

KLAIPĖDOS UNIVERSITETAS
JŪROS TECHNOLOGIJŲ IR GAMTOS MOKSLŲ FAKULTETAS
INFORMATIKOS IR STATISTIKOS KATEDRA

LINAS SILKAUSKAS

**SVEIKATOS STEBĖJIMO SISTEMOS PROTOTIPAS
ANKSTYVAM ŠIRDIES IR KRAUJAGYSLIŲ LIGŲ
DIAGNOZAVIMUI**

**A PROTOTYPE OF THE HEALTH MONITORING SYSTEM FOR EARLY
CARDIOVASCULAR DISEASES DIAGNOSIS**

Techninių informacinių sistemų inžinerijos magistro baigiamasis darbas
Techninių informacinių sistemų inžinerijos studijų programa 621E15004

Klaipėda, 2017

MAGISTRO BAIGIAMOJO DARBO LYDRAŠTIS

Pildo magistro baigiamojo darbo autorius

.....**Linas Silkauskas**.....
(magistro baigiamojo darbo autoriaus vardas, pavardė)

Baigiamojo darbo tema: Sveikatos stebėjimo sistemos prototipas ankstyvam širdies ir kraujagyslių ligų diagnozavimui

(magistro baigiamojo darbo pavadinimas lietuvių kalba)

Patvirtinu, kad bakalauro/magistro baigiamasis darbas parašytas savarankiškai, nepažeidžiant kitiems asmenims priklausančių autorių teisių, visas baigiamasis magistro darbas ar jo dalis nebuvo panaudotas Klaipėdos universitete ir kitose aukštosiose mokyklose.

.....**Linas Silkauskas**.....
(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Sutinku, kad bakalauro/magistro baigiamasis darbas būtų naudojamas neatlygintinai 5 m. Klaipėdos universiteto studijų procese.

.....**Linas Silkauskas**.....
(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Pildo magistro baigiamojo darbo vadovas

Magistro baigiamąjį darbą ginti
(įrašyti – **leidžiu** arba **neleidžiu**)

.....2017..... doc. dr. Violeta Bulbenkienė.....
(data) (magistro baigiamojo darbo vadovės vardas, pavardė ir parašas)

Pildo katedros, kuriojančios studijų programą, administratorius (sekretorius)

Baigiamasis darbas įregistruotas katedroje2017.....
(data)

Laima Brazdeikienė.....
(katedros sekretorės vardas, pavardė ir parašas)

Pildo katedros, kuriojančios studijų programą, vedėjas

Magistro baigiamąjį darbą ginti
(įrašyti – **leidžiu** arba **neleidžiu**)

.....2017..... prof. dr. Arūnas Andziulis.....
(data) (katedros vedėjo vardas, pavardė ir parašas)

Recenzentu (-ais) skiriuprof. dr. Arūnas Andziulis.....
.....lekt. dr. Mindaugas Kurmis.....
.....

(įrašyti recenzento(u) vardą, pavardę)

.....2017..... prof. dr. Arūnas Andziulis.....
(data) (katedros vedėjo vardas, pavardė ir parašas)

PADĖKA

Nuoširdžiai dėkoju darbo vadovei doc. dr. Violetai Bulbenkienei už pagalbą ir kantrybę rengiant baigiamąjį darbą ir mokslinius straipsnius, už konsultacijų metu pateiktas rekomendacijas ir patarimus. Dėkoju katedros vedėjui prof. dr. Arūnui Andziuliui už paskatinimus studijuoti magistrantūroje bei už pasiūlytas idėjas. Taip pat dėkoju KU informatikos ir statistikos katedros dėstytojams už suteiktas žinias studijų metu, ypač lekt. dr. Jurij Tekutov padėjusiam parengti mokslinį straipsnį ir už naudingus patarimus. Dėkoju tėvams ir sesei už palaikymą.

ANOTACIJA

Šiame darbe pateiktas sveikatos stebėjimo informacinės sistemos su grįžtamoju ryšiu prototipas. Jis leidžia medikams nuotoliniu būdu stebėti pacientų širdies ir kraujagyslių veiklą. Sistemą sudaro keturi pagrindiniai komponentai, kuriems parašyti programiniai algoritmai: jutikliams – širdies ritmo fiksavimo, mobiliajai programai – užfiksuotų parametrų siuntimo, debesų serveriui – fiksavimo suvestinės PDF dokumento generavimo, internetiniam puslapiui – užfiksuotų parametrų atvaizdavimo. Jutikliu užfiksuoti asmens sveikatos ir judėjimo parametrai perduodami į mobiliąją programą, o iš jos į debesų serverį. Medicinos personalui internetiniame puslapyje pateikiama paciento parametrų fiksavimo suvestinė. Medikų išvados siunčiamos atgal pacientui. Atliktas sistemos pagrindinių komponentų verifikavimas. EKG signalo filtravimui pasirinktas EMA algoritmas, širdies ritmo nuokrypis nuo sertifikuoto medicininio prietaiso rodmenų lygus nuo 0 iki 2 dūžių. Naudojant mobiliųjį telefoną turintį bateriją, kurios talpa 3300 mAh, įkrauta 100%, mobiliąją programą be pertraukos galima naudotis apie 15h. Vienu metu prie sistemos gali prisijungti apie 120 vartotojų, esant klaidų dažnumui $\leq 3\%$, debesų serverio vidutinis atsako laikas lygus apie 6 s. Internetinio puslapio vidutinis užkrovimo laikas po optimizacijos lygus apie 900 ms, prieš – apie 1100 ms.

PAGRINDINIAI ŽODŽIAI: debesų kompiuterija, bevielė komunikacija, medicininiai jutikliai, nuotolinis sveikatos stebėjimas, parametrų fiksavimas.

ABSTRACT

There is a prototype of the health monitoring information system with feedback presented in this work. Medical staff can remotely monitor cardiovascular activity of patients. The system consists of four main components. In this work develop algorithms: acquire heart rate using sensors, send acquired parameters using mobile application, generate PDF document of acquisition summary using cloud server, visualize acquired parameters using website. Health and movement parameters are acquired using sensor. Parameters are sent to mobile application, from it to cloud server. In the website acquisition summary of patient parameters are supplied to the medical staff. The medical findings are sent back to the patient. The are performed verification of system main components. EMA algorithm are chosen to filter ECG signal, deviation of the heart rate equals from 0 to 2 beats when results are compared with readings of certified medical device. Mobile application without a break can be used about 15h, when using mobile phone having a battery which capacity 3300 mAh, charged 100%. At the same time 120 users can connect to the system, at error rate $\leq 3\%$. Then average response time of the cloud server equals about 6 s. Average load time of web page after optimization equals about 900 ms, before – about 1100 ms.

KEYWORDS: cloud computing, wireless communication, medical sensors, remote health monitoring, parameters acquisition.

TURINYS

Lentelių sąrašas	5
Paveikslėlių sąrašas	6
Terminų ir santrumpų žodynas.....	8
Įvadas	9
I. Analizė.....	10
1.1. Sveikatos parametrų stebėjimas nuotoliniu būdu	10
1.2. Sveikatos parametrų stebėjimas naudojant išmaniuosius telefonus.....	12
1.3. Debesų kompiuterijos infrastruktūros panaudojimas sveikatos priežiūros sistemose	14
1.4. Duomenų saugumas ir daiktų internetas sveikatos priežiūros sistemose.....	17
1.5. Visuotinės sveikatos priežiūros sistemos	19
1.6. Bevielių jutiklių tinklai ant tiriamojo kūno paviršiaus.....	20
1.7. Didelių duomenų apdorojimas sveikatos priežiūros sistemose	21
1.8. Medicininių jutiklių integravimas į vieną bendrą sistemą.....	22
1.9. Automatinės reguliavimo sistemos sveikatos priežiūroje	23
1.10. Analizės apibendrinimas.....	24
II. Projektavimas	28
2.1. Reikalavimų specifikacija.....	28
2.2. Įrangos parinkimas	32
2.3. Sveikatos stebėjimo sistemos prototipo kūrimas	35
2.3.1. Parametrų fiksavimas jutikliais	35
2.3.2. Mobilioji programa fiksuojamų parametrų siuntimui į serverį	39
2.3.3. Duomenų apdorojimas debesų serveryje	47
2.3.4. Internetinis puslapis fiksuojamų parametrų atvaizdavimui.....	54
2.4. Sistemos testavimas ir diegimas	58
III. Eksperimentinė dalis.....	62
3.1. Signalų filtravimo algoritmų lyginamasis tyrimas EKG fiksavimo metu.....	62
3.2. Telefono resursų, tinklo ir energijos sąnaudų tyrimas mobiliosios programos veikimo metu....	64
3.3. Debesų serverio atsako laiko tyrimas esant skirtingam vartotojų skaičiui.....	67
3.4. Internetinio puslapio užkrovimo laiko tyrimas prieš ir po optimizacijos.....	69
3.5. Tolimesnio sistemos tobulinimo, plėtojimo galimybės.....	70
Išvados	71
Literatūra.....	72
PRIEDAI:	

1. Kompaktinis diskas
2. Mokslinė veikla
3. Užfiksuotų parametrų atvaizdavimas sukurtoje programoje naudojant .NET
4. SMA, EMA ir CMA filtravimo algoritmų veikimo principas

LENTELIŲ SĄRAŠAS

1 lentelė. Sveikatos priežiūros sistemos	24
2 lentelė. Integruotų kūrimo aplinkų funkcijos	33
3 lentelė. Valdiklių plokščių techniniai parametrai.....	34
4 lentelė. Fiksuojami parametrai jutikliais.....	39
5 lentelė. Mobiliosios programos paketų paskirtis.....	41
6 lentelė. Mobiliosios programos klasių ir interfeisų paskirtis	44
7 lentelė. REST API resursų paskirtis	48
8 lentelė. Pasirinkti šifravimo algoritmų parametrai	51
9 lentelė. Duomenų bazės lentelių paskirtis.....	52
10 lentelė. Programinių bibliotekų paskirtis sukurtoje sistemoje	59
11 lentelė. Telefono resursų sąnaudos mobiliosios programos veikimo metu	65
12 lentelė. Telefono tinklo sąnaudos mobiliosios programos veikimo metu	66
13 lentelė. Telefono energijos sąnaudos mobiliosios programos veikimo metu	66
14 lentelė. Serverio vidutinis atsako laikas ir pralaidumas esant maksimaliam vartotojų skaičiui	68

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Sistemų grupavimas	10
2 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (1)	30
3 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (2)	31
4 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (3)	32
5 pav. Sistemos įgyvendinimui pasirinkta programinė įranga	33
6 pav. Suprojektuota techninė įranga naudojant SEN-11574 pulso jutiklį.....	36
7 pav. Algoritmas širdies ritmo fiksavimui naudojant SEN-11574 pulso jutiklį.....	36
8 pav. Suprojektuota techninė įranga naudojant e-Health platformą su EKG jutikliu.....	37
9 pav. Techninės įrangos sujungimo schema naudojant e-Health platformą su EKG jutikliu	38
10 pav. Algoritmas širdies ritmo fiksavimui naudojant e-Health platformą su EKG jutikliu	38
11 pav. Širdies ritmo fiksavimas Zephyr HxM BT juosmens diržu, Xiaomi Mi Band 1s apyranke	39
12 pav. Užfiksuotų parametrų, medicininių išvadų atvaizdavimas mobiliojoje programoje	40
13 pav. Mobiliosios programos klasių ir interfeisų saugojimo struktūra	41
14 pav. Mobiliosios programos statinis vaizdas (1)	41
15 pav. Mobiliosios programos statinis vaizdas (2)	42
16 pav. Mobiliosios programos statinis vaizdas (3)	43
17 pav. Mobiliosios programos statinis vaizdas (4)	44
18 pav. Algoritmas fiksuojamų parametrų siuntimui iš mobiliosios programos į debesų serverį.....	46
19 pav. Sukurta REST API, kuri įdiegta debesų kompiuterijos infrastruktūroje (1)	47
20 pav. Sukurta REST API, kuri įdiegta debesų kompiuterijos infrastruktūroje (2)	48
21 pav. Sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas (1).....	50
22 pav. Sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas (2).....	51
23 pav. Suprojektuota duomenų bazės struktūra.....	52
24 pav. Algoritmas fiksavimo suvestinės PDF dokumento generavimui debesų serveryje	53
25 pav. Sugeneruotas PDF dokumentas su paciento fiksuotų parametrų savaitine suvestine	54
26 pav. Širdies ritmo, judėjimo ir vietovės parametrai užfiksuoti tam tikru laiko momentu	55
27 pav. Širdies ritmo ir judėjimo parametrų fiksavimo suvestinė.....	55
28 pav. Gydytojų medicininės išvados apie paciento sveikatos būklę.....	56
29 pav. Algoritmas fiksuojamų parametrų atvaizdavimui internetiniame puslapyje.....	57
30 pav. Internetinio puslapio failų optimizavimo procesas	58
31 pav. Sistemos įgyvendinimui panaudotos programinės bibliotekos (1)	58
32 pav. Sistemos įgyvendinimui panaudotos programinės bibliotekos (2)	59
33 pav. Nuotolinė širdies ir kraujagyslių ligų stebėjimo sistema su grįžtamoju ryšiu	60

34 pav. Sistemos diegimas į galutiniams vartotojams skirtą aplinką.....	61
35 pav. Algoritmas automatizuotam programos diegimui į debesų serverį	61
36 pav. EKG signalo filtravimo rezultatai naudojant SMA, EMA, CMA, RDP ir KZ algoritmus	63
37 pav. Vidutinis širdies ritmo fiksavimo tikslumas naudojant skirtingus filtravimo algoritmus.....	64
38 pav. Telefono resursų sąnaudų kitimas mobiliosios programos veikimo metu	65
39 pav. Telefono tinklo sąnaudų kitimas mobiliosios programos veikimo metu	65
40 pav. Telefono energijos sąnaudų kitimas mobiliosios programos veikimo metu	66
41 pav. Debesų serverio atsako laiko tyrimo rezultatai.....	67
42 pav. Vidutinio atsako laiko, pralaidumo, klaidų dažnumo priklausomybė nuo vartotojų skaičiaus .	68
43 pav. Internetinio puslapio failų skaičius ir jų užimama vieta prieš ir po optimizacijos	69
44 pav. Internetinio puslapio užkrovimo laiko rezultatai prieš ir po optimizacijos.....	69
45 pav. Internetinio puslapio užkrovimo laiko histograma prieš ir po optimizacijos	70

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Aplikacijų programavimo sąsaja (angl. *Application Programming Interface, API*) – tai sąsaja, kuri leidžia skirtingiems sistemos komponentams komunikuoti tarpusavyje.

Autentifikacija (angl. *Authentication*) – asmens atpažinimas prieš suteikiant informaciją.

Autorizacija (angl. *Authorization*) – nustatoma ar identifiikuotas vartotojas turi teisę naudotis paslaugomis.

Daiktų internetas (angl. *Internet of Things, IoT*) – tarpusavyje interneto tinklu susijungę įrenginiai. Perduodant informaciją per tinklą naudojama ne tik „žmogus-žmogus“ ar „žmogus-kompiuteris“ sąveikos, bet ir sąveika „įrenginys-įrenginys“.

Debesų kompiuterija (angl. *Cloud Computing*) – kompiuterinių išteklių nuoma, kai nereikia rūpintis techninės įrangos priežiūra ir palaikymu, infrastruktūros įgyvendinimu. Kintamo dydžio skaičiavimo resursai prieinami per tinklo prieigą, norint keisti jų kiekį nereikia investicijų naujiems serveriams.

Dėvimoji technologija (angl. *Wearable Technology*) – įrenginiai skirti fiksuoti asmens fizinio aktyvumo parametrus nesukeliant nepatogumų, integruoti į aprangą arba priglauti prie kūno (pvz., apyrankė, juosmens diržas, laikrodis).

Dideli duomenys (angl. *Big Data*) – tai terminas skirtas apibūdinti didžiulius duomenų masyvus, kai naudojantis tradiciniais metodais ir būdais tokius duomenų kiekius sunku analizuoti, apdoroti, saugoti, atlikti paiešką, perduoti ir atvaizduoti.

Duomenų išgavimas (angl. *Data Mining*) – tai automatinė informacijos paieška duomenų bazėje pagal šablonus, siekiant didelės apimties duomenų masyvuose rasti naują ir naudingą informaciją.

ISO/IEEE 11073 standartas – komunikacijos standartas, kuris leidžia komunikuoti medicininiais prietaisams, sveikatos priežiūrai skirtiems įrenginiams su kompiuterinėmis sistemomis.

Išmanieji asmeniniai asistentai-padėjėjai (angl. *Intelligent Personal Assistants, IPA*) – tai programinės įrangos agentai, kurie naudojami padėti vartotojams įvykdyti kasdienes užduotis, išklaudyti pageidavimus ir pateikti patarimus.

OMA DM protokolas – tai mobiliesiems įrenginiams skirtas protokolas, kuris leidžia po įrenginio pagaminimo atnaujinti sistemos programinę įrangą, įdiegti pataisymus, ištaisyti programines klaidas ir siųsti klaidos pranešimus iš įrenginio į serverį.

REST (*Representational State Transfer*) – tai architektūrinis stilius internetinių paslaugų, API kūrimui.

Šifro teksto strategija ir atributais paremtas šifravimas (angl. *Ciphertext Policy Attribute based Encryption, CP-ABE*) – šifravimo mechanizmas, kai užšifruotus duomenis gali iššifruoti vartotojas, jeigu jam suteiktas slaptasis raktas turi atributus, kurie atitinka prieigos strategiją.

ĮVADAS

Širdies ir kraujagyslių ligos yra viena iš pagrindinių mirties priežasčių Europos šalyse. Daugumos šių ligų galima išvengti laiku imantis reikalingų priemonių. Nuolatinis sveikatos stebėjimas ypač svarbus diagnozuojant ir gydant šias ligas. Užfiksuoti su sveikata susiję parametrai svarbūs gydytojams ir kitiems medicinos specialistams skiriant asmeniui reikiamą gydymą. Didėjant bevielių ir nešiojamų medicininių prietaisų ir jutiklių skaičiui atsiranda galimybė sukurti mobiliąsias širdies ir kraujagyslių ligų monitoringo sistemas [1].

Siekiant gerinti žmonių gyvenimo kokybę, prailginti sveiko gyvenimo trukmę, sumažinti dėl ligos ar neįgalumo prarastus gyvenimo metus, kuriamos įvairios sveikatos stebėjimo sistemos. Nurodoma, kad sveikatos stebėjimo sistemų, didelių duomenų analizės ir debesų kompiuterijos infrastruktūros apjungimas sukelia problemų [2]. Susiduriama su duomenų apsaugojimo problema [3]. Su paciento sveikata susijusi informacija yra nuolat fiksuojama. Didžiuliai duomenų rinkiniai turi būti saugomi ilgą laikotarpį [4]. Dauguma sveikatos priežiūros sistemų yra centralizuotos. Reikalinga paskirstyta programų sistemų architektūra [5].

Viena iš pagrindinių problemų sveikatos priežiūros sistemose yra skirtingų tiekėjų programinių bibliotekų, technologijų, medicininių jutiklių integravimas į vieną bendrą sistemą. Šiame darbe siekiama apjungti e-Health jutiklių platformą, Zephyr HxM BT juosmens diržą, Xiaomi Mi Band 1s apyrankę, SEN-11574 pulso jutiklį, Android operacinę sistemą, REST architektūrinį stilių, internetines technologijas (PHP, Apache HTTP, MySQL) į vieną bendrą sistemą. Darbe pateikta nuotolinė sveikatos stebėjimo sistema su grįžtamoju ryšiu apimanti tokias funkcijas: asmens sveikatos parametrų fiksavimas jutikliais, bevieliu ryšiu duomenų siuntimas į mobilųjį įrenginį ir debesų serverį, buvimo vietos koordinatų nustatymas esant ekstriniam atvejui, suinteresuotų asmenų informavimas SMS žinute ir elektroniniu laišku apie sveikatos parametrų nukrypimus, apdorotos informacijos atvaizdavimas internetiniame puslapyje.

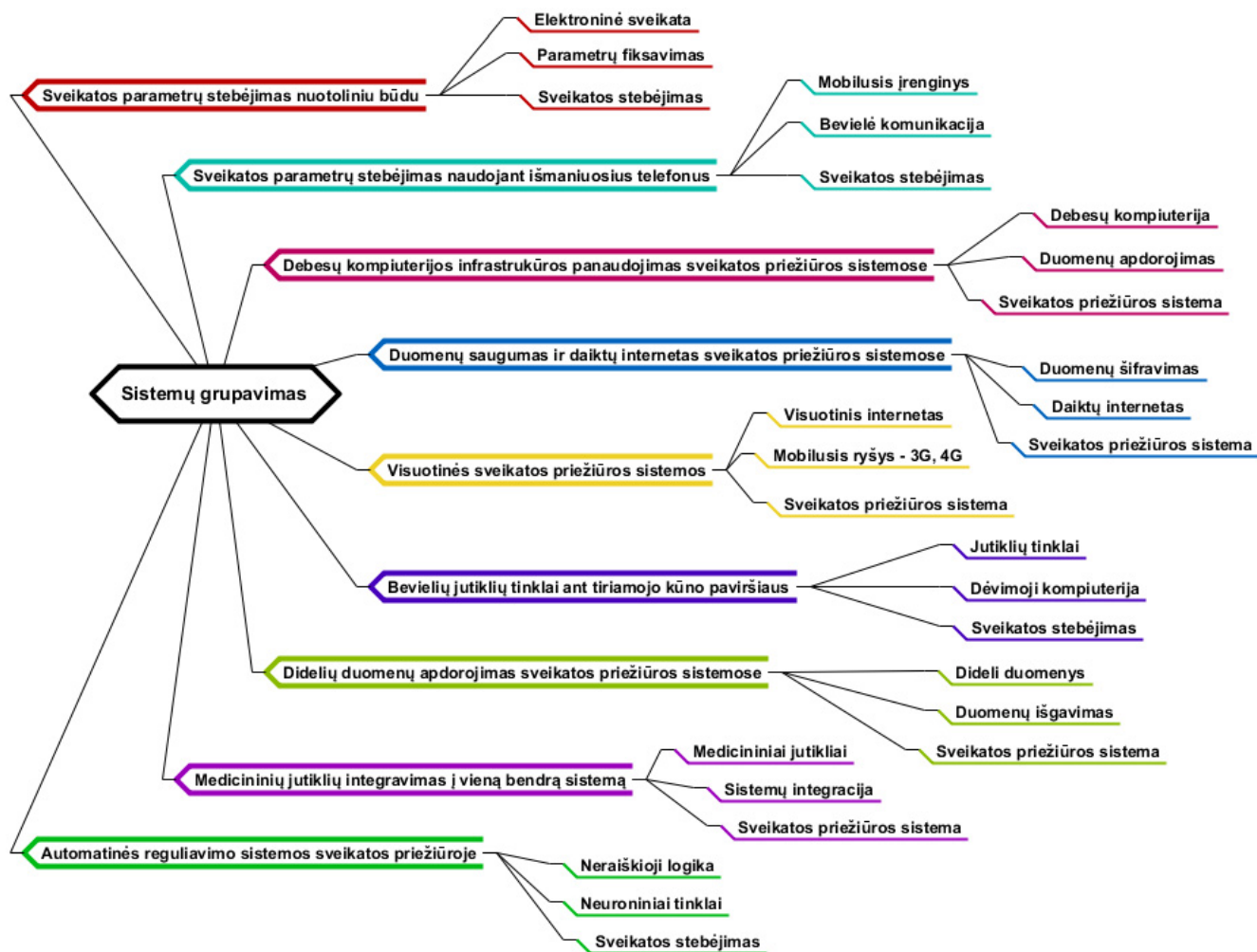
Darbo tikslas – sukurti sveikatos stebėjimo informacinės sistemos su grįžtamoju ryšiu prototipą, leidžianti medikams nuotoliniu būdu stebėti pacientų širdies ir kraujagyslių veiklą.

Siekiant įgyvendinti numatytą tikslą, keliami šie darbo uždaviniai:

1. išanalizuoti sveikatos stebėjimo sistemas;
2. suprojektuoti sveikatos stebėjimo sistemos prototipą ir jam parašyti programinius algoritmus;
3. sukurti sveikatos stebėjimo informacinės sistemos su grįžtamoju ryšiu prototipą, skirtą stebėti pacientų širdies ir kraujagyslių veiklą;
4. verifikuoti sukurtos sistemos pagrindinius komponentus: parametrų fiksavimą jutikliais, mobiliąją programą fiksuojamų parametrų siuntimui į serverį, duomenų apdorojimą debesų serveryje, internetinį puslapį fiksuojamų parametrų atvaizdavimui.

I. ANALIZĖ

Pastaraisiais metais daug dėmesio skiriama moksliniams tyrimams, kurie yra susiję su išmaniosiomis, bevielėmis ir visuotinėmis sveikatos monitoringo sistemomis. Tai rodo didėjantys publikuotų straipsnių skaičiai. Keletas sistemų apžvelgiamos šiame poskyryje. 1 pav. pateikiamas minčių žemėlapis (angl. *mind map*). Jame atvaizduojamas sistemų grupavimas.



1 pav. Sistemų grupavimas

Sistemos sugrupuotos pagal pavadinime naudojamus terminus, raktinius žodžius ir pagal tai kam skiriamas pagrindinis dėmesys. Visos išnagrinėtos sistemos susietos su sveikatos stebėjimu (angl. *health monitoring*), sveikatos priežiūra (angl. *healthcare*).

1.1. Sveikatos parametrų stebėjimas nuotoliniu būdu

Asmens judėjimo ir sveikatos parametrams fiksuoti yra sukurta įvairių prietaisų, kurie uždedami ant vartotojo riešo arba juosmens siekiant fiksuoti kalorijų sunaudojimą, širdies ritmą, kūno temperatūrą ir padėtį (guli, sėdi, stovi). Prietaisas užfiksuotus parametrus nuotoliniu būdu turi perduoti į kitą duomenų peržiūrai skirtą įrenginį, vartotojui nesukeliant nepatogumų. Asmens sveikatos būklės įvertinimui yra sukurta išmanioji veiklos stebėjimo sistema [6]. Ji vadinama išmaniaja, nes asmens

aktyvumo fiksavimo prietaisas gali būti valdomas komandomis nuotoliniu būdu panaudojant serverį. Juo naudojantis atnaujinamos prietaise įdiegtos programos, iš prietaiso į serverį siunčiamos klaidų ataskaitos. Šioms funkcijoms įgyvendinti panaudoti ISO/IEEE 11073 ir OMA DM komunikacijos protokoliai. Užfiksuoti parametrai nuotoliniu būdu gali būti stebimi medicinos personalo.

Lėtinių ligų priežiūros teikimas ne ligoninėse dažnai yra nepakankamas. Todėl kuriamos nuotolinio stebėjimo sistemos, kurios atliktų paciento sveikatos būklės monitoringą, klinikinį vertinimą už ligoninės ribų. Sukurta nuotolinė sveikatos stebėjimo sistema, kuri pagrįsta sąveika tarp skirtingų vartotojų (pacientų, gydytojų, mokslininkų ir inžinierių) [7]. Sistemai reikalingi prietaisai galintys matuoti įvairius fiziologinius parametrus, pavyzdžiui, kraujo spaudimą, širdies susitraukimų dažnį ir temperatūrą. Sistema palaiko nuotolinį programinės įrangos atnaujinimą per aplikacijų platinimo kanalą ir funkcinių komponentų aktyvavimą ir deaktivavimą, kuriuos gali atlikti sveikatos priežiūros specialistas. Sistemos kūrimo metu atlikta reikalavimų specifikacija siekiant sukurti vartotojams kuo tinkamiausią programinę įrangą.

Pastaraisiais metais mobilieji įrenginiai tampa pajėgesni apdoroti daug skaičiavimų resursų reikalaujančius procesus, mobiliojo interneto duomenų perdavimo sparta didėja, ryšio kokybė gerėja, o tai sudaro puikias sąlygas kurti mobiliąsias, nuotolines ir visuotines sveikatos stebėjimo sistemas. Žmogaus gyvenimo kokybė yra vienas iš pagrindinių aspektų išmaniuosiuose miestuose. Daug dėmesio skiriama moksliniams tyrimams, kurie yra susiję su sveikatos stebėjimo sistemomis, kurios yra skirtos pagyvenusių ir neįgalių žmonių priežiūrai. Sveikatos stebėjimo sistemos turi atlikti parametrų fiksavimo, duomenų apdorojimo, saugojimo ir informacijos atvaizdavimo funkcijas. Pagyvenusių ir neįgalių žmonių priežiūrai yra suprojektuotas sveikatos stebėjimo sistemos prototipas [8]. Pagrindinis sistemos uždavinys suteikti realiu laiku medicines paslaugas nelaimingo atsitikimo metu. Sistema sudaryta iš dviejų dalių: lokalsios ir nuotolinės. Nuotolinėje – saugoma informacija apie paciento sveikatos būklę, kuri pateikiama medicinos personalui, lokaliaje – fiksuojami parametrai iš jutiklių. Užfiksuoti parametrai iš Arduino plokštės Bluetooth ryšiu siunčiami į mobilųjį įrenginį, kuris Wi-Fi arba 4G LTE ryšiu perduoda į nuotolinį serverį.

Sveikatos stebėjimo sistemos yra viena iš perspektyviausių sričių norint pagerinti žmonių gyvenimo kokybę. Nuolatinis biomedicininis signalų (kūno temperatūros, širdies ritmo, kraujospūdžio) fiksavimas ypač svarbus diagnozuojant ir gydant širdies ir kraujagyslių ligas. Serveryje saugomi paciento sveikatos parametrai gali būti kaip pagalbinė priemonė gydytojams skiriant pacientui gydymą. Naudojant tinklalapių (angl. *web*) technologijas sukurta nuotolinė sveikatos stebėjimo sistema [9]. Ji fiksuoja paciento širdies ritmą ir kūno temperatūrą, užfiksuoti parametrai saugomi duomenų bazėje. Sistemoje naudojama įranga: Arduino plokštė, LM35 temperatūros jutiklis, LM358 širdies ritmo jutiklis ir CC2500 XBEE modulis. Jutikliais užfiksuoti parametrai iš Arduino plokštės per XBEE modulį perduodami į kompiuterį, o iš jo į serverį, kuriame yra įdiegta duomenų bazė. Paciento

sveikatos parametrai atvaizduojami interneto naršyklėje. Ateityje žadama padidinti sistemos saugumą panaudojant IPv6 protokolą.

Visame pasaulyje per pastarąjį dešimtmetį išaugo pacientų skaičius, kurie suserga sunkiai pagydomomis arba nepagydomomis lėtinėmis ligomis. Pavyzdžiui, neurodegeneracinės ligos (Parkinsono liga, Alzheimerio liga, epilepsija) pažeidžiančios žmogaus nervų sistemą. Europoje šių ligų gydymui skiriamos didelės gydymo išlaidos. Jau egzistuojančios sveikatos stebėjimo sistemos, kurios pagrįstos išmaniaisiais telefonais ir šiuolaikinėmis ryšių technologijomis nėra pritaikytos sprendžiant problemas susijusias su nuolatiniu parametru fiksavimu, didelių duomenų (angl. *big data*) apdorojimu ir integracija su esamų sistemų karkasais. Sunkiai pagydomų arba nepagydomų ligų stebėjimui yra sukurta mobilioji ir „nuo pradžios iki galo“ (angl. *end-to-end*) tinklo protokolu pagrįsta architektūra [10]. Sukurta sąsaja, kuri leidžia architektūrą integruoti su esamomis sistemomis. Elektroencefalografu fiksuojama elektroencefalograma (EEG), kuri perduodama į paciento mobilųjį telefoną. Programa EEG signalą Wi-Fi, 3G arba 4G ryšiu siunčia į tinklalapių serverį. Duomenys saugomi MySQL 5.5 duomenų bazėje, o apdorojami kitame serveryje, kuris pagrįstas debesų kompiuterija. Didelių duomenų apdorojimui naudojami Matlab ir Weka programiniai įrankiai. Medicinos personalui mobiliajame įrenginyje atvaizduojami EEG signalo kanalai, smegenų aktyvumo žemėlapis. Duomenų objektai perduodami JSON formatu, saugumui užtikrinti naudojamas TLS/SSL3 protokolas. Mobiliosios programos kūrimui Android OS, naudota Eclipse IDE 3.7 programinė įranga su ADT įskiepiu (angl. *plugin*).

Bevielė komunikacija, skaitmeninės kameros, išmanieji telefonai, įterptinės sistemos, IoT, visuotinis internetas leidžia sukurti išmaniuosius namus, kuriuose būtų galima vykdyti pacientų priežiūrą, gydymą ir slaugą. Senyvų žmonių slaugai jų pačių namuose yra sukurta sveikatos stebėjimo sistema [11]. Namuose įrengta sistema sudaryta iš dviejų komponentų: jutiklių ir sprendimų priėmimo modulio. Jutikliais fiksuojami sveikatos parametrai, o kamera – asmens emocijos. Parametrai siunčiami į sprendimų priėmimo modulį, kuris juos apdoroja. Jeigu parametrai peržengia leistinas ribas informuojamas slaugytojas. Sprendimų priėmimo modulyje taip pat vykdomas vaizdų apdorojimas. Pagal paciento veido išraišką atpažįstamos emocijos. Jų atpažinimui naudojami penki klasifikavimo metodai: k-artimiausio kaimyno (angl. *k-nearest neighbor*, kNN), sprendimų medis (angl. *decision tree*), neraiškioji logika (angl. *fuzzy logic*), Bajeso tinklai (angl. *Bayesian networks*) ir atraminių vektorių klasifikatorius (angl. *support vector machine*, SVM). Atliktas emocijų atpažinimo tikslumo tyrimas, kurio metu nustatyta, kad geriausi rezultatai gauti naudojant SVM.

1.2. Sveikatos parametru stebėjimas naudojant išmaniuosius telefonus

Šiuolaikiniuose išmaniuosiuose telefonuose integruoti įvairūs jutikliai. Tai leidžia mokslininkams kurti taikomąsias programas norint išspręsti problemas su kuriomis susiduriama

įvairiose srityse. Išnagrinėtos sveikatos parametrų stebėjimo sistemos išmaniesiems telefonams gyvenimo kokybės gerinimui [12]. Telefone integruotas akselerometras gali būti naudojamas kūno laikysenos ir griuvimo stebėjimui, kamera – širdies ir kraujagyslių sistemos parametrų fiksavimui, mikrofonas – kvėpavimo sistemos sutrikimų atpažinimui. Europa turi didžiausią amžiaus vidurkį pasaulyje, kur apie 20% populiacijos yra 65 metų ir vyresnio amžiaus. Išmanieji telefonai gali būti naudojami vykdant raumenų sistemos ligų stebėjimą ir griuvimo aptikimą pagyvenusiems žmonėms. Nuolatinis širdies ritmo ir kraujospūdžio stebėjimas išmaniaisiais telefonais gali padėti užkirsti kelią širdies ir kraujagyslių ligoms. Asmenims nereikėtų su savimi nešioti kitų prietaisų parametrų fiksavimui ir nuolat lankytis medicinos įstaigoje. Žmonių skaičius, sergančių kvėpavimo takų ligomis (astma, lėtine obstrukcine plaučių liga), nuolat didėja. Kosulio ir kvėpavimo analizė gali būti atliekama išmaniajame telefone. Nuolat stebint kvėpavimo takų sistemą paaiškėja koks reikalingas gydymas.

Greitas gyvenimo tempas, nuolatinė laiko stoka sukelia sunkumų rūpintis senyvo amžiaus šeimos nariais. Netolimoje ateityje vienišų ir neprižiūrimų senyvo amžiaus žmonių skaičius išaugs. Naudojant bevielį parametrų fiksavimo įrenginį ir išmanųjį telefoną yra suprojektuotas sveikatos stebėjimo sistemos prototipas [13]. Jis sudarytas iš fiksavimo įrenginio, kuris integruotas į krūtinės diržą ir bazinės stoties, kurios funkcijos sistemoje atlieka išmanusis telefonas. Įrenginys fiksuoja sveikatos ir aplinkos parametrus, kuriuos siunčia į bazinę stotį. Komunikuojama bevieliu ryšiu naudojant Texas Instruments CC2541 modulį su BLE 4.0 (Bluetooth Low Energy). Asmens širdies ritmas fiksuojamas naudojant AFE grandinę. Užfiksuotas signalas daugelyje atvejų yra su triukšmu, todėl jis yra apdorojamas dviejų baigtinių impulso atsako (FIR) žemo dažnio ir aukšto dažnio filtrais. Taip pat sistemoje naudojamas akselerometras norint nustatyti kūno padėtį. Mobiliajame telefone pateikiamas širdies ritmas, kūno padėtis, įspėjimai ir oro temperatūra. Bevielės sveikatos stebėjimo sistemos gali išplėsti vyresnio amžiaus žmonių nepriklausomybės laikotarpį, sumažinti sveikatos priežiūros paslaugų investicijas, palengvinti ir paspartinti diagnostikos procesą. Tokioms sistemoms keliami reikalavimai: maitinimo šaltinio talpa turi būti didelė ir fiksavimo įrenginys turi būti pakankamai mažas, kad vartotojui nesukeltų nepatogumų, netrukdytų atlikti kasdieninius darbus.

Vyresnio amžiaus asmenys susiduria su įvairiomis sveikatos problemomis. Reguliarus ir taisyklingas vaistų vartojimas susilpnina ligos simptomus, pagerina savijautą ir padeda išvengti komplikacijų. Daugiau kaip 50% senyvo amžiaus žmonių susiduria su sunkumais, kurie yra susiję su vaistų vartojimo režimo laikymosi. Senyvo amžiaus žmonėms sukurta sistema, kuri palengvina laikytis vaistų vartojimo režimo [14]. Mobiliajame įrenginyje reguliariai pranešama ir primenama kada pacientas turi vartoti vaistus. Sistema veikia įrenginiuose turinčiuose Android OS. Sistemos įvertinimui buvo atliktas eksperimentas. Vyresnio amžiaus žmonės aštuonias savaites turėjo reguliariai vartoti kalcio tabletę pagal nurodytą režimą. Sistema labiausiai pravertė pirmas keturias savaites, o po to eksperimento dalyviai pamažu patys pradėjo reguliariai laikytis vaistų vartojimo režimo be sistemos

priminimo. Norint įvertinti sukurtą mobiliąją programą iš vartotojo perspektyvos, jiems buvo pateikta anketa su klausimais. Pagal atliktą apklausą buvo nustatyta, kad senyvo amžiaus žmonėms sukurtas sistemos prototipas buvo naudingas laikantis vaistų vartojimo režimo.

Širdies ir kraujagyslių ligos (angl. *cardiovascular diseases*, CVDs) yra viena iš pagrindinių mirties priežasčių. Dauguma šių ligų galima išvengti arba išgydyti, jeigu laiku imtasi reikalingų priemonių, įskaitant diagnostiką, prognozę, analizę. Tačiau šiems metodams atlikti reikalingos didelės išlaidos. Didėjant bevielių ir nešiojamų prietaisų ir jutiklių skaičiui suteikiama galimybė sukurti mobiliąsias širdies ir kraujagyslių ligų stebėjimo sistemas. Atlikta apžvalga medicininių prietaisų ir mobiliųjų įrenginių programų, skirtų fiksuoti su širdies ir kraujagyslių ligomis susijusius parametrus [15]. Išnagrinėti prietaisai kraujo tyrimams atlikti, nustatant cukraus ir cholesterolio kiekį kraujyje, mobiliosios programos: Instant Heart Rate – širdies ritmo fiksavimui ir Instant Blood Pressure – kraujo spaudimo matavimui.

Mobiliųjų prietaisų tobulėjimas, IoT ir visuotinis internetas brandina naują technologiją: išmaniuosius asmeninius asistentus-padėjėjus (angl. *intelligent personal assistants*, IPA). IPA tai programinės įrangos agentai, kurie naudojami padėti vartotojams įvykdyti kasdienes užduotis, išklausyti pageidavimus ir pateikti patarimus. Norint tai atlikti IPA turi analizuoti savo vartotojo elgesį ir kasdienybę, sąveikauti su kitais išmaniaisiais objektais, kad įgytų naujų žinių ir supratimą apie vartotoją. Sveikatos priežiūrai (angl. *m-Health*) yra sukurta mobilioji tinklo sąsaja (angl. *gateway*) [16]. Sistemoje fiksuojama paciento buvimo vieta, širdies ritmas, kūno padėtis (sėdi, guli ir pan.) ir aktyvumas. Buvimo vieta nustatoma su išmaniajame telefone integruotu GPS, o paciento padėtis – akselerometru. Širdies ritmas fiksuojamas Android išmaniuoju laikrodžiu. Fiksuojama informacija perduodama Bluetooth LE ryšiu į išmanųjį telefoną. Aktyvumui nustatyti naudojama Google Activity Recognition API. Naudojant ją nustatoma vartotojo veikla: vaikščiojimas, bėgimas, važiavimas dviračiu, transporto priemonės vairavimas. Pagal šią pirminę informaciją IPA galės nustatyti paciento veiksmus, atitinkamai informuoti reikiamus asmenis. Tinklų sąsaja, kuri atsakinga už informacijos siuntimą paciento priežiūrėtojo IPA platformai, vadinama AMBRO. Tai technologijų platforma, kuri palaiko komunikaciją tarp kitų IPA per visuotinį internetą. Priežiūrėtojas gauna pranešimus ir įspėjimus apie stebimą asmenį.

1.3. Debesų kompiuterijos infrastruktūros panaudojimas sveikatos priežiūros sistemose

Naujausios technologijų tendencijos, pavyzdžiui, debesų kompiuterija, bevielė komunikacija, bevieliai jutikliai tinklai leidžia sukurti sveikatos priežiūros informacinių technologijų paslaugas per internetą. Sveikatos priežiūrai yra suprojektuota mobilioji informacinė rekomendacijos sistema [17]. Ji pagrįsta debesų kompiuterija, mobiliojo ryšio technologija, konteksto supratimo (kur esate, su kuo esate, kokie resursai yra netoliese) technologija ir bevieliais jutiklių tinklais. Sistemą sudaro dvi

paslaugų funkcijos: rekomendacija, kuri pagrįsta užfiksuotų parametrų apdorojimu, pacientų ir gydytojų bendradarbiavimas. Jutikliais užfiksuoti parametrai perduodami į debesų kompiuterijos infrastruktūrą. Joje vykdomas apdorojimas, rezultatai siunčiami atgal vartotojui. Sistema sukurta naudojant MySQL duomenų bazę, HTML5 žymių kalbą, CSS stilių kalbą, jQuery biblioteką ir JavaScript, PHP programavimo kalbas. Sistema pagrįsta tinklalapių technologijomis, todėl mobiliuosiuose įrenginiuose gali būti įdiegta iOS, Android, Windows Phone arba kita OS.

Dauguma sveikatos priežiūros specialistų po studijų lieka dirbti didžiuosiuose miestuose. Rajonų centruose ir mažesnėse gyvenvietėse jaučiamas šių specialistų trūkumas. Vienas iš problemų sprendimų – nuotolinės sveikatos stebėjimo sistemos diegimas. Naudojant debesų kompiuterijos infrastruktūrą sukurta sistema skirta nuotoliniu būdu stebėti nuo didžiųjų miestų nutolusių gyventojų sveikatą [18]. Bluetooth arba Wi-Fi ryšiu jutiklių duomenys perduodami į mobilųjį telefoną arba į planšetinį kompiuterį. Mobilioji programa duomenis siunčia į debesų serverį. Jame tikrinama ar parametrai neperžengia leistinų ribų. Sveikatos priežiūros specialistai ir pacientas informuojami apie sveikatos būklę. Sistemoje vykdomas duomenų šifravimas siekiant užtikrinti paciento sveikatos informacijos privatumą ir konfidencialumą. Naudojamas tapatybe pagrįstas šifravimas (angl. *Identity Based Encryption*, IBE). Tai vienas iš asimetrinių algoritmų tipų.

Mobilieji įrenginiai šiais laikais naudojama ne tik kaip komunikacijos priemonė, bet ir plačiai taikomi sveikatos priežiūros paslaugose, pavyzdžiui, matuojant širdies ritmą, vykdant sveikatos stebėjimą, sunaudotų kalorijų skaičiavimą, fizinio aktyvumo nustatymą ir t. t. Mobiliosios elektroninės sveikatos paslaugos suteikia galimybę medicinos personalui stebėti pacientų būklę ir kasdieninę veiklą, pavyzdžiui, prabudimų skaičių naktimis diagnozuojant miego sutrikimus, fizinį aktyvumą, mitybos įpročius ir t. t. Mobilųjų įrenginių skaičiavimo pajėgumai nuolat didėja, juose įdiegiami didesnės galios procesoriai, vaizdo plokštės, didesnis kiekis operatyviosios atminties, failų saugojimui skirtos atminties ir t. t. Tačiau mobilieji įrenginiai apriboti baterijos veikimo laiku. Daug skaičiavimų resursų reikalaujantys algoritmai, pvz. kompiuterinės regos (angl. *computer vision*), vaizdų, video ar kitos multimedijos apdorojimo, vaizdų atpažinimo, neuroniniai tinklai, greitai eikvoja mobiliojo įrenginio bateriją. Siekiant įveikti šį apribojimą, mobiliosios programos dalis, kurios reikalauja daug skaičiavimų resursų galima perkelti į debesų kompiuterijos infrastruktūrą. Naudojant debesų kompiuterijos infrastruktūrą sukurta vaizdų apdorojimo sistema ir mobilioji multimedijos programa elektroninėms sveikatos paslaugoms [19]. Sistema vykdo vaizdų fiksavimą ir apdorojimą siekiant vaizde aptikti, atpažinti ir klasifikuoti maistą, o pagal tai nustatyti kalorijų skaičių. Vaizdų fiksavimas vykdomas naudojant išmanųjį telefoną. Vaizdas nusiunčiamas į debesų kompiuterijos infrastruktūrą. Užfiksuotame vaizde aptinkami lėkštėje pateikti maisto produktai. Jie klasifikuojami ir atpažįstami. Jeigu produktas sudėtinis analizuojami jo ingredientai. Apskaičiuotas bendras skaičius kalorijų maiste atsiunčiamas atgal į mobilųjį įrenginį. Maisto atpažinimui naudojami du metodai: atraminių vektorių

klasifikatorius (angl. *support vector machines*, SVM) ir gilus mokymas (angl. *deep learning*). Kalorijų nustatymui reikalingas maisto porcijos dydis. Kalibravimas atliekamas remiantis šalia lėkštės padėta liniuote. Vaizdų apdorojimui naudota paskirstyta Amazon Web Services (AWS) EC2 debesų kompiuterijos infrastruktūra. Apdorojimas vykdomas keliuose EC2 serveriuose siekiant vartotojui pateikti atsaką per kuo trumpesnę laiką. Atliktas tyrimas naudojant g2.2xlarge Amazon EC2 debesų infrastruktūrą. Sukurtos keturios virtualios mašinos, įdiegta Ubuntu Server 14.04 LTS OS. Vaizdų apdorojimas atliktas naudojant GPU. Įdiegtos Nvidia tvarkyklės (angl. *drivers*) ir programinės įrangos kūrimo įrankiai CUDA SDK 6.5. Vaizdo apdorojimo algoritmo kūrimui naudota OpenCV biblioteka, o apmokymo modeliui – Python 2.7 programavimo kalba ir bibliotekos Numpy, Scipy. Tyrimo metu stebėta apdorojimo proceso trukmė priklausomai nuo pateiktų vaizdų skaičiaus. Vaizdai buvo apdorjami lygiagrečiai keturiuose VM. Nustatyta, kad didėjant vaizdų skaičiui apdorojimo proceso trukmė ilgėja eksponentiškai.

Lėtinės kvėpavimo takų ligos (astma, lėtinė obstrukcinė plaučių liga, miego apnėja, plautinė hipertenzija) viena iš labiausiai paplitusių sveikatos problemų pasaulyje. Šių ligų gydymas dažniausiai vyksta namuose, pacientas turi būti nuolat prižiūrimas šeimos nario. Ligonio priežiūra namų sąlygomis, nuolatinė rutina apsunkina globėjo kasdieninio gyvenimo veiklą, vienam iš artimųjų dažniausiai tenka atsisakyti darbo. Pastaraisiais metais mobiliosios (angl. *mHealth*) ir elektroninės (angl. *eHealth*) sveikatos priežiūros sistemos pradėtos plačiai taikyti nuotoliniam pacientų stebėjimui. Mažesnis darbo krūvis tenka globėjui arba šeimos nariui, medicinos personalui nuotoliniu būdu prižiūrint pacientą. Naudojant debesų kompiuterijos infrastruktūrą sukurta mobilioji sveikatos priežiūros sistema skirta namų sąlygomis stebėti pacientą sergantį lėtinėmis kvėpavimo ligomis [20]. Sistema sudaryta iš trijų komponentų: debesų serveris – duomenų saugojimui ir apdorojimui, mobilioji programa ir internetinis puslapis – paciento sveikatos parametrų atvaizdavimui medicinos personalui, programa įterptinėje sistemoje – parametrų fiksavimui. Sistemoje naudojama Microsoft Azure debesų kompiuterijos platforma, informacijos saugojimui pasirinkta NoSQL duomenų bazė. Naudojant PubNub technologiją duomenys siunčiami suinteresuotiems asmenims (medicinos personalui, globėjui) naudojantiems skirtingų platformų įrenginius. Vartotojo sąsajoje atvaizduojamas kraujo įsisotinimas deguonimi (SpO₂), širdies ritmas ir paciento ryšys su jutikliu (prijungtas, atjungtas). Sistemoje įdiegtas modulis, kuris generuoja savaitės ataskaitas apie stebimo paciento sveikatą. Pateikiami perspėjimai apie sveikatos nukrypimus. Tai padeda diagnozuojant miego sutrikimus. Nuotolinis ligonio sveikatos parametrų stebėjimas namų sąlygomis leidžia mažinti ligoninių išlaidas. Užfiksuoti paciento sveikatos parametrai Bluetooth ryšiu siunčiami į mobilųjį įrenginį su Android OS, o iš jo Wi-Fi ryšiu – į debesų serverį. Širdies ritmas ir SpO₂ matuojami naudojant BM2000A pulsoksimetrą.

