

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Aivaras Šimulis

**SVEIKATOS PRIEŽIŪROS PASLAUGŲ PERKĖLIMAS Į
ELEKTRONINĘ ERDVĘ**

Baigiamasis bakalauro projektas

Vadovas
Doc. Dalius Makackas

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

**PRIEŽIŪROS PASLAUGŲ PERKĖLIMAS Į ELEKTRONINĘ
ERDVĘ**

Baigiamasis bakalauro projektas
Informatika (612I10004)

Vadovas
Doc. Dalius Makackas

Recenzentas

Projektą atliko
Aivaras Šimulis

KAUNAS, 2015

TECHNINĖ UŽDUOTIS

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Kompiuterių tinklų katedra

TVIRTINU: _____

KTU Informatikos fakulteto

katedros vedėjas

201 5 m. _____ d

BAKALAURO BAIGIAMOJO DARBO UŽDUOTIS

1. Bakalauro kvalifikacinio darbo tema:

Sveikatos priežiūros paslaugų perkėlimas į elektroninę erdvę.

2. Bakalauro kvalifikacinio darbo pateikimo terminai:

vadovui	201	<u>5</u>	m.	_____	_____	d
recenzentui	201	<u>5</u>	m.	_____	_____	d
kvalifikacinei komisijai	201	<u>5</u>	m.	_____	_____	d

3. Funkciniai reikalavimai kuriamam objektui arba sistemai:

- Pacientų nuotolinių stebėjimų ir įrenginių modulis
- Pacientų informavimo ir apsilankymų modulis
- Pacientų mokymų ir švietimo modulis

4. Nefunkciniai reikalavimai kuriamam objektui arba sistemai:

- Klientinė sistemos dalis turi būti prieiname per populiariausias interneto naršykles.
- Sistemos architektūra turi būti realizuota ne mažiau kaip pagal trijų sluoksnį: vaizdavimo lygmenį; veiklos logikos lygmenį; duomenų bazės lygmenį.
- Sistemos duomenys turi būti saugomi reliacinėje duomenų bazių valdymo sistemoje (DBVS).

5. Reikalavimai operacinei sistemai:

Naudojamas serveris – MS Windows Server 2008R2/2012 (IIS Internet Information Server).

6. Reikalavimai programinei įrangai:

- Visual Studio 2013 SP4, Microsoft SQL Server 2012
- Visos plačiai paplitusios interneto naršyklės (ir mobilių įrenginių) bei jų įvairios versijos.
- Java Plug-in (Java Runtime Environment (JRE) komponentas).

7. Reikalavimai tinklinei įrangai ir protokolams:

Elektroninio pašto sistemos integravimui – SMTP, HTTPS protokolai.

8. Reikalavimai testavimui:

Sistema bus testuojama rašant modulių testus ir rankiniu būdu (pagal juodosios dėžės principą).

Kvalifikacinio darbo vadovas _____ 201 5 m. vasario 02 d

Dalius Makackas _____ (parašas)

(Vardas pavardė)

Studentas (studentai) _____ 201 5 m. vasario 02 d

(parašas, parašai)

AKADEMINIO SĄŽININGUMODEKLARACIJA



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos

(Fakultetas)

Aivaras Šimulis

(Studento vardas, pavardė)

Informatika (612I10004)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 15 m. gegužės 22 d.
Kaunas

Patvirtinu, kad mano, **Aivaro Šimlio**, baigiamasis projektas tema „Sveikatos priežiūros paslaugų perkėlimas į elektroninę erdvę“ yra parašytas visiškai savarankiskai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Istatymų nenumatyty piniginių sumų už šį darbą niekam nesu mokėjės.

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

(vardą ir pavardę išrašyti ranka)

(parašas)

Šimulis, A. Sveikatos priežiūros paslaugų perkėlimas į elektroninę erdvę. *Bakalaurinės* baigiamasis projektas / vadovas doc. Dalius Makackas; Kauno technologijos universitetas, Informatikos fakultetas.

Kaunas, 2015. 89 p.

SANTRAUKA

Baigiamasis darbas „Sveikatos priežiūros paslaugų perkėlimas į elektroninę erdvę“ – tai dalis naujosios Kauno klinikų informacinės medicininės sistemos, kuri buvo kuriamą bendradarbiaujant įmonėms – UAB „Amidus“ ir UAB „Blue Bridge“. Šioje sistemoje buvo siekiama pagal užsakovo reikalavimus sukurti tris norimą sveikatos priežiūros paslaugų modulius: nuotolinių stebėjimų, apsilankymų ir švietimo. Sukurtieji moduliai sieks pagerinti Kauno klinikų teikiamą sveikatos priežiūros paslaugų kokybę, sukurti galimybes taupyti lėšas, laiką bei pinigus, sumažinti administracinię naštą ir padidinti Kauno klinikų paslaugų pasiekiamumą ir prieinamumą. Projekto metu į elektroninę erdvę perkeltos paslaugos sukurs papildomą naudą tiek Kauno klinikoms, tiek šios įstaigos pacientams.

Kuriant sistemą ir įgyvendinant jos modulius buvo atlikta panašių sistemų, bei technologijų analizė. Panašių sistemų analizėje pagal pasirinktus kriterijus nuspresta ar verta kurti tokią sistemą, kuo ji yra pranašesnė už jau esamas. Technologijų analizė buvo sudaryta iš programinės įrangos analizės. Programinės įrangos analizėje pagal pasirinktus kriterijus palygintos skirtinės technologijos, duomenų bazės valdymo sistemos, programinės bibliotekos ir standartai, internetinių servisių kūrimo, bei taikymų technologijos. Po atliktos analizės - suprojektuota sistema ir jos moduliai. Apibrėžti sistemos aktoriai: pacientas, atsakingas gydytojas, specialistas, darbuotojas, įrenginių ir mokymų administratorius, serveris, bendradarbiaujanti sistema. Aprašyti aktorių panaudos atvejai, sudaryti funkciniai, bei išskirti nefunkciniai sistemos modulių reikalavimai. Iš sudarytų reikalavimų suprojektuoti sistemos moduliai – sudarytos funkcijų hierarchijos, sekų diagramos, duomenų bazės aprašo schema ir klasių esybų diagramos. Pagal aprašytą projektą sistema realizuota į prototipą, ataskaitoje pateikti realūs vartotojo sąsajos vaizdai. Realizuotas prototipas buvo testuojamas naudojant sudarytą testavimo planą ir paruoštus testavimo atvejus. Pavieniai komponentai buvo testuoti juodos dėžės principu, o komponentų, modulių integracija atlikta „iš apačios į viršų“ integracijos testavimo metodu. Pagal sudarytą planą ir paruošus testinius atvejus sistemos moduliai buvo ištестuoti. Sukurtas ir aprašytas sistemos modulių naudojimo vadovas. Išvadose aprašyta, kas buvo padaryta, pateiktos papildomos pastabos ir pastebėti dėsningumai.

SUMMARY

„The provision of health care services on the internet“ - is a health management information system, which according to the customer's requirements has the desired health care services (patient remote observation, visits, education) functions and is developed by UAB „Amidus“ ir UAB „Blue Bridge“ corporations. The developed system modules will improve Kaunas clinics provided health care services, to create opportunities for cost savings, time and money, reduce the administrative burden and increase Kaunas clinics service availability and accessibility. The created health care services will create an added value for both patients and clinics workers.

Similar systems analysis, and technology analysis was made. Systems analysis was based on the selected criteria and it was decided whether it was worthwhile to develop such a system, as it will be superior to the existing ones. Technology analysis comprises software analysis. Software analysis, based on the selected criteria were based on different technologies of database management systems, software libraries and standards, web services, and web application technologies. When the system analysis was done, system actors were defined (patient, doctor, specialist, worker, devices administrator, education administrator, server, communicating server) and use cases drawn functional system requirements. Also non-functional system requirements were defined. When requirements were done, components diagram, sequences diagrams, database schema and the class diagrams were created. With the project implemented in a prototype system, the report provides a realistic user interface views. The realized prototype was tested using the test plan and test cases. Individual components were tested by using black box approach and the integration of components were tested by using “bottom-up” integration testing method. The conclusions describe what has been done and additional notes or note patterns.

TURINYS

Lentelių sąrašas	9
Paveikslų sąrašas.....	10
Terminų ir santrumpų žodynas	12
Įvadas	13
1. Analizė	14
1.1. Įmonės analizė.....	15
1.1.1. Bendra informacija.....	15
1.1.2. Informacija apie įmonėje naudojamas IT.....	15
1.1.3. Įmonės veiklos procesai	16
1.2. Lietuvoje sukurtų panašių sistemų analizė.....	16
1.2.1. Sveikatingumo sistema „pasveik.lt“	17
1.2.2. Medicininė registracijos sistema „sergu.lt“	18
1.2.3. Medicininė informacinė sistema „santa.lt“	19
1.2.4. Medicininė informacinė sistema „kaunoklinikos.lt“.....	22
1.2.5. Kuriamoji Kauno klinikų medicininė informacinė sistema ir jos palyginimas su analizuojamomis sistemomis	23
1.3. Galimų sprendžiamo uždavinio įgyvendinimo priemonių variantų analizė	25
1.3.1. Programinių įrankių analizė	26
1.3.2. Technologijų analizė	29
1.3.3. Protokolų analizė	33
1.4. Analizės išvados.....	34
2. Projektas.....	35
2.1. Reikalavimų specifikacija	35
2.1.1. Projekto apribojimai.....	35
2.1.2. Funkciniai reikalavimai	36
2.1.3. Nefunkciniai reikalavimai.....	42
2.1.4. Techninė specifikacija.....	44
2.2. Sistemos modulių projektas	45
2.2.1. Sistemos loginė architektūra	45
2.2.2. Sistemos modulių veiklos logika	47
2.2.3. Sistemos modulių dalykinės srities esybių klasės.....	52
2.2.4. Sistemos modulių vartotojo sąsajos modeliai	55
2.3. Projektinės dalies išvados	58
3. Realizacija ir testavimas.....	59
3.1. Sistemos projektų architektūra.....	59
3.2. Sistemos loginės architektūros realizacija	60
3.3. Sistemos duomenų bazės realizacija	64

3.4. Sistemos duomenų keitimosi lygmens realizacija	66
3.5. Sistemos pranešimų siuntimo realizacija	69
3.6. Sistemos realizacijos testavimas	72
3.7. Sukurtos sistemos trūkumai, apribojimai bei tolimesnio plėtojimo galimybės	76
4. Dokumentacija naudotojui	77
4.1. Diegimo vadovas.....	77
4.2. Naudotojo vadovas.....	77
4.2.1. Pacientų naudotojų grupė.....	77
4.2.2. Gydytojų naudotojų grupė	81
4.3. Administravimo vadovas	84
5. Rezultatų apibendrinimas ir išvados	87
6. Literatūra	88

LENTELIŲ SĄRAŠAS

1.1 lentelė. Sistemų palyginimo lentelė	24
1.2 lentelė. Programavimo aplinkų palyginimo lentelė	27
1.3 lentelė. Duomenų bazių valdymo sistemų palyginimo lentelė	28
1.4 lentelė. Sistemos programavimo kalbos palyginimo lentelė	30
1.5 lentelė. Internetinių taikymų kūrimo technologijų palyginimo lentelė.....	31
1.6 lentelė. Skirtingų technologijų, web serviso kūrimui palyginimo lentelė	33
2.1 lentelė. Nuotolinių stebėjimų modulio funkcinių reikalavimų lentelė:	38
2.2 lentelė. Pacientų apsilankymų modulio funkcinių reikalavimų lentelė:	39
2.3 lentelė. Pacientų mokymų modulio funkcinių reikalavimų lentelė:	41
2.4 lentelė. Sistemos modulių naudotojo sąsajos reikalavimų lentelė:.....	42
2.5 lentelė. Sistemos modulių plečiamumų reikalavimų lentelė:	43
2.6 lentelė. Sistemos modulių našumo ir greitaveikos reikalavimų lentelė:.....	43
2.7 lentelė. Sistemos modulių prieinamumo ir patikimumo reikalavimų lentelė:	43
2.8 lentelė. Sistemos modulių saugumo reikalavimų lentelė:.....	44
2.9 lentelė. Techninės specifikacijos reikalavimų lentelė:	44

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Kuriamos medicininės IS nuotolinių stebėjimų modulio vizija	14
1.2 pav. Kuriamos medicininės IS apsilankymų modulio vizija	14
1.3 pav. Kuriamos medicininės IS švietimo modulio vizija	15
1.4 pav. Sveikatingumo sistemos „pasveik.lt“ pagrindinis puslapis	17
1.5 pav. Medicininės registracijos sistemos „sergu.lt“ pagrindinis puslapis	19
1.6 pav. Medicininės informacinės sistemos „santa.lt“ pagrindinis puslapis	20
1.7 pav. Medicininės informacinės sistemos „kaunoklinikos.lt“ pagrindinis puslapis	22
2.1 pav. Nuotolinių stebėjimo modulio funkcijų hierarchijos diagrama	37
2.2 pav. Apsilankymų modulio funkcijų hierarchijos diagrama	39
2.3 pav. Mokymų modulio funkcijų hierarchijos diagrama	41
2.4 pav. Kuriamos sistemos architektūrinis modelis	46
2.5 pav. Sistemos nuotolinių stebėjimų modulio UML veiklos diagrama	49
2.6 pav. Sistemos apsilankymų modulio UML veiklos diagrama	50
2.7 pav. Sistemos mokymų modulio UML veiklos diagrama	51
2.8 pav. Sistemos nuotolinių stebėjimų modulio esybių diagrama	52
2.9 pav. Sistemos apsilankymų modulio esybių diagrama	53
2.10 pav. Sistemos mokymų modulio esybių diagrama	54
2.11 pav. Nuotolinių stebėjimų įrenginių navigacijos planas ir vartotojo sąsajos eskizai	55
2.12 pav. Nuotolinių stebėjimų modulio navigacijos planas ir vartotojo sąsajos eskizai	56
2.13 pav. Apsilankymų modulio navigacijos planas ir vartotojo sąsajos eskizai	57
2.14 pav. Mokymų modulio navigacijos planas ir vartotojo sąsajos eskizai	58
3.1 pav. Sistemos sprendimų ir projektų išskaidymas	59
3.2 pav. Valdiklio sluoksnio realizacijos fragmentas	61
3.3 pav. Serviso sluoksnio realizacijos fragmentas	61
3.4 pav. Serviso sluoksnio metodo realizacijos fragmentas	62
3.5 pav. Modelio sluoksnio realizacijos fragmentas	62
3.6 pav. Vaizdo modelio sluoksnio realizacijos fragmentas	63
3.7 pav. Vaizdo sluoksnio realizacijos fragmentas	64
3.8 pav. Vaizdo dalinio sluoksnio realizacijos fragmentas	64
3.9 pav. „Entity Framework Code First“ duomenų bazės priskyrimo modelio schema	65
3.10 pav. Įrenginio dalykinės srities klasės modelio fragmento pavyzdys	65
3.11 pav. „SiteDataAccess.cs“ klasės fragmento pavyzdys	65
3.12 pav. Duomenų bazės įrenginio (angl. „Device“) lentelės struktūra	66
3.13 pav. Sistemos modulių ryšių diagrama	66
3.14 pav. Sistemos modulių užklausų į „HIS“ sistemą klasės sąrašas	67
3.15 pav. Duomenų mainų metodo iškvietaimo pavyzdys	67
3.16 pav. Sistemos užklausos duomenų klasės pavyzdys	68
3.17 pav. „HIS“ sistemos atsako duomenų klasės pavyzdys	68
3.18 pav. Sistemos užklausos klasės į „HIS“ sistemą implementacijos fragmentas	69
3.19 pav. Apsilankymo modulio apsilankymo atšaukimo pranešimo išsiuntimo pavyzdys	70
3.20 pav. Pranešimų siuntimo serviso metodo fragmento pavyzdys	70
3.21 pav. Sisteminių pranešimų siuntimo serviso metodo fragmento pavyzdys	71
3.22 pav. Elektroninių laiškų siuntimo serviso metodo fragmento pavyzdys	72
3.23 pav. Apsilankymų modulio kontrolerio metodo pranešimų siuntimo pavyzdys	72
3.24 pav. Integracijos testavimo principo „Iš apačios į viršų“ schema	73
3.25 pav. „Juodos dėžės“ testavimo schema	73
3.26 pav. Parašytų testų sąrašas	74
3.27 pav. Testinių duomenų pavyzdys 3.28 pav. Testinių komponentų pavyzdys	75
3.29 pav. Testo logikos implementacijos pavyzdys	75
4.1 pav. Sistemos prisijungimo pradžios puslapis	77

4.2 pav. Paciento nuotolinių stebėjimų sąrašo puslapis	78
4.3 pav. Paciento nuotolinio stebėjimo sukūrimo puslapis	78
4.4 pav. Paciento nuotolinio stebėjimo peržiūros puslapis	78
4.5 pav. Paciento nuotolinio stebėjimo rezultatų puslapis	79
4.6 pav. Paciento nuotolinio stebėjimo rezultatų grafinio atvaizdavimo puslapis	79
4.7 pav. Paciento nuotolinio stebėjimo duomenų pateikimo puslapis	80
4.8 pav. Paciento apsilankymų sąrašo puslapis	80
4.9 pav. Paciento naujo apsilankymo sukūrimo puslapis	80
4.10 pav. Paciento naujo apsilankymo patvirtinimo puslapis	81
4.11 pav. Paciento apsilankymo peržiūros puslapis	81
4.12 pav. Gydytojo naujo nuotolinio stebėjimo sukūrimo puslapis	82
4.13 pav. Gydytojo nuotolinio stebėjimo peržiūros puslapis	82
4.14 pav. Gydytojo susipažinimas su paciento pateiktais stebėjimo duomenimis	83
4.15 pav. Gydytojo rekomendacijos sukūrimas pacientui	83
4.16 pav. Gydytojo konsultacijos sukūrimas kitam gydytojui	83
4.17 pav. Įrenginių sąrašo puslapis	84
4.18 pav. Naujo įrenginio sukūrimo puslapis	84
4.19 pav. Įrenginio peržiūros puslapis	85
4.20 pav. Mokymų sąrašo peržiūros puslapis	85
4.21 pav. Mokymo redagavimo puslapis	86

TERMINŲ IR SANTRUMPŲ ŽODYNAS

DBVS – duomenų bazių valdymo sistema. Kompiuterinė programa ar programų paketas, skirtas duomenų bazės valdymui;

JNA – „Java native access“. Įrankis, leidžiantis „Java“ programoms pasiekti kitų programavimo kalbų bibliotekas ir naudoti jų funkcijas;

OSI modelis – abstraktus ryšio protokolą, naudojamą ryšio ir kompiuteriniuose tinkluose, aprašymas; W3C – „World Wide Web Consortium“. Konsorciumas, leidžiantis programinės įrangos standartus, rekomendacijas žiniatinkliui.

Web servisas – pagal „W3C“: tai abstraktus resursas, kuris reprezentuoja galimybes atligli užduotis kas suformuoja koherentų funkcionalumą stebint iš siuntėjo aprašo ir gavėjo aprašo pozicijų;

SOAP – protokolo, skirto apsikeitimui struktūrizuotais duomenimis, specifikacija;

HTTP/S – hyperteksto perdavimo protokolas/saugus hyperteksto perdavimo protokolas. Taikymu protokolas skirtas duomenų apsikeitimui žiniatinklio pagalba;

El. paštas – elektroninis paštas;

XML – „Extensible Markup Language“. Bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.

RSS – „XML“ failų formatas, skirtas internetiniam duomenų rinkimui iš naujientinklių ir tinklaraščių. IS – informacinė sistema;

IDE – „Integrated Development Environment“. Programos, turinčios daug priemonių, palengvinančių ir pagreitinančių programinės įrangos kūrimo procesą.

KK – kauno klinikos;

MVC – „Model-View-Controller“. Programinės įrangos architektūrinis sprendimas, kuris padalija programinę įrangą į atskirus sluoksnius, siekiant atskirti jų funkcionalumus.

SoC – „Separation of concerns“. Dizaino principas, skirtas išskaidyti kompiuterio programą į atskiras dalis, kuri kiekviena iš jų turėtų po savo dizaino principą.

Ecma – „Ecma International“. Tarptautinė organizacija, kuri kuria standartus informacinėms, komunikavimo technologijoms ir vartojimo elektronikai.

„ISO – „International Organization of Standardization“. Tarptautinė standartizacijos organizacija, kuri apima daug kitų standartizacinių organizacijų .

ASP .NET – tinklalapio struktūros technologija, skirta sukurti dinaminę internetinę svetainę.

PHP – dinaminė interpretuojama programavimo kalba, pritaikyta interneto svetainių kūrimui.

JSP – „JavaServer Pages“. Technologija, leidžianti dinamiškai generuoti HTML, XML ar kito tipo puslapius.

ĮVADAS

„Sveikatos priežiūros paslaugų perkėlimas į elektroninę erdvę“ darbas priklauso studijų programos „Informatika“ Kompiuterių tinklų ir Interneto specializacijai.

Siekiant patenkinti pacientų ir darbuotojų poreikius, kuriamos naujos medicininės informacinės sistemos. Šių sistemų pagrindiniai uždaviniai dažniausiai yra įvairių paslaugų (apsilankymų valdymo, informacijos apie ligas, medikamentus, pateikimo, dokumentų išdavimo) suteikimas ir darbo krūvio darbuotojams (įrenginių priežiūros, pacientų stebėjimų, suvestinių ir ataskaitų kūrimo) sumažinimas, siekiant padidinti aptarnaujamą pacientų kiekį, suteikiamą paslaugų kokybę ir darbuotojų efektyvumą. Kuriamą medicininę informacinię sistemą yra skirta Kauno Klinikų pacientams ir jų darbuotojams. Prie jos kūrimo bus prisdėta įgyvendant tris paslaugų modulius: nuotolinių stebėjimų, apsilankymų ir švietimo. Nuotolinių stebėjimų modulis skirtas stebeti savo sveikatos būklę ir gauti gydytojų patarimus. Apsilankymų modulis skirtas planuoti apsilankymams pas gydytojus taupant savo laiką. Švietimo modulis skirtas sužinoti apie profilaktikos programas ir medicinines procedūras, padėti sau skaitant mokymų medžiagą, pateiktą individualiai kiekvienam vartotojui. Taipogi vartotojas gali matyti grafiškai pateiktus sukauptus duomenis, kuriuose atispindi tam tikri dėsningumai (pvz.: kraujospūdžio pakitimo intervalai), leidžiama jam kontroliuoti savo sveikatos duomenis. Kuriamos sistemos moduliai yra informatyvūs ir naudingi, užsakovo pageidavimu yra numatyta šiemis moduliams suteikti kuo daugiau funkcijų, kurios kuo įmanoma labiau padidintų šių modulių suteikiamą paslaugų tikrąją vertę.

Pagrindinis darbo tikslas – sukurti tris sveikatos priežiūros paslaugų modulius (nuotolinių stebėjimų, apsilankymų, švietimo), juos perkeliant į elektroninę erdvę, sukuriant naujas e-paslaugas ir juos integruoti į kuriamą Kauno Klinikų medicininę informacinię sistemą. Taip bus siekiama pagerinti Kauno klinikų teikiamą sveikatos priežiūros paslaugų kokybę, sukurti galimybes taupyti lėšas, laiką bei pinigus, sumažinti administracinię naštą ir padidinti Kauno klinikų paslaugų pasiekiamumą ir prieinamumą. Projekto metu į elektroninę erdvę perkeltos paslaugos sukurs papildomą naudą tiek pacientams, tiek Kauno klinikoms.

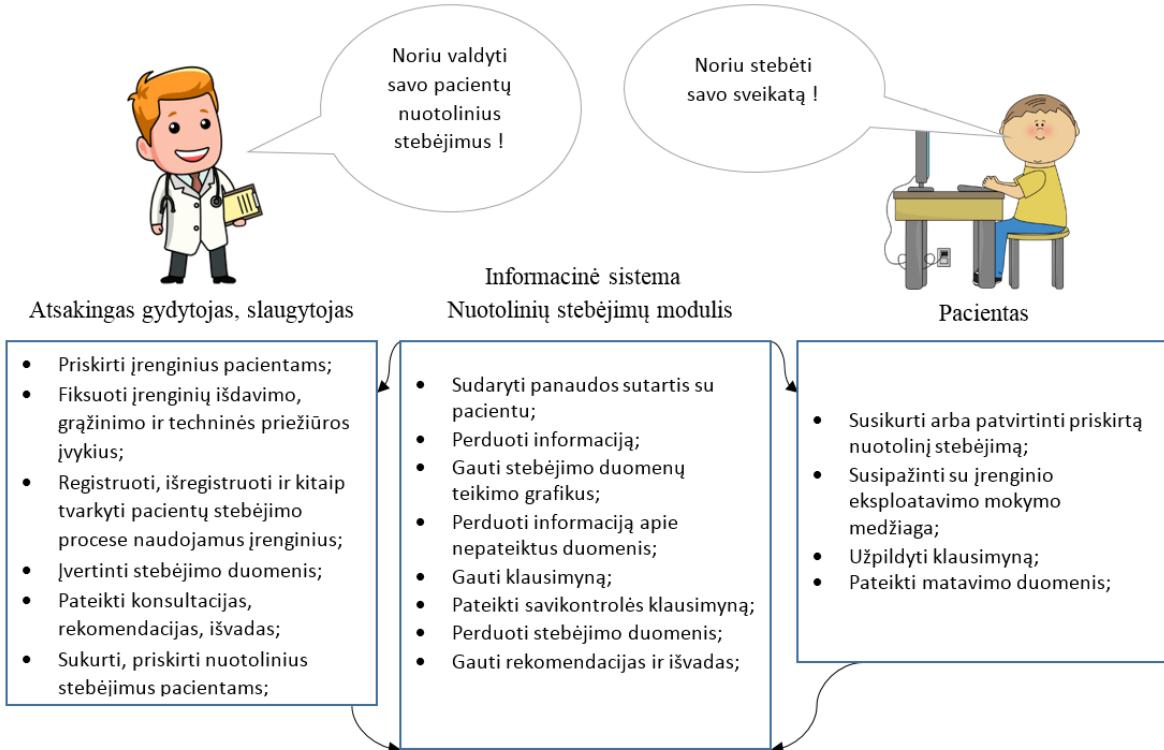
Siekiant tikslo reikalingi įgyvendinti šie *uždaviniai*:

1. Išanalizuoti sukurtas medicinines informacines sistemas ir atliki palyginamają analizę.
2. Išanalizuoti esamas sistemų kūrimo technologijas ir parinkti labiausiai tinkančias.
3. Apibrėžti sistemos specifikaciją ir išskirti funkcinius, bei nefunkcinius reikalavimus.
4. Suprojektuoti sistemą ir jos modulius.
5. Realizuoti medicininės informacinių sistemos modulius (nuotolinių stebėjimų, apsilankymų ir švietimo) ir juos ištenuoti.
6. Paruošti modulių dokumentaciją jų naudotojams.

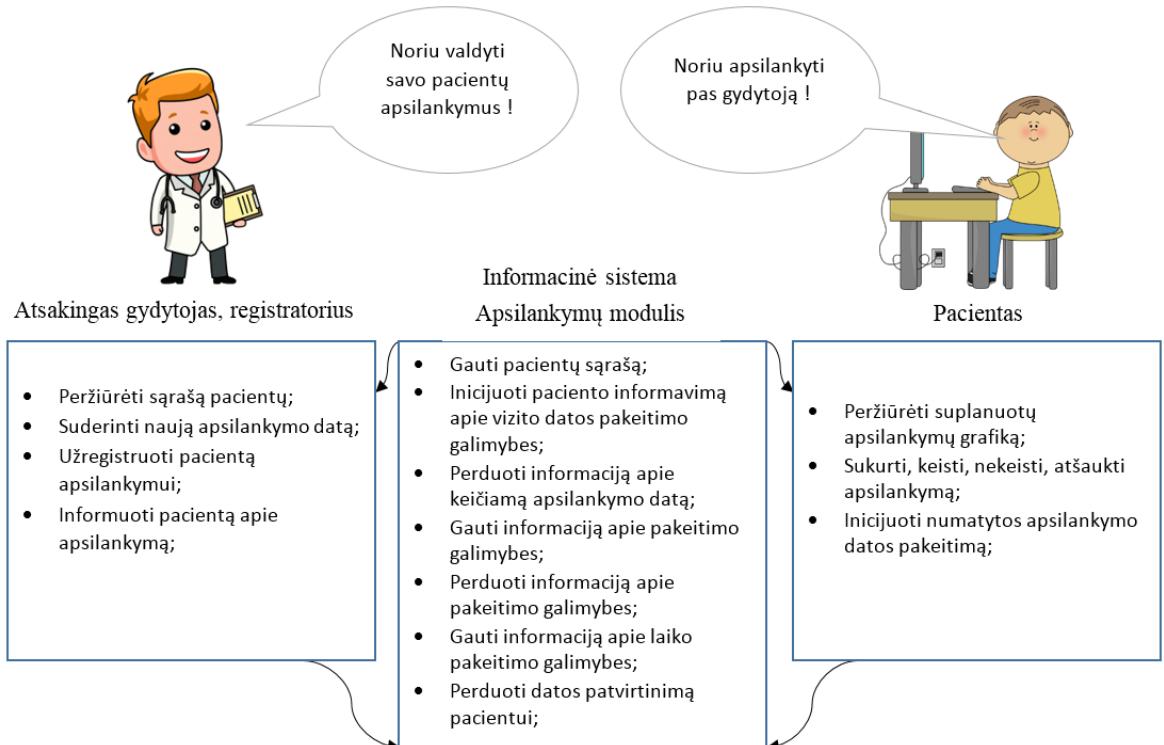
Dokumentą sudaro septynios pagrindinės dalys. Pirmoje jų pristatomas įvadas (pateikiamas dokumento įvadas). Antroje pateikiama įmonės, panašių sistemų, technologijų, programinės įrangos, protokolų analizė. Trečioje aprašomas projektas (pateikiama detalė specifikacija, išskiriant sistemos vartotojų roles, sudarant funkcinius ir nefunkcinius reikalavimus, bei aprašoma projektavimo stadija, pateikiama sistemos komponentų diagrama, duomenų bazės aprašo schema, klasių diagramos). Ketvirtoje aprašoma prototipo realizacija ir testavimas (aprašoma su sukurtų modulių realizacija ir testavimu susijusi informacija, pateikiami realūs vartotojo sąsajų vaizdai, aprašomas testavimo planas ir pateikiami testiniai atvejai). Penktijoje pateikiama naudotojo dokumentacija, papildoma informacija ir nuorodos kaip reikia naudotis sistemos modulių teikiamomis funkcijomis. Šeštojoje pateikiamas išvados (aprašomas darbo metu gautų ir dokumentacijoje užfiksotų rezultatų apibendrinimas ir suformuluojamos išvados). Septintojoje pateikiama literatūra (pateikiamas naudotos literatūros sąrašas).

1. ANALIZĖ

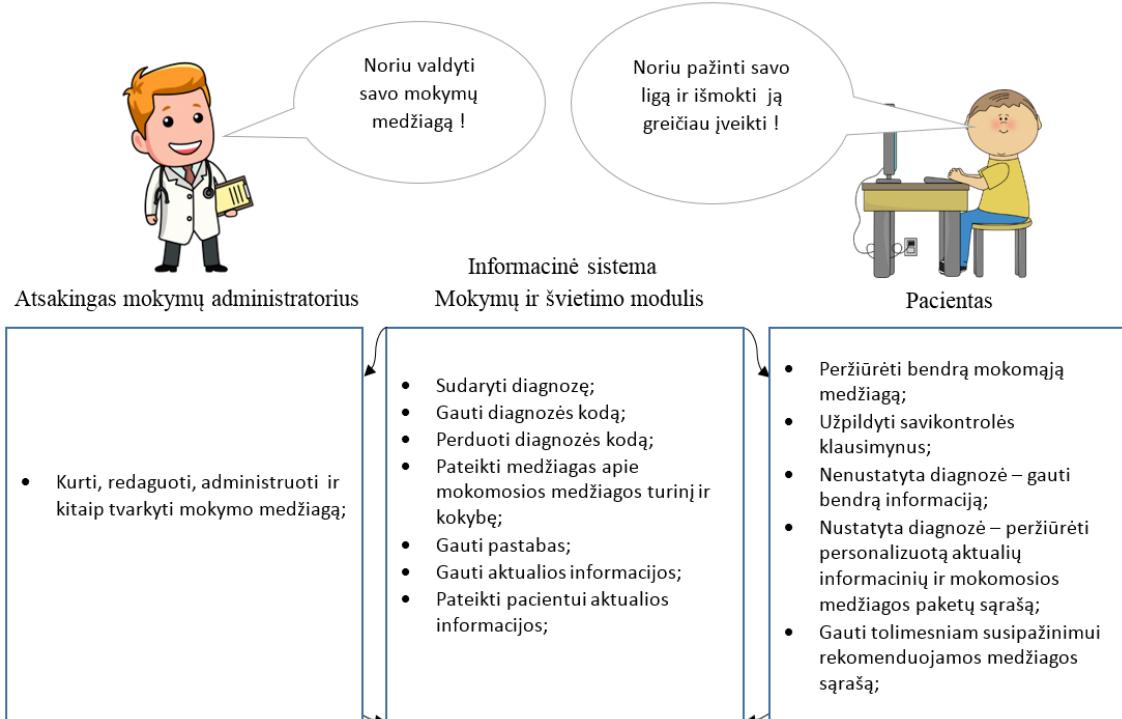
Siekiant perkelti medicininį įstaigų teikiamas paslaugas į elektroninę erdvę kuriamos įvairios tokias paslaugas teikiančios sistemos, kurių pagrindinis tikslas yra sumažinti teikiamų viešujų paslaugų prieinamumą. Nors skirtinges medicininės sistemos siekia panašių tikslų, tačiau viena nuo kitos skiriasi atitinkamų uždavinių akcentavimu ir realizavimo sprendimais. 1.1, 1.2, 1.3 paveiksluose pateikiamą kuriamos sistemos modulių vizija^[1, 2].



1.1 pav. Kuriamos medicininės IS nuotolinių stebėjimų modilio vizija



1.2 pav. Kuriamos medicininės IS apsilankymų modilio vizija



1.3 pav. Kuriamos medicininės IS švietimo modulio vizija

Kuriamos sistemos vizijos paveiksluose atsispindi vartotojų (pacientų, darbuotojų) vaidmuo Kauno klinikų medicininės informacinės sistemos suteikiamų paslaugų modeliuose, kokią naudą sistema jiems galės suteikti ir ką jie su šia sistema galės atliliki.

1.1. Įmonės analizė

Kadangi medicininė informacinė sistema ir jos moduliai bus kuriami realioje darbo aplinkoje, bendradarbiaujant su kitomis įmonėmis ir jų darbuotojais, trumpai išanalizuosime „UAB Amidus“ įmonę, kurios darbo aplinkoje bus kuriami aptarti sistemos moduliai. Prieš apibūdinant įmonę išskeliame šiuos aptarimo kriterijus: veiklos sritis, numatomą ateities vizija, misija, naudojamos technologijos, taikomos metodologijos ir darbo specifika^[3].

1.1.1. Bendra informacija

Įmonės veiklos sritis, specializacija. Įmonė specializuojasi profesionalios programinės įrangos kūrimo srityje.

Įmonės vizija. Įmonė tiki, kad geriausi rezultatai gali būti pasiekti, jei įmonė, jos darbuotojai ir klientai dirba kartu, siekdami to paties tikslų.

Įmonės misija. Siūlyti geriausius sprendimus, bei teikti aukštostos kokybės klientų aptarnavimą. Įmonė siekdamas išgyvendinti savo misiją ir viziją daug dėmesio skiria programinės įrangos kokybei, programinės įrangos palaikymui bei kokybiškam klientų aptarnavimui. Patenkinti įmonės klientai palieka gerus atsiliepimus apie bendradarbiavimą su įmonė. Šie atsiliepimai didina klientų pasitikėjimą ir pritraukia naujus klientus tokiu būdu suteikiant įmonėi galimybę augti ir įveikti vis naujus iššūkius.

1.1.2. Informacija apie įmonėje naudojamas IT

Įmonė daugiausia dėmesio skiria „Microsoft“ platformai. Ji yra „Microsoft“ partnerė, turinti auksinės kompetencijos įvertinimą turinio valdymo srityje, sidabrinės kompetencijos įvertinimus tarpusavio bendradarbiavime, paieškos, verslo, informacinių sistemų kūrimo, programinės įrangos

kūrimo, aplikacijų integracijų srityse. 2011-2013 metais įmonė buvo apdovanota kaip geriausia „Microsoft“ programinės įrangos kūrimo partnerė Lietuvoje.

Dalis technologijų ir įgūdžių naudojamų ir išgyjamų įmonėje:

- „Windows“ aplikacijų kūrimas su „Microsoft .NET Framework 6“.
- „Windows“ tipo aplikacijų kūrimas, testavimas ir išgyvendinimas su „Microsoft Visual C# .NET“ ir „Microsoft Visual Studio .NET“ technologijomis.
- Duomenų bazių projektavimas, sprendimų ieškojimas ir verslo logikos pritaikymas ir palaikymas naudojant „Microsoft SQL Server“.
- Bendravimas tarp įmonės darbuotojų vyksta per elektroninius laiškus arba naudojant susirašinėjimo programę „Skype“, „Lync“.

1.1.3. Įmonės veiklos procesai

Projekto inicijavimas. Marketingo direktorius suranda klientą. Tuomet sistemų analitikas vyksta į pirmajį pokalbį su užsakovu, kuriame išsiaiškinami detalesni sistemių keliami reikalavimai.

Projekto koncepcijos analizė. Įmonės analitikas sudaro programinės įrangos reikalavimų specifikaciją. Nupiešia programinės įrangos vartotojo sąsajos eskizus.

Kainos ir investicijų analizė. Atsižvelgiama į projekto sudėtingumą. Įvertinama kiek laiko užtruks įvykdyti projektą. Taip pat išanalizuojama ar norint įvykdyti projektą įmonei reikės investuoti į naują įrangą susijusią su projekto vystymu. Taip pat projektas palyginamas su jau anksčiau atliktais įmonės ar konkurentų projektais.

Projektų vykdymas, pristatymas ir dokumentacija. Projekto programinės įrangos dokumentacija yra generuojama automatiškai ir sukuriamas „*.doc“ arba „*.pdf“ failas. Taip pat dokumentuojama ir visa medžiaga susijusi su projekto vykdymu, nuo reikalavimų specifikacijos iki aptiktų klaidų ištaisymų ataskaitų. Programinės įrangos architektūra taip pat yra dokumentuojama.

