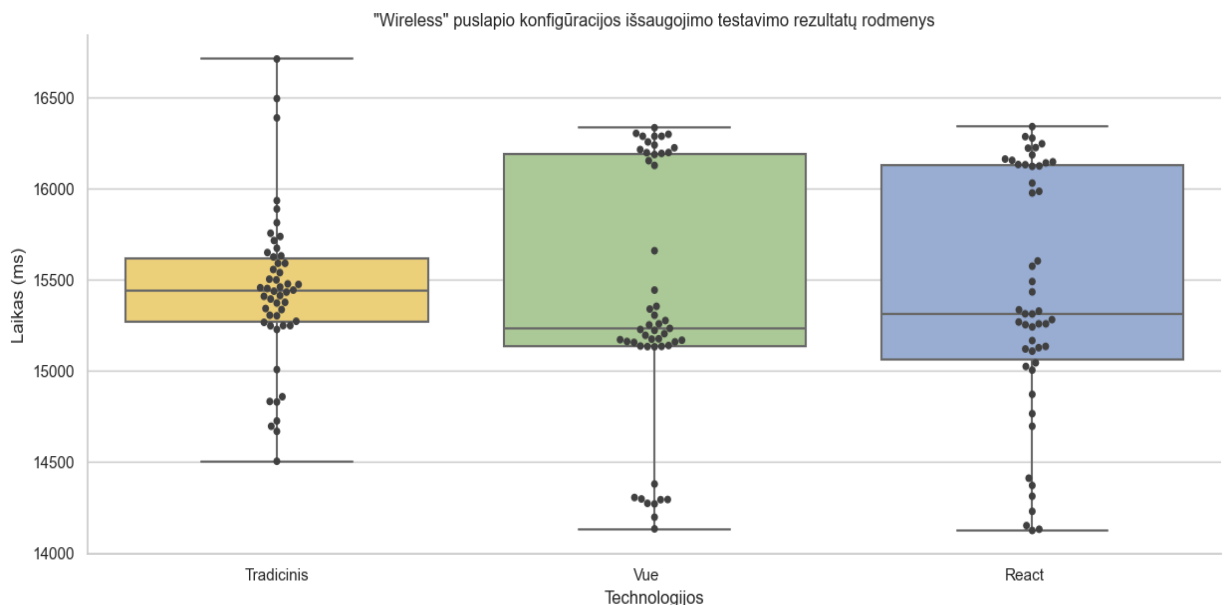
Atsižvelgiant į „Wireless“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramų rezultatus matoma, kad React užkrovė puslapį šiek greičiau. React įverčių pasiskirstymo mediana yra – 108 milisekundžių. Vue mediana yra – 122 milisekundžių, o tradicinio modelio informacinės sistemos įverčių pasiskirstymo mediana yra – 2577 milisekundžių.

„Wireless“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus vidurkio rezultatai pateikiami 94 pav.



94 pav. „Wireless“ puslapio konfigurācijas išsaugojimo testavimo rezultātu diagrama

Tradicionio modelio taikomai informacinėj sistemai konfiguraciją išsaugoti užtruko 15,4 sekundžių, React 14,4 sekundžių ir Vue – 15,3 sekundžių. Konfiguracijos saugojimo rezultatai beveik vienodi, nes šiame puslapyje po išsaugojimo mygtuko „Save“ paspaudimo tradicinėje informacinėje sistemoje nėra perkraunamas puslapis kaip kituose konfiguracijos saugojimo scenarijuose. Atsižvelgus į praėjusius konfiguracijos saugojimo scenarijų rezultatus galime pastebėti, kad kiekvienas naujo puslapio generavimas tradiciniu modeliu paremtai informacinėj sistemai kainuoja daug laiko. 95 pav. pateikiama „Wireless“ puslapio konfiguracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.

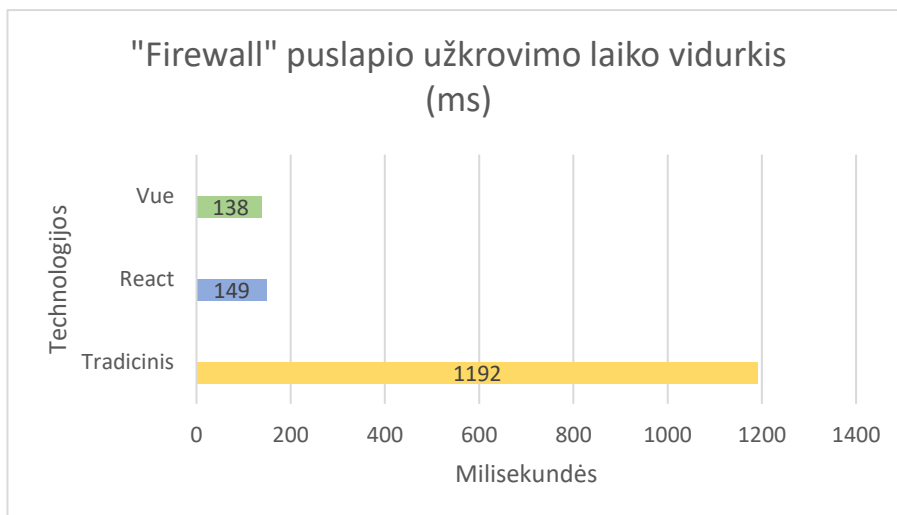


95 pav. „Wireless“ puslapio konfigurācijas išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama



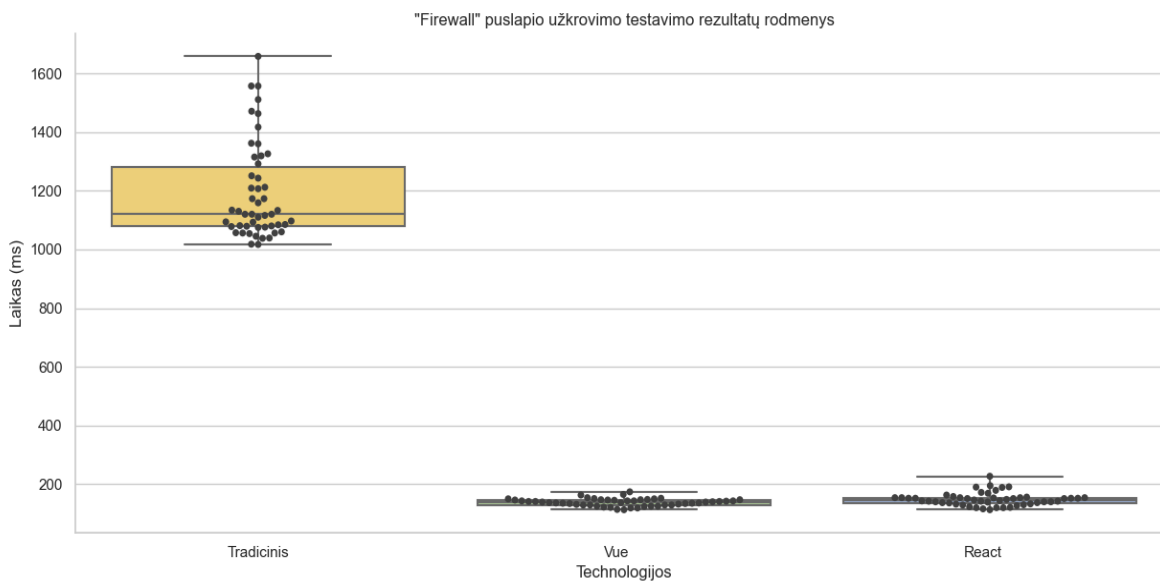
Atsižvelgus į įverčių pasiskirstymo diagrama, matoma, kad konfigūracijos saugojimas svyruoja 1500 milisekundžių intervale ir nėra daug beveik vienodų taškai kaip ankstesnėse diagramose. Tačiau šis svyravimas vyksta dėl maršrutizatoriuje vykstančios konfigūracijos saugojimo logikos, kuri turi perkrauti visus su belaidžiu ryšiu susijusius servisu, kas apkrauna patį maršrutizatorių ir konfigūracijos išsaugojimas svyruoja. Vue mediana – 15230 milisekundžių, React mediana – 15313 milisekundžių, tradicinio mediana – 14440 milisekundžių.

96 pav. pateikiama „Firewall“ puslapio užkrovimo testavimo laiko vidurkio rezultatų diagrama.

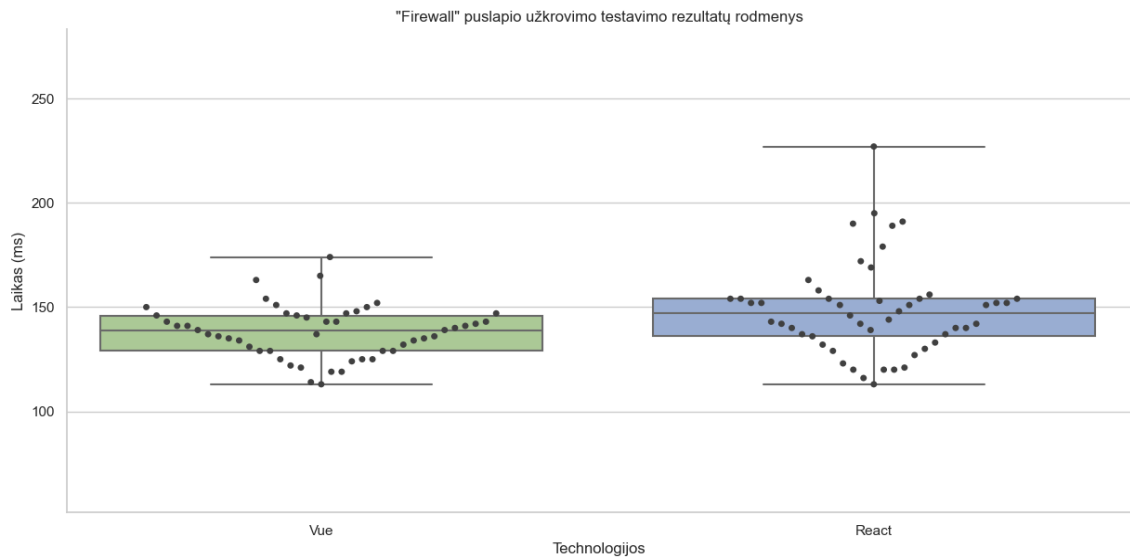


96 pav. „Firewall“ puslapio užkrovimo testavimo rezultatai

Ugniasienės puslapį „Firewall“ vieno puslapio programos modeliams užkrauti prireikė iki 150 milisekundžių, kai tradicinio modelio paremtai grafinėj sąsajai prireikė 1.2 sekundžių. Vieno puslapio programos modeliu paremtos grafinės sąsajos puslapį užkrovė 6 kartus greičiau. 97 pav. ir 98 pav. pateikiamos „Firewall“ puslapio užkrovimo testavimo įverčių pasiskirstymo rezultatų diagramos.



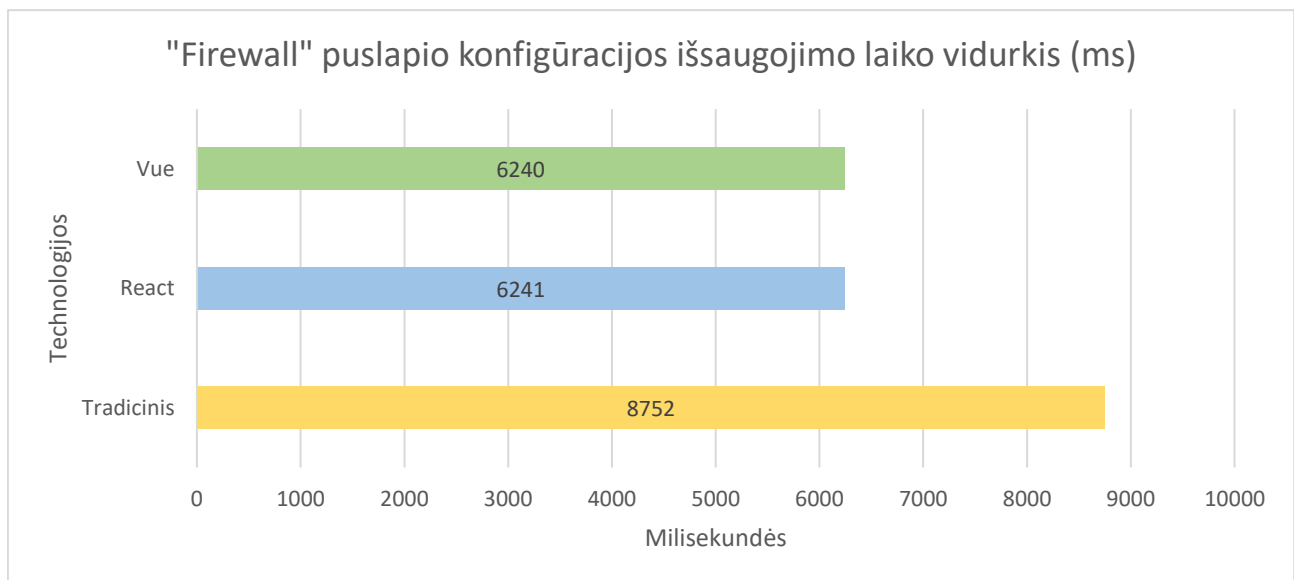
97 pav. „Firewall“ puslapio užkrovimo testavimo įverčių pasiskirstymo rezultatų diagrama



**98 pav.** Vieno puslapio programos modelių „Firewall“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

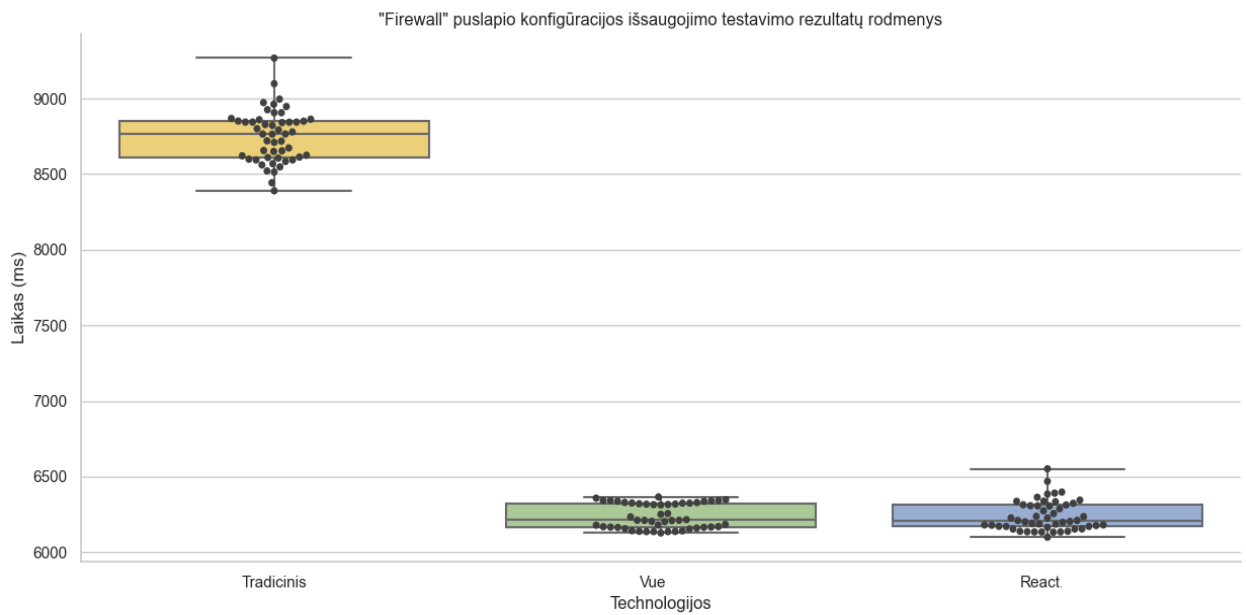
Atsižvelgiant į „Firewall“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo rezultatus matoma, kad Vue mediana buvo – 139 milisekundžių, React – 147 milisekundžių, o tradicinio modelio – 1120 milisekundžių.

„Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus laiko vidurkio rezultatai pateikiami 99 pav.



**99 pav.** „Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus vidurkio diagrama

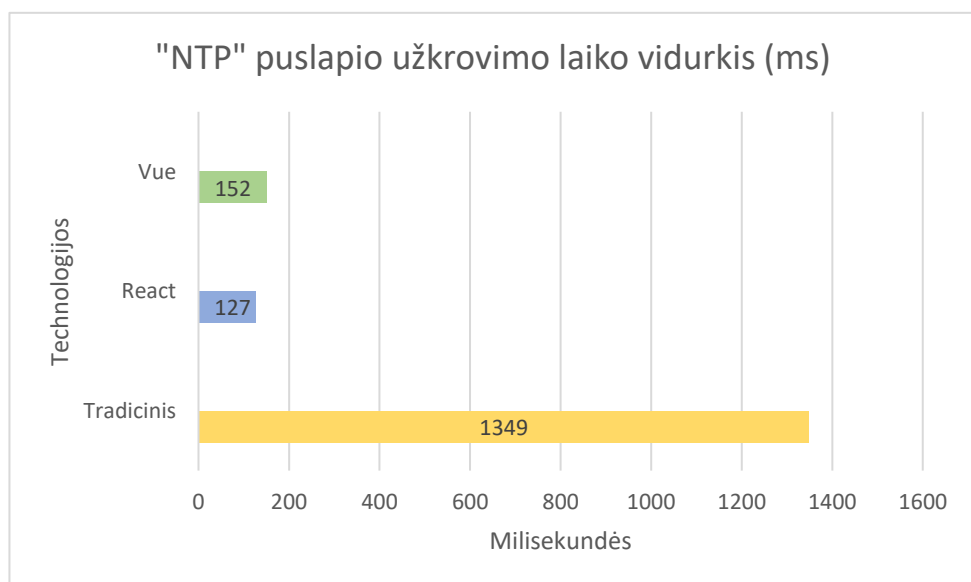
„Firewall“ puslapio konfigūracijos išsaugojimo vidurkis tarp Vue ir React buvo 1 milisekundė, t.y., 6240 milisekundžių ir 6241 milisekundžių, o tradicinio modelio informacinė sistemai prirėkė 8752 milisekundžių. Vieno puslapio programos modelio informacinės sistemos konfigūraciją išsaugojo 1.4 karto greičiau. 100 pav. pateikiama „Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



**100 pav.** „Firewall“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Atsižvelgiant į „Firewall“ puslapio konfigūracijos išsaugojimo įverčių pasiskirstymo diagramą matoma, kad nors ir Vue rezultatas buvo 1 milisekundę mažesnis, tačiau React mediana buvo – 6206 milisekundžių, o Vue – 6212 milisekundžių. Tradicinio modelio mediana buvo – 8765 milisekundžių.

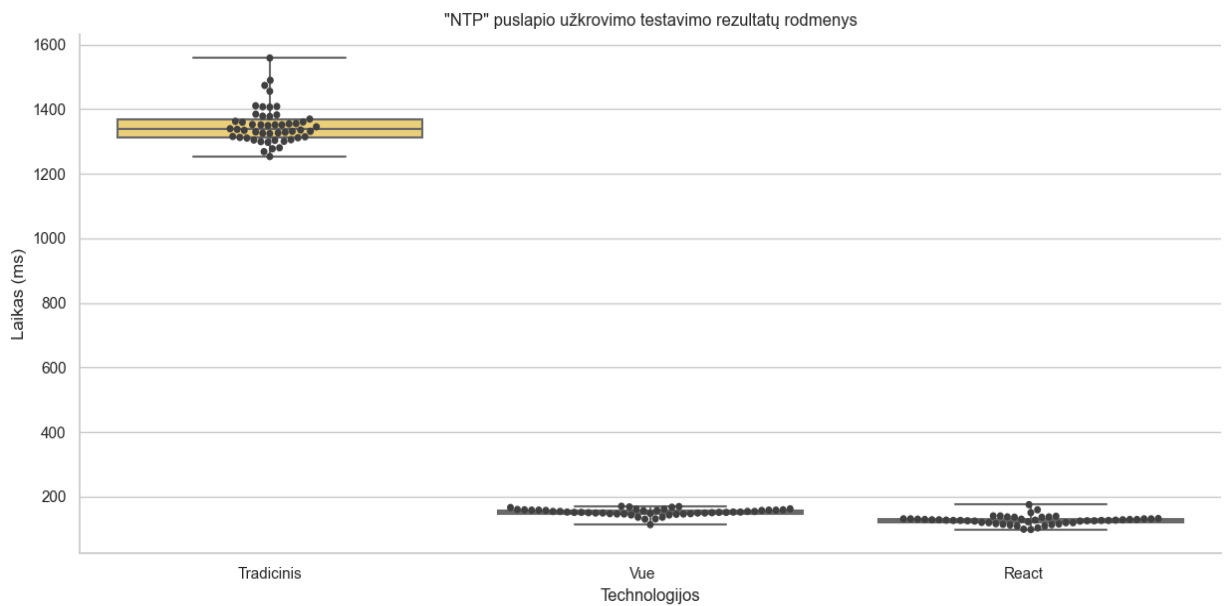
101 pav. pateikiama sistemos laiko „NTP“ nustatymų puslapio užkrovimo testavimo rezultatai.



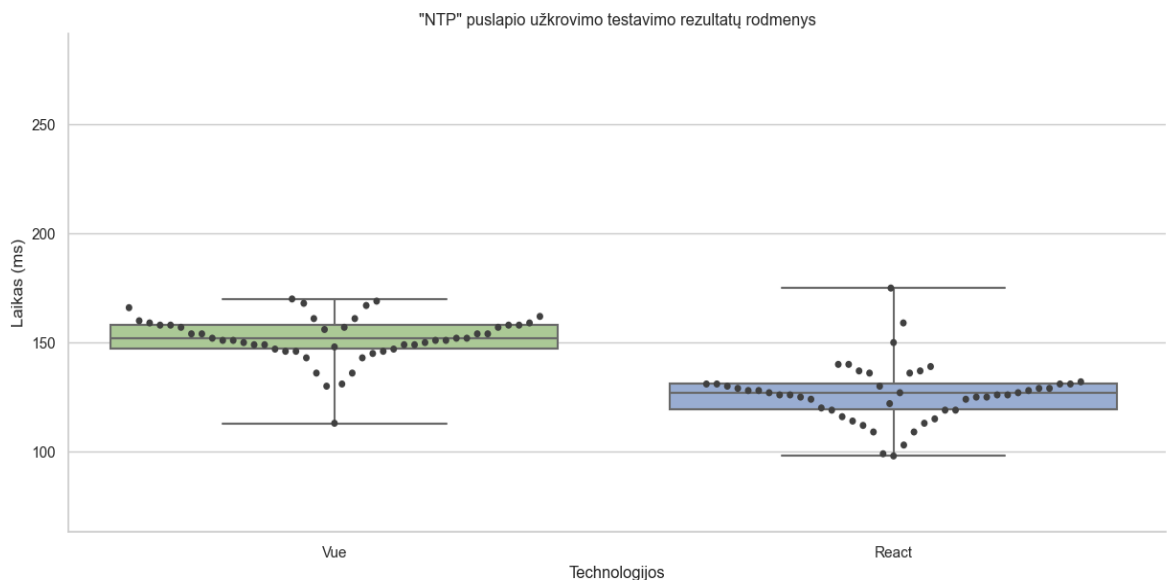
**101 pav.** „NTP“ puslapio užkrovimo testavimo rezultatai

Sistemos laiko „NTP“ nustatymų puslapį vieno puslapio programos modeliu paremtos informacinės sistemos užkrovė mažiau nei 160 milisekundžių, kai tradicinio modelio informacinėi sistemai prireikė 1,4 sekundės. Tradiciniu modeliu paremta informacinė sistema „NTP“ puslapį užkrovė 8 kartus

lėčiau, nei vieno puslapio programos modelio informacinės sistemos. 102 pav. ir 103 pav. pateikiamos „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



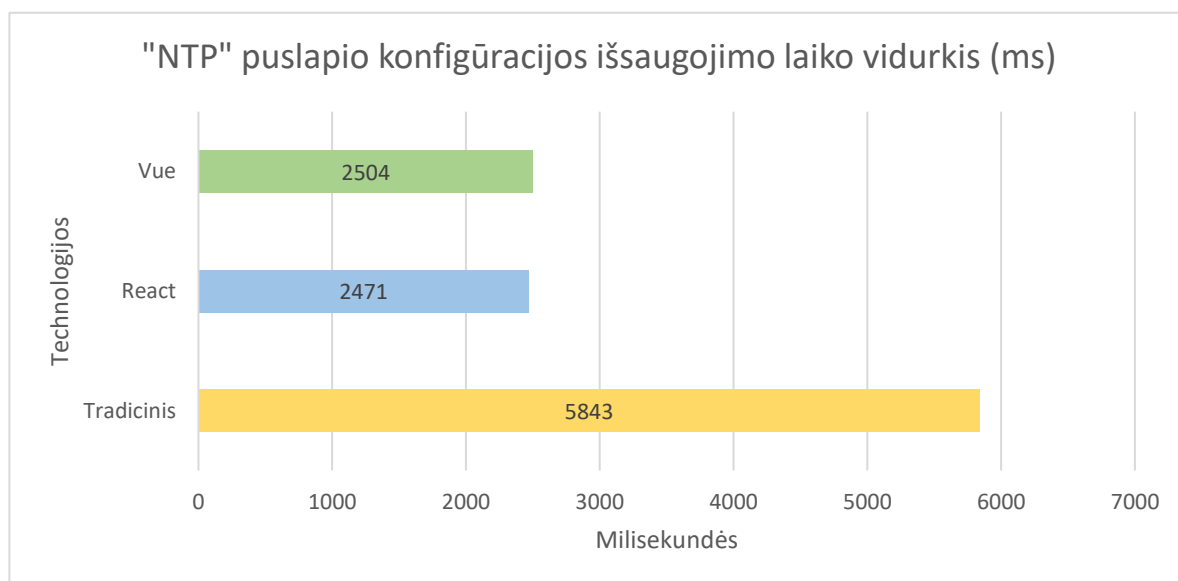
**102 pav.** „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



**103 pav.** Vieno puslapio programų „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

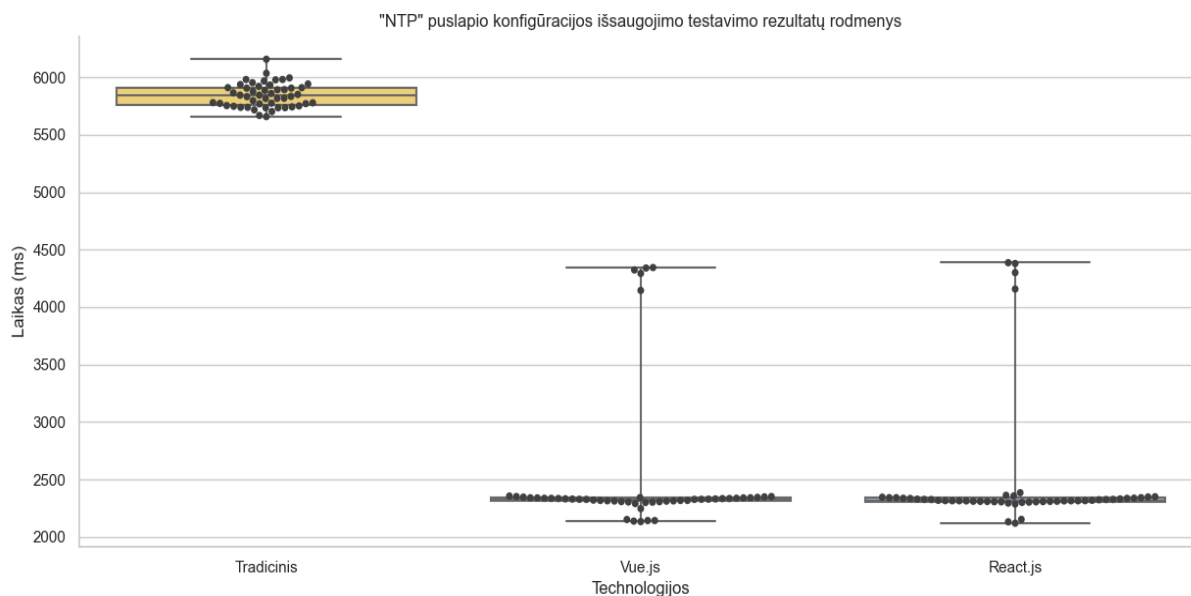
Atsižvelgus į „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos rezultatus matoma, kad React technologijos netik vidurkis, bet ir mediana buvo mažesnė nei Vue. React mediana – 127 milisekundžių, Vue mediana – 152 milisekundžių, o tradicinio modelio mediana – 1337 milisekundžių.

