



**Kauno technologijos universitetas**

Informatikos fakultetas

**Žiniomis grindžiamas kompiuterizuotas naudotojo  
reikalavimų specifikacijos sudarymo metodas**

Baigiamasis magistro projektas

---

**Dovilė Gedminaitė**

Projekto autorė

**Prof. dr. Audrius Lopata**

Vadovas

---

**Kaunas, 2020**



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Žiniomis grindžiamas kompiuterizuotas naudotojo reikalavimų specifikacijos sudarymo metodas**

Baigiamasis magistro projektas

Informacinių sistemų inžinerija (6211BX009)

---

**Dovilė Gedminaitė**

Projekto autorė

**Prof. dr. Audrius Lopata**

Vadovas

**Doc. dr. Lina Čeponienė**

Recenzentė

---

**Kaunas, 2020**



**Kauno technologijos universitetas**

Informatikos fakultetas

Dovilė Gedminaitė

## **Žiniomis grindžiamas kompiuterizuotas naudotojo reikalavimų specifikacijos sudarymo metodas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Dovilės Gedminaitės, baigiamasis projektas tema „Žiniomis grindžiamas kompiuterizuotas naudotojo reikalavimų specifikacijos sudarymo metodas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

---

(vardą ir pavardę įrašyti ranka)

---

(parašas)

Gedminaitė, Dovilė. Žiniomis grindžiamas kompiuterizuotas naudotojo reikalavimų specifikacijos sudarymo metodas. Magistro baigiamasis projektas / vadovas prof. dr. Audrius Lopata; Kauno technologijos universitetas, Informatikos fakultetas.

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

Reikšminiai žodžiai: Volere šablonas, Veiklos modelis, Reikalavimai, Specifikacija

Kaunas, 2020. 121 p.

### **Santrauka**

Reikalavimų specifikavimas ir analizė labai priklauso nuo analitiko žinių ir gebėjimų, dėl to tai didina reikalavimų specifikacijos sudarymo empiriškumo faktorių. Naudotojo reikalavimų specifikavimo procese tikslinga naudoti veiklos metamodelio pagrindu sukurtą žinių bazę, saugančią naudotojo reikalavimų specifikavimui būtinas dalykinės veiklos srities žinias (veiklos modelį), tuo padidinant reikalavimų specifikacijos sudarymo proceso efektyvumą pakartotinai panaudojant šias žinias UML diagramų generavimui ir esminių naudotojo reikalavimų specifikacijos sudarymui. Taip ne tik bus mažinamas empiriškumo faktorius ir klaidų tikimybė, bet ir taupomas analitikų laikas.

Norint įgyvendinti reikalavimų specifikacijos sudarymo metodą reikėjo atlikti programinės įrangos, kuri yra skirta specifikuoti reikalavimus, reikalavimų specifikavimo šablonų ir žiniomis grindžiamų sistemų bei jų veiklos modelio analizę. Bei buvo specifikuotas siūlomas metodas ir suprojektuota eksperimentinė sistema, kuria buvo ištestuotas metodo veikimas su realiais duomenimis.

Tyrimo metu, metode naudojamos reikalavimų aprašymo formos buvo palygintos su naudotojų reikalavimų specifikavimo standartų sudėtimi ir nustatyta, kad, taikant *Volere* šabloną galima padengti iki 97% naudotojo reikalavimų specifikacijos dokumento turinio. Taip pat naudojant aprašymo formose kuo daugiau laukų su apribotomis įvedimo reikšmėmis bei pasirinkimais galima sumažinti reikalavimų specifikacijos sudarymo empiriškumo faktorių.

Gedminaitė, Dovilė. Knowledge-based Computerized User Requirements Specification. Master's Final Degree Project / supervisor prof. dr. Audrius Lopata; Faculty of Informatics, Kaunas University of Technology.

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

Keywords: Volere Template, Enterprise Model, Requirements, Specification.

Kaunas, 2020. 121 pages.

### **Summary**

Requirements specification and analysis is highly dependent on the analyst's knowledge and skills, which in turn increases the empirical factor in requirements specification. Therefore, in the process of user requirements specification it is expedient to use the knowledge base created on the basis of enterprise metamodel, which stores the subject area knowledge (enterprise model), thus increasing the efficiency of requirements specification process by reusing enterprise models knowledge. This will not only reduce the empirical factor and the likelihood of errors, but also save analysts time.

In order to implement the method of creating a requirements specification, an analysis had to be performed: software that is designed to specify requirements, requirements specification templates and knowledge-based systems and their business model. Then the proposed method was specified and an experimental system was designed to test the performance of the method with real data.

In the study, the requirements description forms used in the method were compared with the composition of the user requirements specification standards and it was found that the *Volere* template could be covered up to 97% of the content of the user requirements specification document. Also, using as many fields as possible in the description forms with limited input values and choices can reduce the empirical factor of requirements specification.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>8</b>
<b>Paveikslų sąrašas .....</b>	<b>9</b>
<b>Santrumpų ir terminų sąrašas .....</b>	<b>11</b>
<b>Įvadas.....</b>	<b>12</b>
<b>1. Žiniomis grindžiamos naudotojo reikalavimų specifikacijos sudarymo metodo analizė ...</b>	<b>14</b>
1.1. Analizės tikslas .....	14
1.2. Tyrimo objektas, sritis ir problema .....	14
1.3. Reikalavimų valdymo įrankių palyginimas .....	15
1.3.1. Modern Requirements4TFS įrankio analizė.....	15
1.3.2. IBM Rational DOORS įrankio analizė.....	16
1.3.3. Visure įrankio analizė.....	16
1.3.4. ReQtest įrankio analizė.....	17
1.3.5. Įrankių palyginimas .....	17
1.4. Žiniomis grindžiamos sistemos analizė.....	18
1.4.1. Žiniomis grindžiamų sistemų inžinerijos principai .....	19
1.4.2. Žiniomis grindžiamų sistemų veiklos modelis .....	19
1.5. Reikalavimų specifikavimo šablonų analizė .....	21
1.5.1. <i>Volere</i> šablono analizė.....	22
1.5.2. <i>IEEE 830-1998</i> šablono analizė .....	24
1.5.3. <i>Use Case Approach</i> šablono analizė .....	25
1.5.4. Karl Wieger sukurtas šablonas .....	26
1.5.5. Šablonų palyginimas .....	28
1.6. <i>Volere</i> šablono metamodelio analizė.....	28
1.7. UML panaudojimo atvejų diagramos metamodelio analizė.....	39
1.8. Metodo naudotojų analizė .....	39
1.9. Darbo tikslas, uždaviniai ir siekiami privalumai .....	40
1.10. Siekiamo sprendimo apibrėžimas.....	41
1.11. Analizės išvados .....	43
<b>2. Formalusis metodo reikalavimų specifikacijos ir projekto aprašas .....</b>	<b>44</b>
2.1. Reikalavimų specifikacija.....	44
2.1.1. Metodo veiklų specifikavimas.....	45
2.2. Dalykinės srities modelis.....	53
2.3. Metodo elementų komunikavimas.....	56
2.3.1. Specifikavimo metodo aspektų sekų diagramos.....	58
2.3.2. Specifikacijos redagavimo sekų diagramos.....	60
2.4. Naudotojų sąsajos modelis .....	61
2.5. Formalus sprendimo aprašas .....	64
2.6. Reikalavimų apibendrinimas .....	67
<b>3. Eksperimentinės realizacijos projektas .....</b>	<b>68</b>
3.1. Sistemos reikalavimų specifikacija .....	68
3.1.1. Realizuojamos sistemos veiklos.....	68
3.2. Dalykinės srities modelis.....	80
3.3. Loginė architektūra.....	83

3.4.Panaudojimo atvejų sekų diagramos .....	83
<b>4. Sprendimo realizacija ir testavimas.....</b>	<b>86</b>
4.1.Sprendimo realizacija .....	86
4.2.Sistemos veikimo aprašas .....	87
4.3.Testavimo modelis, duomenys, rezultatai .....	88
<b>5. Eksperimentinis reikalavimų specifikacijos sudarymo metodo tyrimas.....</b>	<b>95</b>
5.1.Eksperimento planas.....	95
5.2.Užpildytų specifikacijų palyginimas su Volere šablono rekomendacijomis.....	95
5.3.Volere šabono atvaizdavimo rezultatai.....	107
5.4.Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas .....	111
5.5.Metodo sudaromos reikalavimų specifikacijos palyginimas su studentų parašytais specifiacijomis .....	118
5.6.Sprendimo taikymo rekomendacijos .....	119
<b>Išvados .....</b>	<b>120</b>
<b>Literatūros sąrašas .....</b>	<b>121</b>
<b>Priedai.....</b>	<b>122</b>
1priedas. Naudotojo instrukcija.....	122
2priedas. Sudaryta specifikacija .....	128

## Lentelių sąrašas

1.1 lentelė Reikalavimų analizės įrankių palyginimo rezultatai.....	17
1.2 lentelė ŽGS veiklos modelio klasių aprašymas.....	21
1.3 lentelė <i>Volere</i> šablonas. Reikalavimo aprašymo forma.....	23
1.4 lentelė <i>Volere</i> šablonas. PA aprašymo forma.....	24
1.5 lentelė <i>IEEE 830-1998</i> šablonas. PA aprašymo forma.....	25
1.6 lentelė <i>Use Case Approach</i> šablonas. Funkcinių reikalavimo aprašymo forma .....	25
1.7 lentelė <i>Use Case Approach</i> šablonas. Pagalbiniai klausimai norint išsiaiškinti apribojimus..	26
1.8 lentelė <i>Use Case Approach</i> šablonas. Projekto tikslo ir priežasties aprašymo forma.....	26
1.9 lentelė Karl Wieger sukurtas šablonas. Funkcinių reikalavimų aprašymo forma .....	27
1.10 lentelė Reikalavimų specifikacijos šablonų palyginimas .....	28
1.11 lentelė <i>Volere</i> šablono aprašymas.....	29
1.12 lentelė Metodo teikiami privalumai naudotojams. ....	40
2.1 lentelė PA Sudaryti specifikaciją pasirinktu aspektu, <i>Volere</i> aprašymas.....	45
2.2 lentelė PA Sukurti elementą, <i>Volere</i> aprašymas.....	48
2.3 lentelė PA Redaguoti elementą, <i>Volere</i> aprašymas .....	49
2.4 lentelė PA Panaikinti elementą, <i>Volere</i> aprašymas .....	51
2.5 lentelė VM klasių priklausomybė su <i>Volere</i> šablonu .....	66
3.1 lentelė PA Prisijungti <i>Volere</i> aprašymas .....	69
3.2 lentelė PA Registruotis <i>Volere</i> aprašymas.....	70
3.3 lentelė PA Peržiūrėti naudotojo projektų sąrašą <i>Volere</i> aprašymas .....	72
3.4 lentelė PA Peržiūrėti visus projektus <i>Volere</i> aprašymas. ....	74
3.5 lentelė PA Sukurti projektą <i>Volere</i> aprašymas .....	75
3.6 lentelė PA Naikinti projektą <i>Volere</i> aprašymas.....	77
3.7 lentelė PA Koreguoti projekto komandą <i>Volere</i> aprašymas.....	78
5.1 lentelė Užpildytų specifikacijų rezultatų palyginimas su <i>Volere</i> šablono turiniu .....	96
5.2 lentelė Palyginimas tarp sukurto reikalavimų specifikavimo metodo ir <i>Volere</i> šablono .....	112
5.3 lentelė Palyginimas situacijos be metodo ir naudojant metodą.....	119