Komandinis darbas. Užduotys tarp komandos narių paskirstomas naudojant „TFS“ (Team Foundation Server) programinį įrankį, taikant „Agile“ metodologijos principus. Kas savaitę vyksta aptarimai apie tai, kaip sekasi atlikti tam tikras užduotis – kiekvienas komandos narys apibendrina ką nuveikė per savaitę. Svarbu yra pasakyti su kokiomis kliūtimis buvo susidurta ir ar problema buvo išspręsta. Jei problema nebuvo išspręsta, tuomet galbūt vienas iš komandos narių išgirdės apie ją galės pagelbėti. Taip pat visada yra galima ir rekomenduotina, jei kyla koks klausimas nedelsiant klausti labiau patyruisių komandos narių.

Inovacijos. Įmonėje dirbama su tokiais projektais kaip Lietuvos Miškų Urėdijos informacinės sistemos projektu, kuriuo siekiama perkelti šios srities paslaugas į elektroninę erdvę, taip pat „Viatender“ projektu skirtu verslui, efektyviam pirkimų valdymui. Stengiamasi sukurti pranašesnes sistemas, nei šiuo metu rinkoje esantys produktai.

Produktų vystymo procesai. Sudaromas pradinis funkcionalumo sąrašas. Jį išgyvendinus produktas yra išleidžiamas į rinką. Kaupiami vartotojų atsiliepimai ir sudaromas naujų funkcijų sąrašas. Atidžiai peržvelgiant šį sąrašą išrenkamas tokis funkcionalumas kuris būtų naudingas platesniams vartotojams ratui. Taip pat kuriami papildiniai kurie skirti vartotojams, kuriems reikia specifinio funkcionalumo.

1.2. Lietuvoje sukurtų panašių sistemų analizė

Kadangi jau yra sukurta ne viena panaši sistema, šiame skyriuje atliksime skirtingų sistemų palyginamąjų analizę. Palyginimui yra parenkami kriterijai, kurie yra aktualūs kuriamos medicininės informacinės sistemos bei jos suteikiamų paslaugų atžvilgiu. Jie padės išsiaiškinti ar sistemą ir ją sudarančius modulius tikrai verta kurti, t.y. ar nėra jau sukurtos visais atžvilgiais pranašesnės sistemos. Siekiant struktūrizuoti analizės rezultatus, jie bus surašomi į lentelę, kurioje bus pateikiamais jau

sukurtos medicininės informacinės sistemos, bei kuriamoji Kauno klinikų informacinė sistema, jų lyginamieji vertinimo kriterijai ir kiekvienos iš šių sistemų vertinimas pagal kriterijus. Prieš atliekant sistemų analizę yra detaliau paaškinami visi pasirinkti panašių medicininių informacinių sistemų palyginimo kriterijai:

- *apsilankymų planavimas* – šis kriterijus apibūdina sistemos gebėjimą valdyti vartotojų apsilankymus pas gydytojus: užsiregistrnuoti, atšaukti, pakeisti datą, laiką, gydytoją, pranešti apie artėjantį, atsilaisvinusį apsilankymą;
- *savo sveikatos būklės stebėjimas* – kriterijus, apibūdinantis sistemos gebėjimą priimti vartotojų pateikiamą informaciją apie jų sveikatos būklę, ją išanalizuoti ir pateikti konkrečius rezultatus: susikurti stebėjimą, gauti rekomendacijas, konsultacijas, išvadas apie esamą sveikatos būklę, peržvelgti gautos duomenis grafiškai, lentelėmis.
- *galimybė sužinoti apie profilaktines programas* – šis kriterijus apibūdina informacinių sistemų teikiamą informaciją apie ligas, vykdomas programas, sveikatinimo akcijas;
- *galimybė padėti sau* – kriterijus, kuris apibūdina ar sistemose yra galimybė skaityti mokymo medžiagą: susipažinti su liga, jos įveikimo etapais ir būdais, sužinoti teikiamų paslaugų procesus ir etapus bei jų teikiamą naudą;
- *paslaugų kokybė* – kriterijus apibūdina bendrą visų suteikiamų paslaugų kokybę: išvaizdingumas (sistemos dizainas, navigaciinių elementų išdėstyMAS), suprantamumas, aiškumas (ar pateikiamą tekstą, grafinius simbolius, elementus yra lengvai suprasti, ar lengva įsiminti jų naudojimą bei naudojimo paskirtį), patogumas (ar norint pasinaudoti pasirinkta paslauga yra lengva ir paprasta, ar yra kitų naudingų pagalbinių funkcionalumų: filtravimas, istorijos peržiūra).

Didžiausia kylanti problema, norint palyginti ir įvertinti panašias, jau sukurtas, ar vis dar plėtojamas sistemas – patekimas į jas. Jų išbandymas sava patirtimi reikalauja išskirtinio prisijungimo, ar registracijos, pateikiant savo asmeninius, valstybinius duomenis, pvz.: prisijungiant per el. valdžios vartus. Patekimas į užsienio šalių informacines sistemas yra dar sudėtingesnis, todėl tai riboja tikrajį sistemų įvertinimą už mūsų valstybės ribų. Atsižvelgdami į tai, kad medicininių informacinių sistemų naudotojai dažniausiai naudojasi savo gyvenamojoje aplinkoje esančių medicininių įstaigų informacinėmis sistemos, o šiuo atveju šia kuriamą sistema labiausiai naudosis žmonės gyvenantys Kauno apskrityje bei likusios Lietuvos teritorijoje, analizei pasirinktos daugiausia savo šalies panašaus tipo informacinės sistemos: „pasveik.lt“, „sergu.lt“, „santa.lt“, „kaunoklinikos.lt“. Toliau pateikiame detalesni šių sistemų aprašymai ir gauti palyginimo rezultatai pagal pasirinktus kriterijus.

1.2.1. Sveikatingumo sistema „pasveik.lt“

Visa „pasveik.lt“ sistemoje pateikiama informacija yra VšĮ „Emedicina“ nuosavybė ir visos teisės į ją priklauso šiai viešajai įstaigai^[4]. Jai taip pat priklauso ir „emedicina.lt“ informaciniis portalas^[5]. Sveikatingumo sistemos „pasveik.lt“ vaizdas pateiktas 1.4 paveiksle.



1.4 pav. Sveikatingumo sistemos „pasveik.lt“ pagrindinis puslapis

Šios sistemos pagrindinis funkcionalumas remiasi statiniu informacijos pateikimu vartotojui:

- rašomi ir skelbiami įvairūs straipsniai, patarimai apie sveikatinimo aktualijas, ligų išvengimo būdus, dietas, grožį;
- pateikiama kontaktinė informacija apie Lietuvos gydymo įstaigas ir vaistines;
- pateikiamas rekomenduojamų perskaityti knygų ir peržiūrėti vaizdo klipų sąrašas;
- sistemos bibliotekoje, žodynose, galima rasti informacijos apie ligas, sindromus ir vaistus;
- skelbiami konkursai;

Sistemos teikiamos dinaminės paslaugos:

- *Galimybė registruotis į poliklinikas.* Registracija į gydymo įstaigas vyksta per kitą informacinę sistemą - „sergu.lt“. Pateikiamas informacinis paveikslėlis ir nuoroda į sistemą;
- *Galimybė atliliki sveikatos testus.* Sveikatos testai turi šešias skirtinges kategorijas: ligų, vaistų, žmogaus anatomijos, mitybos, vaikų sveikatos ir jokiai kategorijai neapibrėžti testai. Kiekviena kategorija turi apie 10-20 testų. Jų atlikimo tvarka visuose testuose yra vienoda – tenka atsakyti į užduodamus klausimus, pasirenkant vieną iš atsakymų. Testo pabaigoje yra pateikiamas teisingai atsakyta klausimų skaičius, priklausomai nuo gauto rezultato yra pateikiama nuoroda į papildomą informacijos suteikimo puslapį;
- *Galimybė paklausti gydytojo.* Pateikiamas apie dešimties gydytojų sąrašas su trumpu kiekvieno iš jų aprašymu. Norint pateikti klausimą, reikia užpildyti siūlomą formą ir iš anksto už tai sumokėti. Suteikiama galimybė peržvelgti kitų vartotojų klausimus ir gydytojų pateiktus atsakymus;
- *Diskusijų/skelbimų forumas.* Suteikiama galimybė bendrauti su kitais sistemos vartotojais, kurti temas aktualiais klausimais;
- *Vaistų kainų palyginimas.* Pateikiamas dešimties vaistų sąrašas, įvairių miestų, miestų rajonų sąrašas, leidžiama įvesti gatvės pavadinimą. Įvykdžius paiešką, gaunamas vaistinių sąrašas, adresas, kaina ir atnaujinimo data;

Sistemoje dažnai matomi netolygūs navigacijos elementų išdėstymai (formos laukai nelygiuojami, tarpusavyje persidengę, nevienodos paraštės, atitraukimai, matomi klaidų pranešimai), daug vietos užima atvaizduojama reklama, suteikiama paieškos ir filtravimo galimybė, tačiau kuri ne visados veikia korektiškai.

Informacinė sistema stengiasi pateikti vartotojams naujausią informaciją, tačiau negali garantuoti, kad visa puslapyje pateikiama informacija yra visiškai teisinga. Sistema neatsako už sistemos vartotojų pateikiamą informaciją. Joje pateiktos informacijos negalima naudoti internete, spaudoje, televizijoje ar radijuje. Šioje sistemoje pateikiama informacija negali atstoti realios gydytojo konsultacijos.

Apibendrinant „pasveik.lt“ yra bazinės, statinės informacijos pobūdžio suteikimo šaltinis, kuris labiausiai siekia savo publikuojamais straipsniais pritraukti kuo daugiau vartotojų, pasižymintis mažą reikšmę turinčiais funkcionalumais (kurių dalis iš jų yra netgi mokami), bei reklamomis užpildytu ir klaidų turinčiu dizainu.

1.2.2. Medicininė registracijos sistema „sergu.lt“

Tai medicininė informacinė sistema, skirta išankstinei pacientų registracijai, kurios tikslas yra apjungti atskirų gydymo įstaigų registracijos sistemas^[6]. Ji yra įgyvendinta nacionaliniu mastu, o tai leidžia suteikti pacientui galimybę rinktis specialistų iš skirtinės gydymo įstaigų paslaugas. Medicininės registracijos sistemos „sergu.lt“ vaizdas pateiktas 1.5 paveiksle.



1.5 pav. Medicininės registracijos sistemos „sergu.lt“ pagrindinis puslapis

Ši sistema leidžia jos vartotojams:

- Rezervuoti gydytojo priėmimo laiką internetu;
- Rinktis pacientams gydytojus iš skirtinų gydymo įstaigų;
- Automatiškai (SMS arba el. paštu) informuoti pacientą apie būsimo apsilankymo būklę, apsilankymo atšaukimą, siųsti priminimus;
- Internete stebeti visų savo planuojamų bei įvykusių apsilankymų būklę;
- Gydytojui iš anksto leidžiama susipažinti su atvyksiančių pacientų duomenimis;
- Siunčiamam gydytojui suformuoti siuntimą, kurį galima atspausdinti bei automatiškai elektroniniu būdu pridėti prie planuoamo paciento apsilankymo duomenų, stebeti savo siųstų pacientų apsilankymų būklę;
- Įstaigos administratoriams matyti gydytojų planuojamą krūvį, pacientų pasiskirstymą, automatiškai gauti statistines ataskaitas ir kitą informaciją, analizuoti pacientų registracijos procesą (kokiu laiku, koks kiekis, kokių pacientų registruojasi ir pan.) ir tuo pagrindu daryti įstaigos valdymo sprendimus;

Sistemoje galima rasti apie 70 medicininių įstaigų, kurios aptarnauja 49 Lietuvos savivaldybių, miestų ir rajonų gyventojus [7]. Siūlomas platus paieškos funkcionalumas: paieška pagal specialybę, gydytoją, šeimos gydytoją. Naudingos sąrašų rikiavimo galimybės: pagal kabinetą, specialybę, gydytoją, laisvų vietų skaičių, atvaizduojamų rezultatų kiekį. Pateikiamas interaktyvus Lietuvos valstybės žemėlapis, kuris yra sudarytas pagal administracinių suskirstymą – iš apskričių ir savivaldybių. Pasirinkus norimą teritoriją, yra atvaizduojamos jose esamos įstaigos. Teritorijos, kuriose yra jai priklausančių įstaigų, yra atvaizduojamos žalsvai, priešingu atveju – bespalvės. Kiekviena įstaiga turi po savo interaktyvų vienos žemėlapį.

Vartotojų identifikacijai yra naudojamos SMS trumposios žinutės – siunčiami identifikacinių slaptažodžių. Ši paslauga yra mokama. Taikomas informacijos konfidencialumas – nejsimenami pateikiami telefonų numeriai.

Apibendrinant „sergu.lt“ yra apsilankymų planavimo sistema, kuria naudojasi didžioji dalis Lietuvos medicininių įstaigų ir gyventojų. Susikoncentravusi į konkrečią paslaugą ji sugeba savo vartotojams suteikti aukščiausios kokybės aptarnavimą, kuris yra vis puoselėjimas, o pati sistema yra nuolat palaikoma ir plečiama. Ieškant būdo užsiregistrnuoti pas gydytoją „sergu.lt“ yra puikus pasirinkimas.

1.2.3. Medicininė informacinė sistema „santa.lt“

Tai medicininė informacinė sistema, skirta Vilniaus universiteto ligoninės Santariškių klinikos vartotojams, kurios tikslas yra suteikti aktualią informaciją apie pačią įstaigą, jos atliekamą veiklą ir teikiamas paslaugas [8]. Tai progresyvi, daugiaprofilinė gydymo įstaiga, savo pacientams teikianti

aukščiausio lygio medicinos paslaugas, atitinkančias naujausius mokslo pasiekimus. Medicininės informacinės sistemos „santa.lt“ vaizdas pateiktas 1.6 paveiksle.



1.6 pav. Medicininės informacinės sistemos „santa.lt“ pagrindinis puslapis

Santariskių klinikų informacinės sistemos struktūra yra išskaidyta į tris aiškius modulius:

- *Statinės informacijos modulį*, kuris padeda sistemos vartotojams susipažinti su sistemos teikiamomis galimybėmis ir šiuo metu įstaigos vykdoma veikla, jos istorija;
- *Pacientų modulį*, skirtą supažindinti galimus ir esamus pacientus su įstaigos struktūra, suteikti informacijos apie atvykimo būdus, suteikiama galimybė užpildyti sveikatos priežiūros paslaugų vertinimo anketa;
- *Elektroninių paslaugų modulį*, kuriame yra pateikiamos klinikų siūlomos medicininės paslaugos;

Plačiau apie kiekvieną iš jų:

Informacijos modulis. Šis modulis yra suskirstytas į 14 skirtingų skyrelių:

- *Apie mus.* Pateikiama bendra informacija apie įstaigos dalininkus, jos veiklą, joje dirbančius darbuotojus, kitus filialus su savo atskiromis informaciniemis sistemomis (vaikuligonine.lt, vpc.lt, iltl.lt), įvertinimus, pasiekimus, prioritetines mokslinės veiklos kryptis, aptarnaujamų pacientų kiekį ir daugelį kitos informacijos;
- *Istorija.* Glaustai pateikiama informacija apie įstaigos istoriją: veiklos pradžią, reorganizacijas, jos vadovus;
- *Struktūra ir kontaktai.* Pateikiama Santariskių klinikų struktūra, susisiekimo duomenys su administracijos darbuotojais, įvairiais buhalterijos ir ekonomikos skyriais bei tarnybomis;
- *Karjera.* Aptariamos karjeros galimybės ir šiuo metu įstaigoje vykdomi konkursai užimti darbuotojo pareigoms;
- *Veikla.* Pateikiama informacija apie vykdomus projektus, sudarytas ataskaitas, darbo užmokestį, viešuosius pirkimus organizacijas, turimas licenzijas;
- *Korupcijos prevencija.* Pateikiamas korupcijos prevencijos programos priemonių vykdymo planas;
- *Rekvizitai.* Pateikiami įstaigos rekvizitai, sąskaitos bankuose;
- *Naujienos/Renginiai.* Skyreliai, skirti talpinti kuriamus straipsnius. Informacija yra atnaujinama bent kartą per savaitę. Galima peržvelgti straipsnių archyvus;
- *Padėkos.* Unikalus skyrelis, skirtas talpinti vartotojų išreikštoms padėkoms įstaigos darbuotojams;

- *Parašykite mums.* Pateikiama duomenų pildymo forma: pildytojo vardo, el. pašto adreso ir klausimų, padėkų, pastebėjimų laukeliai. Formą galima išsiųsti arba įvedus nenorimus duomenis ją išvalyti;
- *Paieška svetainėje.* Suteikiama galimybė ieškoti informacijos po visą informacinę sistemą pagal raktažodį (visų žodžių, bet kurio žodžio, tikslios frazės), pagal tipą (irašus, nuorodas, kontaktus, kategorijas, skyrius, RSS naujienas), ją rūšiuoti (pirmiausia naujausi, pirmiausia seniausi, populariausi, abécéline tvarka, skyrius/kategorijas);
- *Galerija.* Pateikiama 70 nuotraukų galerija, atvaizduojanti įstaigoje atliekamas procedūras, joje dirbančius darbuotojus, turimas technologijas ir įtaisus. Ji yra dažnai atnaujinama.
- *Video reportažai.* Mažas skyrelis, vos su keletą informacinių, reklaminių vaizdo reportažų apie Santaros slėnį, išmaniąją ligoninę ir įstaigos skulptorių darbą;

Pacientų modulis. Suskirstytas į 4 skirtingus skyrelius:

- *Kaip mus rasti.* Pateikiama kontaktinė ir navigacinė informacija apie įstaigos skyrių adresus, atvykimo būdus (iš stoties, iš Vilniaus Oro uosto), telefonus, nuorodas į kitas klinikų filialų informacines sistemas;
- *Centrai ir skyriai.* Skyrelis apibrėžiantis atskirų įstaigos centrų ir skyrių veiklą, kurie yra išvardinti ir atvaizduojami kortelėmis. Iš viso pateikiama 32 centrų/skyrių informacija apie jų vadovus, istoriją, dirbančius medikus, ligas, pasiekimus, mokslinę ir pedagoginę veiklą, kontaktus;
- *Aktuali informacija.* Skyrelyje yra aprašoma įstaigoje priimamų ir aptarnaujamų pacientų guldymo į ligoninę, konsultavimo tvarka, mokoma kaip reikia užsiregistrnuoti apsilankymams, pateikiamos vidaus tvarkos taisyklės, atliekami tyrimai, nuorodos, informacija apie koplytėlę, anesteziją;
- *Paciente apklausos anketa.* Informacinės sistemos vartotojams pateikiama įstaigos sveikatos priežiūros paslaugų vertinimo anketa. Anketą yra prašoma užpildyti baigus gydymą ligoninėje. Pagal ją yra vertinama suteikiamų paslaugų kokybė. Ji yra anoniminė, o pagal gautus duomenis, jie yra lyginami su kitomis Lietuvos ligoninių duomenimis, siekiant padėti pacientams apsispręsti renkantis ligoninę;

Elektroninių paslaugų modulis. Suskirstytas į 4 skirtingus skyrelius:

- *Išankstinė registracija.* Pateikiama nuoroda į Santariškių klinikos gydymo įstaigos puslapį „sergu.lt“ apsilankymų valdymo sistemoje;
- *Internetinė paciento kortelė.* Pateikiama nuoroda į medicininę informacinię sistemą „viva.santa.lt“, kurioje galima matyti savo gydymo įstaigoje atliktus tyrimus, nustatytą diagnozę ir kitus klinikinius dokumentus^[9]. Tai vienintelė pačių Santariškių klinikų teikiamų elektroninių paslaugų;
- *ESP (Medikams).* Pateikiama nuoroda į medicinos personalo apskaitos sistemą „esps.medpas.lt“;
- *Elektroninis paštas.* Pateikiama nuoroda į įstaigos turimą „Microsoft Outlook Web App“ aplikaciją, kuria naudojantis galima rašyti ir siųsti laiškus norimiems įstaigos darbuotojams, skyriams ar filialams;

Apibendrinant „santa.lt“ yra visą reikalingą informaciją apie Vilniaus universiteto ligoninės Santariškių klinikas turinti informacinė sistema, suteikianti vartotojams ne tik ypač reikalingos kontaktinės, procedūrinės informacijos, kurios ypatingai reikia norint pasinaudoti įstaigos teikiamomis paslaugomis, tačiau ir antraruše informacija, kuria yra siekiama įtraukti vartotoją į įstaigos veiklą, padėti jam pažinti jos aplinką ir joje dirbančius darbuotojus. Pati sistema nesiūlo daug įstaigos teikiamų paslaugų elektroninėje erdvėje, naudojasi jau sukurtais sprendimais, kurie yra prieinami daugeliui Lietuvos medicininių įstaigų. Sistemoje nėra siūloma galimybė stebeti savo paties sveikatą, padėti sau skaitant mokomąjį medžiagą. Pateikiama informacija ir navigacija po sistemą yra patogi ir

jsimenant bei nuolat atnaujinama. Suteikiamą galimybę sistemą peržiūrėti neįgaliesiems. Apsilankyt i sistemoje yra išties rekomenduotina ir naudinga prieš apsilankant pačioje gydymo įstaigoje.

1.2.4. Medicininė informacinė sistema „kaunoklinikos.lt“

Tai medicininė informacinė sistema, skirta Lietuvos sveikatos mokslų universiteto ligoninės Kauno klinikų vartotojams, kurios tikslas yra suteikti aktualią informaciją apie pačią įstaigą, jos atliekamą veiklą ir teikiamas paslaugas [10]. Klinikos stengiasi, kad kiekvienam į ligoninę besikreipiančiam žmogui būtų suteikta aukščiausios kokybės medicinos pagalba. Medicininės informacinės sistemos „kaunoklinikos.lt“ vaizdas pateiktas 1.7 paveiksle.



1.7 pav. Medicininės informacinės sistemos „kaunoklinikos.lt“ pagrindinis puslapis.

Informacinėje sistemoje visa pateikiama informacija yra suskirstyta į daugybę mažų skyrelių, kurie nėra korektiškai sugrupuoti ir atvaizduoti. Pagrindinė navigacija po sistemą vyksta per abiejuose sistemos vartotojo sąsajos kraštose esančias meniu juostas.

Aprašysime kairinėje meniu juostoje pateikiamus skyrelius ir jų paskirtį:

- *Apie mus.* Pateikiami įstaigos tikslai ir ateities vizijos. Galima patekti į atskirą kitos informacijos skyrelį, tačiau tame deja nėra pateikiamos jokios informacijos;
- *Struktūra ir kontaktai.* Pateikiama Kauno klinikų struktūra, susisiekimo duomenys su administracijos darbuotojais, įvairiais įvairių skyrių kontaktiniai duomenys (telefonai, el. pašto adresai);
- *Veikla.* Analizės metu patekti į šį skyrelį nebuvo galima;
- *Paslaugos.* Pateikiamos įstaigoje teikiamų mokomų ir nemokamų paslaugų sąrašai, jų kainos, atsiskaitymo būdai. Pateikiama forma, kurioje galima užpildyti savo asmens ir kontaktinius duomenis, tekstinių formatu aprašyti savo diagnozę, trumpą ligos istoriją, atvykimo tikslą, pageidautiną atvykimo laikotarpi;
- *Pacientams ir lankytojams.* Pateikiamas gausus įvairios informacijos: skubioji pagalba, konsultacijos poliklinikoje, kaip pateikti pas gydytoją, išankstinė registraciją per „sergu.lt“ registraciję sistemą, gydytojų darbo grafikai, ką reikėtų pasiimti atvykstant, klinikų vidaus tvarkos taisyklės, kaip atvykti ir kur pastatyti automobilį;
- *Žemėlapiai.* Pateikiami du navigaciniai žemėlapiai, vaizduojantys Kauno klinikų ir jos filialų vietas ir atvykimo schemas;
- *Laikraštis „Ave Vita“.* Pateikiamas įstaigoje išleidžiamų medicininių žurnalų archyvas su galimybe kiekvieną iš jų nemokamai atsisistoti;
- *Apklausos.* Pateikiamas vykdomų apklausų sąrašas, kurios yra atliekamos „apklausa.lt“ sistemoje.

- *Nuorodos.* Pateikiamas įvairių nuorodų sąrašas į įvairias kitas valstybines įstaigas ir mokslines institucijas.
- *Informacija darbuotojams.* Analizės metu patekti į ši skyrelį nebuvvo galima;
- *Administratoriams.* Analizės metu patekti į ši skyrelį nebuvvo galima;
- *El. paštas (darbuotojams).* Patenkama į „Novell GroupWise WebAccess“ serviso sistemą, kuri valdo tam tikros vartotojų grupės bendravimą el. paštą;
- *Informacijos rodyklė.* Pateikiamas informacinės sistemos puslapių nuorodų medis;
- *Laisvos darbo vietas.* Aptariamos karjeros galimybės ir pateikiamos laisvos pozicijos užimti darbuotojo pareigoms;
- *Korupcijos prevencija.* Aprašoma įstaigos korupcijos prevencijos programa, pateikiami informaciniai vaizdo įrašai, rekomendacijos kaip reikėtų išvengti šios problemos;

Aprašysime dešininėje meniu juostoje pateikiamus skyrelius ir jų paskirtį:

- *Poliklinika.* Pateikiama informacija apie Kauno klinikų centrinę konsultacinię polikliniką: registratūrų sąrašas (kontaktinė informacija ir nuorodos į jų atskirus puslapius), adresą, istoriją, darbo grafikus;
- *Skubios pagalbos skyrius.* Pateikiama informacija apie Kauno klinikų skubios pagalbos skyrių: darbo laiką, pagrindinę veiklą, ką turi žinoti žmogus, kuris kreipiasi pagalbos, skyriaus gydytojų sąrašą, skyriaus istoriją, naudingus interneto adresu;
- *Profilinės klinikos.* Pateikiama informacija apie visas Kauno klinikų profilines klinikas. Kiekviena iš jų pateikia kontaktinę informaciją (adresą, telefono numerį, el. paštą), nuorodas į vietas atvaizdavimo žemėlapį pačioje sistemoje ir „maps.lt“ svetainėje;
- *Filialai.* Pateikiama informacija apie visus Kauno klinikų filialus. Kiekviena iš jų pateikia kontaktinę informaciją (adresą, telefono numerį, el. paštą), nuorodas į vietas atvaizdavimo žemėlapį pačioje sistemoje ir „maps.lt“ svetainėje;

Apibendrinant ši „kaunoklinikos.lt“ yra visą reikalingą informaciją apie Lietuvos sveikatos mokslų universiteto ligoninės Kauno klinikas turinti informacinė sistema, suteikianti vartotojams tik reikalingiausią kontaktinę ir susipažinimo su įstaiga informaciją. Sistema nesiūlo jokių sveikatos priežiūros paslaugų elektroninėje erdvėje, naudojasi tik bendro prieinamumo, jau sukurtomis sistemomis. Analizuojant ir naršant po sistemą nebuvvo nepastebėtas ne mažas dėmesį blaškančių, judančių reklamų, paveiksliukų ir animacijų kiekis, sistemoje naudojamų navigacinių elementų persidengimas, mažas skaitomo teksto šriftų dydis ir paraščių neatitikimai. Sistemos skyreliuose dažnai buvo pateikama pasikartojanti informacija, skyriai nebuvvo aiškiai išdėstyti ir įvardinti, dėl to reikiama informaciją buvo rasti kur kas sunkiau. Nebuvvo aptiktas joks skelbiamu naujienų skyrelis. Tai sistema, paremta vien statinės informacijos paslaugos teikimu vartotojams, turinti labai mažai dinamiškumo ir nepasižyminti didelėmis vartotojams teikiamais galimybėmis.

1.2.5. Kuriamoji Kauno klinikų medicininė informacinė sistema ir jos palyginimas su analizuojamomis sistemomis

Kuriama naujoji Kauno klinikų medicininė informacinė sistema ir joje įgyvendinami medicininių paslaugų moduliai nuo anksčiau paminėtų visų pirma išsiskiria tuo, kad nei viena iš apžvelgtų sistemų nesuteikia galimybės vartotojams stebeti savo asmeninės sveikatos būklės, stebėsenos tikslais pildyti duomenų pateikimo formas ir gauti atliekamų analizių rezultatus, pasiūlyti vartotojui reikiamas tekstinės ir vaizdinės informacijos, kuri būtų skirta tik jam ir atrenkama pagal sergamos ligos gydimo etapus, stadijas, naudojamus vaistus ar stebėjimo įrenginius. Priešingai nei „pasveik.lt“, „santa.lt“ ir „kaunoklinikos.lt“ sistemos, kurių funkcionalumas remiasi vien tik informacinių paslaugų suteikimu, kuriamieji naujosios sistemos moduliai labiau atspindės šiuo metu Lietuvos medicininėse įstaigose suteikiamas paslaugas, kurios bus perkeltos į elektroninę erdvę

kuriamos naujosios sistemos pagalba. Kitas svarbus kriterijus, kuo kuriamą sistemą ir jos paslaugų moduliai yra inovatyvūs ir skiriasi nuo kitų aprašytų yra duomenų atvaizdavimas. Kuriamos sistemos moduliai kaups įvestus duomenis ir pagal pageidavimą juos pateiks grafiskai, diagramų pavidalu. Bus galima pasirinkti kelis duomenų atvaizdavimo būdus: stulpelinį grafiką, linijinių grafikų arba lentelę. Paslaugų moduliai bus kuriami taip, kad esant poreikiui, jie galėtų būti panaudojami ir integruojami ir kituose sistemos moduliuose. Jais bus galima kokybiškai dirbti ir naudotis su įvairaus dydžio ekranus turinčiais įrenginiais. Sistemą planuoja kurti kuo įmanomą lankstesnę, lengvai praplečiamą ir suderinamą su kita programine įranga. 1.1 lentelėje pateikiama visų išanalizuotų sistemų ir kuriamos naujosios informacinių sistemos palyginimo kriterijų įvertinimai.

1.1 lentelė. Sistemų palyginimo lentelė

Lyginimo kriterijai	pasveik.lt	sergu.lt	santa.lt	kaunoklinikos.lt	Kuriama sistema
Apsilankymų planavimas	+/-	+	+/-	+/-	+
Savo sveikatos būklės stebėjimas	-	-	-	-	+
Galimybė sužinoti apie profilaktines programas	-	-	+/-	-	+
Galimybė padėti sau	+/-	-	+/-	+/-	+
Paslaugų kokybė	+/-	+	+	+/-	+

Apibendrinime gautos analizės rezultatus pagal išsikeltus palyginimo kriterijus:

Vartotojų registracija apsilankymams. Sistemų teikiamos bazinės funkcijos, tokios kaip užsiregistravimas vizitui pas gydytoją, neturi atitinkamai gilaus funkcionalumo. Užsiregistravimas yra galimas, tačiau dauguma jų nesugeba informuoti pacientų ir gydytojų apie atsilaisvinusius ankstesnius vizitų laikus, pakeisti savo sukurto vizito datą ir laiką, ji atšaukti, gauti rekomendacijas apie profilaktinius apsilankymus (pagal ligos pobūdį arba rizikos grupę), yra naudojamas kitų sistemų suteikiamomis galimybėmis.

Sveikatos būklės stebėjimas. Susipažinti su savo sveikatos būkle ir apsilankymų medicininėse įstaigose istorija – sistemoje retai, kai kuriose išvis neteikiama paslauga. Pacientų noras patiemams stebeti savo sveikatos būklę tokiose sistemoje išvis nėra įgyvendintas. Jie neturi galimybės pateikti savo matavimų duomenų né vienoje iš analizuotų informacinių sistemų. Dėl to gydytojai neturi galimybės informuoti pacientus konkrečioje informacinė sistemoje, pranešti apie jų pateiktus stebėjimo rezultatus, rekomenduoti apsilankymą pas gydytoją, jeigu paciento pateikti matavimo rezultatai viršija normą. Dėl šių priežasčių pacientai neturi galimybės peržiūrėti duomenų stebėjimo rezultatų ir grafikų, atitinkamai pagal pasirinktus parametrus, ar apžvelgti kurias normos ribas jie viršijo ir ką dėl to reikėtų daryti – nepateikiamas jokios rekomendacijos. Medicininėse įstaigose naudojami įrenginiai pacientų būklei stebeti ir jų kaupiama informacija nėra talpinama į jokią informacinę sistemą – nėra jokio automatiško susiejimo tarp naudojamo įrenginio ir informacinių sistemų.

Galimybė sužinoti apie profilaktines programas. Bendras daugelio Lietuvoje sukurtų susijusių sistemų požymis yra tokis, kad dauguma jų pateikia tik statinę, bendrai auditorijai taikomą informaciją vartotojui aktualiaisiais klausimais: konkrečių ligų apibūdinimus, jų gydymo etapus, išvengimo būdus. Tokios sistemos nėra skirtos individualiam vartotojui, o tik bendrai sistemos vartotojus siejančiai ligų grupei. Jos neturi galimybės individualiai įvertinti kiekvieno sistemos vartotojo sveikatos būklę, ją išanalizuoti, ir pagal tai pateikti rekomendacijas, gautus rezultatus, taip padedant suteikti kuo tikslesnę ir naudingesnę paslaugos kokybę individualiai kiekvienam vartotojui. Didelius sunkumus kelia ir tai, kad dauguma jų nėra patogios: senamadiškas dizainas, mažos arba išvis nepateikiamas jokios duomenų filtravimo galimybės, jokio atgalinio atsako vartotojui apie sistemoje atliekamus veiksmus ir jų būklę.

Galimybė padėti sau. Vartotojų mokymai juos dominančiais sveikatos klausimais sistemoje arba nėra įgyvendintas arba yra siūlomas labai paviršutinišku lygiu, tokiu kaip norint sužinoti daugiau

informacijos, reikia užsiregistrnuoti pokalbiui ar apžiūrai pas gydytoją, ar nukeliauti į kitą informacinę sistemą ir perskaityti kokį straipsnį. Figūruoja nevienalytiškumo principas – visa informacija yra pateikta ne nuosekliai, skirtingose vietose.

Atlikus panašių sistemų analizę iš gautų rezultatų galima padaryti išvadą, kad kuriamos sistemos moduliai geriausiai atitinka pageidavimus, ji pagal pasirinktus kriterijus yra geresnė, nei kitos. Pateiksime apibendrintas, galutines visų išanalizuotų sistemų įvertinimo išvados:

- Sistemos nėra orientuotos į individualų vartotoją;
- Sistemose pateikiama informacija ir bendras kontekstas nėra dinamiškas;
- Pagrindinės teikiamos paslaugos yra statinio informacinių pobūdžio;
- Nėra gilaus bendradarbiavimo ir stebėjimo galimybių tarp paciento ir gydytojo;
- Ne visose sistemose pacientams yra suteikiamos galimybės išmokti pažinti savo ligą ir ją atitinkamai įveikti;

Numatomą kuriamos sistemos ir jos modulių viziją ir analizės metu išsiaiškintus kitų sistemų trūkumus mūsų įgyvendinama informacinė sistema stengsis pašalinti ir įgyvendinti naujus, šiuolaikiškus tų problemų sprendimo būdus. Išsiaiškinti paslaugų privalumai, kurie buvo aptarti, bus taipogi įgyvendinami, kurių daugelis iš jų suteiks netgi didesnį funkcionalumą ir naudą sistemos vartotojams. Sistema bus kuriamā kaip naujas produktas, pradedant nuo nulio, o tai leis nevaržomai naudotis naujausiomis technologijomis pagrįstus programinius įrangius, kurie šiandien suteikia daug didesnes galimybes, priešingai nei kitos išanalizuotos sistemos, kurios buvo pradėtos kurti kur kas ankščiau.

Kodėl įgyvendinimui pasirinkti būtent šie konkretūs moduliai ?

- *Nuotolinių stebėjimų* – unikalus ir reikalingas modulis, suteikiantis daug funkcionalumo vartotojams. Tokio funkcionalumo neturi nė viena išanalizuota sistema;
- *Apsilankymų modulis* – įgyvendinamas dėl bendros modulių integracijos naujojoje sistemoje, bendro jų vientisumo ir susietumo. Sistema suteiks galimybę valdyti savo apsilankymus ir per „sergu.lt“ registracinių sistemų, tačiau tuomet nebus siūlomi kiti atskirai šiame modulyje teikiami funkcionalumai;
- *Švietimo modulis* – remsis ne vien tik statinės informacijos pateikimu, tačiau labiau orientuosis į individualų vartotoją, jo turimas diagnozes ir atitinkamai pagal šiuos parametrus bus teikama jam skirta, atrinkta informacija;

1.3. Galimų sprendžiamo uždavinio įgyvendinimo priemonių variantų analizė

Apžvelgus keletą medicininių informacinių sistemų ir atlikus palyginamąjį analizę matyti, kad nors sistemos ir turi panašumų, suteikiamų paslaugų rūšys, kiekis ir kokybė skiriasi. Vienos sistemos orientuoja į vienas suteikiamų paslaugų rūšis, kitos į kitas. Šiame skyriuje apžvelgsime esamas kūrimo technologijas ir atliksime programinių įrankių, technologijų bei protokolų palyginamąjį analizę.

Numatoma, kad sistemos kūrimo įrankių ir technologijų pasirinkimui didelę įtaką turės šie veiksnių:

- Įmonėje taikomų technologijų gausa;
- Žmonių, galinčių dirbti su pasirinktomis technologijomis kiekis;
- Darbo su konkrečiais įrankiais ir technologijomis patirtis;
- Programinių įrankių ekosistemiškumas, integraciškumas ir tarpusavio palaikymas su kitaip pasirinktais įrankiais;
- Atsižvelgimas į kuriamos sistemos ir jos modulių funkcius, nefunkcinius ir specifikacionius, ypatingai techninius reikalavimus;

Kadangi medicininės informacinės sistemos susideda iš skirtingų programinių komponentų (duomenų bazės, valdymo komponentų, tinklo servisų, ir kt.), kiekvieno komponento realizacijai reikia rinktis keletą iš egzistuojančių kūrimo įrankių, technologijų, reikalingų ir naudojamų protokolų. Konkretaus komponento (pvz. programavimo aplinkos) kūrimui renkamasi ta technologija, kuri geriausiai suderinama su visa sistema ir yra pranašesnė už kitas, lyginant pagal tam tikrus pasirinktus kriterijus.

I programinių įrankių palyginamają analizę įeina šie komponentai: programavimo aplinka ir duomenų bazių valdymo sistema. Aptariama pasirinkta sistemos gyvavimo ciklo valdymo sistema, privataus tinklo prieigos sistema, naudojamos internetinės naršykės, serveris.

I technologijų palyginamają analizę įeina šie komponentai: sistemos programavimo kalbos, internetinių taikymų kūrimo technologijos. Aptariamos pasirinktos žymėjimo, skriptų, stilių derinimo, užklausų kalbos, objektinis-reliaciniis konvertuotojas.

I protokolų palyginamają analizę įeina servisių kūrimo technologijų komponentas.

Kiekvienas iš išvardytų komponentų aprašomas detaliau ir yra pateikiami jų analizės rezultatai.