Kalbos signalų apdorojimas ir atpažinimas pastaraisiais metais plačiai taikomos įvairiose srityse: robotų valdyme balso komandomis, automatiniuose pokalbių vertėjuose, kalba į tekstą (angl. *speech-*

to-text) programose, neurodegeneracinių ligų (Alzheimerio, Parkinsono, Huntingtono) diagnozavimui ir t. t. Naudojant debesų kompiuterijos infrastruktūrą sukurtas karkasas (angl. *framework*) Parkinsono ligos diagnozavimui ir stebėjimui [21]. Parkinsono liga pasireiškia judesių, kalbos, mąstymo sutrikimais. Paciento balso signalas užfiksuojamas mobiliąja programa, o iš jos siunčiamas į debesų serverį. Jame vykdomas skaitmeninis signalo apdorojimas (angl. *digital signal processing*, DSP) ir išsaugomas duomenų bazėje. Iš signalo pašalinami triukšmai, išskiriami požymiai (angl. *feature extraction*), o pagal juos signalas klasifikuojamas siekiant nustatyti įtariamą ligą. Balso signalo klasifikavimui naudotas FBAAN. Tai tiesioginio sklidimo dirbtinis neuroninis tinklas (angl. *feedforward artificial neural network*), kurio apmokymui naudotas klaidos sklidimo atgal algoritmas (angl. *backpropagation*). Pagal gautus klasifikavimo rezultatus nustatomi balso sutrikimai – disfonija (angl. *dysphonia*). Esant disfonijai informuojamas atitinkamos specializacijos gydytojas, kuris yra prisijungęs prie debesų infrastruktūros. Gydytojui atsiunčiamas paciento balso signalas pagal kurį atliekama ligos diagnozė. Pacientas automatiškai informuojamas apie gautus tyrimo rezultatus. Sistemoje atliktas tyrimas, kurio metu palyginti trys kalbos signalų požymių klasifikatoriai: FBAAN, atsitiktinių medžių (angl. *random tree*) ir atraminių vektorių klasifikatorius (angl. *support vector machine*, SVM). Geriausi klasifikavimo rezultatai gauti naudojant FBANN. Algoritmo kūrimui naudota Matlab programinė įranga.

1.4. Duomenų saugumas ir daiktų internetas sveikatos priežiūros sistemose

Didelis kiekis duomenų apie paciento sveikatos būklę yra surenkami iš įvairių prietaisų ir jutiklių. Dabartinės sveikatos priežiūros sistemos neišsprendžia problemų, kurios susijusios su duomenų prieinamumu ir saugumu. Informacija apie pacientą ir jo sveikatos būklę turi būti prieinami tik medicinos personalui. Sveikatos priežiūros paslaugoms sukurta architektūra surinkti didelį kiekį duomenų iš jutiklių tinklų [22]. Architektūra sprendžia su saugumu susijusias problemas ir palengvina informacijos apsikeitimą tarp sveikatos priežiūros specialistų įprastomis sąlygomis ir nenumatytais atvejais. Siūlomas saugumo mechanizmas, kuris garantuoja konfidencialumą, vientisumą prieigai prie medicininių duomenų. Šis mechanizmas remiasi šifro teksto strategija ir atributais paremtu šifravimu (angl. *Ciphertext Policy Attribute based Encryption*, CP-ABE). Architektūrą sudaro šie komponentai: bevielis jutiklių tinklas – renka informaciją apie paciento sveikatos būklę, programa – leidžia sveikatos priežiūros specialistams prieiti prie saugomų duomenų, administracija – nurodo ir užtikrina saugumo politiką sveikatos priežiūros įstaigoje, debesų serveriai – juose įdiegta duomenų saugykla. Naudojant pateiktą architektūrą sveikatos priežiūros įstaigai nereikia rūpintis serverių valdymu, kuris jai gali būti sudėtingas. Siekiant užtikrinti prieigos kontrolę naudojamas atributais paremtas šifravimas (ABE). Prieš patalpinant duomenis į debesų kompiuterijos infrastruktūrą, jie yra užšifruojami. ABE integravimas į medicininės sistemas yra vienas iš iššūkių. Naudojant ABE duomenys yra užšifruojami

su prieigos struktūra, kuri yra prieigos strategijos loginė išraiška (pvz., duomenys gali pasiekti kardiologijos skyriaus gydytojai arba slaugytojai). Užšifruotus duomenis gali iššifruoti bet kuris vartotojas, jeigu jam suteiktas slapta raktas turi atributus, kurie atitinka prieigos strategiją. Pačiame serveryje imtis griežtų saugojimo priemonių nebereikia, nes duomenų bazėje visi duomenys užšifruoti. Sistemoje sprendžiama problema kas turėtų generuoti prieigos struktūrą, kuri reglamentuos saugumą ir kas turėtų generuoti ir platinti raktus.

Naujausi pasiekimai informacinėse ir komunikacijų technologijose, įterptinėse sistemose brandina naują technologiją: daiktų internetą (angl. *Internet of Things*, IoT). IoT tai tarpusavyje interneto tinklu susijungę įrenginiai. Perduodant informaciją per tinklą naudojama ne tik „žmogus-žmogus“ ar „žmogus-kompiuteris“ sąveikos, bet ir sąveika „įrenginys-įrenginys“. Didėjančios sveikatos priežiūros išlaidos, lėtinių ligų paplitimas visame pasaulyje, reikalauja pokyčių esamose sveikatos priežiūros sistemose. Gydymas vyksta ligoninėse, o ateityje norima padaryti, kad gydymą būtų galima suteikti ir namuose. Kai tinklai diegiami plačiu mastu atsiranda su saugumu susijusių problemų. IoT pagrįstos sveikatos priežiūros sistemos sąveikauja su paciento sveikata susijusia informacija. Iš jutiklių surinkti duomenys bevieliu ryšiu iš vieno įrenginio perduodami į kitą. Kadangi sveikatos priežiūros sistemose, kurios pagrįstos IoT, vyksta bevielė komunikacija tarp įrenginių, todėl saugumas ir privatumas – pagrindiniai iššūkiai. Tradiciniai saugumo mechanizmai: kriptografijos algoritmai, saugumo protokolai, privatumo užtikrinimas negali būti taikomi IoT pagrįstose sveikatos priežiūros sistemose dėl išteklių apribojimų. Su saugumu susiję sprendimai turi būti efektyvūs, nes medicininių jutiklių mazgai turi apribotą galią, atmintį ir ryšio pralaidumą. Taigi tradicinių kriptografijos metodų naudojimas medicininių jutiklių mazguose tampa neįgyvendinamas dėl to, kad šie metodai naudoja daug resursų reikalaujančius skaičiavimus. Sveikatos priežiūros sistemoms, kurios yra pagrįstos IoT, sukurta saugi ir efektyvi autentiškumo nustatymo architektūra [23]. Architektūrą sudaro šie pagrindiniai komponentai: medicininių jutiklių tinklas – fiksuoja biomedicininis signalus, tinklų sietuvas (angl. *gateway*) – į jį perduodami signalai bevieliu arba laidiniu ryšiu naudojant protokolus: Serial, SPI, Bluetooth, Wi-Fi arba IEEE 802.15.4, debesų kompiuterijos platforma – apima duomenų bazę ir didelių duomenų (angl. *big data*) analizės serverius. Debesų platformoje vyksta sveikatos duomenų apdorojimas. Medicinos darbuotojams paciento sveikatos parametrai atvaizduojami interneto naršyklėje. Architektūroje naudojami saugumo protokolai, kuriais įgyvendinama sistemos vartotojo autentifikacija – asmens atpažinimas prieš suteikiant informaciją ir autorizacija – nustatoma ar identifikuotas vartotojas turi teisę naudotis paslaugomis.

Daiktų internetas, tarpusavyje bevieliu ryšiu sujungti medicininiai prietaisai ir jutikliai, yra populiarėjanti technologija, kuriant naujos kartos sveikatos priežiūros sistemas. Jos leistų pagerinti žmonių gyvenimo kokybę ir prailginti trukmę. EKG signalo fiksavimui mobiliaisiais įrenginiais arba jutikliais sukurtas sveikatos stebėjimo karkasas, kuris pagrįstas IoT [24]. Jutikliais užfiksuoti

parametrai siunčiami į debesų serverį. Jame esanti informacija prieinama sveikatos priežiūros specialistams. Serveryje EKG signalas apdorojamas: vykdomas QRS komplekso aptikimas, RR intervalo ir širdies ritmo apskaičiavimas. Signalu filtravimui naudojama Hilberto transformacija, kuri pastumia signalo dažnio komponentų fazes. Ši transformacija leidžia išspręsti EKG signalo iškraipymo problemą, suvienodinti R bangų amplitudes. R bangos aptinkamos pagal slenkstį (angl. *threshold*). Į EKG signalą įterpiamas skaitmeninis vandens ženklas (angl. *digital watermarking*) siekiant apsaugoti informaciją nuo klastojimo. Žymėjimas vandens ženklu tai procesas, kurio metu įterpiama nedidelė dalis informacijos neiškraipant signalo. Sistema į EKG signalą įterpia kliento registracijos ID.

Informacinių ir komunikacijų technologijų pažanga tapo nepakeičiama priemone daugelyje sektorių šiuolaikinėje visuomenėje, įskaitant sveikatos priežiūrą. Šios technologijos leidžia optimizuoti turimus išteklius, tampa efektyvesnis administracinis ir klinikinis valdymas. Elektroninė sveikata (angl. *eHealth*) tampa alternatyva tradicinėms medicinos sistemoms. El. sveikatos sąvoka apima ne tik paciento sveikatos priežiūrą, bet ir jo paties dalyvavimą gydyme. Daiktų interneto tikslas visus įrenginius sujungti interneto ryšiu, kurie tarpusavyje komunikuoju be žmogaus įsikišimo. Šio pobūdžio komunikacijoje naudojami įrenginys į įrenginį (angl. *Machine to Machine*, M2M) protokolai. Šiuo metu yra sukurta daug M2M protokolų, todėl yra atlikta jų lyginamoji analizė norint nustatyti, kuris atitinka el. sveikatos sistemų reikalavimus [25]. Pasirinktas konkretus atvejis: kvėpavimo reabilitacija lėtinei obstrukcinei plaučių ligai. Nagrinėti du skirtingų tipų el. sveikatos reikalavimai: namų sąlygomis ir ambulatorinis gydymas. Pasirinkti M2M protokolai: DDS, MQTT, CoAP, JMS, AMQP ir XMPP, kurie buvo vertinami pagal įvairius rodiklius: CPU ir atminties sąnaudas, tinklo pralaidumą, uždelsimą ir jo varijavimą (angl. *jitter*). Tyrimas atliktas naudojant Distributed Computing Framework programinį įrankį. Atsižvelgiant į gautus rezultatus ir į el. sveikatos keliamus reikalavimus (greita, patikima, artima realaus laiko komunikacija ir t. t.) ambulatorinio gydymo atveju pasirinktas DDS protokolas. Jis pasižymi mažu uždelsimu ir varijavimu, prarastų duomenų paketų skaičiumi, atsparumu gedimams (angl. *fault tolerance*). Gydymui namų sąlygomis buvo pasirinktas MQTT protokolas. Jis mažai naudoja CPU ir atminties resursų, didelis skaičius įterptinių sistemų ir mobiliųjų įrenginių, kurie turi suderinamumą su protokolu, pasižymi mažomis elektros energijos sąnaudomis. Abu pasirinkti protokolai DDS ir MQTT įvykdo keliamus el. sveikatos saugumo, privatumo reikalavimus.

1.5. Visuotinės sveikatos priežiūros sistemos

Visuotinė sveikatos priežiūros sistema – tai sistema, kurioje sveikatos priežiūros paslaugos prieinamos bet kam, bet kada ir bet kur. Ši sistema gali sumažinti išlaidas ir pagerinti sveikatos priežiūros paslaugų kokybę. Visuotinės sistemos apima plataus masto bevielių tinklų diegimą siekiant pagerinti komunikaciją tarp sveikatos priežiūros specialistų ir pacientų. Visuotinės sveikatos priežiūros

sistemos pavyzdys – mobilioji telemedicina. Sveikatos priežiūros paslaugoms yra pateiktas teorinis informacinės sistemos kelių pakopų karkasas (angl. *framework*), kuris apima etapus: inicijavimą, pasiruošimą tyrimams, tyrimą ir atlikto tyrimo vertinimą [26]. Pateiktas karkasas leistų sumažinti ilgalaikes sąnaudas ir užtikrinti duomenų saugumą. Visuotinė sveikatos priežiūros informacinė sistema leistų sumažinti užkrečiamųjų ligų plitimą, nes naudojant ją nuotoliniu būdu galima stebėti sveikatos parametrus, medicinos personalui komunikuoti su pacientais. Jeigu sveikatos priežiūros įstaigoje nėra įdiegta informacinė sistema, slaugytojos turi pacientams pamatuoti kraujo spaudimą, temperatūrą, kvėpavimo dažnį, pulsą ir kitus parametrus, o visus duomenis rankiniu būdu surašyti į ligos istoriją. Atliekant medicininių dokumentų pildymą rankiniu būdu sugaištama daug laiko ir dažnai padaroma klaidų. Visuotinė sveikatos stebėjimo sistema gali sumažinti duomenų suvedimo laiką ir pagerinti tikslumą.

Greitas bevielų prieigos tinklų ir aukštos kokybės mobiliųjų įrenginių tobulėjimas sudaro puikias sąlygas kurti elektronines sveikatos priežiūros paslaugas. Joms skiriama daug dėmesio, o ypač – mobiliosioms. Nuotolinis paciento sveikatos būklės stebėjimas, telekonsultacijos reikalauja gretimo realaus laiko (angl. *near real time*) komunikacijos ir geros ryšio kokybės (nedidelis vėlinimas, greitas atsako laikas, mažai prarandama duomenų paketų). Didėjant bevielų tinklų prieigos taškų skaičiui atsiranda tinkamos sąlygos kurti mobiliąsias ir visuotines sveikatos priežiūros sistemas. Visuotinėms ir mobiliosioms realaus laiko sveikatos priežiūros paslaugoms sukurta išmanioji bevielio prieigos taško pasirinkimo sistema [27]. Pateiktas sprendimas kaip rasti netoliese esančią bazinę stotį, kurios ryšys aprėpia dabartinę mobiliojo įrenginio buvimo vietą. Iš nešiojamų jutiklių (EKG, širdies ritmo, kūno temperatūros) ir išmaniuosiuose telefonuose integruotų jutiklių (aukštos raiškos kamera, GPS, akselerometras, giroskopas) surenkami duomenys ir nusiunčiami į nuotolinį serverį, kuris prieinamas sveikatos priežiūros centrui. Širdies ritmo matavimui naudojamas Zephyr HxM krūtinės diržas, o vaizdų fiksavimui Samsung Note 3 mobilusis telefonas. Išmaniajame laikrodyje Samsung Gear Fit pateikiami pranešimai, kurie buvo gauti iš ligoninės arba iš gydymo įstaigos. Remiantis mobiliojo telefono būvimo vieta sistema nustato galimus prisijungimo taškus. Sistema prisijungia prie to prieigos taško, kurio stiprumas didžiausias ir pradeda duomenų perdavimą.

1.6. Bevielų jutiklių tinklai ant tiriamojo kūno paviršiaus

Bevielų ant tiriamojo kūno paviršiaus dedamų jutiklių tinklai (angl. *Wireless Body Area Network*, WBAN) naudojami įvairiose srityse: sveikatos priežiūros, stebėjimo, medicinos, sporte ir t. t. WBAN susideda iš medicinos jutiklių (EKG, EEG ir pan.), kurie uždedami ant tiriamojo žmogaus kūno. Jutiklių duomenys perduodami į išmanųjį telefoną ar planšetinį kompiuterį. Šie įrenginiai renka, saugo ir apdoroja paciento sveikatos parametrus (širdies ritmą, kraujo spaudimą, kūno temperatūrą ir pan.) ir perduoda į serverį. Nuotolinės sveikatos stebėjimo sistemos tikslas sumažinti hospitalizacijų

skaičių, sutrumpinti buvimo laiką gydymo įstaigose. Kadangi mobilieji įrenginiai palaiko įvairias bevielio ryšio technologijas: Wi-Fi, 2G, 3G, 4G (LTE) ir pan., todėl reikia pasirinkti prieinamą tinklą norint išlaikyti nuolatinį sveikatos parametrų perdavimą į nuotolinį serverį. Sukurtas algoritmas, kuris paciento mobiliajame įrenginyje dinamiškai parenka tinkamiausią tinklą atsižvelgiant į signalo stiprumą (RSSI), perdavimo spartą, prarastų paketų skaičių ir kitus parametrus [28]. Algoritmas sukurtas naudojant OMNeT++ simuliacijos įrankį. Sudarytas tinklas susidedantis iš 10 mobiliųjų WBAN, simuliacijos plotas $1000 \times 1000 \text{ m}^2$, apręptas aštuoniais Wi-Fi prieigos taškais, keturiomis 3G ir keturiomis LTE bazinėmis stotimis. WBAN juda 0.5 m/s greičiu atsitiktinai parinktu maršrutu ir į serverį siunčia fiksuojamus sveikatos parametrus. Simuliacijos rezultatai parodė, kad pateiktas algoritmas sunaudoja mažiau mobiliojo įrenginio baterijos sąnaudų, mažesnis prarastų paketų skaičius ir užlaikymas, lyginant su DR-VHD ir RSS-VHD tinklo parinkimo algoritmais.

Pažangios bevielio ryšio ir komunikacijų technologijos leidžia įgyvendinti išmaniąsias sveikatos priežiūros paslaugas. Beveliai ant tiriamojo kūno paviršiaus dedamų jutiklių tinklai (WBAN) plačiai taikomi sveikatos priežiūros srityse. Sveikatos parametrų fiksavimas, saugojimas, apdorojimas ir valdymas yra reikšmingi sveikatos priežiūros paslaugų procesai. Naudojant WBAN ir debesų kompiuteriją sukurta tinklo architektūra sveikatos priežiūros paslaugoms [29]. Architektūra sudaryta iš keturių sluoksnių: fiksavimo, tinklo, debesų kompiuterijos ir taikomojo. Su sveikata susiję parametrai fiksuojami jutikliais. Naudojant ZigBee bevielio ryšio technologiją užfiksuoti parametrai perduodami į vietinius duomenų surinkimo mazgus. Juose vykdomas pradinis duomenų atrinkimas ir apdorojimas. Naudojant Wi-Fi bevielį ryšį arba mobiliuosius tinklus (3G, LTE) duomenys siunčiami į debesų serverį. Jame vykdomas duomenų saugojimas, apdorojimas ir valdymas, saugumo ir privatumo užtikrinimas, vartotojo autentifikacija ir autorizacija, apkrovų paskirstymas. Medicinos personalui ir pacientams programoje atvaizduojama reikiama informacija. Atliekamas lėtinių ligų stebėjimas, pooperacinė priežiūra, suteikiamos telemedicinos paslaugas. Architektūra suprojektuota naudojant Opnet tinklų simuliacijos įrankį.

1.7. Didelių duomenų apdorojimas sveikatos priežiūros sistemose

Skaitmeninės informacijos kiekis ir įvairovė nuolat didėja, todėl reikalingas jos apdorojimas vykdant didelių duomenų analizę. Sveikatos priežiūros sistemos saugo didelį kiekį duomenų apie pacientų sveikatą, tyrimus, ligos istoriją ir pan. Daugelyje šių sistemų didelių duomenų analizės metodai nėra pritaikyti. Suprojektuota didelių duomenų analizės architektūra, kurią galima įdiegti į esamas sveikatos priežiūros sistemas [30]. Didelių duomenų analizė apima aprašomąją statistiką ir duomenų išgavimą (angl. *data mining*). Aprašomoji statistika apima duomenų sisteminimą ir grafinį atvaizdavimą. Ji leidžia koncentruotai užrašyti informaciją, esančią dideliuose duomenų masyvuose. Duomenų išgavimas – automatinė informacijos paieška duomenų bazėje (DB) pagal šablonus. Jo

tikslas naujos ir naudingos informacijos radimas. Didelės apimties duomenų saugojimui naudojamos DB valdymo sistemos (MongoDB, MarkLogic, Apache Cassandra), kurios pagrįstos NoSQL technologija. Šios DB pasižymi trumpu atsako laiku vykdant užklausas. Didelių duomenų analizės architektūra sudaryta iš penkių sluoksnių: duomenų, jų surinkimo, analizės, informacijos sisteminimo ir duomenų valdymo. Duomenų sluoksnis apima informacijos šaltinius: elektroninius sveikatos įrašus, jutikliais fiksuojamus sveikatos parametrus, medicininius vaizdus ir t. t. Duomenų surinkimo sluoksnyje vykdomas įvairaus formato, dydžio, dažnumo informacijos surinkimas iš komunikacijos kanalų. Duomenys saugomi Hadoop paskirstytoje failų sistemoje (angl. *Hadoop Distributed File System*, HDFS), NoSQL duomenų bazėje arba Hadoop debesų kompiuterijos platformoje. Analizės sluoksnyje vykdomas apdorojimas ir duomenų išgavimas. Informacijos sisteminimo sluoksnyje atliekamas duomenų vizualizavimas ir jų pateikimas vartotojo sąsajoje medicinos personalui. Duomenų valdymo sluoksnyje vykdomas saugumo ir privatumo užtikrinimas, leidimų suteikimas naudotis informacija.

1.8. Medicininių jutiklių integravimas į vieną bendrą sistemą

Per pastaruosius porą dešimtmečių sveikatos priežiūros technologijų pažanga ypač išaugo. Daug dėmesio skiriama paskirstytų sveikatos priežiūros sistemų kūrimui, kur prietaisai veikia ant skirtingų architektūrų, platformų ir operacinių sistemų. Medicininiai įrenginiai ir jutikliai gaminami įvairių gamintojų, todėl skiriasi jų architektūra, duomenų perdavimo sąsaja. Sveikatos priežiūros sistemose vienas iš iššūkių yra medicininių jutiklių integravimas į vieną bendrą sistemą. Sukurta sistema, kuri surenka duomenis apie paciento sveikatos būklę iš skirtingų gamintojų medicininių prietaisų ir jutiklių [31]. Fiksuojami duomenys RTPS (Real-Time Publish Subscribe) protokolu perduodami į serverį. Jame esanti informacija prieinama medicinos personalui. Sistema sukurta naudojant Java programavimo kalbą ir Netbeans platformą. Atliktas tyrimas, kurio metu duomenys buvo surenkami iš 10 skirtingų medicininių prietaisų ir vienu metu atvaizduojami skirtingam vartotojų skaičiui, kurie naudojami laidiniu arba bevieliu LAN ryšiu. Nustatytas tinklo pralaidumas ir atsako laikas esant skirtingam vartotojų skaičiui. Matavimai atlikti naudojant Wireshark programinę įrangą. Nustatyta, kad didėjant vartotojų skaičiui laidinio LAN ryšio vidutinis pralaidumas didėja, o atsako laikas išliko pastovus. Vidutinis pralaidumas bevielio LAN buvo mažesnis negu laidinio.

Gerėjant gyvenimo sąlygoms ilgėja žmonių gyvenimo trukmė. Apskaičiuota, kad per ateinančius 40 metų 60 ar daugiau metų amžiaus žmonių skaičius patrigubės. Taip pat teigiama, kad žmonių su negalia ir sergančių lėtinėmis ligomis, skaičius nuolat didės. Lėtinių ligų gydymas yra ilgas ir brangus procesas reikalaujantis nuolatinės priežiūros ir medicininių konsultacijų. Sukurta nuotolinė mobilioji paciento sveikatos būklės stebėjimo sistema – TeleCARE [32]. Jutiklių duomenų surinkimo programa įdiegiama į mobilųjį įrenginį turintį Android OS. Sukurta mobilioji programa lengvai suderinama su

naujais medicinos prietaisais. Nuotolinė sveikatos būklės stebėjimo sistema taikoma gydant pacientus su širdies ir kraujagyslių ligomis. Naudojant TeleCARE sistemą gydytojams suteikiama galimybė analizuoti sveikatos duomenis internete. Sistema sudaryta iš trijų pagrindinių komponentų: duomenų surinkimo posistemė – vykdomas jutiklių duomenų fiksavimas iš medicininių prietaisų, duomenų apdorojimo posistemė – atliekama duomenų analizė ir ataskaitų generavimas, internetinis puslapis – atvaizduojama informacija susijusi su pacientų sveikata. Pacientas naudojantis internetine sąsaja gali pateikti naujus sveikatos parametrų matavimus ir peržiūrėti savo ligos istorijos įrašus. Gydytojas naudojantis sistema gali skirti naują gydymo planą priklausomai nuo paciento sveikatos būklės, išrašyti naujus elektroninius receptus, sudaryti naują grafiką sveikatos parametrų matavimui. TeleCARE sistema suinteresuotus asmenis įspėja apie sveikatos parametrų nukrypimus nuo leistinų ribų. Vartotojai mobiliąją TeleCARE programą gali įsidiegti į savo įrenginį pasinaudodami Google Play paslauga. Programa susiejama su mediciniais prietaisais (pulsoksimetrais, išmaniaisiais laikrodžiais, apyrankėmis, krūtinės diržais), kurie Bluetooth ryšiu fiksuojamus sveikatos parametrus perduoda į mobilųjį telefoną. Užfiksuoti parametrai Wi-Fi ryšiu siunčiami į TeleCARE serverį. Mobilioji programa fiksuoja paciento aktyvumą (sėdi, eina, bėga, važiuoja dviračiu) ir jo buvimo vietą naudojant GPS.

1.9. Automatinės reguliavimo sistemos sveikatos priežiūroje

Automatizuotos treniruočių testavimo sistemos yra svarbios norint atlikti širdies ir kvėpavimo analizę fizinės apkrovos metu. Panaudojant neraiškųjį neuroninio tinklo (angl. *fuzzy neural network*) reguliatorių sukurta širdies ritmo reguliavimo sistema treniruočių ir mankštos metu [33]. Pagal fiksuojamą širdies ritmą automatiškai reguliuojamas bėgimo takelio greitis siekiant kontroliuoti sportuojančio asmens širdies ritmą nustatytose ribose. Asmens širdies ritmas fiksuojamas į diržą integruotu jutikliu. Diržas apjuosiamas aplink krūtinę. Apkrovos metu užfiksuotas širdies ritmo signalas Bluetooth ryšiu siunčiamas į dsPIC 30F4011 mikrovaldiklį. Iš jo signalas perduodamas į kompiuterį naudojant RS232 sąsają. Kompiuteryje įdiegta programa, kuri vykdo skaitmeninio signalo apdorojimą ir širdies ritmo reguliavimą panaudojant neraiškųjį neuroninį tinklą. Jis sudarytas iš keturių sluoksnių: įėjimo, narystės, taisyklių ir išėjimo. Pirmajame sluoksnyje apskaičiuojama klaida e tarp norimo ir esamo signalo bei klaidos pokytis Δe . Antrajame sluoksnyje klaidai e sudaromi PE, ZE ir NE neraiškieji rinkiniai (angl. *fuzzy sets*), klaidos pokyčiui Δe – PD, ZD ir ND, o valdymo signalui u – PU, MU, ZU ir SU. Rinkiniams sudaryti naudojamos Gauso narystės funkcijos (angl. *Gaussian membership functions*). Trečiajame sluoksnyje iš neraiškiųjų rinkinių apibrėžiamos devynios neraiškiosios taisyklės (angl. *fuzzy rules*). Ketvirtajame sluoksnyje apskaičiuojami du išėjimai u_S ir u_θ . Norint juos apskaičiuoti trečiajame sluoksnyje gautos reikšmės padauginamos iš svorių (angl. *weights*). Svoriai parenkami neuroninio tinklo apmokymo metu. Išėjimai u_S ir u_θ naudojami keisti bėgimo takelio

variklių greitį. Sistema sukurta naudojant Matlab programinę įrangą. Tyrimo metu palyginti du reguliatoriai: neraiškiojo neuroninio tinklo ir PI. Bėgimo takelio variklių greičiai buvo keičiami norint asmens širdies ritmą susireguliuoti ties 140 dūžių per minutę (angl. *beats per minute*, bpm). Naudojant neraiškųjį neuroninio tinklo reguliatorių širdies ritmą iki nustatytos ribos pavyko greičiau sureguliuoti negu tada, kai buvo naudotas PI reguliatorius. Su neraiškioju neuroniniu tinklu širdies ritmas susireguliuavo maždaug per 450 s, o su PI – per 600 s.

1.10. Analizės apibendrinimas

Išnagrinėtų sistemų apibendrinimas pateikiamas 1 lentelėje.

1 lentelė. Sveikatos priežiūros sistemos

Sistema pateikta literatūroje	Ligų diagnozavimas, stebėjimas	Fiksuojami sveikatos parametrai	Fiksavimui naudota techninė įranga	Užfiksuotų sveikatos parametrų		Duomenų saugumas
				Apdorojimas, saugojimas	Peržiūra	
[6]	Fizinio aktyvumo monitoringas	Sunaudotos kalorijos, širdies ritmas, kūno temperatūra ir padėtis	Veiklos monitoringo prietaisas, kuris uždedamas ant riešo arba juosmens	Apdorojimas vykdomas serveryje	Nuotoliniu būdu medicinos personalui	Neatsižvelgta
[7]	Širdies nepakankamumo stebėjimas už ligoninės ribų	Kraujo spaudimas, širdies susitraukimų dažnis	Kraujo spaudimo matuoklis, pulsoksimetras	Apdorojimas vykdomas serveryje	Nuotoliniu būdu sveikatos priežiūros specialistams	Neatsižvelgta
[8]	Pagyvenusių ir neįgaliųjų žmonių priežiūra	Širdies ritmas	Arduino Uno valdiklių plokštė surenkanti fiksuojamus pulso jutiklio parametrus	Apdorojimas vykdomas serveryje	Nuotoliniu būdu medicinos specialistų	Neatsižvelgta
[9]	Širdies ir kraujagyslių ligos	Širdies ritmas, kūno temperatūra	LM35 temperatūros jutiklis, LM358 širdies ritmo jutiklis. Fiksuojami parametrai surenkami Arduino valdiklių plokštė	Apdorojimas vykdomas tinklalapių (web) serveryje, o saugojimas –tame pačiame serveryje įdiegtoje duomenų bazėje	Nuotoliniu būdu internetinėje naršyklėje	Neatsižvelgta
[10]	Neurodegeneracinės ligos	Elektroencefalograma (EEG), EEG signalo kanalai, smegenų aktyvumo žemėlapis	Elektroencefalografas	Apdorojimas vykdomas duomenų analizės serveriuose, o saugojimas – atskirame serveryje įdiegtoje MySQL 5.5 duomenų bazėje	Nuotoliniu būdu mobiliajame įrenginyje neurologui	Duomenų perdavimo metu saugumui užtikrinti naudojamas TLS/SSL3 protokolas
[11]	Senyvų žmonių slauga namuose	Asmens emocijos, širdies ritmas	Vaizdo kameros įrengtos namuose, išmanusis laikrodis su integruotu jutikliu	Apdorojimas vykdomas sprendimų priėmimo modulyje	Nuotoliniu būdu slaugytojui	Neatsižvelgta
[12]	Kūno laikysenos ir griuvimo stebėjimas, širdies ir	Kūno padėtis, širdies ritmas, garso įrašas	Išmanusis telefonas, kuriame integruoti jutikliai: akselerometras, kamera, mikrofonas	Nevykdomas	Išmaniajame telefone vartotojui	Neatsižvelgta

	kraujagyslių ligos, kvėpavimo sistemos sutrikimai					
[13]	Senyvo amžiaus šeimos narių priežiūra	Širdies ritmas, kūno pozicija	Su integruotu jutikliu diržas, kuris uždedamas ant krūtinės, akcelerometras	Nevykdomas	Mobiliajame įrenginyje vartotojui	Neatsižvelgta
[14]	Reguliarus ir taisyklingas vaistų vartojimas	–	–	Mobiliajame įrenginyje reguliariai pranešama ir primenama kada pacientas turi vartoti vaistus		Neatsižvelgta
[15]	Širdies ir kraujagyslių ligos	Kraujo spaudimas, širdies ritmas	Su išmaniojo telefono kamera naudojant programas: Instant Heart Rate, Instant Blood Pressure	Nevykdomas	Išmaniajame telefone mobiliojoje programoje vartotojui	Neatsižvelgta
[16]	Vartotojo veiklos monitoringas	Paciento buvimo vieta, širdies ritmas, kūno padėtis	Išmanusis telefonas, kuriame integruoti jutikliai: GPS, akcelerometras ir išmanusis laikrodis su integruotu pulso jutikliu	Nevykdomas	Į išmanųjį telefoną slaugytojas gauna pranešimus ir įspėjimus apie stebimą asmenį	Neatsižvelgta
[17]	Sveikatos priežiūros paslaugos internetu	–	–	Apdorojimas vykdomas debesų kompiuterijos infrastruktūroje, o saugojimas – MySQL duomenų bazėje	Vartotojo konsultacija su gydytoju naudojant internetinėje naršyklėje pateikiamu tinklalapiu	Neatsižvelgta
[18]	Nuo didžiųjų miestų nutolusių gyventojų sveikatos monitoringas	Širdies ritmas, elektrokardiograma (EKG)	EKG jutiklis. Bluetooth ryšiu jutiklių duomenys perduodami į mobilųjį telefoną arba į planšetinį kompiuterį	Apdorojimas vykdomas debesų kompiuterijos infrastruktūroje	Kompiuteri-ryje įdiegtoje programoje sveikatos priežiūros specialistams	Vykdomas duomenų šifravimo jų perdavimo metu
[19]	Sveika mityba	–	Išmaniojo telefono kamera užfiksuotas vaizdas su maistu	Paskirstytas vaizdų apdorojimas ir atpažinimas vykdomas debesų kompiuterijos infrastruktūroje naudojant keturias virtualias mašinas	Išmaniajame telefone vartotojui pateikiamas kalorijų skaičius maiste	Neatsižvelgta
[20]	Lėtinės kvėpavimo ligos	Kraujo įsisotinimas deguonimi (SpO2), širdies ritmas	BM2000A OLED pulsoksimetras	Debesų kompiuterijos paslauga skirta duomenų saugojimui ir apdorojimui. Naudota NoSQL duomenų bazė	Internetinis puslapis ir mobilioji programa medicinos personalui ir globėjui	Neatsižvelgta
[21]	Parkinsono liga	Balso signalas	Mikrofonas išmaniajame telefone	Skaitmeninis signalo apdorojimas vykdomas debesų	Internetinė sąsaja gydytojui ir	Neatsižvelgta

				kompiuterijos infrastruktūroje, o rezultatai saugomi duomenų bazėje	pacientui	
[22]	Širdies ir kraujagyslių ligos	Su širdies ligomis susiję sveikatos parametrai	Medicininis jutiklių tinklas	Apdorojimas vykdomas debesų kompiuterijos infrastruktūroje. Duomenų bazėje saugomi šifruoti duomenys	Internetinis puslapis sveikatos priežiūros specialistams ir pacientams	Naudotas atributais paremtas duomenų šifravimas
[23]	Lėtinės ligos	Elektrokardiograma (EKG)	Medicininis jutiklių tinklas	Apdorojimas vykdomas debesų kompiuterijos infrastruktūroje	Vartotojo sąsaja, kuri prieinama interneto naršyklėje gydytojams	Įgyvendinta sistemos vartotojo autentifikacija ir autorizacija
[24]	Pagyvenusių žmonių priežiūra	Elektrokardiograma (EKG)	EKG fiksavimo prietaisas	EKG signalo apdorojimas vykdomas debesų kompiuterijos infrastruktūroje	Kompiuterijoje įdiegtoje programoje medicinos personalui	Į EKG signalą įterpiamas skaitmeninis vandens ženklas (kliento registracijos ID)
[25]	Lėtinė obstrukcinė plaučių liga	Su sveikata susiję parametrai generuojami keičiant fiksavimo dažnį		Apdorojimas vykdomas naudojant programinį įrankį: Distributed Computing Framework	Nepateikta	Parinkti M2M protokoliai el. sveikatos paslaugoms, kurie atitinka el. sveikatos saugumo, privatumo reikalavimus
[26]	Visuotinė sveikatos priežiūra	Sveikatos priežiūros paslaugoms pateiktas teorinis informacinių sistemų kelių pakopų karkasas			Nepateikta	Neatsižvelgta
[27]	Visuotinė sveikatos priežiūra	Širdies ritmas, elektrokardiograma (EKG)	Zephyr HxM krūtinės diržas	Apdorojimas vykdomas serveryje	Išmaniajame laikrodyje vartotojui pateikiami pranešimai gauti iš gydymo įstaigos	Neatsižvelgta
[28]	Sveikatos priežiūra už ligoninės ribų	Generuojami trijų tipų paketai: EM duomenų paketai (EKG, EEG), DS duomenų paketai (EMG, GSR) ir GM duomenų paketai (SpO2, kūno temperatūra)	Bevielių jutiklių tinklai ant tiriamojo kūno paviršiaus (WBAN)	WBAN simuliacija atlikta naudojant OMNeT++ tinklų simuliacijos įrankį	Nepateikta	Neatsižvelgta
[29]	Lėtinės ligos, pooperacinė priežiūra,	Su sveikata susiję parametrai generuojami	Bevielių jutiklių tinklai ant tiriamojo kūno paviršiaus (WBAN)	Debesų kompiuterijos sluoksnyje vykdomas duomenų saugojimas, apdorojimas,	Duomenų peržiūra prieinama taikomajame	Debesyse vykdomas, saugumo ir privatumo

	telemedicina			valdymas ir apkrovų paskirstymas. Architektūros sudarymui panaudotas Opnet tinklų simuliacijos įrankis	sluoksnyje galutiniams vartotojams	užtikrinimas, vartotojo autentifikacija ir autorizacija
[30]	Sveikatos parametrų stebėjimas	Su sveikata susiję parametrai generuojami	–	Apdorojimas vykdomas panaudojant didelių duomenų analizės metodus. Saugojimui naudota duomenų bazė, kuri pagrįsta NoSQL technologija	Duomenų vizualizavimas ir jų pateikimas vartotojo sąsajoje sveikatos priežiūros specialistams	Duomenų valdymo sluoksnyje vykdomas saugumo ir privatumo užtikrinimas
[31]	Širdies ir kraujagyslių ligos	Širdies ritmas, elektrokardiograma (EKG)	Skirtingų gamintojų medicininiai prietaisai	Apdorojimas vykdomas serveryje	Kompiuteryje įdiegtoje programoje medicinos personalui	Neatsižvelgta
[32]	Širdies ir kraujagyslių ligos	Širdies ritmas, elektrokardiograma (EKG)	Medicininiai prietaisai (pulsoksimetrai, išmanieji laikrodžiai, apyrankės, krūtinės diržai)	Apdorojimas vykdomas serveryje. Sistema suinteresuotus asmenis įspėja apie sveikatos parametrų nukrypimus nuo leistinų ribų	Internetinė sąsaja TeleCARE gydytojams ir pacientams	Neatsižvelgta
[33]	Širdies ritmo reguliavimas treniruočių ir mankštos metu	Širdies ritmas	dsPIC 30F4011 mikrovaldiklis surenkantis fiksuojamus parametrus iš jutiklio, kuris integruotas į diržą. Diržas apjuosiamas aplink krūtinę	Kompiuteryje įdiegta programa, kuri vykdo skaitmeninio signalo apdorojimą ir širdies ritmo reguliavimą panaudojant neraiškųjį neuroninį tinklą	Programoje, kurios kūrimui naudotas Matlab programinis paketas	Neatsižvelgta

Išnagrinėtose sistemose pastebėti kai kurie trūkumai. Pateikiamas trūkumų sąrašas:

- nepateiktas sistemos prototipas, atlikta tik jau egzistuojančių sistemų apžvalga;
- sistemos prototipas neįgyvendintas, pateiktas tik jo teorinis modelis;
- neatsižvelgta į duomenų saugumą;
- duomenys nesaugomi duomenų bazėje;
- nesukurta vartotojo sąsaja medicinos personalui;
- nenaudojama debesų kompiuterijos infrastruktūra;
- nemobili sistema;
- nefiksuojami paciento sveikatos parametrai;
- nėra grįžtamojo ryšio pacientui.

Atsižvelgiant į išnagrinėtų sistemų trūkumus šiame darbe buvo sukurtas sveikatos stebėjimo sistemos su grįžtamoju ryšiu prototipas.

II. PROJEKTAVIMAS

Šiame darbe kuriama nuotolinė sveikatos stebėjimo sistema su grįžtamoju ryšiu, kuri sudaryta iš keturių pagrindinių komponentų: parametrų fiksavimas jutikliais, mobilioji programa fiksuojamų parametrų siuntimui į serverį, duomenų apdorojimas debesų serveryje, internetinis puslapis fiksuojamų parametrų atvaizdavimui. Sistemos komponentai nagrinėjami atskirai. Jiems suformuluojami funkciniai ir nefunkciniai reikalavimai, parenkama įranga, atliekami projektavimo ir programavimo darbai, testavimas ir diegimas.

2.1. Reikalavimų specifikacija

Atliktas funkcinų ir nefunkcinų reikalavimų specifikavimas. Funkciniais reikalavimais nusakomas sistemos funkcionalumas, nurodoma ką sistema turi daryti, kokiais duomenimis manipuluoti, kad atlikti numatytas funkcijas. Sistemai suformuluoti tokie funkciniai reikalavimai:

1. sistema jutikliu turi fiksuoti paciento širdies ritmą ir laiko intervalą tarp dūžių (RR intervalas);
2. sistema jutiklinius duomenis iš mikrovaldiklio turi perduoti į mobiliąją programą;
3. pacientui naudojant mobiliąją programą turi būti leidžiama prisijungti prie sistemos nurodant vartotojo vardą ir slaptažodį;
4. pacientui mobiliojoje programoje turi būti leidžiama pasikeisti paskyros nustatymus;
5. sistema mobiliojoje programoje turi leisti atlikti Bluetooth įrenginių paiešką;
6. sistema turi leisti susieti jutiklį su mobiliąja programa;
7. sistema pacientui mobiliojoje programoje turi atvaizduoti fiksuojamus parametrus;
8. sistema mobiliojoje programoje turi atvaizduoti informaciją apie jutiklį, kuris buvo pasirinktas parametrų fiksavimui;
9. sistema turi suteikti galimybę esant ekstriniam atvejui nustatyti paciento buvimo vietos koordinates (platumą, ilgumą);
10. sistema turi tikrinti ar paciento širdies ritmas neperžengia leistinos ribos;
11. sistema turi informuoti suinteresuotus asmenis SMS žinute, elektroniniu laišku jeigu širdies ritmas peržengia leistiną ribą;
12. sistema užfiksuotus parametrus iš mobiliosios programos turi perduoti į debesų serverį;
13. sistema užfiksuotus parametrus turi išsaugoti duomenų bazėje, kuri įdiegta debesų serveryje;
14. sistema turi atlikti duomenų bazės atsarginių kopijų darymą;
15. sistema atsargines duomenų bazės kopijas turi išsaugoti Dropbox serveryje;
16. sistema turi sudaryti fiksavimo suvestinę, kurioje būtų apskaičiuotas minimalus, vidutinis, maksimalus širdies ritmas ir RR intervalas;
17. sistema turi sugeneruoti PDF dokumentą, kuriame būtų pateikiama fiksavimo suvestinė;

18. sistema PDF dokumentą su fiksavimo suvestine turi išsiųsti į paciento el. pašta;
19. medicinos personalui naudojant internetinį puslapį turi būti leidžiama prisijungti prie sistemos nurodant vartotojo vardą ir slaptažodį;
20. medicinos personalui internetiniame puslapyje turi būti leidžiama pasikeisti paskyros nustatymus;
21. sistema medicinos personalui internetiniame puslapyje turi atvaizduoti paciento sveikatos parametrus užfiksuotus tam tikru laiko momentu;
22. sistema medicinos personalui internetiniame puslapyje turi atvaizduoti fiksavimo suvestinę;
23. sistema medicinos personalui turi suteikti galimybę kurti, redaguoti, trinti medicinines išvadas;
24. sistema medicinines išvadas turi pateikti pacientui mobiliojoje programoje;
25. sistemoje turi būti įdiegta galimybė administratoriui registruoti pacientus ir medicinos specialistus pasinaudojant internetiniu puslapiu;
26. sistema registracijos metu pagal nurodytą vardą ir pavardę turi sugeneruoti vartotojo vardą, laikiną slaptažodį;
27. sistema registracijos metu sugeneruotus prisijungimo duomenis turi nusiųsti į vartotojo el. pašta.

Taip pat sistemai apibrėžiami nefunkciniai reikalavimai, kurie vienaip ar kitaip riboja sistemos įgyvendinimo būdą ir nusako nefunkcines sistemos savybes (patikimumą, panaudojamumą ir kt.).

Sistemai suformuluoti tokie nefunkciniai reikalavimai:

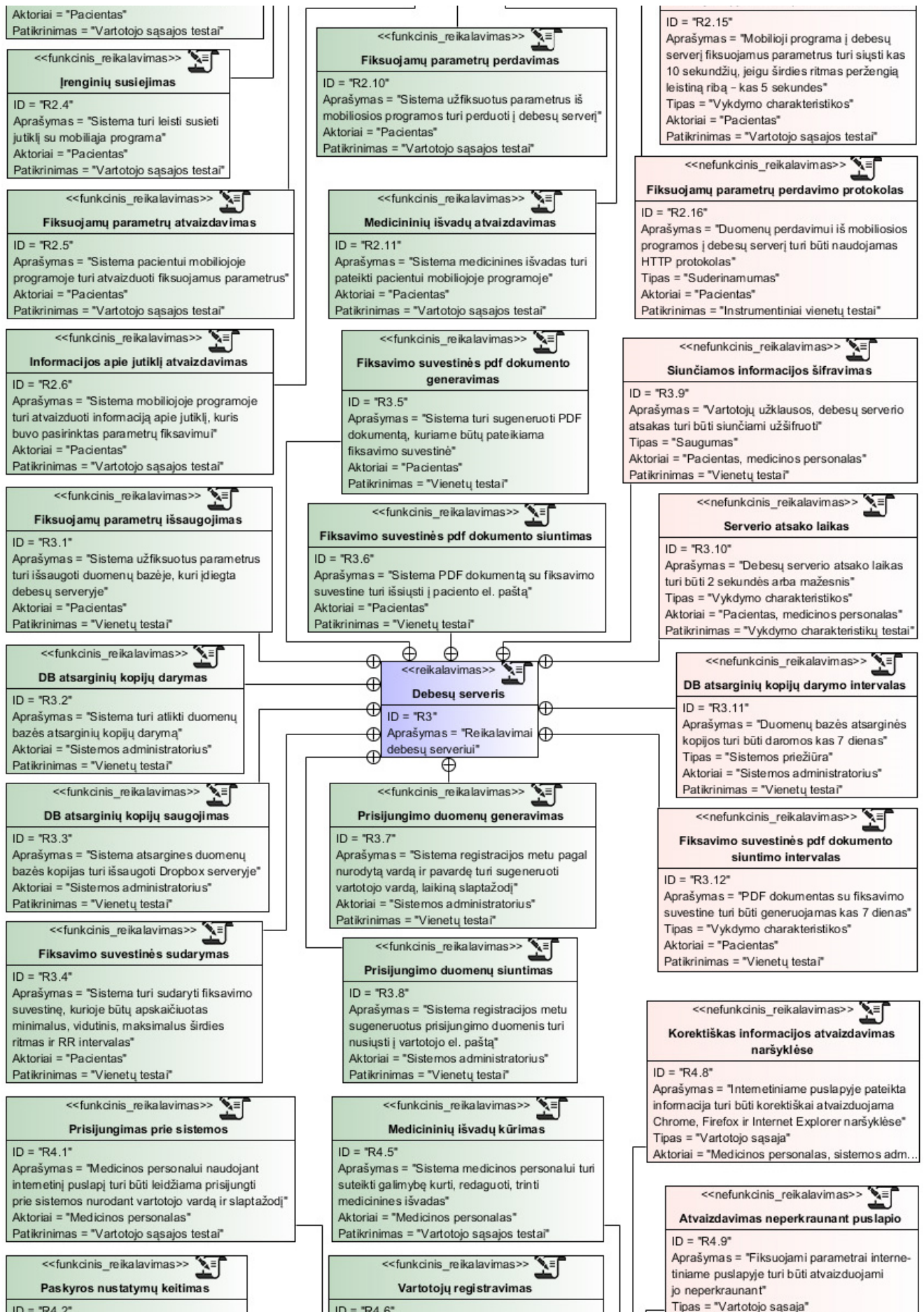
1. duomenų objektai iš mikrovaldiklio į mobiliąją programą turi būti perduodami JSON formatu;
2. RR intervalas į mobiliąją programą turi būti perduodamas milisekundžių tikslumu;
3. jutikliniai duomenys iš mikrovaldiklio į mobiliąją programą turi būti perduodami 115200 bit/s arba didesniu greičiu;
4. jutikliniai duomenys iš mikrovaldiklio į mobiliąją programą turi būti siunčiami Bluetooth ryšiu;
5. fiksuojami parametrai mobiliojoje programoje turi būti atvaizduojami skaitine ir grafine forma;
6. buvimo vietos koordinatės turi būti fiksuojamos kas 10 sekundžių;
7. fiksuojami parametrai iš mobiliosios programos į debesų serverį turi būti siunčiami Wi-Fi arba mobiliuoju interneto ryšiu;
8. mobilioji programa į debesų serverį fiksuojamus parametrus turi siųsti kas 10 sekundžių, jeigu širdies ritmas peržengia leistiną ribą – kas 5 sekundes;
9. duomenų perdavimui iš mobiliosios programos į debesų serverį turi būti naudojamas HTTP protokolas;
10. vartotojų užklausos, debesų serverio atsakas turi būti siunčiami užšifruoti;
11. debesų serverio atsako laikas turi būti 7 sekundės arba mažesnis;
12. duomenų bazės atsarginės kopijos turi būti daromos kas 7 dienas;
13. PDF dokumentas su fiksavimo suvestine turi būti generuojamas kas 7 dienas;

14. internetinio puslapio CSS ir HTML aprašai turi atitikti W3C standartus;
15. internetiniame puslapyje pateikta informacija turi būti korektiškai atvaizduojama Chrome, Firefox ir Internet Explorer naršyklėse;
16. fiksuojami parametrai internetiniame puslapyje turi būti atvaizduojami jo neperkraunant;
17. fiksuojami parametrai internetiniame puslapyje turi būti atnaujinami kas 5 sekundes;
18. paciento vietovės parametrai turi būti atvaizduojami naudojant Google Maps.

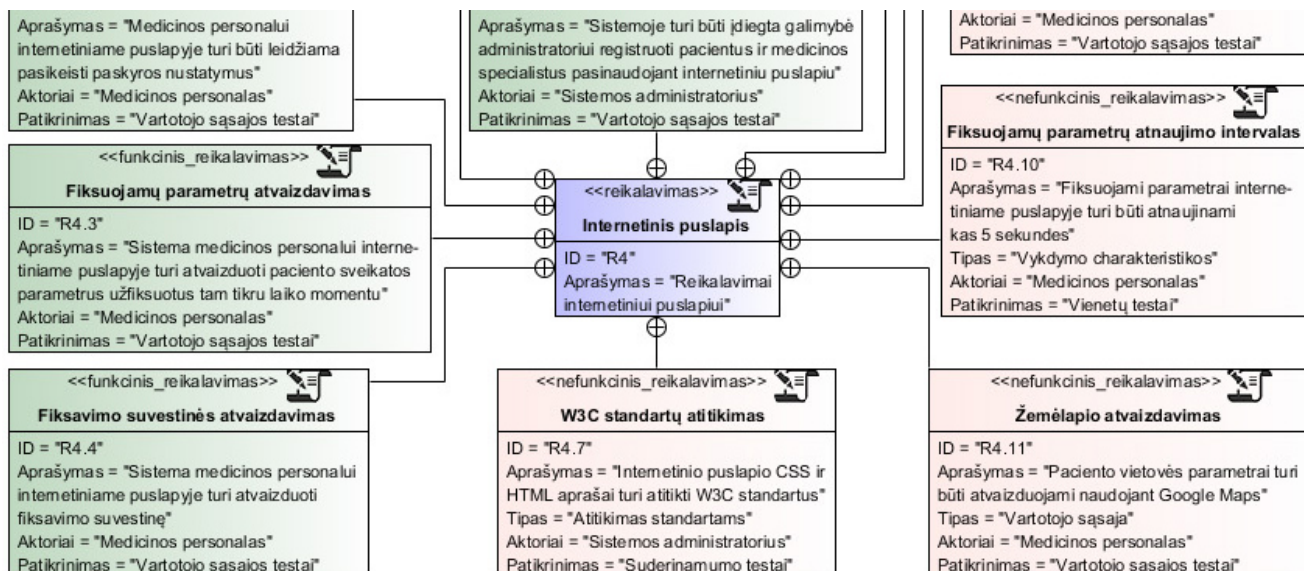
Sudaryta SysML reikalavimų diagrama (angl. *requirement diagram*), kuri pateikiama 2 pav., 3 pav. ir 4 pav. Reikalavimai priskirti keturiems pagrindiniams sistemos komponentams. Pateikti 27 funkciniai reikalavimai ir 18 nefunkcinių reikalavimų. Reikalavimams priskirti unikalūs ID, pateiktas pavadinimas ir aprašymas. Nurodytas aktorius, kuris susietas su reikalavimu. Pateiktas reikalavimo įgyvendinimo patikrinimo metodas. Nefunkciniams reikalavimams nurodytas tipas.