## Paveikslų sąrašas

1.1 pav. Žiniomis grindžiamos sistemos architektūra .....	18
1.2 pav. ŽGS veiklos modelio metamodelis.....	20
1.3 pav. <i>Volere</i> reikalavimų specifikavimo šablono metamodelis .....	38
1.4 pav. UML panaudojimo atvejų diagramos metamodelio schema .....	39
1.5 pav. Veiklos modelio praktinis panaudojimas IS inžinerijoje.....	42
2.1 pav. Reikalavimų specifikacijos sudarymo metodo UML panaudojimo atvejų diagrama.....	44
2.2 pav. PA Sudaryti specifikaciją pasirinktu aspektu UML veiklos diagrama .....	47
2.3 pav. PA Sukurti elementą UML veiklos diagrama.....	49
2.4 pav. PA Redaguoti elementą UML veiklos diagrama .....	51
2.5 pav. PA Panaikinti elementą UML veiklos diagrama .....	53
2.6 pav. UML klasių diagrama aprašanti galimą duomenų bazės struktūrą.....	55
2.7 pav. Apibendrinamoji specifikacijos sudarymo metodo UML sekų diagrama .....	57
2.8 pav. Aspekto Su aktoriumi susijusi informacija UML sekų diagrama.....	59
2.9 pav. Funkcijos Sukurti naują elementą UML sekų diagrama.....	60
2.10 pav. Funkcijos Redaguoti sukurto elemento duomenis UML sekų diagrama.....	61
2.11 pav. Funkcijos Panaikinti elementą UML sekų diagrama.....	61
2.12 pav. Naudotojo sąsajos prototipas. Suinteresuotieji asmenys .....	62
2.13 pav. Naudotojo sąsajos prototipas. Veiklos sudėtis.....	62
2.14 pav. Naudotojo sąsajos prototipas. Duomenų modelis.....	63
2.15 pav. Naudotojo sąsajos prototipas. Sistemos sudėtis.....	63
2.16 pav. Naudotojo sąsajos prototipas. Reikalavimai.....	63
2.17 pav. Naudotojo sąsajos prototipas. Atviros problemos ir klausimai .....	64
2.18 pav. Naudotojo sąsajos prototipas. Uždaviniai.....	64
2.19 pav. Naudotojo sąsajos prototipas. Rizikų įvertinimas .....	64
2.20 pav. Specifikacijos metodo makro-algoritmas .....	65
3.1 pav. Eksperimentinės sistemos UML panaudojimo atvejų diagrama.....	68
3.2 pav. Funkcijos Prisijungti UML veiklos diagrama.....	70
3.3 pav. Funkcijos Registruotis UML veiklos diagrama .....	72
3.4 pav. Funkcijos Peržiūrėti naudotojo projektų sąrašą UML veiklos diagrama.....	73
3.5 pav. Funkcijos Peržiūrėti visus projektus UML veiklos diagrama.....	75
3.6 pav. Funkcijos Sukurti projektą UML veiklos diagrama .....	77
3.7 pav. Funkcijos Naikinti projektą UML veiklos diagrama .....	78
3.8 pav. Funkcijos Koreguoti projekto komandą UML veiklos diagrama .....	80
3.9 pav. VM klasių diagrama .....	81
3.10 pav. <i>Volere</i> šablono UML klasių diagrama .....	82
3.11 pav. Naudotojo duomenų UML klasių diagrama .....	83
3.12 pav. Sistemos loginė architektūra.....	83
3.13 pav. Funkcijos Registruotis UML sekų diagrama .....	84
3.14 pav. Funkcijos Prisijungti UML sekų diagrama.....	84
3.15 pav. Funkcijos Peržiūrėti savo projektus UML sekų diagrama.....	84
3.16 pav. Funkcijos Koreguoti projekto komandą UML sekų diagrama .....	85
4.1 pav. Eksperimentinės sistemos UML diegimo diagrama .....	87
4.2 pav. Ekspertinės sistemos funkcionalumas .....	87

4.3 pav. Funkcionalumo testavimas. Projekto informacijos grąžinimas .....	88
4.4 pav. Funkcionalumo testavimas. Projekto informacijos grąžinimas, projektas neegzistuoja .....	89
4.5 pav. Funkcionalumo testavimas. Projekto funkcijos informacijos grąžinimas .....	89
4.6 pav. Funkcionalumo testavimas. Projekto funkcijos informacijos grąžinimas, funkcija neegzistuoja .....	89
4.7 pav. Funkcionalumo testavimas. Naujo aktorius sukūrimas.....	90
4.8 pav. Funkcionalumo testavimas. Aktoriaus informacijos koregavimas .....	90
4.9 pav. Funkcionalumo testavimas. Aktoriaus informacijos koregavimas, aktorius neegzistuoja ..	91
4.10 pav. Funkcionalumo testavimas. Aktoriaus ištrynimasis .....	91
4.11 pav. Funkcionalumo testavimas. Aktoriaus ištrynimasis, aktorius neegzistuoja .....	91
4.12 pav. Projekto pagrindinės informacijos atvaizdavimas sistemoje .....	92
4.13 pav. Funkcinių naudotojų informacijos atvaizdavimas sistemoje .....	92
4.14 pav. Panaudojimo atvejų aprašomoji forma sistemoje .....	93
4.15 pav. Nefunkcinių reikalavimų atvaizdavimas sistemoje .....	94
4.16 pav. Veiklos kontekstų atvaizdavimas sistemoje .....	94
5.1 pav. Tikslų aprašymas .....	107
5.2 pav. Kitų suinteresuotų asmenų aprašymas .....	107
5.3 pav. Potencialių produkto naudotojų aprašymas .....	107
5.4 pav. Reikalavimų aprašymas .....	108
5.5 pav. Veiklų taisyklių aprašymas .....	108
5.6 pav. Prielaidų aprašymas .....	108
5.7 pav. Veiklos suskaidymo aprašymas .....	108
5.8 pav. Duomenų žodyno aprašymas .....	109
5.9 pav. Panaudojimo atvejų aprašymas.....	109
5.10 pav. Atvirų problemų ir klausimų aprašymas.....	110
5.11 pav. Detalaus kūrimo plano aprašymas .....	110
5.12 pav. Rizikų įvertinimo aprašymas .....	110

## Santrumpų ir terminų sąrašas

### Santrumpos:

- IEEE – Elektros ir elektronikos inžinierių institutas (angl. *Institute of Electrical and Electronic Engineers*);
- ŽGS – žiniomis grindžiama sistema;
- VM – veiklos modelis;
- IS – informacinė sistema;
- PĮ – programinė įranga;
- PA – panaudojimo atvejis;
- RUP (angl. *Rational Unified Process*) – analizės klasių stereotipai.

### Terminai:

- UML (angl. *Unified Modelling Language*) – modeliavimo ir specifikacijų kūrimo kalba, kuri atvaizduoja programų dokumentus;
- panaudojimo atvejų diagrama (angl. *use case diagram*) – diagrama, kuri atvaizduoja, kokie naudotojai gali atlikti, kurias sistemos funkcijas;
- validavimas – veiksmai, kuriais galima įrodyti, kad procedūra ar procesas yra tinkami numatytiems rezultatams gauti;
- metamodelis – konceptualus modelis, kuris apibrėžia sąvokas, ryšius ir semantiką;
- asociacijos ryšys – klasių asociacija reiškia, kad tarp klasių egzempliorių yra ryšys;
- kompozicijos ryšys – agregavimo forma, kai dalis griežtai priklauso visumai ir jos gyvavimo ciklas visiškai priklauso nuo visumos;
- apibendrinimo ryšys – ryšys tarp specifinės klasės (angl. *subclass*) ir apibendrintos klasės (angl. *superclass*), kai subklasė paveldi visas superklasės savybes (atributus, ryšius ir operacijas), ir turi savo papildomas savybes;
- veiklos modelis – CASE įrankio veiklos žinių kaupimo posistemio sudėtinė dalis, kurioje kaupiamos ir saugomos konkrečios kompiuterizuojamos dalykinės srities IS projektinių modelių kūrimui būtinos žinios;
- sudaromos specifikacijos elementas – reikalavimų specifikacijos aprašomasis matas;
- sudaromos specifikacijos aspektas – tipas pagal, kurį yra žinoma, kokią informaciją reikia atrinkti sudarant specifikaciją;
- integralumas – jungimas į visumą;
- atvirojo kodo PĮ – PĮ, kuri pasižymi šiomis savybėmis: nemokama, turi laisvai prieinamus išėties kodus, galima laisvai modifikuoti ir platinti.

## Įvadas

### Darbo problematika ir aktualumas

Reikalavimų specifikuojimas yra vienas svarbiausių informacinių sistemų (IS) kūrimo etapų. Taip pat, šitas etapas yra labiausiai priklausomas nuo specialisto įgudžių ir gebėjimų, kuris atlieka reikalavimų specifikuojimą ir analizę. Kadangi reikalavimų analizė ir specifikuojimas yra labai subjektyvūs veiksmi, žiniomis grindžiamų sistemų (ŽGS) populiarumas išauga. Žiniomis grindžiamos sistemos padeda struktūrizuoti reikalavimus ir juos susisteminti. Taip pat, jei dar galėtume jau turimą ŽGS duomenų bazę papildyti UML diagramų metamodelių ir specifikuojimo elementais, reikalavimų specifikuojimo kokybė būtų mažiau priklausoma nuo reikalavimus surinkinėjančio ir analizuojančio analitiko patirties ir gebėjimų. Tokiu būdu sudarant reikalavimų specifikuojimą būtų galima sumažinti empiriškumo faktorių.