„NTP“ puslapio konfigūracijos išsaugojimo testavimo laiko vidurkių rezultatų diagrama pateikiama 104 pav.



104 pav. „NTP“ puslapio konfigūracijos išsaugojimo testavimo rezultatai

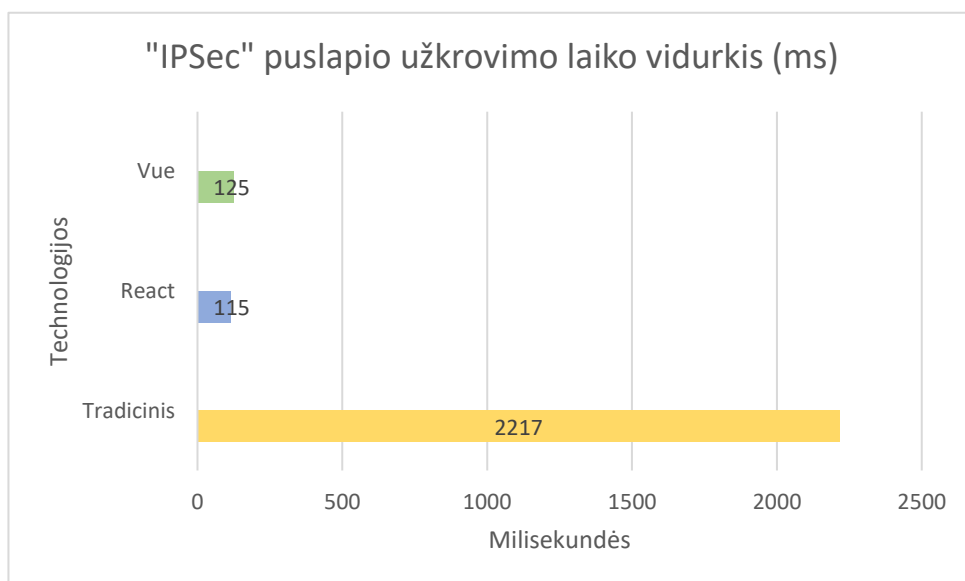
Vieno puslapio programos modelio išsaugojimo rezultatai siekė apie 2,5 sekundės, kai tradicinio modelio informacinė sistema užtruko beveik 6 sekundės. Vieno puslapio programos modelio informacinės sistemos konfigūraciją išsaugojo 2,3 karto greičiau. 105 pav. pateikiama „NTP“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



105 pav. „NTP“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

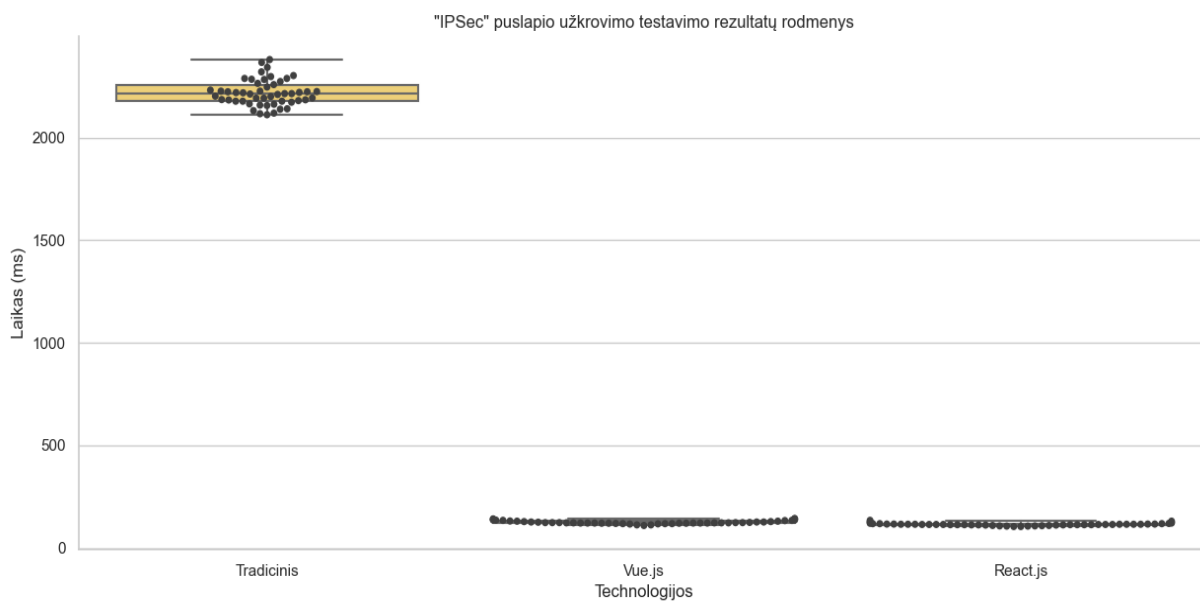
Pagal „NTP“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagramą matoma, kad React mediana yra žemiausių reikšmių. React mediana – 2319 milisekundžių, Vue mediana – 2329 milisekundžių, o tradicinio modelio mediana – 5839 milisekundžių.

Virtualaus vidinio tinklo „IPSec“ puslapio užkrovimo testavimo vidurkių rezultatų diagrama pateikiama 106 pav.

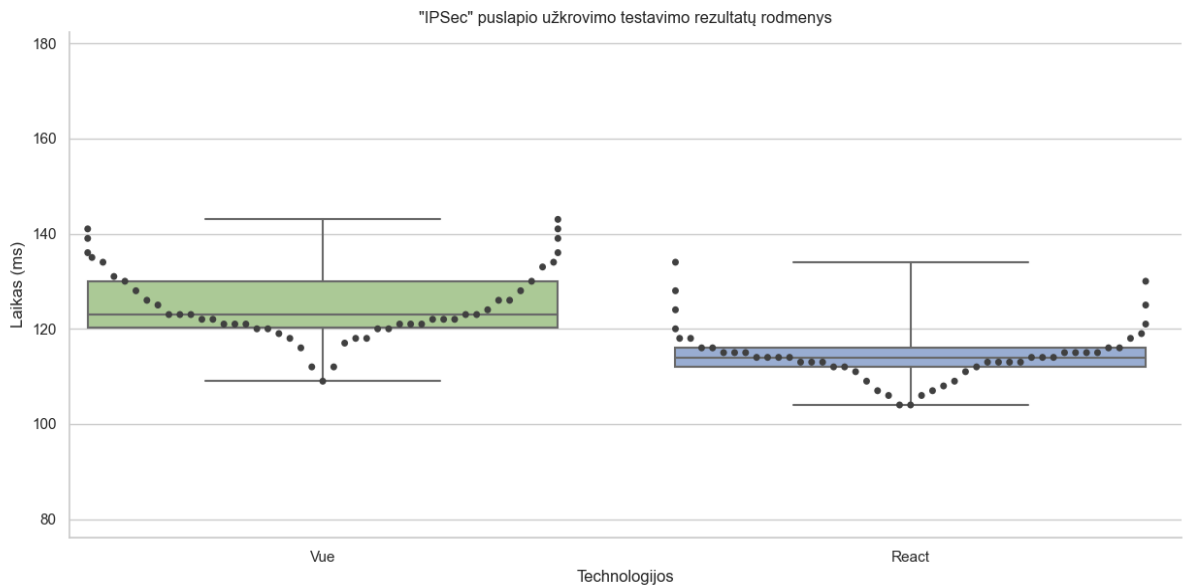


106 pav. „IPSec“ puslapio užkrovimo testavimo rezultatai

„IPSec“ puslapį užkrauti vieno puslapio programos modeliams prireikė ne daugiau nei 130 milisekundžių, kai tradicinio modelio paremtai informacinė sistemai prireikė šiek tiek daugiau nei 2,2 sekundės. Vieno puslapio programos modelio „IPSec“ puslapio užkrovimo laikas užtruko 17 kartų greičiau, nei tradicinio modelio paremtai informacinė sistemai. 107 pav. ir 108 pav. pateikiamos „IPSec“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



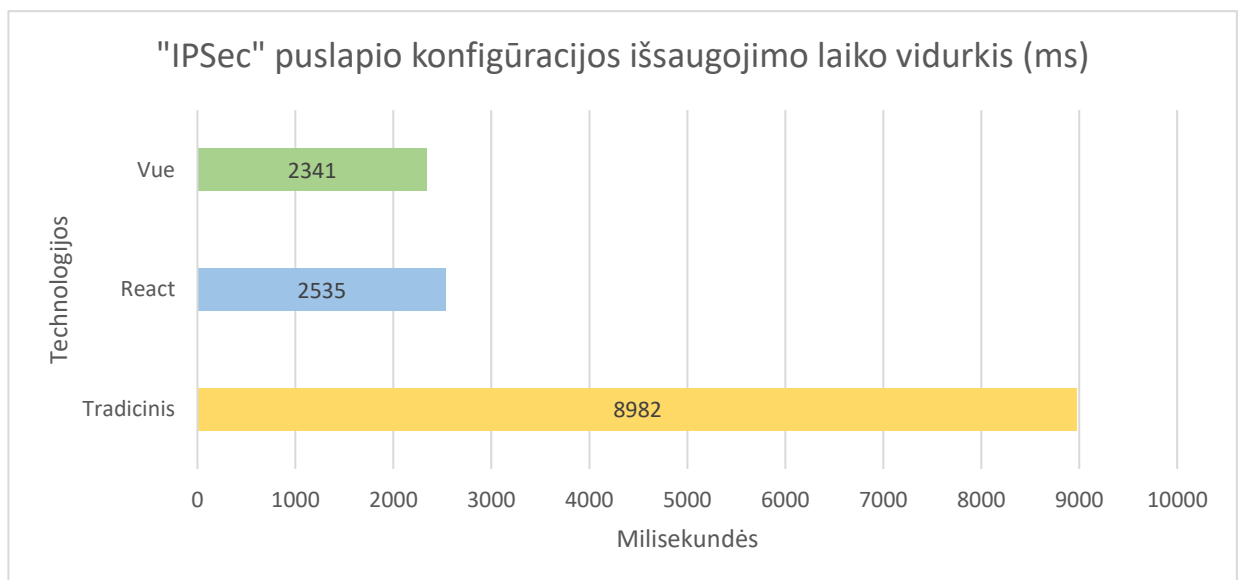
107 pav. „IPSec“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



**108 pav.** Vieno puslapio programų „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Atsižvelgiant į „NTP“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo rezultatus matoma, kad React užfiksavo greičiausius laikus. React mediana siekė – 114 milisekundes, Vue mediana – 123 milisekundes, o tradicinio modelio mediana buvo – 2213 milisekundžių.

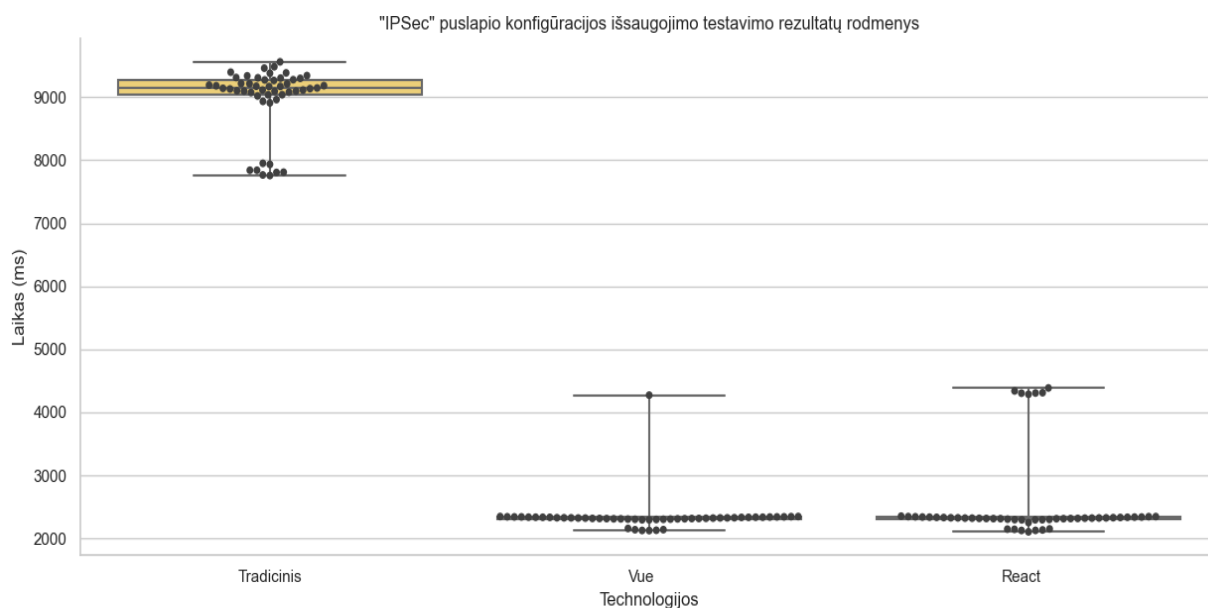
„IPSec“ puslapio konfigūracijos išsaugojimo testavimo vidurkių rezultatų diagrama pateikiama 109 pav.