1.3.1. Programinių įrankių analizė

1.3.1.1. Programavimo aplinkos („Visual Studio 2013“)

Jokia kuriama didesnė informacinė sistema neapsieina be geros sistemos architektūros, kuriai sukurti ir valdyti yra naudojama programavimo aplinka (IDE). Sistemos kūrimui galima pasirinkti vieną iš kelių itin daug funkcionalumų turinčių programavimo aplinkų^[11]. Pasirinkimas didelis, todėl renkantis reikia atkreipti dėmesį į apsibrėžtus kriterijus, pagal kuriuos ir įvertinama, kurią programavimo aplinką vertėtų pasirinkti. Renkantis programavimo aplinkos sistemą analizei iškeliami tokie kriterijai:

- *suderinamumas* – pasirinkta duomenų bazių valdymo sistema turi palaikyti pagrindines programavimo kalbas, tokias kaip „C#“, „C++“ ar „VB“;
- *praplečiamumas* – duomenų bazių valdymo sistema turi veikti serveryje, kurio operacinė sistema gali būti ne tik „Windows“, bet ir „Linux“;
- *patogumas dirbtui* – kriterijus, nurodantis ar tai patentuotas įrankis, už kurį reikia mokėti ar laisvai naudojamas, ar jį nesunku perprasti, ar juo yra lengva išmokti naudotis;

Visual Studio. Ši integruota programavimo aplinka dažniausiai yra naudojama didelių įmonių. Suteikiama galimybė naudotis „.NET“^[12]. Galima pasirinkti kelias programavimo aplinkas, siūlančias skirtingą suteikiamų funkcionalumų kiekį priklausomai pagal vartotojo poreikį (yra ir nemokama versija). Turi „Intellisense“ – programuotojo pagalbininką. Pasižymi automatiniu kodo formatavimu, praplečiamumu – „Visual Studio Gallery“ arba „Nuget“ paketai, kurie suteikia programavimo aplinkai naujų galimybių. Įrankis naudoja daug kompiuterio resursų. Reikia įdarbio norint priprasti prie aplinkos dizaino, jo išdėstymo. Reikia įdėti daug pastangų norint suprasti ir išmokti visas aplinkos teikiamas galimybes.

Eclipse. Geriausia alternatyva, kuriems nepatinka „Visual Studio“ aplinka. Turi švarią ir patogią vartotojo sąsaja. Galima naudoti ir kitas programavimo kalbas, kurių nėra „.NET“ tinklalapio struktūros technologijoje. Taip pat naudojama kuriant „Android“ aplikacijas. Kalbų praplečiamumas – naudojamos kalbos nėra ribojamos ir bet kas šiai aplinkai gali kurti įskiepius, taip palengvinant darbą su šia aplinka. Galimybė ją naudoti „Linux“ sistemose. Nemokama. Reikia įdėti daug pastangų norint suprasti ir išmokti visas aplinkos teikiamas galimybes.

Netbeans. Dar viena alternatyvi programavimo aplinka, kuri yra praplečiama, nemokama ir skirta „Java“ programavimo kalbų. Kartu su šia aplinka yra įdiegiamas ir „Java“ serveris – „GlassFish“, kuris yra dažniausiai naudojamas kaip atskira testavimo aplinka. Pasižymi

praplečiamumu – galima kurti įskiepius. Aplinka nemokama. Naudoja daug kompiuterio resursų. Turi sunkiai įsimenamą dizainą, kuriame daug kur yra pateikiama vienodų formų langų.

Programavimo aplinkų analizės palyginimo rezultatai pateikiami 1.2 lentelėje.

1.2 lentelė. Programavimo aplinkų palyginimo lentelė

	Visual Studio	Eclipse	Netbeans
Suderinamumas	+	+	+/-
Praplečiamumas	+	+	+
Patogumas dirbt	+	+/-	+/-

Apibendrinant analizės rezultatus galima daryti išvadą, jog pagal pasirinktus kriterijus labiausiai tinkama yra „Visual Studio“ programavimo aplinka.

Jrankio pasirinkimo priežastys. Geriausiai atitinka išsikeltus kriterijus. Vieninga ekosistema su daugybe galimų pasirinkti įrankių. Patogi ir greita projektų kodo valdymo kontrolė „Source code control“: kodo rašymas, derinimas, optimizavimas. Galimybė perpanaudoti sukurtą kodą naudojant klasių bibliotekas. Daug įvairių naudingų plėtinių iš „NuGet“ paketų saugyklos.

Platesnis pasirinkto įrankio aprašymas. Modernus įrankių rinkinys, suteikiantis galimybę naudoti daugybę įvairių projektavimui, programavimui, testavimui skirtų įrankių, kurių reikės kuriamos sistemos ir modulių įgyvendinimui pasiekti. Lankstūs planavimo įrankiai, pvz.: apimties planavimo funkcija, užduočių lento, neatliktų darbų valdymo galimybė, leidžianti išnaudoti laipsniško programavimo metodus. Modernūs modeliavimo, aptikimo ir architektūros įrankiai, galimybė aprašyti sistemą ir užtikrinti, jog architektūros vizija bus išlaikyta ją įgyvendinant. Programuotojų ir vykdymo komandos sutelkimas naudojant „IntelliTrace“, „Operations Manager Connector“ bei „Preemptive Analytics“, kurie pagerina kodo kokybę ir sutrumpina laiką, kurio reikia sprendžiant problemas. Galimybė generuoti trikčių aprašus iš įdiegtos programinės įrangos. Naudojimasis skirtingais projektais: servisu, integracijų, internetinės aplikacijos, globalizacijos viename „ASP.NET“ sprendime. Programuojama vienoje ir toje pačioje integruotoje programavimo aplinkoje.

1.3.1.2. Duomenų bazių valdymo sistemas („MS SQL Server 2014“)

Šiais laikais jokia didesnė kompiuterinė sistema neapsieina ir be duomenų bazės, kuriai sukurti ir valdyti naudojama duomenų bazių valdymo sistema (DBVS). Duomenų bazės kūrimui renkamasi viena iš esamų DBVS [13]. Pasirinkimas didelis, todėl renkantis reikia atkreipti dėmesį į apsibrėžtus kriterijus, pagal kuriuos ir įvertinama, kurią technologiją pasirinkti. Renkantis duomenų bazių valdymo sistemą analizei yra iškeliami tokie kriterijai:

- *suderinamumas* – pasirinkta duomenų bazių valdymo sistema turi palaikyti pagrindines programavimo kalbas, tokias kaip „Java“, „C++“, „Python“ ar „PHP“;
- *skirtingų OS palaikymas* – duomenų bazių valdymo sistema turi veikti serveryje, kurio operacinė sistema gali būti ne tik „Windows“, bet ir „Linux“;
- *nemokama DBVS* – kriterijus, nurodantis ar tai patentuotas įrankis, už kurį reikia mokėti ar laisvai naudojamas;

Analizei pasirenkamos „MS SQL Server“, „Oracle“ ir „MySQL“ DBVS. „Oracle“ ir „MySQL“ duomenų bazių valdymo sistemos palaiko daugybę programavimo kalbų, išskaitant „Java“, „C++“, „Python“ ir „PHP“, o „MS SQL Server“ DBVS palaikomą kalbų kiekis gerokai mažesnis, tačiau yra palaikoma „C#“ kalba. Panaši situacija ir su operacių sistemų palaikymu. „Oracle“ ir „MySQL“ gali veikti serveriuose su „Linux“ operacine sistema, o „MS SQL Server“ veikia tik „Windows“ operacinėje sistemoje. „Oracle“ ir „MS SQL Server“ DBVS yra labiau skirtos didelėms komercinėms

sistemoms, todėl kai kurios jų versijos yra mokamos, o „MySQL“ yra įrankis, orientuotas į individualius interesus ir yra nemokamas. Duomenų bazių valdymo sistemų analizės rezultatai pateikiami 1.3 lentelėje.

1.3 lentelė. Duomenų bazių valdymo sistemų palyginimo lentelė

Lyginimo kriterijai	MS SQL Server	Oracle	MySQL
Suderinamumas	+	+	+
Skirtingų OS palaikymas	-	+	+
Nemokama DBVS	+/-	-	+

Apibendrinant rezultatus galima daryti išvadą, jog pagal pasirinktus kriterijus labiausiai tinkama yra „MySQL“ duomenų bazių valdymo sistema, tačiau nusprendėme pasirinkti „MS SQL“ įrankį.

Įrankio pasirinkimo priežastys. Patogi integracija su „Visual Studio“ įrankiu. Daugybė duomenų bazės valdymo pasirinkčių ir galimybių. Informacinių sistemų kurdami „Windows“ operacinėje sistemoje, naudodami „Visual Studio“ programavimo aplinką ir kitus šiame skyriuje išvardintus išanalizuotus ir pasirinktus programinius įrankius ir technologijas (serveri, servisai, programavimo kalbas), kurie tarpusavyje yra suderinti itin gerai, pasirinkome „MS SQL“ duomenų bazių valdymo sistemą.

Platesnis pasirinkto įrankio aprašymas. Įrankis, skirtas kurti ir valdyti šiuolaikines reliacinių duomenų bazių valdymo sistemas, suteikiantis galimybę naudotis įvairiais transakcijų valdymo mechanizmais, užklausų valdymo metodais, greitaveikos optimizavimo funkcijomis ir kitais įrankiais, reikalingais palaikyti kuriamos sistemos duomenų valdymą. Suteikiama galimybė:

- Programavimas „T-SQL“, duomenų ir failų tipais;
- Integruota grafinė duomenų bazių valdymo sistema (angl. „Management Studio“);
- Transakcijų valdymas (angl. „Online Transaction Processing“);
- Integruotos įvairių tipų duomenų jungtys;
- Duomenų integracija („SSIS, designer transforms“);
- Raportų kūrimas;

1.3.1.3. Sistemų gyvenimo ciklo valdymo sistema („Team Foundation Communication“)

Įrankio paskirtis. Įrankis, skirtas kuriamų aplikacijų gyvenimo ciklui valdyti. Galima valdyti projektus, jų gyvendinimo procesus, skirstyti darbus, peržvelgti komandos atliekamą veiklą, kurti ir diegti sistemų funkcionalumus, testuoti jų infrastruktūrą. Visa tai yra pasiekama sistemingai valdant visos komandos darbą.

Įrankio pasirinkimo priežastys. Ši sistema yra integruota kartu su pasirinktu „Visual Studio“ programavimo aplinkos įrankiu. Tai greitas ir paprastas sprendimas siekiant išnaudoti, nukreipti ir paskirstyti visų komandos narių darbą taip leidžiant pasiekti užsibrėžtus tikslus kuo įmanoma greičiau, tuo pačiu neprarandant darbo kokybės. Galimybė prisijungti iš bet kokio tinklo, kuris turi interneto prieigą.

1.3.1.4. Privataus tinklo prieigos sistema („Cisco System VPN Client 5.0“)

Įrankio paskirtis. „Cisco System VPN Client 5.0“ – programa, skirta prisijungimui prie virtualaus privataus tinklo („VPN network“). Klientas paverčia savo lokalius resursus prieinamais nuotoliniu būdu per saugų ir patikimą tinklą, tarsi vartotojas būtų prisijungęs tiesiogiai prie privataus tinklo. Šis įrankis yra reikalingas įmonių tinklų tarpusavio susiejimui.

Įrankio pasirinkimo priežastys: Patikimas ir užtikrintas, „Cisco Systems“ standartais paremtas įrankis.

1.3.1.5. Naršyklės („Internet Explorer“, „Mozilla Firefox“, „Chrome“, „Opera“, „Safari“)

Įrankių paskirtis. Įrankiai, skirti atvaizduoti internetinius puslapius žiniatinklyje, vidiniuose įmonės tinkluose ir savo kompiuteryje. Tai pagrindiniai įrankiai skirti pasiekti kuriamą informacinę sistemą. Sistemos kūrimo metu naršyklės pasitarnaus kaip ypač naudingi testavimo įrankiai, kurie turi itin daug galimybų suteikiančius programuotojų įrankius („Developer tools“): greitaveikos analizė, internetinių puslapių turinio (įskiepių, grafinių elementų, skriptų) peržvelgimas, kodo veikimo tikrinimas, kodo redagavimas ir kt. Visos išvardintos naršyklės bus naudojamos sistemos modulių kūrimo procese.

Įrankių pasirinkimo priežastys. Informacinių sistemos kūrimo procesas ir jos naudojimas yra neatsiejamas nuo internetinių naršyklų naudojimo.

1.3.1.6. Serveris („Internet Information Services (IIS“))

Įrankio paskirtis. Praplečiamų galimybių tinklo serveris, sukurtas „Microsoft“, palaikantis įvairių rūsių protokolus (HTTP, HTTPS, FTP, FTPS, SMTP, ir NNTP).

Įrankių pasirinkimo priežastys. Patogi integracija ir darbas tam skirtoje ir naudojamoje „Windows“ aplinkoje. Palaikomi visi modulių įgyvendinimui ir realizavimui reikalingi protokolai (HTTP, HTTPS, SMTP). Reikalingas norint tenkinti užsakovo turimą infrastruktūrą.

1.3.2. Technologijų analizė

1.3.2.1. Sistemų programavimo kalbos („C#“)

Kuriamos sistemos programavimo kalba yra atsakinga už sistemos realizavimo įgyvendinimą, veikimą ir valdymą [15]. Sistemos modulių programinio kodo rašymui reikia išsirinkti vieną iš galimų programavimo kalbų. Renkantis programavimo kalbą atliekama analizė pagal šiuos pasirinktus kriterijus:

- *suderinamumas* – kalbos sederinamumas su visa sistema;
- *darbo produktyvumas* – sugaisto laiko kuriant, idėtų pastangų ir tam tikrų kokybės aspektų, pvz.: našumo santykis, kuriant pasirinkus tam tikrą technologiją;
- *lankstumas* – kriterijus, nurodantis ar daug pastangų reikia norint sukurti tam tikrą funkcionalumą, jį atnaujinti ar integruoti į kitą sistemą;
- *galimybės* – konkrečios kūrimo technologijos siūlomos galimybės, pvz.: įrankiai, skirti darbui komandoje, kūrimo aplinkos ir pan.;
- *patirtis* – patirtis, naudojant kūrimo technologiją.

Tai gali būti viena iš populiarųjų programavimo kalbų: „VB“, „C#“, „PHP“ arba kokia nors kita kalba, tačiau renkantis tam tikrą specifinę kalbą gali tapti sunku integruoti komponentus į sistemą ir taip nukentėtų sederinamumas. Renkantis specifinę programavimo kalbą nukentėti gali ir lankstumas, kadangi gali susidaryti tokia situacija, jog kažkieno parašytą programinį kodą teks keisti ar papildyti kitam asmeniui, o kai kodas realizuotas tam tikra specifine kalba, valdymo komponentai tampa sunkiai palaikomi – atnaujinimui reikia idėti daug pastangų, o kartais net ir naujai suprojektuoti ir realizuoti kita kalba. Darbo produktyvumas, naudojant specifinę programavimo kalbą, gali ženkliai kristi, kadangi našumo optimizavimui gali tekti sunaudoti daug laiko ir idėti daug pastangų. Paskutinis kriterijus, nurodantis darbo komandoje galimybes, taip pat gali būti sunkiai išpildomas, pasirinkus specifinę programavimo kalbą, kadangi ne visoms kalboms yra sukurti komandinio darbo ar versijų kontrolės palaikymui reikalingi įrankiai. Taigi, atsižvelgiant į apžvelgtus kriterijus geriau rinktis vieną

iš populiarių programavimo kalbų valdymo komponentų kūrimui. Visos keturios paminėtos populiarosios programavimo kalbos yra gerai suderinamos su daugeliu sistemų, tačiau plačiausiai naudojamos yra „PHP“ ir „C#“ kalbos. Kiekviena šių kalbų yra lanksti, sukurtus komponentus lengva redaguoti ir integrnuoti kartu su kitais komponentais, tačiau ne visos turi komandiniam darbui ir versijų kontrolei palaikyti skirtus įrankius. „PHP“ ir „C#“ yra populariausios kalbos kuriant sistemas komandose. Paskutinis, bet ne mažiau svarbus kriterijus yra darbo su konkretičia programavimo kalba patirtis. Didžiausia patirtis kuriant programinius komponentus yra naudojant „C#“ programavimo kalbą. Programavimo kalbų, sistemas kūrimui, analizės rezultatai pateikti 1.4 lentelėje.

1.4 lentelė. Sistemos programavimo kalbos palyginimo lentelė

Lyginimo kriterijai	C#	VB	PHP
Suderinamumas	+	-	+
Darbo produktyvumas	+	+/-	+
Lankstumas	+/-	+	+
Galimybės	+	-	-
Patirtis	+/-	-	+/-

Atsižvelgiant į visų pasirinktų kriterijų palyginimo rezultatus ir didžiausią įtaką pasirenkamoms technologijoms turintiems veiksniams galima daryti išvadą, jog „C#“ yra tinkamiausia kalba valdymo komponentų kūrimui.

Technologijos pasirinkimo priežastys. Pasirinktoje „ASP .NET“ technologijoje palaikoma ir naudojama programavimo kalba. Patvirtinta „Ecma (ECMA-334)“ ir „ISO (ISO/IEC 23270:2006)“ standartu.

Platesnis pasirinktos technologijos aprašymas. Programavimo kalba, pasižyminti objektiškumu, sukurta „Microsoft“ kompanijoje kaip dalis „.NET“ iniciatyvos. Kalba paremta „C++“ bei „Java“ kalbomis. Savybės: Imperatyvi, deklaratyvi, funkcinė, bendarinė, orientuota į komponentiškumą palaikymą.

1.3.2.2. Internetinių taikymų kūrimo technologijų analizė („ASP.NET“)

Kuriama sistema prieigą prie jos suteiks internetinės naršykėlės pagalba naudojant internetinių taikymą. Internetinis puslapis rašomas naršyklei suprantama kalba – „HTML“ ir „CSS“ stilių derinimui, o dinaminio puslapio turinio realizavimui reikia rinktis vieną iš internetinių taikymų kūrimo technologijų. Renkantis technologiją analizei pasirenkami tokie kriterijai:

- *nemokama* – kriterijus, nusakantis ar tai nemokama internetinių taikymų kūrimo technologija;
- *prieiga prie „C#“ bibliotekų* – šis kriterijus nurodo ar naudojant technologiją galima prieiti prie pasirinktos programavimo kalbos – „C#“ bibliotekų;
- *objektiškai orientuota* – kriterijus, kuris nurodo ar technologija yra orientuota objektiškai ar struktūriškai;
- *patirtis* – kriterijus, kuris rodo ar turima patirties naudojant technologiją;

„JSP“ visada buvo nemokama technologija, kaip ir „PHP“ bei „ASP.NET“. Prie „C#“ bibliotekų patogią prieigą turi tik „ASP.NET“. Nei „PHP“, nei „JSP“ tokios galimybės nesuteikia. Tieki „JSP“, tieki „ASP.NET“ yra objektiškai orientuotos technologijos, dėl to parašytas programinis kodas yra aiškesnis, geriau palaikomas, lengviau aptinkamos ir ištaisomos klaidos. „PHP“ taip pat palaiko objektų kūrimą, tačiau objektinis modelis yra labai primityvus ir dauguma puslapių tiesiog ignoruoja šiuos objektus vietoje to naudodami paprastus kintamuosius. Darbo patirties naudojant „ASP.NET“

yra, priešingai nei „JSP“ ir „PHP“, kuriomis naudotis teko kur kas mažiau. Internetinių taikymų kūrimo technologijų analizės rezultatai pateikti 1.5 lentelėje.

1.5 lentelė. Internetinių taikymų kūrimo technologijų palyginimo lentelė

Lyginimo kriterijai	JSP/Servlet	ASP.NET	PHP
Nemokama	+	+	+
Prieiga prie C# bibliotekų	-	+	-
Objektiškai orientuota	+	+	+/-
Patirtis	+/-	+	+/-

Atsižvelgiant į analizės rezultatus, „ASP.NET“ yra priimtiniausia technologija internetinių taikymų kūrimui.

Platesnis pasirinktos technologijos aprašymas. „ASP.NET“ yra tinklalapio struktūros technologija, sukurta „Microsoft“, kurią programuotojai gali naudoti norėdami sukurti dinaminę internetinę svetainę, žiniatinklio konstrukciją arba paslaugą. Tai dalis „Microsoft .NET“ platformos. „ASP.NET“ integruota su Bendros kalbos išpildymo aplinka (angl. "Common Language Runtime", trump. CLR), leidžiančia programuotojams rašyti „ASP.NET“ kodą bet kuria „Microsoft .NET kalba“.

Technologijos pasirinkimo priežastys. Pasirinktas „Visual Studio“ įrankis palaiko šią tinklalapio struktūros technologiją, ji labiausiai yra išplėtota, palaikoma ir propaguojama šioje programavimo aplinkos įrankyje. Technologija suteikia be galio daug sistemų kūrimo procese reikalingo funkcionalumo – standartiškų klasių bibliotekų (laiko, datų, teksto formatavimo, susiejimo su duomenų bazėmis, servisų implementavimo ir kt.).

1.3.2.3. Žymėjimo kalba („HTML5“, „XML“)

Technologijos paskirtis. Tai kompiuterinės žymėjimo kalbos, naudojamos pateikti turinį interneite^[14]. Naujasis standartas smulkiai ir iki galio tiksliai aprašo žymėjimo gaires, nepalikdamas vietos interpretacijai – naršykės turės vienodai atvaizduoti tinklalapius, vienodai elgtis net ir su „HTML“ rašto klaidomis. Visa tai palengvins programuotojų ir dizainerių darbą. „XML“ yra labiau tinkamas aprašyti ir apibūdinti duomenis.

Technologijos pasirinkimo priežastys. Vienintelės ir nepakeičiamos, „W3“ standartizuotos tokio tipo technologijos. Reikalingos norint tenkinti užsakovo techninės specifikacijos reikalavimus.

1.3.2.4. Skriptų kalba („Javascript“, „Jquery“, „AJAX“)

Technologijos paskirtis. „Javascript“ - objektiškai orientuota skriptų programavimo kalba, besiremianti prototipų principu. Dažniausiai kalba naudojama internetinių puslapių interaktyvumo realizacijai, bet taip pat naudojama ir kaip galimybė skriptais manipuliuoti tam tikromis programomis.

Modulių reikalavimų įgyvendinimui pasirinktos naudoti „Javascript“ bibliotekos:

- „Jquery“ (ypatingai duomenų siuntimui ir gavimui skirta „AJAX“ technologija);
- „Select2“ – paieškos laukelio elementas;
- „Picker“ – datos parinkimui skirtas elementas;
- „Bootstrap-ColorPicker“ – spalvų parinkimui skirtas lentelės elementas;
- „CKEditor“ – teksto redaktorius;
- „Owl-Carousel“ – didelėmis galimybėmis pasižyminti slenkanti juosta;
- „FileUpload“ – dokumentų įkėlimui;
- „Validate“ – formų validacijai, duomenų užtikrintumui ir tinkamumui nustatyti;

Technologijos pasirinkimo priežastys: „Javascript“ yra vienintelės ir nepakeičiamą, W3 standartizuota tokio tipo technologija.

1.3.2.5. Stilių derinimo kalba („CSS3“, „Bootstrap CSS Framework“)

Technologijos paskirtis. Internetinių puslapių išvaizdą aprašančių elementų ir taisyklių rinkinys atskirtas nuo „HTML“ kalbos dokumento sandaros žymėjimo, siekiant atskirti teikiamus funkcionalumus. Lengva keisti visų puslapių išvaizdą, paprastesnis pasiekimas, kad puslapius skirtingose naršyklėse rodytų vienodai, atvaizduojant puslapį galima nesiųsti pačio „CSS“ dokumento.

„Bootstrap“ – struktūrinė technologija skirta labiau organizuoti kuriamų internetinių puslapių dizainą. Turinti atitinkamą kiekį savų aprašytų „CSS“ taisyklių rinkinį. Pasižymi: lankstumu – palaiko naujausius priimtus standartus. Patikimumu – yra ištestuota ir nepasižymi jokiomis klaidomis ar netikėtomis elementų išdėstymo ar atvaizdavimo elgsenomis. Palaiko prisitaikantį dizainą – galima naudoti įrenginius su įvairaus dydžio ekranais ir atitinkamai nuo jų atvaizduoti internetinio puslapio išvaizdą. Patogi „tinklelio“ angl. („Grid“) sistema – padeda lengvai ir greitai norimai išdėstyti puslapio elementus. Galimas naudojamų komponentų praplečiamumas ir lengvas jų pritaikymas savoms reikmėms (mygtukų grupės, navigacijos juostos, iššokantys pranešimai ir kt.). Gera ir kokybiška dokumentacija.

Technologijos pasirinkimo priežastys. Vienintelė ir nepakeičiama, „W3“ standartizuota tokio tipo technologija. Reikalinga norint tenkinti užsakovo reikalavimus. Lengvai išmokstama ir įsisavinama. Daug suteikiamų galimybių.

1.3.2.6. Užklausų kalba („Language Integrated Query (LINQ“), „T-SQL“)

Technologijos paskirtis. „LINQ“ – tai „NET“ komponentas, kuris praplečia naudojamą programavimo kalbą, suteikiant galimybę naudojant jos specialius kalbos išsireiškimus sukurti duomenų užklausas. Šie kalbos išsireiškimai yra paverčiami į „SQL“ užklausas, kurių gaunami rezultatai gali būti patogiai talpinami ir naudojami duomenų masyvuose, išvardijamosiose klasėse („enumerable classes“), „XML“ dokumentuose, duomenų bazėse ir kitose struktūrose. „T-SQL“ yra „Microsoft“ sukurtas „SQL“ plėtinys, kuris yra naudojamas ir palaikomas išskirtinai „Microsoft SQL Server“. „T-SQL“ išplečia „SQL“ standartą įtraukiant procedūrinį programavimą, lokalius kintamuosius, daug įvairaus galimybių naujų funkcijų duomenų apdorojimui, pakeitimų pašalinimo (angl. „DELETE“) ir atnaujinimo (angl. „UPDATE“) užklausų sakiniuose.

Technologijos pasirinkimo priežastys. Pasirinktoje „ASP .NET“ technologijoje palaikomas ir naudojamas komponentas. Supaprastina ir palengvina darbą su duomenų užklausomis. „T-SQL“ yra naudojama jau pasirinktame „MS SQL Server“ programiniame įrankyje.

1.3.2.7. Objektinis-reliacinis konvertuotojas („Entity Framework“)

Technologijos paskirtis. Viešo naudojamo ir licencijos projektas, skirtas tarpusavyje nesuderinamų duomenų tipų konvertavimui „Object-relational mapping“ („ORM“). Rezultate yra sukuriama virtuali objektų duomenų bazė, kuri gali būti panaudojama programavimo kalbos pagalba. Ši technologija taip pat yra „NET“ struktūros dalis. Su „Entity Framework“ programuotojai gali dirbti aukštesniame abstrakcijos lygmenyje, (dirbant su duomenimis) ir gali sukurti bei palaikyti sistemas rašant mažiau kodo, pvz.: kuriant esybes, ryšius tarp jų ir bendrą verslo logiką internetiniame projekte, to paties atliliki neberekia duomenų bazės lygmenyje – atitinkami ryšiai, duomenų bazės lentelės yra sukuriama „Entity Framework“ dėka.

Technologijos pasirinkimo priežastys. Pasirinktoje „ASP .NET“ technologijoje palaikomas ir naudojamas komponentas. Palengvina programuotojų darbą, sutupo laiko, sumažina reikalingo kodo galimų padaryti klaidų kiekį.

1.3.3. Protokolų analizė

1.3.3.1. Servisų kūrimo technologijų analizė („Windows Communication Foundation (WCF)“)

Medicininės informacinės sistemos moduliai kuriami taip, kad jų teikiamomis paslaugomis ir funkcionalumais būtų galima naudotis internetu per naršyklę. Tam, kad moduliai galėtų bendradarbiauti tarpusavyje tarp kitų sistemos modulių, komponentų ar gauti informacijos iš kitų reikalingų informacinių sistemų bus naudojami tinkliniai servisai. Prieš kuriant ir naudojant tinklinį servisą reikia pasirinkti kokiomis technologijomis jis bus realizuojamas. Renkantis tinklinio serviso kūrimo technologijas analizei parenkami tokie kriterijai:

- *suderinamumas* – kriterijus, nusakantis ar tinklinio serviso kūrimo technologija palaiko valdymo komponentų kūrimui naudojamą programavimo kalbą – „C#“.
- *mažas programinio kodo kiekis* – kriterijus, nurodantis ar daug programinio kodo reikia parašyti norint sukurti tinklinį servisą;
- *lankstumas* – kriterijus, nurodantis ar yra palaikomas platus tinklinių servisų standartų pasirinkimas;
- *efektyvumas* – kriterijus, kuris nurodo tinklinių servisų kūrimo technologijos efektyvumą, kaip efektyviai panaudojami skaičiavimai;

„Apache Axis2“ ir „Spring WS“ palaiko keletą programavimo kalbų, jų tarpe ir „Java“, tačiau nėra pilnai sunderinami su „JAX-WS“ bei su „C#“ programavimo kalba, taip pat kaip ir „Apache CXF“. Naudojant „WCF“ serviso kūrimui reikia parašyti mažiausiai programinio kodo, „Apache CXF“ jo reikalauja daugiau, o „Apache AXIS2“ daugiausiai iš visų trijų technologijų. „WCF“, „Apache AXIS2“, „Apache CXF“ yra architektūriškai lankstūs, nes palaiko daugumą tinklinių servisų standartų, o „Spring WS“ architektūrinis lankstumas yra daug mažesnis. „WCF“ suteikia didžiausią efektyvumą atliekant mažiausiai papildomų skaičiavimų. Tinklinių servisų kūrimo technologijų analizės rezultatai pateiki 1.6 lentelėje.

1.6 lentelė. Skirtingų technologijų, web serviso kūrimui palyginimo lentelė

Lyginimo kriterijai	Apache AXIS2	WCF	Apache CXF	Spring WS
Suderinamumas	+/-	+	+	+/-
Mažas programinio kodo kiekis	-	+	+/-	+
Lankstumas	+	+	+	-
Efektyvumas	+/-	+	+	+/-

Pagal pasirinktus kriterijus renkamasi labiausiai tinkamą „WCF“ tinklinių servisų kūrimo technologiją.

Platesnis pasirinktos servisų kūrimo technologijos aprašymas. Tai „NET API“, skirtas apjungti kelias tarpusavyje bendraujančias informacines sistemas, siekiant mainytis duomenimis tarp sistemų. Klientas gali naudotis keliais servisais vienu metu, o servisais gali naudotis daug klientų. Servisai turi „WSDL“ interfeisą (angl. „Web Services Description Language“) per kurį bet kuris „WCF“ klientas gali naudoti servisą, nepriklausomai nuo platformos, kurioje yra patalpinta informacinė sistema.

Servisų kūrimo technologijos pasirinkimo priežastys. Kadangi kuriamai sistemai vis dar reikės bendrauti su dar kurį laiką palaikoma ankstesne sistema ir tarpusavyje mainytis duomenis, o ir mūsų pasirinkta „ASP .NET technologija jau turi savyje siūlomą sprendimo būdą į mūsų norimą įgyvendinti funkcionalumą, naudosime „WCF“ įrankį.

1.3.3.2. Kuriamos sistemos protokolų pasirinkimas

Modulių integracija ir reikalingi duomenų mainai su kitomis sistemomis bus reikalingi: apsilankymų modulyje – norint sužinoti konkretaus gydytojo ir jo turimos specialybės laisvų ir užimtų

talonų duomenis (laiką, apsilankymo vietą ir kt.), mokymų modulyje – norint sužinoti ir atnaujinti reikiamą informaciją iš kitoje sistemoje pateikiamos ir ruošiamos mokymo medžiagos.

Pranešimų siuntimas, informuojant sistemos vartotojus, bus reikalingas:

- Apsilankymų modulyje – užregistruvus apsilankymą, priminimui apie artėjantį apsilankymą, priminimui apie atsilaisvinusį ankstesnį taloną;
- Nuotolinių stebėjimų modulyje – informuojant pacientus apie praleistus duomenų pateikimus, esant kritiškiems pateiktų rodiklių rezultatams;

Bus naudojamas „**SMTP**“ (angl. „Simple Mail Transfer Protocol“) protokolas, kurį naudoja pasirinktas internetinis serveris „Internet Information Services (IIS)“ [16].

Pasirinkdami naudoti „WCF“ technologiją, atsižvelgėme į iškeltus techninės specifikacijos reikalavimus, dėl to „WCF“ palaiko visus mums reikalingus ir naudojamus protokolus [17]:

- „**HTTP / HTTPS**“ – komunikacijos protokolas / saugus komunikacijos protokolas;
- „**TCP**“ – standartinis duomenų perdavimo protokolų rinkinys;
- „**MSMQ**“ – pranešimų siuntimų protokolas;
- „**NamedPipes**“ – komunikacijos protokolas tarp serverio ir vieno ar daugelio to serverio klientų;

Atsižvelgiant į reikalavimus, galime numatyti, kuriais OSI modelio lygmenimis kuriami sistemos moduliai naudosios:

- V Sesijos lygis, kuris aprašo duomenų apsikeitimo tarp galinių sistemų taisykles vienos jungties ribose. Prieš užmezgant sesiją, abu komunikujantys taškai turi suderinti ryšio parametrus, kaip greitis, klaidų kontrolė ir pan. Atlieka vartotojų identifikavimą.
- VI Atvaizdavimo lygis, kuris nusako duomenų kodavimo sesijos metu taisykles. Pvz.: „MIME“. Užtikrina, kad informacija pasiusta vienos sistemos taikymo lygio, būtų priimta kitos sistemos taikymo lygio. Šiame lygyje atliekamas informacijos formato keitimas ir vertimas.
- VII Taikymo lygis, kuris apibrėžia tinklo teikiamas paslaugas vartotojų programoms. Šis lygis apibrėžia sistemas, kurioms reikia siusti informaciją tinklu.

1.4. Analizės išvados

Atlikta panašių sistemų ir programinių įrankių, technologijų, naudojamų protokolų analizė. Panašių sistemų analizėje kuriamą sistema palyginta su „pasveik.lt“, „sergu.lt“, „santa.lt“, „kaunoklinikos.lt“ medicininėmis informacinėmis pagal išsikeltus kriterijus – apsilankymų planavimą, savo sveikatos būklės stebėjimą, galimybę sužinoti apie profilaktines programas, galimybę padėti sau, bendrą paslaugą kokybę. Atlikta analizė parodė esamus šalies medicininių informacinių sistemų trūkumus ir kuriamos sistemos modulių galimus jų sprendimo būdus. Pagal išsikeltus kriterijus kuriamą sistema yra pranašesnė už likusias. Atlikta programinių įrankių, technologijų, reikalingų protokolų analizė. Programinių įrankių analizėje pasirinkta „Visual Studio“ programavimo aplinka, „MS SQL Server“ duomenų bazių valdymo sistema, „Visual Studio Team Foundation Server“ sistemos gyvenimo ciklo valdymo sistema, „Cisco System VPN Client“ privataus tinklo prieigos programa, „Internet Explorer“, „Mozilla Firefox“, „Chrome“, „Opera“, „Safari“ interneto naršyklės, IIS serveris. Technologijų analizėje pasirinkta C# programavimo kalba, „ASP.NET“ internetinių taikymų kūrimo technologija, „HTML5“, „XML“, „Javascript“, „Jquery“, „AJAX“, „CSS3“, „Bootstrap“, „LINQ“, „Entity Framework“ technologijos. Protokolų analizėje pasirinktas „Windows Communication Foundation“ servisių kūrimo technologija, išsiaiškinti ir aptarti sistemos moduliams reikalingi „SMTP“, „HTTP/HTTPS“, „TCP“, „MSMQ“ protokolai. Analizės metu įgautos žinios apie kiekvieną įrankį, leidžia labiau suprasti savo įgyvendinamų modulių realizavimo galimybes, galimas kliūtis ir kaip jų bus galima išvengti pritaikant skirtingus, įrankius ir technologijas. Gauti rezultatai padės tolimesniuose sistemos modulių projektavimo etapuose, ir jų realizacijoje.

2. PROJEKTAS

Atlikus analizę prieš realizuojant kuriamos sistemos modulius reikia juos suprojektuoti. Šiame skyriuje detaliai aprašomas projektavimo etapas. Projektuojant pirmiausiai dėmesys yra teikiamas vartotojams. Reikia išsiaiškinti, kas bus sistemos vartotojai, nustatyti ir suderinti jų reikalavimus ir tik tada galima projektuoti sistemos architektūrą.

2.1. Reikalavimų specifikacija

Produkto apibréžimas. Bus prisdėta prie kuriamos medicininės informacinės sistemos, įgyvendinant tris paslaugų suteikimo modulius: pacientų nuotolinių stebėjimų, apsilankymų ir švietimo. Šių modulių funkcionalumas taps aiškus peržiūrėjus kiekvieno iš jų pateikiamus funkcinius reikalavimus, o bendra jų integracija sistemoje išryškės peržvelgus nefunkcinius ir kitus likusius specifikacinius reikalavimus.

Vartotojų apibūdinimas. Informacinės sistemos vartotojas – tai asmuo, besinaudojantis IS. Su informacijos sistema numatyta, kad kuriamais moduliais dirbs/naudosis dvi išskiriančios vartotojų grupės: pacientai ir darbuotojai, kurios funkcinių reikalavimų aprašyme bus detalizuotos labiau.

Sistemos ir modulių vartotojas, norėdamas dirbtį su sistema privalėti elementarius darbo su internetinėmis naršyklėmis įgūdžius. Vartotojui turėtų būti aiški naršyklės lange pateikiama tekstinė informacija (lietuvių arba anglų kalba), jis turėtų turėti bazinius grafikų ir suvestinių skaitymo bei supratimo įgūdžius.

2.1.1. Projekto apribojimai

Sistemos aplinkos. Projekto eigoje turi būti paruoštos ir bus naudojamos sistemos aplinkos: testavimo aplinka, gamybinė aplinka, vystymo aplinka. Vystymo aplinkos saugojimo vieta bus suderinta rengiant projekto planą, turi būti galimybė šią aplinką saugoti pas tiekėją. Kitoms sistemos aplinkoms (testavimo ir gamybinių) resursus suteiks jos užsakovas. Tieka bus atsakingas už aplinką paruošimą, diegimo instrukcijos parengimą ir užsakovo personalo apmokymą valdyti aplinkas. Testavimo aplinka, gamybinė aplinka turi funkcionuoti ir po sistemos starto visą sistemos eksploatacijos laiką šio laiko neribojant. Tieka įsipareigoja suteikti visas reikalingas licencijas tokį aplinką funkcionavimui (išskyrus sisteminię programinę įrangą, kuri bus įsigyjama atskiru pirkimu).