Nors jau yra sukurta daug programinės įrangos [1], skirtos specifikuoti kuriamų sistemų reikalavimus, nei viena jų neturi funkcionalumo, kuris iš turimų dalykinės srities žinių galėtų pusiau automatiškai būdu sugeneruoti norimas diagramas UML notacija. Šis naujai sukurta reikalavimų specifikuojimo sudarymo funkcionalumas būtų naudojamas dviejų naudotojų tipų – sistemų analitikų ir architektų. Analitikai pildytų kuriamų sistemų reikalavimus, o architektai naudotųsi metodo sugeneruotomis diagramomis.

**Darbo tikslas** – pasiūlyti veiklos modeliu (VM) grindžiamą, kompiuterizuotą naudotojo reikalavimų specifikuojimo sudarymo metodą, apimančią UML panaudojimo atvejų diagramų generavimo funkcionalumą, kuris sumažintų empiriškumo faktorių specifikuojant reikalavimus, taip užtikrinant tikslesnį ir pilnesnį reikalavimų specifikuojimo turinį.

### Darbo uždaviniai:

1. išanalizuoti reikalavimų specifikuojimo įrankius ir šablonus bei žiniomis grindžiamų sistemų struktūrą;
2. ištirti UML panaudojimo atvejų diagramos ir reikalavimų specifikuojimo sudarymo galimybes, naudojant veiklos modelį kaip pagrindinį dalykinės srities žinių šaltinį;
3. praturtinti veiklos metamodelio sudėtį integruojant trūkstamus reikalavimų specifikuojimo sudarymui būtinus elementus;
4. sukurti metodo įgyvendinimui būtinus algoritmus, skirtus reikalavimų specifikuojimo sudarymui ir UML panaudojimo atvejų diagramoms generuoti;
5. sukurti prototipą, ir patikrinti jį pasirinktos dalykinės srities testiniais duomenimis;
6. eksperimentiškai patikrinti sukurto metodo funkcionalumo efektyvumą

### Darbo rezultatai ir jų svarba

Papildžius žiniomis grindžiamos sistemos duomenų bazę UML panaudojimo atvejų diagramomis ir reikalavimų specifikuojimo šablono metamodelių elementais, bus galima sukurti metodą, kuris galėtų ne tik struktūrizuoti reikalavimus, ir juos pusiau automatiškai būdu užpildyti bet ir sugeneruoti UML panaudojimo atvejų diagramas. Šis metodas ne tik leistų analitikui sutaupyti laiko specifikuojant reikalavimus, bet ir reikalavimų analizė ir specifikuojimas būtų atliekami standartiniu būdu. Taip būtų sumažintas specifikuojimo sudarymo empiriškumo faktorius ir atliekama mažiau klaidų.

## Darbo struktūra

Žiniomis grindžiamos naudotojo reikalavimų specifikacijos sudarymo metodo analizės skyriuje įsitikinama ar tikrai nėra prieinamas automatizuotas diagramų generavimo funkcionalumas jau sukurtuose ir naudojamuose reikalavimų valdymo įrankiuose. Tai atlikus, toliau yra analizuojamos žiniomis grindžiamos sistemos bei jų veiklos modelis, reikalavimų specifikacijos šablonai, diagramos, kurią planuojama generuoti naudojant siūlomą metodą, metamodelis.

Kuriamo metodo reikalavimų specifikacijos ir projekto aprašymo skyriuje aprašomos galimos metodo funkcijos. Apibrėžiamos galimos klasės, kurios sudarys dvi duomenų bazes. Taip pat komunikacija tarp metodo klasių, kurioms yra priskirti RUP stereotipai. Skyriaus gale atvaizduojami ir aprašomi naudotojo sąsajos prototipai ir makro algoritmas.

Eksperimentinės realizacijos projekto skyriuje aprašoma sistema, kurioje bus panaudotas siūlomas reikalavimų specifikacijos sudarymo metodas. Apibrėžiamos ne tik galimos duomenų bazių struktūros, bet ir papildomos funkcijos, kurios reikalingos norint naudotis specifikacijos sudarymo metodu.

Sprendimo realizacijos ir testavimo skyriuje aprašomos sukurtos sistemos charakteristikos. Taip pat ištestuotas metodo funkcionalumas: ar metodo grąžinamas rezultatas atitinka naudojamo reikalavimų specifikacijos šablono aprašymą ir ar metodas veikia teisingai: ar yra grąžinama egzistuojančio projekto specifikacija ir ar grąžinama klaida, kai prašoma neegzistuojančio projekto specifikacija.

Eksperimentiniame metodo tyrimo skyriuje yra aprašomas lyginamasis tyrimas tarp rekomenduojamų *Volere* šablono skyrių informacijos aprašymo struktūrų ir metodo siūlomo duomenų aprašymo. Palyginus struktūras yra nustatomos metodo kiekybinės savybės, o pagal jas ir rekomendacijos.

## 1. Žiniomis grindžiamos naudotojo reikalavimų specifikacijos sudarymo metodo analizė

Pirmajame skyriuje analizuojamos siūlomo reikalavimų specifikacijos sudedamosios dalys – ŽGS veiklos modelis, reikalavimų specifikacijos šablonai, UML panaudojimo atvejų diagramos struktūra.

### 1.1. Analizės tikslas

Norint suprojektuoti naują metodą, skirtą specifikuoti naudotojo pateiktus reikalavimus, naudojant papildomą žinių bazę, bei automatiškai generuojantį UML panaudojimo atvejų diagramas, pirmiausia reikia įsitikinti ar jau sukurti ir naudojami programos, skirtos reikalavimams valdyti, neturi šio funkcionalumo. Įsitikinus, jog toks funkcionalumas neegzistuoja, reikia išanalizuoti visas siūlomo funkcionalumo sudedamąsias dalis: žiniomis grindžiamas sistemas bei jų veiklos modelį, naudojamo reikalavimų specifikavimo šablono ir generuojamų diagramų metamodelius.

Pasirinkus tinkamiausią specifikavimo šabloną analizei, reikės išnagrinėti jo metamodelį, tam, kad būtų galima suvokti, iš kokių dalių jis yra susidarytas ir kaip kiekviena dalis yra aprašoma. Taip pat, reikės išanalizuoti ir UML panaudojimo atvejų diagramos modelį.

Atlikus šiuos analizės etapus, taps aiškiau kokiais elementais reikės papildyti ŽGS veiklos modelį norint sugeneruoti UML panaudojimo atvejų diagramas iš specifikuotų reikalavimų ir sudaryti sistemos reikalavimų specifikaciją.

### 1.2. Tyrimo objektas, sritis ir problema

**Tyrimo problema** – kuriamos sistemos reikalavimų surinkimas ir specifikacija labai priklauso nuo analitiko, kuris tai atlieka patirties ir gebėjimų. Naudojant žiniomis grindžiamą metodą, reglamentuojantį naudotojo reikalavimų specifikaciją sudarymo ir UML panaudojimo atvejų diagramų generavimo procesą, analitikai greičiau ir patogiau surinktų ir specifikuotų reikalavimus. Taip pat, šis metodas sumažintų klaidų, atsiradusių dėl analitikų subjektyvumo skaičių, ir užtikrintų reikalavimų pilnumą. Tokiu būdu būtų mažinamas empiriškumo faktorius specifikuojant reikalavimus bei modeliuojant UML diagramas.

**Tyrimo objektas** – veiklos žiniomis grindžiamo pusiau automatizuoto reikalavimų specifikacijos sudarymo ir UML panaudojimo atvejų diagramų generavimo metodo procesas.

**Tyrimo sritis** – naudotojo reikalavimų specifikavimo įrankiai, reikalavimų specifikavimo šablonai ir žiniomis grindžiamų sistemų veiklos modelis.

#### Numatomas problemos sprendimas:

1. išanalizuoti ir palyginti jau sukurtus reikalavimų specifikacijos įrankius;
2. išanalizuoti žiniomis grindžiamų sistemų veikimo principus ir veiklos modelį;
3. išnagrinėti ir palyginti reikalavimų surinkimo šablonus;
4. išnagrinėti UML panaudojimo atvejų diagramos kūrimo metodiką bei metamodelį;
5. patobulinti žiniomis grindžiamos dalykinės srities veiklos metamodelį atsižvelgiant į tinkamiausio reikalavimų surinkimo šablono sudėtį ir UML panaudojimo atvejų diagramos metamodelį;
6. sukurti algoritmą, skirtą sudaryti reikalavimų specifikaciją ir generuoti UML panaudojimo atvejų diagramas, naudojant modelyje saugomas žinias;

7. sukurti prototipą, kuris turėtų pusiau automatizuotą naudotojų reikalavimo specifikacijos sudarymo metodą, apimantį UML panaudojimo atvejų diagramų generavimo funkcionalumą;
8. išbandyti sukurtą prototipą su realios srities duomenimis.

### 1.3. Reikalavimų valdymo įrankių palyginimas

Yra skurta daugybė komercinių įrankių [1] skirtų valdyti reikalavimus. Aišku jie nėra vienodi, jie teikia skirtingas funkcijas be kelių standartinių, tokių kaip saugoti užregistruotus reikalavimus, leisti susieti reikalavimus ar aprašyti svarbią informaciją kuriamai programinei įrangai. Šios funkcijos yra privalomos norint valdyti reikalavimus. Toks įrankis taip pat atlieka vieną bene svarbiausių funkcijų – kaupia visą projektui svarbią informaciją vienoje vietoje.

Darbe bus analizuojami keturi įrankiai, kurių pagalba galima palengvinti reikalavimų valdymą. Šie įrankiai buvo pasirinkti todėl, kad yra pakankamai populiarūs ir patenka į geriausių įrankių dvidešimtuką [1]:

- Modern Requirements4TFS;
- IBM Rational DOORS;
- Visure;
- ReQtest.

Įrankiai bus analizuojami pagal pasirinktus kriterijus:

1. netekstinių dokumentų ir failų saugojimas įrankyje – ar galima pridėti dokumentus bei paveikslus prie reikalavimų ar kitų projekto aprašymo dalių;
2. atsekamumas tarp projekto dalių – ar galima susieti panaudojimo atvejus su reikalavimais, panaudojimo atvejus su naudotojų rolėmis, naudotojų roles su reikalavimais, ir kt.;
3. ataskaitų sukūrimas naudojant projekto informaciją;
4. skirtingų šablonų naudojimas;
5. sąsaja su kitomis programomis, kurios leidžia toliau vykdyti IS kūrimą;
6. diagramų modeliavimas/ atvaizdavimas.