2 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (1)



3 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (2)



4 pav. Sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai (3)

2.2. Įrangos parinkimas

Sistemos keturių pagrindinių komponentų įgyvendinimui parinkta reikiama programinė ir techninė įranga. 5 pav. pateikiama SysML struktūros aprašų diagrama (angl. *block definition diagram*), kurioje pavaizduoti pasirinkti programiniai įrankiai ir jų versijos projektavimo ir programavimo darbams. UML ir SysML diagramų modeliavimui pasirinkta Visual Paradigm programinė įranga, elektroninių schemų braižymui – Microsoft Visio, duomenų bazės struktūros projektavimui – MySQL Workbench. Programos, kuri įdiegiama į mikrovaldiklį, kūrimui pasirinkta Visual Studio integruota kūrimo aplinka (angl. *integrated development environment, IDE*), Android Studio – mobiliajai programai, PhpStorm – programai įdiegtai į debesų serverį ir internetiniam puslapiui. Bibliotekų valdymui (angl. *library manager, dependency management*), galutinio programinio kodo generavimui (angl. *code builder, build system*) pasirinkti PlatformIO, Gradle, Composer, Bower ir Gulp komandinės eilutės įrankiai. Programinio kodo versijavimui pasirinkta versijų kontrolės sistema Git. Debesų serverio administravimui naudotas PuTTY įrankis. Fiksuojami paciento parametrai saugomi MySQL duomenų bazėje. Vartotojų užklausoms apdoroti ir serverio atsako grąžinimui pasirinktas tinklalapių serveris (angl. *web server*) – Apache HTTP Server. Nuotolinės sveikatos stebėjimo sistemos kūrimui naudotos C++, Java, PHP ir JavaScript programavimo kalbos. Vartotojo sąsajos kūrimui panaudotos XML, HTML ir CSS kalbos.

Pasirinkti programiniai įrankiai projektavimo darbams	
įrankiai	
Modeliavimo kalba : UML, SysML	
Diagramų modeliavimo įrankis : Visual Paradigm = 14	
Duomenų bazės struktūros projektavimas : MySQL Workbench = 6.3.6	

Pasirinkti programiniai įrankiai programavimo darbams																																		
<table border="1"> <thead> <tr> <th>Programa parametrų fiksavimui iš jutiklių</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">properties</td> </tr> <tr> <td>Programavimo kalba : C++</td> </tr> <tr> <td>Integruota kūrimo aplinka : Visual Studio = 2015</td> </tr> <tr> <td>Bibliotekų valdymas : PlatformIO = 3.2.1</td> </tr> <tr> <td>Galutinio programos kodo generavimas : PlatformIO = 3.2.1</td> </tr> <tr> <td>Versijų kontrolės sistema : Git = 2.8.3</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Programa debesų serveryje</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">įrankiai</td> </tr> <tr> <td>Programavimo kalba : PHP = 7.1.1</td> </tr> <tr> <td>Duomenų bazė : MySQL = 5.7.17</td> </tr> <tr> <td>Tinklalapių serveris : Apache HTTP Server = 2.4.25</td> </tr> <tr> <td>Integruota kūrimo aplinka : PhpStorm = 2016.3.2</td> </tr> <tr> <td>Bibliotekų valdymas : Composer = 1.3.2</td> </tr> <tr> <td>Versijų kontrolės sistema : Git = 2.8.3</td> </tr> <tr> <td>Serverio administravimas : PuTTY = 0.67</td> </tr> </tbody> </table>	Programa parametrų fiksavimui iš jutiklių	properties	Programavimo kalba : C++	Integruota kūrimo aplinka : Visual Studio = 2015	Bibliotekų valdymas : PlatformIO = 3.2.1	Galutinio programos kodo generavimas : PlatformIO = 3.2.1	Versijų kontrolės sistema : Git = 2.8.3	Programa debesų serveryje	įrankiai	Programavimo kalba : PHP = 7.1.1	Duomenų bazė : MySQL = 5.7.17	Tinklalapių serveris : Apache HTTP Server = 2.4.25	Integruota kūrimo aplinka : PhpStorm = 2016.3.2	Bibliotekų valdymas : Composer = 1.3.2	Versijų kontrolės sistema : Git = 2.8.3	Serverio administravimas : PuTTY = 0.67	<table border="1"> <thead> <tr> <th>Mobilioji programa</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">įrankiai</td> </tr> <tr> <td>Programavimo kalba : Java</td> </tr> <tr> <td>Žymėjimo kalba : XML</td> </tr> <tr> <td>Integruota kūrimo aplinka : Android Studio = 2.2.3</td> </tr> <tr> <td>Bibliotekų valdymas : Gradle = 2.14.1</td> </tr> <tr> <td>Galutinio programos kodo generavimas : Gradle = 2.14.1</td> </tr> <tr> <td>Versijų kontrolės sistema : Git = 2.8.3</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Internetinis puslapis</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">įrankiai</td> </tr> <tr> <td>Programavimo kalba : JavaScript</td> </tr> <tr> <td>Žymėjimo kalba : HTML = 5</td> </tr> <tr> <td>Stilių kalba : CSS = 3</td> </tr> <tr> <td>Integruota kūrimo aplinka : PhpStorm = 2016.3.2</td> </tr> <tr> <td>Bibliotekų valdymas : Bower = 1.8</td> </tr> <tr> <td>Galutinio programinio kodo generavimas : Gulp = 1.2.2</td> </tr> <tr> <td>Versijų kontrolės sistema : Git = 2.8.3</td> </tr> </tbody> </table>	Mobilioji programa	įrankiai	Programavimo kalba : Java	Žymėjimo kalba : XML	Integruota kūrimo aplinka : Android Studio = 2.2.3	Bibliotekų valdymas : Gradle = 2.14.1	Galutinio programos kodo generavimas : Gradle = 2.14.1	Versijų kontrolės sistema : Git = 2.8.3	Internetinis puslapis	įrankiai	Programavimo kalba : JavaScript	Žymėjimo kalba : HTML = 5	Stilių kalba : CSS = 3	Integruota kūrimo aplinka : PhpStorm = 2016.3.2	Bibliotekų valdymas : Bower = 1.8	Galutinio programinio kodo generavimas : Gulp = 1.2.2	Versijų kontrolės sistema : Git = 2.8.3
Programa parametrų fiksavimui iš jutiklių																																		
properties																																		
Programavimo kalba : C++																																		
Integruota kūrimo aplinka : Visual Studio = 2015																																		
Bibliotekų valdymas : PlatformIO = 3.2.1																																		
Galutinio programos kodo generavimas : PlatformIO = 3.2.1																																		
Versijų kontrolės sistema : Git = 2.8.3																																		
Programa debesų serveryje																																		
įrankiai																																		
Programavimo kalba : PHP = 7.1.1																																		
Duomenų bazė : MySQL = 5.7.17																																		
Tinklalapių serveris : Apache HTTP Server = 2.4.25																																		
Integruota kūrimo aplinka : PhpStorm = 2016.3.2																																		
Bibliotekų valdymas : Composer = 1.3.2																																		
Versijų kontrolės sistema : Git = 2.8.3																																		
Serverio administravimas : PuTTY = 0.67																																		
Mobilioji programa																																		
įrankiai																																		
Programavimo kalba : Java																																		
Žymėjimo kalba : XML																																		
Integruota kūrimo aplinka : Android Studio = 2.2.3																																		
Bibliotekų valdymas : Gradle = 2.14.1																																		
Galutinio programos kodo generavimas : Gradle = 2.14.1																																		
Versijų kontrolės sistema : Git = 2.8.3																																		
Internetinis puslapis																																		
įrankiai																																		
Programavimo kalba : JavaScript																																		
Žymėjimo kalba : HTML = 5																																		
Stilių kalba : CSS = 3																																		
Integruota kūrimo aplinka : PhpStorm = 2016.3.2																																		
Bibliotekų valdymas : Bower = 1.8																																		
Galutinio programinio kodo generavimas : Gulp = 1.2.2																																		
Versijų kontrolės sistema : Git = 2.8.3																																		

5 pav. Sistemos įgyvendinimui pasirinkta programinė įranga

Integruotų kūrimo aplinkų (IDE) pasirinkimas atliktas įvertinus jų privalumus ir trūkumus. Išanalizuotos jose esančios funkcijos, kurios pateikiamos 2 lentelėje. IDE privalo palaikyti derinimą ir Git versijų kontrolės sistemą. IDE turi leisti vykdyti vienetų testus ir peržiūrėti kodo aprėptį testais. IDE turi palaikyti funkcijas: nuosekliosios sąsajos terminalą, programos vykdymo charakteristikų stebėjimą, REST API testavimą, programinio kodo pakeitimų peržiūrą neperkraunant naršyklės. Šios funkcijos reikalingos kuriant sistemos pagrindinius komponentus.

2 lentelė. Integruotų kūrimo aplinkų funkcijos

Programa parametrų fiksavimui iš jutiklių			
Programinė įranga	Arduino IDE 1.8.2	Visual Studio 2015 + Visual Micro įskiepis	Programino 1.5
Funkcija			
Kodo sintaksės išryškinimas	✓	✓	✓
Automatinis kodo užbaigimas	–	✓	✓
Klaidų aptikimas rašant kodą (kai dar nesukompiliuota)	–	✓	–
Langas su projekte naudojamais failais (angl. <i>Project/Solution explorer</i>)	–	✓	✓
Nuosekliosios sąsajos terminalas	✓	✓	✓
Git versijų kontrolės sistemos integracija	–	✓	–
Derinimas (angl. <i>Debug</i>)	–	✓	–
Mobilioji programa			
Programinė įranga	Android Studio 2.2.3	Eclipse for Android Developers 4.6	Xamarin Studio 6.0
Funkcija			
Android emulatoriaus palaikymas	✓	✓	✓
Programinio kodo pakeitimų įkėlimas į įrenginyje veikiančią programą neperkraunant visos programos ir jos neinstaliuojant iš naujo (angl. <i>Instant run</i>)	✓	–	–
Kodo aprėpties (angl. <i>Code coverage</i>) vienetų testais vizualizavimas	✓	✓	–
Programos APK analizavimas	✓	–	–
Programos vykdymo charakteristikų stebėjimas	✓	–	✓

(angl. <i>Performance monitor</i>)				
Git versijų kontrolės sistemos integracija		✓	✓	✓
Derinimas (angl. <i>Debug</i>)		✓	✓	✓
Programa debesų serveryje, internetinis puslapis				
	Programinė įranga	NetBeans IDE PHP 8.2	Adobe Dreamweaver CC 2017	PhpStorm 2016.3.2
Funkcija				
SSH kliento palaikymas		✓	–	✓
Failų įkėlimas į nuotolinį serverį		✓	✓	✓
Programinio kodo pakeitimų peržiūra neperkraunant naršyklės (angl. <i>Live view</i>)		✓	✓	✓
REST API testavimo palaikymas		–	–	✓
Prisijungimas prie duomenų bazės ir operacijų vykdymas		✓	–	✓
Xdebug Profiler rezultatų vizualizavimas		–	–	✓
Kodo kokybės analizavimas		–	–	✓
Git versijų kontrolės sistemos integracija		✓	✓	✓
Derinimas (angl. <i>Debug</i>)		✓	–	✓

Fiksuojamų duomenų surinkimui iš jutiklių reikalinga valdiklių plokštė su integruotu mikrovaldikliu. Renkantis ją atsižvelgiama į energijos sąnaudas užtikrinant sistemos mobilumą, į UART nuosekliosios sąsajos greitį. Naudojant šią sąsają duomenys perduodami į Bluetooth modulį, o iš jo į mobiliąją programą. Siekiama, kad mobiliojoje programoje fiksuojami parametrai būtų nuolat atnaujinami. Mikrovaldiklis turi turėti pakankamai Flash atminties parašytos programos saugojimui ir SRAM atminties – kintamiesiems. 3 lentelėje pateikti išnagrinėtų valdiklių plokščių techniniai parametrai.

3 lentelė. Valdiklių plokščių techniniai parametrai

Valdiklių plokštė	M168 AVR Development Board	TI MSP-EXP430G2 LaunchPad	Arduino Uno R3
Plokštėje įmontuotas mikrovaldiklis	Atmega168	MSP430G2553	ATmega328P
Taktinis dažnis	20 MHz	16 MHz	16 MHz
Darbinė įtampa	5 V	3,6 V	5 V
Srovės naudojimas esant aktyviam režimui	0,25 mA	0,23 mA	0,2 mA
Skaitmeninių įėjimų/išėjimų (angl. I/O) skaičius	14	14	14 (iš kurių 6 galimi būti panaudoti kaip PWM kanalai)
PWM kanalų	6	8	6
Analoginių įėjimų skaičius	6	8	6
ADC rezoliucija	10-bit	10-bit	10-bit
UART sąsajų skaičius	1	1	1
UART sąsajos greitis	iki 57600 bit/s	iki 9600 bit/s	iki 115200 bit/s
Flash atmintis	16 Kb	16 Kb	32 Kb
EEPROM atmintis	512 b	256 b	1024 b
SRAM atmintis	1 Kb	512 b	2 Kb

Nuotolinei sveikatos stebėjimo sistemai įgyvendinti, Arduino Uno R3 valdiklių plokštė yra tinkamiausia, nes ji atitinka visus prieš tai pateiktus reikalavimus.

Paciento širdies ritmo ir RR intervalo fiksavimui pasirinkti: SEN-11574 pulso jutiklis, e-Health platforma su EKG jutikliu, Zephyr HxM BT juosmens diržas ir Xiaomi Mi Band 1s apyrankė. Fiksuojamiems parametrams perduoti į mobiliąją programą pasirinktas HC-06 Bluetooth modulis. Jam

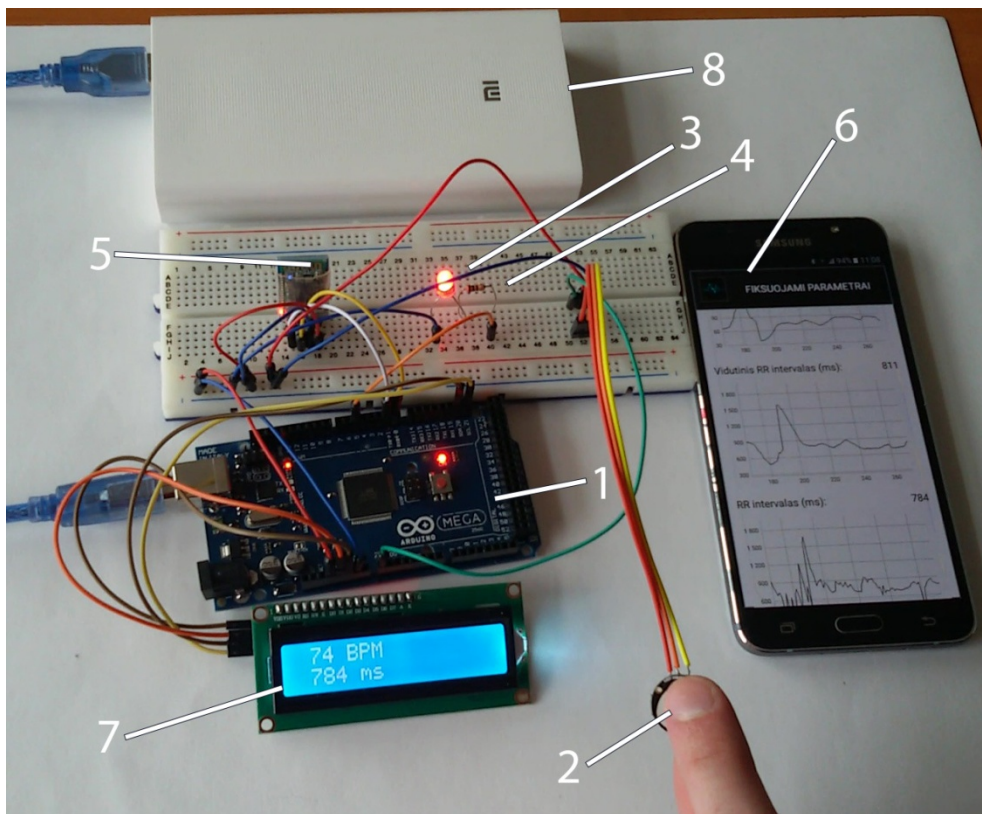
reikalingas 3.3 V maitinimo šaltinis. Duomenys iš mikrovaldiklio į modulį ir atvirkščiai perduodami per TX ir RX išvadus. Širdies ritmo ir RR intervalo atvaizdavimui pasirinktas dviejų eilučių po 16 simbolių (16x2) LCD ekranas HD44780. Prie LCD ekrano įmontuota I2C (TWI) nuoseklios sąsajos plokštė PCF8574T. Komunikacija tarp mikrovaldiklio ir LCD ekrano vykdoma naudojant SDA ir SDL išvadus. LCD ekranui reikalingas 5 V maitinimo šaltinis.

2.3. Sveikatos stebėjimo sistemos prototipo kūrimas

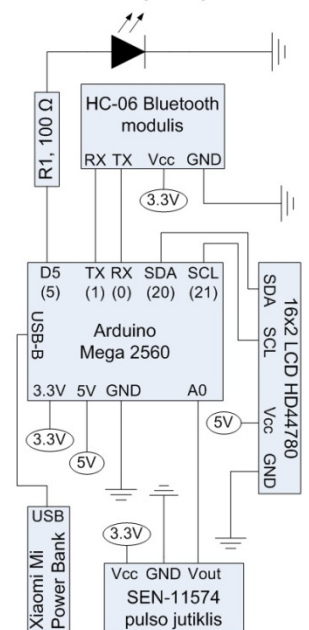
2.3.1. Parametrų fiksavimas jutikliais

Pateiktoje sistemoje asmens širdies ritmas ir laiko intervalas tarp dūžių (RR intervalas) fiksuojamas naudojant keturis jutiklius. Suprojektuota techninė įranga naudojant pulso jutiklį pateikiama 6 pav. 1 numeriu pažymėta Arduino Mega 2560 valdiklių plokštė, kuri fiksuoja EKG signalą iš SEN-11574 pulso jutiklio (2). Pagal fiksuojamą signalą yra nustatomas asmens širdies ritmas ir RR intervalas. Kai užfiksuojamas širdies dūžis užsidega šviesos diodas (3). Kad per šviesos diodą netekėtų per didelė srovė prijungiamas rezistorius (4). 5 numeriu pažymėtas HC-06 Bluetooth modulis. Juo naudojantis jutiklių duomenys perduodami į mobilųjį telefoną (6), kuriame įdiegta sukurta mobilioji programa. Širdies ritmas ir laiko intervalas tarp dūžių pateikiamas HD44780 LCD ekrane (7). Prie Arduino plokštės USB-B tipo jungties prijungtas Xiaomi 20000 mAh Mi Power Bank nešiojamas įkroviklis (8).

Elektroninė schema pateikiama 6 pav. dešinėje pusėje. Pulso jutiklio išėjimo signalas (V_{out}) prijungiamas prie Arduino plokštės analoginio įėjimo (A0). Jutiklis yra prijungiamas prie 3.3 V maitinimo šaltinio. Prie Arduino plokštės skaitmeninio išėjimo (D5) prijungiamas 100 Ω rezistorius, o prie jo šviesos diodas (LED). Su PWM signalu reguliuojamas LED šviesos intensyvumas. Užfiksavus širdies dūžį uždegamas LED ir palaipsniui mažinant išėjimo signalo (D5) įtampą, mažinamas LED intensyvumas. Maksimali šviesos diodo tiesioginė (angl. *forward*) srovė 30 mA, o tiesioginė įtampa 2 V. Arduino plokštės išėjimo (D5) signalas kinta 0 iki 5 V. Apskaičiuojama rezistoriaus varža: $\frac{5V-2V}{0,030A} = 100 \Omega$. Komunikacija tarp Arduino plokštės ir Bluetooth modulio vykdoma naudojant UART nuosekliosios sąsajos RX ir TX išvadus. Modulio TX išvadas prijungiamas prie Arduino plokštės RX (D0) išvado, o RX prie TX (D1). Moduliu reikalingas 3.3 V maitinimo šaltinis. LCD ekranas prijungiamas prie 5 V maitinimo šaltinio. Iš Arduino plokštės į LCD ekraną siunčiamos komandos. Komunikuojama per I²C (TWI) sąsajos išvadus SDA ir SCL.

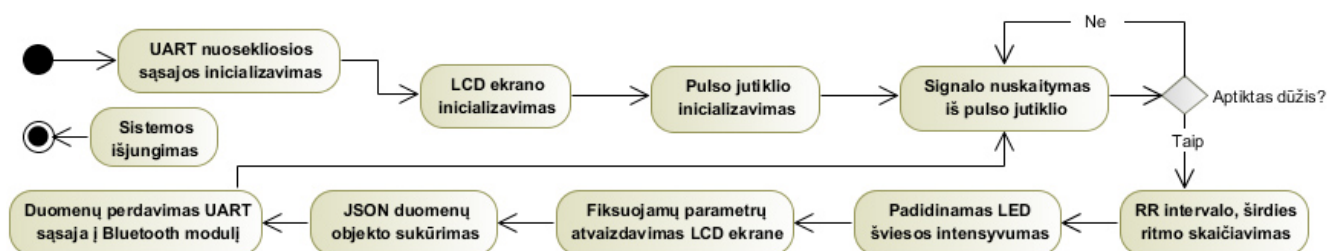


1. Arduino Mega 2560
2. SEN-11574 pulso jutiklis
3. Šviesos diodas
4. 100 Ω rezistorius
5. HC-06 Bluetooth modulis
6. Mobilioji programa telefone
7. HD44780 LCD ekranas
8. Xiaomi nešiojamas įkroviklis



6 pav. Suprojektuota techninė įranga naudojant SEN-11574 pulso jutiklį

UML veiklos diagrama (angl. *activity diagram*) pateikiama 7 pav. Joje pavaizduotas širdies ritmo fiksavimo algoritmas pulso jutikliu. Arduino plokštėje veikiančios programos vykdymas pradedamas nuo UART sąsajos inicializacijos. Atidaromas prievadas, nurodoma perdavimo sparta ir kiti parametrai. Įvykdoma LCD ekrano inicializacija. Nurodomas LCD ekrano I²C (TWI) sąsajos adresas, įjungiamas foninis apšvietimas (angl. *backlight*), ekrane išvedami matavimo vienetai. Inicializuojami pulso jutiklio signalo fiksavimui reikalingi parametrai. Nustatoma, kad LED intensyvumą reguliuojantis skaitmeninis išvadas D5 dirbtų išėjimo režimu. Arduino plokštė kas 2 ms nuskaityo pulso jutiklio signalą iš analoginio įėjimo A0. Signalas kinta intervale nuo 0 iki 1023 (2^{10}).

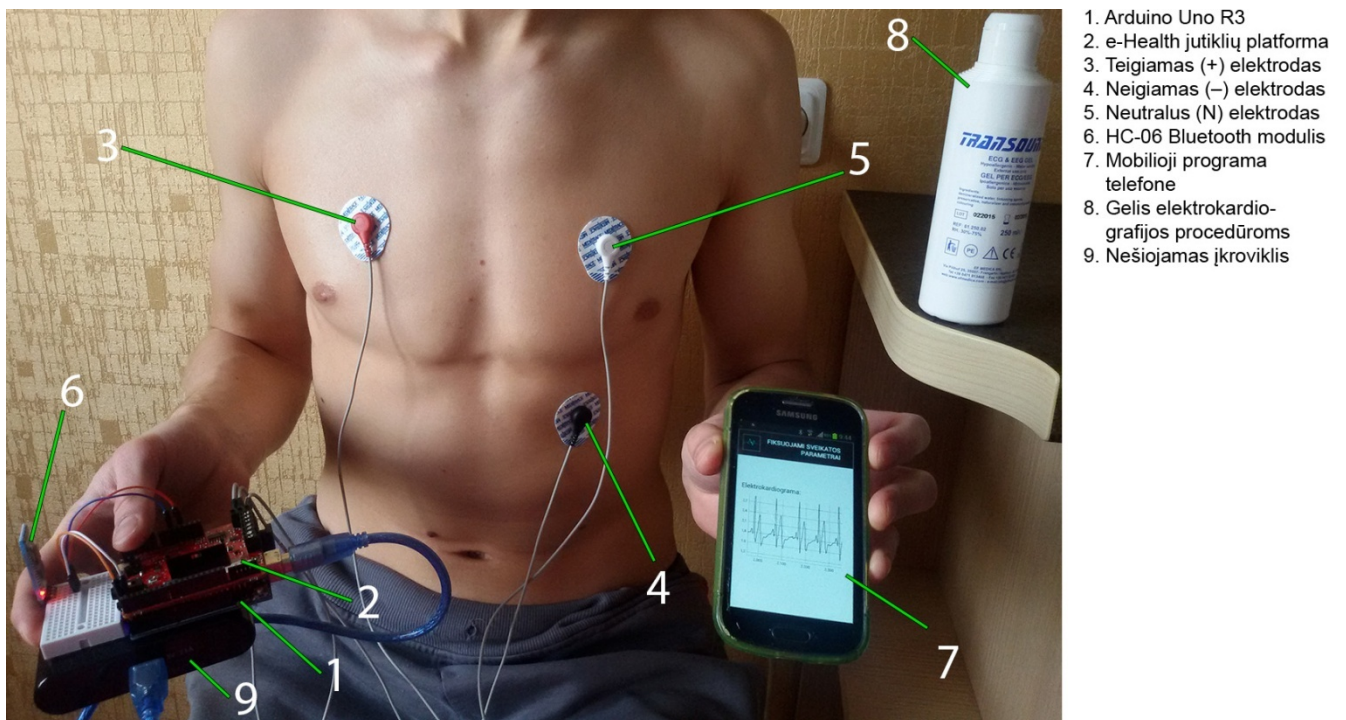


7 pav. Algoritmas širdies ritmo fiksavimui naudojant SEN-11574 pulso jutiklį

Vykdomas R bangų aptikimas. Naujas širdies dūžis užfiksuojamas jeigu signalas viršija slenkstinę vertę 525. Apskaičiuojamas RR intervalas, širdies ritmas. Vidutinis širdies ritmas skaičiuojamas imant dešimt prieš tai užfiksuotų širdies ritmo reikšmių. Po to uždegamas LED ir palaipsniui mažinant išėjimo signalo (D5) įtampą, mažinamas LED intensyvumas. Nustatoma LCD ekrano žymeklio pozicija, atvaizduojami jutiklių duomenys ekrane. Iš fiksuojamų parametų

sukuriamas JSON duomenų objektas. UART sąsaja duomenys perduodami į Bluetooth modulį, o iš jo į mobiliąją programą.

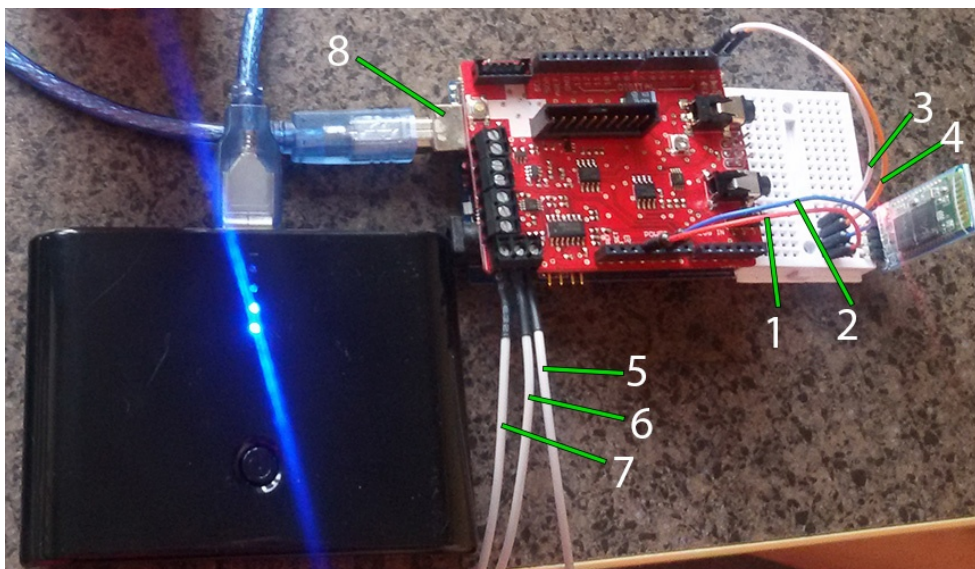
Suprojektuota techninė įranga naudojant e-Health platformą su EKG jutikliu pateikiama 8 pav. 1 numeriu pažymėta Arduino Uno R3 valdiklių plokštė, kuri yra sujungta su 2 numeriu pažymėta e-Health jutiklių platforma. e-Health platforma surenka fiksuojamus sveikatos parametrus iš jutiklių. Prie platformos prijungtas elektrokardiogramos (EKG) jutiklis, kurį sudaro trys elektrodai. 3 numeriu pažymėtas teigiamas (+) elektrodas, 4 numeriu – neigiamas (–), o 5 numeriu – neutralus (N). Užfiksuotas EKG signalas perduodamas į e-Health platformą, o iš jos į Arduino Uno valdiklių plokštę. Iš signalo nustatomas asmens širdies ritmas ir RR intervalas.



8 pav. Suprojektuota techninė įranga naudojant e-Health platformą su EKG jutikliu

Sveikatos parametrai iš Arduino plokštės perduodami į 6 numeriu pažymėtą HC-06 Bluetooth modulį. Iš jo duomenys perduodami į mobiliąjį telefoną. 7 numeriu pažymėta sukurta mobilioji programa, kurioje atvaizduojama elektrokardiograma. Odos laidumo padidinimui naudojamas gelis elektrokardiografijos procedūroms, kuris pažymėtas 8 numeriu. Nešiojamas įkroviklis naudojamas kaip maitinimo šaltinis. Jis pažymėtas 9 numeriu.

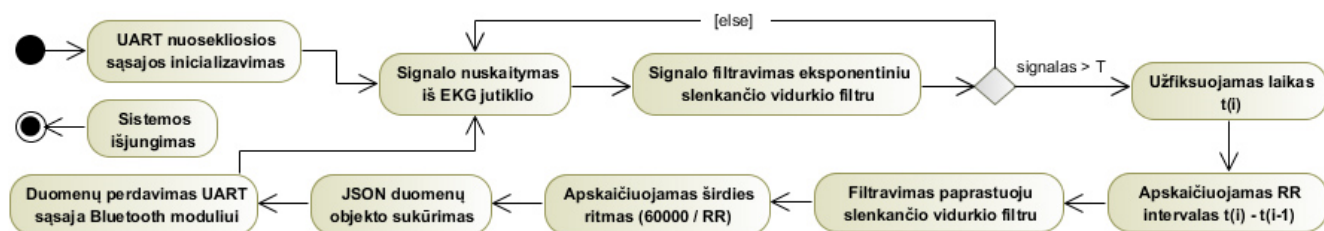
Techninės įrangos sujungimo schema pateikiama 9 pav. 1 numeriu pažymėtas HC-06 Bluetooth modulio teigiamas kontaktas, o 2 – neigiamas. Duomenys iš Arduino plokštės į modulį perduodami naudojant UART sąsają. Modulio RX išvadas (3) sujungiamas su Arduino plokštės TX išvadu, o modulio TX išvadas (4) su Arduino – RX išvadu. Prie e-Health platformos prijungiamas elektrokardiogramos jutiklio teigiamas (5), neigiamas (6) ir neutralus (7) kontaktai. Prie Arduino plokštės USB-B tipo jungties (8) prijungiamas nešiojamas įkroviklis. Įkroviklio išėjimas lygus 5V.



1. HC-06 Bluetooth modulio teigiamas kontaktas
2. HC-06 Bluetooth modulio neigiamas kontaktas
3. HC-06 Bluetooth modulio RX išvadas
4. HC-06 Bluetooth modulio TX išvadas
5. EKG jutiklio teigiamas kontaktas
6. EKG jutiklio neigiamas kontaktas
7. EKG jutiklio neutralus kontaktas
8. Arduino Uno R3 USB-B tipo jungtis

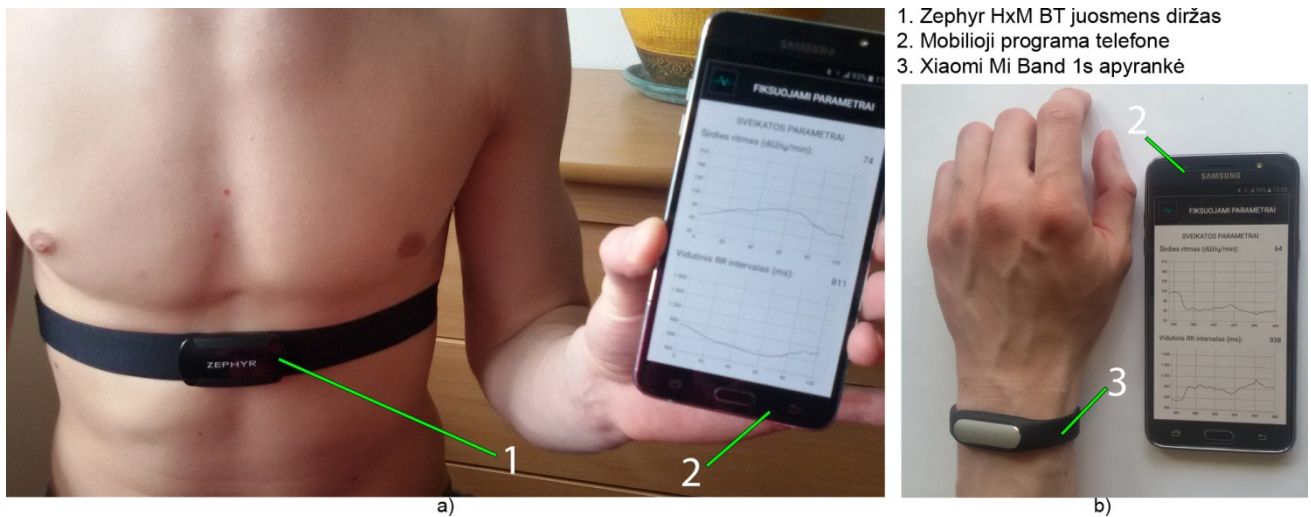
9 pav. Techninės įrangos sujungimo schema naudojant e-Health platformą su EKG jutikliu

Veiklos diagrama pateikiama 10 pav. Joje pavaizduotas širdies ritmo fiksavimo algoritmas naudojant e-Health platformą su EKG jutikliu. Nuskaitytas signalas iš EKG jutiklio filtruojamas naudojant eksponentinį slenkančio vidurkio (angl. *exponential moving average*, *EMA*) filtrą. Koeficientas α pasirinkta, kad būtų lygus 20. Filtravimo rezultatai pateikiami eksperimentinėje dalyje. Jeigu signalo reikšmė didesnė už slenksstinę reikšmę ($T = 650$), tada užfiksuojamas laikas $t(i)$. Apskaičiuojamas RR intervalas iš laiko $t(i)$ atimant prieš tai užfiksuotą laiką $t(i-1)$. RR intervalas filtruojamas naudojant paprastąjį slenkančio vidurkio filtrą (angl. *simple moving average*, *SMA*). Imamos 20 prieš tai buvusių reikšmių skaičiuojant vidutinį RR. Pagal vidutinį RR apskaičiuojamas širdies ritmas. Po to iš užfiksuotų parametrų sukuriama JSON duomenų objektas. Duomenys perduodami Bluetooth moduliui, o iš jo į mobiliąją programą.



10 pav. Algoritmas širdies ritmo fiksavimui naudojant e-Health platformą su EKG jutikliu

Širdies ritmo fiksavimas naudojant Zephyr HxM BT juosmens diržą pateikiamas 11 pav. a. Juosmens diržu (1) fiksuojami asmens sveikatos (širdies ritmas, RR intervalas) ir judėjimo (žingsnių skaičius, nueitas atstumas, momentinis greitis, ritmas – ilgų žingsnių/min) parametrai. Dirže integruotas Bluetooth modulis, kuris duomenis perduoda į mobilųjį telefoną (2). Jame įdiegta sukurta mobilioji programa, kurioje atvaizduojami užfiksuoti sveikatos ir judėjimo parametrai. Širdies ritmo fiksavimas naudojant Xiaomi Mi Band 1s apyrankę pateikiamas 11 pav. b. Apyranke (3) fiksuojami sveikatos (širdies ritmas, RR intervalas) ir judėjimo (žingsnių skaičius) parametrai, kurie Bluetooth ryšiu perduodami į telefoną (2), kuriame įdiegta mobilioji programa.



11 pav. Širdies ritmo fiksavimas Zephyr HxM BT juosmens diržu, Xiaomi Mi Band 1s apyrankė

Pasirinktais jutikliais fiksuojami parametrai pateikiami 4 lentelėje.

4 lentelė. Fiksuojami parametrai jutikliais

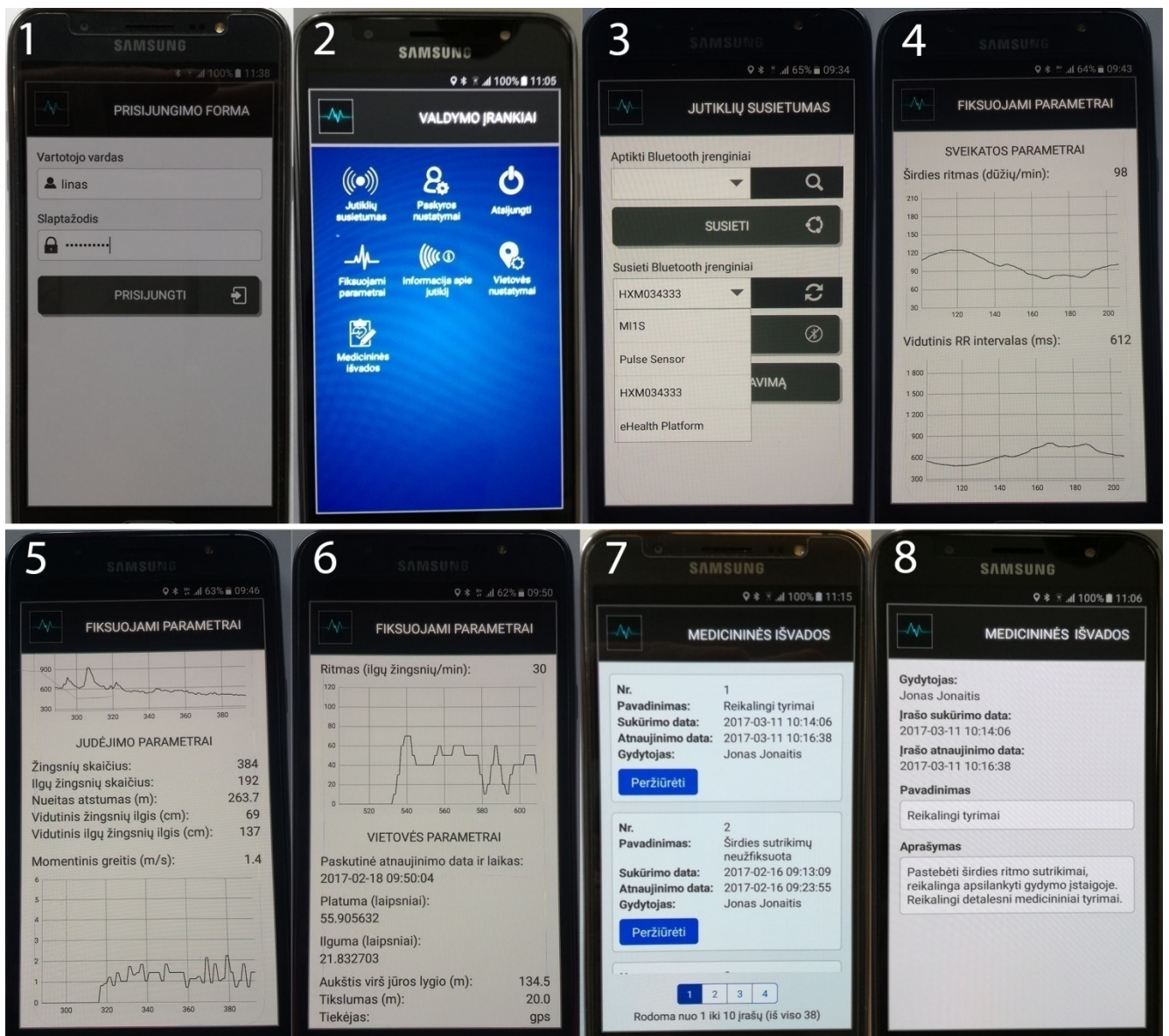
Fiksuojamas parametras	SEN-11574 pulso jutiklis	e-Health platforma su EKG jutikliu	Zephyr HxM BT juosmens diržas	Xiaomi Mi Band 1s apyrankė
Elektrokardiogramos (EKG) signalas (angl. <i>electrocardiogram (ECG) signal</i>), V	✓	✓	–	–
Širdies ritmas (angl. <i>heart rate, HR</i>), dūžių/min Laiko intervalas tarp dūžių (angl. <i>beat-to-beat interval, RR</i>), ms	✓	– (jutiklis fiksuoja tik EKG signalą. HR, RR – reikalinga pačiam įgyvendinti algoritmą)	✓	✓
Žingsnių skaičius (angl. <i>steps</i>) Ilgų žingsnių skaičius (angl. <i>strides</i>)	–	–	✓	✓
Nueitas atstumas (angl. <i>distance</i>), m	–	–	✓	–
Momentinis greitis (angl. <i>instant speed</i>), m/s	–	–	✓	–
Ritmas (angl. <i>cadence</i>), ilgų žingsnių/min	–	–	✓	–

Visi jutikliai vienu metu nenaudojami. Norint užfiksuoti elektrokardiogramą reikalingas SEN-11574 pulso jutiklis arba e-Health platforma su EKG jutikliu. Judėjimo parametrai fiksuojami tik Zephyr HxM BT juosmens diržu arba Xiaomi Mi Band 1s apyrankė.

2.3.2. Mobilioji programa fiksuojamų parametru siuntimui į serverį

Pateiktoje sistemoje jutikliais užfiksuoti parametrai Bluetooth ryšiu perduodami į mobilųjį įrenginį su Android OS. Jame įdiegta sukurta mobilioji programa, kuri pateikiama 12 pav. Norint naudotis programa reikalingas Wi-Fi arba mobilusis interneto ryšys. Įjungus programą pateikiama prisijungimo forma, kuri pažymėta 1 numeriu. Vartotojas norėdamas prisijungti prie sistemos turi suvesti vartotojo vardą ir slaptažodį, o nuspaudus mygtuką „Prisijungti“ įvykdoma užklausa. Sėkmingos autentifikacijos metu pateikiamas programos langas (2) su valdymo įrankiais. Vartotojas gali atlikti jutiklių susietumą, keisti paskyros nustatymus, peržiūrėti fiksuojamus parametrus ir medicininės išvadas, atsijungti nuo sistemos. 3 numeriu pažymėtas jutiklių susietumo langas. Mobiliajame telefone turi būti įgalintas Bluetooth ryšys. Su paieškos mygtuku aptinkami šalia esantys Bluetooth įrenginiai, kurių pavadinimai pateikiami sąrašė. Norimas įrenginys susiejamas su mobiliuoju

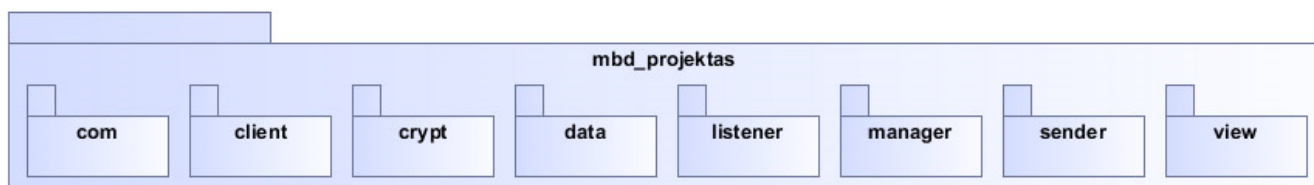
telefonu. Nuspaudus mygtuką „Pradėti fiksavimą“ pradedamas duomenų siuntimas iš jutiklio į mobiliąją programą. 4, 5 ir 6 numeriais pažymėtas fiksuojamų parametrų peržiūros langas. Jame pateikiami sveikatos (širdies ritmas, RR intervalas), judėjimo (žingsnių skaičius, nueitas atstumas, vidutinis žingsnių ilgis, momentinis greitis, ritmas) ir vietovės (platuma, ilguma ir kt.) parametrai. Buvimo vietos koordinatės nustatomos naudojant GPS. Gydytojų nurodymai, rekomendacijos siunčiami pacientui. 7 numeriu pažymėtas programos langas, kuriame pateikiamas sąrašas su medicininėmis išvadomis. Sąrašo viršuje atvaizduojami naujausi įrašai. Pasirinkus norimą įrašą pagal datą ir nuspaudus mygtuką „Peržiūrėti“ atidaromas 8 numeriu pažymėtas langas. Jame pateikiamas aprašymas apie paciento sveikatos būklę.



12 pav. Užfiksuotų parametrų, medicininių išvadų atvaizdavimas mobiliojoje programoje

Mobiliojoje programoje taip pat tikrinama ar užfiksuotas širdies ritmas neperžengia leistinos ribos. Esant nukrypimams SMS žinute ir elektroniniu laiškų informuojami suinteresuoti asmenys. Žinutėje pateikiamas internetinio puslapio adresas peržiūrėti fiksuojamus parametrus.

UML paketų diagrama (angl. *package diagram*) pateikiama 13 pav. Joje atvaizduojama mobiliosios programos klasių ir interfeisų saugojimo struktūra. Mobiliąją programą sudaro 8 paketai. Klasės ir interfeiso pavadinimo sudarymui naudojamas to paketo pavadinimas, kuriame jie yra saugomi.



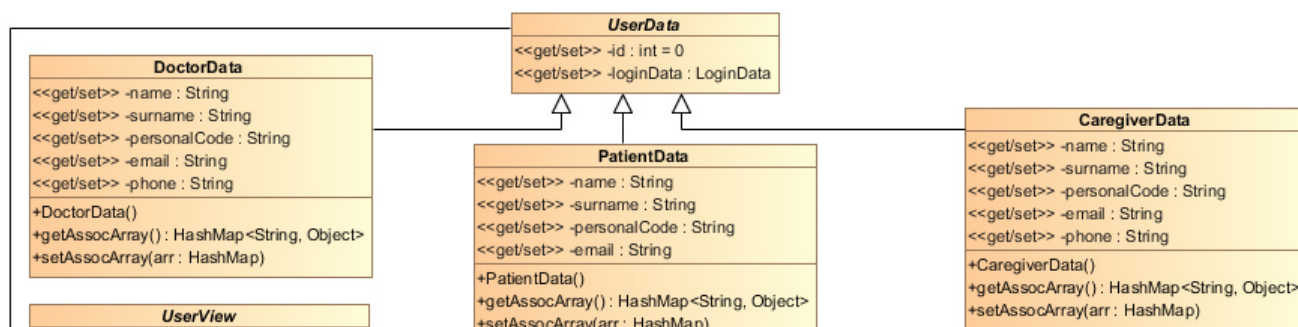
13 pav. Mobiliosios programos klasių ir interfeisų saugojimo struktūra

Mobiliosios programos klasės ir interfeisai sugrupuoti pagal jų paskirtį. 5 lentelėje pateikiama paketų paskirtis.