**109 pav.** „IPSec“ puslapio konfigūracijos išsaugojimo testavimo rezultatai

Virtualaus vidinio tinklo „IPSec“ nustatymų išsaugojimas tradicinio modelio informacinėje sistemai užtruko beveik 9 sekundes. Vieno puslapio programos modeliams konfigūracijos išsaugojimas užtruko ne daugiau, nei 2.6 sekundės. Vieno puslapio programos modeliams konfigūracijos

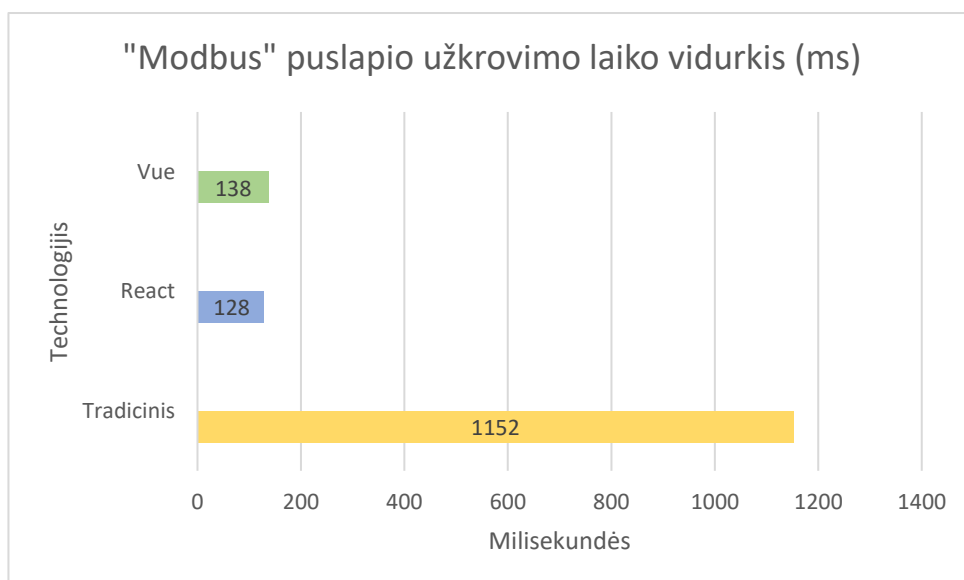
išsaugojimas užtruko 3,5 karto greičiau. 110 pav. pateikiama „IPSec“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



**110 pav.** „IPSec“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Pagal „IPSec“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama matoma, kad tradicinio modelio įverčių mediana yra – 9150 milisekundžių. Vieno puslapio programos modeliu paremtų informacinių sistemų medianos vienodos – 2322 milisekundžių.

Komunikacijos protokolo „Modbus“ puslapio užkrovimo testavimo laiko vidurkio rezultatų diagrama pateikiama 111 pav.

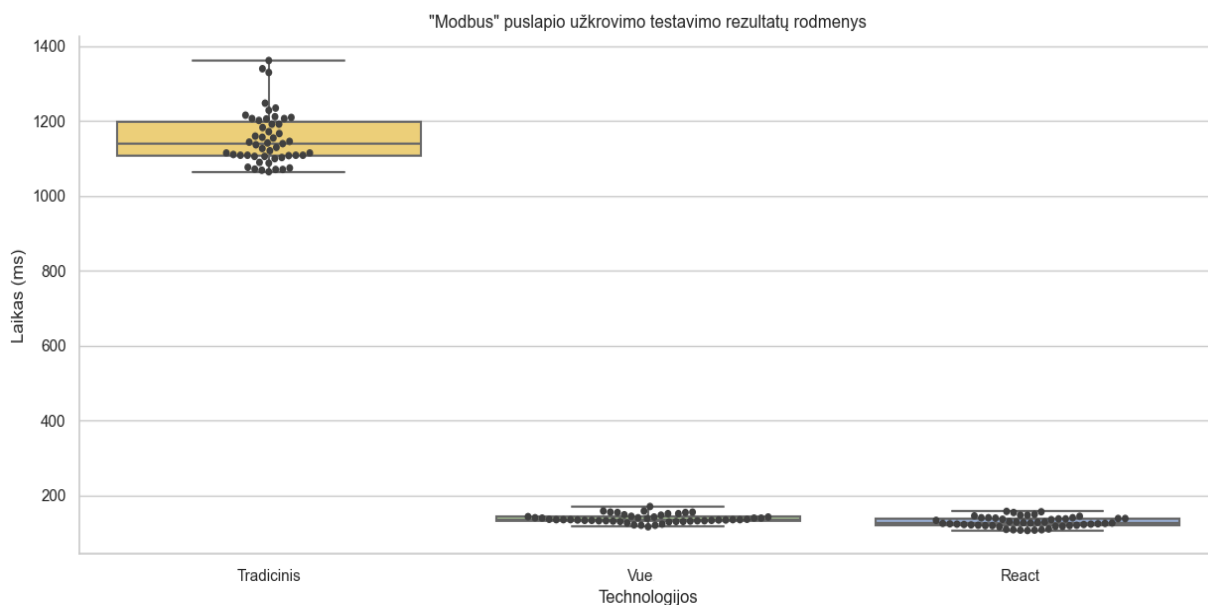


**111 pav.** „Modbus“ puslapio užkrovimo testavimo scenarijaus rezultatų diagrama

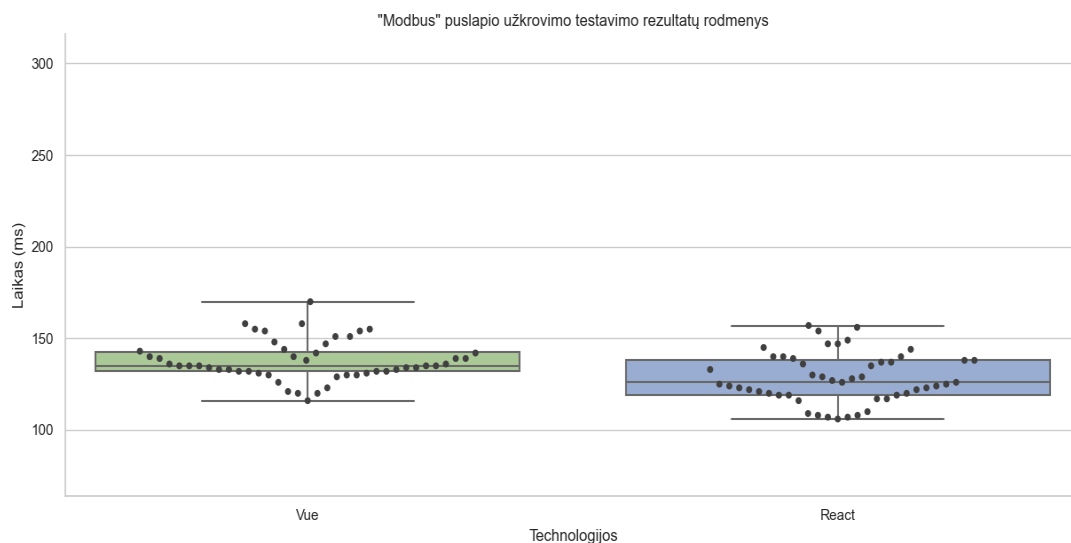
„Modbus“ puslapį užkrauti vieno puslapio programos modeliams prireikė ne daugiau, nei 140 milisekundžių, kai tradicinio modelio paremtai informacinėi sistemai prireikė beveik 1,2 sekundės. Vieno puslapio programoms puslapį užkrauti užtruko 8 kartus greičiau, nei tradicinio modelio



informacinėj sistemai. 112 pav. ir 113 pav. pateikiamos „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



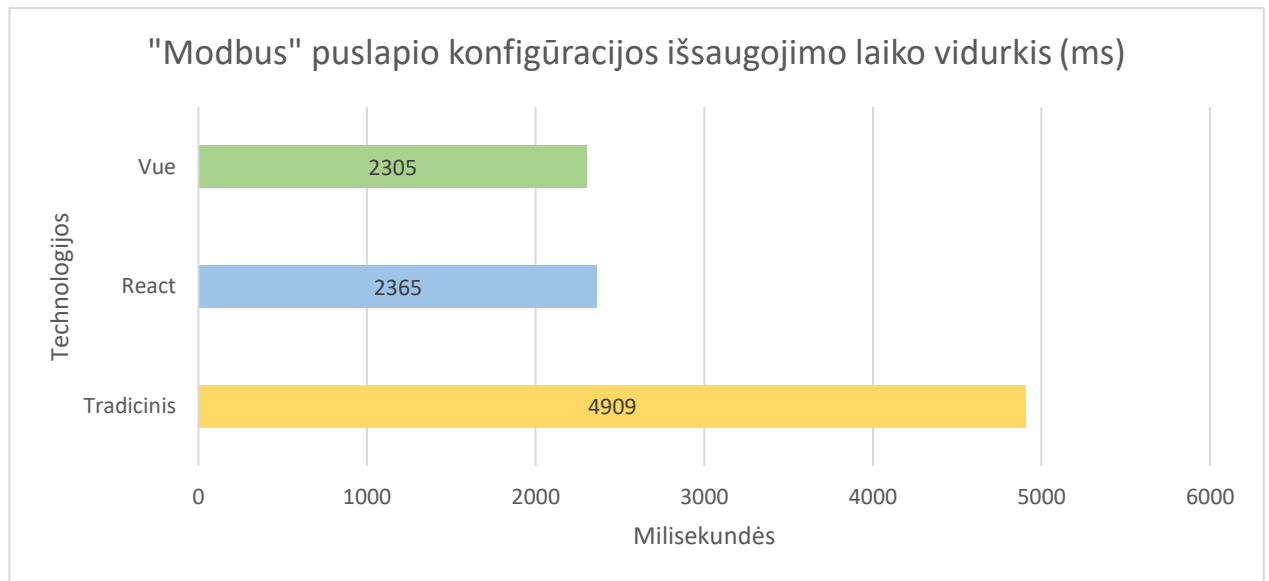
**112 pav.** „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



**113 pav.** Vieno puslapio programos „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

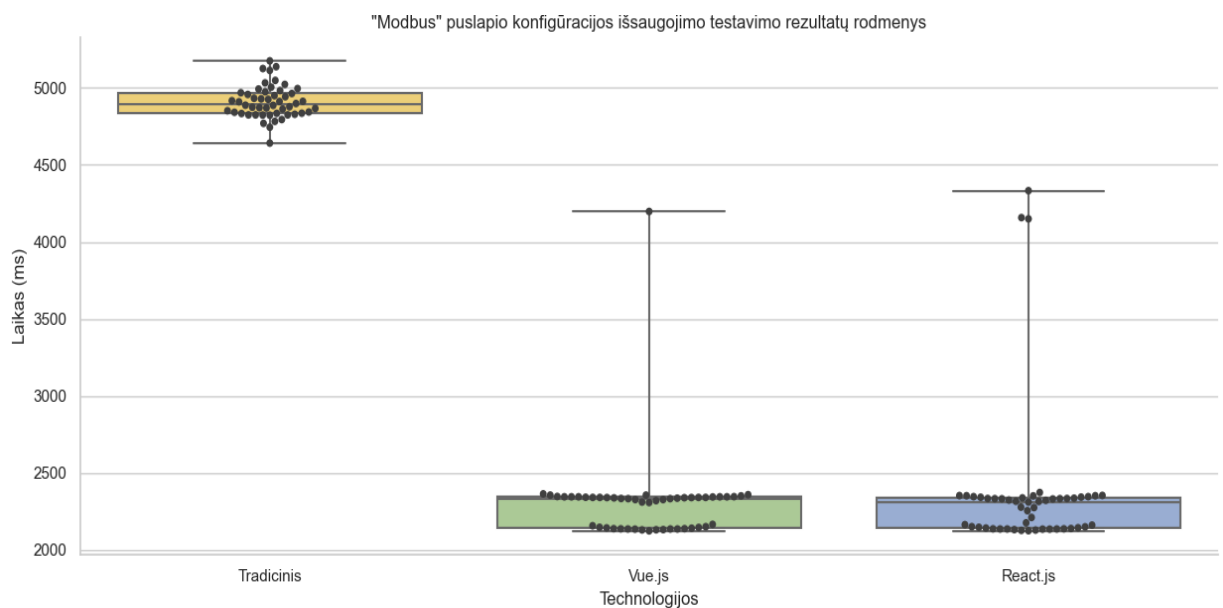
Atsižvelgiant į „Modbus“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos rezultatus matoma, kad React rezultatai greičiausi. React mediana yra – 126 milisekundžių, Vue mediana – 135 milisekundžių, o tradicinio modelio mediana – 1137 milisekundžių.

114 pav. pateikiama komunikacijos protokolo „Modbus“ nustatymų išsaugojimo scenarijaus laiko vidurkio diagrama



**114 pav.** „Modbus“ puslapio nustatymu iĥsaugojimo testavimo diagrama

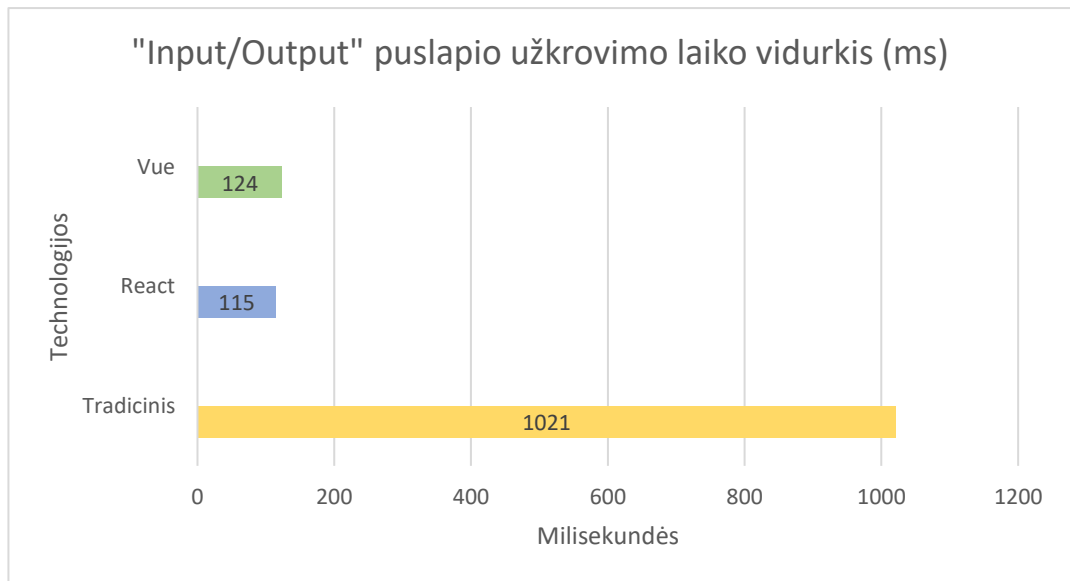
Komunikācijas protokola „Modbus“ konfigurācijas iĥsaugojimo tradicioniu modeġiu paremtai informacinēj sistēmai uĥtruko 4,9 sekundēs. Viena puslapio programos modeġiams prirēķē maĥiau nei 2.4 sekundzīu. Viena puslapio programos informacinēs sistēmas konfigurāciju iĥsaugojo 2 kartus greiĥiau, nei tradicionio modeġio informacinē sistēma. 115 pav. pateikta „Modbus“ konfigurācijas iĥsaugojimo testavimo scenārijaus ģverĥu pasiskirstymo diagrama.