#### 1.3.1. Modern Requirements4TFS įrankio analizė

*Modern Requirements4TFS* įrankis papildo *Microsoft TFS* programą, todėl visa informacija yra iškart saugojama debesyje, kas leidžia lengvai stebėti projekto informacijos pakeitimus. Informacijos saugojimas debesyje tikrai nėra vienintelis šio įrankio svarbus funkcionalumas. Daug svarbesnis funkcionalumas yra įrankio teikiama galimybė kontaktuoti su savo komandos nariais, brėžti panaudojimo atvejų diagramas, generuoti ataskaitas naudojant užpildytus reikalavimus arba atsekamumo lenteles iš susietų reikalavimų su artefaktais.

Vienas šio įrankio minusas yra tai, kad negalima išsaugoti susikurtų reikalavimų specifikacijos šablonų ir po to juos panaudoti kituose projektuose, tačiau visos informacijos pildymas įrankyje nesiskiria nuo darbo tekstiniame dokumente – kuriami skyriai ir poskyriai, kuriuose talpiname artefaktus. Kuriami artefaktai gali būti ne tik tekstinio tipo – taip pat galima kelti paveikslus, *MS Excel* ar *MS Visio* failus, kas leidžia pritaikyti priimtinausią šabloną.

Šis įrankis gali būti susietas ir su kitomis programomis, kaip *Smart Word4TFS*, kas leistų turimam projektui kurti testavimo atvejus, ir *Git*, kas leidžia matyti visas kurtas projekto versijas [2].

### 1.3.2. IBM Rational DOORS įrankio analizė

*IBM DOORS* įrankis labiausiai pritaikytas didelės apimties projektams. Projekto aprašymas įrankyje niekuo nesiskiria nuo tekstinio dokumento aprašo. Jame galima sukurti skyrius ir poskyrius, o skyriuose iškart įkelti artefaktus. Tokiu būdu gaunama hierarchinė projekto aprašymo struktūra.

Įrankyje galima susikurti struktūrą pagal sau priimtina reikalavimų specifikavimo šabloną ir artefaktų tipus bei jų atributus, tačiau vienam projektui galima naudoti tik vieną šabloną. Sukurti šablonai išlieka išsaugoti ir galima juos panaudoti kitiems projektams. Kaip ir šablonai, visa projekto informacija – artefaktai – yra saugojami duomenų bazėje, kurią naudotojas užpildo naudodamas interneto prieigą.

Įrankis neteikia diagramų generavimo funkcionalumo, tačiau galima jas nubraižyti pačiame įrankyje arba įkelti jau sumodeliuotas diagramas atvaizdą.

Turbūt pati svarbiausia įrankio funkcija yra artefaktų susiejimas tarpusavyje. Vienas artefaktas gali turėti keletą skirtingų sąsajų su kitais artefaktais. Atributų susiejimas įgauna prasmės tuomet, kai norime sugeneruoti ataskaitas. Yra trys galimi ataskaitų tipai – poveikio, atsekamumo ir trūkstančių nuorodų.

*IBM DOORS* įrankis kaupia pakeitimų istoriją kiekvienam individualiam objektui, moduliui ir specifinėms redagavimo sesijoms. Taip pat, įrankis turi tiesiogines sąsajas su *Microsoft Project*, *Rational Team Concert*, ir *Rational Rose*, todėl galima savo projektą patobulinti papildomomis galimybėmis bei funkcijomis. Reikia nepamiršti, jei yra reikalingas ir popierinis aprašas, tai *IBM DOORS* yra suderinamas su *MS Word*, ir galima ne tik generuoti aprašo tekstinį dokumentą bet ir į įrankį sukelti informaciją iš *MS Word* formato dokumento [3].

### 1.3.3. Visure įrankio analizė

Amerikiečių sukurtas įrankis – *Visure* – tinkamas ne tik reikalavimų nustatymui, jų analizei bet ir validavimui. Įrankis visą gautą informaciją iš naudotojo saugoja duomenų bazėje, todėl galima lyginti projekto versijas tarpusavyje, taip pat gražinti prieš tai buvusios versijos užpildytą informaciją.

*Visure* įrankyje galima saugoti būsimos sistemos reikalavimus naudojant hierarchinę struktūrą, tačiau negalima pritaikyti norimo šablono. Šiai problemai spręsti, įrankio kūrėjai siūlo veiksmus, kurie padės nustatyti reikalavimus:

- įvertinti sistemos įgyvendinamumą;
- įvertinti iš kur kyla reikalavimai – įvertinti suinteresuotus asmenis;
- įvertinti sistemos tikslus: ar sistema yra autonominė, ar turės sąsają su kitomis sistemomis;
- iš suinteresuotų asmenų galima surinkti reikalavimus įvertinant anksčiau paminėtus aspektus.

Įrankyje yra jau sukurta reikalavimų aprašymo forma, kuri tiks ne tik funkciniam, bet ir nefunkciniam reikalavimams. Ta pati forma tinkama aprašyti ir sistemos apribojimams. Įrankio minusas yra tai, kad galimi tik tekstinės informacijos reikalavimai.



Jau užpildytus reikalavimus galima grupuoti, sieti tarpusavyje bei kurti ataskaitas. Taip pat, reikalavimus galima importuoti ir iš tekstinio dokumento, o ataskaitas matyti ne tik įrankyje, bet ir sugeneruoti *MS Word* ar *PDF* formatais.

*Visure* įrankio funkcionalumą galima išplėsti jį susiejant su kitais įrankiais ar programomis, tokiomis kaip *IBM Rational* produktais, *MS Team Foundation Server*, *Atlassian Jira* [4].

### 1.3.4. ReQtest įrankio analizė

*ReQtest* įrankis pasižymi ne tik reikalavimų valdymu, bet ir jų testavimo ir įvykdymo registravimu. Įrankyje paprasta registruoti reikalavimus pagal pateiktą šabloną. Taip pat, prie reikalavimų galima įkelti dokumentus ar paveikslus, kurie gali suteikti reikalavimams papildomos informacijos ir tikslumo. Įrankyje reikalavimai išrikiuojami hierarchine struktūra.

Visas darbas su reikalavimų keitimu, jų įvykdymu ir testavimu atliekamas lanksčioje ir lengvai keičiamoje lentoje, kurioje matomi reikalavimai su jų prioritetu, tipu, statusu, testavimo atvejų kiekiu bei vykdymo progresu.

Visa reikalavimų informacija saugojama debesyje, todėl dirbant komandoje visuomet bus matomi net naujausi projekto pakeitimai per internetinę sąsają.

Įrankyje pastebimas svarbus trūkumas – negalima susieti vieno reikalavimo su kitu. Sąsaja galima tik tarp testavimo atvejų ir reikalavimų. Taip pat nėra galimybės sugeneruoti panaudojimo atvejų diagramos ar įkelti diagramos paveikslą, tačiau galimybė sugeneruoti įvairias ataskaitas, kurias galima matyti ne tik įrankyje, bet ir atsiųsti įvairiu formatu vis tiek išlieka.

*ReQtest* įrankyje esantį testavimo valdymo funkcionalumą galima patobulinti naudojant *Atlassian Jira* įrankį. Tai yra vienintelis įrankis, su kuriuo yra galima integracija [5].

### 1.3.5. Įrankių palyginimas

Atlikus įrankių analizę, reikia šiuos keturis įrankius tarpusavyje palyginti. Palyginimo rezultatai, pateikiami lentelės forma (1.1 lentelė):

1.1 lentelė Reikalavimų analizės įrankių palyginimo rezultatai

Kriterijus	Modern Requirements4TFS	IBM Rational DOORS	Visure	ReQtest
Ataskaitų generavimas	+	+	+	+
Galimybė prie reikalavimo pridėti papildomą informaciją, kaip <i>MS Excel</i> ar <i>MS Word</i> failai, paveikslai, diagramos	+	+	-	-
Atsekamumas tarp projekto dalių	+	+	+	+/-
Skirtingų šablonų panaudojimas	+/-	+	-	-
Norimo šablono panaudojimas	+	+	-	-
Sąsaja su kitomis programomis	+	+	+	+
Diagramų modeliavimas	+	+	-	-
Automatizuotas diagramų generavimas	-	-	-	-

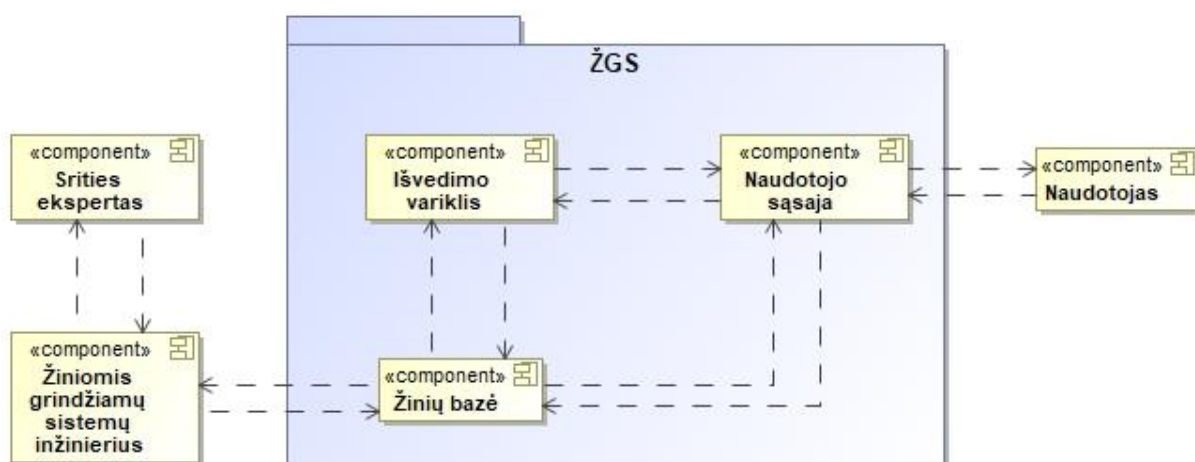
Lyginant šiuos keturis įrankius pagal prieš tai apibrėžtus kriterijus, būtų galima daryti išvadas, kad daugiausia mūsų analizuojamų sąvybių turi *IBM Rational DOORS* ir *Modern Requirements4TFS* įrankiai. Jie atitinka visus išvardintus kriterijus, išskyrus patį svarbiausią – automatizuotą diagramų generavimą, todėl šio darbo tikslas ir yra sukurti trūkstamą funkcionalumą, o ne patobulinti jau esamą.

#### 1.4. Žiniomis grindžiamos sistemos analizė

Norint sukurti naudotojo reikalavimų specifikacijos sudarymo metodą, kuris remtųsi žiniomis grindžiamų sistemų veiklos modeliu, būtina suvokti kaip veikia šios sistemos. Taip pat, išanalizuoti ŽGS veiklos modelį, kurį reikės papildyti elementais, kad būtų galima aprašyti pilną reikalavimų specifikaciją.