Bendradarbiaujančios sistemos. Visos integracinių sąsajos bei duomenų mainai su išorinėmis bei su Klinikų esamomis sistemomis turi būti realizuoti duomenų mainų priemones funkcionalumo pagalba. Duomenų mainų priemonė turi užtikrinti perduodamų duomenų formatų keitimą, transformavimą (jeigu toks reikalingas) ir šiu duomenų perdavimą kitoms informaciniems sistemoms. Duomenų mainų priemonės pagrindinė paskirtis – paskirstyti ir valdyti vykstančius duomenų mainus tarp e. paslaugų teikimo proceso dalyvių. Tuo tikslu turi būti aprašyti visi dalyvaujantys duomenų rinkiniai (šaltiniai). Turi būti galimybė metaduomenims apibréžti realizuotinas duomenų mainų sąsajas su kitomis sistemomis ir pateikti atitinkamus aprašymus duomenų mainų priemonei. Duomenų mainų priemonės teikiamas paslaugos turės būti realizuotos kaip žiniatinklio paslaugos (angl. web services).

Našumas ir greitaveika. Atsako sparta, atmetus duomenų užlaikymą duomenų perdavimo linijose, turi būti (ne mažiau kaip 90% veikimo atvejų). Turi būti užtikrinti sistemos atsako spartos parametrai, įvertinant, kad sistemoje vienu metu, lygiagrečiai veiksmus iniciuos ne mažiau nei 3000 naudotojų. Reikalavimas taikomas darant prielaidą, kad tipinė sistemos naudotojo darbo vieta atitiks šiuos parametrus:

- Pagrindinis procesorius - ne prastesnis Intel Pentium 4 2.0 GHz;
- Operatyvioji atmintis - nemažiau nei 1024 MB;
- Kietojo disko talpa - ne mažesnė nei 40 GB;
- Tinklo pralaidumas – ne mažesnis nei 100 Mbps;

Su sistema turi būti galima dirbti kol vykdomi kiti darbai, pavyzdžiui, atliekamų paketinių užduočių veiksmai, registravimai. Turi būti užtikrinta, kad atliekami sistemos naudotojų veiksmai neblokuotų kitų naudotojų veiksmų ir nedarytų įtakos sistemos greitaveikai (išskyrus atvejus, kai duomenų integralumo sumetimais, naudotojams blokuojama prieiga prie tuo metu kitų naudotojų ar e. paslaugų gavėjų tvarkomų duomenų).

Projekto finansavimo šaltinis. Projektas finansuojamas Europos regioninės plėtros fondo ir Lietuvos Respublikos valstybės biudžeto bendrojo finansavimo lėšomis pagal 2007–2013 m. Ekonomikos augimo veiksmų programos 3 prioriteto „Informacinė visuomenė visiems“ įgyvendinimo priemonę Nr. VP2-3.1-IVPK-14-K „Pažangios elektroninės paslaugos“.

Darbo vietas aplinka. Sistemoje turi būti realizuota naudotojo ir paslaugų vartotojo sėsaja, veikianti žiniatinklio principu (angl. Web–Client), t. y. kai pagrindinė veiklos logika yra realizuota serveryje. Prisijungimas prie sistemos turi vykti interneto naršyklije, nediegiant jokios papildomos programines įrangos. Prisijungti pacientams prie sistemos turi būti galima iš bet kurios kompiuterinės darbo vietas, turinčios prieigą prie interneto. Prisijungimas sistemos darbuotojams ir administratoriams turi būti vykdomas tik iš vidinio kompiuterių tinklo.

2.1.2. Funkciniai reikalavimai

Sudarant funkcinius reikalavimus pirmiausia reikia apsibrėžti, kokie yra kuriamos sistemos modulių aktoriai, kad galėtume sudaryti kiekvieno aktoriaus funkcijų hierarchijos diagramas ir jas detalizuoti. Sistemos moduliuose bus šie aktorių, tikslinių grupių tipai:

- Nuotolinių stebėjimų modulyje – pacientas, įrenginių administratorius, atsakingas gydytojas, gydytojas specialistas.
- Apsilankymų modulyje – pacientas, KK atsakingas darbuotojas.
- Švietimo modulyje – pacientas, mokymų programų administratorius.

Detalizuosime kiekvieną iš esamų sistemos modulių aktorius, tikslines grupes, jų pareigas ir roles. Visas sistemos tikslines grupes bendrai apimanti sąvoka – *sistemos naudotojas*.

1. *Pacientas* – fizinis asmuo, galintis naudotis sistemą sudarančiu posistemų funkcijomis ir sistemoje saugoma informacija, jeigu jie prieš pradedami naudotis sistema patvirtina sutinkantys su sistemos naudojamą reglamentuojančiais dokumentais ir teisės aktais. Jie gali naudotis sistemą sudarančiu nuotolinių stebėjimų, apsilankymų, švietimo posistemų funkcijomis ir sistemoje saugoma informacija, susijusia su jiems teikiama sveikatinimo paslaugomis;
2. *Įrenginių administratorius* – yra Kauno klinikose dirbantis asmuo, turintis teisę sistemoje atliliki jam pavestas funkcijas, pvz.: šiuo atveju registruoti, priskirti įrenginius, fiksuoti išdavimą, grąžinimą, kitaip tvarkyti nuotoliniuose stebėjimuose naudojamus įrenginius);
3. *Atsakingas gydytojas* – Kauno klinikose dirbantis specialistas, kuris užsiima asmens sveikatos priežiūros, visuomenės sveikatos priežiūros, ar kita sveikatinimo veikla, kurios rūšis ir reikalavimus ją vykdantiems subjektams nustato Sveikatos apsaugos ministerija;
4. *Atsakingas specialistas* – yra Kauno klinikose dirbantis specialistas, kuris užsiima asmens sveikatos priežiūros, visuomenės sveikatos priežiūros, ar kita sveikatinimo veikla, kurios rūšis ir reikalavimus ją vykdantiems subjektams nustato Sveikatos apsaugos ministerija;
5. *KK atsakingas darbuotojas* – yra Kauno klinikose dirbantis asmuo vykdantis priskirtos e. paslaugos tvarkymą, pvz.: šiuo atveju tvarko ir peržiūri apsilankymų informaciją, jų susijusį turinį bei duomenų mainų funkcinius komponentus, stebi paslaugos suteikimo procesus;
6. *Mokymų administratorius* - yra Kauno klinikose dirbantis asmuo, turintis teisę sistemoje atliliki

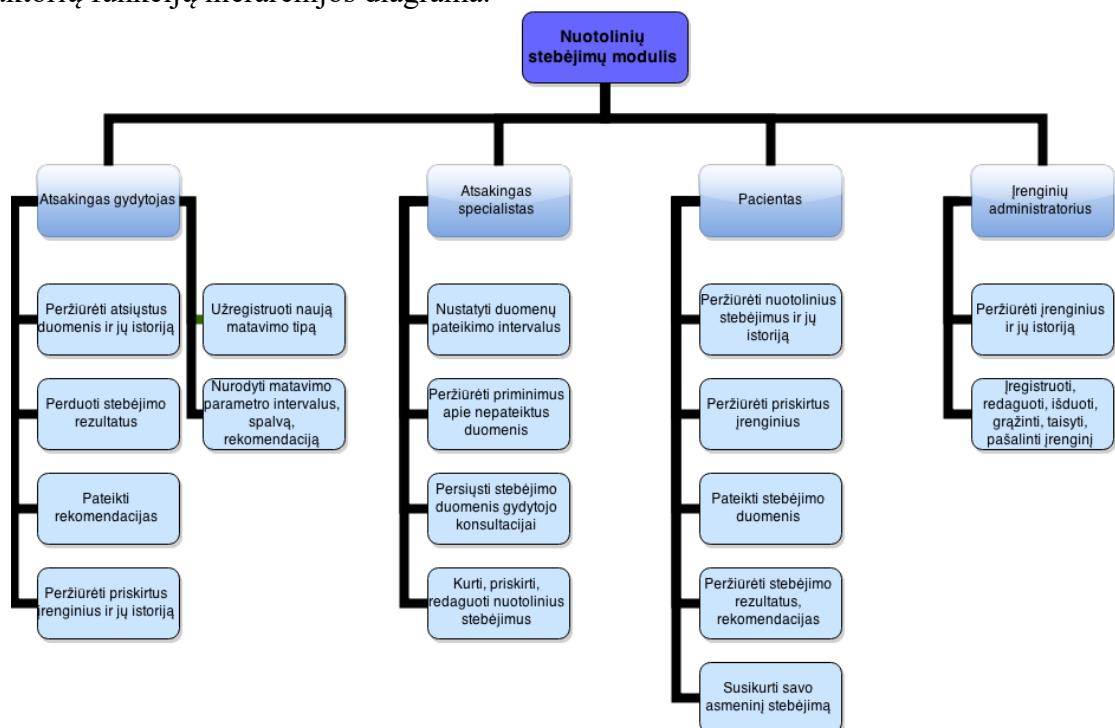
jam pavestas funkcijas, pvz.: šiuo atveju tvarkyti ir peržiūrėti gaunamas pastabas apie mokymų paketus, pateikti pacientui aktualios informacijos. Turi teisę vykdyti priskirtos e. paslaugos informacinį aprūpinimą;

7. *Serveris* – medicininės informacinės sistemos (jų kuriamų modulių) aktorius, atliekantis pagrindinį vaidmenį veikiant sistemai. Serveris surenka duomenis iš teikiamų paslaugų modulių, išsaugo juos duomenų bazėje, paima išsaugotus duomenis iš duomenų bazės ir pagal šiuos duomenis, bei valdymo logiką siunčia ją sistemos naudotojams;
8. *Bendradarbiaujanti sistema (HIS)* – yra kitą sistemą apibūdinantis aktorius, kuris atlikis informacijos šaltinio vaidmenį. Tarp serverio ir bendradarbiaujančios sistemos vyks įvairūs informacijos mainai pvz.: pateikti apsilankymų, pacientų duomenis ir kt.

Toliau aptarsime kiekvieną kuriamą sistemos modulį, plačiau išnagrinėsime jų teikiamą naudą ir ką yra planuojama įgyvendinti jų kūrimo procese. Tuomet pateiksime šių modulių funkcinius reikalavimus [24].

2.1.2.1. Nuotolinių stebėjimų modulio funkciniai reikalavimai:

Šia paslauga siekiama užtikrinti sunkiomis ligų formomis sergančių Lietuvos gyventojų būklės monitoringą ir kontrolę, naudojant elektroninius stebėjimo prietaisus. Ši elektroninė paslauga turi suteikti galimybę pacientui, kuris yra nuolat stebimas ir konsultuojamas Kauno klinikų gydytojų specialistų, nuotoliniu būdu teikti jiems detalią informaciją apie savo sveikatos būklę, jo ligos progresą apibūdinančius simptomus, ilgalaikio sveikatos stebėjimo rodiklius, kuriuos identifikuos bei perduos į sistemą specialūs elektroniniai stebėjimo prietaisai arba patys vartotojai. Sunkių pacientų būklės nuotolinio stebėjimo paslauga leis Kauno klinikų gydytojui specialistui kontroliuoti paciento būklę nuotoliniu būdu, kai šeimos gydytojo kompetencija yra nepakankama. Sistema turi sudaryti galimybes gydytojams interaktyviai individualizuotai konsultuoti pacientus ir keisti gydymo planą. Projekto apimtyje orientuojamasi į sveikatos priežiūros paslaugų teikimą elektroniniu būdu visiems Kauno klinikų pacientams. 2.1 pav. pateikiamas nuotolinių stebėjimų modulio aktorių funkcijų hierarchijos diagrama.



2.1 pav. Nuotolinių stebėjimo modulio funkcių hierarchijos diagrama

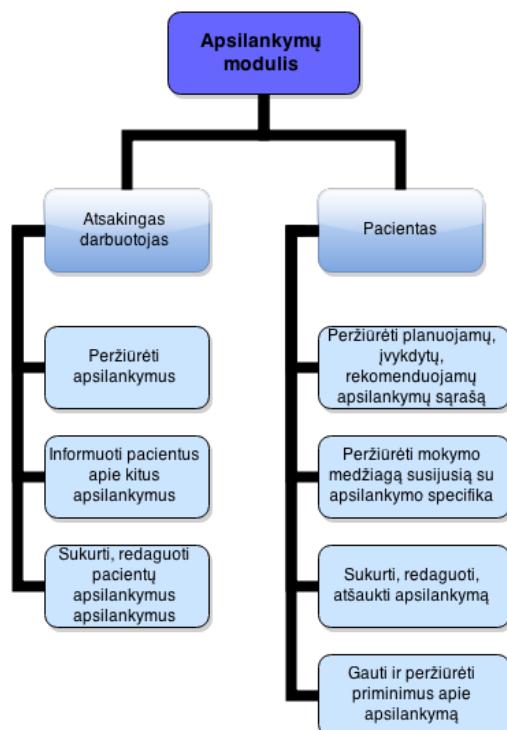
2.1 lentelė. Nuotolinių stebėjimų modulio funkcinių reikalavimų lentelė:

Nr.	Reikalavimas
1.	<p>Peržiūrėti atsiųstus duomenis:</p> <ul style="list-style-type: none"> Atsakingas gydytojas turi turėti galimybę matyti paciento siunčiamų duomenų istoriją ir peržiūrėti atsiųstus matavimo rezultatus, naudojant matavimo įrenginio specializuotą programinę įrangą, peržiūrėti duomenų stebėjimo rezultatus, pateiktus per portalą specializuotą formą, kaip grafiką už pasirinktą periodą. Grafike turi būti vaizduojamas vienas iš pasirinktų matavimo parametrų pokyčiai bei normos viršutinė ir apatinė reikšmės, peržiūrėjęs duomenų stebėjimo rezultatus, juos perduoti gydytojui-specialistui tolimesnei peržiūrai, užregistruoti naują matavimo tipą ir nurodyti skirtinges matavimo parametru intervalus, jų atvaizdavimo spalvą ir pranešimą (rekomendaciją), kurią reikia pateikti naudotojui;
2.	<p>Valdyti paciento duomenų nuolatinės peržiūros procesą (priminimai, kontrolė, prioritetų nustatymas):</p> <ul style="list-style-type: none"> Atsakingas sveikatos priežiūros specialistas turi turėti galimybę nustatyti pacientui rekomenduojamus laiko intervalus, kada jis turi siųsti stebėjimo duomenis, matyti pacientui išsiųstus priminimus apie nepateiktus duomenis, persiųsti stebėjimo duomenis kartu su papildoma informacija apie pacientą gydytojo-specialisto konsultacijai;
3.	<p>Pateikti pacientui rekomendacijas dėl tolimesnių veiksmų (per portalą, el. paštu):</p> <ul style="list-style-type: none"> Atsakingas gydytojas, gydytojas-specialistas turi turėti galimybę pateikti pacientui rekomendacijas dėl tolimesnių veiksmų, pasirenkant rekomendacijas iš standartinio sąrašo (pvz. apsilankytį pas gydytoja) arba surašant laisvos formos tekstą. Atsakingas gydytojas turi turėti galimybę inicijuoti parengtų rekomendacijų siuntimą pacientui nurodytu kanalu;
4.	<p>Registruoti, išregistruoti ir kitaip tvarkyti pacientų stebėjimo procese naudojamus įrenginius:</p> <ul style="list-style-type: none"> Įrenginių administratorius turi turėti galimybę įregistruoti įrenginį sistemoje, redaguoti įrenginio duomenis, pašalinti įrenginį iš sistemos (pažymėti kaip nenaudojamą), matyti aktualiam įrenginių sąraše įrenginio pagrindinius duomenis, įrenginio aktualią būseną (laisvas, priskirtas pacientui, taisomas - būsenos turi būti patikslintos analizes etapo metu), pagrindinius paciento, kuriam priskirtas įrenginys, duomenis, peržiūrėti įrenginių naudojimo istoriją pagal įrenginį, pacientą, priskyrimo laiką ir kitus filtravimo kriterijus, kurie turi būti detalizuoti analizes etapo metu;
5.	<ul style="list-style-type: none"> Atsakingas gydytojas turi turėti galimybę peržiūrėti įrenginių, priskirtų jo kuruojamiams pacientams, aktualų sąrašą, istoriją; Pacientas turi turėti galimybę peržiūrėti jam priskirtų įrenginių duomenis, įrenginių istoriją;
6.	<p>Fiksuoti įrenginių išdavimo, grąžinimo ir techninės priežiūros įvykius:</p> <ul style="list-style-type: none"> Įrenginių administratorius turi turėti galimybę fiksuoti įrenginio išdavimo pacientui faktą, įrenginio grąžinimo faktą, įrenginio perdavimo remontui faktą, nurodydamas gedimo registravimo duomenis (turi būti detalizuojama analizės etapo metu), įrenginio paėmimo iš remonto faktą, planuoti periodinę įrenginių patikrą, apdoroti (detalizuoti ir patvirtinti) paciento inicijuotą gedimo registravimą; Pacientas turi turėti galimybę inicijuoti įrenginio gedimo registravimą;
7.	<p>Priskirti įrenginius pacientams:</p> <ul style="list-style-type: none"> Įrenginių administratorius turi turėti galimybę inicijuoti įrenginio išdavimo procesą iš pacientų sąrašo su kuriais yra sudarytos įrenginių panaudos sutartys; Atsakingas gydytojas turi turėti galimybę sudaryti telemetrinių duomenų teikimo grafiką;

- | | |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> Įrenginių administratorius turi turėti galimybę susirasti pacientą, kuriam norės priskirti įrenginį, pagal pagrindinius paciento duomenis (vardą ir pavardę, diagnozę, priskirtą gydytoją), priskirti įrenginį pacientui, nurodant priskyrimo pabaigos datą arba neterminuotai, pakeisti paciento įrenginį, įrenginį turintį pacientą; |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2.1.2.2. Apsilankymų modulio funkciniai reikalavimai:

Perkėlus paslaugą į elektroninę erdvę, vartotojo profilyje turi būti pateikiama informacija apie paciento profilaktinius patikrinimus, stacionarizavimo laiką, priminimai apie užregistruotus vizitus, informacija apie paskirtą šeimos gydytoją, naujas Kauno klinikose vykdomas prevencines programas, naują įrangą ar naujas teikiamas paslaugas. Pacientams turi atsirasti galimybė aktyviai bendradarbiauti su Kauno klinikų darbuotojais, išspėti ištaigos darbuotojus apie tai, kad negali atvykti numatytu laiku planinei konsultacijai, tyrimui ar gydymui (operaciniam gydymui, chemoterapijai, kt.) – iki šiol minėti klausimai sprendžiami telefonu arba visai nevyksta bendravimas. 2.2 pav. pateikiama apsilankymų modulio aktorių funkcijų hierarchijos diagrama.



2.2 pav. Apsilankymų modulio funkcijų hierarchijos diagrama

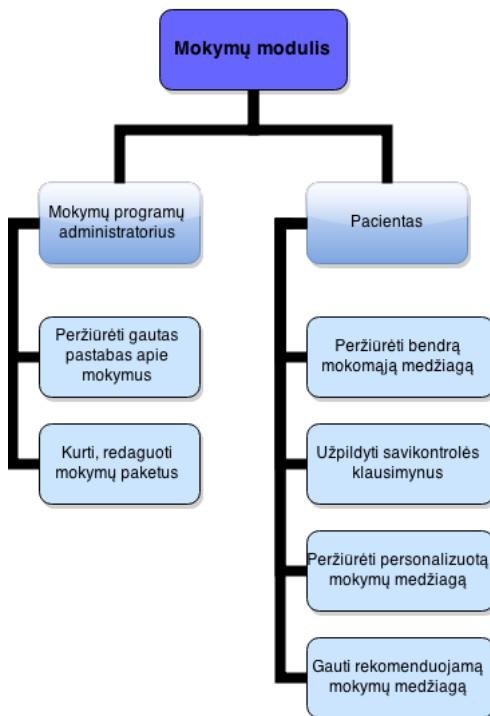
2.2 lentelė. Pacientų apsilankymų modulio funkcinių reikalavimų lentelė:

Nr.	Reikalavimas
1.	Peržiūrėti suplanuotų apsilankymų grafiką: <ul style="list-style-type: none"> Pacientas turi turėti galimybę peržiūrėti sąrašą suplanuotų apsilankymų/vizitų, detalesnę planuojamo vizito informaciją: numatomos paslaugos, data, laikas, gydytojo kabinetas, apsilankymo grafikus, pasirenkant vieną iš sistemojenumatyčių detalumo lygių: vienos dienos, savaites, mėnesio, 3-jų mėnesių;
2.	Peržiūrėti mokymų ir švietimo modulio paketus, susijusius su apsilankymo specifika (pvz. informaciją apie planuojamą procedūrą, ligoninės paciento atmintines ir pan.): <ul style="list-style-type: none"> Pacientas detalios peržiūros lange turi turėti galimybę peržiūrėti su paslaugos teikimu susijusį mokymų paketą ir rekomendacijų sąrašą, inicijuoti susipažinimą su paslaugos suteikimu susijusiu mokymu paketu;

3.	<p>Informuoti apie tai, kad negali atvykti numatytu laiku ir pakeisti numatyta apsilankymo datą ir laiką:</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę atšaukti numatyta vizitą, nurodant atsisakymo priežastį, peržiūrėti kitų galimų apsilankymų kalendorinį grafiką ir pasirinkti jam tinkamą datą; • Pacientas, kuris nori pakeisti apsilankymo datą ir/ar laiką, turi turėti galimybę peržiūrėti įmanomus (sistemos siūlomus) variantus, nurodant pageidaujamus parametrus: datų intervalą, laiko intervalą, savaites dienas, konkretų gydytoją;
4.	<p>Gauti informaciją apie galimybes apsilankytį anksciau, negu buvo numatyta ir pakeisti vizito datą ir laiką:</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę gauti elektroniniu paštu arba matyti portalą pranešimą apie tai, kad atsilaisvino kitas laikas, ankstesnis negu jo pasirinkta data, rezervuoti jam pasiūlytą apsilankymo laiką;
5.	<p>Gauti rekomendacijas apie profilaktinius apsilankymus (pvz. pagal ligos pobūdį arba rizikos grupę):</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę matyti sąrašą jam rekomenduojamų apsilankymų, jeigu tokias rekomendacijas pateike gydytojas ankstesnių vizitų metu, iškarto patvirtinti siūlomo apsilankymo datą ir laiką arba inicijuoti jų pakeitimą. • Sveikatos priežiūros specialistas turi turėti galimybę priskirti pacientui rekomenduojamą apsilankymų grafiką, pasirenkant iš tipinių grafikų sąrašo, priskiriant pradžios datą ir papildomus parametrus (rezervuoti į priekį kiek leidžia rezervavimo sistema, ar fiksuoти konkretią savaitęs dieną, prieš kiek dienų priminti pacientui ir pan.);
6.	<p>Inicijuoti registraciją konsultacijai:</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę inicijuoti registraciją pas pasirinktą specialistą;
7.	<p>Peržiūrėti sąrašą pacientų, kurie negali atvykti ar nori pakeisti apsilankymo datą:</p> <ul style="list-style-type: none"> • Atsakingas darbuotojas turi turėti galimybę peržiūrėti sąrašą pranešimų apie pacientų neatvykimą ir\arba prašymus pakeisti atvykimo laiką, perkelti kitų pacientų apsilankymą į atsilaisvintą datą ir laiką bei informuoti pacientą apie naują vizito datą.;
8.	<p>Inicijuoti kitų pacientų informavimą apie vizito datos pakeitimo galimybes:</p> <ul style="list-style-type: none"> • Atsakingas darbuotojas turi turėti galimybę informuoti pasirinktą pacientų grupę apie galimybę pakeisti apsilankymo datą ir laiką;
9.	<p>Patvirtinti apsilankymo datos pakeitimą:</p> <ul style="list-style-type: none"> • Atsakingas darbuotojas turi turėti galimybę peržiūrėti sąrašą pacientų, kurie patvirtino galimybę atvykti kitu laiku, patvirtinti pasirinkto paciento rezervavimą ir jį apie tai informuoti pasirinktu kanalu, pasirinkti vieną iš atlaisvinto laiko rezervavimo būdą: <ul style="list-style-type: none"> ○ <i>Paciente parinkimas iš sąrašo pacientų kurie patvirtino galimybę atvykti kitu laiku;</i> ○ <i>Automatinis rezervavimas pirmam patvirtinusiam atvykimo galimybę;</i> ○ <i>Kiti analizes metu nustatyti būdai.</i>

2.1.2.3. Mokymų modulio funkciniai reikalavimai:

Siekiant užkirsti kelią sunkių létinių ligų vystymuisi, sveikatos priežiūros paslaugos orientuotos tiek į sergančių ligonių gydymą, tiek ir į susirgimų prevenciją. Kauno klinikos aktyviai dalyvauja mokant ir šviečiant pacientus – taiko įvairias ligų ir susirgimų prevencijos priemones (teikia informaciją ir mokomąjā medžiagą apie riziką susirgti létinėmis ligomis, taiko pacientų saugos principus, informuoja apie prevencinių priemonių taikymo galimybes atsižvelgiant į individualius poreikius, šiu priemonių poveikį ir pan.), tokiu būdu užtikrindamos pagrindinių pacientų mokymo bei švietimo tikslų įgyvendinimą. 2.3 pav. pateikiamas mokymų modulio aktorių funkcijų hierarchijos diagrama.



2.3 pav. Mokymų modulio funkcijų hierarchijos diagrama

2.3 lentelė. Pacientų mokymų modulio funkcinį reikalavimų lentelė:

Nr.	Reikalavimas
1.	<p>Pagal naudotojo profili (rizikos grupę, diagnozę, priskirtą gydytoją ir pan.) peržiūrėti personalizuotą aktualių informaciinių ir mokamosios medžiagos paketų sąrašą:</p> <ul style="list-style-type: none"> Pacientas turi turėti galimybę peržiūrėti pilną jam aktualių informaciinių ir mokamosios medžiagos paketų sąrašą, jam suteiktų ir planuojamų paslaugų sąraše turi matyti požymį, kad sistemoje yra užregistruoti su paslaugos suteikimu susijusiu mokymų paketai ir turėti galimybę peržiūrėti šių paketų sąrašą, jam rekomenduojamų mokamosios medžiagos paketų sąraše turi matyti paketo pavadinimą, trumpą turinio aprašymą, paskutinio atnaujinimo datą, paketo peržiūros būseną ir kitus analizes etapo metu identifikuotus duomenis, inicijuoti pasirinkto paketo peržiūrą;
2.	<p>Peržiūrėti mokomąjį medžiagą (dokumentus, vaizdo įrašus, interaktyvias informacines programas):</p> <ul style="list-style-type: none"> Pacientas turi turėti galimybę peržiūrėti mokymų paketo struktūrą: sričių pavadinimus, prikabintus failus, nuorodas į papildomą medžiagą, klausimynus, mokymų paketo sričių tekstą, vaizdinę ir garsinę medžiagą;
3.	<p>Užpildyti savikontroles klausimynus:</p> <ul style="list-style-type: none"> Pacientas turi turėti galimybę užpildyti savikontrolės klausimynus, susijusius su peržiūreta mokymų medžiaga, peržiūrėti savo atliko testo įvertinimus ir peržiūrėti pateiktas rekomendacijas, pakartotiniai užpildyti klausimyną, jeigu tokią galimybę numate mokymo medžiagos rengėjas;
4.	<p>Išsiusti paklausimus dėl papildomos informacijos:</p> <ul style="list-style-type: none"> Pacientas turi turėti galimybę išsiusti paklausimą dėl papildomos informacijos; Pacientas turi turėti galimybę nurodyti paklausimo tipą („dėl papildomos informacijos“, „Pastebėta klaida“). Paklausimų tipų sąrašas turi būti parengtas ir suderintas su Perkančiąją Organizaciją detalaus projektavimo metu;

5.	<p>Pateikti pastabas apie mokemosios medžiagos turinį ir kokybę:</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę įvertinti visą mokymų medžiagos paketą balais, pateikti pastabas dėl paketo medžiagos kokybės rašant laisvos formos tekštą, kiekvieno mokymo paketo peržiūros pabaigoje užpildyti standartizuotą klausimyną apie mokymų naudą ir kokybę;
6.	<p>Gauti tolimesniams susipažinimui rekomenduojamos medžiagos sąrašą:</p> <ul style="list-style-type: none"> • Pacientas turi turėti galimybę peržiūrėti jam, pagal diagnozę, rizikos grupę, suteiktas paslaugas, susipažinimui rekomenduojamos medžiagos sąrašą ir inicijuoti pasirinktos medžiagos peržiūrą;

2.1.3. Nefunkciniai reikalavimai

Kuriamos sistemos moduliams yra keliami šie nefunkciniai reikalavimai [25]:

2.1.3.1. Reikalavimai naudotojo sasajai

Sistemos modulių naudotojo sasają turi tenkinti šiuos nefunkcinius reikalavimus, kurie yra pateikiami 2.4 lentelėje:

2.4 lentelė. Sistemos modulių naudotojo sasajos reikalavimų lentelė:

Nr.	Reikalavimas
1.	Produkto elgsena ir valdymas turi atitiki šiuolaikinius grafinės vartotojo terpės reikalavimus ir įgyvendinti sąlygas, įtakojančias funkcionalumą.
2.	Sistemos portalas grafinė naudotojo sasaja turi būti pritaikyta neįgaliesiems pagal Europos Sajungos „WAI“ (angl. „Web Accessibility Initiative“) gaires („Web Content Accessibility Guidelines 2.0“).
3.	Naudotojo sasaja turi būti realizuota Interneto naršyklės aplinkoje ir turi būti suderinama su „Microsoft Internet Explorer“, „Mozilla Firefox“, „Opera“ ir kitomis lygiavertėmis plačiai paplitusiomis interneto naršyklėmis (tame tarpe ir mobilių įrenginių) bei jų vėliausiomis versijomis.
4.	Naudotojui įvedant duomenis, dalis jų sistemoje turi būti užpildomi automatiškai (pvz.: duomenų pateikimo data, naudotojo duomenys), jei tai bus įmanoma vienareikšmiškai nustatyti.
5.	Sistemos funkcionalumas turi leisti rūšiuoti duomenis pagal pasirinktus parametrus (parametrai turės būti suderinti diegimo projekto analizės ir projektavimo metu). Duomenys, kuriuos sudaro lietuviški rašmenys, turi būti rūšiuojami pagal lietuvišką abécéle.
6.	Sistemos funkcionalumas turi leisti atlkti įrašų paiešką pagal pasirinktus parametrus.
7.	Meniu nuorodų pavadinimai ir grupavimas turi būti aiškūs. Turi būti aišku, kokiame puslapyje ir kuriame detalizacijos lygmenyje naudotojas yra. Mygtukai, nuorodos ir tekstai turi skirtis ir būti aiškiai atpažistami.
8.	<p>Turi būti realizuotas kitas naudojimo patogumą užtikrinantis funkcionalumas:</p> <ul style="list-style-type: none"> • Langų/objektų išdestymas turi atitiki naudotojų veiklos seką; • Realizuota TAB klavišo seka einant per įvedimo laukus; • Turi būti naudojamas užuominos (žymeklis, „ToolTips“, mygtukų būsenos „onmouseover“ ir pan.);
9.	Sistema turi išlaikyti vientisą vartotojo sasajos stilių ir elementų išsidėstymą per visus kuriamus modulių puslapius. Dizaino grafikos elementai turėtų būti imami ir naudojami iš jau sistemoje sukurtų komponentų sąrašo, o prireikus naujo dizaino sprendimo reikėtų pasitarti su projekto dizaineriu.

2.1.3.2. Reikalavimai sistemos plečiamumui

Sistemos modulių plečiamumas turi tenkinti šiuos nefunkcinius reikalavimus, kurie yra pateikiami 2.5 lentelėje:

2.5 lentelė. Sistemos modulių plečiamumų reikalavimų lentelė:

Nr.	Reikalavimas
1.	Sistemos programinė įranga negali būti ribojantis veiksny, didinant sistemos našumą. Turi būti užtikrintas sistemos našumo padidinimas pridedant papildomus techninius išteklius, nekeičiant programinės įrangos išeities tekštų. Techninės įrangos pajėgumų didinimas turi būti atliekamas nestabdant, kiek tai įmanoma, sistemos darbo.
2.	Prie sistemos pridėjus papildomą funkcinį komponentą neturi reikėti perprogramuoti visos sistemos. Sistema turi būti suprojektuota ir realizuota taip, kad sistema būtų lanksti modifikuojant – realizavus funkcionalumo pakeitimus vienoje ar keliose funkcinėse srityse, pakeitimai neturi būti visos sistemos perkūrimo priežastimi.
3.	Informacinių sistemos programinės įrangos išorinių naudotojų skaičius ir apdorojamos informacijos apimtys neturi būti ribojamas licencijomis.

2.1.3.3. Reikalavimai našumui ir greitaveikai

Sistemos modulių našumas ir greitaveika turi tenkinti šiuos nefunkcinius reikalavimus, kurie yra pateikiami 2.6 lentelėje:

2.6 lentelė. Sistemos modulių našumo ir greitaveikos reikalavimų lentelė:

Nr.	Reikalavimas
1.	Atsako sparta, atmetus duomenų užlaikymą duomenų perdavimo linijose, turi būti (ne mažiau kaip 90% veikimo atvejų): <ul style="list-style-type: none">• Sistemos naudotojų veiksmų vidutinė atlikimo trukmė būtų ne ilgesnė nei 3 sekundės;• Vidiniame tinkle navigacija tarp skirtingų naudotojo sąsajos langų turi trukti ne ilgiau kaip 2 sekundės (išskyrus atvejus, kai generuojama ataskaita);• Sistemos duomenų mainuose dalyvaujančių žiniatinklio paslaugų atsakymai turi būti pateikiami per ne ilgiau kaip 2 sekundes;• Skaitmeninio turinio paieška turi trukti ne ilgiau kaip 2 sekundes;

2.1.3.4. Reikalavimai sistemos prieinamumui ir patikimumui

Sistemos modulių prieinamumas ir patikumas turi tenkinti šiuos nefunkcinius reikalavimus, kurie yra pateikiami 2.7 lentelėje:

2.7 lentelė. Sistemos modulių prieinamumo ir patikumo reikalavimų lentelė:

Nr.	Reikalavimas
1.	Sistema turi būti technologiškai funkcionali pagal principą „24 valandos per dieną, 7 dienos per savaitę, 365 dienos per metus“. Iš viso per metus sistema dėl gedimų ar profilaktikos gali neveikti ne daugiau 80 valandų. Turi būti užtikrintas informacinių sistemos prieinamumas ne mažiau kaip 96 proc. laiko visą parą, ir 99% prieinamumą dienos metu nuo 8:00 iki 20:00.

<p>2. Sistema turi užtikrinti korekтиšką avarinių situacijų, kurias sukėlė neteisingi sistemos naudotojo veiksmai, neteisingas įvedamų duomenų formatas arba neleidžiamos įvedamų duomenų reikšmės, valdymą. Šiai nurodytais atvejais, sistema turi rodyti atitinkamus avarinius pranešimus ir po to grįžti į darbinę būklę. Sistemos architektūriniai komponentai turi būti placiai naudojami praktikoje ir stabilūs. Diegėjo siūloma sistemos programinė architektūra turi užtikrinti pakeičiamumo principą, t.y. įvykus vieno ar kelių komponentų gedimams, sistema turi tapti darbą su esamais ištekliais.</p>

2.1.3.5. Reikalavimai saugumui

Sistemos modulių saugumas turi tenkinti šiuos nefunkcinius reikalavimus, kurie yra pateikiami 2.8 lentelėje:

2.8 lentelė. Sistemos modulių saugumo reikalavimų lentelė:

Nr.	Reikalavimas
1.	Sistemos posistemės turi būti prieinamos naudojantis vieningomis saugos komponentėmis teikiamomis saugos priemonėmis, vieningo prisijungimo (angl. Single Sign On - SSO) principu.
2.	Sistema naudotojams turi pateikti tik tas duomenų tvarkymo priemones, kurias naudotojas turi teisę naudoti.
3.	Naudotojui turi būti leidžiama keisti tik tuos įrašus, kuriuos jis turi teisę keisti.
4.	Naudotojui pateikdama užklausų rezultatus, Sistema turi rodyti tik tuos įrašus, kuriuos naudotojas turi teisę peržiūreti. Sistema turi būti apsaugota nuo: <ul style="list-style-type: none"> • neautentikuotos prieigos; • nesankcionuoto naudotojo sesijos peremimo; • nesankcionuoto duomenų peremimo ar jų įterpimo; • žalingo kodo įterpimo (angl. Injection, XSS (Cross-sitescripting)); kitų saugumo pažeidimų, kurie įvardijami OWASP TOP 10 (https://www.owasp.org) sąraše (arba lygiaverčiame);
5.	Sistema turi veikti saugiai ir patikimai, t. y. nekelti grėsmės kitiems sistemos moduliams, duomenims, aparatinėi įrangai, vartotojo sveikatai, netrukdyti kitų sistemų (integraciinių) darbui; iškilusios išimtinės situacijos turi būti tinkamai apdorotos programos viduje, apie iškilusias problemas informuojant vartotoją.

2.1.4. Techninė specifikacija

Kuriamų sistemos modulių techninės specifikacijos reikalavimai yra pateikiami 2.9 lentelėje:

2.9 lentelė. Techninės specifikacijos reikalavimų lentelė:

Nr.	Reikalavimas
1.	Sistemos architektūra turi būti realizuota ne mažiau kaip pagal trijų sluoksnių (angl. three-tier, 3-tier) programų architektūros modelį, išskiriant: <ul style="list-style-type: none"> • Vaizdavimo lygmenį; • Veiklos logikos lygmenį; • Duomenų bazes lygmenį;
2.	Sistema turi būti realizuota remiantis „SOA“ (angl. „Service–Oriented Architecutre“) principais, išlaikant kuo didesnę ją sudarančių komponentų tarpusavio nepriklausomybę.
3.	Sistema turi palaikyti ir būti suderinama su „XML“ (angl. „eXtensible Markup Language“) ir „XML“ žiniatinklio paslaugomis.
4.	Sistemos duomenys turi būti saugomi reliacinėje duomenų bazių valdymo sistemoje (DBVS).

5.	Visos sistemos dalys turi būti tarpusavyje integruotos. Visi informacijos pasikeitimai vienoje dalyje turi atspindėti susijusiose dalyse be papildomų sistemos naudotojų veiksmų.
6.	Sistema turi veikti užsakovo techninėje infrastruktūroje. Jeigu paslaugų teikėjas numatys keisti tarnybinių stočių platformą, turi būti naudojamos labiausiai paplitusios „Windows“, „UNIX“ arba „Linux“ operacinės sistemos.
7.	Sistema turi būti suderinama su užsakovo esama infrastruktūra, t.y. turi būti užtikrintas užsakovo turimos infrastruktūros panaudojimas ir sederinamumas (DBVS, kitos IS, darbo vietų operacines sistemos ir tt.). Visos darbo vietas naudoja „Windows“ pagrindu veikiančias operacines sistemas.
8.	Sistemos duomenų mainų priemones turi būti pagrįstos: „XML“, „HTTP(S)“, „SOAP“, „WSDL“, „WS-*“ (žr. kitą reikalavimą) standartais ir turi būti sederintos su „ESPBI IS“ duomenų mainams keliamais reikalavimais.
9.	Duomenų mainų saugos, patikimumo ir sąveikumo technologiniame lygmenyje užtikrinimui, realizuojamos žiniatinklio paslaugos turi naudoti „WS-*“ standartų grupes protokonus: „WS-Security“, „WS-Secure Conversation“, „WS-SecurityPolicy“, „WS-MetadataExchange“, „WS-Trust“, „WS-AtomicTransaction“, „WS-ReliableMessaging“, „WS-I(nteroperability)“.
10.	Visi sistemos funkciniai komponentai turi palaikyti „Unicode“ (UTF – 8) standartą.