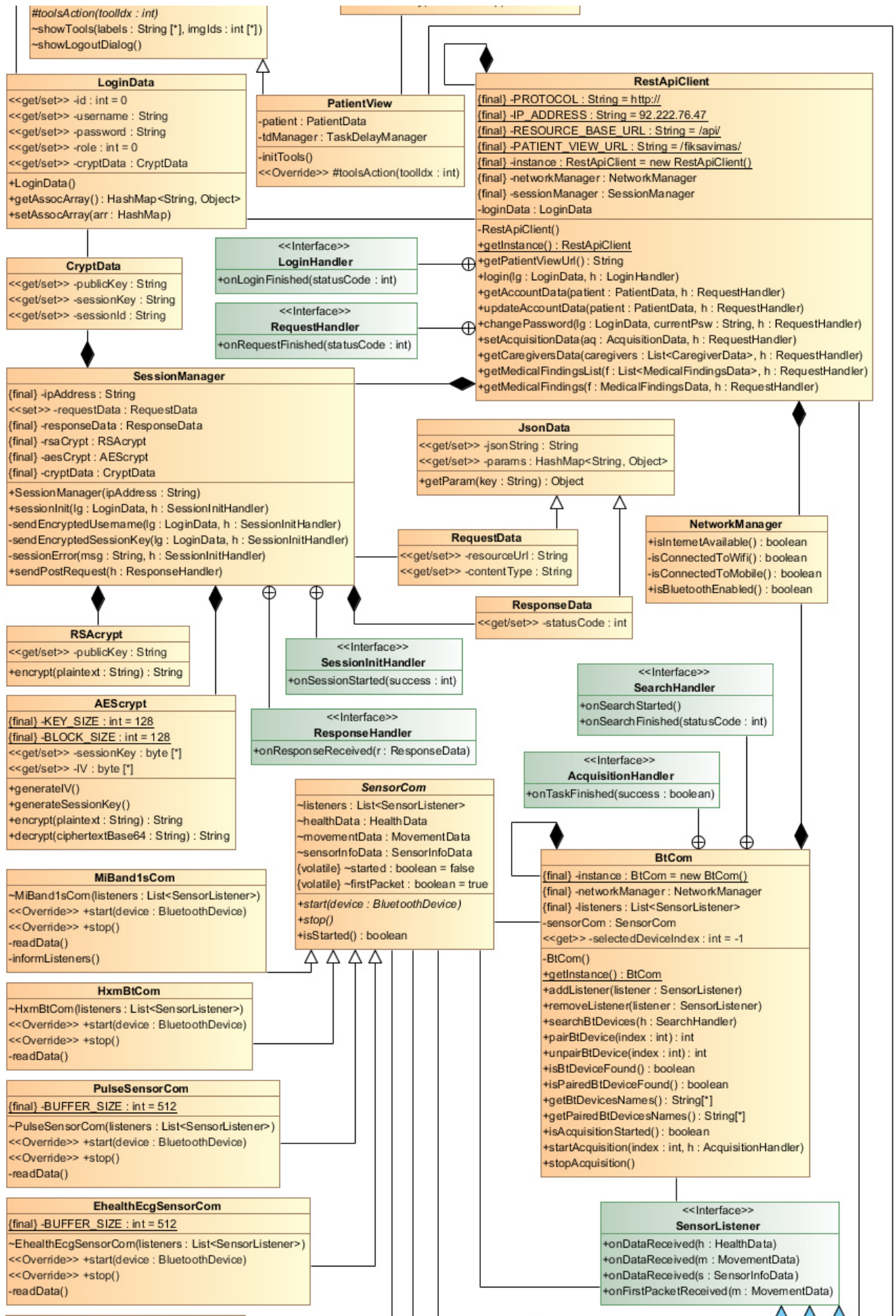
5 lentelė. Mobiliosios programos paketų paskirtis

Paketas	Paskirtis	Paketas	Paskirtis
com	saugomos klasės, kurios naudojamos komunikacijai su jutikliais	listener	saugomi interfeisai, kurie naudojami perduoti fiksuojamus parametrus iš vienos klasės į kitą
client	saugoma klasė, kuri atsakinga už komunikaciją su debesų serveriu	manager	saugomos klasės, kurios naudojamos valdyti vartotojo prisijungimo sesiją, buvimo vietos koordinacių fiksavimą, užduočių vykdymą nustatytu laiko intervalu
crypt	saugomos klasės, kurios naudojamos šifruoti siunčiamą informaciją	sender	saugomos klasės, kurios atsakingos už asmenų informaciją SMS žinute, el. laišku
data	saugomos klasės, kurių atributuose laikomi programos duomenys: fiksuojami parametrai, informacija apie susietą jutiklį, iš debesų serverio gauti duomenys ir t.t.	view	saugomos klasės, kurios naudojamos atvaizduoti programos langus ir grafinės sąsajos elementus

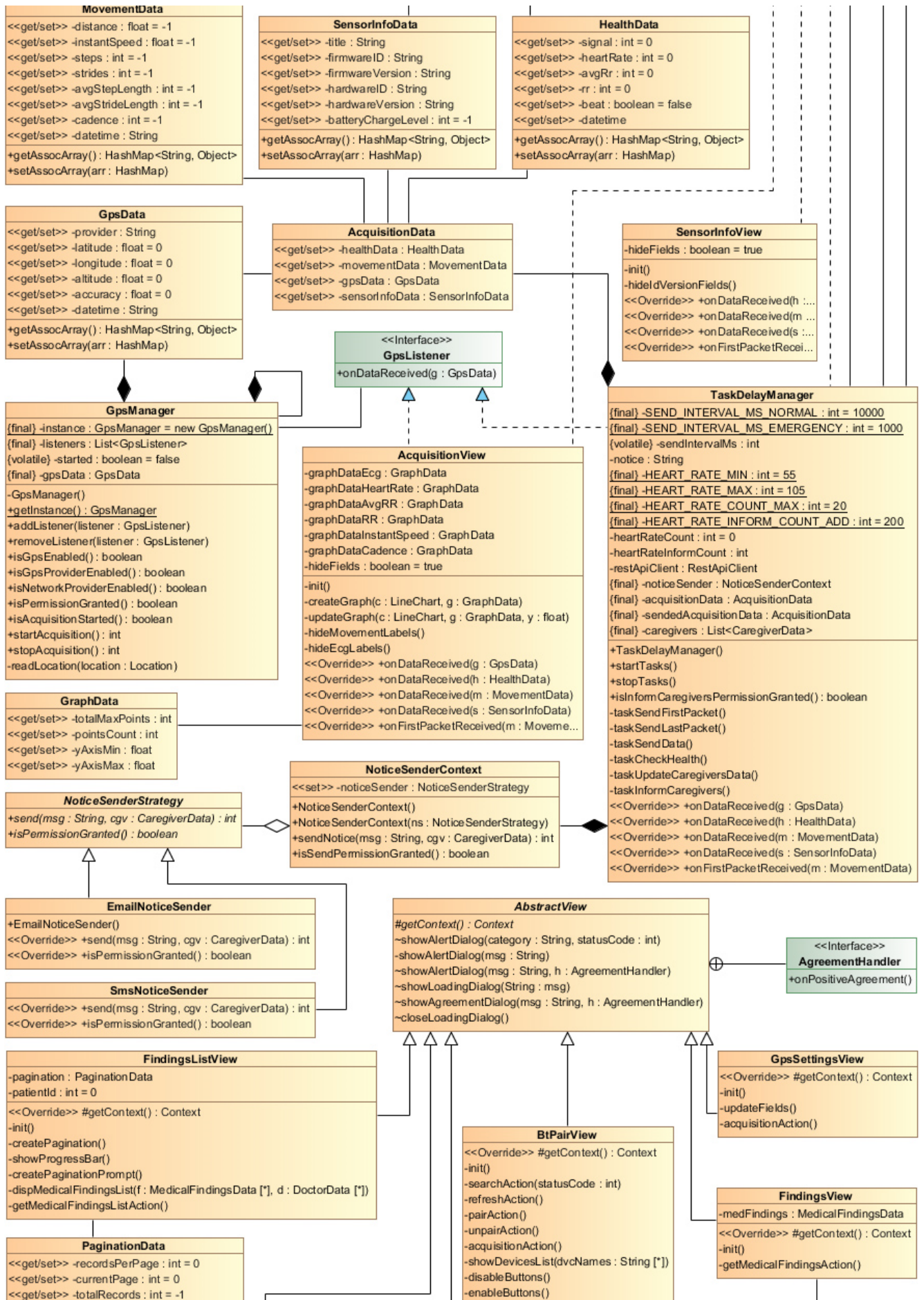
UML klasių diagrama (angl. *class diagram*) pateikiama 14 pav., 15 pav., 16 pav. ir 17 pav. Joje pavaizduotas mobiliosios programos statinis vaizdas. Klasių *RestApiClient*, *BtCom* ir *GpsManager* sukūrimui panaudotas singletono (angl. *singleton*) projektavimo šablonas, tam kad būtų galima sukurti tik vieną vartotojo prisijungimo sesiją prie debesų serverio, jutiklio ir mobiliajame įrenginyje integruoto GPS. Asmenų informavimas SMS žinute, el. laišku esant širdies ritmo nukrypimams, įgyvendintas panaudojant strategijos (angl. *strategy*) projektavimo šabloną. Tam sukurtos keturios klasės: abstrakti – *NoticeSenderStrategy*, ją paveldinčios – *EmailNoticeSender*, *SmsNoticeSender* ir taikymo – *NoticeSenderContext*.



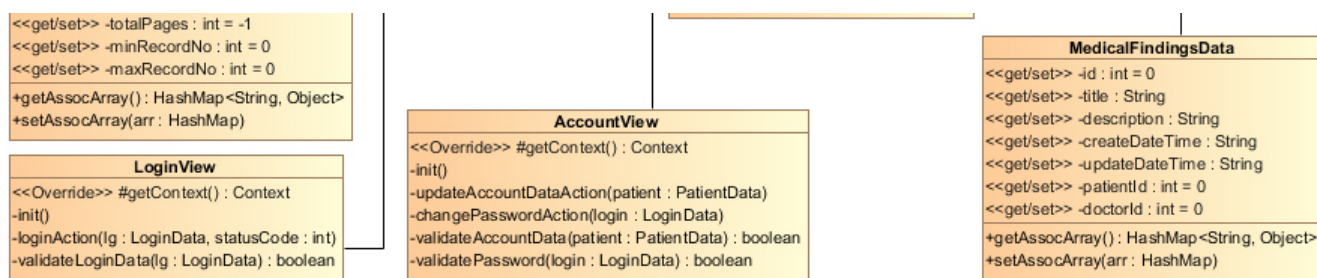
14 pav. Mobiliosios programos statinis vaizdas (1)



15 pav. Mobiliosios programos statinis vaizdas (2)



16 pav. Mobiliosios programos statinis vaizdas (3)



17 pav. Mobiliosios programos statinis vaizdas (4)

Visų mobiliuojoje programoje naudojamų klasių ir interfeisų paskirtis pateikiama 6 lentelėje.

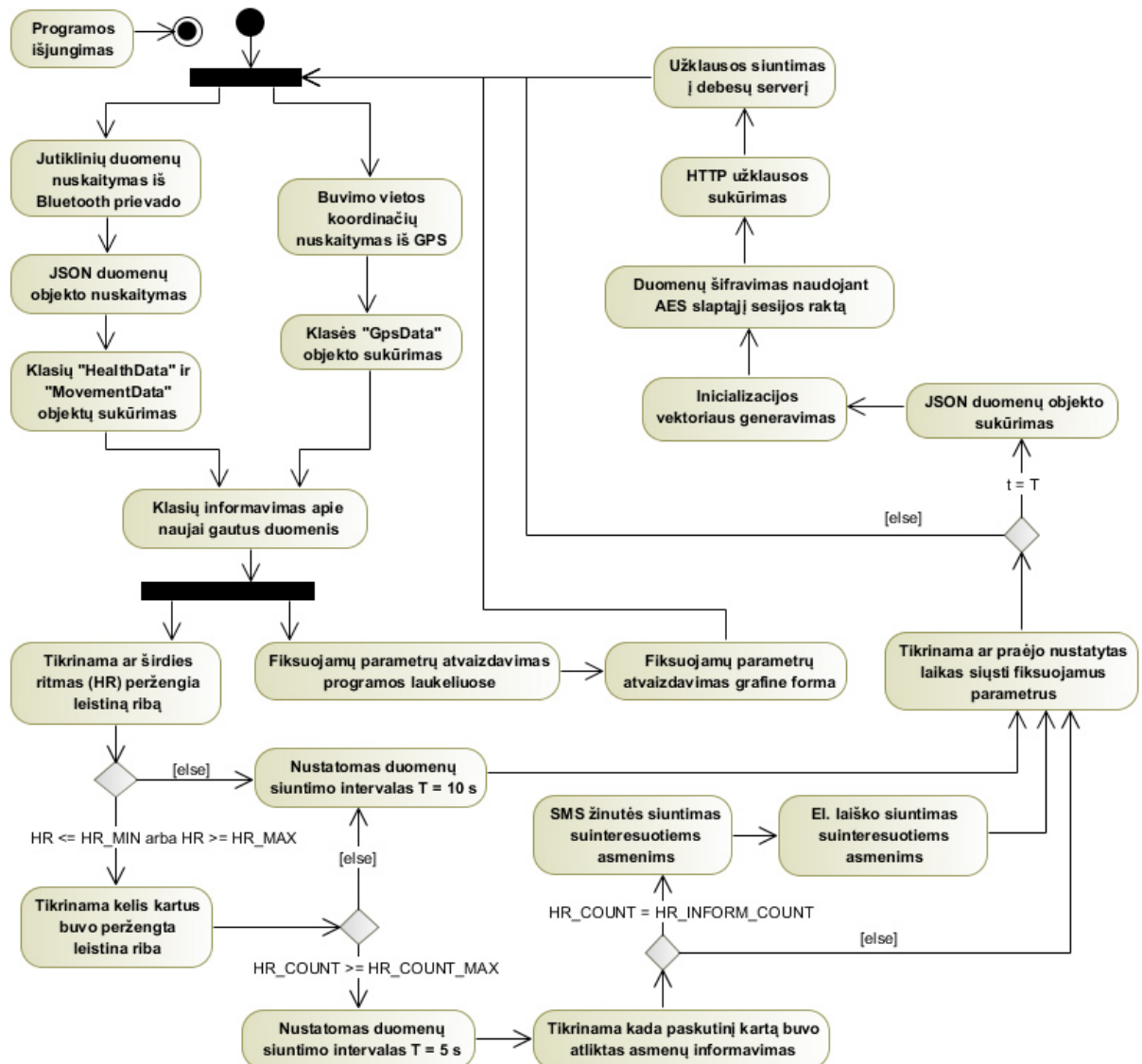
6 lentelė. Mobiliosios programos klasių ir interfeisų paskirtis

Klasė, interfeisas	Paskirtis
BtCom	atliekama Bluetooth įrenginių paieška, jų susiejimas ir atsiejimas su mobiliuoju telefonu, parametų fiksavimo pradėjimas ir sustabdymas
EhealthEcgSensorCom	surenkami duomenys iš Arduino valdiklių plokštės, kuri fiksuoja sveikatos parametrus naudojant e-Health platformą su EKG jutikliu
MiBand1sCom	surenkami fiksuojami sveikatos ir judėjimo parametrai iš Xiaomi Mi Band 1s apyrankės
HxmBtCom	surenkami fiksuojami sveikatos ir judėjimo parametrai iš Zephyr HxM BT juosmens diržo
PulseSensorCom	surenkami duomenys iš Arduino valdiklių plokštės, kuri fiksuoja sveikatos parametrus naudojant SEN-11574 pulso jutiklį
SensorCom	abstrakti klasė, kurią paveldi už komunikaciją su jutikliais atsakingos klasės. Joje apibrėžiami fiksavimo pradėjimo ir sustabdymo metodai, kurie turi būti įgyvendinti
RestApiClient	aprašytas REST API klientas. Juo naudojantis vykdomas vartotojo prisijungimas prie sistemos, paskyros informacijos atnaujinimas, slaptažodžio keitimas, fiksuojamų parametų siuntimas į debesų serverį, medicininių išvadų parsisiuntimas į mobiliąją programą iš serverio. Klasėje saugomi prisijungimo duomenys, komunikacijos protokolas, debesų serverio IP, REST API resursų adresas
AEScrypt	užšifruojama siunčiama informacija į debesų serverį ir iššifruojamas gautas atsakas. Generuojamas inicializacijos vektorius (angl. <i>initialization vector</i> , <i>IV</i>) ir sesijos slaptasis raktas (angl. <i>session secret key</i>). Naudojamas AES simetrinis šifravimo algoritmas. Rakto ir šifro bloko dydis 128-bit
RSACrypt	užšifruojamas sesijos slaptasis raktas naudojant debesų serverio viešąjį raktą (angl. <i>public key</i>). Naudojamas RSA asimetrinis šifravimo algoritmas
AcquisitionData	saugomi jutiklių ir GPS fiksuojami parametrai
CaregiverData	saugoma informacija apie asmenis, kurie informuojami el. laišku ir SMS žinute jeigu paciento širdies ritmas peržengia leistiną ribą
CryptData	saugomi viešasis raktas, slaptas sesijos raktas ir sesijos ID
DoctorData	saugoma informacija apie gydytoją
GpsData	saugomos buvimo vietos koordinatės: platumas ir ilgumas, fiksavimo tikslumas, aukštis virš jūros lygio, data ir laikas kada paskutinį kartą atnaujinta fiksuojama informacija
GraphData	saugomi grafiko duomenys: maksimalus skaičius taškų grafike, kintamasis skirtas skaičiuoti taškus, y ašies minimali ir maksimali reikšmės
HealthData	saugomi sveikatos parametrai
JsonData	saugomi simbolių eilutė JSON formatu, duomenys raktas-reikšmė pavidalu. Duomenys paverčiami JSON formatu ir atvirkščiai
LoginData	saugomi prisijungimo duomenys: id, vartotojo vardas, slaptažodis ir vartotojo grupės numeris
MedicalFindingsData	saugoma informacija apie medicinines išvadas
MovementData	saugomi judėjimo parametrai
PaginationData	saugomi medicininių išvadų puslapiavimo duomenys: įrašų skaičius puslapyje, dabartinio puslapio numeris, iš viso įrašų, iš viso puslapių
PatientData	saugoma informacija apie pacientą
RequestData	saugomi siunčiamos užklausos duomenys: REST API resurso adresas, siunčiamo turinio formatas

ResponseData	saugomi atsako duomenys: HTTP būsenos kodas (200, 400, 404, 405)	
SensorInfoData	saugoma informacija apie jutiklį, kuris buvo pasirinktas fiksuoti parametrus	
UserData	abstrakti klasė, kurią paveldi vartotojų grupės	
GpsListener	interfeisas, kuriuo naudojantis fiksuojami buvimo vietos parametrai perduodami iš vienos klasės į kitą	
SensorListener	interfeisas, kuriuo naudojantis fiksuojami sveikatos ir judėjimo parametrai perduodami iš vienos klasės į kitą	
GpsManager	pradedamas ir sustabdomas buvimo vietos koordinacijų fiksavimas. Taip pat tikrinama ar GPS mobiliajame telefone įjungtas, ar GPS fiksavimo leidimai suteikti programai	
NetworkManager	tikrinama ar yra prisijungta prie Wi-Fi arba mobiliojo interneto ryšio, ar įrenginyje įgalintas Bluetooth ryšys	
SessionManager	įvykdoma vartotojo prisijungimo sesijos inicializacija. HTTP protokolu siunčiamos užklausos į debesų serverį, nuskaitomas gražintas atsakas. Naudojamas POST metodas	
TaskDelayManager	nustatytu laiko intervalu siunčiami fiksuojami parametrai į debesų serverį, tikrinama ar širdies ritmas neperžengia leistinos ribos, siunčiama SMS žinutė, el. laiškas suinteresuotiems asmenims esant širdies ritmo nukrypimams	
EmailNoticeSender	siunčiamas nurodytas el. pranešimas į el. paštą	
NoticeSenderContext	klasė, kuria valdomos EmailNoticeSender ir SmsNoticeSender klasės	
NoticeSenderStrategy	abstrakti klasė, kurią paveldi pranešimų siuntimą įgyvendinančios klasės	
SmsNoticeSender	siunčiamas nurodytas SMS pranešimas į mobiliųjį telefoną	
AbstractView	abstrakti klasė, kurią paveldi grafinės sąsajos elementus atvaizduojančios klasės. Jos paveldi metodus, kurie skirti atvaizduoti įspėjimą, laukimo ir patvirtinimo langus	
AccountView	atvaizduojamas paskyros nustatymų langas	
AcquisitionView	atvaizduojamas fiksuojamų parametru peržiūros langas	
BtBairView	atvaizduojamas Bluetooth įrenginių paieškos, susietumo su mobiliuoju telefonu langas	
FindingsListView	atvaizduojamas programos langas, kuriame pateikiamas sąrašas su medicininėmis išvadomis	
FindingsView	atvaizduojamas programos langas, kuriame pateikiamas aprašymas apie paciento sveikatos būklę	
GpsSettingsView	atvaizduojamas GPS nustatymų langas	
LoginView	atvaizduojamas programos langas su prisijungimo forma	
PatientView	atvaizduojamas paciento valdymo langas su įrankių sąrašu	
SensorInfoView	atvaizduojamas programos langas, kuriame pateikiama informacija apie fiksavimui pasirinktą jutiklį	
UIView	abstrakti klasė, kurią paveldi vartotojų grupių įrankius atvaizduojančios klasės. Jos paveldi metodą atvaizduojantį atsijungimo langą nuo sistemos	
LoginHandler	RestApiClient klasės vidiniai interfeisai, kuriais naudojantis informuojamos kitos klasės	apie vartotojo prisijungimą prie sistemos
RequestHandler		apie užklausos įvykdymą debesų serveryje
SessionInitHandler	SessionManager klasės vidiniai interfeisai, kuriais naudojantis informuojamos kitos klasės	apie prisijungimo sesijos inicializavimą
ResponseHandler		apie atsako gavimą iš debesų serverio
SearchHandler	BtCom klasės vidiniai interfeisai, kuriais naudojantis informuojamos kitos klasės	apie Bluetooth įrenginių paieškos pradžią ir pabaigą
AcquisitionHandler		apie parametru fiksavimo pradėjimą ir sustabdymą
AgreementHandler	AbstractView klasės vidinis interfeisas, kuriais naudojantis informuojamos kitos klasės	apie patvirtinimo mygtuko nuspaudimą pateiktame lange

Veiklos diagrama pateikiama 18 pav. Joje pavaizduotas fiksuojamų parametru siuntimo į debesų serverį algoritmas naudojant mobiliąją programą. Diagramoje pateikiamas programos vykdymas nuo to momento, kai vartotojas prisijungia prie sistemos, pasirinktą jutiklį susieja su mobiliuoju telefonu ir nuspausdamas mygtuką pradeda parametru fiksavimą. Iš Bluetooth prievado (angl. *socket*) nuskaitomi jutiklių duomenys. JSON duomenų objektas konvertuojamas į masyvą raktas-reikšmė pavidalu. Tuo

pačiu momentu taip pat vykdomas buvimo vietos koordinacių nuskaitymas iš GPS. Vietovės parametrų fiksavimas vykdomas kas 10 sekundžių, o sveikatos, judėjimo – nuolatos. Iš fiksuojamų parametrų sukuriama klasių *HealthData*, *MovementData* ir *GpsData* objektai. Informuojamos interfeisus realizuojančios klasės apie naujai gautus duomenis.



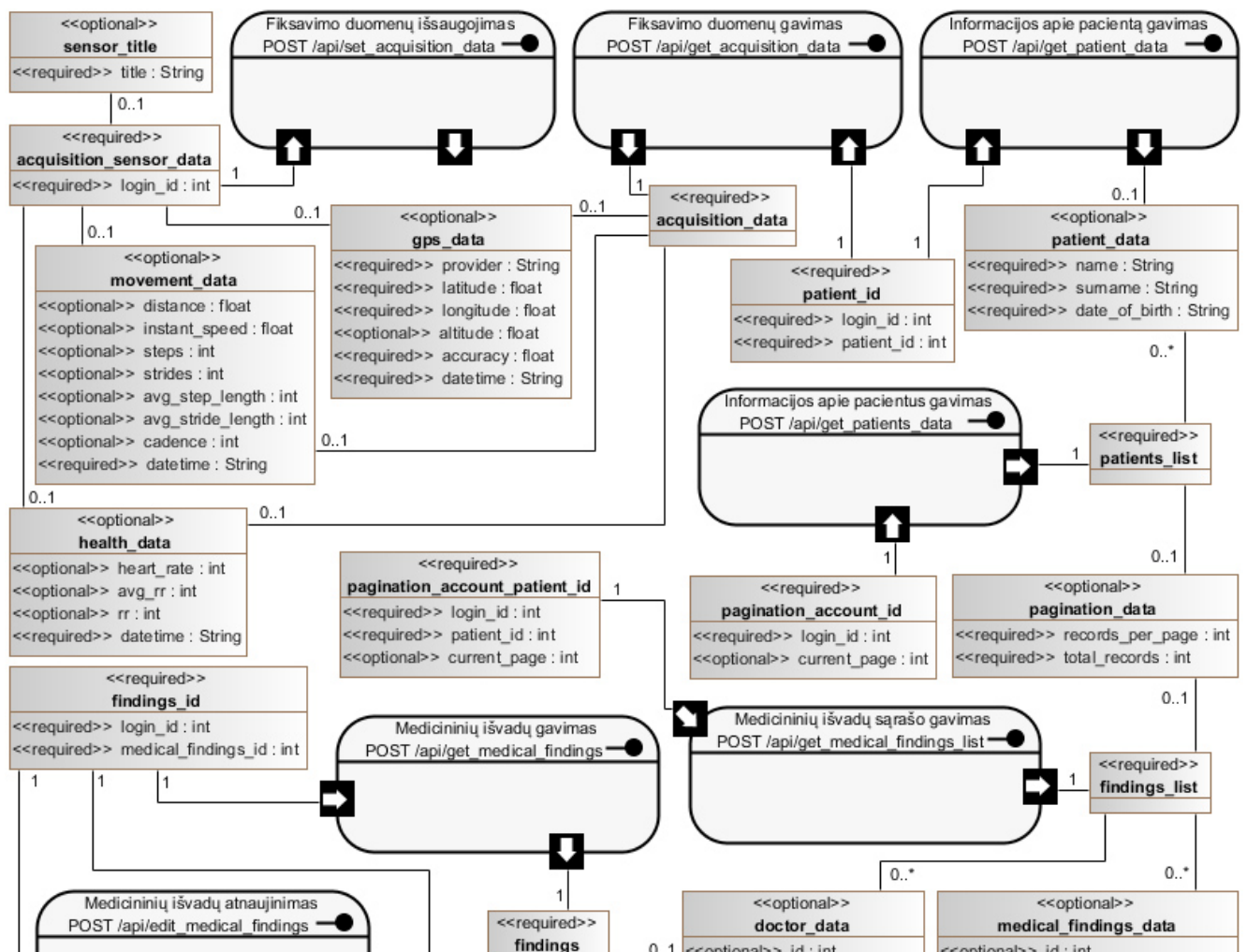
18 pav. Algoritmas fiksuojamų parametrų siuntimui iš mobiliosios programos į debesų serverį

Fiksuojami parametrai atvaizduojami skaitine forma programos laukeliuose ir grafine forma – kreivėse. Tuo pačiu momentu tikrinama ar širdies ritmas peržengia nurodytą leistiną ribą ($HR_{MIN} = 55$, $HR_{MAX} = 105$). Jeigu sąlyga tenkinama tikrinama kiek kartų buvo peržengta leistina riba ($HR_{COUNT_MAX} = 20$). Sąlygai tenkinant tikrinama kada paskutinį kartą buvo atliktas asmenų informavimas (pradžioje $HR_{INFORM_COUNT} = 20$, išsiuntus pranešimą padidinama 200). Jeigu trečioji sąlyga tenkinama SMS žinute ir el. laišku informuojami suinteresuoti asmenys. Fiksuojami parametrai siunčiami į debesų serverį kas 10 sekundžių, jeigu širdies ritmas peržengią leistiną ribą – kas 5 sekundes. Iš siunčiamų duomenų sukuriamas JSON duomenų objektas. Sugeneruojamas inicializacijos vektorius (IV). JSON objektas užšifruojamas naudojant AES simetrinį šifravimą

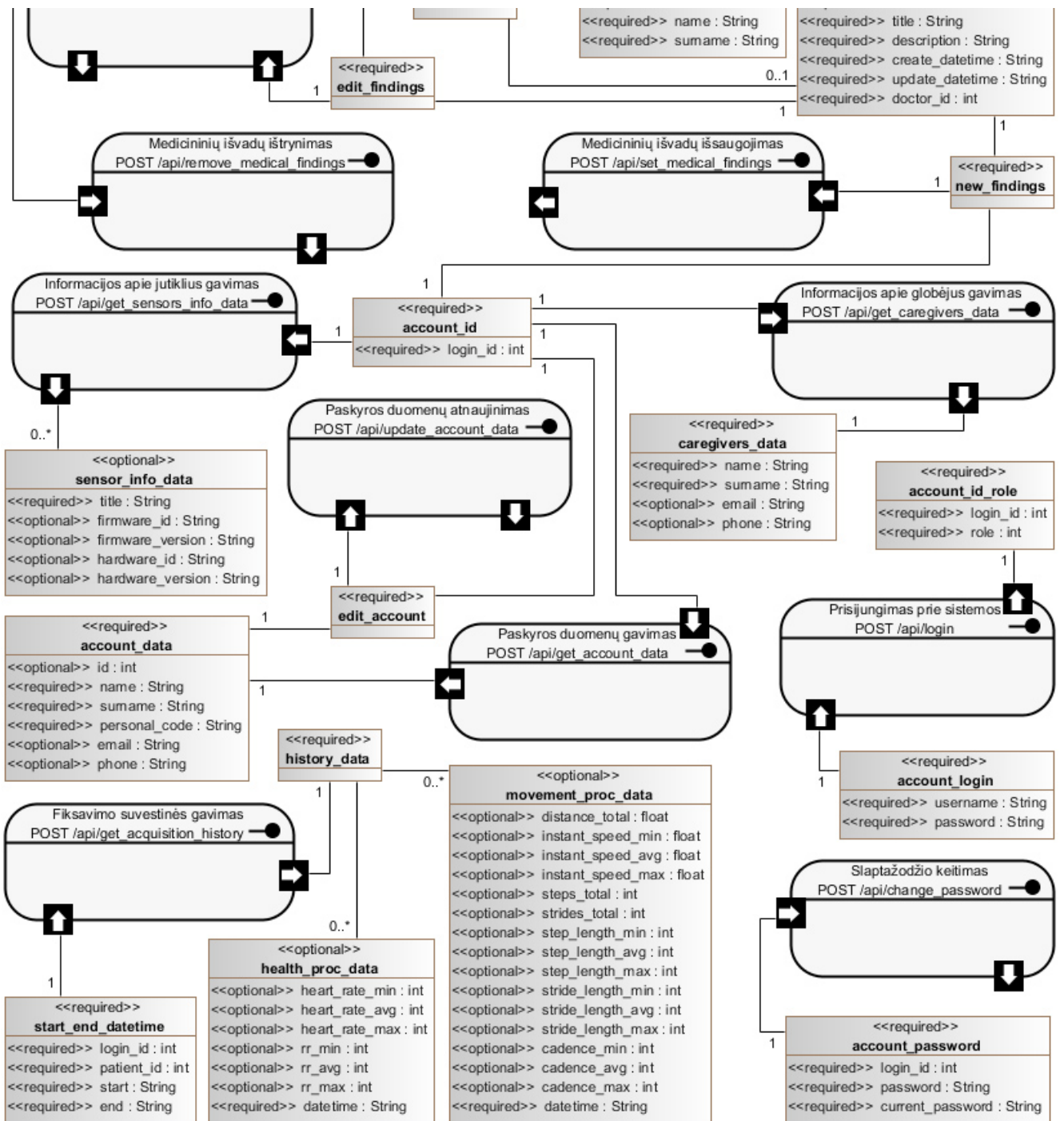
algoritmą su kintamu IV ir slaptu sesijos raktu. Sukuriama HTTP užklausa, kurią sudaro sesijos ID, inicializacijos vektorius atviro teksto pavidalu ir užšifruotas JSON duomenų objektas. Užklausa išsiunčiama į debesų serverį.

2.3.3. Duomenų apdorojimas debesų serveryje

Pateiktoje sistemoje užfiksuoti sveikatos, judėjimo ir vietovės parametrai Wi-Fi arba mobiliuoju interneto ryšiu siunčiami į debesų serverį. Jame įdiegta aplikacijų programavimo sąsaja (angl. *Application Programming Interface, API*), kurios kūrimui panaudotas REST (angl. *Representational State Transfer*) architektūrinis stilius. Naudojant sukurtą REST API vykdoma komunikacija tarp mobiliosios programos ir debesų serverio, tarp internetinio puslapio ir serverio. Vartotojų užklausa, debesų serverio atsakas siunčiami HTTP (angl. *HyperText Transfer Protocol*) protokolu naudojant POST metodą. Duomenų objektai perduodami JSON formatu. REST API diagrama pateikiama 19 pav. ir 20 pav. Joje pavaizduoti REST resursai (angl. *resources*), kurie identifikuojami URI (angl. *Uniform Resource Identifier*) adresais. Jais naudojantis mobilioji programa ir internetinis puslapis kreipiasi į debesų serverį, kuris išsaugo, grąžina, atnaujina ir ištrina reikalingą informaciją. Diagramoje pateikiama iš kokių parametrų sudarytos vartotojų užklausa ir serverio atsakas.



19 pav. Sukurta REST API, kuri įdiegta debesų kompiuterijos infrastruktūroje (1)



20 pav. Sukurta REST API, kuri įdiegta debesų kompiuterijos infrastruktūroje (2)

REST API resursų paskirtis pateikiama 7 lentelėje.

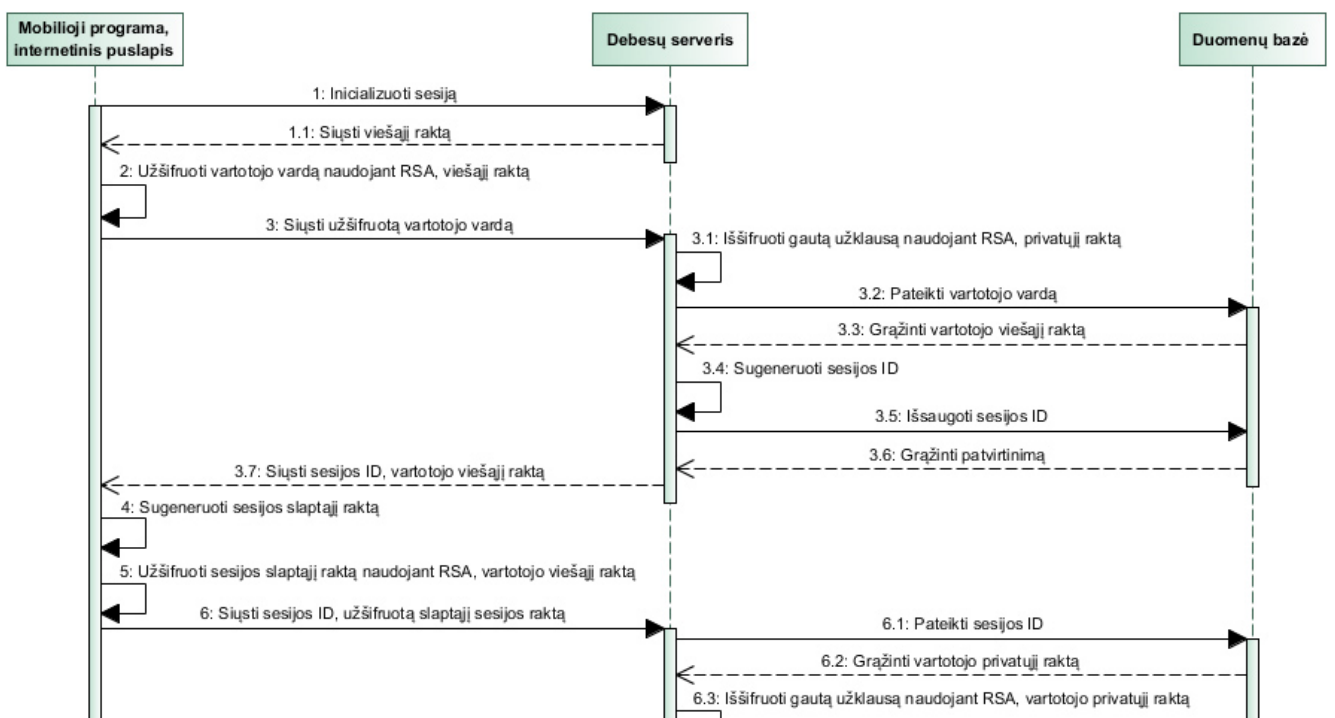
7 lentelė. REST API resursų paskirtis

Resursas	Paskirtis
/api/set_acquisition_data	duomenų bazėje išsaugomi fiksuojami paciento parametrai. Vartotojo užklausa sudaryta iš sveikatos, judėjimo, vietovės parametrų ir fiksavimui pasirinkto jutiklio pavadinimo
/api/get_acquisition_data	užklauskos siuntėjui atsake grąžinamas naujausias duomenų bazės įrašas, kuris sudarytas iš fiksuojamų parametrų. Užklausoje pateikiamas paciento ID
/api/get_patient_data	pagal paciento ID grąžinama informacija apie jį
/api/get_patients_data	užklauskos siuntėjui grąžinamas atsakas su įrašų sąrašu, kuriame pateikiami sistemoje užregistruoti pacientai. Jeigu užklausoje nurodytas

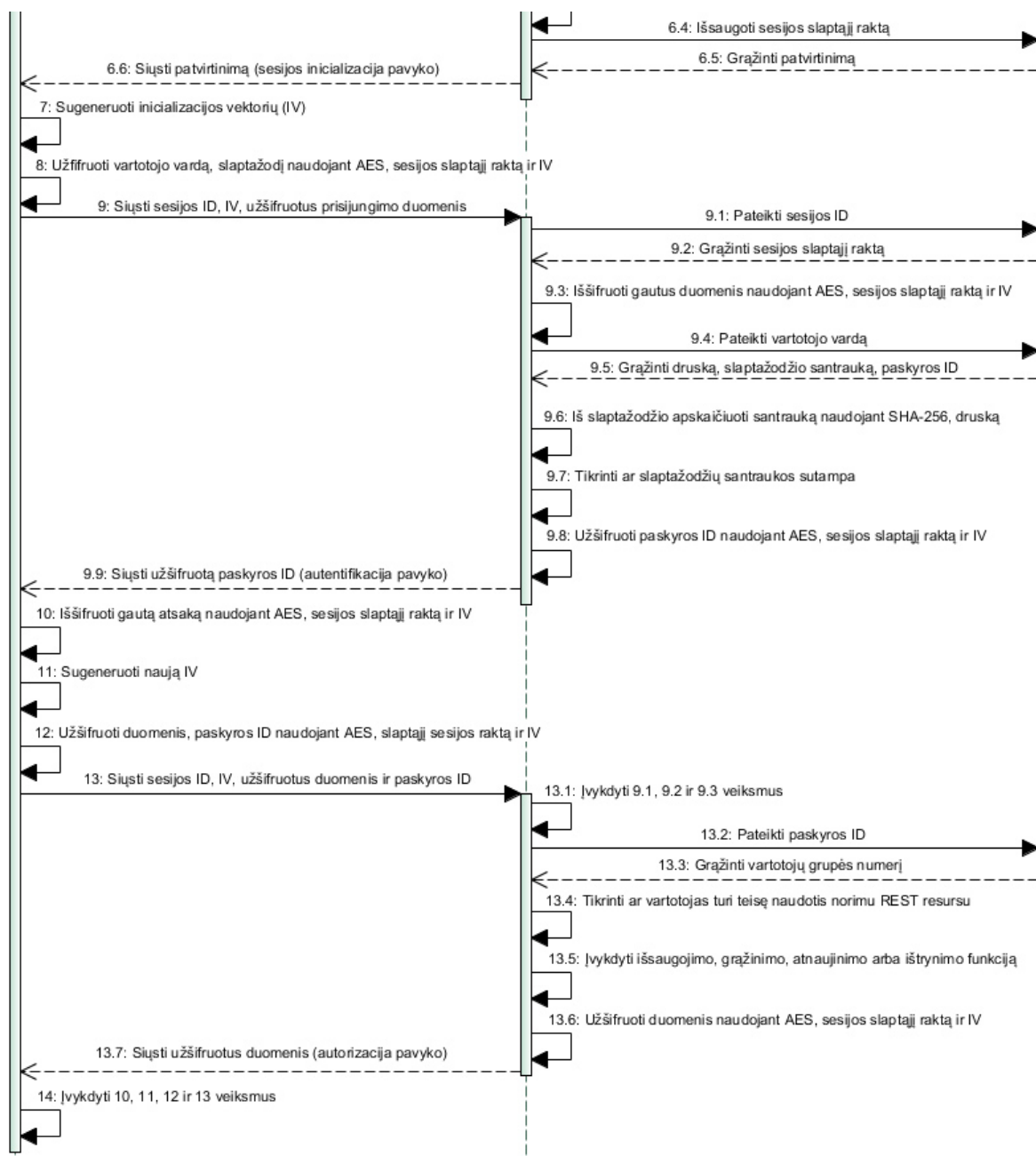
	parametras <i>current_page</i> , grąžinama tik dalis sąrašo. Taip pat serveris atsake nurodo kiek iš viso įrašų yra duomenų bazėje, norint gauti sąrašo likusias dalis
/api/get_medical_findings	atsake grąžinama medicininė išvada, mediko vardas ir pavardė. Išvadoje nurodomas mediko pateiktas aprašymas apie paciento sveikatos būklę. Užklausoje nurodomas medicininės išvados ID
/api/get_medical_findings_list	užklauskos siuntėjui grąžinamas atsakas su įrašų sąrašu, kuriame pateikiamos paciento medicininės išvados. Užklausoje nurodomas paciento ID. Atsake pateikiama dalis sąrašo jeigu užklausoje nurodytas parametras <i>current_page</i>
/api/edit_medical_findings	duomenų bazėje atnaujinamas medicininės išvados įrašas. Užklausoje pateikiamas medicininės išvados ID, pakoreguotas pavadinimas ir aprašymas apie paciento sveikatos būklę
/api/remove_medical_findings	iš duomenų bazės ištrinamas medicininės išvados įrašas pagal jos ID
/api/set_medical_findings	duomenų bazėje išsaugoma mediko pateikta medicininė išvada apie paciento sveikatą
/api/get_caregivers_data	užklauskos siuntėjui grąžinamas atsakas su sąrašu, kuriame pateikiami asmenų el. pašto adresai ir telefono numeriai, kurie informuojami apie paciento širdies ritmo nukrypimus
/api/get_sensors_info_data	atsake grąžinama informacija apie jutiklius, kurie naudojami fiksuoti paciento sveikatos ir judėjimo parametrus
/api/update_account_data	duomenų bazėje atnaujinama informacija apie paskyrą pagal jos ID
/api/get_account_data	užklauskos siuntėjui grąžinamas atsakas su informacija apie jo paskyrą
/api/change_password	pakeičiamas vartotojo slaptažodis. Užklausoje nurodomas dabartinis ir naujas slaptažodis
/api/login	įvykdoma vartotojo autentifikacija pagal užklausoje pateiktą vartotojo vardą ir slaptažodį. Atsake grąžinamas paskyros ID (<i>login_id</i>) pagal, kurį vykdoma REST resursų autorizacija
/api/get_acquisition_history	pagal užklausoje nurodytą paciento ID, pradžios ir pabaigos datą, laiką grąžinamas atsakas su fiksavimo suvestine

UML sekų diagrama (angl. *sequence diagram*) pateikiama 21 pav. ir 22 pav. Joje pavaizduotas sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas. Vartotojo prisijungimo prie sistemos metu inicializuojama sesija. Debesų serveryje iš *.pem* failo nuskaitomas viešasis raktas (angl. *public key*), kuris nusiunčiamas vartotojui. Su RSA (Rivest-Shamir-Adleman) asimetriniu algoritmu ir viešuoju raktu užšifruojamas (angl. *encrypt*) vartotojo vardas. Šifruotas tekstas (angl. *ciphertext*) nusiunčiamas į debesų serverį. Gauta užklausa iššifruojama (angl. *decrypt*) naudojant RSA ir privatųjį raktą (angl. *private key*), kuris nuskaitomas iš *.pem* failo. Debesų serveryje pagal iššifruotą vartotojo vardą iš duomenų bazės (DB) paimamas vartotojo viešasis raktas (angl. *user public key*). Sugeneruojama sesijos ID (angl. *session ID*), kuri išsaugoma duomenų bazėje. Vartotojo viešasis raktas ir sesijos ID atviro teksto (angl. *plaintext*) pavidalu nusiunčiami vartotojui. Mobiliojoje programoje, internetiniame puslapyje sugeneruojamas sesijos slaptasis raktas (angl. *session secret key*), kuris užšifruojamas naudojant RSA algoritmą ir vartotojo viešąjį raktą. Sesijos ID ir užšifruotas slaptasis raktas nusiunčiami į debesų serverį. Pagal sesijos ID iš duomenų bazės paimamas vartotojo privatusis raktas (angl. *user private key*). Iš gautos užklauskos iššifruojamas sesijos slaptasis raktas naudojant RSA ir vartotojo privatųjį raktą. Sesijos slaptasis raktas išsaugomas duomenų bazėje. Vartotojui nusiunčiamas atsakas, kuris patvirtina, kad pavyko inicializuoti sesiją. Abu vartotojas ir debesų serveris turi sesijos slaptąjį raktą, todėl pradedama vykdyti vartotojo autentifikacija (angl.

authentication). Mobiliojoje programoje, internetiniame puslapyje sugeneruojamas inicializacijos vektorius (angl. *initialization vector, IV*). Su AES (angl. *Advanced Encryption Standard*) simetriniu algoritmu, sesijos slaptuoju raktu ir IV užšifruojamas vartotojo vardas ir slaptažodis. Sesijos ID, IV ir užšifruoti prisijungimo duomenys nusiunčiami į debesų serverį. Pagal sesijos ID iš duomenų bazės paaimamas slaptasis raktas. Iš užklauso iššifruojami prisijungimo duomenys naudojant AES algoritmą, sesijos slaptąjį raktą ir IV. Pagal iššifruotą vartotojo vardą iš DB paaimama druska (angl. *salt*), slaptažodžio santrauka/maišos reikšmė (angl. *digest/hash value*) ir paskyros ID (*login_id*). Iš vartotojo atsiųsto slaptažodžio apskaičiuojama santrauka naudojant SHA-256 maišos funkciją (angl. *hash function*) ir druską. Tikrinama ar slaptažodžių santraukos (iš DB paaimta ir apskaičiuota iš atsiųsto slaptažodžio) sutampa. Jeigu sąlyga tenkinama – autentifikacija pavyko. Tada paskyros ID užšifruojamas naudojant AES, sesijos slaptąjį raktą ir IV, o šifruotas atsakas nusiunčiamas vartotojui. Gautas atsakas iššifruojamas su AES algoritmu. Abu vartotojas ir debesų serveris turi paskyros ID, todėl pradeda vykdyti vartotojo autorizacija (angl. *authorization*) REST API užklauso siuntimo metu. Sugeneruojamas naujas IV. REST API užklausa ir paskyros ID užšifruojami naudojant AES, sesijos slaptąjį raktą ir naująjį IV. Sesijos ID, IV, užšifruota REST API užklausa ir paskyros ID nusiunčiami į debesų serverį. Jame pagal sesijos ID iš DB paaimamas slaptasis raktas ir iššifruojama REST API užklausa. Duomenų bazei pagal pateiktą paskyros ID gaunamas vartotojų grupės numeris. Tikrinama ar vartotojas turi teisę naudotis norimu REST resursu. Jeigu sąlyga tenkinama – autorizacija pavyko. Tokiu atveju įvykdoma norima sistemos funkcija ir vartotojui grąžinamas AES algoritmu užšifruotas REST API atsakas. Jeigu vartotojas nori siųsti naują REST API užklauso kartojami 22 pav. pateikti 10, 11, 12 ir 13 veiksmai.



21 pav. Sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas (1)



22 pav. Sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas (2)

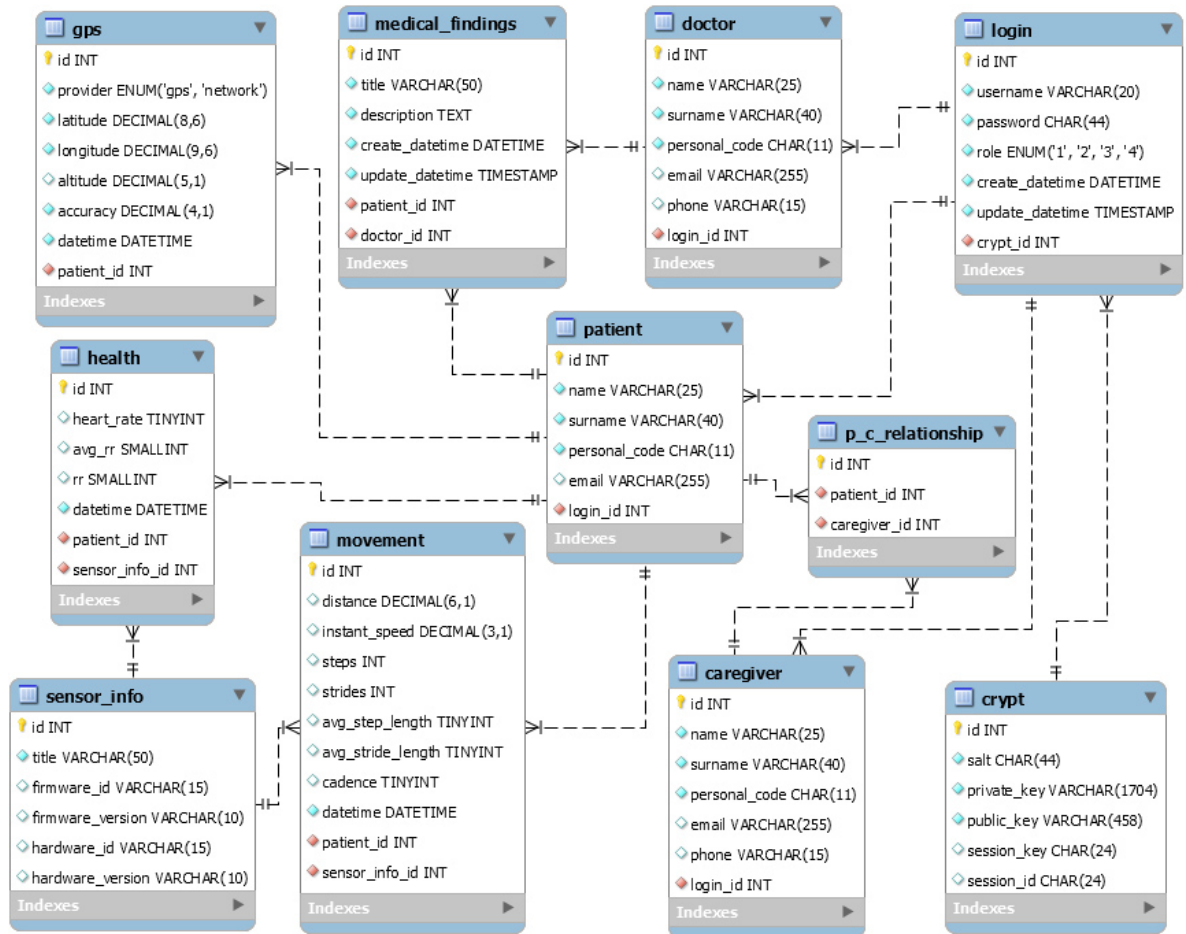
Pasirinkti šifravimo algoritmų parametrai pateikiami 8 lentelėje. Slaptažodžio keitimo metu sugeneruojama nauja druska, vartotojo viešojo ir privačiojo raktų pora. Šie parametrai atnaujinami duomenų bazėje. Siunčiant REST API užklausas visada sugeneruojamas naujas IV, o sesijos inicializacijos metu – naujas slapstasis raktas ir sesijos ID.

8 lentelė. Pasirinkti šifravimo algoritmų parametrai

Parametras		Reikšmė	Parametras	Reikšmė
RSA modulio (angl. <i>modulus</i>) ilgis		2048-bit	RSA ir AES šifruoto teksto kodavimas	
RSA viešojo ir privačiojo rakto	formatas	PKCS#1	SHA-256 santraukos	ilgis
	kodavimas (angl. <i>encoding</i>)	Base64		kodavimas
AES sesijos slaptojo rakto	ilgis	128-bit	SHA-256 druskos	ilgis
	kodavimas	Base64		kodavimas
AES šifruoto teksto bloko ilgis		128-bit	SHA-256 druska pridama prie atviro teksto	
			pabaigos	

AES inicializacijos vektorius (IV)	ilgis	128-bit	Sesijos ID	ilgis	128-bit
	kodavimas	Base64		kodavimas	Base64

Esybių-ryšių diagrama (angl. *entity-relationship diagram*, ERD) pateikiama 23 pav. Joje pavaizduota suprojektuota duomenų bazės struktūra. Sukurta 11 lentelių, kurios susietos ryšiais.



23 pav. Suprojektuota duomenų bazės struktūra

Duomenų bazės lentelių paskirtis pateikiama 9 lentelėje.