Norint užtikrinti, kad žiniomis grindžiama sistema (ŽGS) veiktų gerai, reikia ne tik išmanyti programinės įrangos projektavimą, bet ir duomenų bazes, dirbtinį intelektą, duomenų gavybą ir sprendimų palaikymo sistemas. Šios sistemos yra naudojamos problemoms spręsti, nes ŽGS gali užpildyti savo veiklos modelį ne tik tiesiogiai gaunant informaciją iš naudotojo, bet ir netiesiogiai. Sistemos duodami rezultatai yra vis tiek glaudžiai susiję su asmenimis, kurie naudojami sistema, nes problemų sprendimo būdai pateikiami remiantis žmonių mokymais, sprendimų priėmimu ir veiksmis [6].

Žiniomis grindžiamos sistemos pagrindiniai komponentai ir išoriniai naudotojai, kurie bendrauja su sistema, yra atvaizduojami diagramoje, kuri sukurta pagal [6] šaltinį (1.1 pav.):



1.1 pav. Žiniomis grindžiamos sistemos architektūra

Pagal [6] ir [7] šaltinius ŽGS, sudaro šie pagrindiniai komponentai:

- išvedimo variklis – kuriame yra saugojami metodai, pagal kuriuos yra analizuojama informacija iš žinių bazės, bei priimamos išvados iš gautos informacijos;
- žinių bazė – vieta, kur yra talpinamos ir grupuojamos probleminės srities žinios bei taisyklės, pagal kurias yra sprendžiamos problemos. Žinių bazę gali užpildyti ŽGS inžinierius, kuris informaciją gauna iš srities eksperto arba sistema gali pati užsipildyti žinių bazę, jei sistemoje yra įdiegtas apsimokymo modulis;
- naudotojo sąsaja – skirta komunikacijai tarp sistemos ir naudotojo.

#### 1.4.1. Žiniomis grindžiamų sistemų inžinerijos principai

Atsižvelgiant į IS inžinerijos raidos analizę bei ŽGS inžinerijos komponentų sudėtį, buvo sukurti penki principai, pagal kuriuos galima nustatyti kokybiškus ŽGS inžinerijos metodus. Pagal [8] šaltinį, pateikiami penki pagrindiniai ŽGS principai.

**P1. Teorinė ŽGS paskirtis** – analizuoti ir kompiuterizuoti tiriamos veikos dėsningumą, kuris išreiškia veiklos esmę.

**P2. Veikos modelio paskirtis** – specifikuoti veiklos valdymo procesus, kurie pasitaiko veiklos srityje. Taip pat atlikus veiklos srities dėsningumo specifikaciją, galima sudaryti koncepcinį veiklos modelį. Šis modelis yra reikalavimų inžinerijos pagrindas.

**P3. Žiniomis grindžiamų CASE metodų paskirtis** – apibrėžti IS gyvavimo ciklo etapus. Kiekvienas gyvavimo etapas yra apibrėžiamas veiklos metamodeliu.

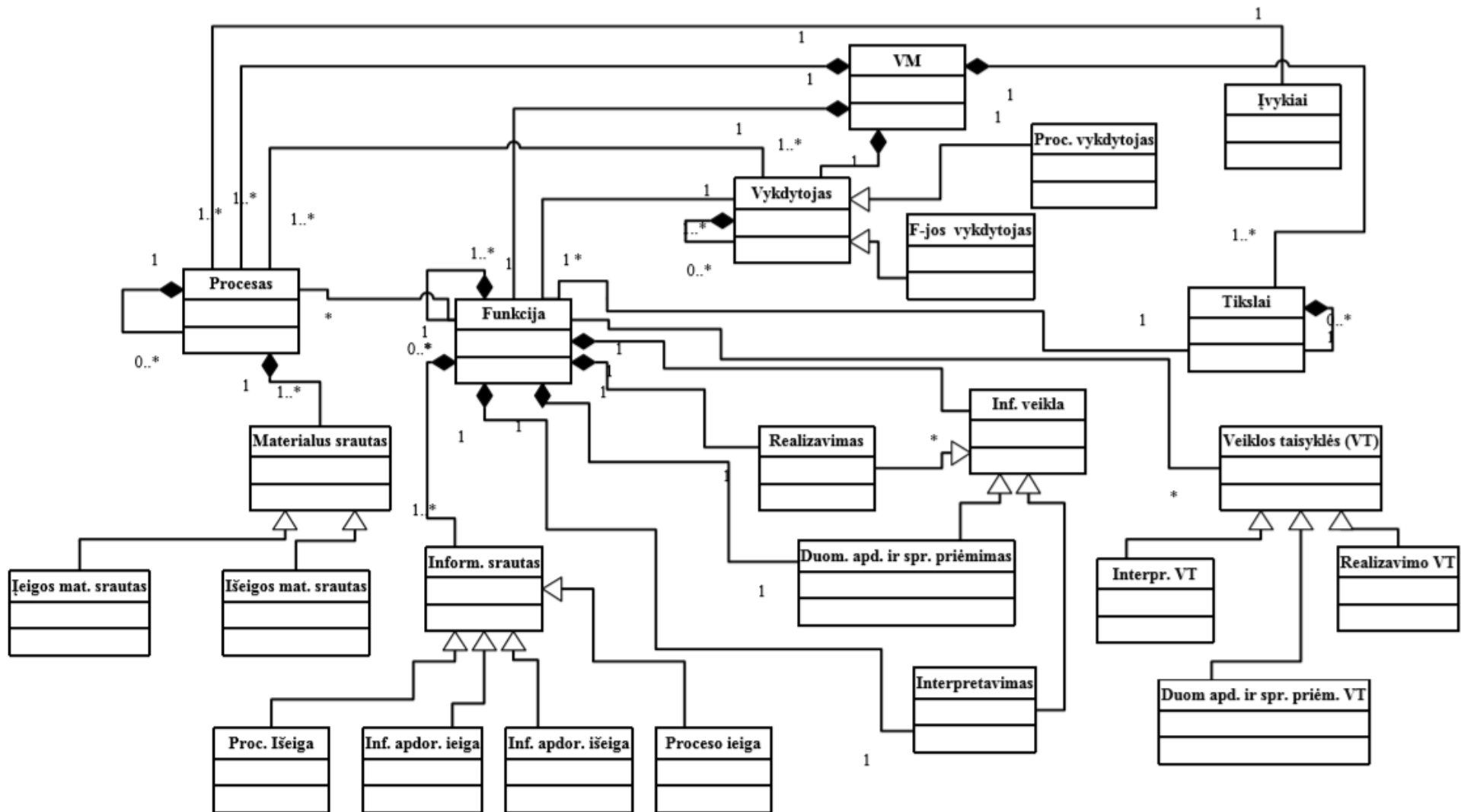
**P4. Žiniomis grindžiamos CASE sistemos privalomus komponentus galima apibrėžti įvertinus veiklos metamodelį bei formalizuotą veiklos modelį.** Yra reikalingi du komponentai:

1. žinių posistemė – saugykla skirta veiklos modelio specifikacijai bei formalizuoto veiklos modelio kūrimui;
2. žinių bazė – teorinio veiklos modelio specifikacijos saugykla, kuri taip pat saugoja formalizuotą konkrečios veiklos srities modelį.

**P5. Žiniomis grindžiamos CASE sistemos paskirtis** – tikrinti sprendimus, kuriuos priima sistema kiekviename IS kūrimo etape. Kadangi žinių bazėje yra saugojamas formalizuotas veiklos modelis, kuris aprašo konkrečios veiklos valdymo procesus, IS projektą galima verifikuoti objektyviai.

#### 1.4.2. Žiniomis grindžiamų sistemų veiklos modelis

Šiame darbe bus naudojamosi KTU mokslininkų sukurtu veiklos modeliu, kuris yra tinkamas kuriamam naudotojo reikalavimų specifikacijos sudarymo metodui, kadangi jo struktūra leidžia aprašyti bet kokios sistemos dalykinę sritį. Bet norint įgyvendinti norimą metodą, yra būtina patobulinti šį modelį, todėl reikia jį išanalizuoti, t.y. susipažinti su visomis modelio klasėmis ir suvokti kokie ryšiai sieja jas. Veiklos modelio analizė bus atliekama remiantis [9] ir [10] disertacijomis bei [11], [12] šaltiniais. veiklos modelis atvaizduojamas UML klasių diagrama (angl. *UML class diagram*) – 1.2 pav.



Šaltinis: [9].

1.2 pav. ŽGS veiklos modelio metamodelis

Veiklos modelis susideda iš 24 klasių, iš kurių dvi pagrindinės klasės yra: *Procesai* ir *Funkcijos*. Modelis vadovaujasi principu – kiekviena funkcija valdo procesą. *Funkcija* tai sudėtinis elementas su hierarchine struktūra, kuris valdo *Procesus*. *Procesai* yra valdomi siunčiant duomenis *Funkcijoms*, kurios, naudojant *Informacinius srautus*, gali apdoroti gautą informaciją. *Procesas* – veiksmas, kuris atliekamas norint gauti materialų rezultatą. Tiek *Funkcijos*, tiek *Procesai* turi *Vykdytojus*.

Pagrindinės veiklos modelio klasės plačiau aprašomos 1.2 lentelėje.