2.2. Sistemos modulių projektas

2.2.1. Sistemos loginė architektūra

Sistemos paslaugų modulių architektūroje bus taikomas bazinis „MVC“ architektūros modelis, pritaikant keliis papildomus architektūros sluoksnius: „ViewModel“ ir „Service“ [19]. Šio modelio veikimo principas yra paremtas „Separation of Concerns“ (SoC) principu, tai reiškia, kad programa yra išskaidoma į tokias dalis, kad jų funkcionalumas ir poreikiai nepersidengtū – izoliuojama sistemos logika nuo vartotojo sąsajos. Tai leidžia rašyti geriau organizuotą, tvarkingesnį ir lengviau prižiūrimą kodą, galima lengviau palaikyti sistemą ir ją tobulinti. Objektiškai orientuotame programavime šis principas atispindi išskaidant užduočių atlikimą į objektus ir jų metodus. Tai plačiai taikomas, gerai įvertintas ir laiko išbandytas sprendimas, kuris yra ganétinai lankstus, leidžiantis pritaikyti savus norimus įgyvendinti sprendimus. Naudojamas „Visual Studio“ programinis įrankis „MVC“ architektūrą siūlo kaip numatytają.

Privalumai. Leidžia keisti duomenis nepriklausomai nuo jų vaizdavimo būdo ir atvirkšciai. Palaiko tų pačių duomenų vaizdavimą skirtingais būdais keičiant duomenis tik vienoje vietoje [20].

Trūkumai. Gali reikėti sudėtingo programos kodo net ir tada, kai duomenų modelis bei sąveika tarp duomenų yra labai paprasti.

Aptarsime kiekvieną numatomą architektūrinį komponentą:

„Model“ – tai tam tikros srities duomenų valdymo/pateikimo vienetas, kurį nusako įvairūs programavimo kalbos sukurti kintamieji, objektai, jų sąrašai, masyvai ir kitos imtys, atspindintys norimą atvaizduoti vartotojui informaciją ar realaus pasaulio objektą/esybę.

„ViewModel“ – tarpinis sluoksnis tarp modelio ir vaizdo sluoksnio, kuris turi apibrėžtą konkrečią aibę modelio parametrų ir tik būtent jie yra atvaizduojami vaizdo sluoksnuje, norint atvaizduoti tik tai reikiamus tam vaizdui duomenis (išvengiama nereikalingų duomenų pateikimo).

„View“ – šis sluoksnis suteikia aprašytam modeliu išvaizdą, kuri yra tinkama suprasti vartotojui (formos, įvairūs sąrašai). Pateikiamas vaizdas yra sugeneruotas HTML/CSS/Javascript pavidalu.

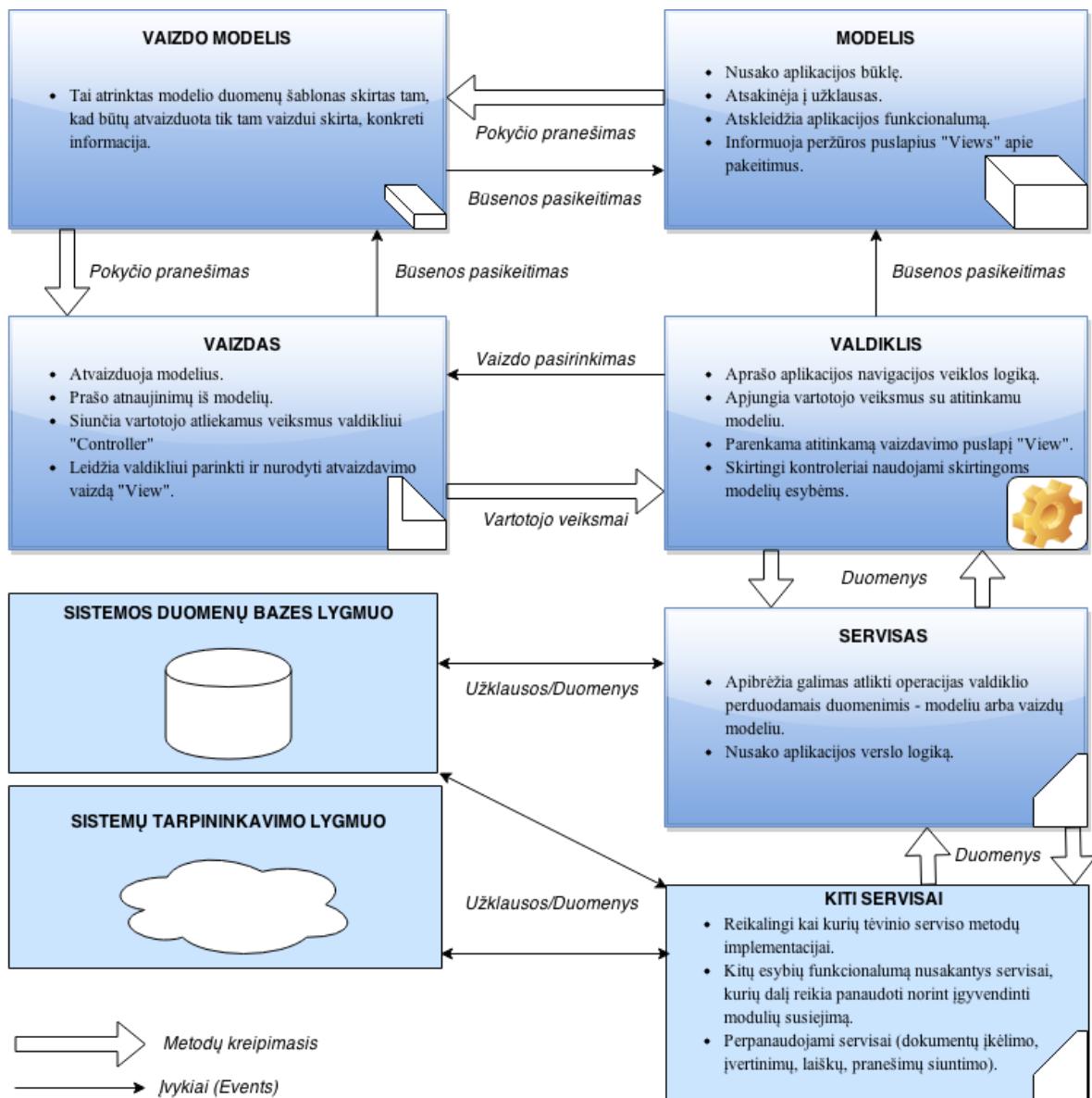
„Controller“ – šis sluoksnis yra atsakingas už modelių valdymą ir vaizdų pateikimą atsižvelgiant į atitinkamus vartotojo veiksmus. Kontroleriuose bus valdomi su sistemos navigacija susiję sprendimai, bus bendradarbiaujama tarp servisių (angl. „Service“) ir vaizdų (angl. „View“) sluoksnii.

„Service“ – šis sluoksnis yra susietas su „Controller“ sluoksniu, tačiau jি išskiriant jis palengvins ir atskirs dalį darbo ir veikos logikos nuo kontrolerio sluoksnio. Jo paskirtis atributi duomenų gavimą, atfiltravimą ir apiforminimą nuo kontrolerio ir jam palikti atligli tik tai norimos logikos vaidmenį su iš serviso gaunama duomenų imtimi (modeliu).

Sistemos duomenų bazės lygmuo (angl. „Data access layer“) – apibrėžia sistemoje naudojamų duomenų prieinamumą prie duomenų bazės (paémimą, atnaujinimą, ištrynimą), naudojant „Entity Framework“ technologiją.

Sistemos tarpininkavimo lygmuo (angl. „Service layer“) – apibrėžia sistemos bendravimo logiką su kitomis sistemomis „WCF“ technologija – sutartis, duomenų išsiuntimo, gavimo elgseną, kontraktus, saugumą.

Sistemos architektūrinio modelio schema yra pavaizduota 2.4 pav.



2.4 pav. Kuriamos sistemos architektūrinis modelis

2.2.2. Sistemos modulių veiklos logika

Į sistemos paslaugų modulių specifikaciją taip pat įeina jų elgsenos aprašymai. Elgsena aprašoma UML veiklos diagramomis.

2.2.2.1. Nuotolinių stebėjimų modulio veiklos logika:

Pacientas, prisijungęs prie portalo, nuotolinių stebėjimų modulyje, jam suteiktos sąsajos pagalba gali priimti arba susikurti atitinkamas paslaugas - nuotolinį stebėjimą, prisiskirti įrenginių. Atlikto veiksmo informacija yra siunciama įrenginių administratoriui. Prisiskyrus nuotolinį stebėjimą arba jo metu naudojamą įrenginį, yra suteikiama galimybė susipažinti su įrenginio naudojimo medžiaga - nuotolinio stebėjimo duomenimis, duomenų teikimo grafiku. Pateikinėjant stebėjimui duomenis, priklausomai nuo to, ar pateikimas vėluoja ar ne, sistemoje yra atliekami skirtini skirtini veiksmai. Nepateikus laiku duomenų, apie tai yra informuojamas įrenginių administratorius, nuotoliniui stebėjimui priskirtas gydytojas. Pateikinėjant duomenis yra pateikiamas pildytu klausimynas. Klausimyną sudaro įvairūs anketos ir sveikatos rodiklių duomenys. Nuotolinio stebėjimo eigoje yra gaunamos įvairios rekomendacijos apie vykdomą stebėjimą ir gaunamus analizės rezultatus.

Įrenginių administratorius prisijungęs prie portalo, jam suteiktos sąsajos pagalba gali priskirti atitinkamus įrenginius pacientams, fiksuoti įvairius veiksmus - išduoti, grąžinti, remontuoti, peržiūrėti, kitaip tvarkyti visą su įrenginiais susijusią informaciją.

Atsakingas gydytojas tvarko jam priskirto paciento nuotolinio stebėjimo nustatymus, tačiau paciento pateikiamą duomenų keisti negali, peržiūri paciento pateikiamą informaciją, ją analizuoją, įvertina, priima atitinkamą sprendimą ar yra reikalingi atlikti papildomus veiksmus (rekomenduoti, perspėti pacientą).

Sistema atlieka informacijos apdorojimo, įvertinimo ir pateikimo vaidmenį, kokybiškai greitai ir tiksliai pateikiant reikiama informaciją atitinkamiams portalo naudotojams ir jų rolių grupėms.

Nuotolinių stebėjimų modulio veiklos diagrama yra pavaizduota 2.5 pav.

2.2.2.2. Apsilankymų modulio veiklos logika:

Pacientas, prisijungęs prie portalo, jam suteiktos sąsajos pagalba gali peržiūrėti suplanuotų apsilankymų grafiką arba jau turimos paslaugos (apsilankymo) būseną. Ši informacija yra gaunama iš kitos bendradarbiaujančios HIS informacinės sistemos. Peržiūrint apsilankymų grafiką, galimi atlikti keli skirtini veiksmai su apsilankymais, priklausomai nuo paciento noro atlikti konkretų veiksmų: nekeisti registracijos laiko (tiesiog peržiūrėti turimą užregistruotą apsilankymą), negalėti atvykti sutartu laiku (inicijuoti apsilankymo datos pakeitimą, sistemai įvertinant tos datos pakeitimo galimybes), galėti atvykti kitu laiku (atlaisvinti turimą apsilankymą kitiems sistemos vartotojams bei užsirezervuoti laiką kitam apsilankymui).

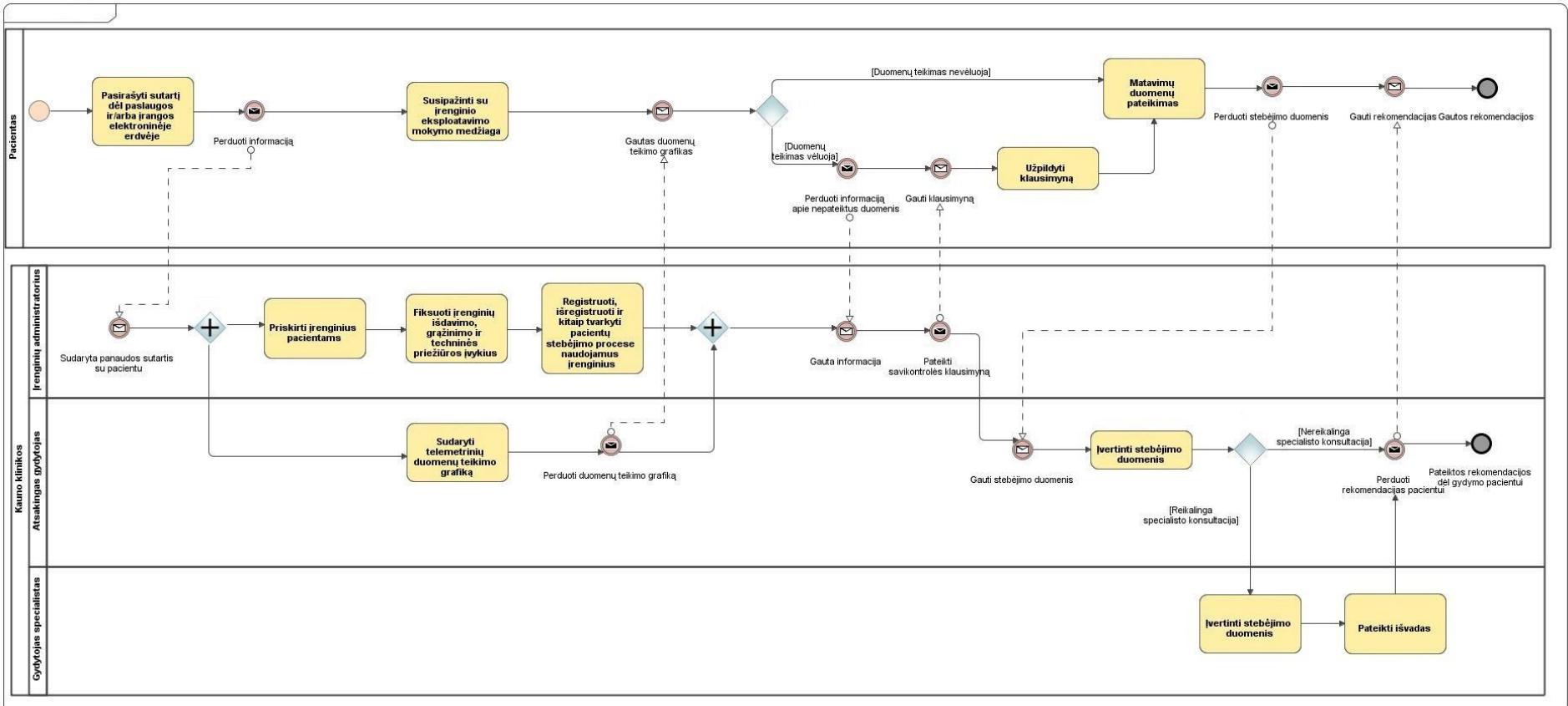
Klinikų atsakingas darbuotojas prisijungęs prie portalo, jam suteiktos sąsajos pagalbą gali peržiūrėti sąrašą pacientų, kurie negali atvykti ar nori pakeisti apsilankymo datą. Informacija gaunama sistemos pagalba. Darbuotojas (taip pat ir pati sistema) gali bet kada inicijuoti atitinkamus pacientus apie vizito datos pakeitimo galimybes (galioja ir atšaukimas), rekomenduoti kitus apsilankymus.

Sistema atlieka informacijos apdorojimo ir įvertinimo vaidmenį ar atitinkamais atvejais yra galima atlikti vizito datos pakeitimo, atšaukimo veiksmus, ar atitinkami vartotojai nori atlikti veiksmus ir gauti informaciją apie tai. Taip pat yra įvertinamos laisvų laikų ir vietų galimybės, atlaisvinami ir vėl užrezervuojami apsilankymai. Apsilankymų modulio veiklos diagrama yra pavaizduota 2.6 paveiksle.

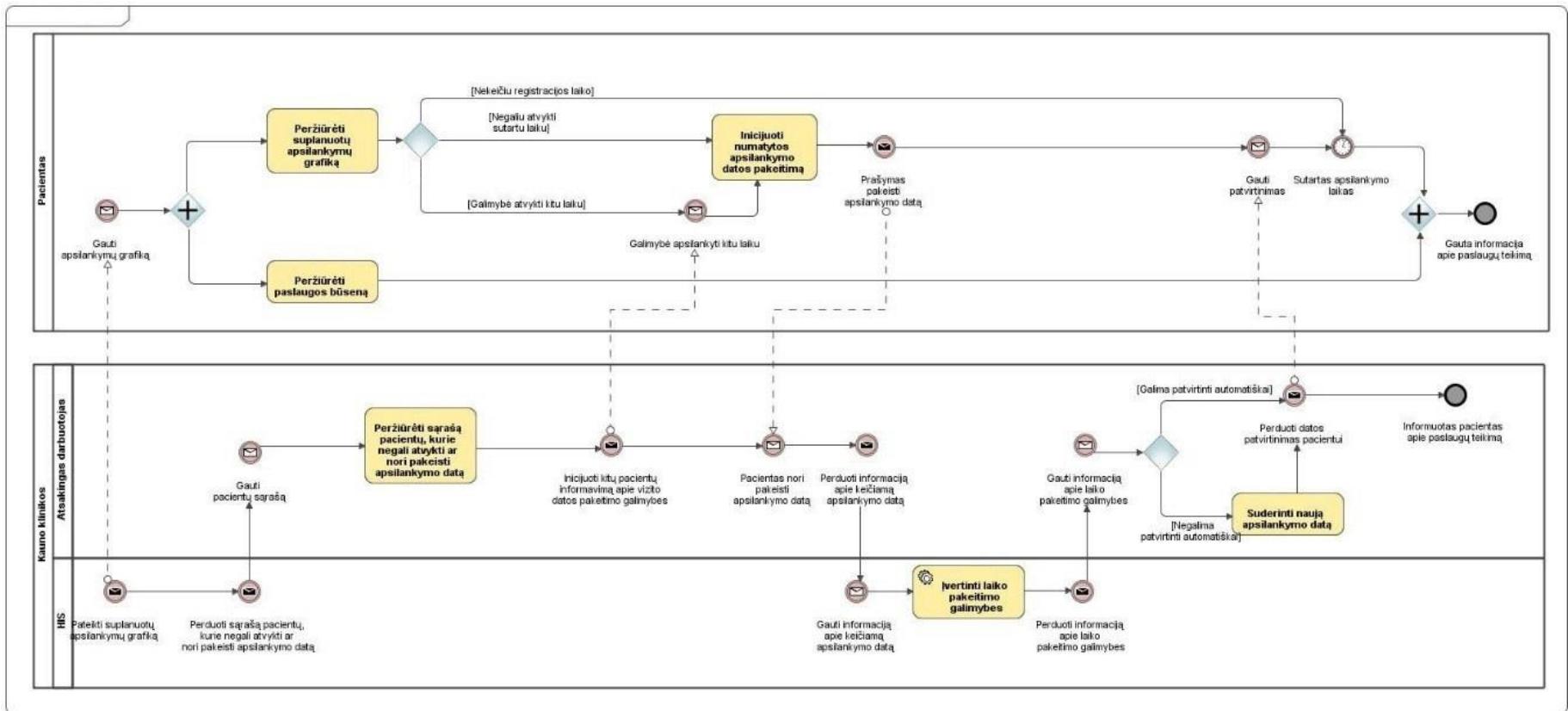
2.2.2.3. Mokymų modulio veiklos logika:

Vartotojas, prisijungęs prie portalo, jam suteiktos sąsajos pagalba peržiūri bendrą mokomąjį medžiagą, užpildo savikontrolės klausimynus. Priklausomai nuo sistemos priskiriamo ar nepriskiriamo diagnostės kodo sistema atlieka atitinkamus veiksmus ir skaičiavimus norimai informacijai gauti. Priskiriant diagnostės kodą, jis yra gaunamas iš kitos bendradarbiaujančios HIS informacinės sistemos. Tuomet vartotojas pagal jam nustatyta diagnostę gali peržiūrėti savo personalizuotą mokymų informaciją, gauti rekomenduojamos peržiūrėti medžiagos sąrašą. Peržiūrint personalizuotą informaciją vartotojas gali pateikti pastabas apie jos turinį ir kokybę sistemos mokymų administratoriui. Pastarasis atsižvelgdamas į pastabas gali tobulinti ir keisti mokymo medžiagą atitinkamiems diagnostės tipams. Nepriskyrus diagnostės kodo arba jo neįdentifikavus, vartotojui sistema pateikia bendrą informaciją.

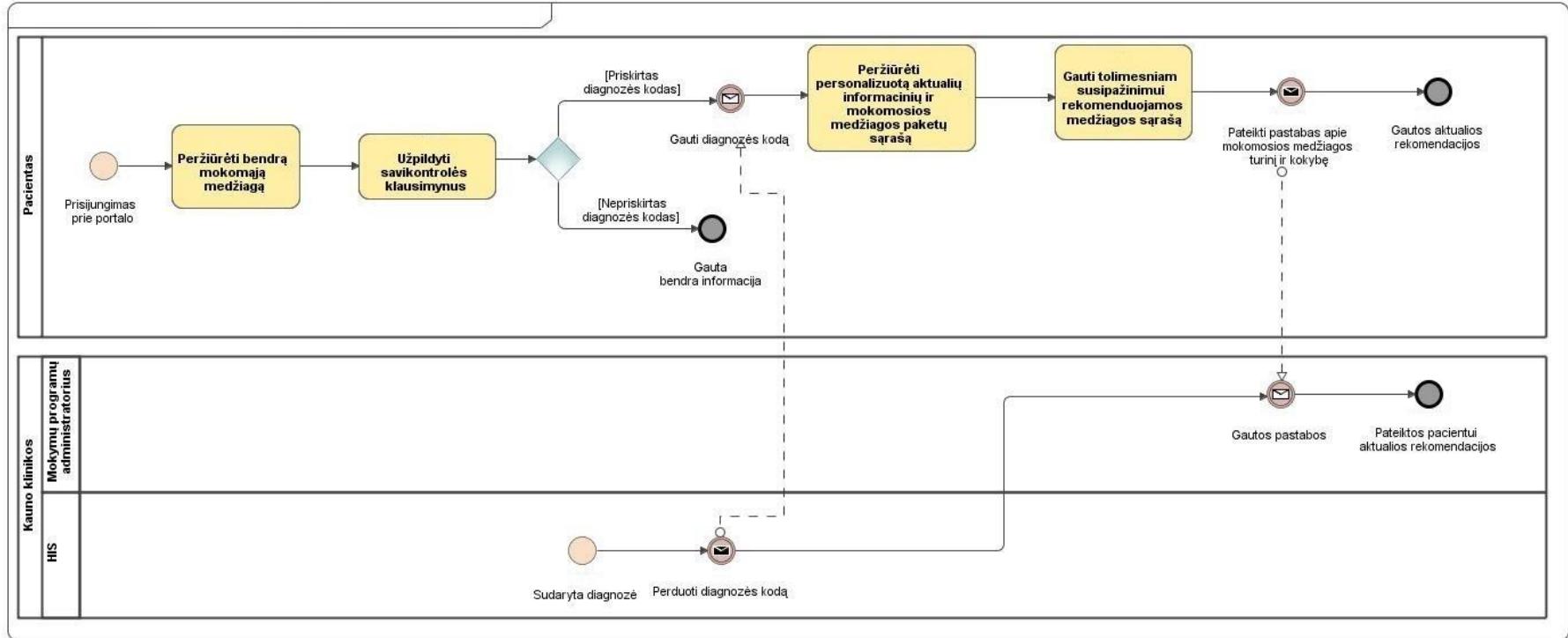
Mokymų modulio veiklos diagrama yra pavaizduota 2.7 pav.



2.5 pav. Sistemos nuotolinių stebėjimų modulio UML veiklos diagrama



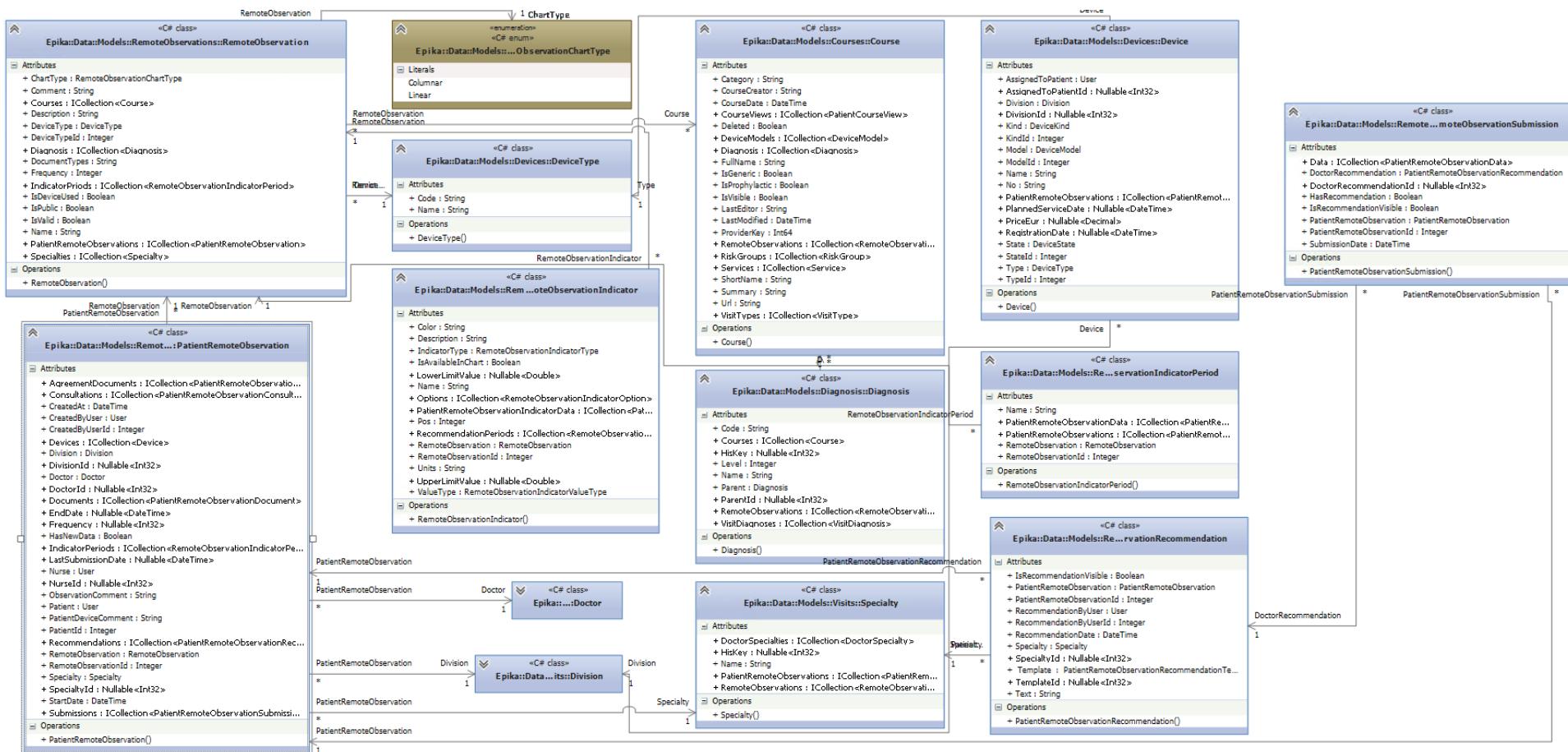
2.6 pav. Sistemos apsilankymų modulio UML veiklos diagrama



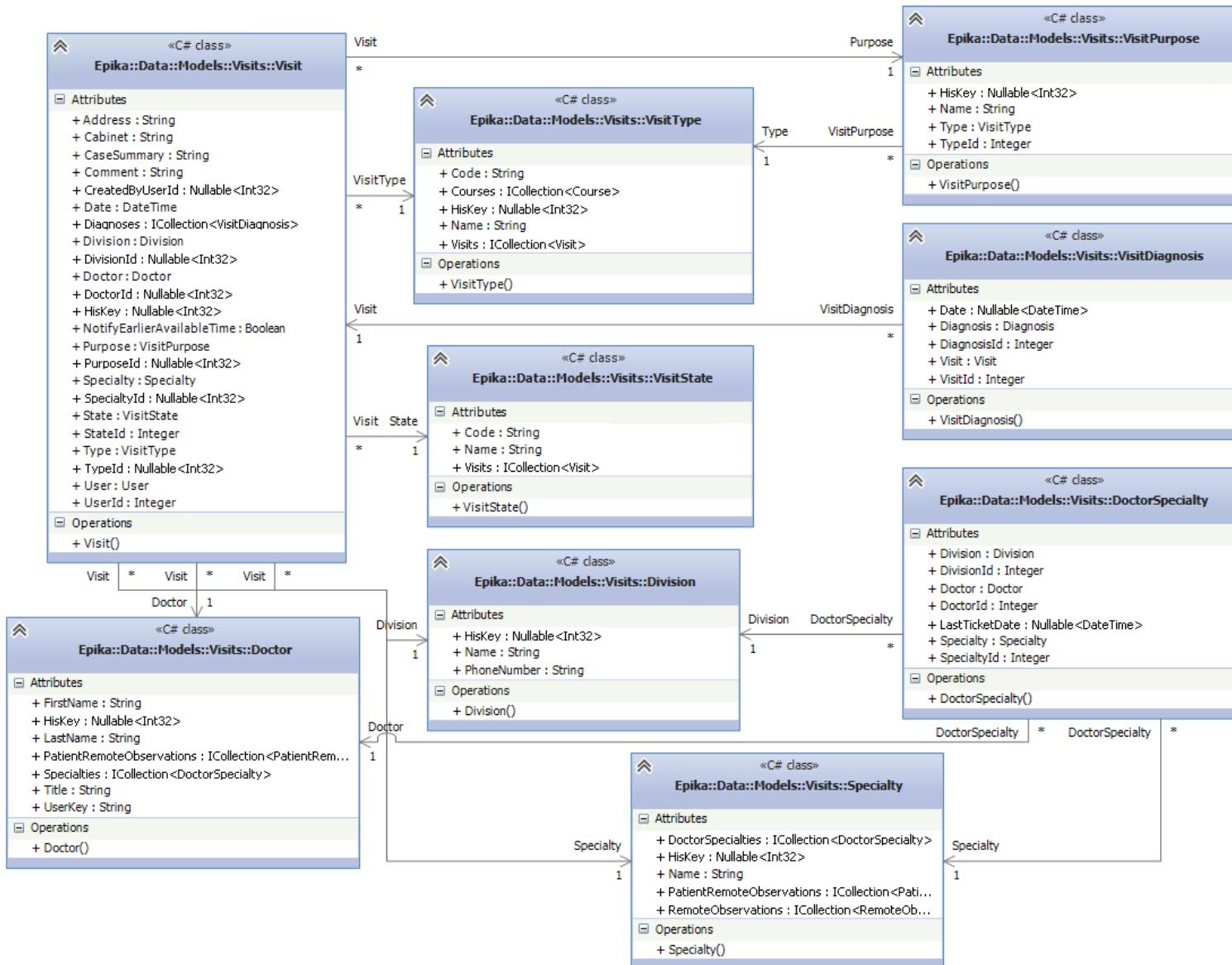
2.7 pav. Sistemos mokymų modulio UML veiklos diagramma

2.2.3. Sistemos modulių dalykinės srities esybių klasės

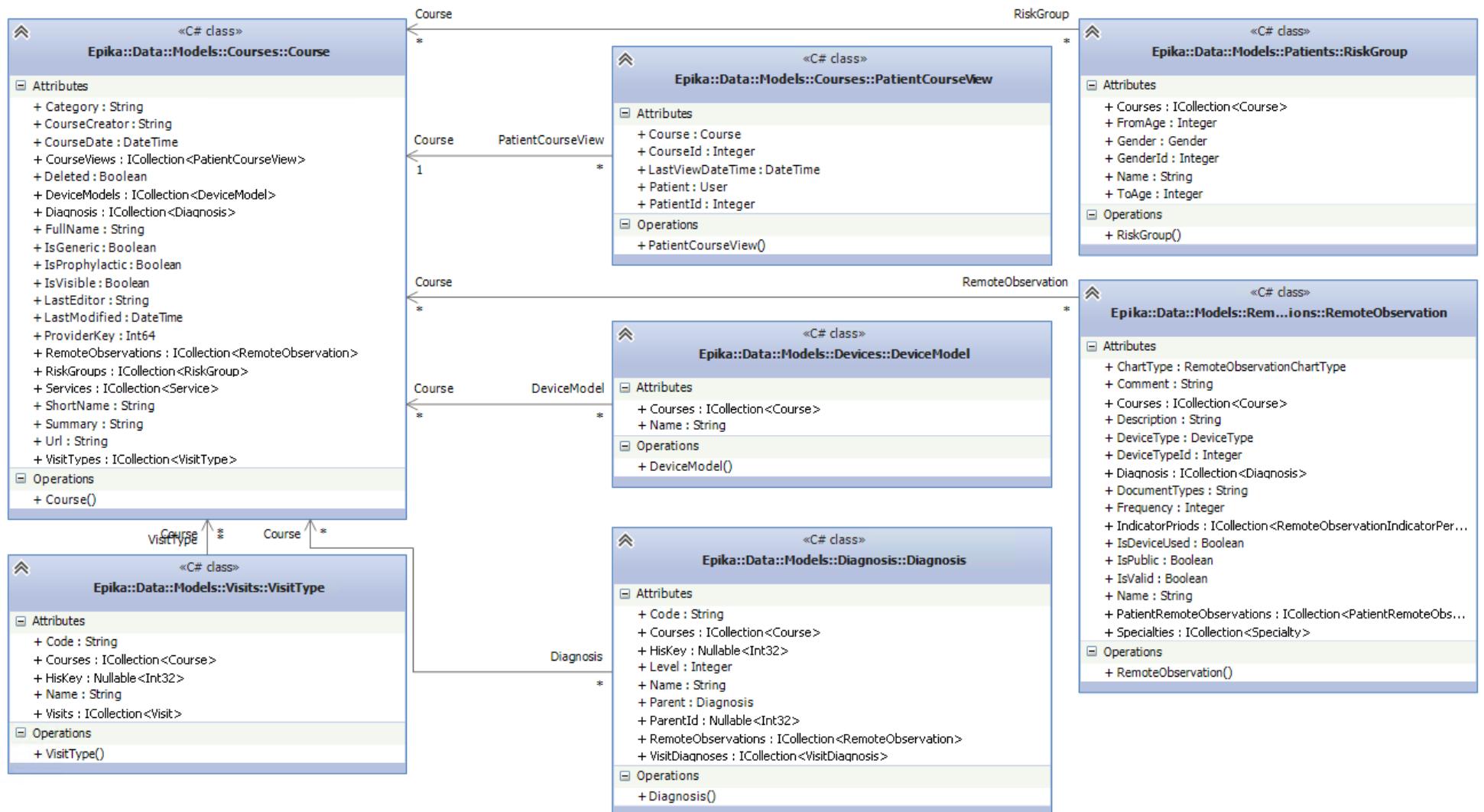
Sistemos modulių modelių ryšiai tarp jų atvaizduoja numatomos klasų diagrammos, kurios yra pateikiamos 2.8, 2.9 ir 2.10 paveiksluose. Jose yra pateikiami galimi numatomi modulių klasų aprašai, atributai, ryšiai. Šiu parametru pavadinimai buvo verčiami į anglų kalbą iš medicininėse išstaigose naudojamų techninės kalbos žodžių. Jos salyginai atspindi duomenų bazėje naudojamas lenteles. Kadangi klasų diagramma yra per didelę tam, kad būtų kokybiškai atvaizduota viename paveiksliuke, ji suskaidyta į atskiras dalis pagal modulius.



2.8 pav. Sistemos nuotolinio stebėjimų modulio esybių diagramma



2.9 pav. Sistemos apsilankymų modulio esybių diagramma



2.10 pav. Sistemos mokymų modulio esybių diagrama

2.2.4. Sistemos modulių vartotojo sąsajos modeliai

Pateiksime sistemos modulių vartotojo sąsajos navigavimo planą taip pat pavaizduosime numatomus vartotojo sąsajos eskizus (maketus), rezultatų formas.