**115 pav.** „Modbus“ konfigurācijas iĥsaugojimo testavimo scenārijaus ģverĥu pasiskirstymo diagrama

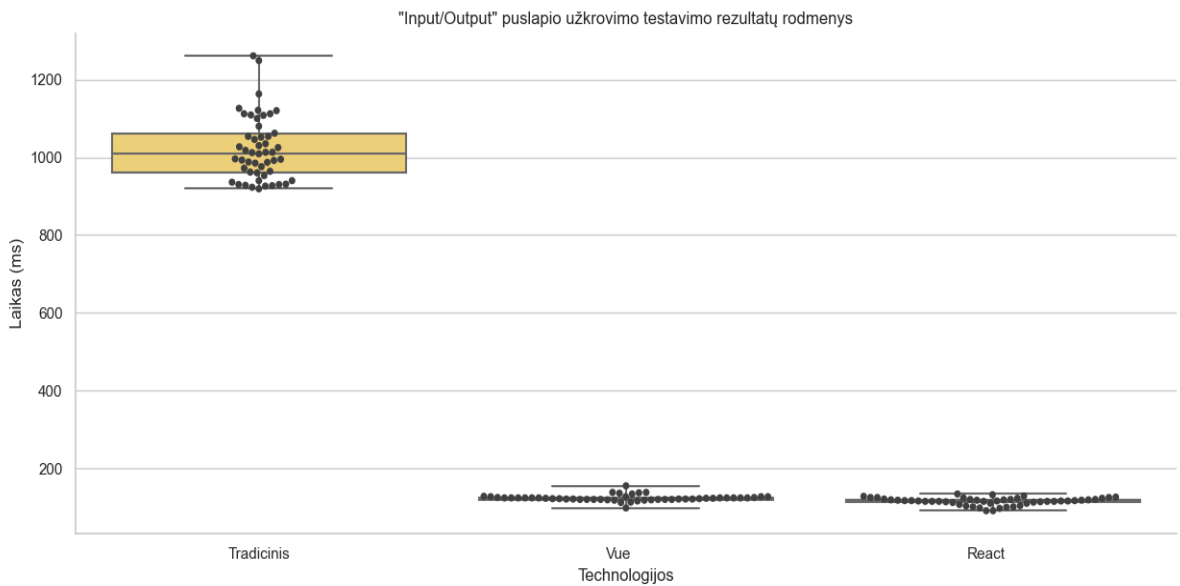
Atsiĥvelģiant ģ „Modbus“ konfigurācijas iĥsaugojimo testavimo scenārijaus ģverĥu pasiskirstymo diagramas rezultātus matoma, ka React mediana ģra – 2313 milisekundzīu, Vue mediana ģra – 2333 milisekundzīu, o tradicionio modeġio mediana – 4894 milisekundzīu.

Maršrutizatoriaus ģvesties ir iĥvesties „Input/Output“ puslapio uĥzkrovimo testavimo scenārijaus laiko vidurķio rezultāti pateikti 116 pav.

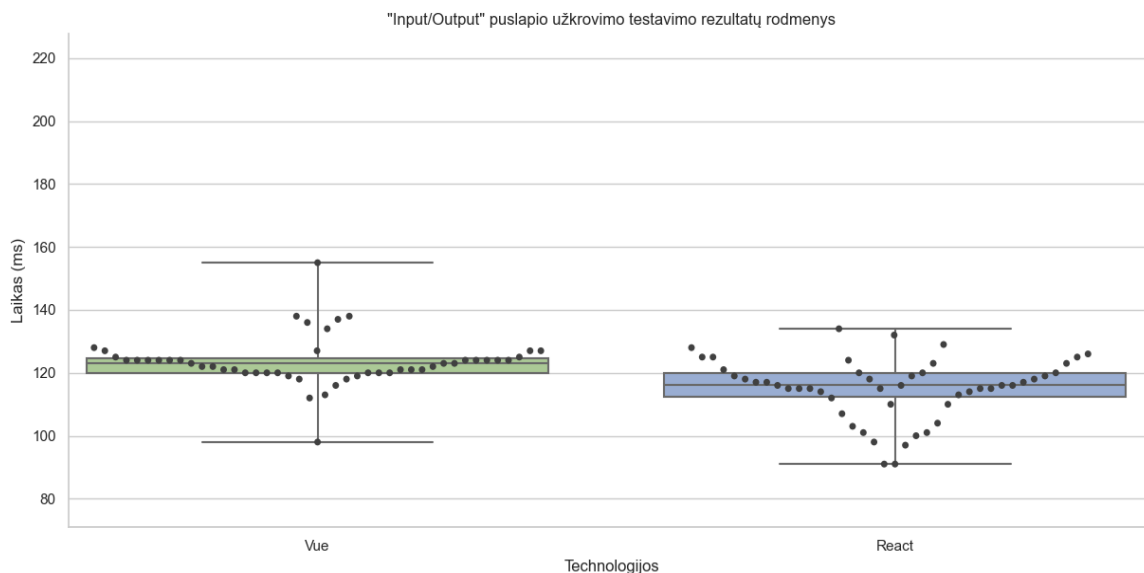


**116 pav.** „Input/Output“ puslapio užkrovimo testavimo rezultatai

Įvesties ir išvesties puslapio užkrovimas vieno puslapio programos užtruko ne daugiau, nei 125 milisekundžių, kai tradicinio modelio informacinėi sistemai prirėikė 1021 milisekundžių. Vieno puslapio programos puslapį užkrovė 9 kartus greičiau. 117 pav. ir 118 pav. pateikiamos „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramos.



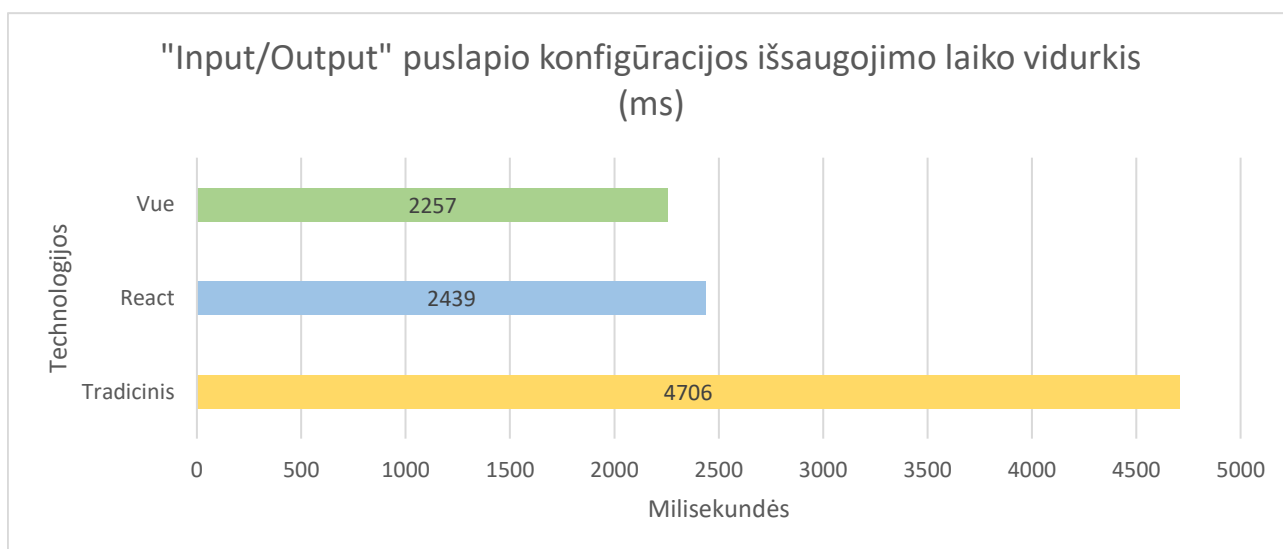
**117 pav.** „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama



**118 pav.** Vieno puslapio programos modelio „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagrama

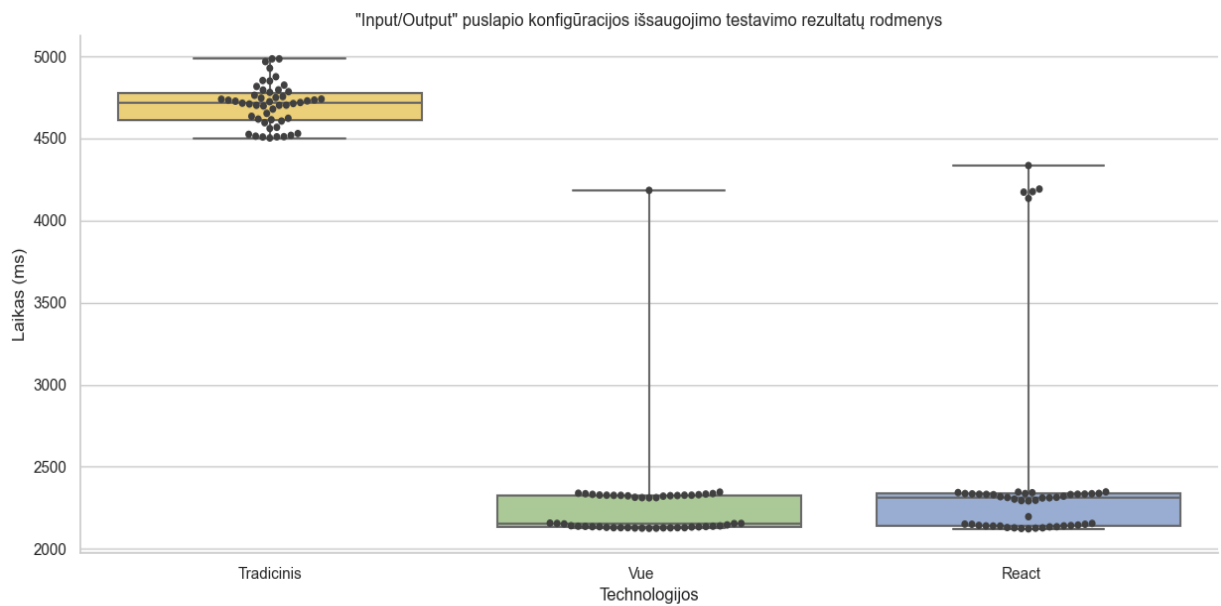
Atsižvelgiant į „Input/Output“ puslapio užkrovimo testavimo scenarijaus įverčių pasiskirstymo diagramą matoma, kad tradicinio modelio mediana yra – 1010 milisekundžių, Vue mediana – 123 milisekundžių, o React mediana – 116 milisekundžių.

119 pav. pateikiami „Input/Output“ konfigūracijos išsaugojimo scenarijaus laiko vidurkio testavimo rezultatai.



**119 pav.** „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo rezultatai

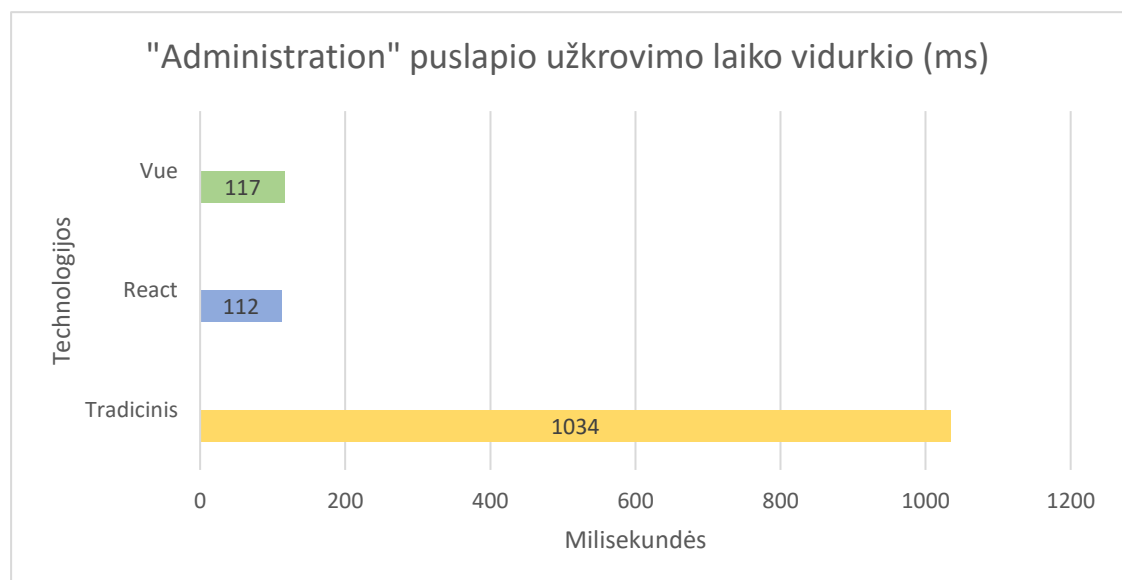
Pagal 120 pav. matomus rezultatus vieno puslapio programos modeliams konfigūraciją išsaugoti užtruko ne daugiau nei 2.5 sekundės, kai tradicinio modelio parentai informacinėj sistemai prireikė 4,7 sekundžių. Vieno puslapio programos modeliu parentos informacinės sistemos konfigūraciją išsaugojo 2 kartus greičiau. 120 pav. pateikiama „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.



**120 pav.** „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Pagal „Input/Output“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagramą matoma, kad tradicinio modelio mediana yra – 4715 milisekundžių, React mediana – 2311 milisekundžių, o Vue mediana – 2154 milisekundžių.