**1.2 lentelė ŽGS veiklos modelio klasių aprašymas**

Klasės pavadinimas	Klasės aprašymas	Klasės ryšiai su kitomis klasėmis
<b>VM</b>	Veiklos modelio klasė.	Kompozicijos ryšys su <i>Funkcija</i> , <i>Procesas</i> , <i>Tikslai</i> ir <i>Vykdytojai</i> klasėmis.
<b>Procesas</b>	Žingsnių seka, kuria siekiama įgyvendinti rezultatą. Procesai naudoja materialiuosius resursus tam, kad sukurti kitus materialius resursus.	Kompozicijos ryšys su <i>Procesas</i> , <i>Materialius srautas</i> klasėmis. Asociacijos ryšys su <i>Funkcija</i> , <i>Įvykiai</i> ir <i>Vykdytojai</i> klasėmis.
<b>Įvykiai</b>	Saugojami duomenys įvykių, kurie gali sukelti pakeitimus sistemos procesuose.	-
<b>Materialus srautas</b>	Saugojami procesų materialiniai įėjimo ir išėjimo rezultatai.	Apibendrinimo ryšys su <i>Įeigos mat. srautas</i> , <i>Išeigos mat. srautas</i> klasėmis.
<b>Funkcija</b>	Detaliau aprašo dalykinės srities funkcijas, kadangi ši klasė yra kompleksinės struktūros. Ją papildo informaciniai srautai, veiklos taisyklės ir informacinė veikla.	Kompozicijos ryšys su <i>Funkcija</i> , <i>Inform. srautas</i> , <i>Interpretavimas</i> , <i>Duom. apdor. ir spr. priėmimas</i> , <i>Realizavimas</i> klasėmis. Asociacijos ryšys su <i>Procesas</i> , <i>Tikslas</i> , <i>Vykdytojas</i> klasėmis.
<b>Inform. srautas</b>	Saugojami funkcijų informaciniai įėjimo ir išėjimo rezultatai. Ši informacija susieja procesus su funkcijomis.	Apibendrinimo ryšys su <i>Proc. įeiga</i> , <i>Inf. apdor. įeiga</i> , <i>Inf. apdor. išeiga</i> , <i>Proc. įeiga</i> klasėmis.
<b>Veiklos taisyklės (VT)</b>	Veiklos taisyklės tai sąlygos, apribojimai ar matematinės išraiškos, kurie gali papildyti veiklos elementą.	Apibendrinimo ryšys su <i>Interpr. VT</i> , <i>Realizavimo VT</i> , <i>Duom. apdor. Ir spr. priėmimo VT</i> klasėmis.
<b>Int. veikla</b>	Aprašo informacines veiklas, tokias kaip duomenų apdorojimas, sprendimų priėmimas, žingsniai, kurie pakeičia proceso būseną, informacija, kurią funkcija priima ir/ar grąžina.	Apibendrinimo ryšys su <i>Realizavimas</i> , <i>Duom. apdor. ir spr. priėmimas</i> , <i>Interpretavimas</i> klasėmis.
<b>Vykdytojas</b>	Aprašo, koks asmuo yra atsakingas už funkcijų arba procesų vykdymą.	Apibendrinimo ryšys su <i>Proc. vykdytojas</i> ir <i>F-jos vykdytojas</i> klasėmis. Kompozicijos ryšys su <i>Vykdytojas</i> klase.
<b>Tikslai</b>	Saugojami dalykinės srities tikslai, kurie bus įgyvendinami su funkcijomis.	Kompozicijos ryšys su <i>Tikslai</i> klase.

### 1.5. Reikalavimų specifikavimo šablonų analizė

Norint užtikrinti struktūrizuotą reikalavimų aprašymą metode, reikia atlikti reikalavimų specifikavimo šablonų analizę.

Šiuo metu yra sukurta panašių ir skirtingų reikalavimų specifikacijos šablonų. Kiekvienas iš jų turi savo trūkumų ir privalumų. Šiame darbe bus analizuojami keturi populiariausi, labiausiai analizuojami ir minimi literatūroje [13] šablonai:

- Volere;
- IEEE 830-1998;
- Use Case Approach;
- Karl Wieger.

Šie šablonai bus analizuojami remiantis bendrais reikalavimų specifikacijos kriterijais, labiau išsiplečiant ties tais kriterijais, kurie yra susiję su UML panaudojimo atvejų diagramomis (angl. *UML use case diagram*), kurias turėtų generuoti siūlomas metodas. Kriterijai, kurių aprašymo struktūra bus analizuojama:

- 1. funkciniai reikalavimai:** kokias funkcijas sistema turi atlikti, kokius žingsnius reikia įvykdyti norit įgyvendinti funkcijas, kokie apribojimai skirti sistemos funkcijoms;
- 2. nefunkciniai reikalavimai:** kaip aprašoma, kaip sistema turi atrodyti, sistemos kokybė: patogumas, patikimumas, greitaveika, saugumas, palaikomumas, palaikomos operacinės sistemos, būtinos integracijos;
- 3. apribojimai:** viso projekto apribojimai, tokie kaip laikas, kaina, būtinos panaudoti technologijos, darbo vietos apribojimai;
- 4. projekto informacija:** koks yra sistemos tikslas, kokie yra projekto suinteresuoti asmenys, projekto pagrindimas, kaip yra vykdomi darbai, kuriuose bus integruota sistema.

#### 1.5.1. Volere šablono analizė

Pirmoji *Volere* šablono versija pasaulį išvydo 1995 metais. Šablonas tinkamas ne tik programinei įrangai aprašyti bet ir aparatinei įrangai, paslaugoms ir organizacijoms. *Volere* šablono įvertinimas pagal kriterijų aprašomąsias struktūras [14]:

- 1. funkciniai ir nefunkciniai reikalavimai:** šablone vienodai aprašinėjami funkciniai ir nefunkciniai reikalavimai. Šie reikalavimai aprašinėjami pildant specifišką kortelę, kurioje aprašoma visa reikalinga informacija:
  - tipas ir numeris – funkcinis reikalavimas, nefunkcinis reikalavimas, apribojimas;
  - nuoroda į panaudojimo atvejį, jei tai funkcinis reikalavimas; nuoroda į funkcinį reikalavimą, jei tai nefunkcinis reikalavimas arba apribojimas;
  - aprašymas;
  - pagrindimas – kodėl yra reikalingas reikalavimas;
  - šaltinis – kas inicijavo šį reikalavimą;
  - atlikimo kriterijus – aprašoma, kaip bus įsitikinama ar reikalavimas įvykdytas;
  - naudotojo tenkinimas ir netenkinimas – skaitinė reikšmė 1-5;
  - prioritetas;
  - ryšiai tarp reikalavimų;
  - konfliktai su kitais reikalavimais;
  - šaltiniai – įstatymai, formos;
  - paskutinio atnaujinimo data.

Funkcinių ir nefunkcinių reikalavimų aprašymo kortelės struktūra (1.3 lentelė):

### 1.3 lentelė *Volere* šablonas. Reikalavimo aprašymo forma

<b>Reikalavimas #:</b>	<b>Reikalavimo tipas:</b>	<b>PA/FR#:</b>
<b>Aprašymas:</b>		
<b>Pagrindimas:</b>		
<b>Šaltinis:</b>		
<b>Atlikimo kriterijus:</b>		
<b>Naudotojo tenkinimas:</b>	<b>Naudotojo netenkinimas:</b>	
<b>Prioritetas:</b>	<b>Konfliktai:</b>	<b>Ryšiai:</b>
<b>Šaltiniai:</b>		
<b>Paskutinio atnaujinimo data:</b>		

Taip pat tik funkciniai reikalavimai, kurie yra atvaizduojami UML panaudojimo atvejų diagramoje yra aprašomi dar viena struktūra (1.4 lentelė):

- panaudojimo atvejo numeris ir pavadinimas;
- tikslai ir (ar) uždaviniai;
- aprašymas;
- sąlyga prieš;
- naudotojo rolė;
- sužadinimo sąlygos – kokias funkcijas atlikus sistemoje, bus galima vykdyti šią funkciją;
- susiję panaudojimo atvejai:
  - o išplečiantys panaudojimo atvejai – tai panaudojimo atvejai susieti <<*extend*>> ryšiu;
  - o apimami panaudojimo atvejai – tai panaudojimo atvejai susieti <<*include*>> ryšiu;
  - o apibendrinti panaudojimo atvejai – tai panaudojimo atvejai susieti apibendrinimo ryšiu.
- pagrindinis scenarijus – veiksmai, kuriuos atlieka naudotojas ir sistema, norint įvykdyti šią funkciją;
- rezultatas – rezultatai atlikus funkciją;
- alternatyvūs scenarijai – aprašomi veiksmai, kuomet funkcijos vykdyme galimi loginiai išsiskyrimai.

#### 1.4 lentelė *Volere* šablonas. PA aprašymo forma

PA		
Tikslas/uždavinys.		
Aprašymas.		
Sąlyga prieš		
Aktorius		
Sužadinimo sąlyga		
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Apibendrinti PA	
Pagrindinis scenarijus		
Naudotojo veiksmai		Sistemos veiksmai
Rezultatas		
Alternatyvūs scenarijai		

#### 2. apribojimai:

- apribojimai sprendimui – atvaizduojami taip pat, kaip funkciniai ir nefunkciniai reikalavimai, naudojant specialiąją kortelę (1.3 lentelė);
- kiti apribojimai – aprašomi laisva tekstine forma arba naudojant UML diagramas.

#### 3. projekto informacija:

- projekto pagrindimas – aprašomas laisva tekstine forma;
- projekto tikslai – aprašomi trijų kriterijų junginiu, kuris susideda iš trumpai apibūdinto tikslo, privalumų, jei bus įgyvendintas šis tikslas, ir išmatavimo būdų;
- suinteresuoti asmenys – aprašomi lentelėmis arba laisva teksto forma;
  - o projekto užsakovas ir pirkėjas aprašomi tekstine forma;
  - o į suinteresuotų asmenų sąrašą, reikėtų įtraukti ne tik įmonės, kuri atliks sistemos kūrimą darbuotojus, bet ir pirkėjų įmonės darbuotojus, kurie prisidės prie sistemos vykdymo. Lentelė susideda iš suinteresuotų asmenų vaidmens aprašymo, pagrindimo, dalyvių kiekio, etapo, kuriame suinteresuoti asmenys prisidės prie projekto vykdymo ir kokias su projektu susijusias žinias jie turi;
  - o potencialūs sistemos naudotojai taip pat aprašomi lentelės forma, kurioje galima matyti šią informaciją apie rolę: pagrindinės funkcijos, pagrindimas, amžiaus ribos ir IT raštingumas.
- dabartinė veikla be sistemos (veiklos kontekstas) – laisvos formos diagrama, kurioje atvaizduojama duomenų kaita tarp veikloje dalyvaujančių šalių. Taip pat, dar rekomenduojama lentelės struktūra paaiškinti kiekvieną duomenų srautą. Srautas apibūdinamas pavadinimu, aprašymu, kas atlieka duomenų srauto perdavimą, kas priima srautą.

#### 1.5.2. *IEEE 830-1998* šablono analizė

Vienas žinomiausių reikalavimų specifikavimo šablonų yra *IEEE 830-1998*. Šablono vertinimas pagal kriterijus [15]:

##### 1. funkciniai reikalavimai – panaudojimo atvejai aprašomi struktūrizuota forma (1.5 lentelė):

- numeris;
- panaudojimo atvejo pavadinimas;



- prieš sąlyga;
- koks naudotojas atlieką šį panaudojimo atvejį;
- pagrindimas;
- nuo kurių kitų panaudojimo atvejų priklauso aprašomas panaudojimo atvejis;
- pagrindinis scenarijus.

#### 1.5 lentelė *IEEE 830-1998* šablonas. PA aprašymo forma

<b>ID</b>	
<b>PA</b>	
<b>Rolė</b>	
<b>Pagrindimas</b>	
<b>Priklausomybės</b>	
<b>PA</b>	<b>Pagrindinis scenarijus</b>

2. **nefunkciniai reikalavimai** – aprašomi laisva tekstine forma. Taip pat, nefunkcinius reikalavimus galima aprašyti naudojant pagalbines diagramas arba atvaizdus, bet nėra tiksliai nurodoma kokius;

3. **apribojimai** – aprašomi laisva tekstine sąrašo forma;

4. **projekto informacija** – aprašoma laisva tekstine forma.