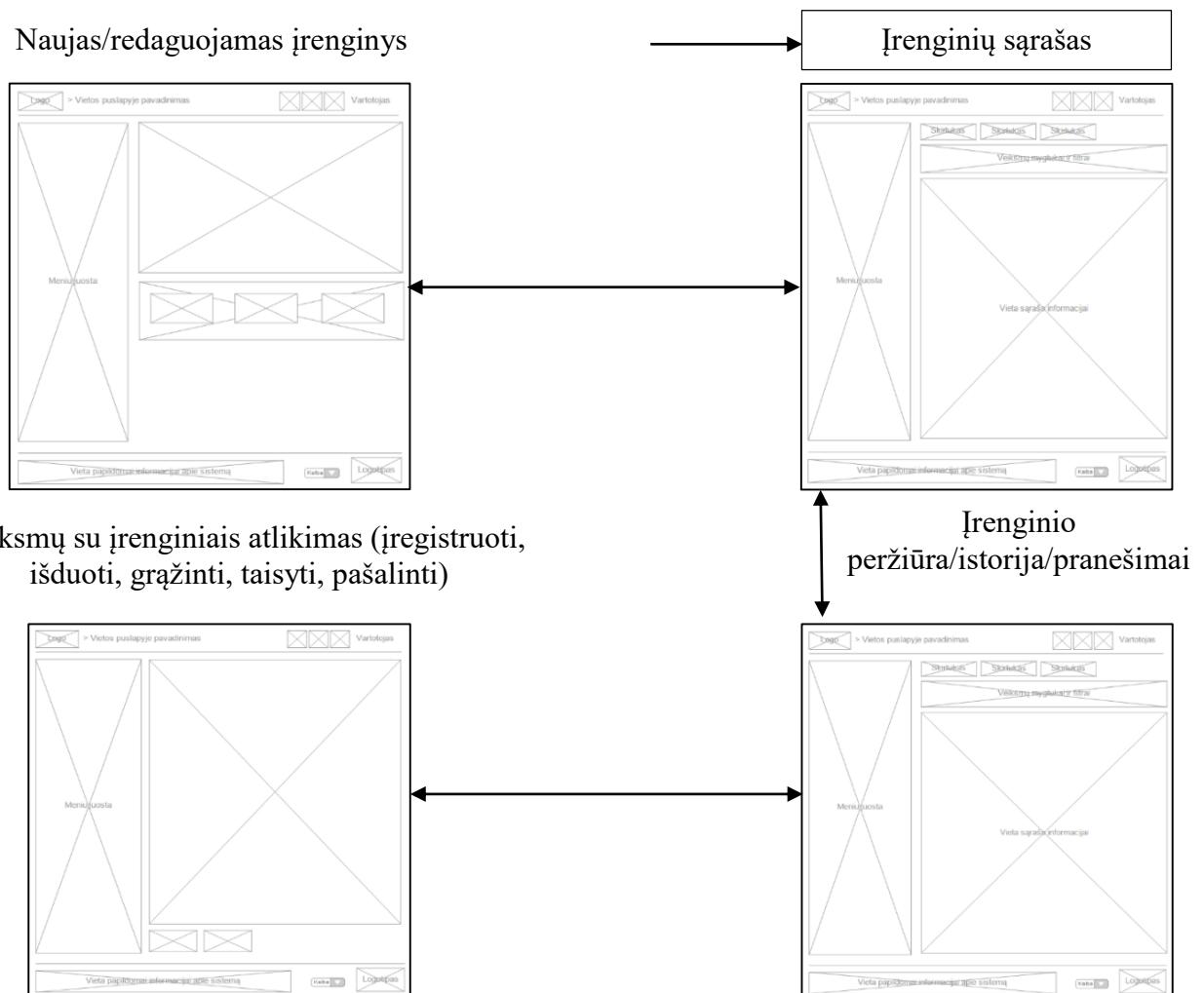
Navigavimo planas atspindi kaip langai yra susieti su kuriamos informacinės sistemos moduliais. Rodyklės tarp vartotojo sąsajos langų vaizduoja sistemos modulių langų navigaciją vienas tarp kito. Mygtukų bei nuorodų pagalba galima pasiekti sekančius langus ir grįžti atgal. Naudojami įvairūs simboliai ir geometrinės formos padeda atvaizduoti numatomus įvedamą arba atvaizduojamą duomenų formų laukus, mygtukų pozicijas, galimas atvaizduoti informacijos vietas ir padėtis. Stengiamasi išlaikyti vieningą šablono stilių, pakeičiant tik konkrečiam funkcionalumui įgyvendinti reikalingą atvaizdavimą.

Visuose moduliuose navigacijos planas prasideda iš jų sąrašo puslapių. Virš maketo vaizduojamas tekstinis laukas su pajuodintu kontūru aplink ji. I stebėjimo sąrašus galima patekti visuose eskizuose matomoje meniu juosteje, esančioje kairėje sistemos puslapių pusėje.

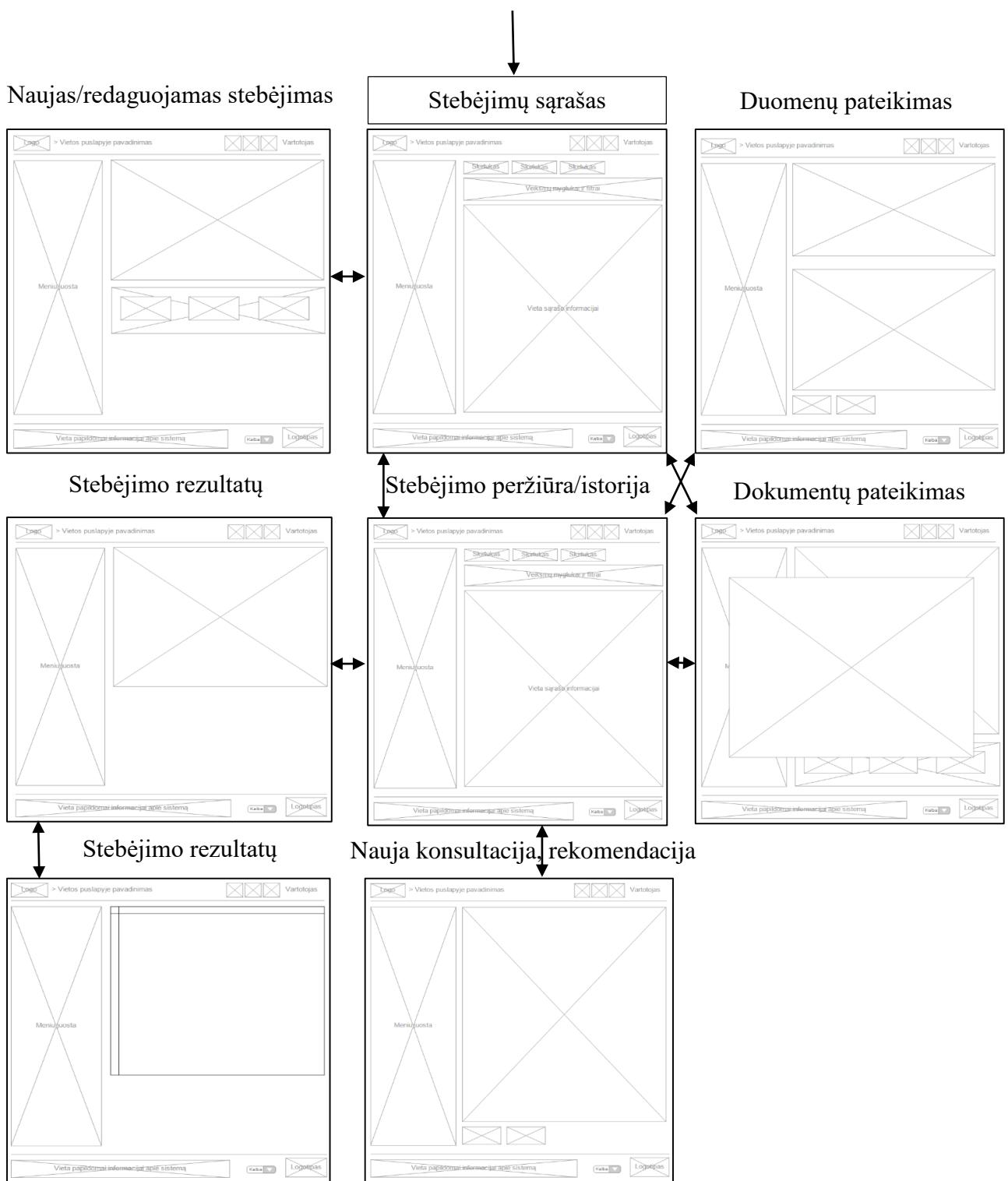
Dėl mūsų projekte apsibrėžtų tikslų ir uždavinijų, kad kurdami sistemą prisdėsime prie nuotolinių stebėjimų, apsilankymų ir mokymų modulių įgyvendinimo, šių modulių navigacijos planų atvaizdavimą parodysime tik iš šių modulių vidaus, t.y. nepateiksime jų sąveikos su kitais sistemoje kuriamais moduliais.

2.11, 2.12, 2.13 ir 2.14 paveiksluose yra pateikiami sistemos modulių vartotojo sąsajų modeliai.

2.2.4.1. Nuotolinių stebėjimų modulio vartotojo sąsajos modelis

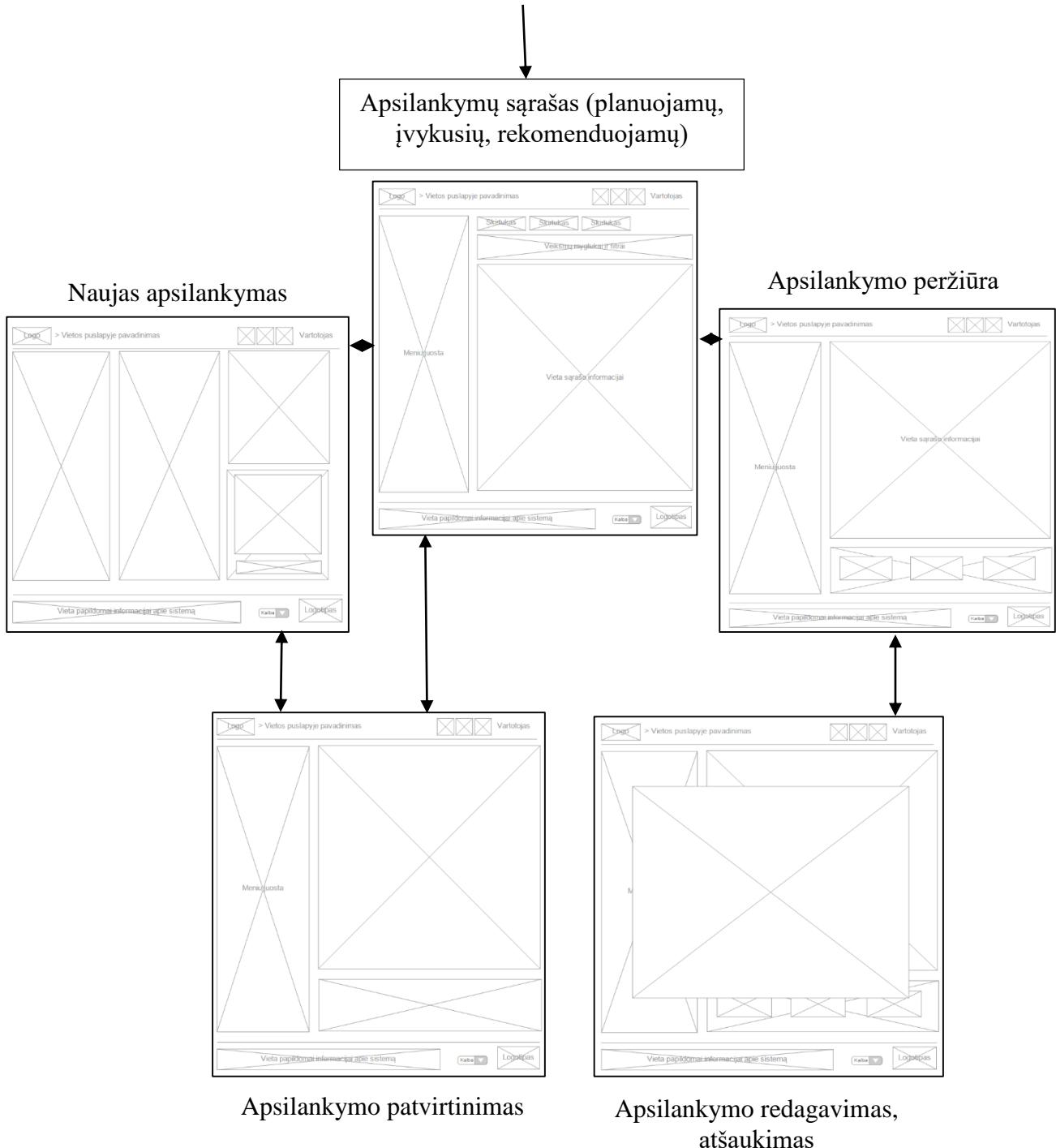


2.11 pav. Nuotolinių stebėjimų įrenginių navigacijos planas ir vartotojo sąsajos eskizai



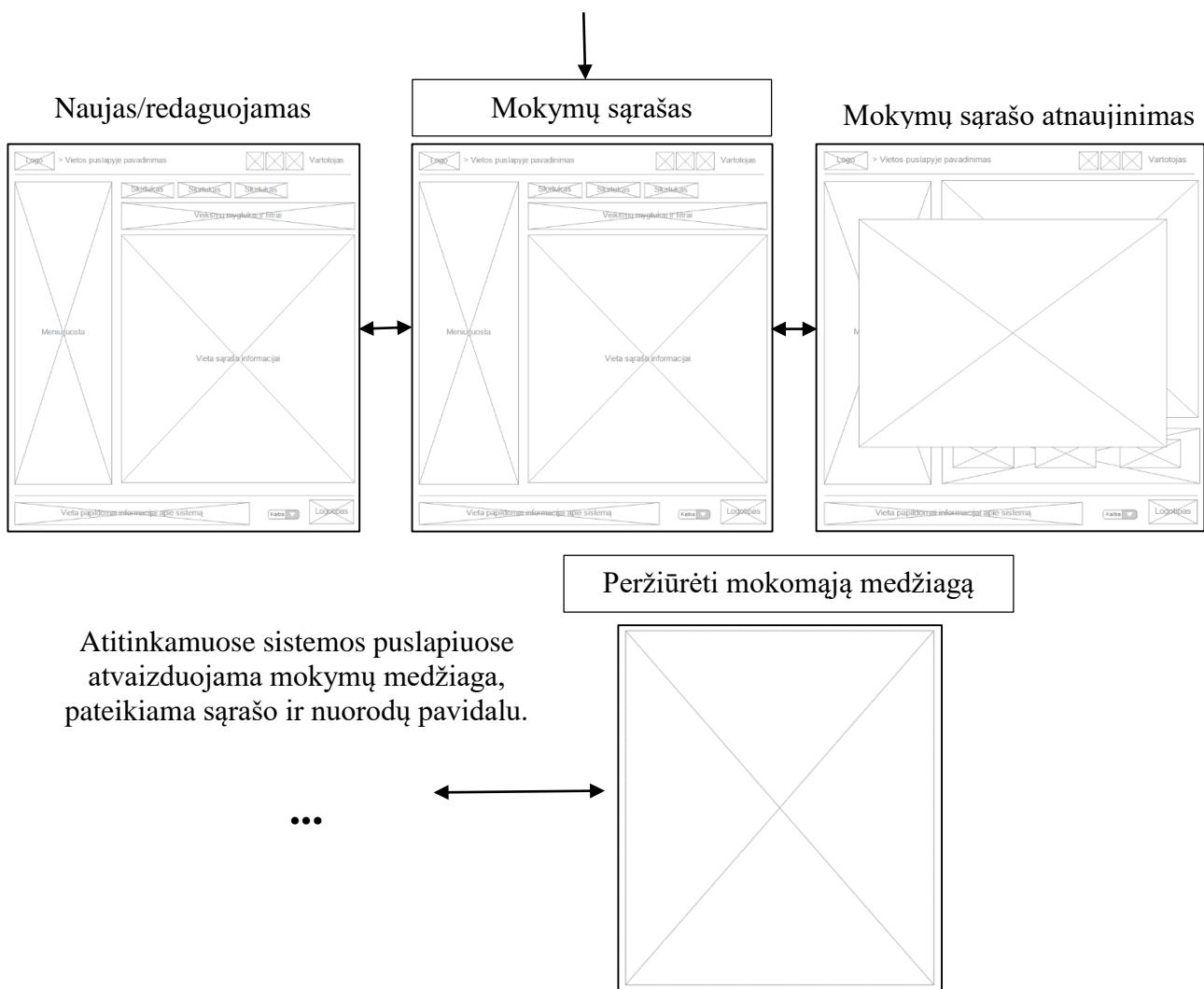
2.12 pav. Nuotolinių stebėjimų modulio navigacijos planas ir vartotojo sąsajos eskizai

2.2.4.2. Apsilankymų modulio vartotojo sasajos modelis



2.13 pav. Apsilankymų modulio navigacijos planas ir vartotojo sasajos eskizai

2.2.4.3. Mokymų modulio vartotojo sąsajos modelis



2.14 pav. Mokymų modulio navigacijos planas ir vartotojo sasajos eskizai

2.3. Projektinės dalies išvados

Sistemos kuriamuose moduliuose numatomi aštuoni aktoriai: pacientas, įrenginių administratorius, atsakingas gydytojas, atsakingas specialistas, KK atsakingas darbuotojas, mokymų administratorius, serveris, bendradarbiaujanti sistema (HIS), kurių kiekvienas gali atlkti tik jam būdingus veiksmus. Detalizuojant sistemos modulių atliekamus veiksmus išskirti funkciniai reikalavimai, taip pat aprašyti nefunkciniai reikalavimai naudotojo sąsajai, sistemos praplečiamumui, našumui ir greitaveikai, prieinamumui ir patikimumui, saugumui. Aprašytas sistemos projekto architektūra, kuriame nurodoma kokie sluoksniai turėt būti realizuojami. Pateiktos modulių veiklų logikos, planuojamos dalykiniai sričių esybių, vartotojo sąsajų diagramos. Visus techninėje bakalauro darbo užduotyje bei analitinėje dalyje suformuotus reikalavimus buvo stengiamasi įvertinti. Pagal visus pateiktus projekto aprašymus toliau interaktyvią medicininę sveikatos paslaugų informacinių sistemą ir jos modulius reikia realizuoti ir ištenuoti.

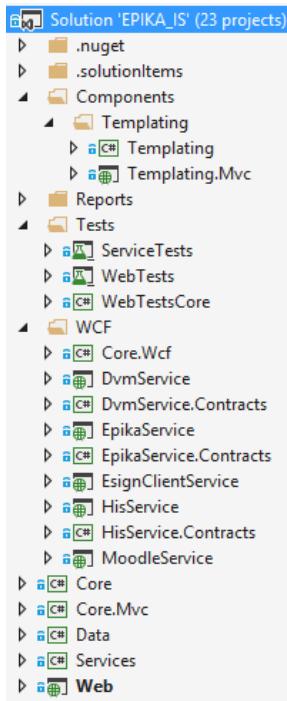
3. REALIZACIJA IR TESTAVIMAS

Atlikus palyginamąjį analizę ir suprojektavus kuriamą sistemą sekantis etapas yra realizacija. Pagal pateiktus aprašymus „Sveikatos priežiūros paslaugų perkėlimo į elektroninę erdvę“ sistema realizuojama ir testuojama.

3.1. Sistemos projektų architektūra

„Visual Studio“ irankis teikia du konteinerius, siekiant padėti efektyviai valdyti elementus, kurie reikalingi sistemos kūrimui [21]. Šie konteineriai yra vadinami sprendimai („Solutions“) ir projektai („Projects“). Sprendimų konteineris padeda efektyviai valdyti kuriamos sistemos projektus.

Sistemą ir ją sudarantys moduliai bus išskaidyti ir sudaryti iš keletos atskirtų ir kiekvienas tik savo skirtą paskirtį turinčių projektų. Projektai yra naudojami logiškai valdyti, kurti, derinti sistemą sudarančias dalis. Šios dalys gali būti įvairūs failai, nuorodos į bibliotekas, duomenų ryšiai, aplankai. Kai kurios jų atstovauja fizinių elementų, kurį galima saugoti talpykloje (kietajame diske), kitos atstovauja nuorodas į kitus objektus, paslaugas, modulius, ar komponentus. Projekto tipų yra įvairių: klasės, valdiklio, bibliotekos, servisų, „Windows“ aplikacijos, „Web“ projekto. Aptarsime kokius projektus išskyriame kuriamos sistemos įgyvendinimui. Kuriamos sistemos sprendimų konteineris yra pavaizduotas 3.1 paveiksle.



3.1 pav. Sistemos sprendimų ir projektų išskaidymas

Kuriami sistemos moduliai naudosis:

Internetinės aplikacijos projektai:

- „Web“ – visos pagrindinės sistemą sudarančios dalys – valdikliai, modeliai, vaizdų modeliai, vaizdai, servisai, įvairūs plėtiniai („plugins“, „helpers“), sistemos grafikos elementai (paveikslėliai, šablonai), konfigūraciniai failai, nuorodos į naudojamus servisus.
- „WCF“ aplankė „HisService“ – sistemos bendradarbiavimui ir duomenų apsikeitimui skirtas projektas, su savo atskirais modeliais, prižiūrėtojais („handlers“), padėjėjų klasėmis („helpers“), ir konfigūracija.

- „Components“ aplanke „Templating.Mvc“ – sistemoje naudojamų tekstinių šablonų valdymui skirtas projektas, kurį sudarys atskiri valdikliai, vaizdų modeliai, vaizdai, servisai ir konfigūraciniai failai.

Klasių bibliotekos projektai:

- „Services“ – nusakoma visų sistemoje naudojamų išorinių servisių, paslaugų bendradarbiavimo veiklos logika. Taip pat apibrėžiami įvairūs sistemos modulių darbai („Jobs“), pranešimų siuntinėjimas, duomenų atnaujinimas iš išorinių sistemų ir kt.
- „Data“ – sistemos duomenų projektas, skirtas bendradarbiavimui su sistemos duomenų baze. Valdoma duomenų bazės prieiga, užklausos iš servisių sluoksnių, duomenų priskyrimas (“mapping”), duomenų bazės kitimo migracijos.
- „Core“ – talpinamas sistemos visuose projektuose naudojamas bendras funkcionalumas: lokalizacija, saugumo, navigacijos logika, formatavimo sprendimai ir kt.
- „Core.Mvc“ – pagalbinis „Core“ projektas, skirtas apibrėžti gilesnę „Core“ projekte naudojamą funkcionalumą logiką ir infrastruktūrą.
- „WCF“ aplanke „Core.Wcf“ – talpinamas servisių veiklą apibrėžianti būsenos logika: klaidos pranešimų, pranešimų siuntimo, užklausų, veiksmų eigos, rezultatų būsenos („logging“).
- „WCF“ aplanke „HisService.Contracts“ – talpinami servisių siunčiamų užklausų, pranešimų ir gaunamų atsakymų duomenų modeliai.
- „Components“ aplanke „Templating“ – talpinama šablonų lokalizacija, modeliai, šablonų servisas, šablonų duomenų bazės lentelių migracijos.

Testavimo projektai:

- „ServiceTests“ – sistemos „WCF“ servisių testavimui skirtas projektas, turintis ir aprašantis šių servisių testavimo logiką, testavimui reikalingus netikrus duomenis, konfigūraciją.

3.2. Sistemos loginės architektūros realizacija

Šiame skyriuje aprašoma visa realizuota sistemos loginė architektūra ir pateikiami konkretūs pavyzdžiai, nusakantys kiekvieną realizuotą architektūrinį lygmenį. Pavyzdžiai bus pateikiami iš nuotolinių stebėjimo modulio, kurie aprašys nuotolinių stebėjimo sąrašo duomenų puslapio pateikimo procesą ir vykdymo eiga.

Valdiklio sluoksnio realizacijos fragmentas yra atvaizduotas 3.2 paveiksle. Pavyzdje yra parodyta „Create“ metodų realizacija. Pirmasis metodas yra skirtas duomenų formos „Get“ atvaizdavimui, antrasis tų duomenų „Post“ pateikimui. „SiteBaseServiceController“ klasė – bendra visus sistemos servisių kontrolerius aprašanti klasė. Šiuo atveju šis valdiklis paveldi šią klasę ir save aprašantį servisą „RemoteObservartionService“. Kreipiantis į pirmajį metodą iš serviso yra grąžinamas atitinkamam „Create“ puslapiui atvaizduoti skirtas vaizdų modelis. Antrajame metode yra patikrinamas pateikiamų duomenų teisingumas, serviso metodu „Save“ vaizdų modelyje talpinamus duomenis yra siekiama išsaugoti į duomenų bazę, šio veiksmo atlikimo rezultatai yra grąžinami ir yra patikrinami, ar veiksmas buvo įvykdytas sėkmingai. Sėkmingu atveju, vartotojas yra nukreipiamas atitinkamą puslapį su pranešimu apie sėkmingą veiksmo atlikimą, klaidos atveju – iš naujo duomenimis yra užpildomas vaizdų modelis ir nukreipiama atgal į „Create“ vaizdą.

```

public partial class RemoteObservationController : SiteBaseServiceController<RemoteObservationService>
{
    ...
    public virtual ActionResult Create()
    {
        var model = Service.GetCreateModel(); // Kreipimasis į serviso lygmenį
        return View(model);
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public virtual ActionResult Create(RemoteObservationCreateEditViewModel model)
    {
        if (ModelState.IsValid)
        {
            var result = Service.Save(model); // Kreipimasis į serviso lygmenį
            if (result.Success)
            {
                return RedirectToActionSuccess(Mvc.RemoteObservation.Edit(result.Id));
            }
            AddErrors(result);
        }
        Service.FillCreateEditModel(model); // Kreipimasis į serviso lygmenį
        return View(model);
    }
}

```

3.2 pav. Valdiklio sluoksnio realizacijos fragmentas

Serviso sluoksnio realizacijos fragmentai yra atvaizduoti 3.3 ir 3.4 paveiksluose. Pavyzdyje yra parodyta „Create“ valdiklio metodo kviečiamas serviso metodo „Save“ realizacija. „BaseService“ klasė – bendra visus sistemos servisus aprašanti klasė. Šis servisas paveldi šią klasę. Servise yra aprašomi visi kiti šio serviso metodų implementacijos logikoje reikalingi servisai. Metodas „Save“ sukuria naują objektą, kuriame bus talpinami iš vaizdų modelio gauti formos duomenys. Objektą užpildžius duomenimis, jis yra bandomas išsaugoti duomenų bazėje, kreipiantis į duomenų bazę lygmeni. Veiksmo atlikimo būsena yra grąžinama atgal į valdiklį.

```

public class RemoteObservationService : BaseService
{
    private readonly LookupService lookupService;
    private readonly ILocalizationService localizationService;
    private readonly RemoteObservationIndicatorValueTypeFormatter indicatorFormatter;
    private readonly PatientRemoteObservationService patientRemoteObservationService;

    public RemoteObservationService(ISiteDataAccess dataAccess, LookupService lookupService,
        : base(dataAccess)
    {
        this.lookupService = lookupService;
        this.localizationService = localizationService;
        this.indicatorFormatter = indicatorFormatter;
        this.patientRemoteObservationService = patientRemoteObservationService;
    }

    public RemoteObservationCreateEditViewModel GetCreateModel()
    {
        var observation = new RemoteObservationCreateEditViewModel
        {
            IsValid = true
        };
        FillCreateEditModel(observation);
        return observation;
    }
}

```

3.3 pav. Serviso sluoksnio realizacijos fragmentas

```

public · ServiceResult · Save(RemoteObservationCreateEditViewModel · model)
{
    ···· var · result ·=· new · ServiceResult();
    ···· var · observation ·=· new · RemoteObservation(); Modelio sluoksnio atitinkmuo

    ···· if ·(!result.Success)
    ···· {
    ····· return · result;
    ···· }

    ···· observation.Name ·=· model.Name;
    ···· observation.DeviceTypeId ·=· model.DeviceTypeId.GetValueOrDefault();
    ···· observation.IsValid ·=· model.IsValid;
    ···· observation.IsPublic ·=· model.IsPublic;
    ···· observation.IsDeviceUsed ·=· model.IsDeviceUsed;
    ···· observation.Comment ·=· model.Comment;
    ···· observation.Description ·=· model.Description;
    ···· observation.ChartType ·=· model.ChartType;
    ···· observation.Frequency ·=· model.FrequencyValue.GetValueOrDefault();
    ···· observation.DocumentTypes ·=· model.DocumentTypes;

    ···· observation.IndicatorPeriods ·=· SaveIndicatorPeriods(observation, · model.Periods);
    ···· SaveDiagnosis(model, · observation);
    ···· SaveSpecialties(model, · observation);

    ··· DataAccess.Insert(observation); Kreipimasis į duomenų bazės lygmenį
    ··· DataAccess.Save();
    ··· result.Id ·=· observation.Id;

    ··· return · result;
}

```

3.4 pav. Serviso sluoksnio metodo realizacijos fragmentas

Modelio sluoksnio realizacijos fragmentas yra atvaizduotas 3.5 paveiksle. Pavyzdyme yra parodyta nuotolinio stebėjimo modelis su save apibūdinančiais parametrais ir juos nusakančiais atributais. Atributas „[Required]“ nusako, kad parametras, nusakantis modelio įrašą turi būti visuomet egzistuojantis (angl. „not null“). Atributas „[StringLength]“ nusako tekstinės reikšmės parametru maksimalų ilgi. „Virtual“ parametru reikšmės tipas nurodo ryšį su „DeviceType“ modeliu ir šis modelis gali būti pasiekiamas būtent per ši parametrą. Konstruktorius „RemoteObservation()“ užpildo naujai sururiamą modelio objektą atitinkamais duomenimis: matavimo rodiklių periodų, specialybių sąrašais, diagnozės įrašu.

```

public · class · RemoteObservation ·:· BaseEntity
{
    ···· – references
    ···· public · RemoteObservation()
    ···· {
    ····· · IndicatorPeriods ·=· new · HashSet<RemoteObservationIndicatorPeriod>();
    ····· · Diagnosis ·=· new · HashSet<Diagnosis.Diagnosis>();
    ····· · Specialties ·=· new · HashSet<Specialty>();
    ···· }

    ···· [Required]
    ···· [StringLength(400)]
    ···· – references
    ···· public · string · Name ·{· get; · set; ·}

    ···· [Required]
    ···· – references
    ···· public · int · DeviceTypeId ·{· get; · set; ·}
    ···· – references
    ···· public · virtual · DeviceType · DeviceType ·{· get; · set; ·}

    ···· [Required]
    ···· – references
    ···· public · bool · IsValid ·{· get; · set; ·}

    ···· [Required]
    ···· – references
    ···· public · bool · IsPublic ·{· get; · set; ·}

    ···· – references
    ···· public · bool · IsDeviceUsed ·{· get; · set; ·}

    ···· – references
    ···· public · string · Comment ·{· get; · set; ·}
}

```

3.5 pav. Modelio sluoksnio realizacijos fragmentas

Vaizdo modelio sluoksnio realizacijos fragmentas yra atvaizduotas 3.6 paveiksle. Pavyzdyme yra parodytas nuotolinio stebėjimo „Create“ vaizdui atvaizduoti skirtas jo duomenų modelis. Jis apibrėžia tik šiame vaizde atvaizduojamų duomenų rinkinį, pateikiant tik tai jam skirtą ir reikiama informaciją. Atributas „[LocalizedDisplay]“ nusako, lokalizuotą parametru reikšmės pavadinimą, kuris yra atvaizduojamas prieš atitinkamos reikšmės duomenų įvedimų formoje lauką. Atributas „[LocalizedStringLength]“ nusako tekstinės reikšmės parametru maksimalų įvedamą ilgį, „[LocalizedRequired]“ – nurodo privalomą užpildyti formos lauką. Neįvykdžius laukų reikalavimų yra atvaizduojami lokalizuoti klaidos pranešimai.

```
public class RemoteObservationCreateEditViewModel
{
    --references
    --public RemoteObservationCreateEditViewModel()
    --{
    --    Periods = new List<PeriodMreItemViewModel>();
    --}

    --references
    --public int Id { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.Name)]
    --[LocalizedRequired]
    --[LocalizedStringLength(400)]
    --references
    --public string Name { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.DeviceType)]
    --[LocalizedRequired]
    --references
    --public int? DeviceTypeId { get; set; }
    --references
    --public string InitDeviceType { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.IsValid)]
    --references
    --public bool IsValid { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.IsPublic)]
    --references
    --public bool IsPublic { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.IsDeviceUsed)]
    --references
    --public bool IsDeviceUsed { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.Comment)]
    --references
    --public string Comment { get; set; }

    --[LocalizedDisplay(Msg_eCore.RemoteObservations.Description)]
    --references
    --public string Description { get; set; }
}
```

3.6 pav. Vaizdo modelio sluoksnio realizacijos fragmentas

Vaizdo sluoksnio realizacijos fragmentas yra atvaizduotas 3.7 paveiksle. Pavyzdyme yra parodytas nuotolinio stebėjimo „Create“ vaizdą sudarantys „HTML“, „Razor“, „CSS“ ir „Bootstrap“ kodo turinys. Vaizdo kodo aprašo pradžioje „@model“ nurodo vaizde naudojamą vaizdo modelį, iš kurio bus atvaizduojami formos elementai ir duomenys. „ViewBag.Title“ nurodo naršyklės skirtuko antraštę, „PageTitle()“ pagrindinę puslapyje atvaizduojamą antraštę. „@using“ sakiny apibrėžia „HTML“ formos atitikmenį, nurodo kreipimosi tipą, elementui būdingas klasses. „HiddenFor“ paslepia norimą vaizdo modelio duomenis formoje, kad jie išliktų po duomenų pateikimo ir nebūtų prieinami redagavimui. „HelpDesktop()“ visoje sistemoje naudojamas informacinis pagalbos komponentas vartotojui. „Partial“ nurodo kitą, dalinį vaizdo puslapį, kuris yra saugomas kitame faile. Šio vaizdo fragmentas yra pateikiamas 3.8 paveiksle. Tokio puslapio tikslas – perpanaudojamumas keliuose skirtinguose puslapiuose. Šiuo atveju šis dalinis vaizdas yra naudojamas ir „Create“ ir „Edit“ puslapiuose. „DisplaySofButtonsFor“ mygtukus atvaizduojantis komponentas. „HelpMobile“ informacinis pagalbos komponentas vartotojui mobiliojoje aplinkoje. Vaizdo puslapio apačioje yra aprašomi ir iškviečiami puslapiui reikalingi skriptai.

Daliname puslapyje „AntiForgeryToken()“ komponentas skirtas galimam formos duomenų klastojimui patikrinti. „LabelRequiredFor“ nurodo formos lauko antraštės pavadinimą su privalomu „*“ simboliu. „TextBoxFor“ atvaizduoja tekstinio tipo duomenų formos laukelį. „ValidationMessageFor“ atvaizduoja informacinių pranešimą neteisingų formos laukelio duomenų pateikimo atveju.

```
@using Epika.Core.Mvc.Infra.Link
@model Epika.Web.ViewModels.RemoteObservations.RemoteObservationCreateEditViewModel
@{
    ViewBag.Title = L(Msg_eCore.RemoteObservations.Register);
    Layout = Mvc.Shared.Views._Layout;
}

@Html.PageTitle()

@using (Html.BeginForm(Mvc.RemoteObservation.Create(), FormMethod.Post, new { @class = "sof-sof-sm-form" }))
{
    @Html.HiddenFor(x => x.Id)
    @Html.HelpDesktop()
    <div class="sof-content">
        @Html.Partial(Mvc.RemoteObservation.Views._EditPartial)
    </div>
    @Html.Epika().DisplaySofButtonsFor(
        new BackLink(Html.Epika().Back(Mvc.RemoteObservation.Index())),
        new PrimaryLink(Html.Epika().SubmitButton(L(Msg_Core.SharedStrings.Save), 'V'), width: 16))
}

@Html.Action(Mvc.Help.HelpMobile(isButtonShown: false))

@section Scripts {
    <script type="text/javascript">
        $(document).ready(function () {
            if (typeof initRemoteObservationEditPartial != 'undefined') {
                initRemoteObservationEditPartial.init();
            }
        });
    </script>
}
```

3.7 pav. Vaizdo sluoksnio realizacijos fragmentas

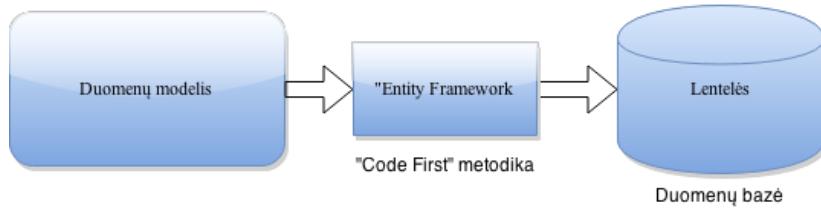
```
@using Epika.Core.Enums
@model Epika.Web.ViewModels.RemoteObservations.RemoteObservationCreateEditViewModel
@Html.AntiForgeryToken()
<div class="row">
    <div class="col-sm-12">
        <div class="form-group">
            @Html.LabelRequiredFor(model => model.Name)
            @Html.TextBoxFor(model => model.Name, new { @class = "form-control" })
            @Html.ValidationMessageFor(model => model.Name)
        </div>
    </div>
    <div class="col-sm-12">
        <div class="form-group">
            @Html.LabelRequiredFor(m => m.DeviceTypeId)
            @Html.TextBoxFor(m => m.DeviceTypeId, new { @class = "form-control-deviceType", data_init = "", data_init_select = Json.Encode(new { url = Url.Action(Mvc.Lookup.DeviceTypes()), placeholder = L(Msg_eCore.RemoteObservations.ChooseDeviceType).ToString(), dropdownAutoWidth = true, initSelection = Model.InitDeviceType } ) })
            @Html.ValidationMessageFor(m => m.DeviceTypeId)
        </div>
    </div>
</div>
```

3.8 pav. Vaizdo dalinio sluoksnio realizacijos fragmentas

3.3. Sistemos duomenų bazės realizacija

Pateiksime duomenų bazės lenteles ir ryšius tarp jų atvaizduojančią aprašo schemą. Ši schema atspindi duomenų struktūras „MS SQL“ serverio duomenų bazėje.

Sistemos duomenų bazė buvo kuriama pasitelkiant sistemos projektavimo etape numatytais dalykinės srities esybių klasėmis. Naudojant „Entity Framework“ technologijos teikiamomis galimybėmis iš šių klasių ir jas sudarančių atributų ir ryšių buvo generuojama duomenų bazės aprašo schema pritaikant „Code First“ kūrimo modelį, kuris yra pavaizduotas 3.9 paveiksle [22].



3.9 pav. „Entity Framework Code First“ duomenų bazės priskyrimo modelio schema

3.10 paveiksle pateikiamas nuotolinio stebėjimo modulio įrenginio dalykinės sritys klasės modelio pavyzdys. Šis modelis yra duomenų bazės apraše esančios lentelės atitinkmuo.

```

public class Device : BaseEntity
{
    ...
    public Device()
    {
        ...
        PatientRemoteObservations = new HashSet<RemoteObservations.PatientRemoteObservation>();
        ...
    }

    [Required]
    [StringLength(200)]
    ...
    public string Name { get; set; }

    ...
    public int? TypeId { get; set; }
    ...
    public virtual DeviceType Type { get; set; }

    ...
    public int? StateId { get; set; }
    ...
    public virtual DeviceState State { get; set; }

    ...
    public int? KindId { get; set; }
    ...
    public virtual DeviceKind Kind { get; set; }

    ...
    [StringLength(100)]
    ...
    public string No { get; set; }

    ...
    public DateTime? RegistrationDate { get; set; }

    ...
    public int? DivisionId { get; set; }
    ...
    public virtual Division Division { get; set; }

    ...
    public int? AssignedToPatientId { get; set; }
    ...
    public virtual User AssignedToPatient { get; set; }

    ...
    [Required]
    ...
    public int? ModelId { get; set; }
    ...
    public virtual DeviceModel Model { get; set; }

    ...
    public DateTime? PlannedServiceDate { get; set; }

    ...
    [InverseProperty("Devices")]
    ...
    public virtual ICollection<RemoteObservations.PatientRemoteObservation> PatientRemoteObservations { get; set; }
    ...
    public decimal? PriceEur { get; set; }
}

```

3.10 pav. Įrenginio dalykinės sritys klasės modelio fragmento pavyzdys

Sistemoje naudojamoje „SiteDataAccess.cs“ klasėje yra nurodoma, kas turi būti sukuriama duomenų bazėje. Ši klasė skirta sąveikauti tarp sistemos modelių ir duomenų bazėje esančių lentelių tarp jų palaikant ryšį. Ji gali pastebeti galimus pakitimus modelyje, tuos pakitimus aprašyti migracijomis ir juos tuo pačiu atispindėti duomenų bazės struktūroje. 3.11 paveiksle parodomos „SiteDataAccess.cs“ klasėje apibrėžtas modelio duomenų kontekstas, kuris turės būti atkuriamas duomenų bazėje, jos struktūros pavidalu.