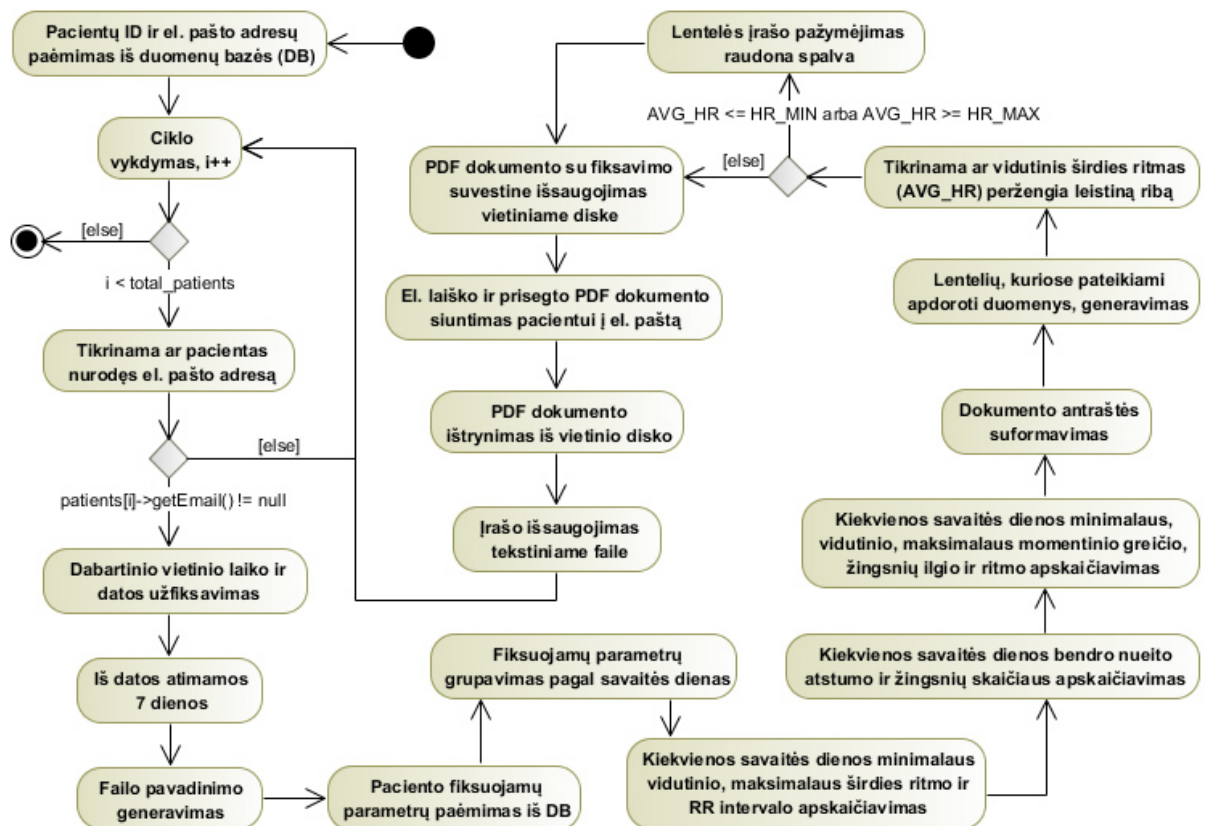
9 lentelė. Duomenų bazės lentelių paskirtis

DB lentelė	Paskirtis
caregiver	saugoma informacija apie asmenis, jų kontaktiniai duomenys, kurie informuojami SMS žinute, el. laiškų jeigu paciento širdies ritmas peržengia leistiną ribą
crypt	saugomi šifravimo algoritmų parametrai: druska, vartotojo viešasis ir privatusis raktai, laikini sesijos ID ir slaptasis raktas
doctor	saugoma informacija apie gydytojus
gps	saugomi fiksuojami vietovės parametrai: platumas, ilgumas, fiksavimo tikslumas, aukštis virš jūros lygio, fiksavimo data ir laikas
health	saugomi fiksuojami sveikatos parametrai: širdies ritmas, RR intervalas, fiksavimo data ir laikas, jutiklio ID su kuriuo buvo fiksuojami parametrai
login	saugomi sistemos vartotojų prisijungimo duomenys: paskyros ID, vartotojo vardas, slaptažodis, vartotojų grupės numeris, paskyros sukūrimo ir atnaujinimo datos
medical_findings	saugomos gydytojų pateiktos medicininės išvados: pavadinimas ir aprašymas apie paciento sveikatos būklę, įrašo sukūrimo ir atnaujinimo datos
movement	saugomi fiksuojami judėjimo parametrai: atstumas, momentinis greitis, žingsnių, ilgų žingsnių skaičius ir vidutinis ilgis, ritmas, fiksavimo data ir laikas, jutiklio ID su kuriuo buvo fiksuojami parametrai
patient	saugoma informacija apie pacientus

p_c_relationship	saugomi ryšiai tarp pacientų ir asmenų, kurie turi būti informuojami esant širdies ritmo nukrypimams
sensor_info	saugoma informacija apie sistemoje naudojamus jutiklius: pavadinimas, įrenginio ID ir versija, įrenginio programinės įrangos ID ir versija

Debesų serveryje naudojant *cron* įrankį aprašytos komandos, kurios periodiškai vykdo užduotis. Sukurtas programinis skriptas, kuris atlieka duomenų bazės atsarginių kopijų darymą. Kopijos automatiškai siunčiamos į Dropbox serverį. Skriptas automatiškai vykdomas kas savaitę. Sužadinus skriptą iškviečiama funkcija, kuri *mysqldump* įrankiui paduoda parametrus ir duomenų bazę išeksportuojama į *.sql* failą. Failas archyvuojamas į *.zip* archyvą, kurio pavadinimas sudaromas iš datos ir laiko. Naudojant Dropbox API duomenų bazės kopija *.zip* archyve įkeliami į serverį.

Taip pat sukurtas programinis skriptas generuojantis PDF dokumentą, kuriame pateikiama paciento savaitinė fiksavimo suvestinė. Ji automatiškai siunčiama pacientui į el. pašta. Veiklos diagrama pateikiama 24 pav. Joje pavaizduotas fiksavimo suvestinės PDF dokumento generavimo algoritmas debesų serveryje.



24 pav. Algoritmas fiksavimo suvestinės PDF dokumento generavimui debesų serveryje

Iš duomenų bazės paimami prie sistemos priregistruotų pacientų ID ir el. pašto adresai. Vykdomas ciklas ir tikrinama ar paciento paskyroje nurodytas el. pašto adresas. Tenkinant sąlygai užfiksuojamas dabartinis vietinis laikas ir data. Iš datos atimamos 7 dienos. Sugeneruojamas failo pavadinimas. Pagal paciento ID, pradžios ir pabaigos datas iš DB paimami paciento fiksuojami parametrai, kurie sugrupuojami pagal savaitės dienas. Apdorojami paciento sveikatos ir judėjimo parametrai. Apskaičiuojami kiekvienos savaitės dienos minimalus, vidutinis, maksimalus širdies ritmas,

RR intervalas, momentinis greitis, žingsnių ilgis ir ritmas. Taip pat apskaičiuojama koks bendras nueitas atstumas ir žingsnių skaičius kiekvienos savaitės dienos. Iš pradžios ir pabaigos datų suformuojama PDF dokumento antraštė, o iš apdorotų duomenų sugeneruojamos lentelės. PDF dokumentas su fiksavimo suvestine išsaugomas vietiniame diske. Pacientui į el. paštą išsiunčiamas el. laiškas prie kurio prisegamas dokumentas. Tekstiniame faile *.txt* išsaugojamas įrašas (angl. *log*) patvirtinantis apie fiksavimo suvestinės išsiuntimą. 25 pav. pateikiamas sugeneruotas PDF dokumentas su paciento fiksuotų parametrų savaitine suvestine. Sukurtas programinis skriptas sugeneruoja visų prie sistemos priregistruotų pacientų fiksavimo suvestines ir jis yra vykdomas kas savaitę.

Fiksavimo suvestinė nuo 2017-03-16 09:00:00 iki 2017-03-23 09:00:00

Sveikatos parametrai

Data ir laikas	Širdies ritmas, dūžių/min			RR intervalas, ms		
	Min.	Vid.	Maks.	Min.	Vid.	Maks.
2017-03-23	103	138	167	359	445	583
2017-03-22	68	81	91	659	743	882
2017-03-21	55	76	86	698	797	1091
2017-03-20	59	81	95	632	758	1017

Judėjimo parametrai

Data ir laikas	Žingsnių skaičius	Ilgų žingsnių skaičius	Nueitas atstumas, m	Žingsnių ilgis, cm			Ilgų žingsnių ilgis, cm			Momentinis greitis, m/s			Ritmas, ilgų žingsnių/min		
				Min.	Vid.	Maks.	Min.	Vid.	Maks.	Min.	Vid.	Maks.	Min.	Vid.	Maks.
2017-03-23	450	225	323.7	0	79	85	0	157	170	0	1.2	3.4	0	49	80
2017-03-22	612	306	387.6	0	60	68	0	120	135	0	0.9	1.4	0	44	60
2017-03-21	848	424	510.7	0	62	67	0	123	135	0	0.7	1.4	0	37	60
2017-03-20	420	210	254.4	0	56	64	0	111	128	0	0.6	1.4	0	33	60

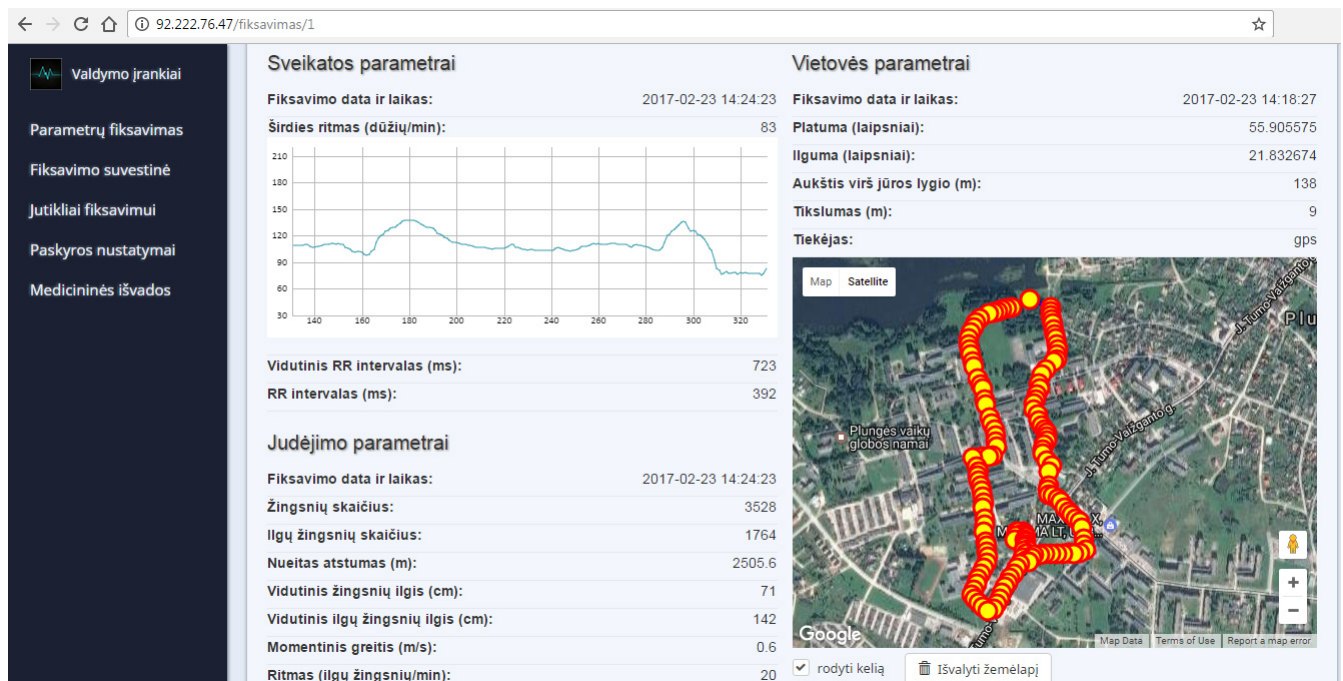
25 pav. Sugeneruotas PDF dokumentas su paciento fiksuotų parametrų savaitine suvestine

Lentelių eilutėse pateikiami apdoroti sveikatos ir judėjimo parametrai kiekvienos savaitės dienos. Suvestinėje raudona spalva pažymima vidutinė širdies ritmo reikšmė jeigu ji peržengia leistiną ribą.

2.3.4. Internetinis puslapis fiksuojamų parametrų atvaizdavimui

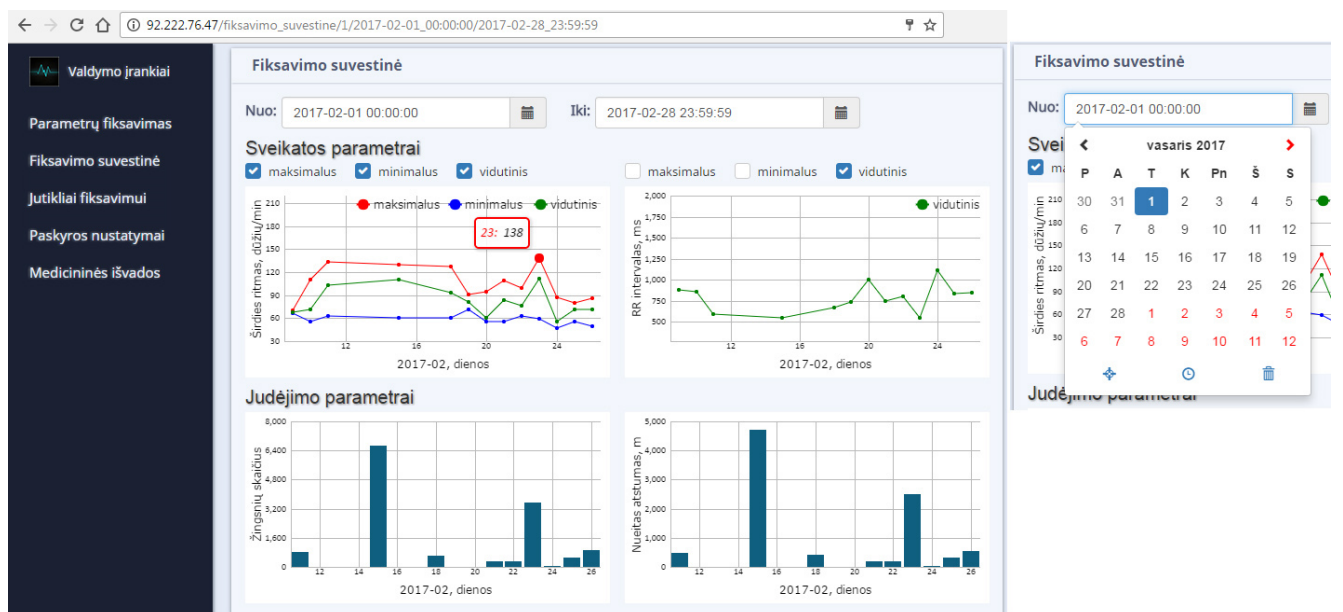
Sukurtas internetinis puslapis fiksuojamų parametrų atvaizdavimui. Naršyklėje suvedus puslapio adresą pateikiama prisijungimo forma, kurioje nurodomas vartotojo vardas ir slaptažodis. Sėkmingos autentifikacijos metu pateikiamas puslapio langas, kuriame atvaizduojami valdymo įrankiai. Medicinos personalas gali peržiūrėti paciento fiksuojamus parametrus, fiksavimo suvestinę, sistemoje naudojamus jutiklius, keisti paskyros nustatymus, atsijungti nuo sistemos ir pateikti medicininę išvadą apie paciento sveikatos būklę. Pasirinkus valdymo įrankį „Parametrų fiksavimas“ pateikiamas sistemoje užregistruotų pacientų sąrašas. Pasirinkus pacientą pateikiamas internetinio puslapio langas, kuriame atvaizduojami sveikatos, judėjimo ir vietovės parametrai užfiksuoti tam tikru laiko momentu (26 pav.). Puslapyje pateikiama data ir laikas kada buvo atliktas parametrų fiksavimas. Kreivėje atvaizduojamas paciento širdies ritmo kitimas. Skaitine forma pateikiami užfiksuoti judėjimo parametrai. Pasinaudojus užfiksuotomis platumos ir ilgumos koordinatėmis Google žemėlapyje

atvaizduojama paciento buvimo vieta. Fiksuojami parametrai internetiniame puslapyje atnaujinami kas 5 sekundes. Panaudota AJAX technologija atnaujinant puslapio turinį jo neperkraunant.



26 pav. Širdies ritmo, judėjimo ir vietovės parametrai užfiksuoti tam tikru laiko momentu

Internetiniame puslapyje medicinos personalui pateikiama parametų fiksavimo suvestinė (27 pav.). Pasirinkus norimą laikotarpį suformuojama užklausa, kuri nusiunčiama į debesų serverį. Jame apdorojami užfiksuoti paciento sveikatos ir judėjimo parametrai. Iš serverio grąžinamas atsakas su apdorotais duomenimis. Kreivėse atvaizduojamas minimalus, vidutinis, maksimalus širdies ritmas ir RR intervalas. Stulpelinėse diagramose pateikiamas bendras nueitas atstumas ir žingsnių skaičius.



27 pav. Širdies ritmo ir judėjimo parametų fiksavimo suvestinė

Priklausomai nuo pasirinkto laikotarpio sudaromos metų, mėnesio, dienos ar valandos fiksavimo suvestinės. Pasirinkus skirtingų metų datas grafikų x ašyje pateikiami metai, o pasirinkus to paties

mėnesio datas – dienos. Datose pasirinkus konkrečią dieną x ašyje pateikiamos valandos, o pasirinkus konkrečią valandą – minutės.

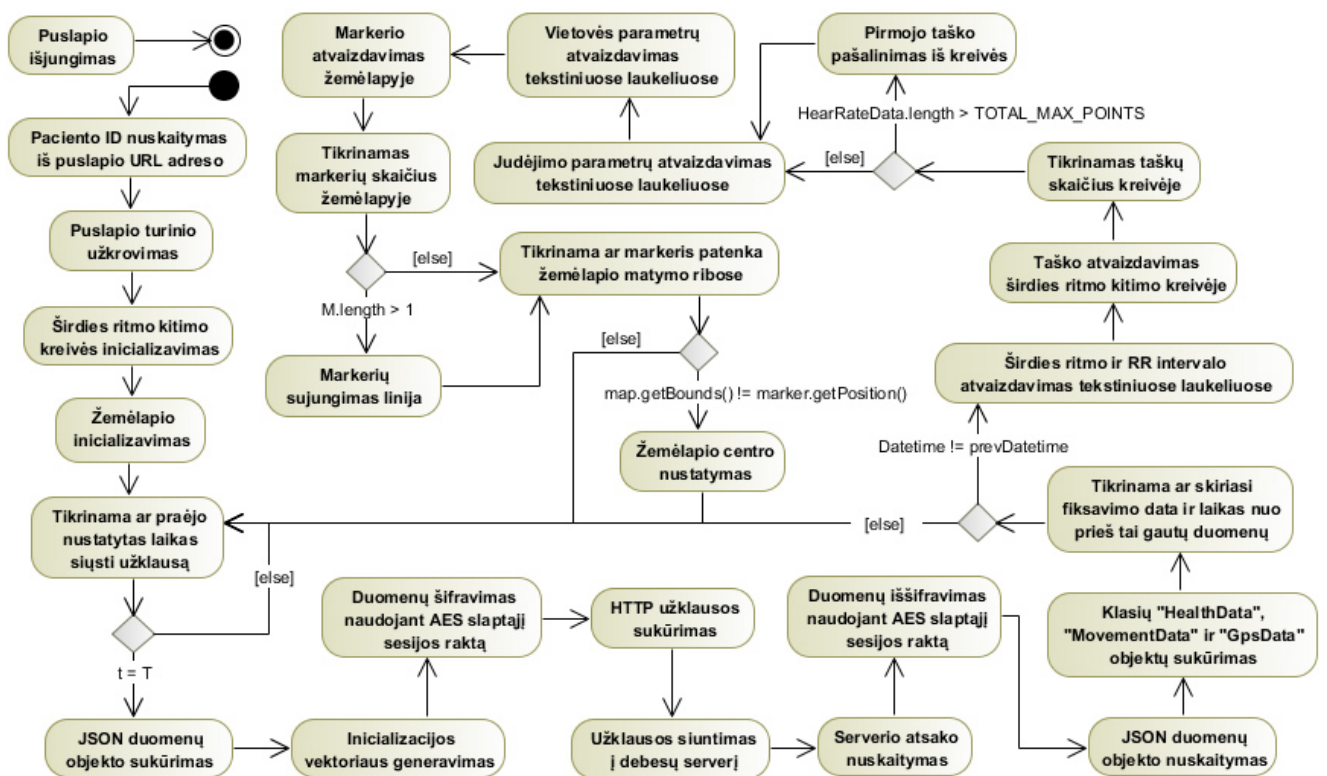
Medicinos personalas internetiniame puslapyje gali pateikti medicininės išvadas apie paciento sveikatos būklę. Jos saugomos debesų serveryje įdiegtoje duomenų bazėje ir siunčiamos pacientui į mobiliąją programą. 28 pav. pateikiamas puslapio langas, kuriame atvaizduojamas sąrašas su pasirinkto paciento medicininėmis išvadomis. Sąrašo viršuje atvaizduojami naujausi įrašai. Pasirinkus norimą įrašą pagal datą ir nuspaudus mygtuką „Peržiūrėti“ atidaromas puslapio langas, kuriame pateikiamas aprašymas apie paciento sveikatos būklę. Gydytojas gali peržiūrėti ir kitų gydytojų pateiktas medicininės išvadas, tačiau redaguoti ir ištrinti gali tik savo.

Nr.	Pavadinimas	Sukūrimo data	Atnaujinimo data	Gydytojas	Veiksmai
1	Reikalingi tyrimai	2017-03-11 10:14:06	2017-03-11 10:16:38	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
2	Širdies sutrikimų neužfiksuota	2017-02-16 09:13:09	2017-02-16 09:23:55	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
3	Pacientas sveikas	2017-02-08 15:30:10	2017-02-08 15:32:52	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
4	Sveikatos sutrikimų nerasta	2017-02-01 13:10:25	2017-02-01 13:15:37	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
5	Širdies ritmo sutrikimų nėra	2017-01-25 10:30:44	2017-01-25 10:31:45	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
6	Širdies veikla normali	2017-01-17 16:20:05	2017-01-17 16:25:00	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti
7	Profilaktinis patikrinimas	2017-01-11 12:20:01	2017-01-11 12:20:01	Jonas Jonaitis	Peržiūrėti Redaguoti Ištrinti

28 pav. Gydytojų medicininės išvados apie paciento sveikatos būklę

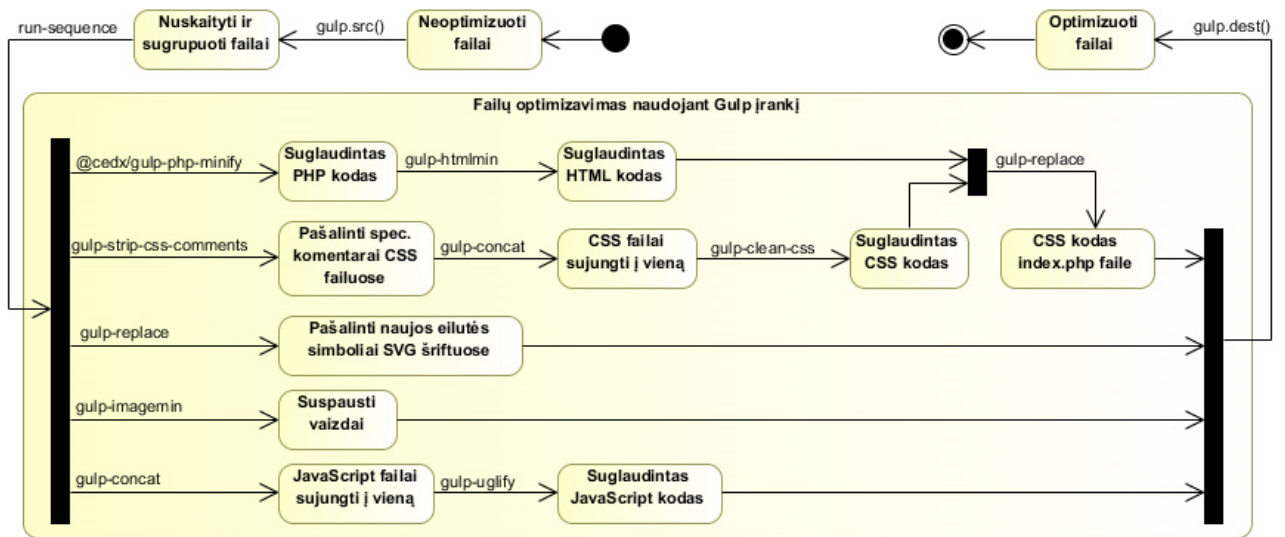
Internetinis puslapis sukurtas naudojant vieno puslapio aplikacijos (angl. *Single-page application*, SPA) metodiką. Puslapis nėra perkraunamas, atnaujinama tik kintanti turinio dalis. Kai vartotojas prisijungia prie puslapio parsisiunčiami CSS, HTML, JavaScript failai, užkraunama antraštė (angl. *header*), šoninė juosta (angl. *sidebar*) su valdymo įrankiais ir blokas su puslapio turiniu (angl. *content*). Pasirinkus valdymo įrankį keičiamas puslapio URL adresas, o pagal jį užkraunamas kintantis turinys. Iš naujo nesiunčiami tie patys failai, neužkraunama antraštė, šoninė juosta. Naudojant SPA sumažinamas puslapio užkrovimo laikas. 29 pav. pavaizduota veiklos diagrama, kurioje pateiktas fiksuojamų parametrų atvaizdavimo algoritmas internetiniame puslapyje. Iš URL adreso nuskaitymas paciento ID. Užkraunamas kintamas puslapio blokas, kuriame pateikiami tekstiniai laukeliai skirti atvaizduoti fiksuojamus parametrus. Inicializuojamas žemėlapis ir širdies ritmo kitimo kreivė. Iš paciento ID, paskyros ID sukuriama JSON duomenų objektas. Sugeneruojamas inicializacijos vektorius (IV). Duomenys užšifruojami naudojant AES, sesijos slaptažodį ir IV. Sukuriama HTTP

užklausa, kuri išsiunčiama į debesų serverį. Gautas serverio atsakas nuskaitomas ir duomenys iššifruojami naudojant AES. JSON duomenų objektas konvertuojamas į masyvą raktas-reikšmė pavidalu, iš kurio sukuriama klasių *HealthData*, *MovementData* ir *GpsData* objektai. Duomenys puslapyje atnaujinami tik tokiu atveju jeigu skiriasi fiksavimo data ir laikas nuo prieš tai gautų duomenų. Tekstiniuose laukeliuose atvaizduojami sveikatos parametrai. Širdies ritmo kreivėje atvaizduojamas naujas taškas. Jeigu taškų skaičius viršija nurodytą ribą pašalinamas pirmasis kreivės taškas. Atvaizduojami judėjimo ir vietovės parametrai. Pagal platumos ir ilgumos koordinatas žemėlapyje atvaizduojamas markeris, kuris sujungiamas linija su prieš tai atvaizduotu markeriu. Jeigu markeris nepatenka į žemėlapio matymo ribą pakeičiamas jo centras. Algoritmas kartojamas, vėl tikrinama ar praėjo nustatytas laikas siųsti naują užklausa.



29 pav. Algoritmas fiksuojamų parametrų atvaizdavimui internetiniame puslapyje

Atliktas internetinio puslapio optimizavimas siekiant sumažinti jo užkrovimo laiką. Naudojant *Gulp* įrankį ir įskiepius (angl. *plugins*) sukurtas skriptas, kuris optimizuoja puslapio failus. UML būsenų diagrama (angl. *state machine diagram*) pateikiama 30 pav. Joje pavaizduotas internetinio puslapio failų optimizavimo procesas. Jo metu siekiama: sumažinti failų skaičių – į serverį siunčiama mažiau HTTP užklauso, sumažinti failų užimamą vietą – failai greičiau parsisiunčiami iš serverio. Failai nuskaitomi ir sugrupuojami pagal plėtinį (angl. *extension*). PHP failuose suglaudinas PHP ir HTML kodas. Pašalinami komentarai, naujos eilutės simboliai, atitraukimai (angl. *tabs*) ir t. t. CSS failuose pašalinami specialūs komentarai, failai sujungiami į vieną, kodas suglaudinas. Išorinio (angl. *external*) CSS failo kodas perkeliamas į *index.php* failo *head* dalį kaip vidinis (angl. *internal*).



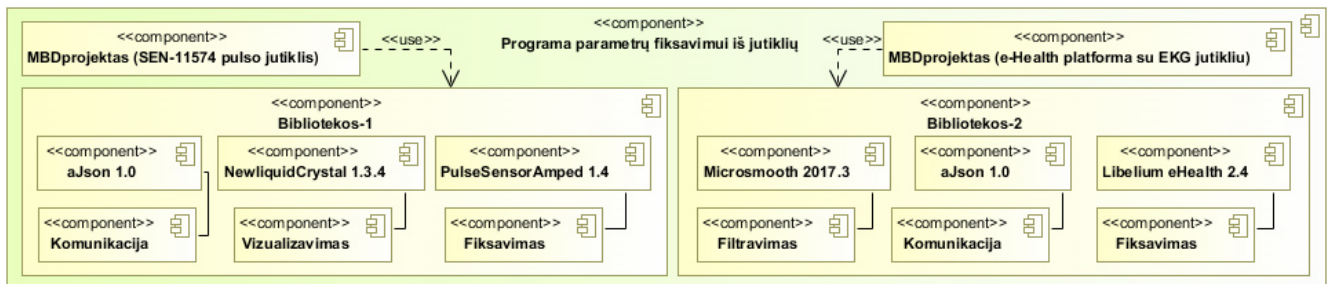
30 pav. Internetinio puslapio failų optimizavimo procesas

Atliekamas SVG šriftų (angl. *fonts*) optimizavimas ir vaizdų suspaudimas (angl. *compression*). JavaScript failai sujungiami į vieną, suglaudinas kodas. Optimizuoti failai išsaugomi. Internetinio puslapio uzkrovimo laikas prieš ir po optimizacijos pateikiami eksperimentinėje dalyje.

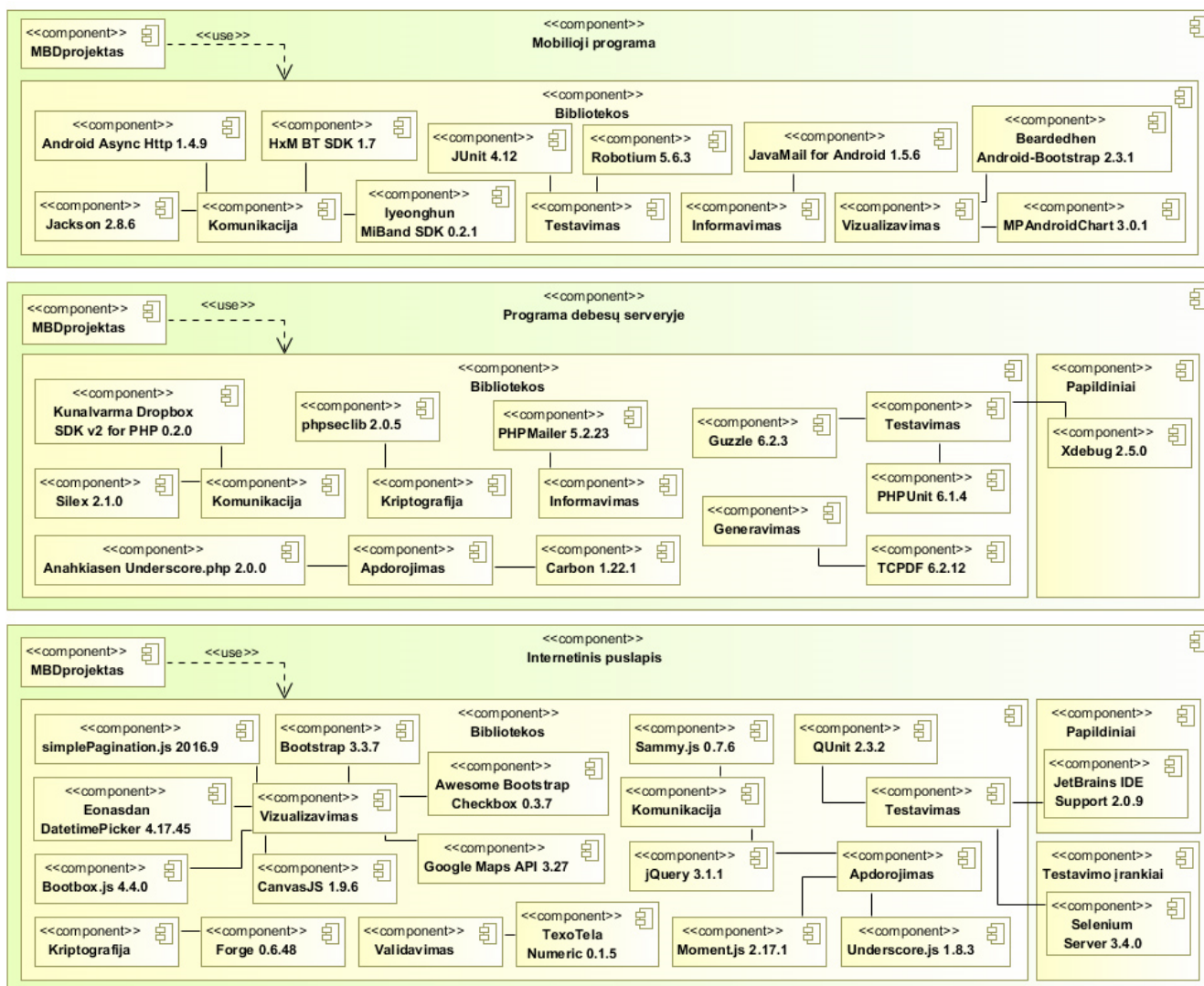
2.4. Sistemos testavimas ir diegimas

Sistemos kūrimo metu atliktas programinio kodo versijavimas naudojant Git įrankį. Kuriant sistemą parašyti vienetų (angl. *unit*) ir vartotojo sąsajos (angl. *user interface*) testai. Jie vykdomi prieš įkeliant programinio kodo pakeitimus (angl. *commit*) į repozitoriją (angl. *repository*), prieš diegiant sistemą į galutiniams vartotojams skirtą aplinką (angl. *production environment*). Mobiliajai programai parašyti instrumentiniai vienetų (angl. *instrumented unit*) testai naudojant JUnit programinę biblioteką, o vartotojo sąsajos testai – Robotium biblioteką. Debesų serveryje įdiegtai programai parašyti vienetų testai naudojant PHPUnit. REST API testavimui panaudota Guzzle biblioteka – HTTP užklausų siuntimui. PHP programinio kodo aprėptis testais (angl. *code coverage*) lygi 100%. Kodo derinimas (angl. *debug*) atliktas naudojant Xdebug papildinį (angl. *extension*). Internetiniam puslapiui parašyti vienetų testai naudojant QUnit biblioteką, o vartotojo sąsajos testai – Selenium Server įrankį.

UML komponentų diagrama (angl. *component diagram*) pateikiama 31 pav. ir 32 pav. Joje pavaizduotos programinės bibliotekos, kurios panaudotos sistemos įgyvendinimui.



31 pav. Sistemos įgyvendinimui panaudotos programinės bibliotekos (1)



32 pav. Sistemos įgyvendinimui panaudotos programinės bibliotekos (2)

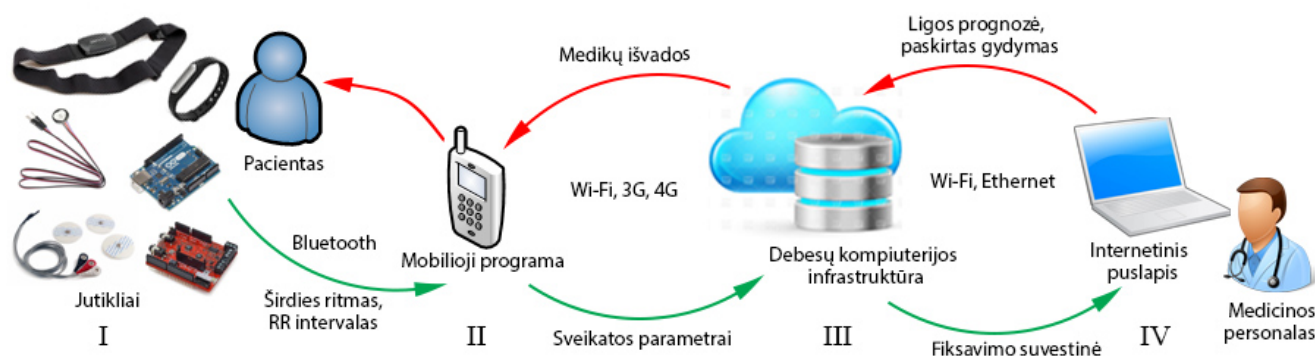
Programinių bibliotekų paskirtis sukurtoje sistemoje pateikiama 10 lentelėje.

10 lentelė. Programinių bibliotekų paskirtis sukurtoje sistemoje

Biblioteka	Paskirtis
aJson	iš užfiksuotų parametru generuojamas JSON duomenų objektas
NewliquidCrystal	atvaizduojamas širdies ritmas ir RR intervalas LCD ekrane
PulseSensorAmped	vykdomas R bangų aptikimas, RR intervalo nustatymas, širdies ritmo skaičiavimas
Microsmooth	filtruojamas užfiksuotas elektrokardiogramos signalas
Libelium eHealth	iš EKG jutiklio fiksuojamas elektrokardiogramos signalas, kuris iš e-Health platformos perduodamas į Arduino valdiklių plokštę
Android Async Http	siunčiamos HTTP užklausos į debesų serverį, sukurtas REST API klientas
HxM BT SDK	fiksuojami sveikatos ir judėjimo parametrai iš Zephyr HxM BT juosmens diržo
Iyeonghun MiBand SDK	fiksuojami sveikatos ir judėjimo parametrai iš Xiaomi Mi Band 1s apyrankės
Jackson	siunčiami užklausos parametrai paverčiami JSON formatu, gautas atsakas JSON formatu konvertuojamas į duomenis raktas-reikšmė pavidalu
JavaMail for Android	suinteresuoti asmenys informuojami el. laišku apie paciento širdies ritmo nukrypimus
Beardedhen Android-Bootstrap	programos lange, kuriame pateikiamas medicininių išvadų sąrašas, atvaizduojamas puslapiavimas
MPAndroidChart	atvaizduojamas širdies ritmas, RR intervalas, momentinis greitis ir ritmas kreivėse
Kunalvarma Dropbox SDK v2 for PHP	į Dropbox serverį siunčiamos duomenų bazės atsarginės kopijos
Silex	sukurtas REST API serveris
phpseclib	vykdomas siunčiamos informacijos šifravimas RSA ir AES algoritmais, viešojo ir

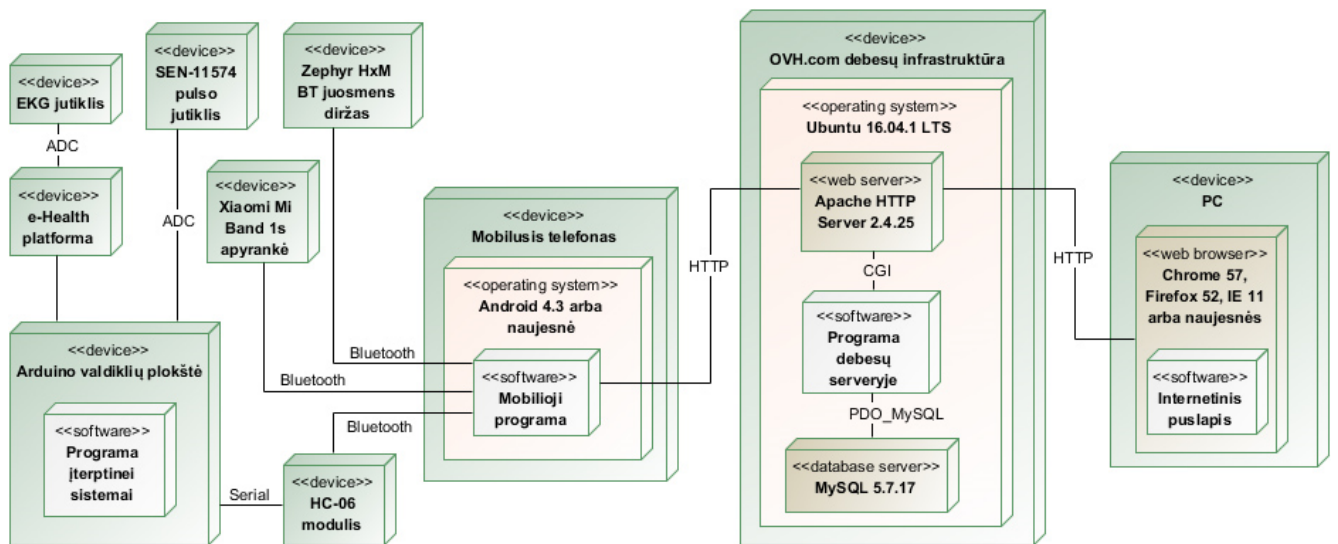
	privačiojo raktų generavimas
Anahkiasen Underscore.php	generuojant fiksavimo suvestinę sugrupuojami sveikatos ir judėjimo parametrai pagal datas
Carbon	vykdomas datų apdorojimas fiksavimo suvestinės generavimo metu
PHPMailer	į paciento el. paštą siunčiamas laiškas su prisegtu PDF dokumentu, kuriame pateikiama savaitinė fiksavimo suvestinė
TCPDF	vykdomas fiksavimo suvestinės PDF dokumento generavimas
simplePagination.js	internetiniame puslapyje, kuriame pateikiamas medicininių išvadų sąrašas, atvaizduojamas puslapiavimas
Eonasdan DatetimePicker	fiksavimo suvestinės puslapyje atvaizduojami datų pasirinkimo laukai
Bootbox	atvaizduojami išpėjiamieji iššokantys langai
Bootstrap	šoninės juostos, mygtukų, tekstinių laukelių, lentelių dizainas
CanvasJS	atvaizduojamas širdies ritmas, RR intervalas kreivėse, nueitas atstumas, žingsnių skaičius – stulpelinėse diagramose
Awesome Bootstrap Checkbox	fiksavimo suvestinės puslapyje atvaizduojamas žymimasis laukelis (angl. <i>checkbox</i>)
Google Maps API	atvaizduojamas Google žemėlapis, buvimo vieta
TexoTela Numeric	patikrinamas tekstinis laukelis, kad būtų galima įvesti tik skaičius
Forge	vykdomas siunčiamos informacijos šifravimas RSA ir AES algoritmais, sesijos slaptojo rakto ir IV generavimas
Sammy.js	nuskaitomas puslapio URI adresas ir jo parametrai, kurie skirti suformuoti užklausą
jQuery	į debesų serverį siunčiamos HTTP užklauskos panaudojant AJAX technologiją, sukurtas REST API klientas, apdorojamas HTML DOM (angl. <i>Document Object Model</i>)
Moment.js	vykdomas datų apdorojimas
Underscore.js	masyve ieškomas objektas pagal ID siekiant nustatyti, kuris gydytojas pateikė medicininę išvadą apie paciento sveikatos būklę

Sukurta nuotolinė sveikatos stebėjimo sistema su grįžtamuju ryšiu, kuri leidžia medikams stebėti pacientų širdies ir kraujagyslių veiklą. Sistemos raiškūs pavėikslėlis pateikiamas 33 pav. Ją sudaro keturi pagrindiniai komponentai. Jutikliais fiksuojami sveikatos, judėjimo parametrai (I). Užfiksuoti parametrai iš mobiliosios programos (II) siunčiami į debesų infrastruktūrą (III). Medicinos personalui internetiniame puslapyje (IV) pateikiama parametų fiksavimo suvestinė. Medikų išvados, nurodymai siunčiami pacientui.



33 pav. Nuotolinė širdies ir kraujagyslių ligų stebėjimo sistema su grįžtamuju ryšiu

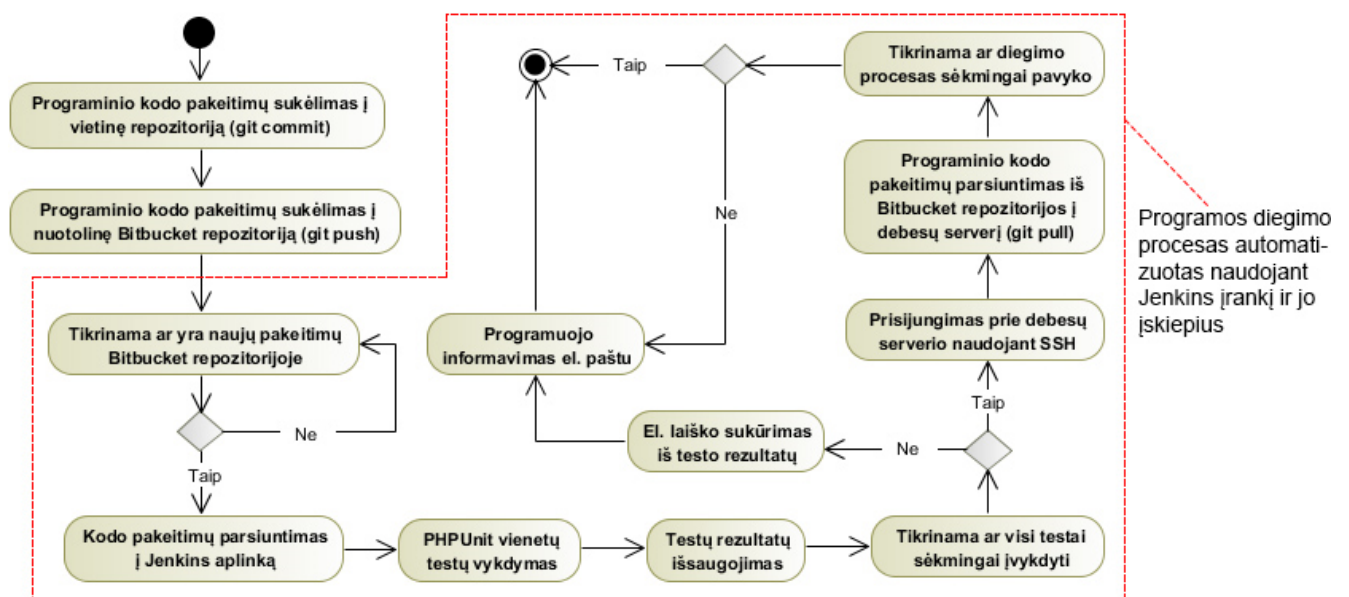
UML diegimo diagrama (angl. *deployment diagram*) pateikiama 34 pav. Joje pavaizduotas sistemos diegimas į galutiniams vartotojams skirtą aplinką. Įterptinei sistemai sukurta programa įdiegta į Arduino valdiklių plokštę, o mobilioji programa – į telefoną su Android operacine sistema, kuriai reikalinga 4.3 arba naujesnė versija.



34 pav. Sistemos diegimas į galutiniams vartotojams skirtą aplinką

Serveryje veikianti programa įdiegta į OVH.com debesų kompiuterijos infrastruktūrą. Pasirinkta Ubuntu 16.04 operacinė sistema. Testavimo metu patikrinta, kad internetiniame puslapyje pateikta informacija korektiškai atvaizduojama Chrome 57, Firefox 52 ir Internet Explorer (IE) 11 naršyklėse. Naudojant W3C Markup klaidų tikrintuvą nustatyta, kad internetinio puslapio CSS ir HTML aprašai atitinka W3C standartus. Google PageSpeed Insights testavimo rezultatai pateikė, kad internetinis puslapis tenkina 99% rekomendacijų skirtų puslapio optimizavimui.

Šiame darbe sukurtas algoritmas leidžiantis automatiškai įdiegti programą į debesų serverį (35 pav.). Jo kūrimui panaudotas Jenkins įrankis. Programuotojas kodo pakeitimus sukelia į vietinę repozitoriją, o po to – į nuotolinę Bitbucket. Jenkins įrankis į savo aplinką parsisiunčia programinio kodo pakeitimus, įvykdo vienetų testus ir išsaugo rezultatus. Jeigu visi testai sėkmingai įvykdyti – prisijungia prie debesų serverio ir įdiegia programos pakeitimus, priešingu atveju – informuoja programuotoją el. paštu, kad programiniame kode yra klaidų.



35 pav. Algoritmas automatizuotam programos diegimui į debesų serverį

III. EKSPERIMENTINĖ DALIS

Atliktas sukurtos sistemos keturių pagrindinių komponentų verifikavimas:

- norint nustatyti su kuriuo signalo filtravimo algoritmu gaunama tikslesnė širdies ritmo reikšmė (lyginama su sertifikuoto medicininio prietaiso rodikliu) atliktas signalo filtravimų algoritmų lyginamasis tyrimas EKG fiksavimo metu;
- norint nustatyti ar mobilioji programa reikalauja daug skaičiavimo, atminties, tinklo ir energijos resursų atliktas telefono sąnaudų tyrimas mobiliosios programos veikimo metu;
- norint nustatyti maksimalų vartotojų skaičių, kuris gali prisijungti prie sistemos atliktas debesų serverio atsako laiko tyrimas esant skirtingai apkrovai;
- norint nustatyti per kiek laiko vartotojui pateikiama informacija internetiniame puslapyje atliktas jo užkrovimo laiko tyrimas prieš ir po optimizacijos.