#### 1.5.3. *Use Case Approach* šablono analizė

Šablono *Use Case Approach* įvertinimas pagal kriterijus [16]:

1. **funkciniai reikalavimai** – aprašomi pildant panaudojimo atvejo aprašymo lentelę (1.6 lentelė):

- panaudojimo atvejo pavadinimas;
- trumpas aprašymas;
- pagrindinis scenarijus;
- alternatyvus scenarijus;
- sąlyga prieš ir po įvykių atlikimo;
- specialūs reikalavimai.

#### 1.6 lentelė *Use Case Approach* šablonas. Funkcinių reikalavimo aprašymo forma

	<b>Aprašymas</b>
<b>PA pavadinimas</b>	
<b>Trumpas aprašymas</b>	
<b>Pagrindinis scenarijus</b>	
<b>Alternatyvus scenarijus</b>	
<b>Sąlyga prieš</b>	
<b>Sąlyga po</b>	
<b>Specialūs reikalavimai</b>	

2. **nefunkciniai reikalavimai** – aprašomi laisva tekstine forma. Šablonas pateikia galimus nefunkcinių reikalavimų tipus su atitinkamais klausimais, į kuriuos atsakius, bus galima aprašyti visus nefunkcinius reikalavimus;

3. **apribojimai** – suvokiami atsakant į klausimus, kurie yra aprašyti 1.7 lentelėje. Klausimai sugrupuoti pagal apribojimų grupes:

### 1.7 lentelė Use Case Approach šablonas. Pagalbiniai klausimai norint išsiaiškinti apribojimus

Apribojimo grupė	Klausimai
Ekonominiai	Koks yra sistemos biudžetas? Ar reikės pirkti licencijas?
Politiniai	Ar yra vidiniai ar išoriniai politiniai nesklaidumai, kurie trukdytų projekto sėkmei? Ar yra vidinių įmonės problemų?
Techniniai	Ar yra privaloma naudoti specifines technologijas? Ar bus privaloma dirbti su nurodytomis priemonėmis? Ar bus draudžiama naudoti naujausias technologijas? Ar bus privaloma naudotis jau nupirkta programine įranga?
Sisteminiai	Ar kuriama sistema bus autonominė? Ar reikės daryti integraciją tarp kitų sistemų? Kokios operacinės sistemos turės būti naudojamos?
Tvarkaraščio ir išteklių	Ar tvarkaraštis jau yra numatytas ir jo keisti negalima? Ar galima keisti reikalingų resursų kiekius?

Atsakius į šiuos klausimus galime užpildyti apribojimų lentelę, kuri atvaizduoja tokią informaciją: apribojimo grupė, pavadinimas ir paaiškinimas.

#### 4. projekto informacija:

- problemos, kurias panaikintų projektas ir tikslai aprašomi struktūriniu lentele (1.8 lentelė):

### 1.8 lentelė Use Case Approach šablonas. Projekto tikslo ir priežasties aprašymo forma

	Apibūdinimas
Problema/Tikslas	
Kuriems asmenis tai daro įtaką	
Rezultatas	
Privalumai	

- svarbūs projekto asmenys, naudotojai ir pirkėjas yra aprašomi laisva tekstine forma, atsakant į šiuos klausimus:

1. Kas naudosis šia sistema?
2. Kas yra šios sistemos pirkėjas?
3. Ką dar paveiks ši sistema?
4. Kas nuspręs ar sistema veikia taip kaip turėtų veikti?
5. Kas prižiūrės ir sukurs sistemą?

#### 1.5.4. Karl Wieger sukurtas šablonas

Skyriuje analizuojamas šablonas buvo aprašytas dviejų autorių, *Karl Wieger* ir *Joy Beatty*, šaltinyje [17], kuris yra išvestas iš *IEEE* šablono .

1. **funkciniai reikalavimai** – aprašyti funkciniai reikalavimai yra naudojama struktūrizuota forma (1.9 lentelė), kuri susideda iš:

- ID – nepasikartojanti reikalavimo identifikatorius;
- pavadinimas;
- užregistravo – vardas ir pavardė, asmens, kuris užregistravo reikalavimą;

- užregistravimo data;
- pirminis aktorius – aktorius, kuris atlieka funkciją;
- antriniai aktoriai – aktoriai, kurie yra priklausomi nuo funkcijos rezultato;
- aprašymas;
- trigeriai – priežastys, dėl kurių būtų vykdoma funkcija;
- sąlyga prieš – kas turi būti įvykdyta aktoariaus prieš vykdant funkciją;
- sąlygos po – funkcijos vykdymo rezultatas;
- pagrindinis srautas – žingsnių seka tarp sistemos ir aktoariaus;
- alternatyvūs srautai – alternatyvių žingsnių seka tarp sistemos ir aktoariaus;
- išimtys – žingsnių seka tarp sistemos ir aktoariaus;
- prioritetas;
- naudojimo dažnumas;
- veiklos taisyklės – panaudojimo atvejų numeriai;
- papildoma informacija;
- prielaidos.

Taip pat visi funkciniai reikalavimai privalo būti susieti su panaudojimo atveju.

#### 1.9 lentelė Karl Wieger sukurtas šablonas. Funkcinių reikalavimų aprašymo forma

<b>ID ir Pavadinimas</b>			
<b>Užregistravo</b>		<b>Užregistravimo data</b>	
<b>Pirminis aktorius</b>		<b>Antriniai aktoriai</b>	
<b>Aprašymas</b>			
<b>Trigeris</b>			
<b>Sąlyga prieš</b>			
<b>Sąlyga po</b>			
<b>Pagrindinis srautas</b>			
<b>Alternatyvūs srautai</b>			
<b>Išimtys</b>			
<b>Prioritetas</b>			
<b>Naudojimo dažnumas</b>			
<b>Veiklos taisyklės</b>			
<b>Papildoma informacija</b>			
<b>Prielaidos</b>			

**2. nefunkciniai reikalavimai** aprašomi kaip apribojimai. Taip pat nefunkciniai reikalavimai yra susiejami su panaudojimo atvejais;

**3. apribojimai** aprašomi tekstine taisyklių forma. Taisyklėse nurodomi veiksmai:

- kuriuos privaloma įvykdyti;
- kuriuos draudžiama atlikti;
- kuriuos tik tam tikri naudotojai, gali atlikti.

**4. projekto priežastys:**

- projekto pagrindimas – aprašomas laisva tekstine forma;
- suinteresuoti asmenys – aprašomi lentelėmis arba laisva teksto forma. Būtinai aprašant šiuos punktus, kiekvienai svarbiai asmenų grupei:
  - priežastys, kodėl jiems reikalingas kuriamas projektas;

- galimas nusistatymas prieš/už projektą;
- pagrindinės susidomėjimo priežastys;
- apribojimai, kuriuos jie gali išsakyti.

### 1.5.5. Šablonų palyginimas

Norint išsirinkti reikalavimų specifikavimo šabloną iš keturių analizuotų, kuriuo bus naudojama siūlomą reikalavimų specifikacijos sudarmo metodą, reikia juos tarpusavyje palyginti. Reikalavimų specifikavimo šablonų palyginimas aprašytas 1.10 lentelėje:

**1.10 lentelė Reikalavimų specifikacijos šablonų palyginimas**

Palyginimo kriterijus	Volere	IEEE 830-1998	Use Case Approach	Carl Wieger šablonas
Funkciniai reikalavimai aprašomi struktūriniu forma nenaudojant PA	+	-	-	+
Funkciniai reikalavimai aprašomi pagal PA	+	+	+	+
Nefunkciniai reikalavimai aprašomi struktūriniu forma	+	+/-	-	-
Apribojimai aprašomi struktūriniu forma	+/-	-	+	-
Projekto problemos ir tikslai aprašomi struktūriniu forma	+/-	-	+	-
Projekto suinteresuotieji asmenys ir naudotojai aprašomi struktūriniu forma	+	-	+/-	+/-

Kaip matome iš pateiktos lentelės, remiantis metodui svarbiais kriterijais, kurie yra susiję su reikalavimų specifikavimu bei panaudojimų atvejų diagramos informacijos aprašymu, Volere šablonas atitinka daugiausia kriterijų, lyginant su kitais trimis šablonais. Svarbiausia yra struktūrizuoti kuo daugiau informacijos ir naudoti tą pačią pildymo formą. Taip daroma norint išvengti subjektyvumo ir nereikalingos informacijos rašymo, ką ir suteikia Volere šablonas.

### 1.6. Volere šablono metamodelio analizė

Analizuojant Volere reikalavimų specifikacijos šablono skyrius, kurie yra pateikti [14] knygoje, buvo sudarytas šablono aprašymas (1.11 lentelė). Aprašymas susideda iš skyriaus pavadinimo, jo aprašymo, siūlomo informacijos pateikimo būdo bei ar skyriaus informacija yra padengiama VM.

Išanalizavus visus šablono skyrius buvo galima sudaryti Volere šablono metamodelį (1.3 pav.). Metamodelis sumodeliuotas naudojantis UML klasių diagrama. Šablono modelis buvo modeliuojamas, nes nebuvo surastas jau sukurtas metamodelis, todėl nes Volere šablonas yra skirtas naudoti tradiciniu būdu – pildant dokumentaciją rankiniu būdu.

Norint, kad sudaroma specifikacija padengtų visą Volere šabloną, reikės veiklos modelio metamodelį papildyti trūkstantomis klasėmis. Padengiamos yra tik tos klasės, kurios yra būtinos aprašyti sistemos funkcionalumui.