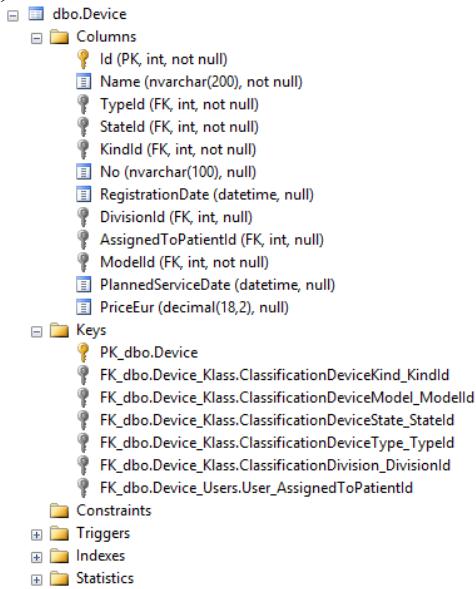
```

    public DbSet<Device> Devices { get; set; }

```

3.11 pav. „SiteDataAccess.cs“ klasės fragmento pavyzdys

Po sistemos kodo sukompiliavimo, aptiktos migracijos sukūrimo ir jos įvykdymo, duomenų bazėje yra sukuriamas nurodytą modelį atspindis duomenų bazės lentelės atitinkmuo, kuris yra pavaizduotas 3.12 paveiksle. Modelio parametrai, jų turimi atributai, ir ši modelių sudarantys kitų modelių objektai yra perverčiami į duomenų bazės tipinius duomenų rinkinius: lentelės stulpelius, jų tipus, apribojimus, ryšius, pirminius, išorinius raktus.

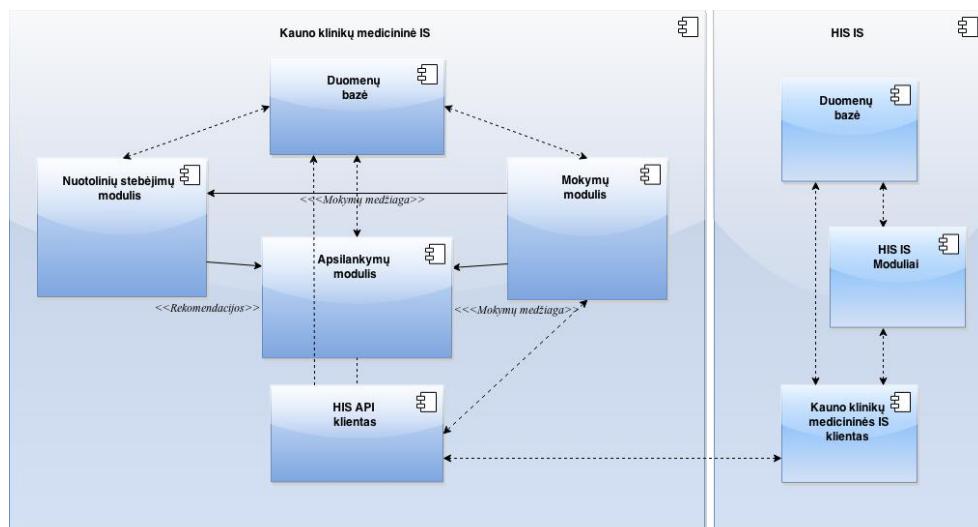


3.12 pav. Duomenų bazės įrenginio (angl. „Device“) lentelės struktūra

Sistemoje įgyvendintų paslaugų modulių duomenų lentelių schemas nepateikinėsime, kadangi duomenų bazės schemą dėl šiame skyrelyje išvardintų priežasčių puikiai atspindi sistemos modulių dalykinės srities esybių klasių diagramos, kurios yra pateiktos sistemos projektavimo etape.

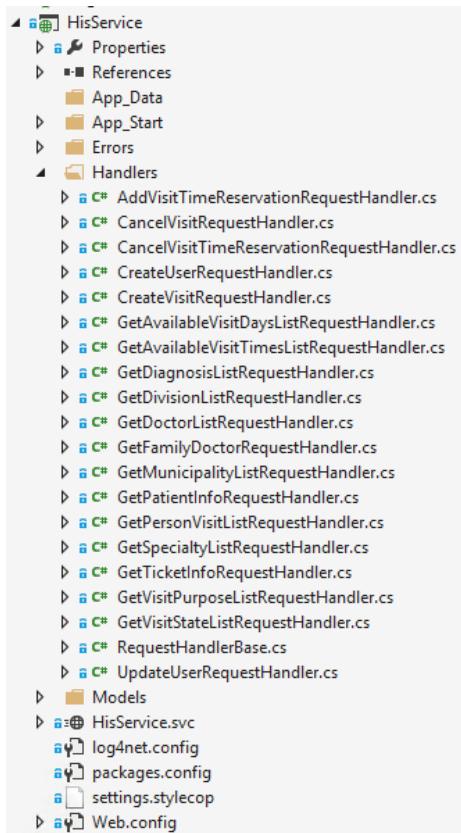
3.4. Sistemos duomenų keitimosi lygmens realizacija

Sistemos loginės architektūros projektavimo metu buvo paminėtas naudojamas „Sistemų tarpininkavimo lygmuo“. Šis lygmuo, kaip ir buvo aptarta, yra atsakingas už duomenų siuntimą žinutėmis tarp serviso pradžios taško, kuri šiuo atveju yra mūsų sukurtoji sistema, ir serviso pabaigos taško – kita „HIS“ medicininė informacinė sistema, kuri dalijasi mūsų sistemai reikalingais duomenimis. Šiam duomenų mainų proceso įgyvendinimui buvo panaudota „WCF“ technologija. Sistemos modulių duomenų mainus atvaizduoja modulių ryšių diagrama, pavaizduota 3.13 paveiksle.



3.13 pav. Sistemos modulių ryšių diagrama

Visus atvaizduotus modulių ryšius aprašo 3.14 paveiksle pateikiamos visos sistemos modulių naudojamos ir aprašytos užklausų klasės. Jų pavadinimai atspindi savo turimą logiką, atliekamą funkcionalumą ir prasmę.



3.14 pav. Sistemos modulių užklausų į „HIS“ sistemą klasų sąrašas

Pateiksime vieno iš 3.14 paveiksllo sąraše esančių užklausų iškvietimo pavyzdį. „GetTicketInfo()“ užklausos iškvietimas yra vykdomas iš apsilankymų modelių nusakančio apsilankymo serviso. Šiame serwise aprašyto metodo fragmentas yra pateikiamas 3.15 paveiksle. Metode yra suformuojamas reikiamas užklausai pradėti vykdyti duomenų objektas. Visas užklausas apibūdinantis „HisServiceClient“ klientas šiuos duomenis siunčia jų vykdymui.

```
public TicketInfoResponse GetTicketInfo(Guid? sessionId, int ticketId)
{
    using (var client = new HisServiceClient())
    {
        var request = new TicketInfoRequest
        {
            SessionId = sessionId.ToString(),
            TicketId = ticketId
        };

        var response = client.GetTicketInfo(request);
        return response;
    }
}
```

3.15 pav. Duomenų mainų metodo iškvietimo pavyzdys

3.18 paveiksle yra pateikiama vienas duomenų mainų užklausos klasės implementacijos logikos fragmento pavyzdys. Visų užklausos klasų logiką galime išskaidyti į tris pagrindines dalis:

- *Užklausos duomenų suformavimo.* Klasė siunčia užklausos duomenis į „HIS“ sistemą. Užklausos duomenų pavyzdys yra pateiktas 3.16 paveiksle. Šiuo atveju siunčiami duomenys yra sesijos identifikacinis numeris („SessionId“) ir talono identifikacinis numeris („TicketId“). Iš šios duomenų užklausos klasės „DataContract“ atributo pagalba yra

suformuojamas duomenų kontraktas, kuris yra atpažįstamas „WCF“ naudojamo serviso. Kiekvieno aprašyto parametru „DataMember“ atributai nusako jų atpažinimo pavadinimus „HIS“ informacineje sistemoje, o „Order“ atributas apibūdiną jų eiliukumo tvarką.

- *Bendradarbiavimo veiksmų logikos.* Logika aprašo sistemos kreipimosi į kitą sistemą ir iš kitos sistemos gaunamo atsako veiksmų apipavidalinimą. 3.18 paveiksle pateiktame pavyzdje „CallHisApi()“ metode, yra aprašoma ši veiksmų seka. Visų pirma yra sukuriamas kliento objektas „HisApiClient()“, kuris turi reikiamas informacijos apie sistemų jungimosi informaciją: duomenų bazių duomenis, perpanaudojamus metodus tarp skirtingų užklausų klasių. Per ši klientą vyksta pagrindiniai informacijos mainai tarp sistemų. Yra iškviečiamas metodas, gauti talonų duomenims, nusiunčiant į „HIS“ sistemą talono identifikacinį numerį. Gaunami talono duomenys, tai sistemos atsakas į mūsų sistemos išsiustą užklausą. Metode taip yra gaunama patikslinta informacija ir apie talono apsilankymo įvykimo adreso ir kabineto duomenis.
- *Atsako duomenų formavimo.* Gauti duomenys iš „HIS“ sistemos yra įvertinami ir talpinami į atitinkamą atsako klasę. Duomenų atsako klasės pavyzdys yra pateikiamas 3.17 paveiksle.

```
[DataContract(Namespace = Namespaces.Contracts)]
public class TicketInfoRequest
{
    [DataMember(Name = "SESSION_ID", Order = 10)]
    public string SessionId { get; set; }

    [DataMember(Name = "TICKET_ID", Order = 20)]
    public int TicketId { get; set; }
}
```

3.16 pav. Sistemos užklausos duomenų klasės pavyzdys

```
[DataContract(Namespace = Namespaces.Contracts)]
public class TicketInfoResponse
{
    [DataMember(Name = "TICKET_ID", Order = 10)]
    public int Id { get; set; }

    [DataMember(Name = "VISIT_TYPE_ID", Order = 20)]
    public int VisitTypeId { get; set; }

    [DataMember(Name = "VISIT_PURPOSE_ID", Order = 30)]
    public int VisitPurposeId { get; set; }

    [DataMember(Name = "START_TIME", Order = 40)]
    public DateTime StartTime { get; set; }

    [DataMember(Name = "CABINET", Order = 50)]
    public string Cabinet { get; set; }

    [DataMember(Name = "DIVISION_ID", Order = 60)]
    public int DivisionId { get; set; }

    [DataMember(Name = "DOCTOR_ID", Order = 70)]
    public int DoctorId { get; set; }

    [DataMember(Name = "SPECIALTY_ID", Order = 80)]
    public int SpecialtyId { get; set; }

    [DataMember(Name = "RESERVATION_SECONDS_LEFT", Order = 90)]
    public int ReservationSecondsLeft { get; set; }

    [DataMember(Name = "ADDRESS", Order = 100)]
    public string Address { get; set; }
}
```

3.17 pav. „HIS“ sistemos atsako duomenų klasės pavyzdys

```

public class GetTicketInfoRequestHandler : RequestHandlerBase<TicketInfoResponse>
{
    private readonly TicketInfoRequest request;

    public GetTicketInfoRequestHandler(TicketInfoRequest request)
    {
        this.request = request;
    }

    protected override TicketInfoResponse ExecuteCore()
    {
        var param = new { sessionId = request.SessionId, ticketId = request.TicketId };
        return Db.Query<TicketInfoResponse>("epika.GetTicketInfo", param, commandType:
            CommandType.StoredProcedure).FirstOrDefault();
    }

    protected override TicketInfoResponse CallHisApi()
    {
        using (var hisClient = new HisApiClient())
        {
            var ticket = hisClient.GetTicketById(request.TicketId);

            var resources = hisClient.GetResources(new Criteria() { StartDate = DateTime.Now });
            var reservationTime = hisClient.GetLockTime();

            var place = resources.Places[ticket.PlaceId.ToString()];
            var cabinet = place.Substring(place.LastIndexOf("-") + 3);

            var result = new TicketInfoResponse()
            {
                DivisionId = ticket.StunId,
                DoctorId = ticket.StaffId,
                Id = ticket.Id,
                StartTime = ticket.DateAndTime,
                SpecialtyId = ticket.ProfessionId,
                VisitTypeId = 1, // TODO: join
                VisitPurposeId = ticket.ScheduleConfigId,
                Cabinet = cabinet,
                ReservationSecondsLeft = (int)new TimeSpan(0, reservationTime, 0).TotalSeconds,
                Address = resourcesBuildings[ticket.BuildingId.ToString()]
            };

            return result;
        }
    }
}

```

3.18 pav. Sistemos užklausos klasės į „HIS“ sistemą implementacijos fragmentas

3.5. Sistemos pranešimų siuntimo realizacija

Sistemoje buvo plačiai naudojamas sisteminiai pranešimai, elektroniniai laiškų siuntimo funkcionalumas, kurį teko panaudoti mūsų kuriamiems moduliams. Šiame skyrelyje aptarsime pranešimų siuntimo logiką ir pateiksime tai atvaizduojantį pavyzdį. Nuotolinių stebėjimų ir apsilankymų moduliuose, priklausomai nuo atitinkamo veiksmo sėkmingo įvykdymo, reikia išsiusti tai iliustruojantį informacinį pranešimą.

Pateiksime detalų šiu išsiunčiamų pranešimo sąrašą.

Apsilankymų modulyje:

- SendVisitEarlierTimeNotificationPatients – siunčiamas pranešimas pacientams, kai gydytojas informuoja atitinkamus pacientus apie ankstesnio apsilankymo pasirinkimą.
- SendVisitCreateDoctorVisit – siunčiamas pranešimas pacientams apie jų arba gydytojų sukurtą apsilankymą.
- SendVisitEditDoctorVisit - siunčiamas pranešimas pacientams apie jų arba gydytojų pakeistą apsilankymo datą.
- SendVisitCancelDoctorVisit - siunčiamas pranešimas pacientams apie jų arba gydytojų atšauktą apsilankymą.

Nuotolinių stebėjimų modulyje:

- SendRecommendationMessage – siunčiamas pranešimas pacientui apie gydytojo pateiktą rekomendaciją.

- SendConsultationMessage – siunčiamas pranešimas gydytojui apie kito gydytojo sukurtą konsultaciją.
- SendFileUploadedMessage – siunčiamas pranešimas gydytojui apie paciento įkeltus stebėjimo duomenis.

Kiti pranešimai, kurie yra išsiunčiami kas 24 valandas:

- SendVisitEarlierTimeNotificationPatientsService – servisas, išsiunčiantis pranešimus pacientams, kurie pažymėjo sukurtame apsilankyme požymį, kad nori gauti informaciją apie galimybę apsilankytį ankščiau.
- SendVisitReminderMailService - servisas, išsiunčiantis pranešimus pacientams, norint priminti apie artėjančią apsilankymą.

Pateiksime apsilankymų modulyje atliekamo apsilankymo atšaukimo veiksmą. Šio veiksmo įvykdymo metu apsilankymo įrašo būseną yra pakeičiamā į atšauktą ir po sėkmingų duomenų išsaugojimo yra išsiunčiamas informacinis pranešimas šio apsilankymo pacientui. Pranešimo siuntimo veiksmo logiką yra pateikta 3.19 paveiksle.

```
private void SendCancelDoctorVisitNotification(Visit visit, string cancelComment)
{
    var container = new
    {
        Date = VisitFormatter.FormatDateWithoutSeconds(visit.Date),
        Specialty = visit.Specialty.Name,
        Doctor = visit.Doctor.FirstName + " " + visit.Doctor.LastName,
        CancelReason = cancelComment
    };

    messageSender.Send(Templates.VisitCancelDoctorVisit, container, visit.UserId,
        deliveryType: MessageDeliveryType.Email | MessageDeliveryType.System);
}
```

3.19 pav. Apsilankymo modulio apsilankymo atšaukimo pranešimo išsiuntimo pavyzdys

Apsilankymų servisas, naudojantis pranešimų siuntimo servisu „messageSender“, prieš tai suformuoja reikiamų duomenų siuntimui objektą. Siuntimo servisiui yra nurodomas pranešimo siuntimui naudojamas šablono tipas, siuntimo duomenų objektas, pranešimo gavėjas ir pranešimo išsiuntimo būdai. 3.20 paveiksle yra pateikta tolimesnė pranešimo siuntimo serviso metodo logika.

```
public ServiceResult Send(Templates type, object container, int? userId, int? organizationId = null,
    MessageDeliveryType? deliveryType = null)
{
    if (!deliveryType.HasValue)
    {
        deliveryType = GetDefaultDeliveryTypes(userId);

        var languageCode = templatingService.GetLanguageCode(null, userId);
        var msg = templatingService.Format(templateService.GetTemplate(type.Value, languageCode), container);
        return Send(msg, userId, deliveryType.Value, organizationId);
    }
}
```

3.20 pav. Pranešimų siuntimo serviso metodo fragmento pavyzdys

Metodas įvertina norimus išsiuštį pranešimų siuntimo būdus, vėliau nustato kokia kalba pranešimas turi būti siunčiamas, atsižvelgiant į sistemos vartotojo nustatymuose pasirinktą kalbos parametru. Tuomet šablonų serviso pagalba iš pasirinkto šablonų tipo ir pateiktų duomenų yra suformuojamas pranešimo tekstas. I pasirinktą šablonų tipą yra įterpiami pateikti informacinių duomenys. Suformuotas pranešimas yra pateikiamas kitam siuntimo metodui.

Tolimesnė pranešimo siuntimo logika įvertina ar pranešimas nėra tuščias, ar pranešimo gavėjas egzistuoja sistemoje, ar buvo nurodytas nors koks pranešimo išsiuntimo būdas. Sėkmingai patenkinus šiuos punktus, pranešimai yra išsiunčiami nurodytais išsiuntimo būdais. Sistemos pranešimai yra saugomi atitinkamoje sistemos duomenų bazės lentelėje su atitinkamu išsiuntimo

priežasties tipu (ar tai buvo apsilankymo, ar nuotolinio stebėjimo pranešimas ir kt.) Sisteminio pranešimo išsiuntimo logikos fragmentas yra pateikiamas 3.21 paveiksle

```
public ServiceResult SendMessage(string subject, string body, int userId, string userFullName, int? organizationId = null)
{
    var result = new ServiceResult();
    var messageType = GetMessageTypes().FirstOrDefault();
    try
    {
        var message = new Message
        {
            Subject = subject,
            Body = body,
            Receivers = new[] { new Receiver { UserId = userId, Name = userFullName } },
            MessageType = messageType,
            CreateDate = DateTime.Now,
            SenderId = 1
        };

        CreateMessage(message);

        if (organizationId != null)
        {
            var organizationMessage = new OrganizationMessage
            {
                MessageId = message.Id,
                OrganizationId = organizationId.Value,
            };

            siteDataAccess.Insert(organizationMessage);
            siteDataAccess.Save();
        }

        result.Id = message.Id;
    }
    catch (DbEntityValidationException ex)
    {
        result.AddError(ExceptionHelper.GetDbEntityValidationMessages(ex));
    }
    catch (Exception ex)
    {
        result.AddError(ExceptionHelper.GetInnerExceptionMessages(ex));
    }
}

return result;
}
```

3.21 pav. Sisteminį pranešimų siuntimo serviso metodo fragmento pavyzdys

Metode yra atsižvelgiama ir į sistemoje prisijungusios vartotojo ar organizacijos požymį, kas ir kam pranešimas turi būti išsiunčiamas. Nesėkmingi pranešimų išsiuntimai yra taip pat saugomi duomenų bazės lentelėje bei jie taipogi yra traktuojami sistemos klaidų pranešimų įrašų sąrašuose (angl. „logs“).

Elektroninių laiškų siuntimo metodo fragmento pavyzdys yra pateikiamas 3.22 paveiksle. Jame yra naudojama „SmtpClient“ ir „MailMessage“ klasės, kurios yra suteikiamas naudoti „Microsoft .NET“ karkaso. „SmtpClient“ klasė leidžia siusti pranešimus naudojant paprastąjį pašto perdavimo protokolą, o „MailMessage“ atitinka laiško objektą, kurį galime siusti nurodytaja „SmtpClient“ klasės pagalba. Pavyzdyme pateiktas metodas suformuoja laiško išsiuntimo objektą iš mūsų ankščiau pateiktų duomenų. „MailAddress“ klasė atitinka laiško gavėjo arba siuntėjo objektą, naudojamą „SmtpClient“ klasėje. Šiuo atveju metode yra nurodomas laiško gavėjas. Šis metodas laiškus gali siusti asinchroniškai arba sinchroniškai, priklausomai nuo pasirinkto laiško siuntimo būdo. Nepavykė laiškų išsiuntimai yra nuolat traktuojami ir saugomi sistemoje. „ServiceResult“ klasės pagalba siuntimo būsena yra įrašoma ir išsaugoma šios klasės sukurtame objekte ir jis yra grąžinamas atgal per visus aptartus servisų sluoksnius iki apsilankymų serviso, kuris šį rezultatą grąžiną kontroleriui, o šis tuomet atitinkamai atvaizduoja vartotojui klaidos pranešimą arba sėkmingą veiksmo atlikimą vaizdo peržiūros sluoksnje. Siuntimo rezultatų gražinimą kontroleriui ir rezultatų įvertinimo logiką galite pamatyti 3.23 paveiksle.

```

private ServiceResult SendMail(SendEmailModel model)
{
    var result = new ServiceResult();
    var client = new SmtpClient();

    var email = new MailMessage { Subject = model.Subject, Body = model.Body, IsBodyHtml = true };
    email.To.Add(new MailAddress(model.Recipient.Address, model.Recipient.FullName));

    if (model.Async)
    {
        try
        {
            Task.Factory.StartNew(() =>
            {
                try
                {
                    client.Send(email);
                }
                catch (Exception ex)
                {
                    Logger.Error("SendMail.async.failed.with.exception", ex);
                    var dataAccess = ServiceLocator.Resolve<ISiteDataAccess>();
                    SaveFailedMessage(dataAccess, model.Subject, model.Body, model.Recipient.Address,
                        model.Recipient.FullName, MessageDeliveryType.Email, ExceptionHelper.GetInnerExceptionMessages(ex));
                }
            });
        }
        catch (Exception e)
        {
            Logger.Error("SendMailAsync.exception", e);
            result.AddError("Mail.sending.failed", ExceptionHelper.GetInnerExceptionMessages(e));
        }
    }
    else
    {
        try
        {
            client.Send(email);
        }
        catch (Exception ex)
        {
            Logger.Error("SendMail.exception", ex);
            result.AddError("Mail.sending.failed", ExceptionHelper.GetInnerExceptionMessages(ex));
        }
    }
}

return result;
}

```

3.22 pav. Elektroninių laiškų siuntimo serviso metodo fragmento pavyzdys

```

public virtual ActionResult CancelDoctorVisit(DoctorVisitDetailsViewModel model)
{
    if (model.CancelComment != null)
    {
        var result = Service.CancelDoctorVisit(model, true);
        if (result.Success)
        {
            return RedirectWithSuccess(Mvc.Visit.DoctorVisit(),
                LocalizationService.Get(Msg_eCore.Visits.CancelSuccessMessage));
        }
        return RedirectWithErrors(Mvc.Visit.Details(model.Id), result.ToString());
    }
    return RedirectWithErrors(Mvc.Visit.Details(model.Id),
        LocalizationService.Get(Msg_eCore.Visits.NoVisitCancelComment));
}

```

3.23 pav. Apsilankymų modulio kontrolerio metodo pranešimų siuntimo pavyzdys

3.6. Sistemos realizacijos testavimas

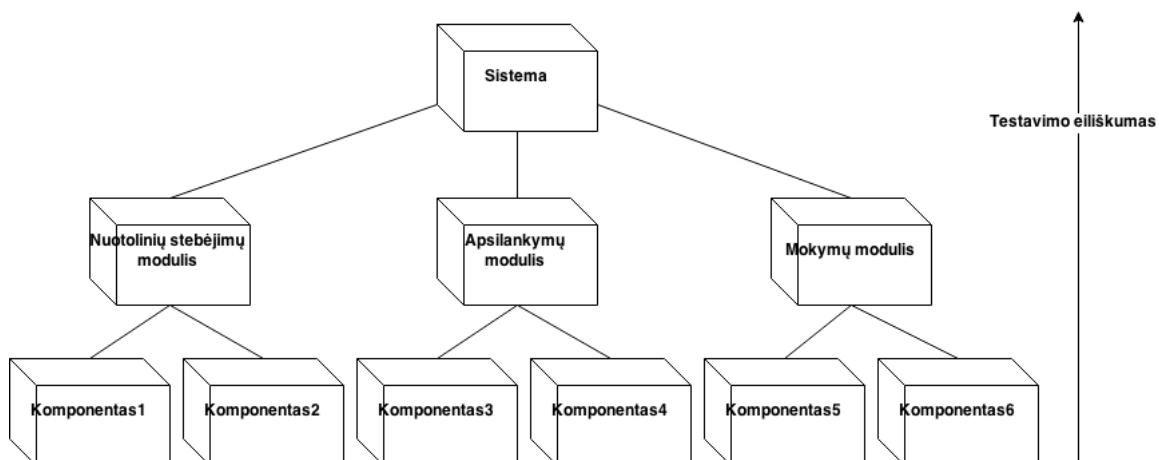
Realizavus medicininės informacinės sistemos modulius ją reikia ištesti. Šiame skyriuje aprašoma visa su sukurtos sistemos testavimu susijusi informacija – testavimo planas (aplinka ir metodika, kaip bus testuojama), pateikiami testiniai atvejai ir atlikti vienetiniai testai [30].

Prieš testuojant sukurtą sistemą reikia susidaryti testavimo planą – kaip sistema bus testuojama. Pirmiausiai reikia apsibrėžti testavimo aplinką. Programinė testavimo aplinkos įranga susideda iš :

- Serverio – kompiuteris su įdiegtu „IIS“ serveriu („Windows“ operacinė sistema) ir „MS SQL Server“ duomenų bazių valdymo sistema, kuriame veikia sukurtoji informacinė sistema ir joje implementuoti moduliai;
- Kliento kompiuterio su „Visual Studio 2013 Ultimate“ programavimo aplinka, kuri turi „Unit“ testams atliskti skirtą testavimo automatizavimo programinę įrangą, ir internetinės naršyklos;

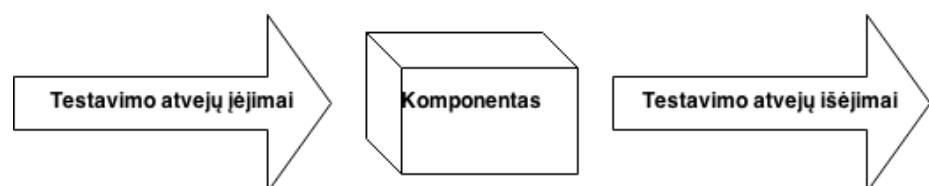
Sistemos testuotojai. Atliskti sistemos testavimo procesus jos kūrimo metu buvo įpareigoti visi programuotojai. Užbaigus kiekvieną savo atliekamą užduotį (angl. „task“) buvo privaloma išsitestuoti atlisktos užduoties gautą rezultatą – ar viskas veikia korektiškai, ar neatsirado kokių nesklandumų, klaidų. Išitikinus, kad atliska užduotis veikia tikslingai ir gaunami rezultatai yra korektiški, užduotis būdavo paruošta testavimui, realioje testavimo aplinkoje. Užduotims, kurios atlkdavo svarbesnes sistemos funkcijas, tokias kaip pranešimų siuntimus, duomenų mainus tarp sistemų, buvo rašomi vienetiniai testai (angl. „Unit tests“) [31].

Testavimo planas. Sukurti sistemos moduliai testuojami pirmiausiai atliekant atskirų programinių komponentų testavimą, tada atliekant integracijos testus principu „iš apačios į viršų“ – testuojama komponentų integracija sudarant didesnius komponentus, kol galiausiai prieinama iki visų sistemos modulių veikimo testavimo. Tokio testavimo pagrindas yra tas, kad būtų atliskti visi reikalingi testai, parodantys ar sukurtos sistemos infrastruktūra veikia patikimai ir stabliai [32]. Žemiau, 3.24 paveiksle, pavaizduota „iš apačios į viršų“ integracijos testavimo principo schema.



3.24 pav. Integracijos testavimo principo „iš apačios į viršų“ schema

Sistemos modulių komponentai testuojami atliekant „juodos dėžės“ testavimą. Tai dinaminis testavimo būdas, kada tikrinama ar yra išpildomi funkciniai programinės įrangos reikalavimai, o pats testuojamas komponentas įsivaizduojamas kaip „juoda dėžė“, nevertinant jo struktūros (žinoma tik apie tai, ką komponentas priima ir ką grąžina, bet nieko nežinoma apie tai, kas yra jo viduje). Žemiau, 3.25 paveiksle, pavaizduota juodos dėžės testavimo metodikos principinė schema.



3.25 pav. „Juodos dėžės“ testavimo schema

Juodos dėžės principu testuojami tokie sistemos modulių komponentai kaip duomenų bazė, pagalbinės klasės (pvz. tekštinių duomenų formatavimo klasė), paruošiant jiems testinius atvejus pagal funkciuose reikalavimuose aprašytus reikalavimus, kurie patikrintų komponento veikimą esant pvz. ribinėms reikšmėms. Komponento testavimas atliekamas tikrinant jo elgseną esant paruoštiems

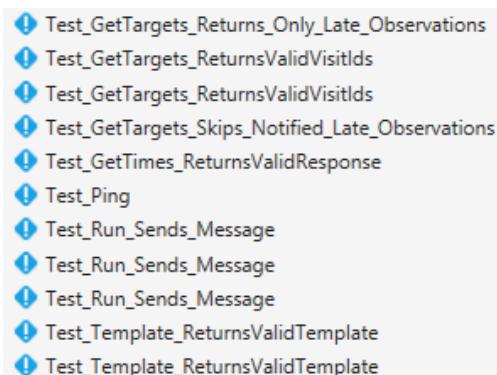
testavimo atvejams ir lyginant su elgsena, kurios tikimąsi. Programinio komponento klaida aptinkama, kai prie tam tikrų testavimo sąlygų komponento elgsena yra ne tokia, kokios tikimąsi.

Integravus atskirus modulių komponentus į didesnius modulius ar į visą sistemą, testavimas vykdomas „baltos dėžės“ principu. Tai taipogi dinaminis testavimo būdas skirtas aptiki loginėms klaidoms. Testiniai rinkiniai gaunami iš sistemos struktūros. Papildomų testinių atvejų nustatymui naudojamos žinios apie programą. „Baltos dėžės“ testavimo tikslas yra išbandyti visus įmanomus komponento skaičiavimo kelius, operatorius, įvykdyti ciklus ir vidines duomenų struktūras. „Baltos dėžės“ testavimas įvykdomas, kai patikrinamas visų nepriklausomų kelių veikimas.

Vartotojo sąsają buvo testuojama rankiniu būdu, išbandant visus sasajos valdymo elementus, patikrinant, ar nėra neveikiančių mygtukų, ar teisingai atvaizduojami duomenys ir rezultatai.

Sudarius testavimo planą lieka tik paruošti komponentams testavimo atvejus ir pagal sudarytą planą naudojant šiuos atvejus sukurtą sistemą ištstuoti. Sistemos modulių komponentų testavimas „juodos dėžės“ principu ir jų testavimo atvejų paruošimas vyko atsižvelgiant į sistemos modulių išsikeltus ir apibrėžtus funkcinius reikalavimus, kurie yra pateikiami 2.1, 2.2 ir 2.3 lentelėse. Kiekvieno modulio funkcinis reikalavimas atitinka atskirą testavimo atvejį, kuris turi patenkinti aprašytą sistemos modulio funkcionalumą. Patenkinus testavimo atvejo funkcionalumą ir įsitikinus, kad jis veikia korektiškai, testinis atvejis yra laikomas sėkmingu. Nefunkcinių reikalavimų būdavo prisilaikoma visų sistemos modulių kūrimo laikotarpiu. Jų testavimas atispindės ir bus vykdomas visos sistemos testavimo metu.

Sistemos modulių pranešimų siuntimo ir duomenų mainų su kitomis sistemomis ištavimui buvo rašomi vienetiniai testai. Naudojant „Visual Studio 2013 Ultimate“ programavimo aplinką ir joje esančius „Unit tests“ automatizuoto testavimo įrankius pagal atitinkamus testavimo atvejus parašomi testai. Testais patikrinamas įgyvendintas funkcionalumas, kodo logika ir veikimas, įsitikinama ar nėra jokių klaidas iššaukiančią konstrukciją. Parašytų testų sąrašas pateikiamas 3.26 paveiksle.



3.26 pav. Parašytų testų sąrašas

Aptarsime aprašytų testų testavimo paskirtį, pateiksime keletą jų pavyzdžių:

„RemoteObservationLateSubmissionMessageTest.cs“ klasė aprašo pranešimų išsiuntimo pacientams logiką dėl pavėluoto pateikti stebėjimo duomenų:

- „Test_GetTargets_Returns_Only_Late_Observations()“ – ar grąžinamas tinkamas šablonas.
- „Test_GetTargets_Skips_Notified_Late_Observations()“ – ar nesiunčiama išsiuistiems.
- „Test_Run_Sends_Message()“ – ar pranešimas yra išsiunčiamas.

„VisitEarlierTimeNotificationSendTest.cs“ klasė aprašo pranešimų išsiuntimo pacientams logiką dėl galimybės apsilankytį ankščiau pas gydytoją:

- „Test_Template_ReturnsValidTemplate()“ – ar grąžinamas tinkamas šablonas.
- „Test_GetTargets_ReturnsValidVisitIds()“ – ar gaunami tinkami apsilankymai.
- „Test_GetTimes_ReturnsValidResponse()“ – ar grąžinami
- „Test_Run_Sends_Message()“ – ar pranešimas yra išsiunčiamas.

,,VisitReminderMailSendTest.cs“ klasė aprašo pranešimų išsiuntimo pacientams logiką dėl informavimo apie artėjančią apsilankymą:

- „Test_Template_ReturnsValidTemplate()“ – ar grąžinamas tinkamas šablonas.
- „Test_GetTargets_ReturnsValidVisitIds()“ – ar gaunami tinkami apsilankymai.
- „Test_Run_Sends_Message()“ – ar pranešimas yra išsiunčiamas.

Visos klasės turi savo apsibrėžtus ir reikalingus netikrų (angl. „fake“) duomenų testinius rinkinius, kurių pavyzdys yra pateikiamas 3.27 paveiksle. Prieš testo inicijavimą yra sukuriami reikalingi sistemos komponentų, naudojamų servisų netikri atitikmenys. Jų pavyzdys yra pateiktas 3.28 paveiksle. Aprašomų testų logika yra atskiriama į tris savo tikslais išsiskiriančias dalis: pasiruošimą (angl. „Act“), vykdymą (angl. „Act“), palyginimą (angl. „Assert“).

```
private User CreatePatient(ISiteDataAccess dataAccess)
{
    var entity = new User
    {
        FullName = "Vardenis Pavardenis",
        FirstName = "Vardenis",
        LastName = "Pavardenis",
        PersonCode = null,
        CreateOn = DateTime.Today,
        FailedPasswordAttemptWindowStart = DateTime.Today,
        FailedPasswordAttemptWindowStart = DateTime.Today,
        LastActivityDate = DateTime.Today,
        LastLockoutDate = DateTime.Today,
        LastLoginDate = DateTime.Today,
        LastPasswordChangedDate = DateTime.Today,
        Email = DefaultPatientEmail,
        UserName = "varpa001",
        LanguageId = 1
    };
    dataAccess.Insert(entity);
    return entity;
}
```

3.27 pav. Testinių duomenų pavyzdys

```
private static void InitServices()
{
    var kernel = new StandardKernel();
    kernel.Bind().To<SiteDataAccess>();
    kernel.Bind().To<SiteLocalizationService>();
    kernel.Bind

```

3.28 pav. Testinių komponentų pavyzdys

Testo inicijavimo metu, yra įvykdoma aprašyto testo logika. Ji sukuria realią situaciją atspindinčią veiksmų seką, kuria yra patikrinamas norimas ištstuoti funkcionalumas. Šiuo atveju yra tikrinama, ar iš bendradarbiaujančios sistemos bus grąžinami korektiški laisvų apsilankymų duomenys. Išsiūsti duomenys – paciento gydytojo identifikacinis numeris, jo specialybės identifikacinis numeris, tuo metu buvusi data ir paciento apsilankymo laikas. Gauti duomenys – apsilankymų sąrašas ir vėliausio apsilankymo data. Testo logikos implementacija yra pateikiama 3.29 paveiksle.

```
public void Test_GetTimes_ReturnsValidResponse()
{
    using (ShimsContext.Create())
    {
        // Arrange
        var now = DateTime.Now;
        Epika.Core.Mvc.Security.Fakes.ShimSiteWorkContext.AllInstances.CurrentLanguageCodeGet = ctx => "lt-LT";

        Visit visit;
        using (var dataAccess = new SiteDataAccess())
        {
            DoVisitCleanup(dataAccess);

            var patient = CreatePatient(dataAccess);
            var doctor = CreateDoctor(dataAccess, ValidDoctorHisKey); // Pass valid doctor HisKey
            var specialty = CreateSpecialty(dataAccess, ValidSpecialtyHisKey); // Pass valid doctor specialty HisKey
            dataAccess.Save();

            visit = CreateVisit(dataAccess, visitStateRegisteredId, now.AddDays(7), patient, doctor, specialty, true);
            dataAccess.Save();

            CreateUserMailHistory(dataAccess, visit.Id, now.AddDays(-7));
            dataAccess.Save();
        }

        // Act
        var service = ServiceLocator.Resolve<VisitEarlierTimeNotificationSendService>();
        var response = service.GetTimeHandler(visit.DoctorId.GetValueOrDefault(), visit.SpecialtyId.GetValueOrDefault(),
            now, visit.Date);

        // Assert
        Assert.IsNotNull(response.LastTicketDate, "Nebuvo gautas vėliausias laisvas laikas.");
        Assert.IsNotNull(response.List, "Gautų laisvų laiku sąrašas tuččias.");
        Assert.IsTrue(response.List.Count() != 0, "Gautų laisvų laiku sąrašo skaičius yra lygus 0");
    }
}
```

3.29 pav. Testo logikos implementacijos pavyzdys

3.7. Sukurtos sistemos trūkumai, apribojimai bei tolimesnio plėtojimo galimybės

Pagal suprojektuotus komponentus sistema realizuota sukuriant ir išskiriant reikalingus projektus ir klasses, kurių reprezentacijos pateikiamas naudojant vizualią priemonę - sistemos projektų architektūros diagramą ir klasių esybių diagramas. Iš klasių diagramų matomi jų atributai, metodai, matomumas ir sąryšiai su kitomis klasėmis. Sistema buvo realizuota pagal Modelio-Vaizdo-Valdiklio loginius architektūros lygmenis, prie jų pritaikant ir kitus servisų ir vaizdų modelio sluoksnius. Sistemos duomenų bazės realizacija atitinka sistemos klasių esybių modelius iš kurių buvo generuojama duomenų bazės aprašo schema. Įgyvendintas „Sistemų tarpininkavimo lygmuo“, kuris yra atsakingas už duomenų siuntimą žinutėmis tarp mūsų sukurtos sistemos modulių, ir kitos „HIS“ medicininės informacinės sistemos. Modiliuose buvo pritaikytas sisteminių pranešimų, elektroninių laiškų siuntimo funkcionalumas.