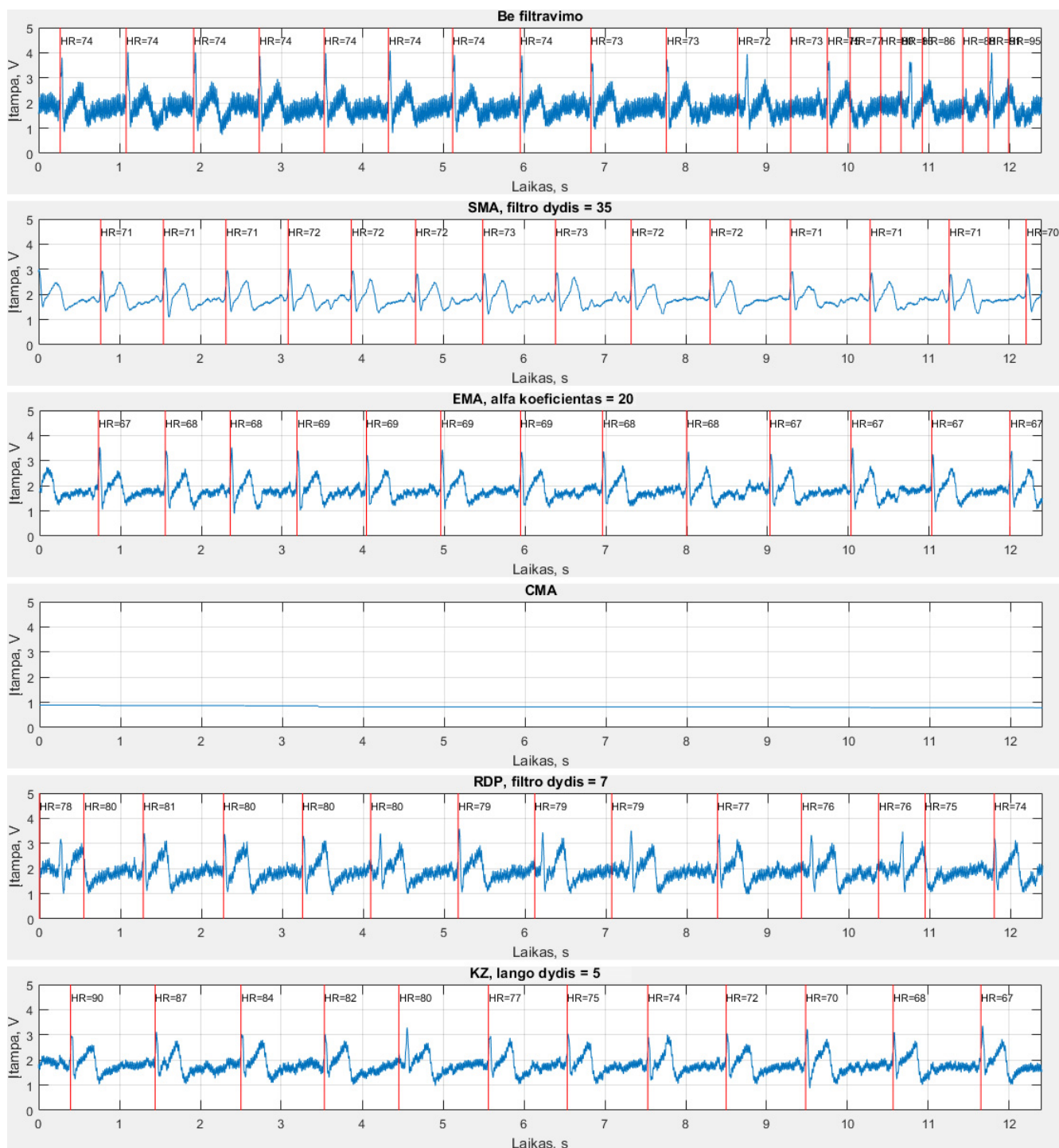
3.1. Signalo filtravimo algoritmų lyginamasis tyrimas EKG fiksavimo metu

Naudojant e-Health platformą su EKG jutikliu užfiksuojamas neapdorotas (angl. *raw*) elektrokardiogramos (EKG) signalas. Sukurtas algoritmas, kuris signale aptinka R bangas. Pagal užfiksuotus R bangų laiko momentus apskaičiuojamas RR intervalas ir širdies ritmas (HR). Nuo EKG signalo kokybės priklauso širdies ritmo tikslumas. Atliktas signalo filtravimo algoritmų lyginamasis tyrimas EKG fiksavimo metu. Pasirinkti penki filtravimo algoritmai:

- paprastasis slenkančio vidurkio (angl. *simple moving average*, SMA);
- eksponentinis slenkančio vidurkio (angl. *exponential moving average*, EMA);
- kaupiamasis slenkančio vidurkio (angl. *cumulative moving average*, CMA);
- Ramer–Douglas–Peucker algoritmas (RDP);
- Kolmogorov–Zurbenko filtras (KZ).

Tyrimo metu keičiami filtrų parametrai: SMA filtro dydis lygus 15, 35, 55 ir 75, EMA α koeficientas – 10, 20, 30 ir 40, RDP filtro dydis – 7 ir 14, KZ lango dydis – 3, 5 ir 7. Pagal apdorotą EKG signalą nustatomas širdies ritmas, kurio reikšmė lyginama su sertifikuoto medicininio prietaiso Microlife BP A100 gautu rodikliu. EKG signalo filtravimas, R bangų aptikimas, RR ir HR skaičiavimas vykdomas Arduino Uno valdiklių plokštėje, o ne kompiuteryje naudojant simuliaciją. Jame vykdomas tik parametrų vizualizavimas. Užfiksuoti duomenys nuosekliaja (angl. *serial*) sąsaja iš valdiklių plokštės perduodami į kompiuterį. Naudojant sukurtą PowerShell skriptą *.ps1* iš prievado (angl. *port*) nuskaityti duomenys įrašomi į tekstinį failą *.txt*. Nuskaitoma 10000 įrašų. Gauti filtravimo rezultatai atvaizduojami naudojant Matlab programinę įrangą, keletas jų pateikiami 36 pav. X ašyje atvaizduojamas fiksavimo laikas, y ašyje – signalo įtampa. Raudona vertikalia linija pažymėtas laiko momentas, kada užfiksuota R banga. Prie linijos nurodytas apskaičiuotas širdies ritmas. 36 pav. viršuje

pateiktas neapdorotas EKG signalas, o kituose – signalas po filtravimo. Kadangi valdiklių plokštė apribota skaičiavimo ir atminties resursais vienu momentu signalas filtruojamas naudojant vieną algoritmą, užfiksavus parametrus pakeičiamas kitu. Tyrimas atliktas fiksuojant paciento EKG rambybės būsenoje.



36 pav. EKG signalo filtravimo rezultatai naudojant SMA, EMA, CMA, RDP ir KZ algoritmus

Naudojant SMA filtrą iš signalo pašalinti triukšmai, tačiau R bangos amplitudė sumažėjo, todėl kartais aptinkama T banga, vietoj R. Nustatyta, kad didinant filtro dydį triukšmų kiekis mažėja, bet taip pat mažėja R bangos amplitudė. CMA filtras EKG signalą visiškai nuslopino. RDP filtras pašalina nedidelį kiekį triukšmų. Nustatyta, kad didinant filtro dydį apdorojimo proceso trukmė ženkliai išauga.

Kai filtro dydis buvo lygus 14, 10000 įrašų surinkta per ~50 s, naudojant kitus filtrus per ~13 s. KZ filtras dalį triukšmų pašalina, tačiau R ir T bangų amplitudės tapo beveik lygios. Geriausi filtravimo rezultatai gauti naudojant EMA filtrą. Didžioji dalis triukšmų pašalinta, R bangos amplitudė didesnė už T bangos. Nustatyta, kad didinant koeficientą α iš signalo pašalinama mažiau triukšmų, o pernelyg sumažinus signalas slopinamas.

Norint įvertinti algoritmus nustatyta širdies ritmo reikšmė lyginama su sertifikuoto medicininio prietaiso gautu rodikliu. Širdies ritmas išmatuotas po tris kartus naudojant tą patį algoritmą ir jo parametą. Matavimai e-Health platforma su EKG jutikliu ir sertifikuotu prietaisu atlikti tuo pačiu metu. Iš viso atlikti 45 matavimai. Vidutinis širdies ritmo tikslumas skaičiuojamas pagal formules:

$$D_1 = |HRj_1 - HRs_1|, \quad D_2 = |HRj_2 - HRs_2|, \quad D_3 = |HRj_3 - HRs_3| \quad (1)$$

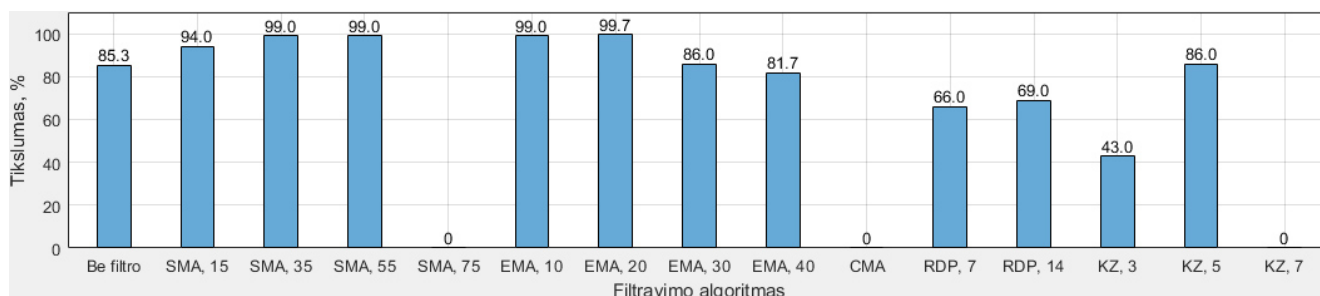
$$\text{jei } D_1 > 100, \text{ tai } D_1 = 100, \quad \text{jei } D_2 > 100, \text{ tai } D_2 = 100, \quad \text{jei } D_3 > 100, \text{ tai } D_3 = 100 \quad (2)$$

$$T_1 = 100 - D_1, \quad T_2 = 100 - D_2, \quad T_3 = 100 - D_3 \quad (3)$$

$$T_{vid} = \text{vidurkis}(T_1, T_2, T_3) \quad (4)$$

Kur, $HRj_{1,2,3}$ – širdies ritmo matavimai e-Health platforma su EKG jutikliu, $HRs_{1,2,3}$ – širdies ritmo matavimai sertifikuotu prietaisu, $D_{1,2,3}$ – matavimų absoliutus skirtumas, $T_{1,2,3}$ – širdies ritmo tikslumas konkretaus matavimo metu, T_{vid} – vidutinis širdies ritmo tikslumas, nuo 0 iki 100%.

Tyrimo metu gautas vidutinis širdies ritmo fiksavimo tikslumas naudojant skirtingus filtravimo algoritmus ir jų parametrų reikšmes, pateikiamas 37 pav.



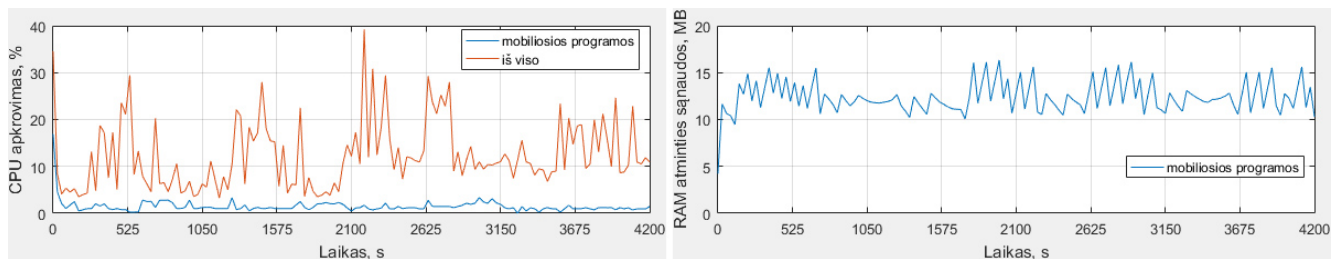
37 pav. Vidutinis širdies ritmo fiksavimo tikslumas naudojant skirtingus filtravimo algoritmus

X ašyje pateikiama filtravimo algoritmo santrumpa ir pasirinkta jo parametro reikšmė, o y ašyje – tikslumas. Geriausias vidutinis širdies ritmo fiksavimo tikslumas gautas, kai EKG signalo filtravimui pasirinktas EMA algoritmas ir α koeficientas lygus 20. Naudojant EMA algoritmą pasiektas apie 99% tikslumas, širdies ritmo nuokrypis nuo medicininio prietaiso rodmenų lygus nuo 0 iki 2 dūžių.

3.2. Telefono resursų, tinklo ir energijos sąnaudų tyrimas mobiliosios programos veikimo metu

Atliktas tyrimas siekiant nustatyti telefono resursų, tinklo ir energijos sąnaudas sukurtos mobiliosios programos veikimo metu. Tyrimo metu Zephyr HxM BT juosmens diržas nuolatos Bluetooth ryšiu į mobiliąją programą siuntė fiksuojamus sveikatos ir judėjimo parametrus. Mobilioji

programa fiksavo asmens buvimo vietos koordinates naudojant GPS. Užfiksuoti sveikatos, judėjimo ir vietovės parametrai buvo siunčiami į debesų serverį kas 10 s naudojant „Ežio“ mobiliųjį internetą. Mobilioji programa įdiegta Samsung Galaxy J7 (J710FN) mobiliajame telefone. Tyrimas atliktas mobiliajai programai be pertraukos veikiant 1h 10min. Telefono CPU apkrovimo ir RAM atminties sąnaudų kitimas mobiliosios programos veikimu metu pateikiamas 38 pav.



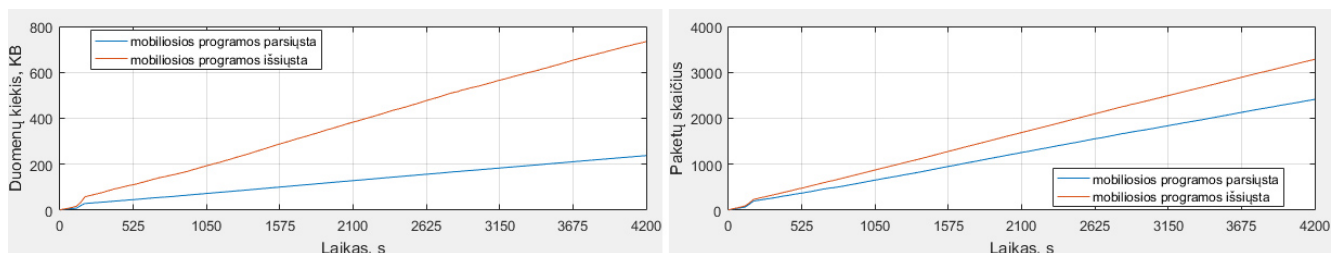
38 pav. Telefono resursų sąnaudų kitimas mobiliosios programos veikimo metu

Pradžioje mobiliosios programos CPU apkrovimas lygus apie 17%. Tada buvo prisijungta prie sistemos, susietas jutiklis su telefonu, pradėtas sveikatos, judėjimo ir vietovės parametų fiksavimas. Įvykdžius programos inicializaciją jos CPU apkrovimas nukrito iki 2%, kuris visą programos veikimo laikotarpį išliko panašus. Mobiliosios programos RAM atminties sąnaudos kito nuo 10 iki 15 MB. Atminties sąnaudoms pakilus daugiau kaip 15 MB automatiškai įvykdomas šiukšlių surinkimas (angl. *garbage collection*). Jo metu iš atminties ištrinami programos kintamieji, klasių objektai ir t. t. Mobiliosios programos vidutinis CPU apkrovimas ir vidutinės RAM atminties sąnaudos pateikiamos 11 lentelėje.

11 lentelė. Telefono resursų sąnaudos mobiliosios programos veikimo metu

Nr.	Parametras	Reikšmė
1	Trukmė, s	4200
2	Vidutinis CPU apkrovimas iš viso, %	12.3
3	Mobiliosios programos vidutinis CPU apkrovimas, %	1.5
4	Mobiliosios programos vidutinės RAM atminties sąnaudos, MB	12

Iš tyrimo rezultatų nustatyta, kad mobilioji programa nereikalauja daug skaičiavimo ir atminties resursų. Programos vidutinės sąnaudos: 1.5% CPU apkrovimas ir 12 MB RAM. Tinkle perduotų duomenų kiekio ir paketų skaičiaus kitimas mobiliosios programos veikimu metu pateikiamas 39 pav.



39 pav. Telefono tinklo sąnaudų kitimas mobiliosios programos veikimo metu

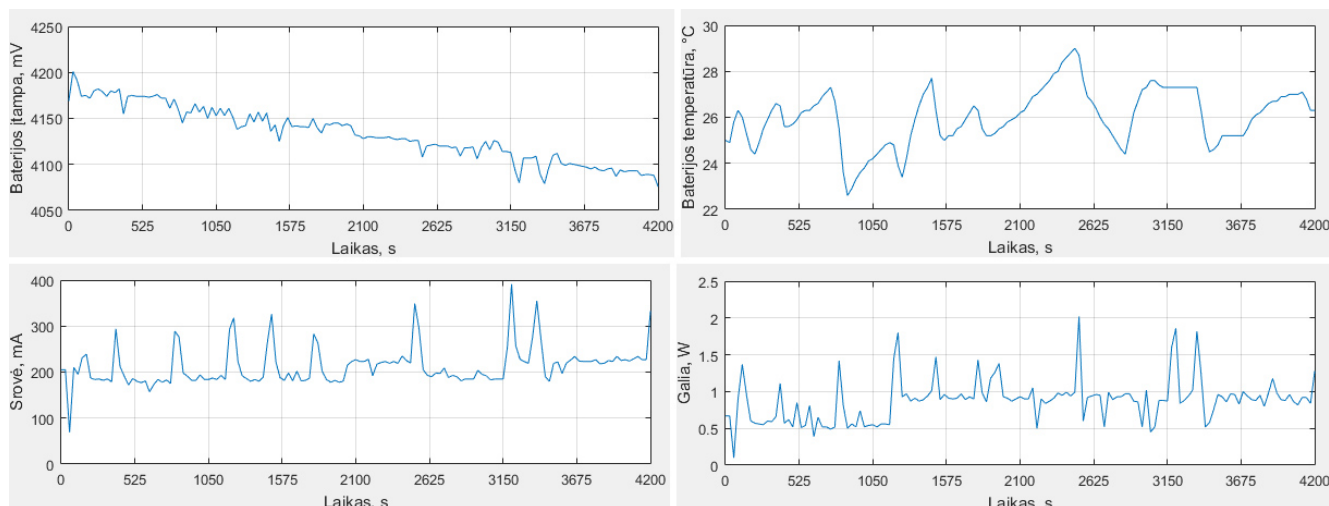
Prisijungimo prie sistemos metu, kai iš debesų serverio parsųstami paskyros duomenys, tinkle perduodamų duomenų kiekis lygus apie 31 KB, o paketų skaičius – 135. Įvykdžius programos

inicializaciją į debesų serverį nuolatos siunčiamos užklauskos su fiksuojamais parametrais. Tada tinkle perduodamų duomenų kiekis lygus apie 7 KB, o paketų skaičius – 42. Programos išsiųstų duomenų kiekis didesnis negu parsųstų. Mobiliosios programos tinkle perduotų duomenų kiekis ir paketų skaičius per jos veikimo laiką pateikiamas 12 lentelėje.

12 lentelė. Telefono tinklo sąnaudos mobiliosios programos veikimo metu

Nr.	Parametras	Reikšmė
1	Trukmė, s	4200
2	Mobiliosios programos parsųstų duomenų kiekis, KB	238
3	Mobiliosios programos išsiųstų duomenų kiekis, KB	735
	Iš viso:	973
4	Mobiliosios programos parsųstų paketų skaičius	2413
5	Mobiliosios programos išsiųstų paketų skaičius	3285
	Iš viso:	5698

Iš tyrimo rezultatų nustatyta, kad tinkle siunčiamų duomenų kiekis, mobiliajai programai veikiant nuolatos 30 dienų lygus apie 586 MB. Telefono baterijos įtampos ir temperatūros, srovės bei galios kitimas pateikiamas 40 pav.



40 pav. Telefono energijos sąnaudų kitimas mobiliosios programos veikimo metu

Baterijos įtampa per mobiliosios programos veikimo laikotarpį sumažėjo nuo 4200 iki 4076 mV. Baterijos temperatūra kito nuo 23 iki 29°C. Srovė svyravo ties 180 mA, o kartais esant didesniai CPU apkrovimui – 340 mA. Galia kito nuo 0.5 iki 1.8 W. Vidutinės telefono energijos sąnaudos mobiliosios programos veikimu metu pateikiamos 13 lentelėje.

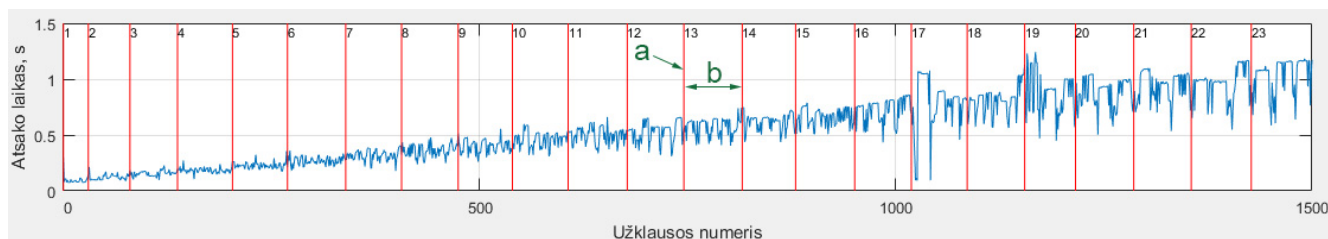
13 lentelė. Telefono energijos sąnaudos mobiliosios programos veikimo metu

Nr.	Parametras	Reikšmė
1	Trukmė, s	4200
2	Baterijos įkrovimo lygio sumažėjimas, %	7
3	Baterijos vidutinė temperatūra, °C	26
4	Baterijos vidutinė įtampa, mV	4133
5	Vidutinė srovė, mA	213
6	Vidutinė galia, W	0.88

Iš tyrimo rezultatų nustatyta, kad naudojant mobilųjį telefoną turintį bateriją, kurios talpa 3300 mAh, įkrauta 100%, mobiliąja programa be pertraukos galima naudotis apie 15h 30min.

3.3. Debesų serverio atsako laiko tyrimas esant skirtingam vartotojų skaičiui

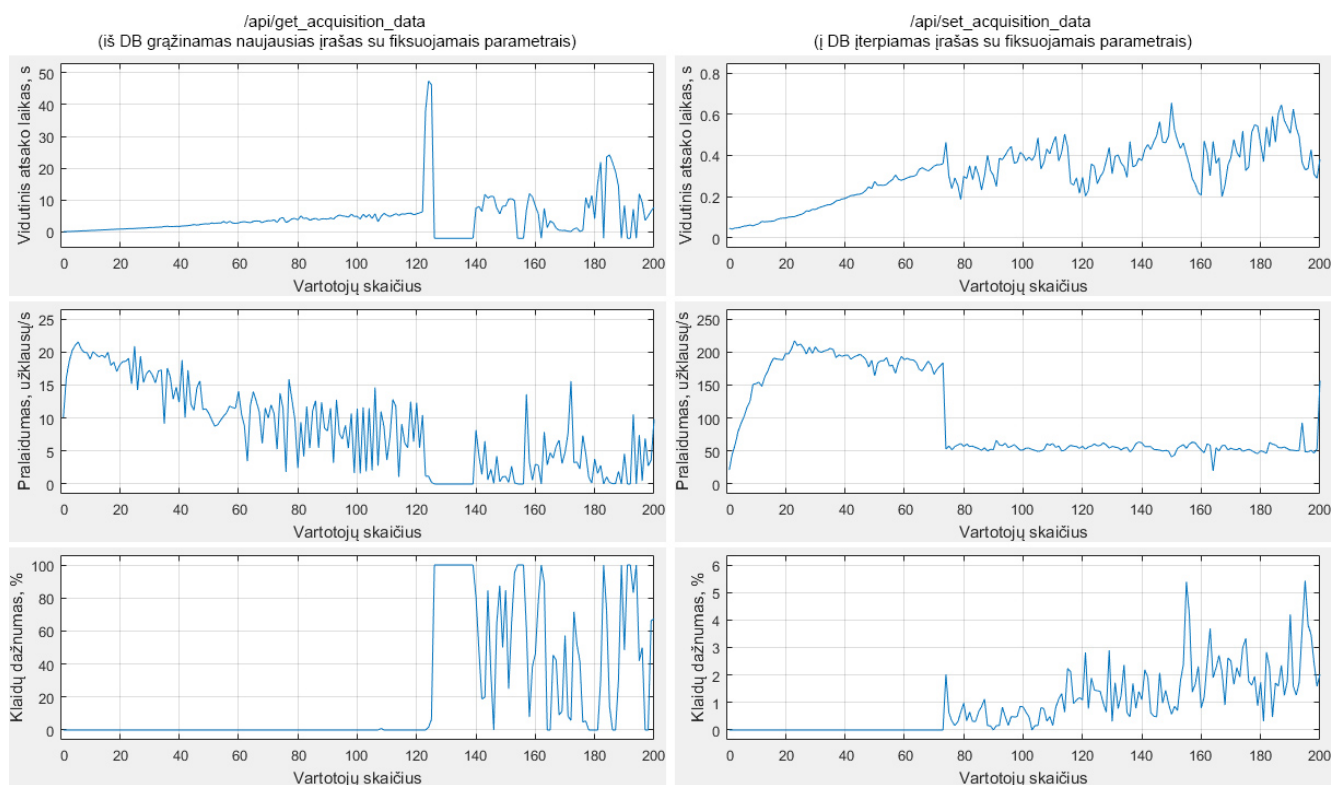
Atliktas tyrimas siekiant nustatyti debesų serverio atsako laiką esant skirtingam vartotojų skaičiui. Tyrimui pasirinkti du REST API resursai: `/api/set_acquisition_data` (į DB įterpiamas įrašas su fiksuojamais parametrais) ir `/api/get_acquisition_data` (iš DB grąžinamas naujausias įrašas su fiksuojamais parametrais). Pasirinkti todėl, kad jie vykdomi nuolatos: mobilioji programa kas 10 s (5 s – esant širdies ritmo nukrypimams) siunčia fiksuojamus parametrus į debesų serverį, kurie DB išsaugomi, internetinis puslapis kas 5 s kreipiasi į serverį, kuris iš DB paima ir grąžina naujausią įrašą. Tyrimas atliktas naudojant JMeter 3.1 programinį įrankį. Tyrimo metu pasirinkta 200 vartotojų (angl. *threads, users*), kurie generuojami kas 4 s. Visų vartotojų sugeneravimo trukmė (angl. *ramp-up period*) 600 s. Kai vartotojas sugeneruojamas, jis siunčia užklausas į serverį kartu su kitais jau sugeneruotais vartotojais. Visų vartotojų užklausos siunčiamos be uždelsimo: išsiunčiama užklausa, gaunamas serverio atsakas, vėl siunčiama užklausa. Dalis tyrimo rezultatų pateikiami 41 pav., likusi – priede, a – sugeneruojamas 13-asis vartotojas, b – visi 13 vartotojų siunčia užklausas į serverį.



41 pav. Debesų serverio atsako laiko tyrimo rezultatai

Debesų serverio techniniai parametrai: 1x 2.4 GHz CPU, 2 GB RAM, 10 GB SSD, 100 Mbps duomenų srautas, lokalizacija – Strasbūras. Serverio vidutinio atsako laiko (angl. *average response time*), pralaidumo (angl. *throughput*) ir klaidos dažnumo (angl. *error rate*) priklausomybė nuo vartotojų skaičiaus pateikiama 42 pav. Neigiamas vidutinis atsako laikas reiškia, kad serveris neatsakinėjo į užklausas (gautos klaidos: *timeout, internal server error, failed to respond*).

REST API užklausa pagal kurią grąžinami fiksuojami parametrai reikalauja apie 10 kartų daugiau skaičiavimo resursų negu užklausa pagal kurią išsaugomi fiksuojami parametrai. Kai pagal užklausą grąžinami parametrai ir vartotojų skaičius didesnis negu 125, tada klaidų dažnumas lygus nuo 40 iki 100%, o kai pagal užklausą išsaugomi parametrai – klaidų dažnumas lygus nuo 2 iki 6%. Abiem atvejais simuliacija buvo vykdoma apie 10 min 20 s. Parametrų grąžinimo atveju iš viso išsiųsta 13563 užklausų, 9995 užklausoms grąžintas atsakas be klaidos (74%), o parametrų išsaugojimo atveju iš viso išsiųsta 124941 užklausų, 123729 užklausoms grąžintas atsakas be klaidos (99%).



42 pav. Vidutinio atsako laiko, pralaidumo, klaidų dažnumo priklausomybė nuo vartotojų skaičiaus

Tyrimo metu nustatyta, kad esant tam tikram vartotojų skaičiui, debesų serverio atsako laikas, pralaidumas ir klaidų dažnumas priklauso nuo serverio techninių parametrų, pasirinktų programinių bibliotekų, įrankių ir nuo to kaip suprogramuota REST API. Debesų serverio vidutinis atsako laikas ir pralaidumas pateikiamas 14 lentelėje. Maksimalus vartotojų skaičius nustatytas įvertinus klaidų dažnumą.

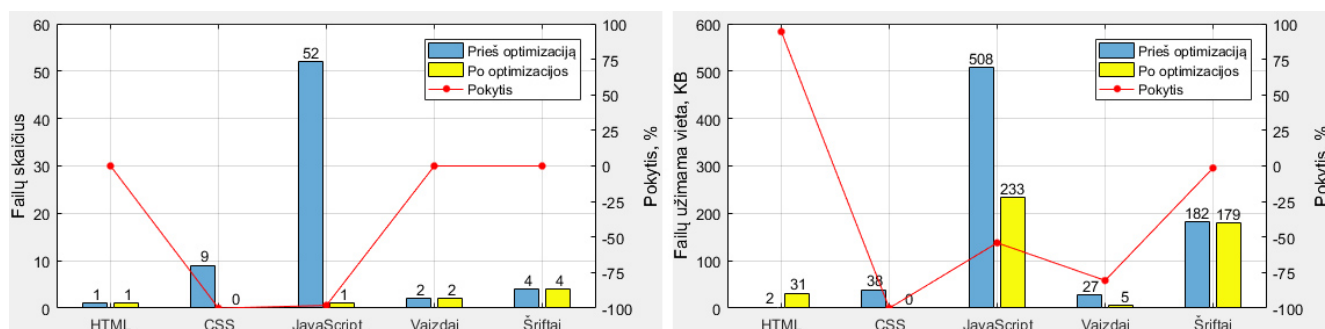
14 lentelė. Serverio vidutinis atsako laikas ir pralaidumas esant maksimaliam vartotojų skaičiui

Nr.	Parametras	Reikšmė	
		/api/get_acquisition_data iš DB grąžinamas naujausias įrašas su fiksuojamais parametrais	/api/set_acquisition_data į DB įterpiamas įrašas su fiksuojamais parametrais
1	Maksimalus vartotojų skaičius (klaidų dažnumas $\leq 3\%$)	122	154
2	Vidutinis atsako laikas, s (maksimalus vartotojų skaičius)	6.28	0.46
3	Pralaidumas, užklausų/s (maksimalus vartotojų skaičius)	10	60

Iš tyrimo rezultatų nustatyta, kad vienu metu prie sistemos gali prisijungti 122 vartotojai, kuriems internetiniame puslapyje atvaizduojami paciento fiksuojami parametrai. Tada debesų serverio vidutinis atsako laikas lygus 6.28 s, pralaidumas 10 užklausų/s, o klaidų dažnumas $\leq 3\%$. Naudojant mobiliąją programą vienu metu prie sistemos gali prisijungti 154 vartotojai ir siųsti fiksuojamus sveikatos, judėjimo ir vietovės parametrus. Tada debesų serverio vidutinis atsako laikas lygus 0.46 s, pralaidumas 60 užklausų/s, o klaidų dažnumas $\leq 3\%$.

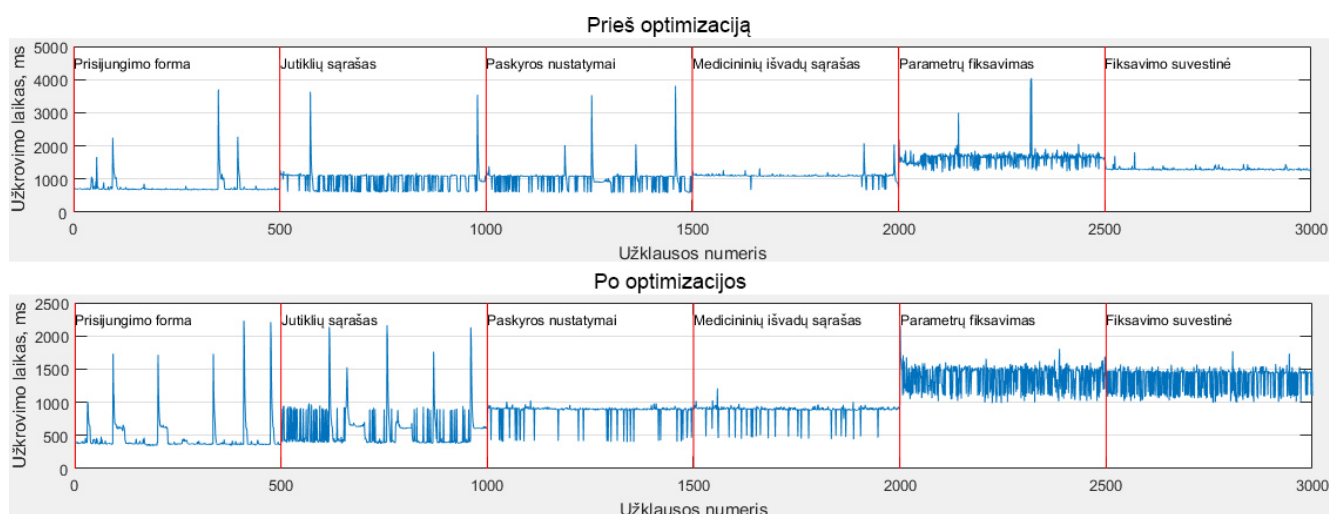
3.4. Internetinio puslapio užkrovimo laiko tyrimas prieš ir po optimizacijos

Atliktas tyrimas siekiant nustatyti internetinio puslapio užkrovimo laiką prieš ir po optimizacijos. Naudojant Gulp įrankį ir įskiepius sukurtas skriptas, kuris optimizuoja puslapio failus. Atskiri failai sujungiami į vieną failą – sumažinamas HTTP užklausų skaičius, kodas suglaudinas – sumažinama failų užimama vieta. 43 pav. pateikiamas internetinio puslapio failų skaičius ir jų užimama vieta prieš ir po optimizacijos. Linijinėje diagramoje pateikiamas procentinis pokytis. Po optimizacijos CSS kodas perkeltas į HTML antraštinę (angl. *head*) dalį.



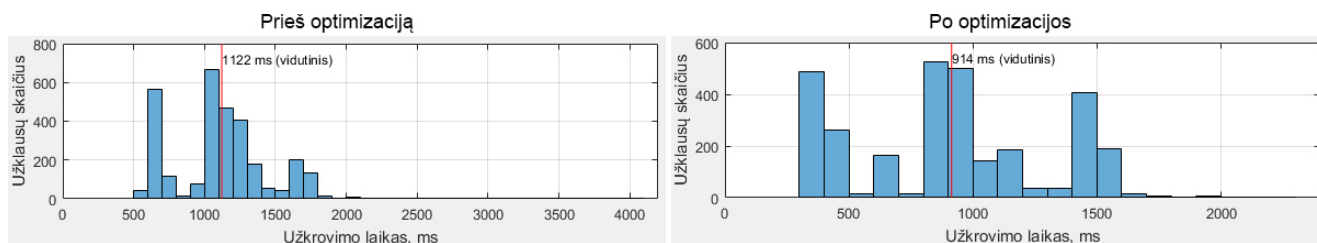
43 pav. Internetinio puslapio failų skaičius ir jų užimama vieta prieš ir po optimizacijos

Tyrimas atliktas naudojant Selenium 3.3.1 programinį įrankį. Sukurtas Java programinis skriptas, kuris automatiškai per internetinę naršyklę atidaro puslapį. Nurodytą kartų skaičių puslapis perkraunamas ir matuojamas jo užkrovimo laikas. Perkrovimas vykdomas atvaizdavirus puslapio turinį. Rezultatai surašomi į tekstinį failą ir apdorojami naudojant Matlab. Tyrimui atlikti pasirinkta Chrome 57 internetinė naršyklė. Jos susietumui su Selenium panaudotas Chrome Driver 2.29. Pasirinkti šeši internetinio puslapio langai, kuriuose pateikiama informacija: prisijungimo forma, jutiklių sąrašas, paskyros nustatymai, medicininių išvadų sąrašas, parametrų fiksavimas ir fiksavimo suvestinė. Kiekvienam iš jų 500 kartų išmatuotas užkrovimo laikas. Iš viso atlikta 3000 matavimų. Gauti rezultatai pateikiami 44 pav.



44 pav. Internetinio puslapio užkrovimo laiko rezultatai prieš ir po optimizacijos

Raudona vertikalia linija atskiriami puslapio langų užkrovimo laikai. Iš visų rezultatų apskaičiuojamas vidutinis internetinio puslapio užkrovimo laikas prieš ir po optimizacijos. 45 pav. pateikiama internetinio puslapio užkrovimo laiko histograma prieš ir po optimizacijos. Joje nurodoma kelis kartus puslapis buvo užkrautas per nurodytą laiko intervalą, kurie didinami kas 100 ms. Tyrimo metu po optimizacijos internetinis puslapis intervale nuo 300 iki 400 ms užkrautas 487 kartus.



45 pav. Internetinio puslapio užkrovimo laiko histograma prieš ir po optimizacijos

Iš tyrimo rezultatų nustatyta, kad internetinio puslapio vidutinis užkrovimo laikas po optimizacijos lygus 914 ms, prieš – 1122 ms, įvertinus šešis puslapio langus. Užkrovimo laiko pokytis -19%.

3.5. Tolimesnio sistemos tobulinimo, plėtojimo galimybės

Numatomi ateities darbai:

- Internetiniame puslapyje atvaizduojamo pacientų sąrašo rūšiavimas pagal klinikinės būklės sudėtingumo lygmenį. Sąrašo viršuje pateikiami pacientai, kurių klinikinės būklės sudėtingumo lygmens tikimybė yra didžiausia:
 - pradinės mokymo imties paieška „UCI Machine Learning Repository“ duomenų bazėje;
 - klasifikatorių (angl. *classifiers*) lyginamasis tyrimas;
 - pasirinkto klasifikatoriaus apmokymas su pradine mokymo imtimi – prižiūrimo mokymo (angl. *supervised learning*) įgyvendinimas;
 - mobiliosios programos lango sukūrimas, kuriame pacientas suveda klinikinės būklės sudėtingumo lygmens tikimybei nustatyti reikalingus parametrus (sistolinis ir diastolinis kraujospūdis, svoris, amžius, lygis, gliukozės ir cholesterolio kiekis kraujyje ir kt.);
 - klasifikatoriaus mokymasis pagal mediko pateiktas išvadas apie paciento sveikatos būklę – mokymo su paskatinimu (angl. *reinforcement learning*) įgyvendinimas. Jeigu medikas internetiniame puslapyje pažymi varnele, kad pacientui reikalingi detalesni tyrimai, jam išsiunčiamas informuojantis el. laiškas. Šis mediko veiksmas būtų kaip išorinis atsakas klasifikatoriui.

IŠVADOS

1. Atlikus mokslinių darbų analizę pasirinktos ir į vieną bendrą sistemą apjungtos technologijos: dėvimoji (Xiaomi apyrankė, Zephyr juosmens diržas), e-Health platforma, pulso jutiklis, Android OS, debesų kompiuterijos infrastruktūra, REST API, vieno puslapio aplikacija (SPA), Git ir TDD.
2. Suprojektuotas sveikatos stebėjimo sistemos prototipas, kuriam parašyti keturi programiniai algoritmai: širdies ritmo fiksavimo, užfiksuotų parametrų siuntimo, fiksavimo suvestinės PDF dokumento generavimo, užfiksuotų parametrų atvaizdavimo.
3. Sukurtas sveikatos stebėjimo informacinės sistemos su grįžtamuju ryšiu prototipas leidžiantis medikams nuotoliniu būdu stebėti pacientų širdies ir kraujagyslių veiklą. Jis susideda iš jutiklių – parametrų fiksavimui, mobiliosios programos – parametrų siuntimui, debesų serverio – duomenų apdorojimui, internetinio puslapio – informacijos atvaizdavimui.
4. Atliktas sistemos verifikavimas. EKG signalo filtravimui pasirinktas EMA algoritmas ($\alpha = 20$), širdies ritmo nuokrypis nuo sertifikuoto medicininio prietaiso rodmenų lygus nuo 0 iki 2 dūžių. Mobiliosios programos vidutinis CPU apkrovimas lygus apie 2%, RAM atminties sąnaudos – apie 12 MB, programai veikiant 30 dienų tinkle siunčiamų duomenų kiekis – apie 600 MB, jeigu baterija 3300 mAh įkrauta 100% programa be pertraukos galima naudotis apie 15h. Vienu metu prie sistemos gali prisijungti apie 120 vartotojų, esant klaidų dažnumui $\leq 3\%$, debesų serverio vidutinis atsako laikas lygus apie 6 s, o pralaidumas – apie 10 užklausų/s. Internetinio puslapio vidutinis užkrovimo laikas po optimizacijos lygus apie 900 ms, prieš – apie 1100 ms.

LITERATŪRA

1. Lietuvos respublikos švietimo ir mokslo ministerija. Nacionalinė mokslo programa „Sveikas senėjimas“. 2015, 18 p.
2. Yang J. J., Li J., Mulder J. Emerging information technologies for enhanced healthcare. *Computers in Industry*, 2015, volume 69, p. 3–11. ISSN 0166-3615.
3. Han N. D., Han L., Tuan D. M. A scheme for data confidentiality in Cloud-assisted Wireless Body Area Networks. *Information Sciences*, 2014, volume 284, p. 157–166. ISSN 0020-0255.
4. Kaur P. D., Chana I. Cloud based intelligent system for delivering health care as a service. *Computer Methods and Programs in Biomedicine*, 2014, volume 113, issue 1, p. 346–359. ISSN 0169-2607.
5. Andriopoulou F., Birkos K., Lymberopoulos D. P2Care: A dynamic peer-to-peer network for collaboration in personalized healthcare service delivery. *Computers in Industry*, 2015, volume 69, p. 45–60. ISSN 0166-3615.
6. Park K., Lim S. A multipurpose smart activity monitoring system for personalized health services. *Information Sciences*, 2015, volume 314, p. 240–254. ISSN 0020-0255.
7. Triantafyllidis A., Velardo C., Chantler T. A personalised mobile-based home monitoring system for heart failure: The SUPPORT-HF Study. *International Journal of Medical Informatics*, 2015, volume 84, issue 10, p. 743–753. ISSN 1386-5056.
8. Hussain A., Wenbi R., Lopes da Silva A. Health and emergency-care platform for the elderly and disabled people in the Smart City. *The Journal of Systems and Software*, 2015, volume 110, p. 253–263. ISSN 0164-1212.
9. Risodkar Y. R., Sangole M. K., Amruta. R. Web Based Health Monitoring System. *International Journal of Advanced Research in Electronics and Communication Engineering*, 2015, volume 4, issue 1, p. 85–89. ISSN 2278 – 909X.
10. Serhani M. A., Menshawy M. E. SME2EM: Smart mobile end-to-end monitoring architecture for life-long diseases. *Computers in Biology and Medicine*, 2016, volume 68, p. 137–154. ISSN 0010-4825.
11. Mano L. Y., Faical B. S., Nakamura L. Exploiting IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition. *Computer Communications*, 2016, volume 89–90, p. 178–190. ISSN 0140-3664.
12. Lamonaca F., Polimeni G., Barbé K., Grimaldi D. Health parameters monitoring by smartphone for quality of life improvement. *Measurement*, 2015, volume 73, p. 82–94. ISSN 0263-2241.

13. Wcislik M., Pozoga M., Smerdzynski P. Wireless Health Monitoring System. *IFAC-PapersOnLine*, 2015, volume 48, issue 4, p. 312–317. ISSN 1474-6670.
14. Yu Z., Liang Y. Facilitating medication adherence in elderly care using ubiquitous sensors and mobile social networks. *Computer Communications*, 2015, volume 65, p. 1–9. ISSN 0140-3664.
15. Hu J., Cui X., Gong Y. Portable microfluidic and smartphone-based devices for monitoring of cardiovascular diseases at the point of care. *Biotechnology Advances*, 2016, volume 34, issue 3, p. 305–320. ISSN 0734-9750.
16. Santos J., Rodrigues J., Denisov V. An IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments. *Journal of Network and Computer Applications*, 2016, volume 71, p. 194–204. ISSN 1084-8045.
17. Wang S. L., Chen Y. L., Kuo A. M. Design and evaluation of a cloud-based Mobile Health Information Recommendation system on wireless sensor networks. *Computers & Electrical Engineering*, 2016, volume 49, p. 221–235. ISSN 0045-7906.
18. Sailunaz K., Alhussein M., Shahiduzzaman M. CMED: Cloud based medical system framework for rural health monitoring in developing countries. *Computers and Electrical Engineering*, 2016, volume 53, p. 469–481. ISSN 0045-7906.
19. Peddi S. V., Kuhad P., Yassine A. An intelligent cloud-based data processing broker for mobile e-health multimedia applications. *Future Generation Computer Systems*, 2017, volume 66, p. 71–86. ISSN 0167-739X.
20. Risso N. A., Neyem A., Benedetto J. I. A cloud-based mobile system to improve respiratory therapy services at home. *Journal of Biomedical Informatics*, 2016, volume 63, p. 45–53, ISSN 1532-0464.
21. Mamun K. A. A., Alhussein M., Sailunaz K. Cloud based framework for Parkinson's disease diagnosis and monitoring system for remote healthcare applications. *Future Generation Computer Systems*, 2017, volume 66, p. 36–47. ISSN 0167-739X.
22. Lounis A., Hadjidj A., Bouabdallah A. Healing on the cloud: Secure cloud architecture for medical wireless sensor networks. *Future Generation Computer Systems*, 2016, volume 55, p. 266–277. ISSN 0167-739X.
23. Moosavi S. R., Gia T. N., Rahmani A. M. SEA: A Secure and Efficient Authentication and Authorization Architecture for IoT-Based Healthcare Using Smart Gateways. *Procedia Computer Science*, 2015, volume 52, p. 452–459. ISSN 1877-0509.
24. Hossain M. S., Muhammad G. Cloud-assisted Industrial Internet of Things (IIoT) – Enabled framework for health monitoring. *Computer Networks*, 2016, volume 101, p. 192–202. ISSN 1389-1286.

25. Talaminos-Barroso A., Estudillo-Valderrama M. A., Ortega-Ruiz F. A Machine-to-Machine protocol benchmark for eHealth applications – Use case: Respiratory rehabilitation. *Computer Methods and Programs in Biomedicine*, 2016, volume 129, p. 1–11. ISSN 0169-2607.
26. Yang Z., Kankanhalli A., Ng B. Y. Examining the pre-adoption stages of healthcare IT: A case study of vital signs monitoring systems. *Information & Management*, 2015, volume 52, issue 4, p. 454–467. ISSN 0378-7206.
27. Varga N., Piri E., Bokor L. Network-assisted Smart Access Point Selection for Pervasive Real-time mHealth Applications. *Procedia Computer Science*, 2015, volume 63, p. 317–324. ISSN 1877-0509.
28. Elhadj H. B., Elias J., Chaari L. Multi-Attribute Decision Making Handover Algorithm for Wireless Body Area Networks. *Computer Communications*, 2016, volume 81, p. 97–108. ISSN 0140-3664.
29. Hassan M. M., Lin K., Yue X. A multimedia healthcare data sharing approach through cloud-based body area network. *Future Generation Computer Systems*, 2017, volume 66, p. 48–58. ISSN 0167-739X.
30. Wang Y., Kung L., Byrd T. A. Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting & Social Change*, 2016, in press, p. 1–11. ISSN 0040-1625.
31. Almadani B., Saeed B., Alroubaiy A. Healthcare systems integration using Real Time Publish Subscribe (RTPS) middleware. *Computers & Electrical Engineering*, 2016, volume 50, p. 67–78. ISSN 0045-7906.
32. Szydło T., Konieczny M. Mobile and wearable devices in an open and universal system for remote patient monitoring. *Microprocessors and Microsystems*, 2016, volume 46, Part A, p. 44–54. ISSN 0141-9331.
33. Lu C. H., Wang W. C., Tai C. C. Design of a heart rate controller for treadmill exercise using a recurrent fuzzy neural network. *Computer Methods and Programs in Biomedicine*, 2016, volume 128, p. 27–39. ISSN 0169-2607.

SANTRAUKA

Linas Silkauskas. Sveikatos stebėjimo sistemos prototipas ankstyvam širdies ir kraujagyslių ligų diagnozavimui. Techninių informacinių sistemų inžinerijos magistro baigiamasis darbas. Vadovė doc. dr. V. Bulbenkienė. Klaipėdos universitetas, Jūros technologijų ir gamtos mokslų fakultetas, Informatikos ir statistikos katedra. Klaipėda, 2017, 76 p.

Šiame darbe pateiktas sveikatos stebėjimo informacinės sistemos su grįžtamoju ryšiu prototipas. Jis leidžia medicams nuotoliniu būdu stebėti pacientų širdies ir kraujagyslių veiklą. Sistemą sudaro keturi pagrindiniai komponentai. Jutikliais fiksuojami sveikatos, judėjimo parametrai (1). Užfiksuoti parametrai iš mobiliosios programos (2) siunčiami į debesų infrastruktūrą (3). Medicinos personalui internetiniame puslapyje (4) pateikiama fiksavimo suvestinė. Medikų išvados, nurodymai siunčiami pacientui. Šiame darbe į vieną bendrą sistemą apjungiami e-Health jutiklių platforma, Zephyr HxM BT juosmens diržas, Xiaomi Mi Band 1s apyrankė, SEN-11574 pulso jutiklis, Android operacinė sistema, REST architektūrinis stilius, internetinės technologijos (PHP, Apache HTTP, MySQL). Taip pat darbe išnagrinėtos išmaniosios, bevielės, mobiliosios ir visuotinės sveikatos monitoringo sistemos, kurios yra pagrįstos debesų kompiuterija, IoT, kriptografija. Sukurtai sistemai apibrėžti funkciniai ir nefunkciniai reikalavimai, parinkta programinė ir techninė įranga, atlikti projektavimo ir programavimo darbai, testavimas ir diegimas. Programinio kodo versijavimas atliktas naudojant Git įrankį. Sistemai parašyti vienetų ir vartotojo sąsajos testai. Įterptinei sistemai sukurta programa įdiegta į Arduino valdiklių plokštę, mobilioji programa – į telefoną su Android operacine sistema, serveryje veikianti programa – į OVH.com debesų kompiuterijos infrastruktūrą, internetinis puslapis korektiškai atvaizduojamas Chrome, Firefox ir IE naršyklėse. Atliktas sukurtos sistemos pagrindinių komponentų verifikavimas. Geriausias vidutinis širdies ritmo fiksavimo tikslumas gautas, kai EKG signalo filtravimui pasirinktas EMA algoritmas ir α koeficientas lygus 20. Naudojant EMA algoritmą pasiektas apie 99% tikslumas, širdies ritmo nuokrypis nuo medicininio prietaiso rodmenų lygus nuo 0 iki 2 dūžių. Iš tyrimo rezultatų nustatyta, kad mobilioji programa nereikalauja daug skaičiavimo ir atminties resursų. Programos vidutinės sąnaudos: apie 2% CPU apkrovimas ir 12 MB RAM. Mobilijai programai veikiant nuolatos 30 dienų, tinkle siunčiamų duomenų kiekis lygus apie 600 MB. Naudojant mobilųjį telefoną turintį bateriją, kurios talpa 3300 mAh, įkrauta 100%, mobiliąja programa be pertraukos galima naudotis apie 15h. Vienu metu prie sistemos gali prisijungti apie 120 vartotojų, esant klaidų dažnumui $\leq 3\%$. Tada debesų serverio vidutinis atsako laikas lygus apie 6 s, o pralaidumas – apie 10 užklausų/s. Internetinio puslapio vidutinis užkrovimo laikas po optimizacijos lygus apie 900 ms, prieš – apie 1100 ms. Užkrovimo laiko pokytis lygus apie -19%.