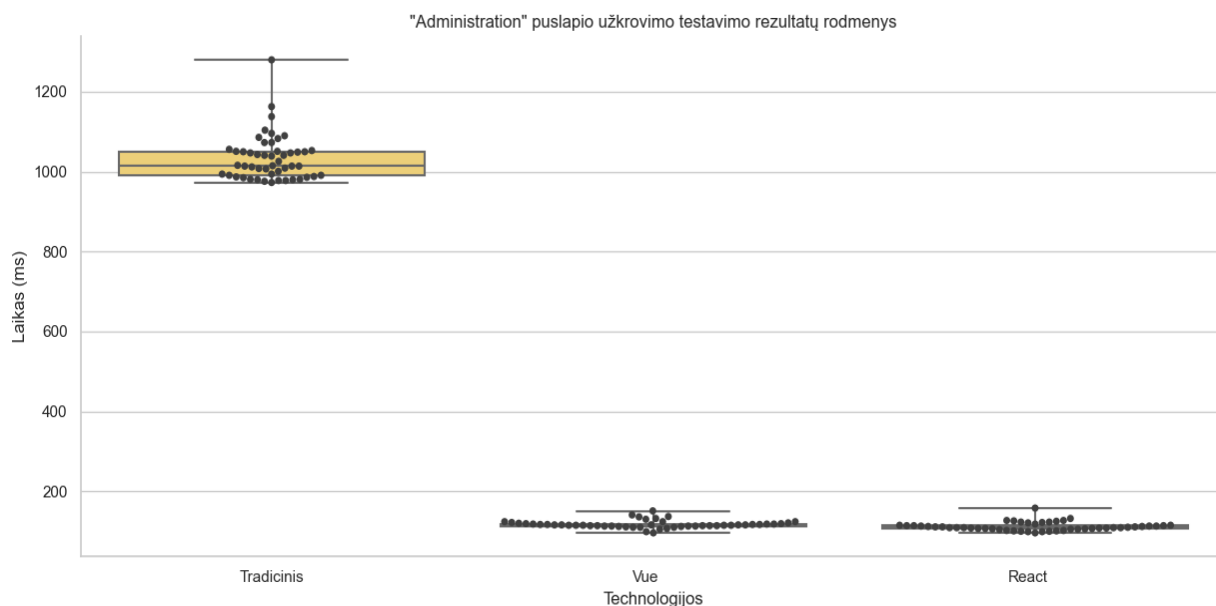
121 pav. pateikiama „Administration“ puslapio užkrovimo scenarijaus laiko vidurkio testavimo diagrama.



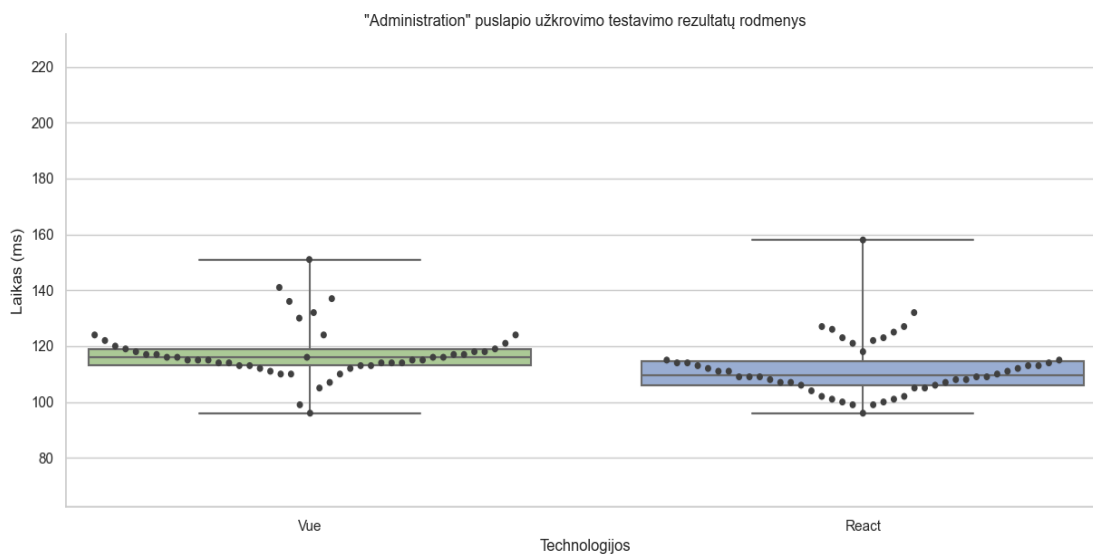
**121 pav.** „Administration“ puslapio užkrovimo testavimo duomenys

Maršrutizatoriaus bendrinių nustatymų „Administration“ puslapį vieno puslapio programoms užkrauti prireikė nedaugiau, nei 120 milisekundžių. Tradicinio modelio informacinėi sistemai prireikė, šiek tiek daugiau nei 1 sekundės, o tai buvo 8 kartais lėčiau, nei vieno puslapio programos

modeliu paremtomis informacinėmis sistemomis. 122 pav. ir 123 pav. pateikiamos „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių paskirstymo diagramos.



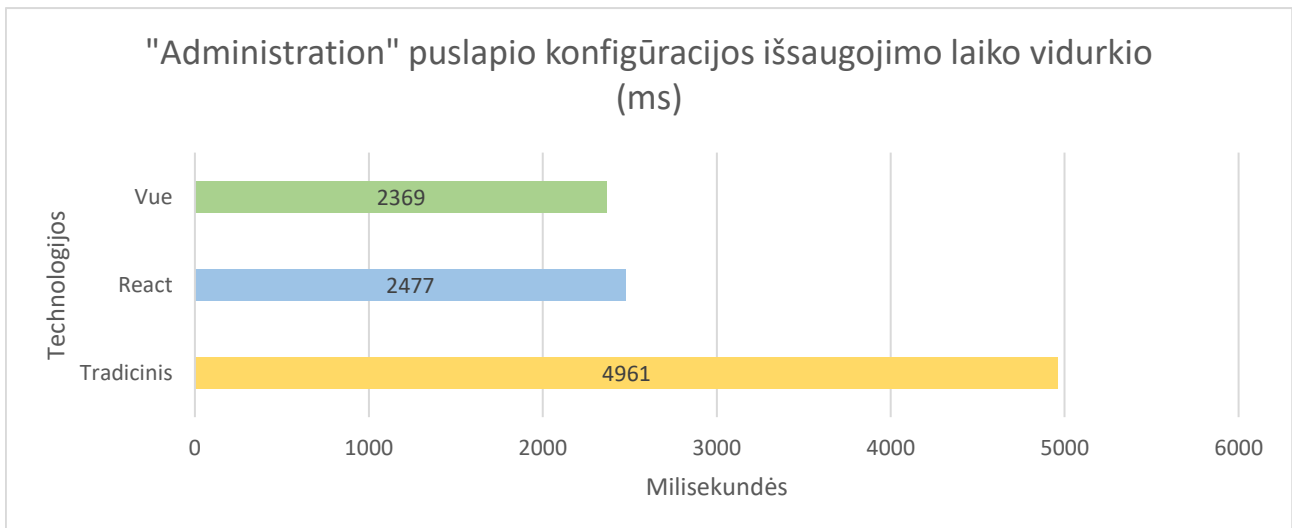
122 pav. „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių paskirstymo diagrama



123 pav. Vieno puslapio programų „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių paskirstymo diagrama

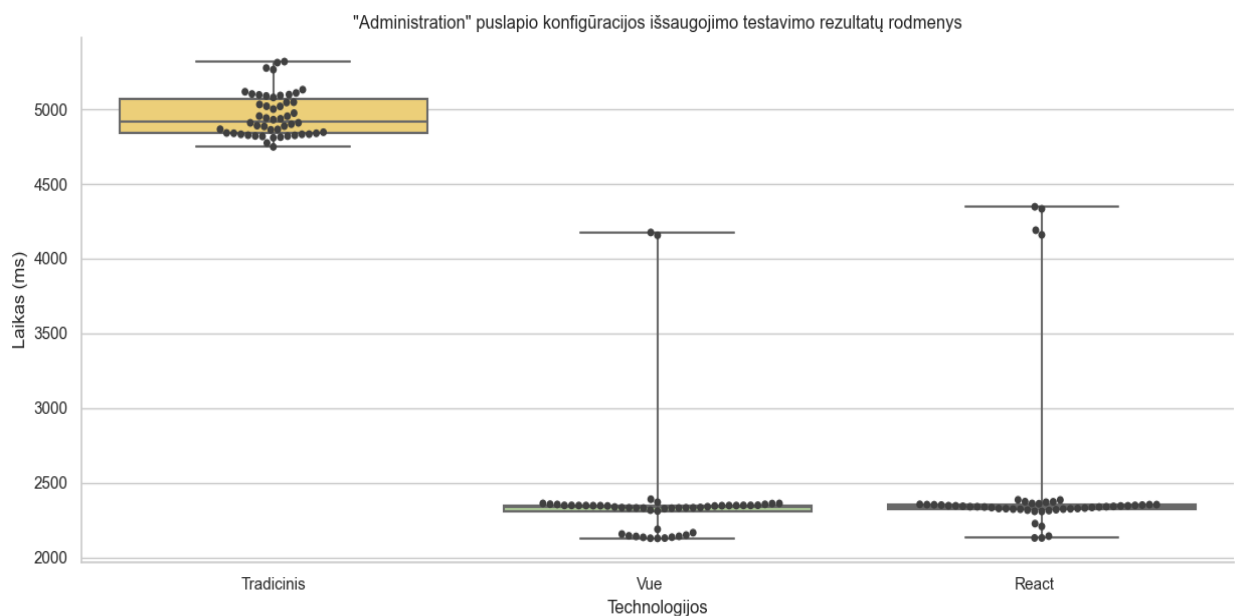
Atsižvelgiant į „Administration“ puslapio užkrovimo testavimo scenarijaus įverčių paskirstymo diagramą matoma, kad Vue mediana yra – 116 milisekundžių, React mediana – 110 milisekundžių, o tradicinio modelio mediana – 1015 milisekundžių.

Maršrutizatoriaus bendrinių nustatymų „Administration“ puslapio konfigūracijos išsaugojimo testavimo rezultatai pateikiami 124 pav.



124 pav. „Administration“ puslapio konfigūracijos išsaugojimo testavimo rezultatai

„Administration“ puslapio konfigūracijos išsaugojimas vieno puslapio programos modelių paremtoms informacinėms sistemoms užtruko ne daugiau, nei 2.5 sekundės. Tradicinio modelių paremtai informacinėms sistemoms prireikė 4,9 sekundžių. Vieno puslapio programos modelio informacinės sistemos konfigūraciją išsaugojo 2 kartus greičiau. 125 pav. pateikiama „Administration“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama.

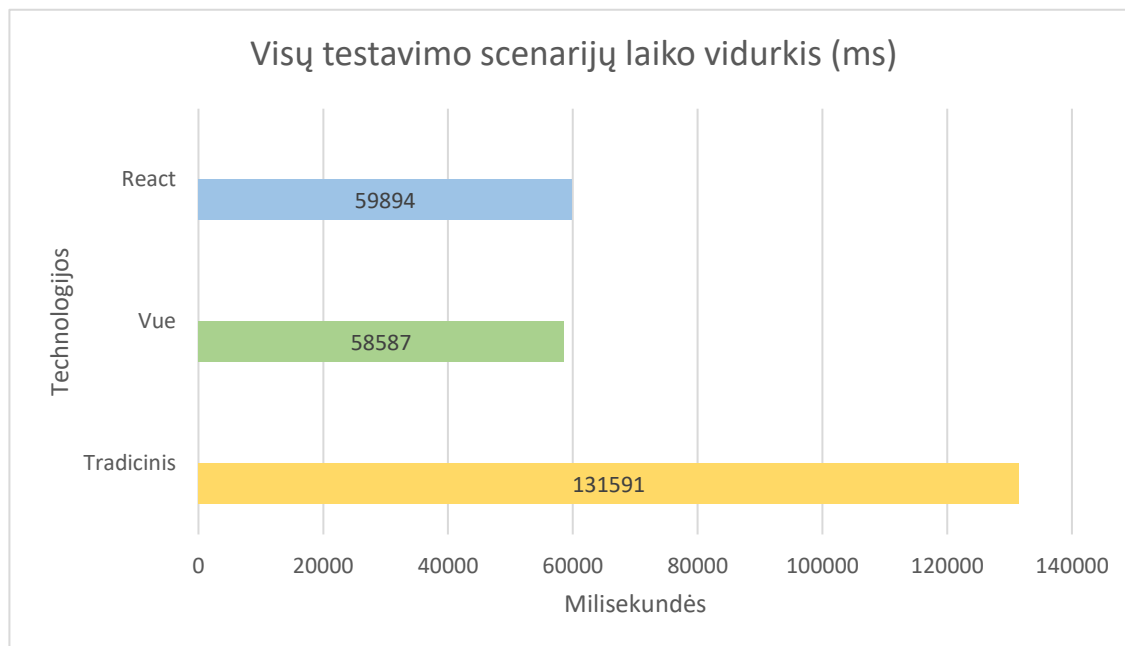


125 pav. „Administration“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagrama

Pagal „Administration“ puslapio konfigūracijos išsaugojimo testavimo scenarijaus įverčių pasiskirstymo diagramą matoma, kad vieno puslapio programos modeliuose yra keli taškai dvigubai viršijantys vidurkį bei medianą. Šie taškai yra užfiksuoti dėl konfigūracijos išsaugojimo klaidų, dėl kurių testavimo metu buvo pastebėti didesni nukrypimai nuo vidutinių reikšmių. Šio testavimo

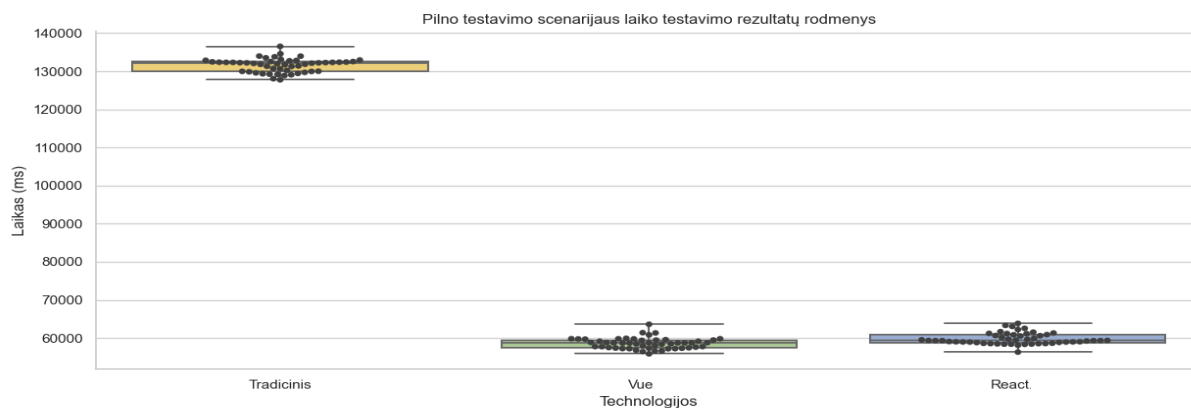
scenarijaus tradicinio modelio mediana yra – 4920 milisekundžių, o vieno puslapio programų modelio technologijoms įverčių pasiskirstymo mediana buvo vienoda – 2340.