Sistemos moduliai buvo ištестuoti naudojant pasirinktą testavimo planą. Testuojami atskiri komponentai „juodos dėžės“ principu, tada testuojama komponentų integracija ir visa sistema, naudojant „baltos dėžės“ testavimo metodiką. Tam, kad sistema būtų ištestuota pagal numatyta planą, pranešimų siuntimo komponentams sudaryti testiniai atvejai. Šie testiniai atvejai realizuoti naudojant „Visual Studio“ automatizuoto testavimo priemones. Išttestavus pašalintos klaidos ir įsitikinta, jog sukurti sistemos moduliai veikia patikimai ir stabiliai.

Sistema buvo kuriama laikantis numatytių funkcinių ir nefunkcinių reikalavimų. To pasekmė, kad sistemą tobulinti bus ganėtinai lengva ir patogu, nes sukurti sistemos moduliai ir jų implementacija numatytoje sistemos loginėje architektūroje atskiria kiekvieno jų funkcionalumą ir paskirtį ir jie tarpusavyje netrukdo vienas kitam. Dėl to sukurtų modulių suteikiamas galimybes bus lengva panaudoti kitose, ateityje galbūt kuriamose paslaugose, o dabartinius sukurtus modulius bus lengva tobulinti ir taisyti jų tobulinimo metu iškilusias problemas.

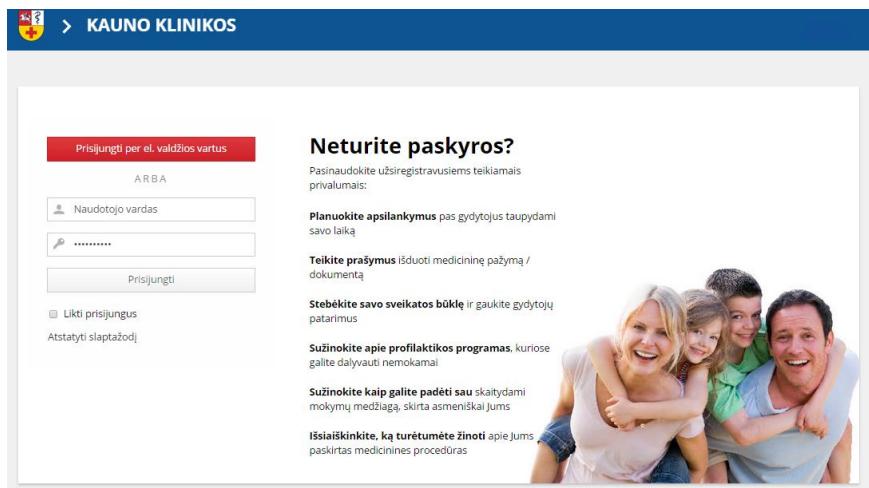
4. DOKUMENTACIJA NAUDOTOJUI

4.1. Diegimo vadovas

Kadangi bakalaurinio darbo tikslas buvo sukurti tris medicininės informacinių sveikatos priežiūros paslaugų sistemos modulius (nuotolinių stebėjimų, apsilankymų, švietimo), diegimo procesas ir galutinis jo rezultatas turės tik su šiais moduliais susijusias sistemos dalis. Diegimo procesas turi du etapus:

1. Serverinėje dalyje instaliuojama „MS SQL Server 2014“ programa. Paleidus programą, prisijungiant prie serverio savo kompiuterio lokalais duomenimis. Tada šioje programoje yra atnaujinama nurodoma duomenų bazė („Object Explorer-Databases-Restore Database...-Device-Nurodyti *.bak failą. Taip sukuriamos lentelės, nurodoma jų struktūra ir įrašomos saugomos procedūros.
2. Klientinėje dalyje instaliuojama „Visual Studio 2013“ programa. Jos pagalba bus galima peržiūrėti norimus sistemos modulių failus.

Kitas, lengvesnis būdas išbandyti sukurtus sistemos modulius - prisijungti prie pačios sistemos <http://portalas.kaunoklinikos.lt/> adresu. Prisijungimo procese įgysite paciento teises, galēsite pasinaudoti jums tuo metu prieinamomis paslaugomis. 4.1 paveiksle yra pateikiamos sistemos pradžios prisijungimo puslapis.



4.1 pav. Sistemos prisijungimo pradžios puslapis

4.2. Naudotojo vadovas

Atsižvelgdami į naudotojų grupes sudarančių vartotojų kiekius, kurie naudosis sukurtaja informacine sistema, vienos didžiausių iš jų bus pacientų ir gydytojų teises turinčios naudotojų grupės. Šios grupės taip pat geriausiai atspindi sukurtų paslaugų modulių prasmę, naudą ir teikiamą funkcionalumą. Aprašysime šioms grupėms naudotojo vadovą. Naudotojai turi prisijungti prie sistemos adresu <http://portalas.kaunoklinikos.lt/>.

4.2.1. Pacientų naudotojų grupė

Nuotolinių stebėjimų modulis. Modulis pasiekiamas per kairinėje meniu juostoje esančią nuorodą „Nuotoliniai stebėjimai“, kuris yra pavaizduotas 4.2 paveiksle. Paspaudus nuorodą, pacientas yra nukeliamas į savo nuotolinių stebėjimų sąrašo puslapį. Šiame puslapyje galima peržvelgti stebėjimų pavadinimo, administruojamo gydytojo, priklausančių įrenginių, pateikimo datos informaciją. Galima filtruoti duomenis pagal visus išvardintus kriterijus. Iš šio puslapio galime patekti į savo nuotolinių stebėjimų kūrimo puslapį, paspaudus „Sukurti stebėjimą“ mygtuką, į pasirinkto

nuotolinio stebėjimo peržiūros puslapį, paspaudus ant pasirinkto nuotolinio stebėjimo įrašo ir pateikti stebėjimų duomenų failus, jeigu sąraše yra atitinkamo nuotolinio stebėjimo tipas su duomenų kaupimu („Įkelti duomenis“). Galima pasirinkti tarp keletos stebėjimų skirtukų – „Vykdomi“ ir „Ivykdyti“.

4.2 pav. Paciento nuotolinį stebėjimų sąrašo puslapis

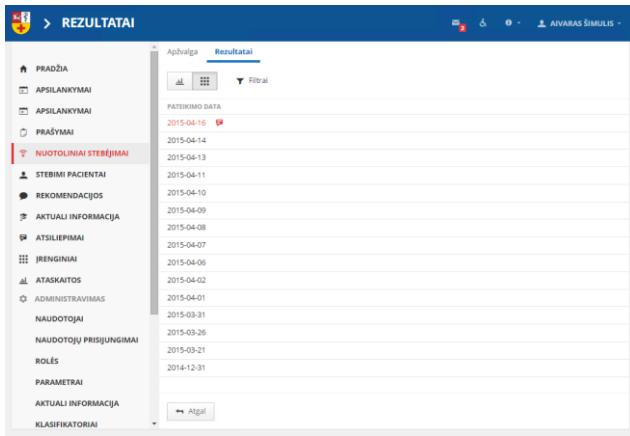
Patekus į naujo stebėjimo sukūrimo puslapį, pavaizduotą 4.3 paveiksle, yra suteikiama galimybė pasirinkti vieną iš pateikiamų susikurti stebėjimų, įvesti stebėjimo pradžios datą.

4.3 pav. Paciento nuotolinio stebėjimo sukūrimo puslapis

Patekus į stebėjimo peržiūros puslapį, kuris yra pavaizduotas 4.4 paveiksle, tame skirtukų pagalba gali peržvelgti bazinius stebėjimo duomenis ir pateikiamų duomenų rezultatus. Kiek žemiau yra pateikiami stebėjimo rekomendacinių pranešimai, naujausių pateiktų duomenų sąrašas, stebėjimo laikotarpis, administruojantis gydytojas ir kita informacija. Puslapyje galime įvesti, peržiūrėti pateiktus duomenis, peržvelgti gautus rezultatus.

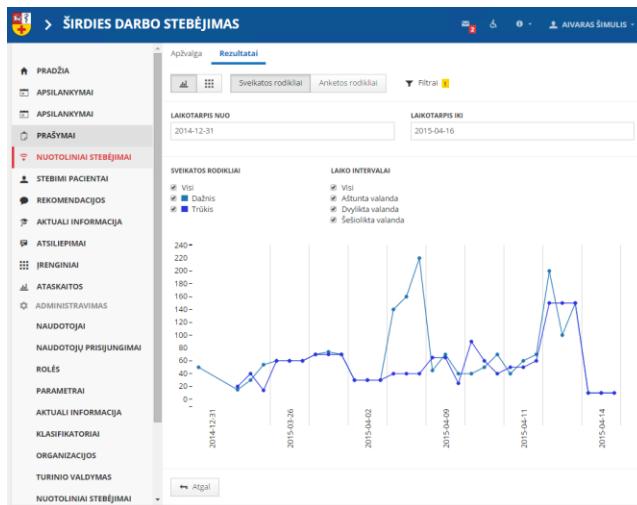
4.4 pav. Paciento nuotolinio stebėjimo peržiūros puslapis

Rezultatų skirtuke, kuris yra pavaizduotas 4.5 paveiksle, galime peržvelgti visų stebėjime pateiktų duomenų sąrašą, juos filtruoti. Iš šio puslapio galime nukeliauti į stebėjimo rezultatų atvaizdavimo puslapį grafiniu pavidalu.



4.5 pav. Paciento nuotolinio stebėjimo rezultatų puslapis

Rezultatų diagramų apžvalgos puslapyje, kuris yra pavaizduotas 4.6 paveiksle, yra pateikiamas detalus stebėjimo grafikas. Grafikas kinta priklausomai nuo pasirinktų atvaizduoti rodiklių, laiko intervalų. Užvedus ant vieno iš taškų galima pamatyti tos dienos įvesto rodiklio duomenis. Grafiko tipas gali būti stulpelinis arba linijinis, priklausomai nuo to, kokį tipą pasirinko stebėjimą administruojantys gydytojas. Vienu metu galima peržiūrėti sveikatos rodiklių arba anketos rodiklių grafike pateikiamus duomenis.



4.6 pav. Paciento nuotolinio stebėjimo rezultatų grafinio atvaizdavimo puslapis

Duomenų pateikimo puslapyje, kuris yra pavaizduotas 4.7 paveiksle, yra pateikiamas stebėjimo turimų rodiklių formos puslapis. Pateikiamus duomenis nurodo stebėjimą administruojantys gydytojas. Išskiriame du pagrindiniai matavimo duomenų pateikimo tipai: sveikatos rodikliai – pateikiami lentelėje ir anketos rodikliai – pateikiami vertikalios formos pavidalu. Priklasomai nuo paciento įvedamų reikšmių, patvirtinus duomenų įvedimą yra atliekama išsami duomenų analizė ir pacientui, jo stebėjimo peržiūros lange yra pateikiami gauti analizės rezultatai. Duomenis galima pateikti bet kuriai pasirinktinai datai nuo stebėjimo vykdymo pradžios iki dabartinės datos. Siūloma įvedinėti korektiškus ir savo matavimo paskirtį atitinkančius duomenis.

The screenshot shows a web-based medical application interface. At the top, there's a header with the logo of the Ministry of Health of the Republic of Lithuania and the title 'ŠIRDIES DARBO STEBĖJIMAS'. On the left, a vertical sidebar menu lists various medical modules: PRADŽIA, APSILANKYMAI, PRAŠYMAI, NUOTOLINIAI STEBĖJIMAI (highlighted in red), STEBIMI PACIENTAI, REKOMENDACIJOS, AKTUALI INFORMACIJA, ATSILIEPIMAI, IRENGINIAI, ATASKAITOS, ADMINISTRAVIMAS, NAUDOTÖJAI, NAUDOTÖJU PRISIJUNGIMAI, ROLËS, PARAMETRAI, AKTUALI INFORMACIJA, and KLASIFIKATORIAI.

The main content area contains a form for entering monitoring data. It includes fields for 'IVESTI DUOMENIS UŽ' (Date: 2015-04-16) and sections for 'LAIKAS', 'DAŽNIS', and 'TRŪKIS'. Below these are fields for 'AŠTUNTA VALANDA', 'DIVLYTKITA VALANDA', and 'ŠEŠIOLIKTA VALANDA'. On the left, there are input fields for 'Pulsas', 'Tankumas', 'Kietumas', and 'Tvirtumas'. At the bottom right are buttons for 'Atgal' (Back) and 'Saugoti' (Save).

4.7 pav. Paciento nuotolinio stebėjimo duomenų pateikimo puslapis

Apsilankymų modulis. Modulis pasiekiamas per kairinėje meniu juosteje esančią nuorodą „Apsilankymai“. Paspaudus nuorodą, pacientas yra nukeliamas į savo apsilankymų sąrašo puslapį kuris yra pavaizduotas 4.8 paveiksle. Šiame puslapyje galima peržvelgti apsilankymų datą, gydytoją, būseną. Galima filtruoti duomenis pagal visus išvardintus kriterijus. Iš šio puslapio galime pateikti į apsilankymo užsiregistravimo puslapį, kuris yra pavaizduotas 4.9 paveiksle, paspaudus mygtuką „Užsiregistruti“, į apsilankymo patvirtinimo puslapį, kuris yra pavaizduotas 4.10 paveiksle, paspaudus "Tęsti" mygtuką, į apsilankymo detalesnės informacijos peržiūros puslapį kuris yra pavaizduotas 4.11 paveiksle, paspaudus ant vieno iš sąrašo apsilankymų. Skirtukų pagalba galima peržvelgti planuojamus, įvykusius/atšauktus, gydytojų, registratorių rekomenduojamus apsilankymus.

This screenshot shows a list of scheduled appointments. The table has three columns: 'DATA' (Date: 2015-04-14 08:30), 'GYDYTOJAS' (Doctor: J. SIUDIKIENE, Endodontologas), and 'BŪSENA' (Status: patvirtinta). Above the table, there are buttons for 'Planuojami', 'Įvykė/Atšaukti' (selected), and 'Rekomenduojami'. There are also buttons for 'Užsiregistruoti' (Schedule), 'Filtrai' (Filters), and 'Pagalba' (Help). The sidebar on the left is identical to the one in figure 4.7.

4.8 pav. Paciento apsilankymų sąrašo puslapis

This screenshot shows the process of creating a new appointment. On the left, a dropdown menu lists doctors by specialty: Akusėris gine (Administratorius), Dermatovenerologas (ADRIJUS KRIMELIS), Endodontologas (ADRIUS GAURILČIKAS), Genetikas (AGNÉ STOLYGAIITÉ), Hematologas (AIDA KUNIGELIENĖ), Infekologas, Kardiologas, Kraujagyslių chirurgas, Krūtinės chirurgas, and Nefrologas. The selected doctor is ADRIJUS KRIMELIS. To the right, a calendar for 'Balandis 2015' shows dates from 30 to 31. Below the calendar, a time selection dropdown shows '08:15', '08:30', and '08:45'. A 'Tęsti' (Continue) button is at the bottom right.

4.9 pav. Paciento naujo apsilankymo sukūrimo puslapis

4.10 pav. Paciento naujo apsilankymo patvirtinimo puslapis

4.11 pav. Paciento apsilankymo peržiūros puslapis

Mokymų modulis. Modulis pacientui nėra atskirai išskiriamas meniu juostos nuoroda. Jo teikiamomis paslaugomis galima pasinaudoti daugelyje puslapyje teikiamų paslaugų arba kitų modulių informacijos pateikimo formose, pvz.: užsiregistrnuojant atitinkamam apsilankymui pas konkretios specialybės gydytoją, sistema parenka būtent vizito tikslui skirtą paruošiamają medžiagą, su kuria pacientas prieš atvykdamas pas gydytoją turėtų susipažinti. Nuotolinio stebėjimuose gauti analizės rezultatai gali rekomenduoti peržiūrėti atrinktą mokymo medžiagą. Prieš atliekant stebėjimą gali būti siūloma peržvelgti reikiama mokymo medžiagą, kitu atveju to nuotolinio stebėjimo nebus įmanoma atlikti. Nuorodos nukreipia vartotojus į atskirą, kito sistemos modulio („Moodle“) aplinką.

4.2.2. Gydytojų naudotojų grupė

Šią naudotojų grupę sudaro šias roles turintys vartotojai - atsakingas gydytojas, atsakingas specialistas, KK atsakingas darbuotojas. Jų aptartos teisės skiriasi nežymiai, todėl jų naudotojų vadovų dėl vengiamo pateikti per didelį informacijos kiekį ir vis pasikartojančią informaciją atskirai neišskirsime.

Nuotolinių stebėjimų modulis. Visų pirma yra siūloma peržvelgti pacientų naudotojų grupės naudotojo vadovą. Siekdami nekartoti jau pateiktos informacijos ir tiek pacientams, tiek gydytojams

prieinamų ir galimų atlikti tą pačią funkciją, aptarsime ką šiame modulyje gali atlikti tik gydytojų naudotojų grupės vartotojai.

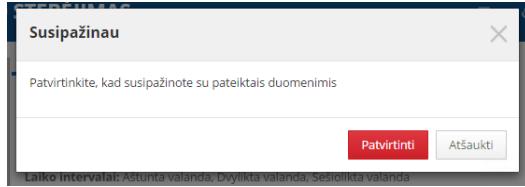
Modulis pasiekiamas per kairinėje meniu juosteje esančią nuorodą „Stebimi pacientai“. Yra patenkama į gydytojo nuotolinių stebėjimų sąrašo puslapi. Jo išvaizda yra adekvati pacientų nuotolinių stebėjimų sąrašo puslapiui. Gydytojas šiame puslapyje gali sukurti stebėjimą pacientui, peržvelgti savo ir kitų gydytojų stebėjimus, filtruoti duomenis.

Nuotolinio stebėjimo sukūrimo puslapyje, kuris yra pavaizduotas 4.12 paveiksle, yra pateikiama kur kas daugiau nustatymų turinti forma, negu pacientų puslapyje. Suteikiama galimybė nurodys norimą pacientą, stebėjimo tipą, stebėjimą, kliniką, gydytojo specialybę, gydytoją, stebėjimo intervalą, rekomendaciją pacientui, priskirti slaugytoją ir stebėjimo įrenginius.

4.12 pav. Gydytojo naujo nuotolinio stebėjimo sukūrimo puslapis

Gydytojų peržiūros puslapis, kuris yra pavaizduotas 4.13 paveiksle, išsiskiria tuo, kad Jame yra suteikiama galimybė peržvelgti gydytojų tarpusavio konsultacijų sąrašą, istoriją, galima redaguoti esamą stebėjimą, informuoti pacientą, kad su jo pateikta informacija buvo susipažinta, sukurti pacientui rekomendaciją – puslapis yra pavaizduotas 4.15 paveiksle ir konsultuotis su kitu gydytoju – puslapis yra pavaizduotas 4.16 paveiksle.

4.13 pav. Gydytojo nuotolinio stebėjimo peržiūros puslapis



4.14 pav. Gydytojo susipažinimas su paciento pateiktais stebėjimo duomenimis

4.15 pav. Gydytojo rekomendacijos sukūrimas pacientui

4.16 pav. Gydytojo konsultacijos sukūrimas kitam gydytojui

Gydytojo kuriami pranešimai yra išsiunciāmi į sistemos pranešimų dėžutę, taip pat yra atvaizduojami jiems skirtiems atskiruose moduliuose bei pačių nuotolinių stebėjimų informacijos peržvelgimo anketose.

Apsilankymų modulis. Modulis pasiekiamas per kairinėje meniu juostoje esančią nuorodą „Apsilankymai“. Paspaudus nuorodą, gydytojas yra nukeliamas į savo apsilankymų sąrašo puslapį. Jo išvaizda yra adekvati pacientų apsilankymų sąrašo puslapiui. Šiame puslapyje galima peržvelgti apsilankymų datą, pacientą, būseną. Galima filtruoti duomenis pagal visus išvardintus kriterijus. Iš šio

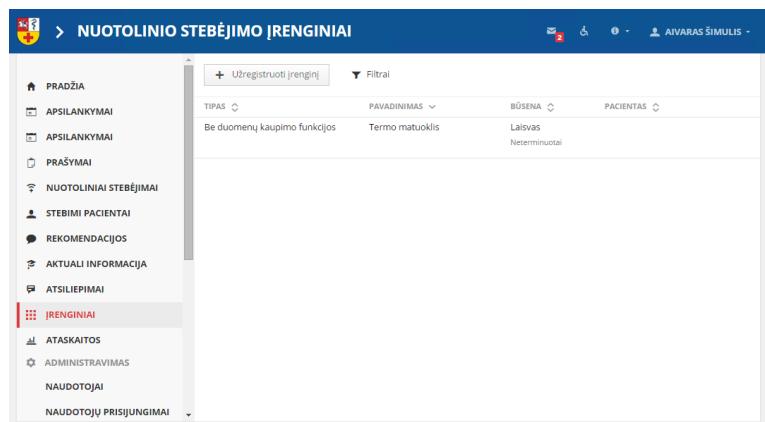
puslapio galime pateikti į apsilankymo užsiregistravimo puslapį pacientui („Registruoti“), į apsilankymo detalesnės informacijos peržiūros puslapį (paspaudus ant vieno iš sąrašo apsilankymų).

Gydytojų apsilankymų funkcijos nuo pacientų skiriasi toliau pateikiamais aspektais. Apsilankymų kūrimo lange galima pasirinkti pacientą, kuriam yra sukuriamas apsilankymas. Tik gydytojui yra prieinamas informavimo puslapis, kuriamo yra pateikiamas pacientų sąrašas, kurie yra užsiregistravę apsilankymams. Galima pasirinkti norimus pacientus ir juos informuoti apie kitas apsilankymų galimybes. Apsilankymų peržiūros pusliai yra identiški.

Mokymų modulis. Modulis gydytojų naudotojų grupei nėra aktualus. Jis yra skirtas sistemos pacientams.

4.3. Administravimo vadovas

Įrenginių administratoriui. Modulis pasiekiamas per kairinėje meniu juostoje esančią nuorodą „Įrenginiai“. Paspaudus nuorodą, administratorius yra nukeliamas į sistemoje, nuotoliniose stebėjimuose naudojamų įrenginių sąrašo puslapį, kuris yra pavaizduotas 4.17 paveiksle. Šiame puslapyje galima peržvelgti įrenginių tipą, pavadinimą, būseną, įrenginį naudojantį pacientą. Galima filtruoti duomenis pagal visus išvardintus kriterijus. Iš šio puslapio galime patekti į įrenginio užsiregistravimo puslapį pacientui („Užregistruoti įrenginį“), į įrenginio detalesnės informacijos peržiūros puslapį (paspaudus ant vieno iš sąraše esančių įrenginių).



4.17 pav. Įrenginių sąrašo puslapis

Patekus į naujo įrenginio sukūrimo puslapį, kuris yra pavaizduotas 4.18 paveiksle, yra suteikiamā galimybė suteikti naujam įrenginiui pavadinimą, tipą, modulį, rūšį, numerį, vertę, registracijos datą, planuojamą patikros datą, kliniką.

4.18 pav. Naujo įrenginio sukūrimo puslapis

Patekus į įrenginio peržiūros puslapį, kuris yra pavaizduotas 4.19 paveiksle, Jame skirtukų pagalba galima peržvelgti bazinius įrenginio duomenis, veiksmų istoriją, pranešimų apie gedimą sąrašą. Kiek žemiau yra pateikiami visi galimi atlkti veiksmai su atitinkamoje būsenoje esančiu įrenginiu: išduoti, remontuoti, įvykdyti patikrą, nustatyti nenaudojamu. Atlirkus su įrenginiu vieną iš nurodytų veiksmų, jo būsena pasikeičia, tuo pačiu ir galimas atlkti veiksmų su įrenginiu kiekis ir įvairumas.

4.19 pav. Įrenginio peržiūros puslapis

Mokymų administratoriui. Modulis pasiekiamas per kairinėje meniu juostoje esančią administruavimo skiltį, paspaudus nuorodą „Aktuali informacija“. Šiame puslapyje, kuris yra pavaizduotas 4.20 paveiksle, yra pateikiamas visų galimų mokymų sąrašas, kuris yra gaunamas iš kito sistemas modilio, esančio „Moodle“ aplinkoje. Šiame puslapyje galima peržvelgti mokymų pavadinimus, kategorijas, atnaujinimo datą. Galima filtruoti duomenis pagal visus išvardintus kriterijus. Mokymų sąrašą galima atnaujinti „Atnaujinti“ mygtuko pagalba. Paspaudus ant vieno iš mokymo sąrašo įrašų, patenkama į mokymo redagavimo puslapį.

4.20 pav. Mokymų sąrašo peržiūros puslapis

Mokymo redagavimo puslapyje, kuris yra pavaizduotas 4.21 paveiksle, yra pateikiami visi galimi konfigūruoti mokymų medžiagos kriterijai, jų bendra informacija. Atsiliepimų skirtuke pateikiami pacientų atsiliepimai apie mokymų medžiagą. Iš šios informacijos administratoriai gali spręsti apie medžiagos tobulinimo, keitimo galimybes „Moodle“ sistemoje. Mokymų medžiagos sukūrimas vyksta kito modilio „Moodle“ aplinkoje.

The screenshot shows a web-based application titled 'DOKUMENTAS'. At the top, there is a navigation bar with icons for email, search, and user profile ('EXAMPLE USER'). Below the header, a sidebar on the left lists various categories: PRADŽIA, APSILANKYMAI, APSILANKYMAI, PRAŠYMAI, NUOTOLINIAI STEBĖJIMAI, STEBIMI PACIENTAI, KONSULTACIJOS, REKOMENDACIJOS, AKTUALI INFORMACIJA, ATSILIEPIMAI, IRENGINIAI, ATASKAITOS, ADMINISTRAVIMAS, NAUDOTAJI, NAUDOTOU PRISJUNGIMAI, ROLES, PARAMETRAI, and AKTUALI INFORMACIJA (highlighted with a red border). The main content area is titled 'Kursas' and 'Atsiliepimai'. It displays course details: PASKAITOS TRUMPINYS (DOKAS), KURSO KATEGORIJA (MOKOMOJI MEDŽIAGA), KURSO KÜREJAS (Example User), KURSO NUORODA (http://moodle.bbc.local/course/view.php?id=90), and MOKYMU PAKETO SANTRAUKA. Below this, sections for 'DIAGNOZĖS IR DIAGNOZIŲ GRUPĖS (TLK-10-AM)', 'RIZIKOS GRUPĖS', 'APSILANKYMO TIPOS' (with tabs for Ambulatorinis, Stacionarinis, Skubi pagalba), 'NUOTOLINIO STEBĖJIMO IRENGINIŲ MODELIAI', 'PRAŠYMAI', and 'NUOTOLINIAI STEBĖJIMAI' are shown. At the bottom right are 'Atgal' and 'Saugoti' buttons.

4.21 pav. Mokymo redagavimo puslapis

5. REZULTATŪ APIBENDRINIMAS IR IŠVADOS

1. Atlikus medicininių informacinių sistemų analizę paaškėjo, jog dauguma jų, skirtingai nuo kuriamos Kauno klinikų medicininės informacinės sistemos, neturi nuotolinių stebėjimų funkcijos, jos nėra orientuotos į individualų vartotoją, pagrindinės teikiamas paslaugos yra statinio informacinio pobūdžio. Išanalizuotos sistemos siūlo daug standartiskų funkcijų, tokų, kaip užsiregistravimas apsilankymui, informacinių straipsnių ir naujienu paskelbimas, tačiau tai neatitinka užsakovo keliamų reikalavimų, kadangi nėra reikiamų nuotolinių stebėjimų ir mokymo funkcijų, o papildomų nereikalingų funkcijų yra per daug.
2. AtlIkta sistemos kūrimo programinių įrankių, technologijų ir protokolų analizė parodė, kad jas renkantis egzistuoja daugelis alternatyvų. Atlirkus analizę buvo įsitikinta, kad tik gerai apsibrėžti analizės kriterijai, pagal kuriuos lyginamos skirtinges technologijos ar programiniai įrankiai, gali padėti išsirinkti labiausiai tinkantį sprendimą sistemos kūrimui. Buvo pastebėta, kad didėjant pasirenkamų ir naudojamų įrankių ir technologijų kiekiui, suderinamumo tarp jų įtaka didėja, todėl analizuojamų technologijų pasirinkimas atitinkamu kiekiu sumažėja. Analizės metu įgautos žinios apie kiekvieną įrankį, leido labiau suprasti savo įgyvendinamų modulių realizavimo galimybes, galimas kliūtis ir kaip jų bus galima išvengti pritaikant skirtinges įrankius ir technologijas.
3. Sistemos specifikacijoje išskirti aktoriai ir jų roles padėjo lengviau suvokti kiekvieno aktoriaus sąveiką su skirtingais sistemos moduliais. Detalizuojant kiekvieno apsibrėžto sistemos aktoriaus panaudos atvejus sudaryti funkciniai sistemos reikalavimai, padėjo išliaiskinti kokio funkcionalumo reikės numatomų modulių įgyvendinimui. Nefunkciniai reikalavimai parodė bendrus sistemos veikimo požymius, pvz.: našumą, patikimumą, o tai leido iš anksto numatyti bendrą galimos sistemos veikimo ir realizacijos vaizdą.
4. Atlirkus analizę ir apsibrėžus specifikaciją buvo suprojektuoti sistemos moduliai, kurių metu buvo patobulinti įgūdžiai naudotis įvairiais projektavimo įrankiais pvz.: „draw.io“, „Microsoft Visio“. Suprojektuota sistemos loginė architektūra, apibrėžta ir veiklos diagramomis pavaizduota sistemos modulių veiklos logika, sistemos modulių modelių dalykinės srities esybių klasės, vartotojo sąsajos modeliai ir navigacijos planai, funkcijų hierarchijos diagramos. Projektavimo etape panaudoti ganėtinai nauji architektūriniai sprendimai, pvz.: „MVC“ modelis, realizavimo etape buvo labai patikimi ir lankstūs bei nesukėlė papildomų problemų, tokų kaip reikiamais informacijos trūkumas internete.
5. Realizacijoje buvo sukurti sistemos moduliai, pritaikyta projektavimo etape numatyta loginė architektūra. Sistemos duomenų bazės realizacija atitinko suprojektuotus sistemos klasų esybių modelius. Iš jų buvo sugeneruota duomenų bazės aprašo schema, o tai leido suraupyti svarbaus programavimo laiko ir nesukėlė papildomos galimų atsirasti klaidų grėsmės. Sékminga sistemos modulių realizacija leido įsitikinti ir patvirtinti, kad analizės metu pasirinktos technologijos, įrankiai bei projektavimo etape pasirinktos įvairios metodologijos ir projektavimo sprendimai buvo teisingi, gerai įvertinti ir apgalvoti. Pavaizduoti ir aprašyti įdomesni sistemos realizacijos atvejai, padėjo įgauti vertingos darbo patirties. Įgyvendinti realūs vartotojo sąsajos vaizdai turi mažų nukrypimų nuo specifikacijos. Atliekant sistemos modulių testavimą paaškėjo, jog testuojant labai svarbus yra testavimo planas, kuris ne kartą padėjo atskleisti neišsiunčiamų pranešimų, negaunamų bendradarbiavimo duomenų tarp HIS sistemos problemas. Atlirkas galutinis testavimas parodė, kad informacinės sistemos moduliai veikia patikimai, todėl ateityje informacinės sistemos modulių funkcionalumą bus galima išplėsti, o gera dabartinė modulių realizacija padės išvengti daugiau klaidų galimuose tolimesniuose sistemos kūrimo etapuose.
6. Paruošta modulių dokumentacija svarbiausioms sistemos aktorių, naudotojų grupėms, leidžia ir suteikia galimybę susipažinti su sistemos modulių naudojimo, įdiegimo ir valdymo procesais. Dokumentacijos kūrime atsižvelgta į projektavimo etapu metu sudarytą vartotojo sąsajos ir navigacijos planą bei pagal ją realizuotų galutinių sistemos modulių įgyvendinimą. Pateikti kiekvieno sukurto sistemos modulių lango vaizdai, jų aprašai, navigacijos paaiškinimai. Aprašyti kiekvienai vartotojų grupei galimos atlirkti funkcijos, jų reikalaujami pradiniai duomenys ir iš jų gaunami rezultatai.

6. LITERATŪRA

- [1] Clip Art Lord, „Free cartoon doctor clip art“ 2015. [Tinkle]. Available: <http://www.clipartlord.com/free-cartoon-doctor-clip-art-4/> [Kreiptasi 05 kovo 2015].
- [2] Clip Arts, „Student using computer clip art“ 2015. [Tinkle]. Available: <http://cliparts.co/clipart/2304594> [Kreiptasi 05 kovo 2015].
- [3] „Our Vision“, 2014. [Tinkle]. Available: <http://www.amidus.lt/about/our-vision/>. [Kreiptasi 11 kovo 2015].
- [4] „Naudojimusi puslapiu taisykłės“ 2010. [Tinkle]. Available: <http://www.pasveik.lt/lt/v/kontaktai/>. [Kreiptasi 12 kovo 2015]
- [5] „e medicina“, 2006. [Tinkle]. Available: <http://www.emedicina.lt/index.php>. [Kreiptasi 12 kovo 2015].
- [6] „Išankstinės pacientų registracijos sistema“. 2015 [Tinkle]. Available: http://sergu.lt/apie_projekta_lt.asp. [Kreiptasi 13 kovo 2015].
- [7] „Lietuvos administracinis suskirstymas“. [Tinkle]. Available: http://lt.wikipedia.org/wiki/Lietuvos_administracinis_suskirstymas. [Kreiptasi 13 kovo 2015].
- [8] „Santariškių klinikos“. [Tinkle]. Available: <http://santa.lt/index.php>. [Kreiptasi 13 kovo 2015].
- [9] „Internetinė paciento kortelė“. [Tinkle]. Available: <https://viva.santa.lt/ikp/apie.asp>. [Kreiptasi 13 kovo 2015].
- [10] „Kauno klinikos laukia pacientų“. [Tinkle]. Available: <http://www.kaunoklinikos.lt/apie.php>. [Kreiptasi 13 kovo 2015].
- [11] Greg R. Jacobs, „Top 5 integrated development environments“, 24 rugpjūčio 2010. [Tinkle]. Available: <http://hackaday.com/2010/08/24/top-5-integrated-development-environments/>. [Kreiptasi 14 kovo 2015].
- [12] „Visual Studio Ultimate su MSDN“, 2015. [Tinkle]. Available: http://www.microsoftstore.com/store/mseea1/lt_LT/pdp/%E2%80%9EVisual-Studio-Ultimate-2013%E2%80%9C-su-MSDN/productID.288483800. [Kreiptasi 14 kovo 2015].
- [13] „SQL Server Editions“, 2015. [Tinkle]. Available: <http://www.microsoft.com/en-us/server-cloud/products/sql-server-editions/>. [Kreiptasi 14 kovo 2015].
- [14] „HTML“, 07 kovo 2015. [Tinkle]. Available: <http://lt.wikipedia.org/wiki/HTML>. [Kreiptasi 14 kovo 2015].
- [15] „C Sharp programming language“, 10 vasario 2015. [Tinkle]. Available: [http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)). [Kreiptasi 14 kovo 2015].
- [16] „Smtp client class“, 2015. [Tinkle]. Available: [https://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient(v=vs.110).aspx). [Kreiptasi 14 kovo 2015].
- [17] „OSI modelis“, 07 kovo 2015. [Tinkle]. Available: http://lt.wikipedia.org/wiki/OSI_modelis. [Kreiptasi 18 kovo 2015].
- [18] „MVC Architecture“, 2015. [Tinkle]. Available: https://developer.chrome.com/apps/app_frameworks. [Kreiptasi 04 balandžio 2015].
- [19] „Model-view-controller“, 2015. [Tinkle]. Available: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Kreiptasi 04 balandžio 2015].
- [20] I.Sommerville, „Architektūros projektavimas“, 24 spalio 2010. [Tinkle]. Available: http://www.techmat.vgtu.lt/konspektai/PSI/psi_13_sistemos_architektura.pdf. [Kreiptasi 04 balandžio 2015].
- [21] Microsoft, „Solutions and Projects“, 2015. [Tinkle]. Available: <https://msdn.microsoft.com/en-us/library/b142f8e7.aspx>. [Kreiptasi 08 balandžio 2015].
- [22] Data Developer Center, „Code First to a New Database“, 2015. [Tinkle]. Available: <https://msdn.microsoft.com/en-us/data/jj193542.aspx>. [Kreiptasi 09 balandžio 2015].

- [23] „Use case“, [Tinkle]. Available: http://en.wikipedia.org/wiki/Use_case. [Kreiptasi 17 balandžio 2015].
- [24] „Functional requirement“, [Tinkle]. Available: http://en.wikipedia.org/wiki/Functional_requirement. [Kreiptasi 17 Balandžio 2015].
- [25] „Non-functional requirement“, [Tinkle]. Available: http://en.wikipedia.org/wiki/Non-functional_requirement. [Kreiptasi 17 Balandžio 2015].
- [26] „UML Components diagram“, [Tinkle]. Available: <http://www.ibm.com/developerworks/rational/library/dec04/bell/>. [Kreiptasi 17 balandžio 2015].
- [27] „UML Sequence diagram“, [Tinkle]. Available: <http://www.ibm.com/developerworks/rational/library/3101.html>. [Kreiptasi 17 Balandžio 2015].
- [28] „Entity-relationship model“, [Tinkle]. Available: http://en.wikipedia.org/wiki/Entity%20relationship_model. [Kreiptasi 24 Balandžio 2015].
- [29] „UML class diagram“, [Tinkle]. Available: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/> . [Kreiptasi 24 Balandžio 2015].
- [30] „Testavimo planas“, [Tinkle]. Available: http://proin.ktu.lt/wiki/doku.php?id=t120b111:ld1:ld1_-_testavimo_planas. [Kreiptasi 25 Balandžio 2015]
- [31] „Software test methods“, [Tinkle]. Available: http://www.tutorialspoint.com/software_testing/testing_methods.htm. [Kreiptasi 28 Balandžio 2015]
- [32] „Integration testing“, [Tinkle]. Available: http://en.wikipedia.org/wiki/Integration_testing. [Kreiptasi 28 Balandžio 2015].