PAGRINDINIAI ŽODŽIAI: debesų kompiuterija, bevielė komunikacija, medicininiai jutikliai, nuotolinis sveikatos stebėjimas, parametrų fiksavimas.

SUMMARY

Linas Silkauskas. A prototype of the health monitoring system for early cardiovascular diseases diagnosis. Master's thesis of Technical Information Systems Engineering. Supervisor doc. dr. V. Bulbenkienė. Klaipeda University, Faculty of Marine Technology and Natural Sciences, Department of Informatics and Statistics. Klaipeda, 2017, 76 p.

There is a prototype of the health monitoring information system with feedback presented in this work. Medical staff can remotely monitor cardiovascular activity of patients. The system consists of four main components. Health, movement parameters are acquired using sensors (1). Acquired parameters from the mobile application (2) are sent to the cloud infrastructure (3). In the website (4) acquisition summary are supplied to the medical staff. The medical findings, instructions are sent to the patient. In this work e-Health sensors platform, Zephyr HxM BT chest strap, Xiaomi Mi Band 1s bracelet, SEN-11574 pulse sensor, Android operating system, REST architectural style, web technologies (PHP, Apache HTTP, MySQL) are combined in single system. Furthermore, there are analyzed smart, wireless, mobile and ubiquitous health monitoring systems which are based on cloud computing, IoT, cryptography. The are defined the functional and non-functional requirements, chosen software and hardware, performed design and programming, testing and deployment for developed system. The are performed source code version control using Git tool. Unit and user interface tests are written for system. Developed application for embedded system are deployed to the Arduinio board, mobile application – to the phone with Android operating system, application that operates in the server – to the OVH.com cloud computing infrastructure, website are correctly displayed in Chrome, Firefox and IE browsers. The are performed verification of developed system main components. The best acquisition accuracy of average heart rate achieved when ECG signal filtered using EMA algorithm and α coefficient equals 20. Using EMA algorithm achieved about 99% accuracy, deviation of the heart rate equals from 0 to 2 beats when results are compared with readings of medical device. From research results determined that mobile application does not require a lot of computing and memory resources. The average consumption of the program: about 2% CPU load and 12 MB RAM. The amount of data sent over the network equals about 600 MB when mobile application running continuously for 30 days. Mobile application without a break can be used about 15h, when using mobile phone having a battery which capacity 3300 mAh, charged 100%. At the same time 120 users can connect to the system, at error rate $\leq 3\%$. Then average response time of the cloud server equals about 6 s, throughput – about 10 requests/s. Average load time of web page after optimization equals about 900 ms, before – about 1100 ms. The change of load time equals about -19%.

KEYWORDS: cloud computing, wireless communication, medical sensors, remote health monitoring, parameters acquisition.

Kompaktinis diskas

Kompaktiniame diske pateikiami:

1. Darbo aprašymas:
 - a. „*Linas_Silkauskas_magistrinio_darbo_aprasymas.docx*“;
 - b. „*Linas_Silkauskas_magistrinio_darbo_aprasymas.pdf*“;
2. Darbo pristatymas:
 - a. „*Linas_Silkauskas_magistrinio_darbo_pristatymas.pptx*“;
3. Darbo modeliai ir diagramos:

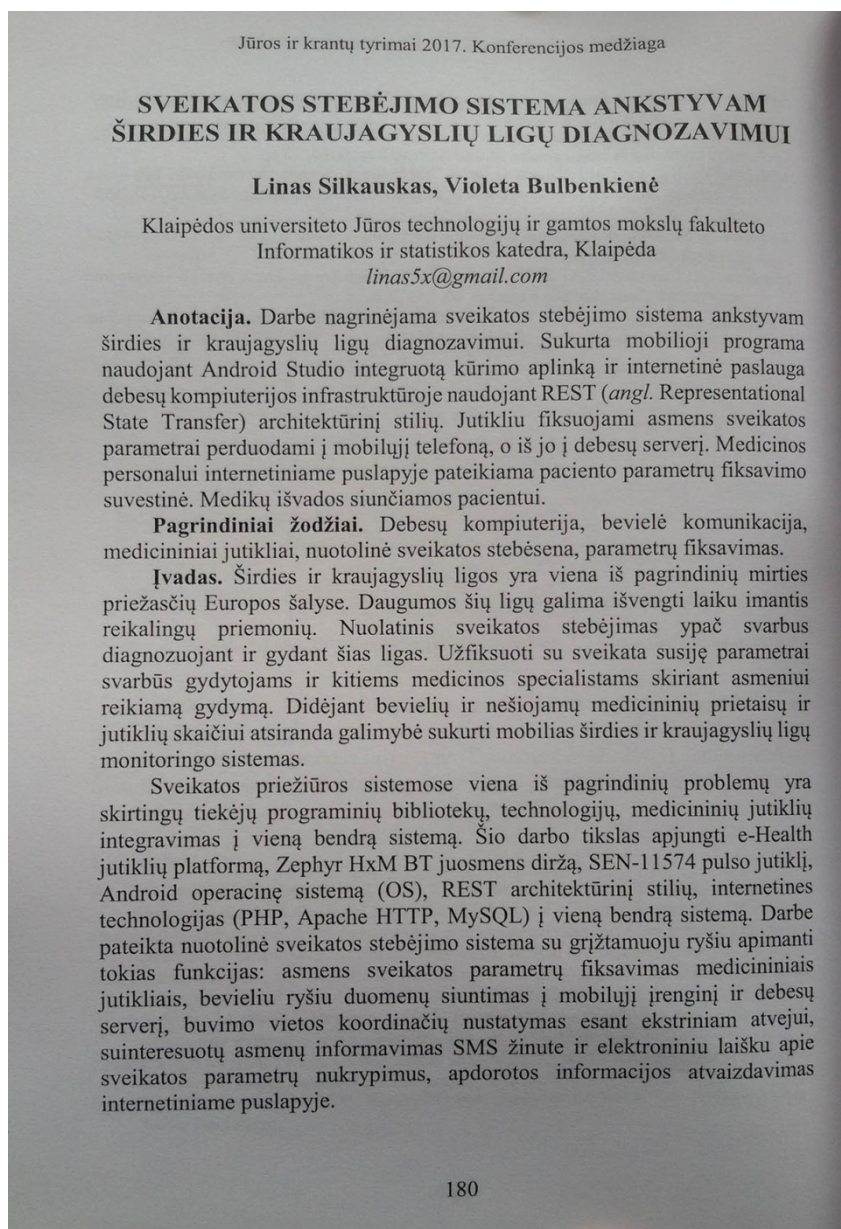
Pavadinimas	Failas
Sistemų grupavimas	„ <i>Modeliai/Minciu_zemelapis.jpg</i> “
Sistamai apibrėžti funkciniai ir nefunkciniai reikalavimai	„ <i>Modeliai/Reikalavimu_diagrama.jpg</i> “
Sistemos įgyvendinimui pasirinkta programinė įranga	„ <i>Modeliai/Strukturos_aprasu_diagrama.jpg</i> “
Suprojektuota techninė įranga naudojant SEN-11574 pulso jutiklį	„ <i>Modeliai/Elektronine_schema_pulso_jutiklis.jpg</i> “
Algoritmas širdies ritmo fiksavimui naudojant SEN-11574 pulso jutiklį	„ <i>Modeliai/Veiklos_diagrama_pulso_jutiklis.jpg</i> “
Algoritmas širdies ritmo fiksavimui naudojant e-Health platformą su EKG jutikliu	„ <i>Modeliai/Veiklos_diagrama_eHealth_platforma_EKG_jutiklis.jpg</i> “
Mobiliosios programos klasių ir interfeisų saugojimo struktūra	„ <i>Modeliai/Paketu_diagrama_mobilioji_programa.jpg</i> “
Mobiliosios programos statinis vaizdas	„ <i>Modeliai/Klasiu_diagrama_mobilioji_programa.jpg</i> “
Algoritmas fiksuojamų parametrų siuntimui iš mobiliosios programos į debesų serverį	„ <i>Modeliai/Veiklos_diagrama_mobilioji_programa.jpg</i> “
Sukurta REST API, kuri įdiegta debesų kompiuterijos infrastruktūroje	„ <i>Modeliai/REST_API_diagrama.jpg</i> “
Sistemoje įgyvendintas siunčiamų duomenų užšifravimas ir iššifravimas	„ <i>Modeliai/Seku_diagrama_debesu_serveris.jpg</i> “
Suprojektuota duomenų bazės struktūra	„ <i>Modeliai/Esybiu_rysiu_diagrama.jpg</i> “
Algoritmas fiksavimo suvestinės PDF dokumento generavimui debesų serveryje	„ <i>Modeliai/Veiklos_diagrama_debesu_serveris.jpg</i> “
Algoritmas fiksuojamų parametrų atvaizdavimui internetiniame puslapyje	„ <i>Modeliai/Veiklos_diagrama_internetinis_puslapis.jpg</i> “
Internetinio puslapio failų optimizavimo procesas	„ <i>Modeliai/Busenu_diagrama_internetinis_puslapis.jpg</i> “
Sistemos įgyvendinimui panaudotos programinės bibliotekos	„ <i>Modeliai/Komponentu_diagrama.jpg</i> “
Nuotolinė širdies ir kraujagyslių ligų stebėjimo sistema su grįžtamoju ryšiu	„ <i>Modeliai/Raiskuisis_paveikslelis.jpg</i> “
Sistemos diegimas į galutiniams vartotojams skirtą aplinką	„ <i>Modeliai/Diegimo_diagrama.jpg</i> “
Algoritmas automatizuotam programos diegimui į debesų serverį	„ <i>Modeliai/Veiklos_diagrama_diegimas.jpg</i> “

Mokslinė veikla

1. Straipsnis, žodinis pranešimas

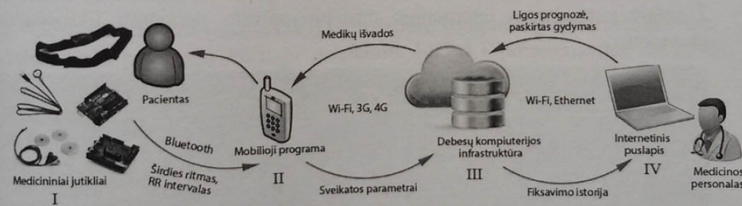
Dalyvauta, 10-oje nacionalinėje jūros mokslų ir technologijų konferencijoje „Jūros ir krantų tyrimai 2016“, kuri vyko balandžio 26–28 d. Parengtas straipsnis ir žodinis pranešimas „Sveikatos stebėjimo sistema ankstyvam širdies ir kraujagyslių ligų diagnozavimui“. Darbo vadovė doc. dr. Violeta Bulbenkienė. Žodinis pranešimas pristatytas Palangos Kurhauzo salėje. Straipsnis publikuotas konferencijos pranešimų medžiagoje:

Silkauskas L., Bulbenkienė V. Sveikatos stebėjimo sistema ankstyvam širdies ir kraujagyslių ligų diagnozavimui. *Jūros ir krantų tyrimai 2017*. Klaipėda: Klaipėdos universiteto leidykla, 2017, p. 180–184. ISSN 2538-7243.



Sveikatos stebėjimo sistemos. Siekiant gerinti žmonių gyvenimo kokybę, kuriamos įvairios sveikatos stebėjimo sistemos. Viena iš jų tai tinklo architektūra, kuri pagrįsta bevieliais ant tiriamojo kūno paviršiaus dedamais jutiklių tinklais (WBAN) ir debesų kompiuterija (Hassan et al., 2017). Taip pat yra sukurta mobili tinklo sąsaja, kuri pagrįsta daiktų internetu (IoT). Renkama informacija apie asmens buvimo vietą, širdies ritmą ir aktyvumą, kuri siunčiama į mobilųjį įrenginį. (Santos et al., 2016). Norint gydytojams palengvinti širdies ir kraujagyslių ligų diagnostiką yra sukurta nuotolinė mobilioji paciento sveikatos būklės stebėjimo sistema – TeleCARE (Szydło et al., 2016). Panaudojant neraiškųjį neuroninio tinklo reguliatorių yra sukurta širdies ritmo reguliavimo sistema treniruočių ir mankštos metu (Lu et al., 2016).

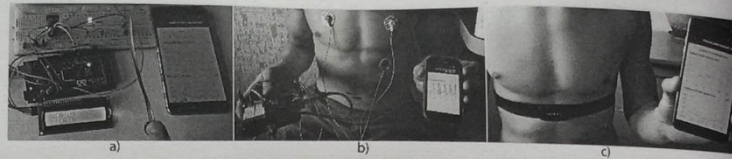
Sistemos principinė schema. Sukurta nuotolinė sveikatos stebėjimo sistema su grįžtamoju ryšiu, kurią sudaro keturios pagrindinės dalys (1 pav.). Medicininiais jutikliais fiksuojami sveikatos, judėjimo parametrai (I). Užfiksuoti parametrai iš mobiliosios programos (II) siunčiami į debesų serverį (III). Medicinos personalui internetiniame puslapyje (IV) pateikiama fiksuojamo suvestinė. Medikų išvados, nurodymai siunčiami pacientui.



1 pav. Nuotolinė sveikatos stebėjimo sistema su grįžtamoju ryšiu.

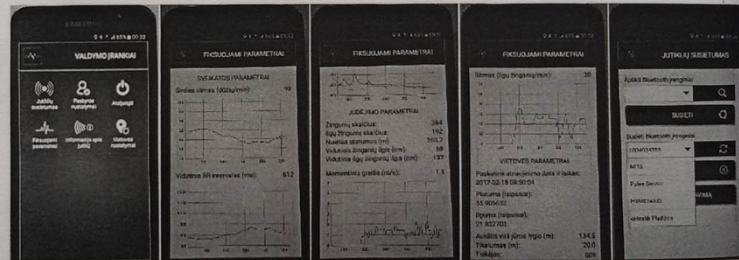
Sveikatos parametrų fiksavimas medicininiais jutikliais. Pateiktoje sistemoje asmens širdies ritmas ir laiko intervalas tarp dūžių (RR) fiksuojamas naudojant tris jutiklius (2 pav.): SEN-11574 pulso jutiklį (a), e-Health jutiklių platformą (b), Zephyr juosmens diržą (c). Jutikliniai duomenys perduodami į mobilųjį telefoną. Prie e-Health platformos prijungtas elektrokardiogramos (EKG) jutiklis, kurį sudaro trys elektrodai. Užfiksuotas EKG signalas perduodamas į e-Health platformą, o iš jos į Arduino Uno valdiklių plokštę. Iš signalo nustatomas asmens širdies ritmas ir RR. Sveikatos parametrai iš Arduino perduodami į HC-06 Bluetooth modulį, o iš jo į mobilųjį telefoną. Juosmens diržu fiksuojami asmens sveikatos (širdies ritmas, RR intervalas) ir judėjimo (žingsnių skaičius, nueitas atstumas, momentinis greitis, ritmas – žingsnių/min) parametrai. Dirže integruotas Bluetooth modulis, kuris

duomenis perduoda į mobilųjį telefoną. Jame įdiegta sukurta mobilioji programa, kurioje atvaizduojami užfiksuoti sveikatos parametrai.



2 pav. Širdies ritmo fiksavimas pulso jutikliu (a), e-Health platforma (b) ir Zephyr juosmens diržu (c).

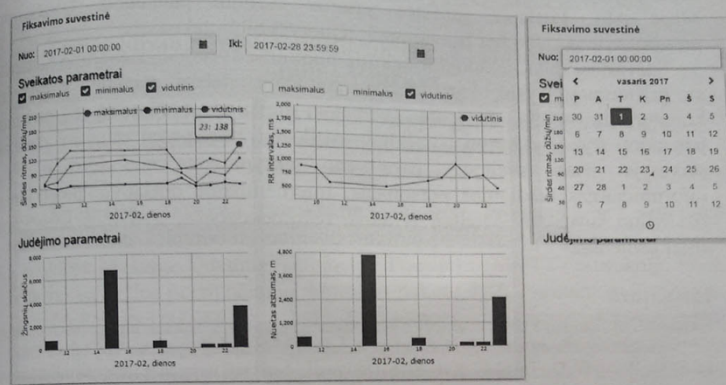
Užfiksuotų parametų atvaizdavimas mobilijoje programoje. Pateiktoje sistemoje medicininiais jutikliais užfiksuoti parametrai Bluetooth ryšiu perduodami į mobilųjį įrenginį su Android OS. Jame įdiegta sukurta mobilioji programa (3 pav.). Naudojant GPS nustatomos buvimo vietos koordinatės (platuma, ilguma). Mobiliojoje programoje taip pat tikrinama ar užfiksuotas širdies ritmas neperžengia leistinos ribos. Esant nukrypimams SMS žinute ir elektroniniu laišku informuojami suinteresuoti asmenys. Žinutėje pateikiamas internetinio puslapio adresas peržiūrėti fiksuojamus parametrus.



3 pav. Užfiksuotų parametų atvaizdavimas mobilijoje programoje.

Užklausų siuntimas į debesų serverį, duomenų apdorojimas. Užfiksuoti sveikatos, judėjimo ir vietovės parametrai Wi-Fi arba mobiliuoju internetu ryšiu siunčiami į debesų serverį. Komunikacijos kūrimui panaudotas REST architektūrinis stilius. Vartotojo užklausos, debesų serverio atsakas siunčiami HTTP (*angl.* HyperText Transfer Protocol) protokolu naudojant POST metodą. Duomenų objektai perduodami JSON formatu. Debesų serveryje surinkti parametrai išsaugomi MySQL 5.7 duomenų bazėje.

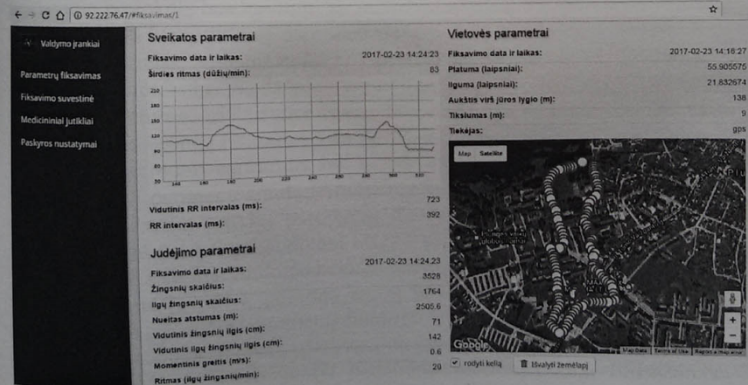
Informacija apdorojama ir internetiniame puslapyje pateikiama fiksavimo suvestinė (4 pav.).



4 pav. Fiksavimo suvestinė.

Sukurta sistema įdiegta OVH.com debesų kompiuterijos infrastruktūroje. Internetinės paslaugos kūrimui panaudota PHP 7.1 programavimo kalba ir Silex 2 karkasas (angl. framework).

Internetinis puslapis fiksuojamų parametų atvaizdavimui. Sukurtas internetinis puslapis, kuriame atvaizduojami sveikatos, judėjimo ir vietovės parametrai užfiksuoti tam tikru laiko momentu (5 pav.).



5 pav. Internetinis puslapis fiksuojamų parametų atvaizdavimui.

Panaudota AJAX technologija atnaujinant puslapio turinį jo neperkraunant. Buvimo vieta atvaizduojama žemėlapyje naudojant Google Maps API. Puslapio grafinės sąsajos kūrimui panaudota JavaScript programavimo kalba bei Bootstrap 3, jQuery 3 bibliotekos.

Išvados. Suprojektuota ir sukurta nuotolinė sveikatos stebėjimo sistema su grįžtamoju ryšiu, kuri leidžia medikams stebėti pacientų širdies ir kraujagyslių veiklą. Paciento sveikatos, judėjimo parametrų fiksavimui panaudoti trys medicininiai jutikliai. Užfiksuoti parametrai saugomi debesų serveryje. Esant sveikatos parametrų nukrypimams nuo leistinų ribų informuojamas medicinos personalas. Internetiniame puslapyje pateikiama fiksavimo suvestinė. Medikų išvados, rekomendacijos, gydymas siunčiami atgal pacientui.

Literatūra

- Hassan M. M., Lin K., Yue X. et al. 2017. A multimedia healthcare data sharing approach through cloud-based body area network. *Future Generation Computer Systems*, Vol. 66, 48–58.
- Lu C. H., Wang W. C., Tai C. C. et al. 2016. Design of a heart rate controller for treadmill exercise using a recurrent fuzzy neural network. *Computer Methods and Programs in Biomedicine*, Vol. 128, 27–39.
- Santos J., Rodrigues J., Denisov V. et al. 2016. An IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments. *Journal of Network and Computer Applications*, Vol. 71, 194–204.
- Szydło T., Konieczny M. 2016. Mobile and wearable devices in an open and universal system for remote patient monitoring. *Microprocessors and Microsystems*, Vol. 46, 44–54.

Dalis konferencijos programos:



10-oji nacionalinė jūros mokslų ir technologijų konferencija JŪROS IR KRANTŲ TYRIMAI 2017

Kurhauzo salė (Grafių Tiškevičių al. 1, Palanga)

Programa

Balandžio 28 d., penktadienis	
<i>V sesija: Pajūrio ir jūros flora bei fauna - sava ir svetima</i> <i>Moderatorius prof. dr. Darius Daunys</i>	
9.00 – 9.20	M. Bučas , Z. Sinkevičienė, M. Kataržytė, D. Vaičiūtė, J. Petkuvienė, R. Ilginė, V. Stragauskaitė. Kokios aplinkos sąlygos palankiausios maurabraginiams dumbliams Kuršių mariose?
9.20 – 9.40	M. Kataržytė , G. Umgiesser, G. Schernewski, R. Staponkienė, S. Liaugaudaitė, N. Čerkasova, J. Mėžinė. Maudyklos Lietuvos pajūryje: mikrobiologinės taršos ir jos sklaidos tyrimai
9.40 – 10.00	R. Chuševė , D. Daunys. Kaip vertinamas makrofaunos jautrumas ir koks jo tikslumas?
10.00 – 10.20	S. Olenin , A. Narščius, G. Srėbalienė. Naujas indeksas vertinant aplinkos apsaugos priemonių efektyvumą svetimkraščių rūšių atžvilgiu
10.20 – 10.40	G. Srėbalienė , S. Olenin, A. Narščius. Svetimkraštės rūšies dvigeldžio moliusko <i>Rangia cuneata</i> invaziškumo potencialo ir rizikos vertinimas
10.40 – 11.00	Kavos pertrauka
<i>VI sesija: Jūrinės technologijos mokslui ir gyvenimui</i> <i>Moderatorė doc. dr. Rima Mickevičienė</i>	
11.00 – 11.20	N. Nika , A. Kontautas N. Ar visada žuvinaiša duoda naudą?
11.20 – 11.40	A. Skaisgirienė , L. Lupeikaitė, J. Janutėnienė, H. Yukhnevich. Biopreparatų panaudojimas mažose nuotekų valyklose
11.40 – 12.00	D. Šateikienė , J. Janutėnienė. Laivo balastinio vandens valymo įrangos parinkimo metodika
12.00 – 12.20	L. Silkauskas , V. Bulbenkienė. Sveikatos stebėjimo sistema ankstyvam širdies ir kraujagyslių ligų diagnozavimui
12.20 – 12.50	Kavos pertrauka
12.50 – 13.10	A. Štuopys , V. Lupandinas, A. Tadžijevs , Klaipėdos (Memelio) vandens kelių tarnybos slėptuvės įranga: ankstyvųjų plieno suvirinimo technologijų pavyzdžiai
13.10 – 13.30	P. Bogdevičius , I. Zygaitė, O. Prentkovskis, M. Bogdevičius. Transmisijos su kardaniniais velenais dinaminė charakteristikų tyrimas
13.30 – 13.50	V. Daukšys , S. Lebedevas. Dyzelino ir suslėgtų gamtinių dujų mišiniu dirbančio variklio darbo proceso skaitinis modeliavimas.
13.50 – 14.00	Apibendrinimas, konferencijos uždarymas.

2. Straipsnis, standinis pranešimas

Dalyvauta 9-oje nacionalinėje jūros mokslų ir technologijų konferencijoje „Jūros ir krantų tyrimai 2016“, kuri vyko balandžio 27–29 d. Parengtas straipsnis ir standinis pranešimas „*Simbolių atpažinimo sistema pagrįsta vaizdų apdorojimu naudojant MATLAB*“. Darbo vadovė doc. dr. Violeta Bulbenkienė. Standinis pranešimas pristatytas Klaipėdos universiteto Auditorijų ir konferencijų salės komplekse „Aula Magna“. Straipsnis publikuotas konferencijos pranešimų medžiagoje:

Silkauskas L., Bulbenkienė V. Simbolių atpažinimo sistema pagrįsta vaizdų apdorojimu naudojant MATLAB. *Jūros ir krantų tyrimai 2016*. Klaipėda: Klaipėdos universiteto leidykla, 2016, p. 166–170. ISBN 978-9955-18-901-5.

SIMBOLIŲ ATPAŽINIMO SISTEMA PAGRĮSTA VAIZDŲ APDOROJIMU NAUDOJANT MATLAB

Linas Silkauskas, Violeta Bulbenkienė

Klaipėdos universiteto Jūros technologijų ir gamtos mokslų fakulteto
Informatikos ir statistikos katedra, Klaipėda
linas5x@gmail.com, bulbenkiene@gmail.com

Anotacija. Darbe pateikta simbolių atpažinimo sistema sukurta naudojant Matlab programinę įrangą ir vaizdų apdorojimo modulį. Sistema pritaikyta atpažinti valstybinius numerius. Atlikti atpažinimo tikslumo ir proceso trukmės tyrimai esant 6-ioms skirtingoms aplinkos sąlygoms ir 5-ioms skirtingoms vaizdų rezoliucijoms.

Pagrindiniai žodžiai. Simbolių atpažinimas, vaizdų apdorojimas, filtravimas, šablono atitikimas, bottom-hat metodas.

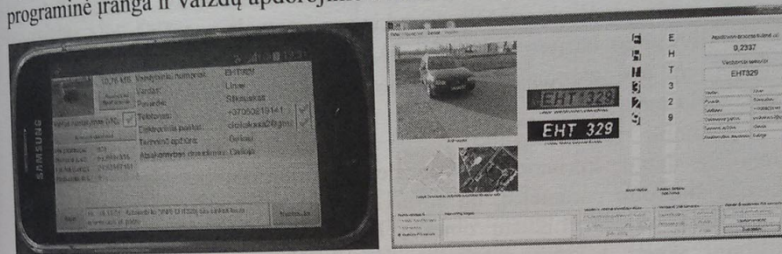
Įvadas. Simbolių atpažinimas yra plačiai naudojamas daugelyje sistemų: atpažįstant skenuotose, nufotografuotose knygose, vaizduose pateiktus užrašus, transporto priemonių valstybinius numerius, jūrinių konteinerių prefixą ir numerį, asmens identifikavimo dokumentuose pateiktus duomenis ir t.t.

Sistemos galinčios nusiųsti vaizdus iš vieno įrenginio į kitą, nustatyti buvimo vietos koordinatas, pagal jas pateikti žemėlapių vaizdą, atpažinti valstybinius numerius, automatiškai nusiųsti SMS žinutę į telefoną ar laišką į elektroninį paštą, internetiniame puslapyje pateikti norimą informaciją, registruoti vartotojus, redaguoti jų informaciją, jau egzistuoja. Tačiau šios visos sistemos yra skirtos tam tikriems procesams atlikti. Šiame darbe pateikta atpažinimo sistema, kuri pritaikyta atpažinti transporto priemonių valstybinius numerius. Sistema atlieka visas prieš tai įvardintas funkcijas.

Atpažinimo sistemos. Automatinis valstybinių numerių atpažinimas yra vienas iš svarbiausių transporto priemonių identifikavimo būdų. Tam šiuo metu naudojami įvairūs metodai: naudojant Gabor filtrą yra sukurtas programinis algoritmas numerių aptikimui vaizduose (Tadic et al., 2016); dviejų etapų simbolių segmentavimo metodas (Tian et al., 2015); numerių aptikimo metodas pagrįstas Wavelet transformacija (Yu et al., 2015); panaudojant SIFT metodą yra sukurtas valstybinių numerių atpažinimo algoritmas (Wang et al., 2015).

Vaizdų fiksavimas mobiliąja programa ir numerių atpažinimas Matlab aplinkoje. Pateiktoje sistemoje vaizdai fiksuojami mobiliuoju įrenginiu su Android OS bei sukurta mobiliąja programa. Vaizdai į serverį siunčiami Wi-Fi arba mobiliuoju interneto ryšiu. Buvimo vietos koordinatės (platumas, ilguma) nustatomos su GPS. Serveris pagal atpažintus numerius grąžina automobilio savininko duomenis, kurie pateikiami programos laukeliuose. Pagal gautą telefono numerį savininkui nusiunčiama informuojanti SMS žinutė, kad jo

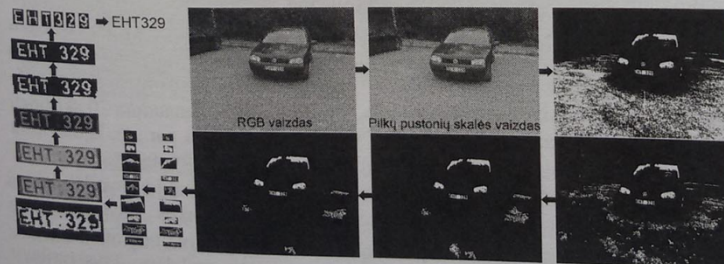
automobilis užfiksuotas stovint neleistinoje vietoje. Žinutėje pateikiami prisijungimo duomenys prie nurodyto internetinio puslapio. Puslapyje pateikiamas užfiksuotas vaizdas su automobiliu ir žemėlapis su pažymėta stovėjimo vieta. Pateiktoje sistemoje vaizdai apdorojami su valstybinių numerių atpažinimo programa (1 pav.). Programos kūrimui naudota Matlab R2014b programinė įranga ir vaizdų apdorojimo modulis Image Processing Toolbox.



1 pav. Mobilioji programa vaizdų fiksavimui ir programa numerių atpažinimui.

Atpažinimo sistemą sudaro trys etapai: numerių aptikimas, simbolių segmentavimas ir jų atpažinimas pagal šablonus.

Iš serverio gautas spalvotas vaizdas (RGB) yra paverčiamas į pilkų pustonų skalės vaizdą naudojantis vieno spalvinio kanalo (raudonojo) metodu (2 pav.). Gauti pilkų pustonų skalės vaizdai yra paverčiami į dvejetainius. Pirmojo ciklo metu slenksčio reikšmė lygi 0,73. Ji pasirinkta todėl, kad apie ~40 % vaizduose su šia reikšme randami numeriai pirmosios ciklo iteracijos metu. Gautuose dvejetainiuose vaizduose yra daug nereikalingų objektų. Naudojantis gretimų pikselių žymėjimo (*angl.* Connected Components Labeling) metodu randami šie objektai.



2 pav. Vaizdų apdorojimo etapai.

Kadangi numerių sritis turi visa pilnai matytis, todėl vaizde pašalinami visi objektai, kurie liečia vaizdo kraštus. Maži objektai yra šalinami atsižvelgiant į jų plotą. Jeigu objekto plotas yra mažesnis už nurodytą ribą, jis pašalinamas. Ploto reikšmė pagal kurią yra šalinami maži objektai yra lygi vaizdo pločio ir aukščio

sandaugai padalintai iš 1500. Taip pat pašalinami objektai, kurių plotis mažesnis už aukštį.

Likę objektai pagal apskaičiuotus Bounding Box parametrus yra iškerpami iš dvejetainio ir iš RGB vaizdo. Kiekvienas iškirptas dvejetainis vaizdas yra pasukamas pagal apskaičiuotą orientacijos kampą. Pasukti objektai filtruojami atsižvelgiant į realios valstybinių numerių lentelės matmenis. Vaizde paliekami objektai, kurių pločio ir aukščio santykis yra intervale nuo 3,64 iki 5,66. Jeigu po šio filtravimo barjero nebelieka nė vieno objekto, tai keičiama slenksčio reikšmė, tol kol atsiranda bent vienas objektas tenkinantis visus prieš tai pateiktus kriterijus.

Simbolių segmentavimo metu likusiuose RGB vaizduose ieškomi simboliai. Jeigu, kuriame nors vaizde rasti objektai netenkina numeriuose esančių simbolių parametrų, tai atitinkamas RGB vaizdas yra pašalinamas iš pretendentų sąrašo. Likusių objektų RGB vaizdai yra paverčiami į pilkų pustonių skalės vaizdus naudojantis Luminosity metodu. Pirmiausia pilkų pustonių skalės vaizdas apdorojamas bottom-hat metodu, o po to su Otsu metodu paverčiamas į dvejetainį. Dvejetainiame vaizde objektai filtruojami. Jie turi tenkinti valstybiniuose numeriuose esančių simbolių parametrus. Naudojami du filtrai. Su pirmuoju filtru pašalinami objektai, kurių aukštis pakankamai mažas lyginant su vaizdo aukščiu, o su antruoju – pašalinami objektai, kurių aukštis pakankamai skiriasi nuo kitų objektų aukščių. Sritis, kurioje lieka objektų laikoma valstybinių numerių lentele. Iš vaizdo simboliai iškerpami į atskirus dvejetainius vaizdus.

Simboliai yra atpažįstami naudojant koreliacijos metodą. Kiekvienas iš vaizdo iškirptas simbolis yra lyginamas su simbolių šablonais, skaičiuojant koreliacijos koeficientą. Pasirenkamas to šablono tekstinis atitikmuo su kuriuo gauta didžiausia koreliacijos koeficiento reikšmė.

Atpažinimo tikslumo ir proceso trukmės tyrimai. Atlikti sistemos tyrimai įvairiomis aplinkos sąlygomis, kurių metu nustatytas atpažinimo tikslumas ir atpažinimo proceso trukmė. Tyrimai šviesiu paros metu atlikti su 5-iomis skirtingomis vaizdų rezoliucijomis. Tyrimui atlikti panaudota 20 skirtingų vaizdų kiekvienai rezoliucijai (viso 100 vaizdų). Rezultatai parodė, kad šviesiu paros metu bendrasis atpažinimo tikslumas svyruoja 95÷99 % intervale. Didžiausias bendrasis tikslumas gautas su 2048x1536 rezoliucijos vaizdais.

Taip pat atlikti tyrimai 5-iomis skirtingomis aplinkos sąlygomis. Du tyrimai atlikti šviesiu paros metu, tačiau 1-ame tyrime valstybiniai numeriai yra purvini ar apdulkėję, o 2-ame – yra saulės atspindžių trukdžių. Trys tyrimai atlikti tamsiu paros metu (temstant, prietemoje, naktį). Tyrimo tikslas – nustatyti kokią įtaką sukurta atpažinimo sistemai turi aplinkos sąlygos. Iš viso tyrimams panaudota 100 vaizdų (1 lentelė).

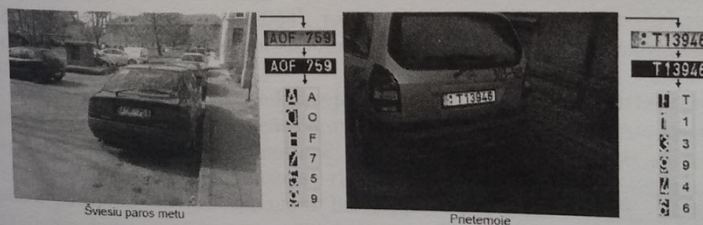
1 lentelė

Tyrimų rezultatai esant skirtingoms aplinkos sąlygoms

Aplinkos sąlyga		Tikslumas, %				Bendra atpažinimo proceso trukmė, s
		Numerių aptikimas	Simbolių		Bendras	
			Segmentavimas	Atpažinimas		
Šviesiu paros metu	Purvai, dulkės	84	59	77	73	1,1
	Atspindžiai	75	70	93	79	2,8
Temstant		89	76	90	85	1,0
Prietoje		99	94	90	94	0,5
Naktį		75	59	88	74	1,1

Esant skirtingoms aplinkos sąlygoms bendrasis atpažinimo tikslumas svyruoja 73÷94 % intervale. Didžiausias bendrasis tikslumas yra prietoje.

Taip pat atliktas tyrimas, kurio metu nustatyta atpažinimo sistemos viso apdorojimo proceso trukmė. Tyrimas atliktas šviesiu paros metu, naudojant 640x480 rezoliucijos vaizdus ir „Ežio“ mobilųjį internetą, kurio maksimalus greitis 0,5 Mbps. Užfiksuota 10 vaizdų. Nustatyta, kad viso apdorojimo proceso trukmė lygi 16,1 s. Ją sudaro vaizdo siuntimo trukmė (79 %), atpažinimo proceso trukmė (4 %) ir kitų procesų (buvimo vietos koordinatų, savininko duomenų, užklausų Google serveriui siuntimas, žemėlapiu generavimas ir t.t.) trukmė (17 %). Keletas tyrimo metu naudotų vaizdų pateikiami 3 pav.



3 pav. Tyrimų metu naudoti vaizdai.

Išvados. Išanalizavus simbolių atpažinimo sistemas bei vaizdų apdorojimo metodus pateikiame sukurta atpažinimo sistemą, kurioje simboliai išskiriami iš vaizdo naudojant bottom-hat ir Otsu metodus, o atpažinami pagal šablonus. Ši sistema buvo pritaikyta atpažinti transporto priemonių valstybinius numerius. Vaizdų fiksavimas vykdomas naudojant sukurta mobiliąją programą. Atlikti atpažinimo tikslumo ir proceso trukmės tyrimai esant 6-ioms skirtingoms aplinkos sąlygoms ir 5-ioms skirtingoms vaizdų rezoliucijoms. Tyrimai parodė, kad šviesiu paros metu bendrasis atpažinimo tikslumas svyruoja 95÷99 % intervale, o trukmė – 0,7÷2,5 s, priklausomai nuo rezoliucijos. Esant

sudėtingesnėms aplinkos sąlygoms tikslumas svyruoja $73 \div 94$ % intervale, o trukmė – $0,5 \div 2,8$ s. Viso apdorojimo proceso trukmė lygi 16,1 s.

Pakeitus atpažinimo sistemoje naudojamų metodų koeficientus, parametrus, ribines reikšmes, galima ją pritaikyti atpažįstant jūrinių konteinerių prefiksą ir numerį bei kitose panašiose sistemose.

Literatūra

- Yu S., Li B., Zhang Q. et al. 2015. A novel license plate location method based on wavelet transform and EMD analysis. *Pattern Recognition*, Vol. 48, Issue 1, 114–125.
- Tadic V., Popovic M., Odry P. 2016. Fuzzified Gabor filter for license plate detection. *Engineering Applications of Artificial Intelligence*, Vol. 48, 40–58.
- Tian J., Wang R., Wang G. et al. 2015. A two-stage character segmentation method for Chinese license plate. *Computers & Electrical Engineering*, Vol. 46, 539–553.
- Wang Y., Ban X., Chen J. et al. 2015. License plate recognition based on SIFT feature. *Optik*, Vol. 126, Issue 21, 2895–2901.

9-oji mokslinė – praktinė konferencija
JŪROS IR KRANTŲ TYRIMAI 2016



Auditorijų ir konferencijų salės kompleksas „Aula Magna“
(H. Manto 84, Klaipėda, „Studlandas“).

Programa

* Stendiniai pranešimai
V. Andrašūnas, R. Jankauskienė. Juodažiočio grundalo (<i>Neogobius melanostomus</i> , Pallas 1811) nerštavietės Kuršių mariose ir jų erdvinė priklausomybė nuo žiemojimui tinkamų gylių
D. Bagočius, A. Narščius, V. Rukšėnienė. Laivybos povandeninio triukšmo modeliavimas Lietuvos jūrinėje teritorijoje, panaudojant laivų automatinės pozicionavimo sistemos (AIS) duomenis
Ž. Berneckytė. Pajūrinės stoklės (<i>Cakile maritima</i> L.) populiacijos būklė Lietuvos pajūrio zonoje ties Melnrage
V. Burškytė. Aplinkos apsaugos aspektų atsaja jūrų uostų pavyzdžiu
N. Čerkasova, G. Umgiesser, A. Ertürk. Minijos baseino hidrologinio modelio sudarymo ypatumai
J. Dainys, E. Jakubavičiūtė, Ž. Pūtys, L. Ložys. Lietuvos ungurių populiacijos būklė: sukauptos energetinės atsargos gonadų subrandinimui ir sėkmingai nerštinei migracijai
J. Galinienė, D. Verkulevičiūtė-Kriukienė, S. Gadal. Pajūrio urbanizuotų teritorijų plėtros iššūkiai landšaftui.
I. Gudaitytė. Sunkiųjų dyzelinių variklių ekologinių rodiklių eksperimentiniai tyrimai
M. Kataržytė, G. Umgiesser, G. Schernewski, N. Čerkasova, E. Baltrūnaitė, S. Liaugaudaitė, J. Mėžinė. Naujos maudyklos Kuršių mariose – sisteminiu požiūriu grindžiamos struktūros taikymas kranto zonos tyrimams ir valdymui
Ž. Kregždys. Žuvų jauniklių gausumo Baltijos jūros Lietuvos priekrantėje tyrimai
R. Naujikienė, D. Dzemydienė. Darnios plėtros sprendimai ir kompleksinės priemonės vertinant vandens nuotekų užterštumą
J. Navašinskienė. Pavojingi vėjai ir reiškiniai Klaipėdos uosto akvatorijoje
R. Nekrošienė, L. Razmuvienė. Pajūrio kopų augalų ekspozicija Klaipėdos universiteto Botanikos sode: rūšių įvairovė ir tyrimų perspektyvos
N. Nika, J. Lesutienė, E. Grinienė, A. Kontautas, R. Statkus, Z. Gasiūnaitė. Jūrinės ekologijos ir žuvininkystės mokslo tyrimų ir eksperimentinės plėtros naujos galimybės Klaipėdoje
E. Petkevičiūtė, A. Razinkovas-Baziukas. Žuvų parazitologinių-ekologinių tyrimų svarba
D. Preibys, I. Bagdanavičiūtė. Aplinkos tinkamumo vertinimas vėjo jėgainių parkų įrengimui taikant GIS erdvinį modeliavimą
N. Remeikaitė-Nikienė, G. Lujanienė, V. Malejevas, R. Barisevičiūtė, G. Garnaga-Budrė, A. Stankevičius. Sezoninė skendinčios organinės medžiagos $\delta^{13}C$ ir $\delta^{15}N$ kaita Kuršių mariose
V. Rukšėnienė, I. Dailidienė, L. Kelpšaitė-Rimkienė. Meteorologinių parametrų įtaka paviršinio vandens temperatūrai pietryčių Baltijos jūroje
A. Samuilovienė, A. Šiaulyš, A. Zaiko. Genetinė midijų (<i>Mytilus</i> spp.) populiacijų struktūra ir populiacijų jungimosi analizė mikrosatelitinių DNR žymenų pagalba
L. Silkauskas, V. Bulbenkienė. Simbolių atpažinimo sistema pagrįsta vaizdų apdorojimu naudojant MATLAB
S. Solovjova. Kumuliacinių kreivių taikymas vertinant Kuršių marių ir Baltijos jūros pietrytinės dalies makrozoobentosos ilgalaikio monitoringo duomenis
D. Špučys, V. Bulbenkienė. Klaipėdos miesto rūšiuojamų šukšlių konteinerių stebėjimo ir duomenų rinkimo sistema
D. Vaičiūtė, M. Bresciani, C. Giardino, A. Razinkovas-Baziukas. INFORM – Europos vidaus vandenų ekologinio statuso stebėsenos ir prognozavimo gerinimas derinant naujus distancinius tyrimų ir modeliavimo metodus

Stendinis pranešimas pristatytas Klaipėdos universiteto Auditorijų ir konferencijų salės komplekse „Aula Magna“.



3. Straipsnis, žodinis pranešimas

Dalyvauta 8-oje respublikinėje mokslinėje-praktinėje studentų konferencijoje „Verslas, technologijos, biomedicina: inovacijų išvalgos 2016“, kuri vyko gegužės 17 d. Parengtas straipsnis ir žodinis pranešimas „UAB „EMTN“ įmonės veiklos modeliavimas bei informacinių technologijų paslaugų valdymo funkcijos kompiuterizavimas“. Darbo vadovas doc. dr. Jurij Tekutov. Pranešimas pristatytas Klaipėdos valstybinėje kolegijoje. Straipsnis publikuotas konferencijos straipsnių elektroniniame leidinyje:

Silkauskas L., Tekutov J. UAB „EMTN“ įmonės veiklos modeliavimas bei informacinių technologijų paslaugų valdymo funkcijos kompiuterizavimas. *Verslas, technologijos, biomedicina: inovacijų išvalgos 2016*. Klaipėda: Klaipėdos valstybinė kolegija, Nr. 1 (T. 7), 2016, p. 328–338. ISSN 2351-5902.

UAB „EMTN“ ĮMONĖS VEIKLOS MODELIAVIMAS BEI INFORMACINIŲ TECHNOLOGIJŲ PASLAUGŲ VALDYMO FUNKCIJOS KOMPIUTERIZAVIMAS

Linas Silkauskas¹, darbo vadovas dr. Jurij Tekutov^{1, 2, 3}

¹Klaipėdos universitetas, Jūros technologijų ir gamtos mokslų fakultetas, Informatikos ir statistikos katedra; ²Klaipėdos valstybinė kolegija, Technologijų fakultetas, Informacinių technologijų katedra;

³Lietuvos verslo kolegija, Informatikos katedra.

ANOTACIJA

Straipsnyje pateikta įmonės UAB „EMTN“ veiklos analizė struktūriniu-funkciniu metodu panaudojant duomenų srautų diagramas. Atliktas veiklos modeliavimas BPMN metodu, pateiktas duomenų bazės projektavimo procesas. Objektinė modeliavimo UML kalba panaudota projektuojant „Internetinių puslapių testavimo“ programą. Sudaryta panaudojimo atvejų diagrama, parodanti sukurtoje programoje įgyvendintas funkcijas, veiklos diagrama, aprašanti programos algoritmo veikimą. Pateikta sukurtos programos naudotojo dokumentacija.

Pagrindiniai žodžiai: organizacijos analizė, veiklos modeliavimas, informacinių poreikių specifikuojimas, programos projektavimas ir kūrimas, internetinių puslapių testavimas.

ĮVADAS

Programinės įrangos testavimas yra svarbi projekto sudedamoji dalis, turinti didelę įtaką kuriamam produktui. Testavimo metu siekiama aptikti klaidas anksčiau, negu programą pradeda naudoti jos galutiniai naudotojai, rasti skirtumus tarp gautų ir laukiamų rezultatų, nes kuo anksčiau aptinkama klaida, tuo jos ištaisymas kainuoja mažiau.

Nagrinėjamos įmonės dalykinė darbo sritis – interneto tinklalapių, informacinių portalų kūrimas, elektroninės komercijos sprendimai ir marketingas internete (įmonės UAB „EMTN“ tinklalapis, 2015). Internetinė svetainė testuojama naudojant kelias testavimo paslaugas: *Google PageSpeed Insights*, *W3C Markup Validation*, *WebPagetest* ir t. t. Atliekant testavimą, reikia pereiti iš vienos testavimo paslaugas teikiančios svetainės į kitą, o dėl to sugaištamas laikas. Testavimo rezultatai nėra išsaugomi, o išjungus internetinę naršyklę, testavimą reikia pakartoti. Kai testavimo paslaugas teikiantis serveris yra apkrautas, testavimas užtrunka pakankamai nemažai laiko.

Šiame tyrime pateikta „Internetinių puslapių testavimo“ programa, kuri iš testavimo paslaugų tiekėjų surenka testavimo rezultatus ir pateikia programuotojams. Testavimo rezultatai saugomi standžiajame diske *.docx* failo pavidalu arba lokaliaje *SQL Server* duomenų bazėje. Programa atlieka internetiniame puslapyje pateikiamų neoptimizuotų vaizdų suglaudinimą ir neoptimizuotų CSS ir

JavaScript failų užimamos vietos sumažinimą.

Tyrimo objektas – įmonė UAB „EMTN“.

Tyrimo tikslas – atlikus UAB „EMTN“ įmonės veiklos modeliavimą, kompiuterizuoti pasirinktą jos veiklos sritį (toku būdu sumažinti internetinės svetainės testavimo atlikimo laiką ir išvengti pakartotinių testavimų).

Tyrimo metodai: įmonės veiklos ir informacinių poreikių analizė, informacinės sistemos modeliavimas.

Straipsnio struktūra: *pirmoje dalyje* atliekama įmonės veiklos analizė struktūriniu-funkciniu metodu, veiklos modeliavimas BPMN metodu, duomenų bazės projektavimas. „Internetinių puslapių testavimo“ programos modeliavimas, taikant objektinę modeliavimo UML kalbą, pateikiamas *antroje dalyje*, o sukurtos programos dokumentacija *trečioje dalyje*.

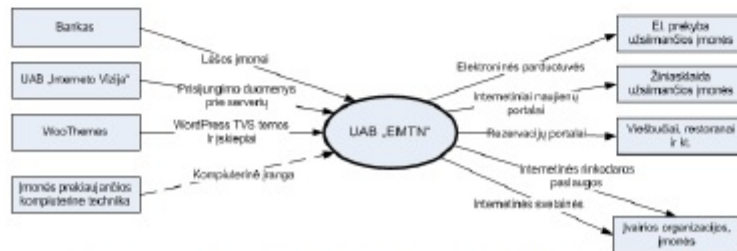
1. ORGANIZACINĖS SISTEMOS VEIKLOS ANALIZĖ IR GRAFINIAI VEIKLOS PROCESŲ MODELIAI

Šiame skyriuje pateiktas atliktas UAB „EMTN“ įmonės veiklos modeliavimas struktūriniu-funkciniu metodu (pasirinkta duomenų srautų diagramų technologija), grindžiamu sistemos dekomponavimo principu „iš viršaus žemyn“ (angl. *Top-down Approach*). Taip pat sudarytas vertės grandinės modelis, atliktas veiklos modeliavimas BPMN (angl. *Business Process Modeling Notation*) metodu, suprojektuota duomenų bazė.

1.1. Veiklos analizė struktūriniu-funkciniu metodu

Pasak S. Gudo (2008), duomenų srautų diagramos (angl. *Data Flow Diagrams*) (toliau vadinama – DFD) skirtos veiklos sričiai apibrėžti, t. y. sistemos funkcijoms (procesams) ir jų sąveikoms (technologiniams ir duomenų srautams) vaizduoti.