Visų testavimo scenarijų bendro laiko vidurkio testavimo rezultatai pateikiami 126 pav.



**126 pav.** Visų testavimo scenarijų bendro laiko vidurkio testavimo rezultatai

Atsižvelgiant į visų testavimo scenarijų bendro laiko vidurkio testavimo rezultatus matoma, kad vienas visų testavimo scenarijų įvykdymas Vue užtruko - 58 sekundės, React – 59 sekundes, o tradicinio modelio paremtai informacinė sistemai – 131 sekundes, kas yra lėčiau nei du kartus lyginant su vieno puslapio programos modelio informacinėmis sistemomis. 127 pav. pateikiama viso testavimo bendro laiko įverčių pasiskirstymo diagrama.

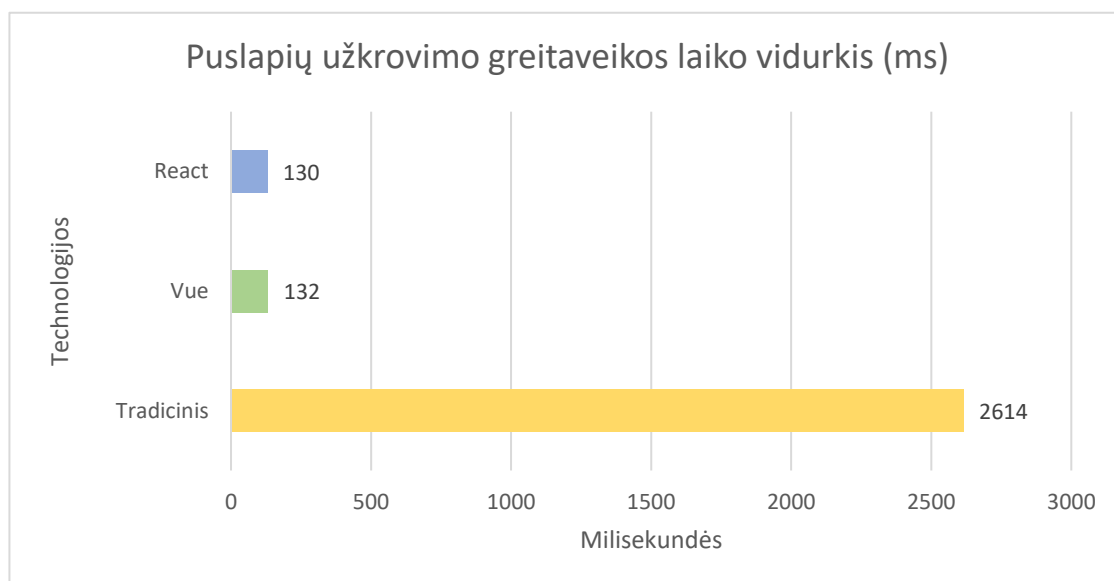


**127 pav.** Viso testavimo bendro laiko įverčių pasiskirstymo diagrama

Atsižvelgiant į visą testavimo scenarijaus bendro laiko įverčių pasiskirstymo diagramą matoma, kad tradicinio modelio mediana yra – 132100 milisekundžių, React mediana – 59313 milisekundžių, o Vue mediana – 58685 milisekundžių.



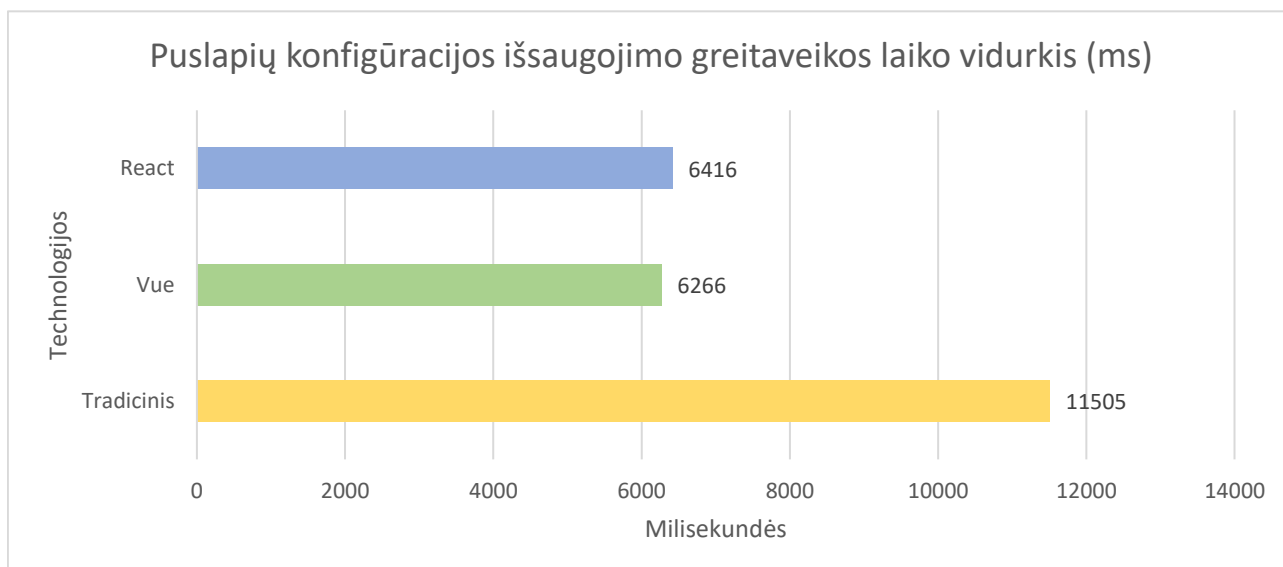
Visus testavimo scenarijus galima išskaidyti į du tipus: puslapio užkrovimą ir konfigūracijos išsaugojimą. Visų scenarijų puslapio užkrovimo vidurkio testavimo rezultatai pateikiami 128 pav.



**128 pav.** Visų puslapių užkrovimo greitaveikos vidurkio rezultatų diagrama

Atsižvelgiant į puslapių užkrovimo greitaveikos rezultatus matoma, kad vidutiniškai React biblioteka turinį užkrovė greičiau per 130 milisekundes, o Vue atsiliko tik per 2 milisekundes. Tradicinio modelio puslapių užkrovimo vidurkis siekė 2.6 sekundės. Vieno puslapio programos modelio puslapių užkrovimas užtruko 19 kartų greičiau, nei tradicinio modelio paremta informacinė sistema.

Visų scenarijų konfigūracijos išsaugojimo greitaveikos vidurkio testavimo rezultatai pateikiami 129 pav.



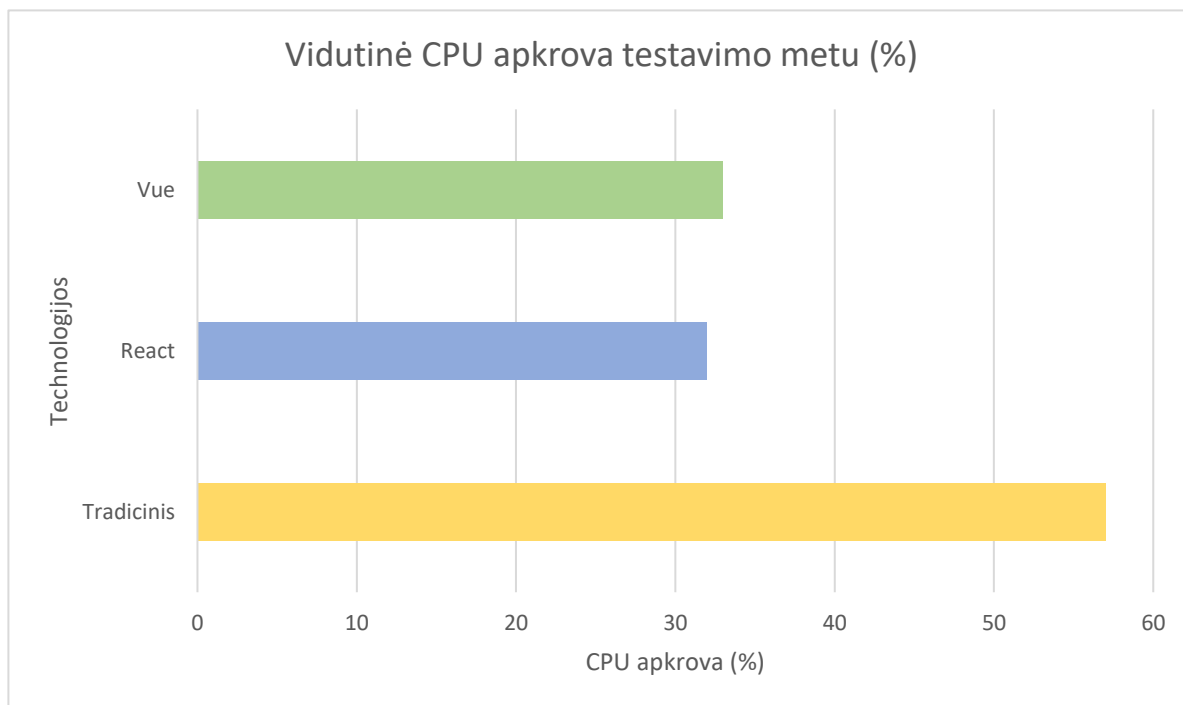
**129 pav.** Konfigūracijos išsaugojimo greitaveikos vidurkio testavimo rezultatai

Puslapių konfigūracijos išsaugojimo greitaveikos vidurkių rezultatai parodė, kad tradicinio modelio paremtai informacinė sistemai konfigūraciją vidutiniškai užtruko išsaugoti per 11,5 milisekundes. React užtruko 6415 milisekundes, o Vue konfigūraciją vidutiniškai išsaugodavo per 6266

milisekundes. Vieno puslapio modelio konfigūracijos išsaugojimas užtruko 1.75 karto greičiau, nei tradicinio modelio informacinėj sistemai.

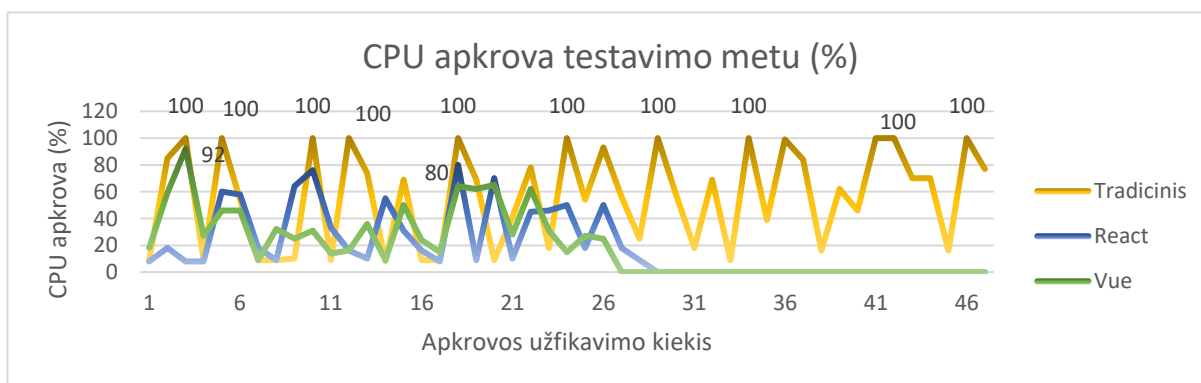
### 5.2.2. Procesoriaus apkrova ir atminties poreikis

Testavimo scenarijų metu, taip pat buvo atliekamas ir procesoriaus apkrovos tikrinimas, kuomet buvo tikrinama procesoriaus apkrova kas 3 sekundes visą testavimo scenarijaus laiką. Vidutinės procesoriaus apkrovos testavimo rezultatai pateikiami 130 pav.