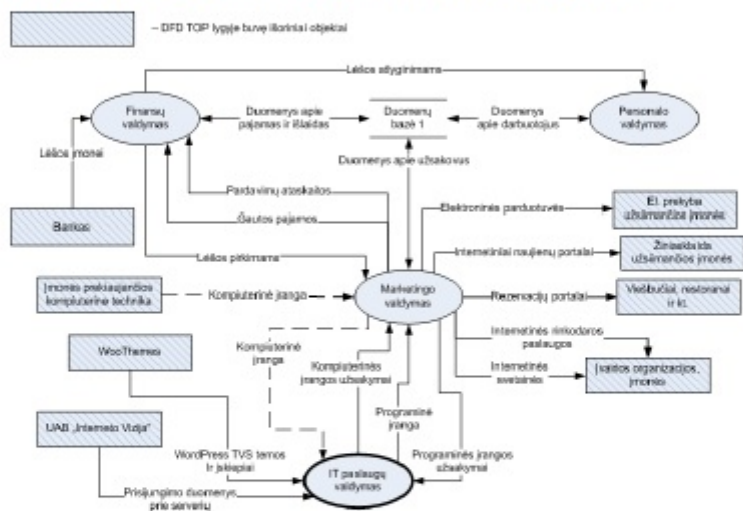
Pirmiausiai sukuriami aukščiausio (TOP) lygmens DFD, kuri aprašo nagrinėjamos įmonės sąveiką su išorine aplinka (Gudas ir Tekutov, 2008). TOP lygmens DFD pateikiama 1 paveiksle. Sudarymui pasirinktas programinis paketas „Microsoft VisioTM“ (Microsoft kompanija, 2015a).



1 pav. Aukščiausio lygmens (TOP) DFD (Microsoft VisioTM priemonės aplinkoje)

Įmonės sukurti internetiniai produktai tiekiami el. prekyba, žiniasklaida užsiimančioms įmonėms, viešbučiams, restoranams ir įvairioms kitoms organizacijoms (baldų gamintojams, viešojo maitinimo įstaigoms, automobilių dalių, poilsio prekių parduotuvėms ir t. t.). Elektroninių parduotuvių, svetainių kūrimui naudojama *WordPress* turinio valdymo sistema (TVS). Kompanija *WooThemes* įmonei tiekia šios TVS temas (tinklapių dizainų šablonus) ir įskiepius (praplečia TVS funkcionalumą). Iš įmonės UAB „Interneto Vizija“ užsakomos svetainių talpinimo ir serverių nuomos paslaugos. TVS temų, įskiepių, techninės įrangos pirkimo, darbuotojų atlyginimo lėšas įmonei tiekia bankas.

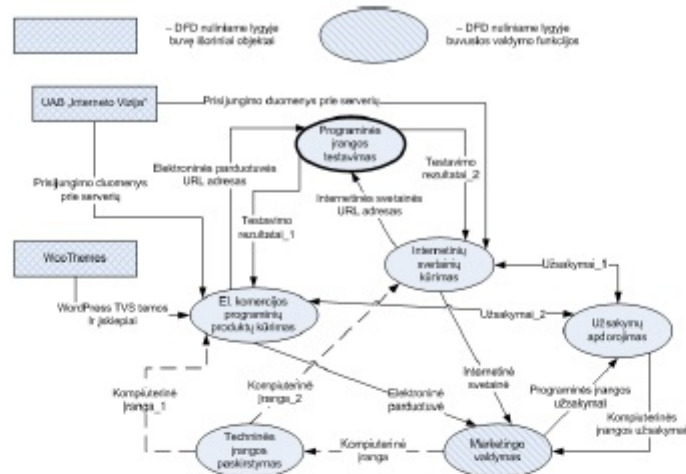
TOP lygmens DFD detalizuojama naudojant nulinio lygmens DFD. Joje pateikiamos pagrindinės valdymo funkcijos ir ryšiai tarp jų. Nulinio lygmens DFD pateikiama 2 paveiksle.



2 pav. Nulinio lygmens DFD

Įmonėje UAB „EMTN“ yra keturios pagrindinės veiklos valdymo funkcijos: marketingo, finansų, personalo ir IT paslaugų. Marketingo skyrius registruoja užsakymus iš klientų, tiekia IT paslaugų skyrius sukurtus internetinius projektus užsakovams, ieško naujų klientų ir tiekėjų. Finansų skyrius kontroliuoja pinigų srautus, atlieka buhalterinę apskaitą, įmonės pelno skaičiavimą, duomenų bazėje kaupia duomenis apie pajamas ir išlaidas, įvertina įmonės finansinę būklę. Personalo skyrius atlieka naujų darbuotojų paiešką, vykdo darbuotojų darbo valandų apskaitą, sudaro darbo grafikus, rūpinasi darbuotojų kvalifikacijos kėlimu. IT paslaugų skyrius kontroliuoja internetinių projektų kūrimo procesą, programinės įrangos testavimą.

Nulinio lygmens DFD detalizuojama naudojant 1-ojo lygmens DFD. Joje pateikiami pasirinktą valdymo funkciją sudarantys procesai. 1-ojo lygmens DFD (funkcijos „IT paslaugų valdymas“) pateikiama 3 paveiksle.

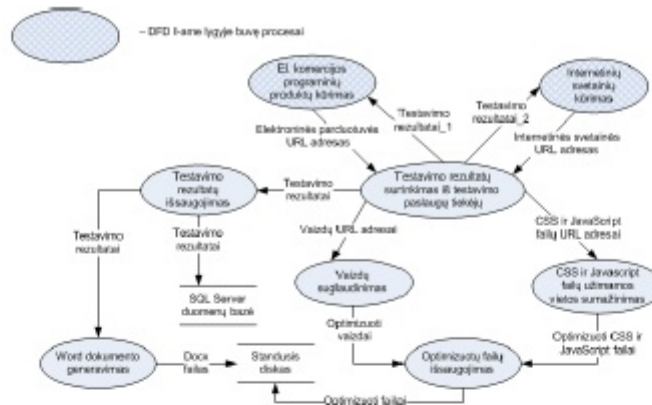


3 pav. 1-ojo lygmens DFD (funkcijos „IT paslaugų valdymas“)

Nagrinėjama IT paslaugų valdymo funkcija. Joje vykdomas techninės įrangos paskirstymas darbuotojams, atliekamas užsakovų pateiktų užsakymų apdorojimas. Priklausomai nuo pateiktų reikalavimų, užduotys suskirstomos programuotojams. Internetinių svetainių kūrimui naudojamos *PHP*, *JavaScript* programavimo kalbos, *CSS* stilių kalba, *HTML* žymių kalba, *AJAX* technologija, *MySQL* duomenų bazė ir įvairios bibliotekos: *jQuery*, *Bootstrap* ir t. t. Elektroninės (el.) komercijos programinių produktų (elektroninių parduotuvių, internetinių aukcionų, skelbimų ir t. t.) kūrimui naudojama *WordPress* turinio valdymo sistema, temos, įskiepiai ir *Magento* el. komercijos platforma. Sukurti programiniai produktai testuojami. Atlikus testavimą programinė įranga perduodama marketingo skyriui.

1-ojo lygmens DFD detalizuojama naudojant 2-ojo lygmens DFD. Joje pateikiami kompiuterizuojami uždaviniai. 2-ojo lygmens DFD (proceso „programinės įrangos testavimas“) pateikiama 4 paveiksle.

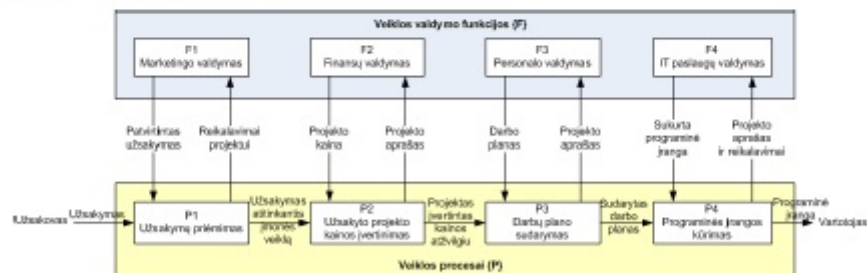
Programinės įrangos testavimas sudarytas iš 6 procesų: testavimo rezultatų surinkimo iš testavimo paslaugų tiekėjų, testavimo rezultatų išsaugojimo, *MS Word* dokumento generavimo, vaizdų suglaudrinimo, *CSS* ir *JavaScript* failų užimamos vietos sumažinimo, optimizuotų failų išsaugojimo. Pagal pateiktą internetinės svetainės *URL (Uniform Resource Locator)* adresą iš testavimo paslaugų tiekėjų surenkami testavimo rezultatai. Internetinių produktų kūrimo darbuotojai gautus testavimo rezultatus gali išsaugoti *SQL Server* duomenų bazėje, o vėliau prireikus peržiūrėti. Jeigu internetiniame puslapyje yra pateikiami neoptimizuoti, daug vietos užimantys vaizdai ir yra nesuglaudintų *CSS* ir *JavaScript* failų, tai galima įvykdyti šių failų optimizaciją ir juos išsaugoti diske. Tai atlikus galima sumažinti tinklu parsinečiamų baitų dydį, serverio atsako laiką, paspartinti tinklalapio užkrovimą.



4 pav. 2-ojo lygmens DFD (proceso „programinės įrangos testavimas“)

1.2. Vertės grandinės modeliavimas

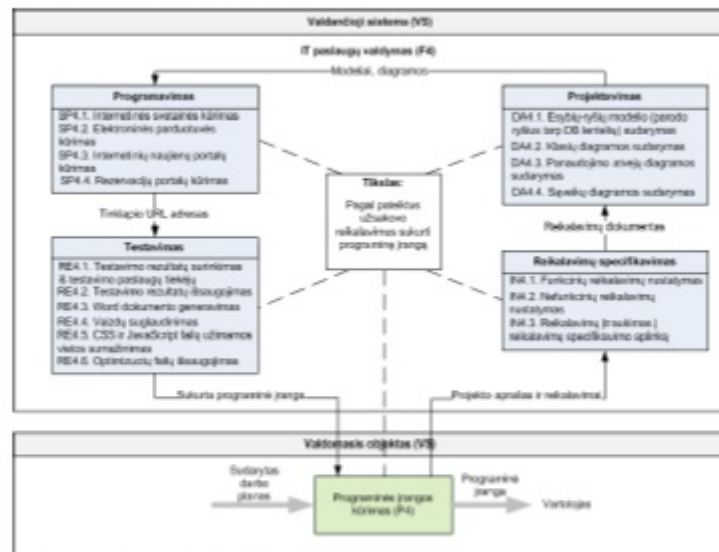
Vertės grandinės modelis (toliau vadinama – VGM) yra organizacijos veiklos modelis, kurio esmė – pagrindinių veiklos procesų bei veiklos valdymo sąveika (Gudas, 2012). Šis modelis atvaizduoja esamų momentu funkcionuojančias veiklas (procesus ir funkcijas) bei ryšius tarp jų. Apibendrintas IT paslaugų teikimo proceso informacinis modelis, naudojant VGM, pateikiamas 5 paveiksle. Marketingo valdymas atsakingas už užsakymų priėmimą (P1), peržiūrą ir patvirtinimą, finansų – už projekto kainos įvertinimą (P2), personalo – už darbų plano sudarymą (P3), o IT paslaugų – už programinės įrangos kūrimą (P4).



5 pav. Apibendrintas IT paslaugų teikimo proceso informacinis modelis naudojant VGM

Modeliuojamos veiklos valdymo funkcijos F4 = „IT paslaugų valdymas“, kuri valdo veiklos procesą P4 = „Programinės įrangos kūrimas“, elementarus valdymo ciklas išdėstomas 6 paveiksle. Programinės įrangos kūrimo veiklos procesas (P4) susideda iš keturių etapų: reikalavimų specifikavimo, projektavimo, programavimo ir testavimo. Reikalavimų specifikavimo etape iš užsakovo pateikto projekto aprašo ir bendrų reikalavimų nustatomi funkciniai ir nefunkciniai

reikalavimai, kurie įtraukiami į reikalavimų specifikavimo aplinką. Reikalavimų dokumentas perduodamas projektavimo darbams. Projektavimo metu sudaromi esybių-ryšių, klasių, panaudojimo atvejų ir sąveikų diagramos, kurios perduodamos programuotojams. Atliekamas programinių produktų kūrimas. Sukurta programinė įranga testuojama.

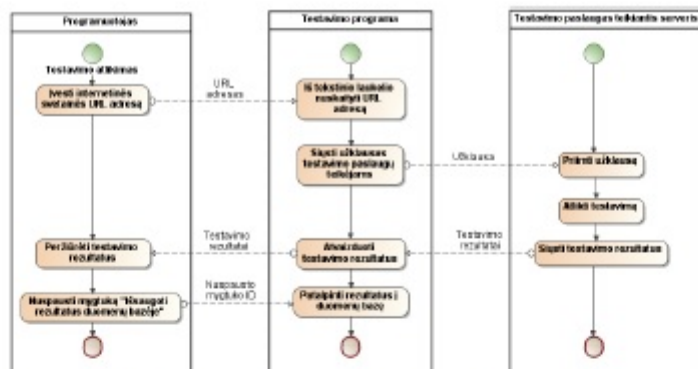


6 pav. Detalus valdymo funkcijos F4 = „IT paslaugų valdymas“ ir proceso P4 = „Programinės įrangos kūrimas“ sąveikos modelis

1.3. Veiklos modeliavimas BPMN metodu

BPMN notacijos paskirtis yra standartizuoti veiklos procesų modeliavimą (OMG konsorciumo tinklalapis, 2015), įvertinant daugumos populiarių ir skirtingų procesų modeliavimo notacijų privalumus, tokių kaip *UML Activity Diagram*, *IDEF*, *Activity-Decision Flow (ADF) Diagram*, *Event-Process Chains (EPCs)* ir kitų. Veiklos modelis BPMN notacija parodomas 7 paveiksle.

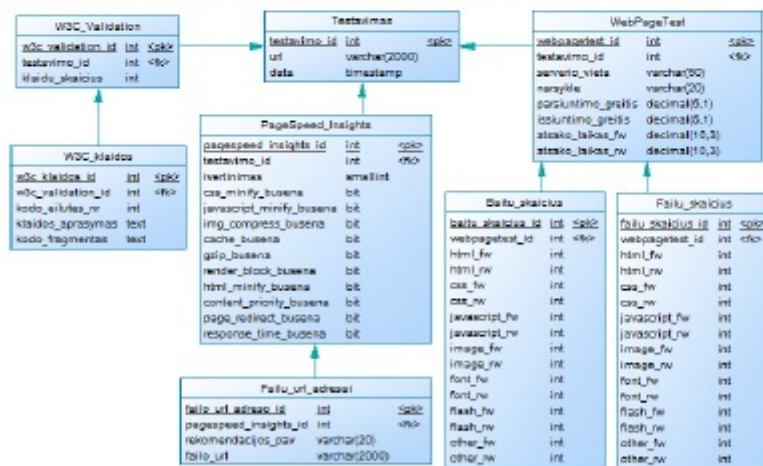
Norint atlikti internetinės svetainės testavimą programoje reikia įvesti URL adresą. Testavimo programa iš tekstinio laukelio nuskaito URL adresą ir nusiunčia užklausas *Google PageSpeed Insights*, *W3C Markup Validation* ir *WebPagetest* serveriams. Serveriai priima užklausą, atlieka testavimą ir grąžina programai testavimo rezultatus. Programa gautus rezultatus atvaizduoja programuotojui. Jam peržiūrėjus ir nuspaudus rezultatų išsaugojimo duomenų bazėje mygtuką, programa rezultatus patalpina į *SQL Server* duomenų bazę.



7 pav. Veiklos modelis sudarytas naudojant BPMN notaciją (*MagicDraw™* priemonės aplinkoje)

1.4. Duomenų bazės projektavimas

Fizinis duomenų modelis – tai toks modelis, kurį kuriant yra žinomas ne tik duomenų bazės (DB) tipas (kaip loginio modeliavimo atveju), bet ir konkreti DB valdymo sistema. Sudaromas fizinis duomenų modelis, nusakantis DB lenteles ir ryšius tarp jų, kuris pateikiamas 8 paveiksle.



8 pav. Fizinis duomenų modelis (*PowerDesigner* priemonės aplinkoje)

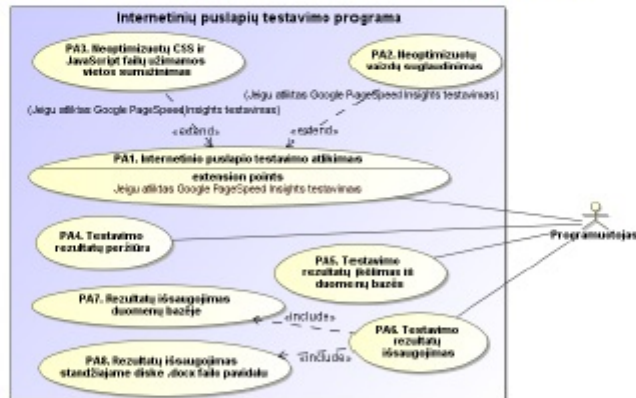
Testavimo duomenų saugojimui naudojamos 8 duomenų bazės lentelės. Bendroji su testavimu susijusi informacija (svetainės URL adresas, atlikimo data) saugoma lentelėje „Testavimas“. „PageSpeedInsights“ lentelėje saugomos *Google* rekomendacijų internetiniam puslapiui būsenos. Būseną „true“ reiškia, kad testuojamas internetinis puslapis tenkina rekomendaciją, „false“ – priešingai. Taip pat šioje lentelėje nurodomas puslapio įvertinimas intervale nuo 0 iki 100. Lentelėje

„Failu_url_adresai“ saugomi neoptimizuotų failų URL adresai. Lentelės „W3C_Validation“ ir „W3C_klaidos“ naudojamos saugoti klaidų skaičių, jos aprašymą, programinio kodo eilučių numerius ir fragmentus, kuriuose aptikta klaida. Lentelėje „WebPageTest“ saugoma informacija susijusi su serveriu (vieta, naudota naršyklė, greičiai) ir serverio atsako laikas puslapio pirmos ir pakartotinos peržiūros metu. Dydžiai nurodantys kiek tam tikrų failų buvo parsųsta iš nurodyto puslapio ir šių failų užimama vieta baitais, saugomi lentelėse „Baitu_skaicius“ ir „Failu_skaicius“.

Tokiu būdu šiame tyrime bus kompiuterizuojama informacinių technologijų paslaugų valdymo funkcija, t. y. kuriama „internetinių puslapių testavimo“ programa.

2. „INTERNETINIŲ PUSLAPIŲ TESTAVIMO“ PROGRAMOS MODELIAVIMAS TAIKANT OBJEKTINĘ MODELIAVIMO UML KALBĄ

Panaudojimo atvejų diagrama naudojama identifikuoti pirminius sistemos elementus ir procesus, kurie formuoja projektuojamą programą (Nemuraitė, 2008). Sudaroma „Internetinių puslapių testavimo“ programos panaudojimo atvejų (angl. *Use Case*) diagrama, kuri pateikiama 9 paveiksle. Sudarymui pasirinktas programinis paketas „MagicDrawTM“ (NO MAGIC kompanija, 2015).

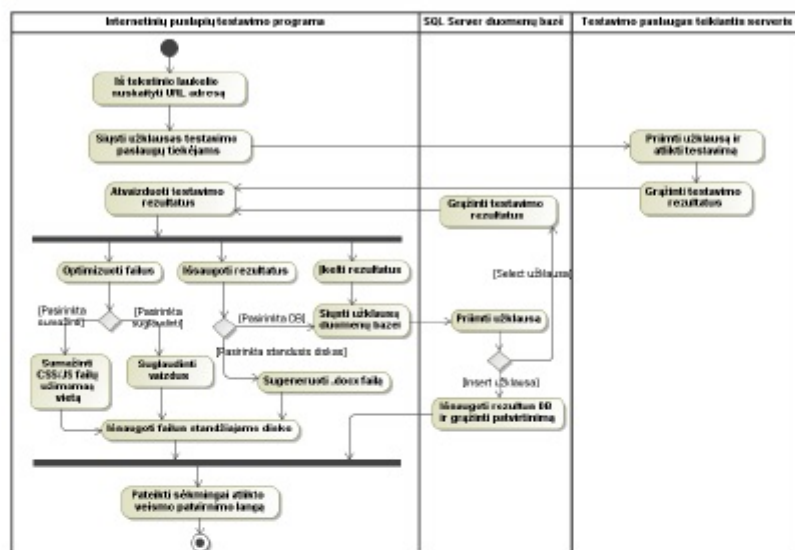


9 pav. Internetinių puslapių testavimo programos panaudojimo atvejų diagrama (MagicDrawTM priemonės aplinkoje)

Vienas iš elgsenos diagramų tipų – veiklos diagramos (angl. *activity diagram*), kurios vaizduoja darbų srautus sistemoje ir aprašo objektų kitimą juos apdorojant, kokios nors užduoties realizavimo būdą. „Internetinių puslapių testavimo“ programos veiklos diagrama pateikiama 10 paveiksle.

Pateikta veiklos diagrama aprašo šiame tyrime sukurtos testavimo programos veikimo algoritmą. Iš tekstinio laukelio nuskaitytas internetinio puslapio URL adresas nusiunčiamas testavimo paslaugas teikiantiems serveriams (*Google, W3C, Webpagetest*). Juose atliekamas testavimas ir grąžinami

testavimo rezultatai programai, kuri juos atvaizduoja. Po to priklausomai nuo programuotojo pasirinkto veiksmo, optimizuojami failai (sumažinama *CSS* arba *JavaScript* failų užimama vieta, suglaudunami vaizdai), išsaugomi rezultatai (diske *.docx* pavidalu, DB) arba iš DB įkeliami prieš tai išsaugoti rezultatai.



10 pav. „Internetinių puslapių testavimo“ programos veiklos diagrama

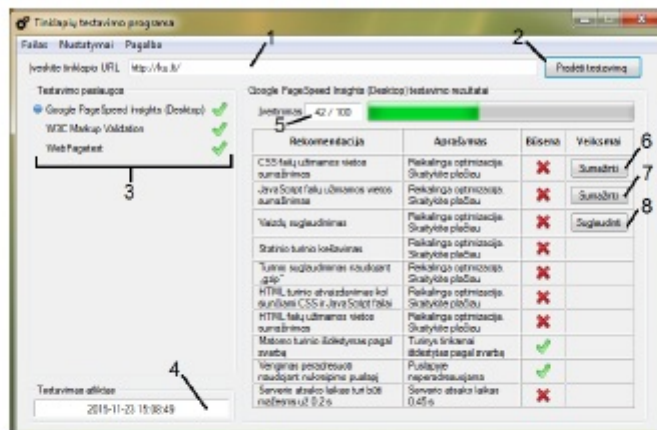
3. „INTERNETINIŲ PUSLAPIŲ TESTAVIMO“ NAUDOTOJO DOKUMENTACIJA

Sukurta programa leidžia atlikti internetinio puslapio testavimą nurodant jo URL adresą (Microsoft kompanijos tinklalapis, 2015b). Programa iš testavimo paslaugų tiekėjų (*Google PageSpeed Insights*, *W3C Markup Validation*, *WebPagetest*) surenka testavimo rezultatus ir juos atvaizduoja programos blokuose. Testavimo rezultatus galima išsaugoti standžiajame diske *.docx* failo pavidalu arba lokaloje *SQL Server* duomenų bazėje. Taip pat sukurta programa leidžia atlikti internetiniame puslapyje pateikiamų neoptimizuotų vaizdų suglaudimą ir neoptimizuotų *CSS* ir *JavaScript* failų užimamos vietos sumažinimą.

„Internetinių puslapių testavimo“ pagrindinis programos langas pateiktas 11 paveiksle.

Norint atlikti internetinio puslapio testavimą į 1 numeriu pažymėtą tekstinį laukelį reikia įvesti puslapio URL adresą ir nuspausti 2 numeriu pažymėtą mygtuką. Jeigu testavimą sėkmingai pavyko atlikti, ties meniu punktais (3) pateikiamas žalios spalvos patvirtinimas. Nuspaudus ant tam tikro meniu punkto (3) pateikiami testavimo rezultatai, kurie buvo gauti iš atitinkamo testavimo paslaugų

tiekejo (*Google, W3C, Webpagetest*). Testavimą gali nepavykti atlikti jeigu nurodytas neteisingas tinklalapio URL adresas, testavimo serveris tuo momentu išjungtas arba didelė jo apkrova. *Webpagetest* serverio rezultatų kartais reikia ilgiau palaukti. 4 numeriu pažymėtoje srityje pateikiama testavimo atlikimo data ir laikas. Kadangi šiuo atveju pasirinktas pirmas meniu punktas, todėl 5 numeriu pažymėtoje srityje pateikiamas puslapio įvertinimas, kuris gautas iš *Google*.



11 pav. „Internetinių puslapių testavimo“ pagrindinis programos langas

Lentelės pirmajame stulpelyje pateikiamos rekomendacijos puslapiui. Jeigu jis tenkina rekomendaciją, tai trečio stulpelio atitinkamoje eilutėje pateikiamas žalios spalvos patvirtinimas. Nuspaudus lentelėje ant užrašo „Reikalinga optimizacija ...“ pateikiama programos langas, kuriame atvaizduojami puslapyje esančių failų URL adresai, kuriems reikalinga optimizacija.

IŠVADOS

1. Atlikus įmonės UAB „EMTN“ veiklos analizę struktūriniu-funkciniu metodu paaiškėjo, kad reikia kompiuterizuoti informacinių technologijų paslaugų valdymo funkciją. Nustatyta, kad įmonėje dirbantys programuotojai internetinius puslapius testuoja naudojantis skirtingose svetainėse pateikiamomis testavimo paslaugomis. Kadangi reikia pereiti iš vienos testavimo paslaugos teikiančios svetainės į kitą, sugaištamas laikas. Testavimo rezultatai nesaugomi, testavimą reikia pakartoti jeigu internetinė naršyklė išjungžiama ir po to vėl įjungžiama.
2. Sukurta programa, kuri iš testavimo paslaugų tiekėjų surenka testavimo rezultatus ir juos pateikia programuotojams. Testavimo rezultatai saugomi standžiajame diske .docx failo pavidalu arba lokaliaje SQL Server duomenų bazėje. Programa leidžia atlikti internetiniame puslapyje pateikiamų neoptimizuotų vaizdų suglaudinimą ir neoptimizuotų CSS ir JavaScript failų užimamos vietos sumažinimą. Sukurta programa leidžia sumažinti internetinės svetainės

testavimo atlikimo laiką ir išvengti pakartotinių testavimų.

Šis straipsnis prisideda prie *Klaipėdos valstybinėje kolegijoje* vykdomo taikomojo mokslinio tyrimo „Veiklos valdymo modelių taikymas žiniomis grindžiamoje informacinių sistemų inžinerijoje“ (veiklos sritis: darnios aplinkos vystymas; kryptis: inovatyvios technologijos; tematika (šaka): išmaniosios komunikacijų technologijos, TMV IF-027).

SUMMARY

In this paper presented activity analysis of UAB „EMTN“ company using structural-functional method. Analysis was done using data flow diagrams (DFD) notation. Created IT services process information model using value chain model and presented elementary management cycle of activity management function. Performed activity modeling using BPMN method and database design. „Web page testing“ program modeling was done using objective modeling UML language. Program functions are shown in use case diagram, program algorithm are presented in activity diagram. Also described user documentation of created program.

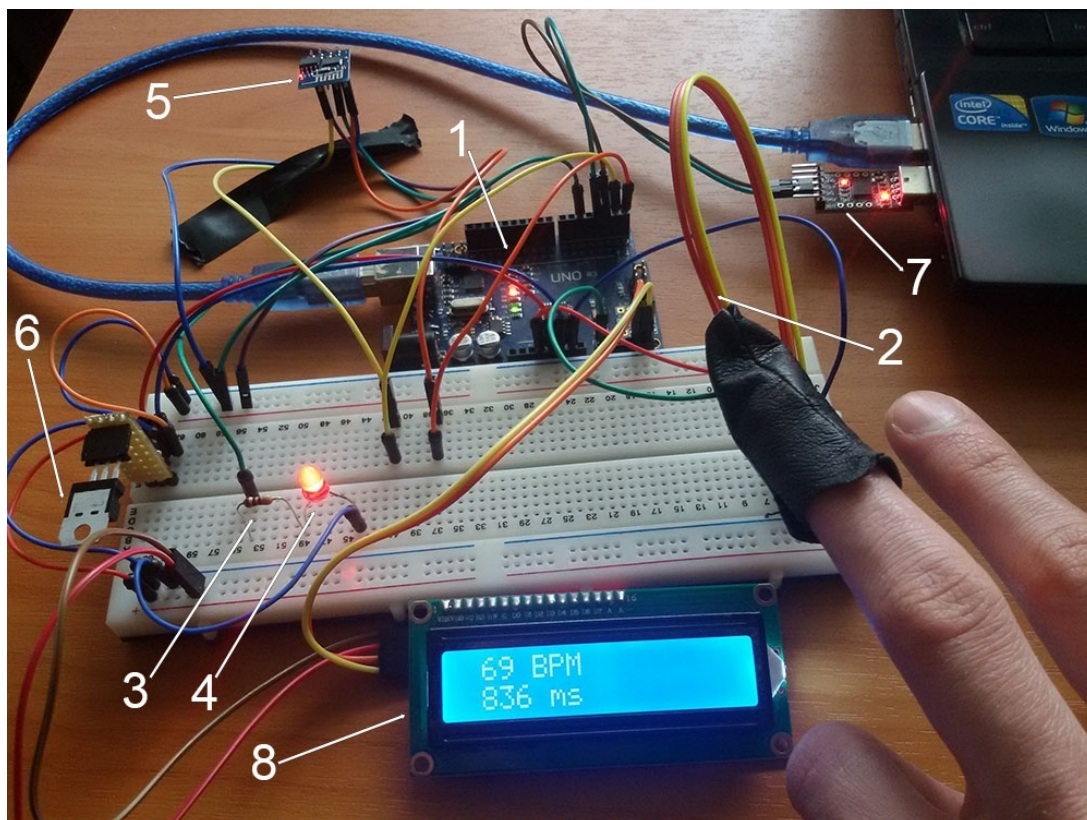
LITERATŪRA

1. GUDAS, S. *Organizacijų veiklos modeliavimo praktiniai darbai*. Klaipėda: Klaipėdos universiteto leidykla, 2008, 201 p. ISBN 978-9955-18-348-8.
2. GUDAS, S. *Informacijos sistemų inžinerijos teorijos pagrindai*. Monografija. Vilnius: Vilniaus universiteto leidykla, 2012, 384 p. ISBN 978-609-459-075-7.
3. GUDAS, S; ir TEKUTOV, J. *Organizacijų veiklos modeliavimas*: el. metodinė medžiaga, nuotolinis kursas, T120M015 [interaktyvus]. Klaipėda: Klaipėdos universitetas, 2008 [žiūrėta 2015 m. gruodžio 11 d.]. Prieiga per internetą: <<http://vma.ku.lt>>.
4. ĮMONĖS UAB „EMTN“ TINKLALAPIS. *Informacija apie nagrinjamą įmonę*. 2015 [žiūrėta 2015 m. rugsėjo 22 d.]. Prieiga per internetą: <<http://www.emotion.lt/>>.
5. OMG KONSORCIUMO TINKLALAPIS. *Business Process Model and Notation (BPMN): Version 2.0* [interaktyvus]. 2015 [žiūrėta 2015 m. lapkričio 18 d.]. Prieiga per internetą: <<http://www.omg.org/spec/BPMN/2.0/>>.
6. MICROSOFT KOMPANIJOS TINKLALAPIS. *MS Visio priemonės informacija* [interaktyvus]. 2015 [žiūrėta 2015 m. rugsėjo 24 d.]. Prieiga per internetą: <<http://office.microsoft.com/en-us/visio/>>.
7. MICROSOFT KOMPANIJOS TINKLALAPIS. *C# programavimo kalbos specifikacija*. 2015 [žiūrėta 2015 m. spalio 17 d.]. Prieiga per internetą: <<https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>>.
8. NEMURAITĖ, L. *Informacinių sistemų programinės įrangos projektavimas*. Klaipėda: Klaipėdos universiteto leidykla, 2008, 381 p. ISBN 978-9955-18-347-1.
9. NO MAGIC, INC KOMPANIJOS TINKLALAPIS. *MagicDraw priemonės specifikacija* [interaktyvus]. No Magic, Inc, 2015 [žiūrėta 2015 m. lapkričio 14 d.]. Prieiga per internetą: <<http://www.nomagic.com/products/magicdraw.html>>.

3 PRIEDAS

Užfiksuotų parametrų atvaizdavimas sukurtoje programoje naudojant .NET

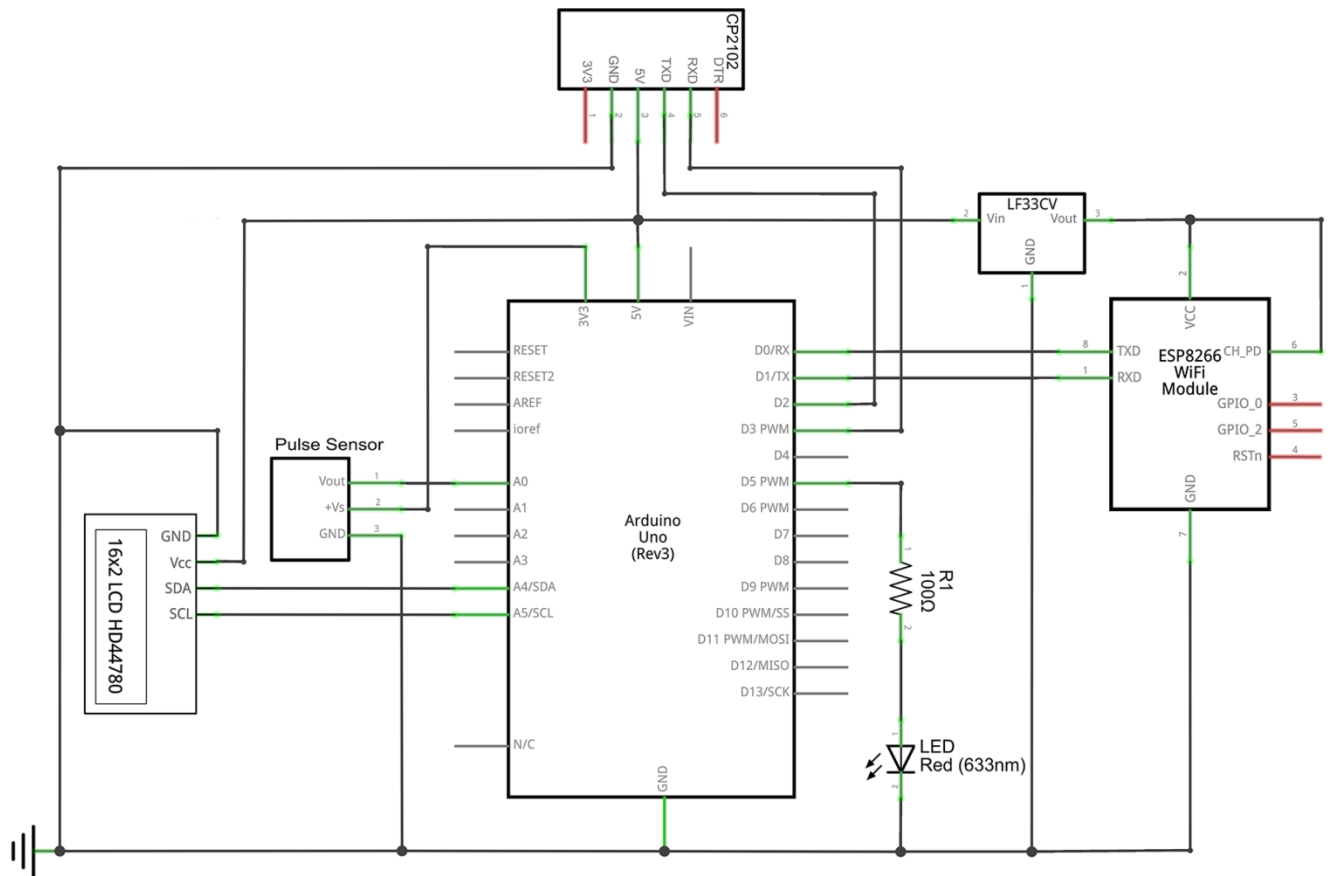
Nuotolinės sveikatos stebėjimo sistemos įgyvendinimas buvo pradėtas nuo SEN-11574 pulso jutiklio duomenų atvaizdavimo programoje, kuri buvo kuriama naudojant C# programavimo kalbą, .NET 4.5.1 karkasą, Visual Studio 2013 IDE ir Visual Micro 1603.30.4 papildinį (angl. extension). Pateikiama suprojektuota techninė įranga naudojant pulso jutiklį:



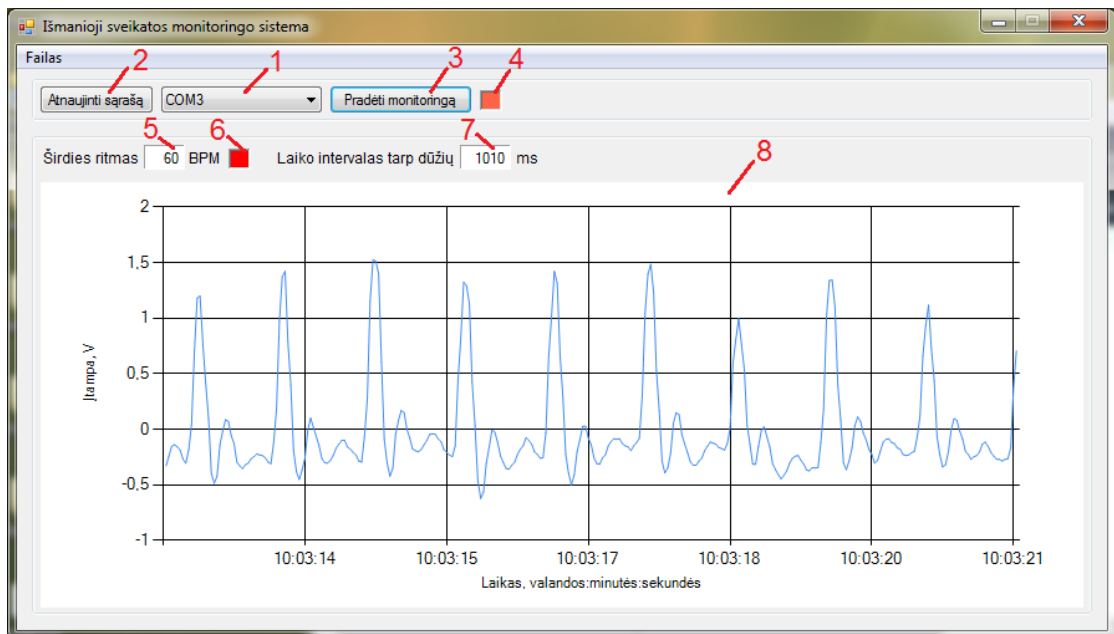
Projektavimui naudota techninė įranga:

1. Arduino Uno R3 valdiklių plokštė;
2. SEN-11574 pulso jutiklis;
3. $100\ \Omega$ rezistorius;
4. šviesos diodas;
5. ESP8266-01 Wi-Fi modulis;
6. LF33CV įtampos reguliatorius (5 V į 3.3 V);
7. CP2102 USB-UART keitiklis;
8. LCD ekranas HD44780.

Pateikiama nubraižyta elektroninė schema:



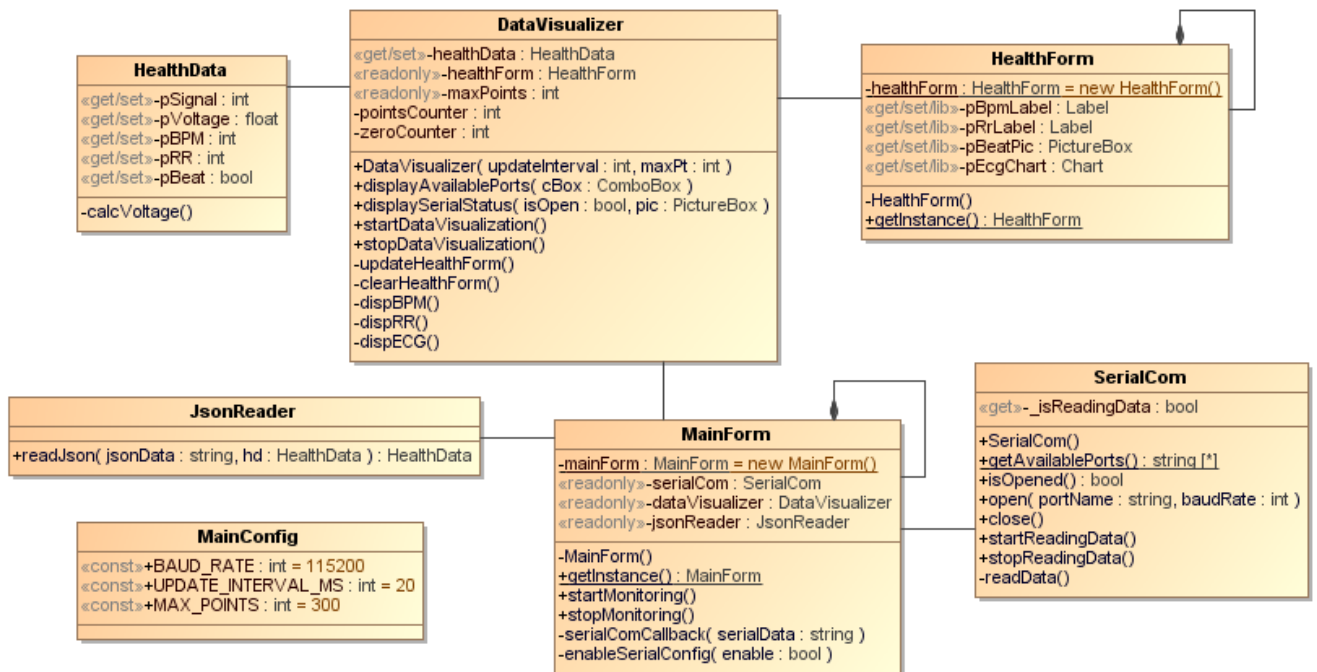
Užfiksuotų parametrų atvaizdavimas sukurtoje programoje (duomenų atnaujinimas programoje vykdomas kas 20 ms):



1. iškrentantis sąrašas su prievadais (angl. *ports*);
2. mygtukas skirtas atnaujinti prievadų sąrašą;
3. mygtukas skirtas pradėti arba sustabdyti duomenų atvaizdavimą;
4. informuojantis blokas: raudona – atvaizdavimas ne pradėtas arba sustabdytas, žalia – atvaizdavimas pradėtas;

- laukelis, kuriame pateikiamas širdies ritmas;
- blokas, kuris informuoja kada užfiksuotas naujas širdies dūžis;
- laukelis, kuriame pateikiamas laiko intervalas tarp dūžių;
- sritis, kurioje atvaizduojama elektrokardiograma.

Programos klasių diagrama:



Klasių paskirtis:

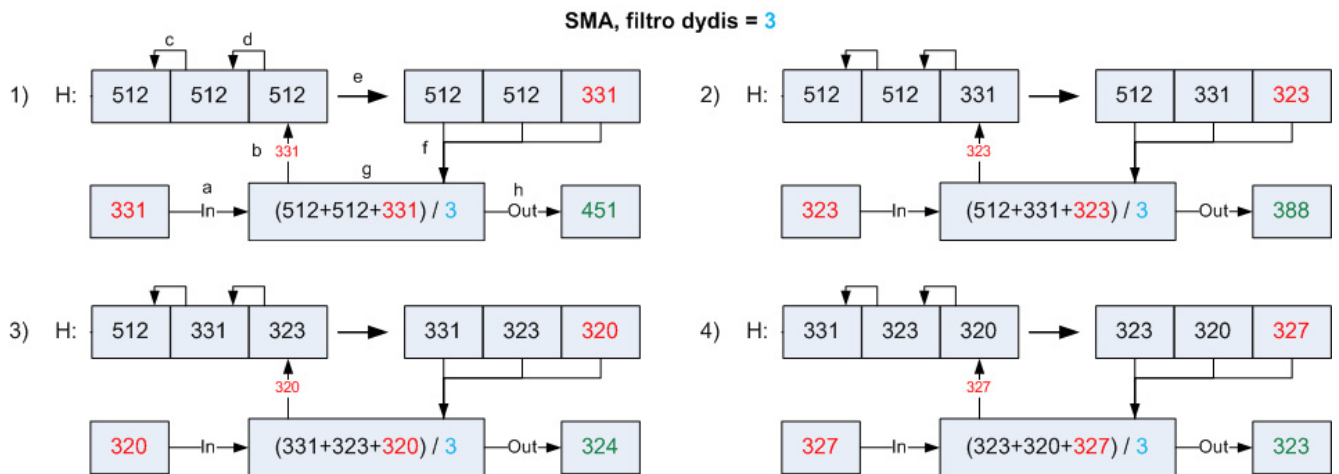
- MainConfig – saugomos programos konstantos (UART sąsajos greitis, duomenų atvaizdavimo intervalas, maksimalus taškų skaičius elektrokardiogramoje);
- SerialCom – atsakinga už kompiuterio komunikaciją su Arduino Uno valdiklių plokšte;
- JsonReader – naudojama nuskaityti JSON duomenų objektą, kuris buvo gautas iš Arduino Uno plokštės. Iš JSON objekto nuskaitytų duomenų formuojamas HealthData klasės objektas.
- HealthData – joje saugomi iš jutiklio gauti sveikatos parametrai;
- DataVisualizer – naudojama jutiklių duomenų atvaizdavimui;
- HealthForm – joje saugomi vartotojo sąsajos laukelių objektai;
- MainForm – pradedamas programos vykdymas. Ji apjungia klases, kurios atsakingos už UART komunikaciją, JSON duomenų objekto nuskaitymą ir duomenų atvaizdavimą.

MainForm ir HealthForm panaudotas singletono projektavimo šablonas. Siekiama, kad formos duomenų prieinamumas būtų visose klasėse.

SMA, EMA ir CMA filtravimo algoritmų veikimo principas

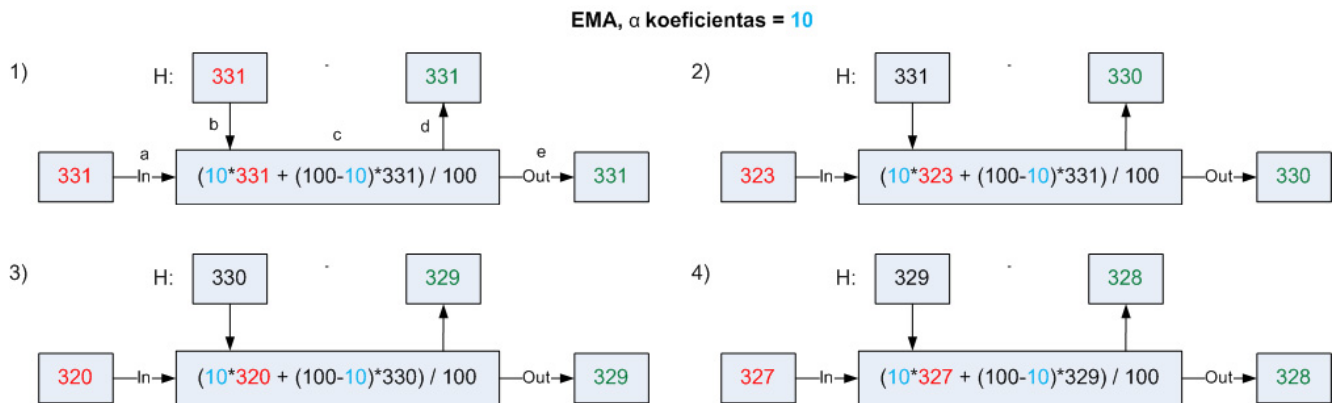
Išnagrinėti eksperimentinėje dalyje EKG signalo filtravimui pasirinkti algoritmai: SMA, EMA ir CMA. Jų veikimo principas pateikiamas vizualiniu būdu. EKG signalo reikšmė kinta nuo 0 iki 1023.

- paprastasis slenkančio vidurkio (angl. *simple moving average*, SMA):



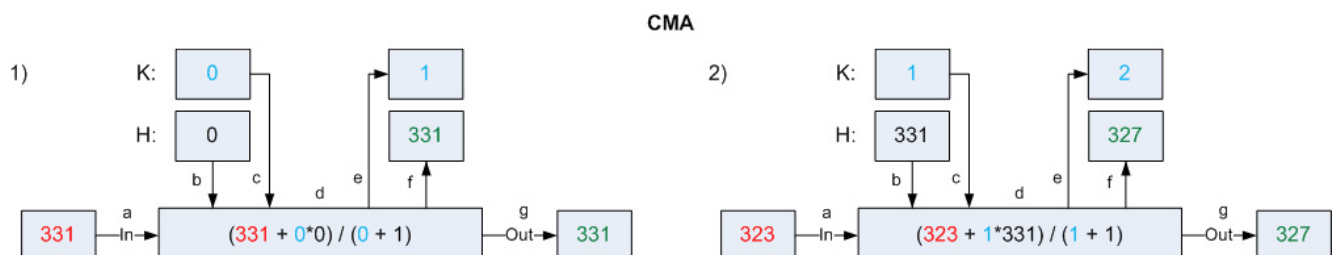
Kur, H – prieš tai buvusios įėjimo reikšmės (masyvas pradžioje užpildomas 512, tačiau galima ir kita reikšme), In – įėjimas (signalo reikšmė prieš filtravimą), Out – išėjimas (signalo reikšmė po filtravimo). SMA filtro dydis gali būti lygus 2, 3, 4 ir t. t.

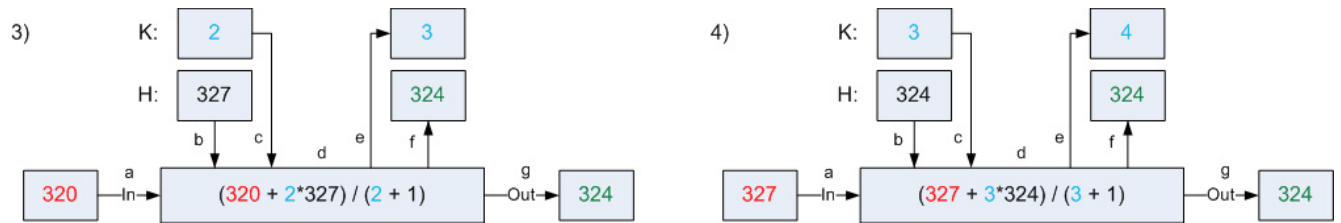
- eksponentinis slenkančio vidurkio (angl. *exponential moving average*, EMA):



Kur, H – prieš tai buvusi išėjimo reikšmė (kintamasis pradžioje lygus įėjimui), In – įėjimas (signalo reikšmė prieš filtravimą), Out – išėjimas (signalo reikšmė po filtravimo). EMA α koeficientas lygus nuo 0 iki 99.

- kaupiamasis slenkančio vidurkio (angl. *cumulative moving average*, CMA):





Kur, H – prieš tai buvusi išėjimo reikšmė (kintamasis pradžioje lygus 0), K – kaupiklis (pradžioje lygus 0, po to didinamas po 1), In – įėjimas (signalo reikšmė prieš filtravimą), Out – išėjimas (signalo reikšmė po filtravimo).