130 pav. Vidutinė procesoriaus apkrovos testavimo metu diagrama

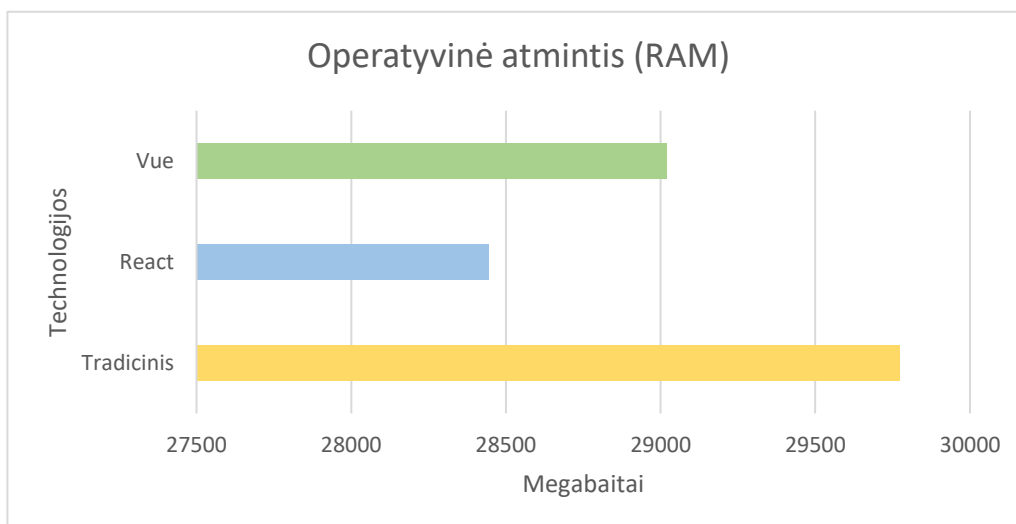
Testavimo metu surinktais duomenimis React parodė 32 % vidutinę procesoriaus apkrovą testavimo metu. Vue parodė 33 %, o tradicinio modelio vidutinė procesoriaus apkrova siekė 57 %. Kadangi procesoriaus apkrova kinta labai trumpais intervalais ir dėl to vidurkio reikšmės neperteikia tikrosios procesoriaus apkrovos vaizdo, todėl realesnė procesoriaus apkrovos situacija yra matoma detalizavus konkretų vieną testavimo scenarijų. 131 pav. pateikiama vieno scenarijaus procesoriaus apkrovos testavimo metu diagrama.



131 pav. Procesoriaus apkrova testavimo metu

Pagal testavimo metu surinktus duomenis matoma, kad tradicinio modelio paremta informacinė sistema 10 kartų pasiekė 100% procesoriaus užimtumą, kai Vue maksimaliai pasiekė 92% vieną kartą ir React pasiekė 80% maksimalią procesoriaus apkrovą. Pagrindinė priežastis, kodėl tradicinio modelio paremtos informacinės sistemos procesoriaus apkrova yra tokia didelė, dėl informacinės sistemos modelio tipo, kuomet kiekvienos užklauskos metu visas internetinis puslapis turi būti sugeneruojamas iš naujo. Silpnų resursų įrenginiui šis procesas reikalauja labai daug resursų.

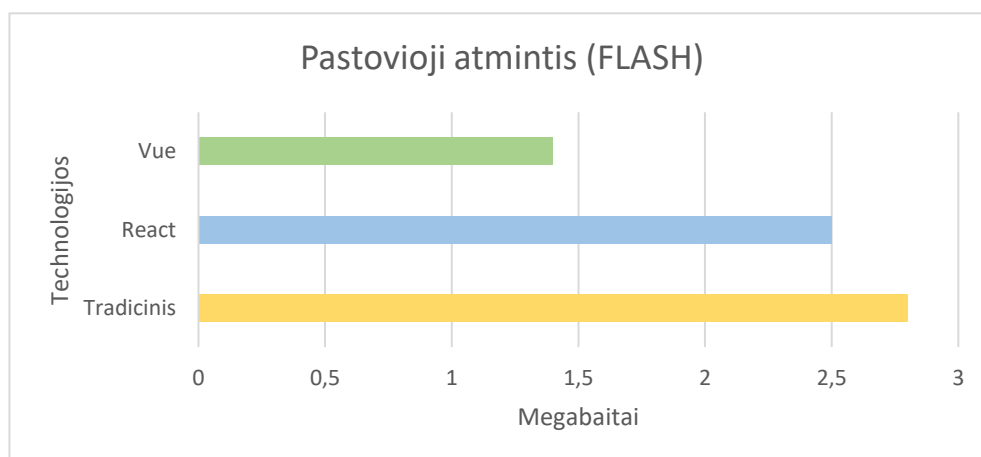
Kartu su procesoriaus apkrovos testavimu buvo tikrinama ir gaminio operatyvioji atmintis visų testavimo scenarijų metu tikrinant kokią įtaką sistemai turi informacinė sistema. Operatyviosios atminties vidurkio rezultatai pateikiami 132 pav.



132 pav. Operatyviosios atminties panaudojimo vidurkio diagrama

Atsižvelgiant į operatyviosios atminties vidurkio rezultatus, matoma, kad informacinės sistemos didelės įtakos operatyviajai atminčiai neturi, kadangi vidurkio rezultatai skiriasi vieno megabaito intervale. React – 28,4 MB, Vue – 29 MB, tradicinio modelio – 29,7 MB.

Kartu su operatyviają atmintimi buvo ir paskaičiuotą pastovioji atmintis apskaičiuojant kiekviena technologija užimta vietos įrenginyje. Pastoviosios atminties rezultatų diagrama pateikiama 133 pav.



133 pav. Technologijų pastoviosios atminties diagrama

Atsižvelgiant į pastoviosios atminties diagrama matoma, kad Vue gaminyje vietos užimta mažiausiai. Vue užėmė 1,4 megabaitus vietos, React užėmė 2.5 megabaitus, o tradicinio modelio informacinės sistema gaminyje užėmė 2.8 megabaitus.

### 5.3. Eksperimento apibendrinimai ir išvados

Apibendrinant eksperimento metu atliktus tyrimus 45 lentelėje pateikiama pagrindinė technologijų gautų rezultatų santrauka.

45 lentelė. Eksperimento metu gautų rezultatų santrauka

	Pavadinimas	Tradicinis modelis	React	Vue
	Pastovioji atmintis (MB)	2,8	2,5	1,4
<b>Vidurkis</b>	Puslapių užkrovimo greitaveika (ms)	2614	130	132
	Konfigūracijos išsaugojimo greitaveika (ms)	11505	6416	6266
	Procesoriaus apkrova (%)	57	32	33
	Operatyvioji atmintis (MB)	29,7	28,4	29
<b>Mediana</b>	Puslapių užkrovimo greitaveika (ms)	1349	126	125
	Konfigūracijos išsaugojimo greitaveika (ms)	8752	2535	2504
	Procesoriaus apkrova (%)	62	20	24
	Operatyvioji atmintis (MB)	29,8	28,3	29

Eksperimento apibendrinimai ir išvados:

1. Atsižvelgus į analizės dalyje analizuotus internetinių puslapių kūrimo modelius ir iškeltus funkcinis reikalavimus buvo realizuotos dvi vieno puslapio programos modeliu paremtos grafinės sąsajos panaudojant Vue karkasą ir React biblioteką.
2. Eksperimentui įvykdyti buvo realizuota automatizuota testavimo sistema grafinės sąsajos puslapių greitaveikai patikrinti.
3. Pagal automatizuotos testavimo programos vidurkio rezultatus matomas net iki 19 kartų naujai sukurtų grafinių sąsajų pranašumas užkraunant ir atvaizduojant internetinius puslapius, lyginant su tradicinių modeliu paremta grafine sąsaja.
4. Atsižvelgiant į eksperimento rezultatus matoma, kad vieno puslapio programos modeliu paremtos informacinės sistemos leidžia sumažinti procesoriaus apkrovą, dėl modelio veikimo nereikalaujant serverio dalyje sugeneruoti naujo puslapio, kuris didina įrenginio procesoriaus apkrovą.
5. Atsižvelgiant į eksperimento rezultatus matoma, kad didžiausią greitaveikos įtaką tradicinio internetinių puslapių modelio grafinei sąsajai sudaro naujo puslapio generavimas kuomet visa puslapio generavimo dalis atliekama serverio dalyje.
6. Kuriant grafinę sąsają rekomenduotina pasirinkti vieno puslapio programos modeliu paremtą technologiją, nes tai leidžia sumažinti puslapių užkrovimo greitaveiką lyginant su tradicine daugelio puslapių technologija.
7. Eksperimento rezultatai parodė, kad ženklus greitaveikos skirtumas tarp testuotų vieno puslapio technologijų nėra, todėl renkantis tarp jų vertėtų atsižvelgti į papildomus faktorius: naujumą, bendruomenės dydį ir dokumentacijos išpildymą.

## Išvados

1. Analizės metu buvo nagrinėtos tradicinės ir vieno puslapio modeliu grįstos internetinių puslapių kūrimo technologijos. Išrinktos dvi vieno puslapio modeliu grįstos technologijos pasirodžiusios geriausiai pagal iškeltus kriterijus.
2. Atsižvelgus į analizės dalyje analizuotus internetinių puslapių kūrimo modelius ir iškeltus funkcinis reikalavimus buvo realizuotos dvi vieno puslapio programos modeliu paremtos grafinės sąsajos panaudojant Vue karkasą ir React biblioteką.
3. Realizuotos informacinės sistemos įvykdo visus iškeltus funkcinis ir nefunkcinis reikalavimus.
4. Eksperimentui įvykdyti buvo realizuota automatizuota testavimo sistema grafinės sąsajos puslapių greitaveikai ir resursų panaudojimui patikrinti.
5. Atliktas eksperimentas, kurio metu buvo atliekamas internetinių puslapių greitaveikos, maršrutizatoriaus procesoriaus apkrovos, operatyviosios atminties ir pastoviosios atminties testavimas.
6. Atsižvelgiant į eksperimento rezultatus galima teigti, kad vieno puslapio programos modeliu paremtos informacinės sistemos sumažina procesoriaus apkrovą, dėl vieno puslapio programos modelio veikimo kuomet internetinių puslapių atvaizdavimo logika atliekama naršyklės dalyje.
7. Atsižvelgus į eksperimento rezultatus galima teigti, kad didžiausią greitaveikos įtaką tradicinio internetinių puslapių modelio grafinėj sąsajai sudaro naujo puslapio generavimas, kuomet visa puslapio generavimo dalis atliekama serverio dalyje.
8. Atsižvelgus į eksperimento rezultatus, galima daryti išvadą, kad vieno puslapio programos modelis pagreitina IoT maršrutizatoriaus internetinių puslapių užkrovimą vidutiniškai iki 19 kartų.
9. Vieno puslapio programos modelis tinkamas pakeisti šiuo metu maršrutizatoriuje esančios programinės įrangos tradicinių internetinių puslapių modelį.
10. Eksperimento rezultatai parodė, kad ženklus greitaveikos skirtumas tarp testuotų vieno puslapio technologijų nėra, todėl renkant tarp jų vertėtų atsižvelgti į papildomus faktorius: naujumą, bendruomenės dydį ir dokumentacijos išpildymą.

## Literatūros sąrašas

1. A. Mesbah ir A. van Deursen. Migrating Multi-page Web Applications to Single-page AJAX Interfaces. 2007, [žiūrėta 2020-11-21]. Prieiga per:  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4145036>
2. S. Aghaei, M. Ali Nematbakhsh ir H. Khosravi Farsami. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. [interaktyvus], 2012. [žiūrėta 2020-11-21]. Prieiga per:  
<http://airccse.org/journal/ijwest/papers/3112ijwest01.pdf>
3. G. Wu, S. Talwar, K. Johnsson, N. Himayt, ir K. D. Johnson. M2M: From mobile to embedded internet. 2011. [žiūrėta: 2020-11-21]. Prieiga per:  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5741144>
4. J. James Garret. AJAX: A New Approach to Web Applications. 2005. [žiūrėta: 2020-11-21]. Prieiga per: [https://www.scriptol.fr/AJAX/AJAX\\_adaptive\\_path.pdf](https://www.scriptol.fr/AJAX/AJAX_adaptive_path.pdf)
5. M. A. Jadhav, B. R. Sawant ir A. Deshmukh. Single Page Application using AngularJS. 2015. [žiūrėta: 2020-11-23]. Prieiga per:  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.4771&rep=rep1&type=pdf>
6. S. Holm Jensen, A. Moller, P. Thiemann. Type Analysis for JavaScript. 2009. [žiūrėta: 2020-11-23]. Prieiga per: <https://cs.au.dk/~amoeller/papers/tajs/paper.pdf>
7. C. Quist Adamsen, S. Alimadadi, A. Moller, F. Tip. Practical AJAX Race Detection for JavaScript Web Applications. 2018. [žiūrėta: 2020-11-24]. Prieiga per:  
<https://dl.acm.org/doi/pdf/10.1145/3236024.3236038>
8. C. Severance. Discovering JavaScript Object Notation. 2013. [žiūrėta: 2020-11-24]. Prieiga per:  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6178118>
9. AM Vipul, P. Sonpatki. ReactJS by Example – Building Modern Web Applications with React. 2015. [žiūrėta: 2020-11-24]. Prieiga per:  
[https://demo.smarttrainerlms.com/uploads/0003/trainings/course/43/modules/ReactJS%20by%20Example\\_150994330836904296.pdf](https://demo.smarttrainerlms.com/uploads/0003/trainings/course/43/modules/ReactJS%20by%20Example_150994330836904296.pdf)
10. A. Gore. Full-Stack Vue.js 2 and Laravel 5: Bring the frontend and backend together with Vue, Vuex, and Laravel. Birmingham: 2017. ISBN 978-1-78829-958-9.
11. J. Zhu, D. S. Chan, M. Suryanarayana Prabhu, P.i Natarajan, H.o Hu ir F. Bonom. Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture. 2013. [žiūrėta: 2020-11-25]. Prieiga per: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6525539>
12. T.Jin. OpenWrt Development Guide. 2012. [žiūrėta: 2020-12-18]. Prieiga per:  
[http://www.ccs.neu.edu/home/noubir/Courses/CS6710/S12/material/OpenWrt\\_Dev\\_Tutorial.pdf](http://www.ccs.neu.edu/home/noubir/Courses/CS6710/S12/material/OpenWrt_Dev_Tutorial.pdf)
13. F. Fui-Hoon Nah. A study on tolerable witing time: how long Web users willing to wait? 2007. [žiūrėta: 2020-12-19]. Prieiga per:  
<https://www.tandfonline.com/doi/pdf/10.1080/01449290410001669914>

14. B. Green, S. Sesbadri. AngularJS. 2013. ISBN: 978-1-449-34485-6.
15. M. Levlin. DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue and Svelte. 2020. [žiūrēta: 2022-04-26]. Prieiga per:  
[https://www.doria.fi/bitstream/handle/10024/177433/levlin\\_mattias.pdf](https://www.doria.fi/bitstream/handle/10024/177433/levlin_mattias.pdf)
16. Front-end Framerowks. 2021. [žiūrēta: 2022-04-26]. Prieiga per: <https://2021.stateofjs.com/en-US/libraries/front-end-frameworks>