### 1.11 lentelė *Volere* šablono aprašymas

Šablono skyrius	Apibūdinimas	Aprašymo struktūra	
<b>1 Sistemos paskirtis</b>		Skyrius atsako į klausimą kodėl reikia sistemos.	
<b>1a Analizuojama veikla ir/ar projekto priežastys</b>	Aprašoma įmonės veikla, priežastys kodėl yra reikalinga sistema.	Veikla ir priežastys aprašomos laisva forma.	Yra atitikmuo VM
<b>1b Tikslai</b>	Tikslai, kurie bus pasiekti naudojantis sistema.	Tikslą sudaro: tikslas, paaiškinimas, tikslo rezultato išmatavimas.	Yra atitikmuo VM
<b>2 Suinteresuoti asmenys</b>		Aprašomos projekto sistemos suinteresuotos šalys.	
<b>2a Užsakovas</b>	Sistemos užsakovas, dažnai tai būna sistemos finansinis rėmėjas.	Pažymimas vardas (pavadinimas), sprendimai už kuriuos užsakovas yra atsakingas.	Nėra atitikmens VM
<b>2b Pirkėjas</b>	Sistemos pirkėjas, dažnai užsakovas yra ir pirkėjas	Pažymimas vardas (pavadinimas), sprendimai už kuriuos pirkėjas yra atsakingas.	Nėra atitikmens VM
<b>2c Kiti suinteresuoti asmenys</b>	Asmenys, kurie prisideda prie sistemos kūrimo arba jai daro įtaką. Pavyzdžiai: <i>programuotojai, testuotojai, dabartinės sistemos naudotojai, teisininkai.</i>	Informacija aprašoma lentelės pavidalu, nurodant: 1. asmens rolę; 2. pagrindimą; 3. dalyvaujančių projekte kiekį; 4. kur šis asmuo dirba; 5. dalyvavimą projekte; 6. turimas žinias.	Nėra atitikmens VM
<b>2d Potencialūs produkto naudotojai</b>	Asmenys, kurie naudosis sistema.	Informacija aprašoma lentelės pavidalu, nurodant: 1. naudotojo rolę; 2. pagrindines atliekamas funkcijas (dalyvavimą projekte); 3. pagrindimą; 4. amžiaus ribas; 5. IT raštingumą; 5. kitą reikšmingą informaciją.	Yra atitikmuo VM
<b>2e Personazai</b>	Galimų naudotojų aprašymai. Šis punktas naudojamas, kai kuriama sistema yra skirta naudotis plačiai žmonių bendruomenei.	Informacija aprašoma pasakojamuoju stiliumi pridėdant galimo naudotojo nuotrauką ar piešinį. Tekste turėtų būti paminėta ši informacija: 1. amžius; 2. darbovietė; 3. šeima; 4. laisvalaikio praleidimo būdai; 5. gyvenamoji vieta; 6. mėgstami/ nemėgstami dalykai; 7. požiūris į pinigus/ technologijas; 8. kita svarbi informacija, kuri gali paveikti požiūrį į kuriamą produktą;	Nėra atitikmens VM

<b>2f Naudotojams priskirti prioritetai</b>	Nustatomas prioritetas kiekvienai naudotojų rolei.	Naudotojų rolės skirstomos į: 1. Pagrindiniai naudotojai; 2. Antraeiliai naudotojai; 3. Nesvarbūs naudotojai Galima pridėti prie 2d skyriaus.	Yra atitikmuo VM
<b>2g Naudotojų dalyvavimas projekte</b>	Kuriems projekto kūrimo etapams ir kurioms funkcijoms naudotojo rolė galės/privalės suteikti sistemai reikalavimus.	Aprašoma lentelės forma, galima pridėti prie 2d skyriaus.	Yra atitikmuo VM
<b>2h Sistemos aptarnaujantis personalas</b>	Personalas, kuris prižiūrės ir keis sistemą, taip pat naudotojai, kurie administruos ją.	Šią informaciją galima pridėti atitinkamai prie 2c ir 2d skyrių.	Nėra atitikmens VM
<b>3 Apribojimai</b>		Sistemos kūrimo apribojimai.	
<b>3a Apribojimai sprendimui</b>	Nurodoma kaip sistemos kūrimo problema turi būti išspręsta. Apibūdinamos privalomos technologijos ar sprendimo būdas, pridedami paaiškinimai.	Galima aprašyti naudojantis reikalavimų kortele (1.3 lentelė), būtina paminėti pagrindimą ir tinkamumo kriterijų.	Nėra atitikmens VM
<b>3b Diegimo aplinka</b>	Apribojimai skirti apibūdinti sistemos/technologijos diegimo aplinkai.	Patogiausia atvaizduoti UML diegimo diagrama (angl. <i>UML deployment diagram</i> ).	Nėra atitikmens VM
<b>3c Komunikuojančios sistemos</b>	Aprašomos susijusios/ bendraujančios sistemos.	Patogiausia atvaizduoti UML diegimo diagrama.	Nėra atitikmens VM
<b>3d Prieinama specializuota programinė įranga (PI)</b>	Nurodoma kokia nemokama, komercine ar atvirojo kodo PI reikia naudoti kuriant produktą.	PI dokumentacijos, taip pat nurodoma kokias funkcijas, reikalavimus apima PI.	Nėra atitikmens VM
<b>3e Numatoma darbo vietos aplinka</b>	Aprašoma numatoma sistemos naudotojų darbo vieta, būtina nurodyti savybes, kurios gali daryti įtaką darbui su sistema. Pavyzdžiai: <i>darbo vietoje yra vienas spausdintuvas, todėl dokumentų spausdinimas turi būti minimalus; darbo vieta yra triukšminga, dėl to negali būti jokių garsinių signalų sistemoje;</i>	Darbo vietos aprašymas, galima pridėti nuotraukų ar vaizdo įrašų.	Nėra atitikmens VM
<b>3f Sistemos kūrimo terminas</b>	Aprašomi visi žinomi ir svarbūs terminai: projekto, etapų, itin svarbių darbų. Taip pat reikia pagrindimo kodėl būtent iki tos datos turi būti atlikti darbai ir pasekmės jei nebus įvykdyta iki termino pabaigos. Šis punktas nėra privalomas	Aprašomas rašytine forma iš trijų dalių: terminas, pagrindimas terminui, pasekmės neįvykdžius darbų iki termino pabaigos.	Nėra atitikmens VM

<b>3g Sistemos kūrimo biudžeto apribojimai</b>	Aprašomas projekto biudžetas, arba pinigine išraiška arba ištekliais. Šis punktas nėra privalomas.	Aprašomas rašytine forma iš dvejų dalių: pinigine išraiška, šaltinis iš kur gaunamas biudžetas.	Nėra atitikmens VM
<b>3 h Organizacijos/politiniai apribojimai</b>	Apribojimai, kurie nėra artimi projekto tikslui, bet yra svarbūs sistemos užsakovui. Kaip <i>produktas turi būti pagamintas tik iš amerikietišku dalių; visos sistemos funkcijos turi būti pasiekiamos generaliniam direktoriui;</i>	Galima aprašyti naudojantis reikalavimų kortele (1.3 lentelė), būtina paminėti pagrindimą ir tinkamumo kriterijų.	Nėra atitikmens VM
<b>4 Sąvokos ir santrumpos</b>	Nurodomos sąvokos, terminai ir trumpiniai, kuriuos naudoja suinteresuotos projekto šalys.	Sąrašo forma, nurodant terminą ir jo reikšmę.	Nėra atitikmens VM
<b>5 Svarbūs faktai ir prielaidos</b>		Svarbūs specifikacijos faktai, kurie nėra apibrėžiami kitose šablono skyriuose.	
<b>5a Faktai</b>	Informacija, kuri daro įtaką projektui, bet nėra nei reikalavimai, nei apribojimai.	Laisva forma.	Nėra atitikmens VM
<b>5b Veiklos taisyklės</b>	Informacija, kuri suteikia papildomos informacijos apie dalykinę sritį, nuo kurios gali priklausyti funkcijų veikimas. Kaip <i>vairuotojo pamaina negali viršyti 12 val.; papildoma nuolaida suteikiama klientams, kurių užsakymo suma viršija 100 eurų.</i>	Laisva forma, kuri būtina aprašo veiklos taisyklę, priešastį taisyklei ir kas įgalino šią taisyklę.	Yra atitikmuo VM.
<b>5c Prielaidos</b>	Prielaidos, kurios yra susijusios tiek su pačiu projekto vykdymu, tiek su dalykinės srities pokyčiais.	Laisva forma, kuri būtina turi aprašyti prielaidą ir pasekmes, jei prielaida išsipildytu. Priklausomai nuo prielaidos sudėtingumo, gali reikėti pateikti papildomų šaltinių, kurie suteiktų daugiau informacijos.	Nėra atitikmens VM
<b>6 Veiklos sudėtis</b>		Aprašoma projekto dalykinė sritis ir kaip projektas įsilieja į suinteresuotų šalių aplinką.	
<b>6a Esama padėtis</b>	Analizuojami dabartiniai procesai nesinaudojant kuriu produktu. Į analizę įtraukiami visi procesai, kuriuose bus naudojama nauju produktu.	Siūloma naudoti UML veiklos diagramas (angl. <i>UML activity diagram</i> ), BPMN veiklos procesų diagramas.	Yra dalis informacijos VM
<b>6b Veiklos kontekstas</b>	Dalykinės srities analizė, kuri leidžia suprasti kokios suinteresuotos šalys yra atsakingos už kurią informacijos dalį, kuri yra naudojama būsimos sistemos. Informacija nurodoma dar nenaudojant naujo produkto.	Siūloma braižyti laisvos formos diagramą, kuri susideda iš duomenų srautų tarp suinteresuotų šalių ir galimo produkto.	Nėra atitikmens VM

<b>6c Veiklos suskaidymas (įvykių specifikacija)</b>	Sąrašas duomenų srautų, kurie yra atvaizduojami 6b skyriuje.	Lentelės pavidalu, nurodant: 1. pavadinimą; 2. duomenų srauto siuntėją; 3. duomenų srauto gavėją; 4. duomenų srauto aprašymą.	Nėra atitikmens VM
<b>7 Duomenų modelis ir jo elementų žodynas</b>			
<b>7a Duomenų modelis</b>	Atvaizduojamos klasės ir jų ryšiai. Kad palengvinti, klasių atrinkimą, rekomenduojama įtraukti tas klases, kurių informacija bus atvaizduojama sistemoje ir (arba) koreguojama.	Rekomenduojama naudoti UML klasių modelį (angl. <i>UML class diagram</i> ), subjektų ryšių diagrama (angl. <i>entity relationship diagram</i> ).	Nėra atitikmens VM
<b>7b Duomenų žodynas</b>	Aprašoma pavaizduota informacija 7a skyriuje.	Lentelės forma su informacija: 1. klasės pavadinimas; 2. atributai, jų aprašymai ir galimų reikšmių apribojimai; 3. klasės ryšiai su aprašymais.	Nėra atitikmens VM
<b>8 Sistemos sudėtis (panaudojimo atvejų modelis)</b>			
<b>8a Sistemos ribos (panaudojimo atvejų diagrama)</b>	Identifikuojami ryšiai tarp sistemos naudotojų (aktorių) ir sistemos funkcijų.	Atvaizduojama UML panaudojimo atvejų diagrama.	Nėra atitikmens VM
<b>8bc Panaudojimo atvejai</b>	Aprašomas kiekvienas panaudojimo atvejis.	Atvaizduojama lentelės forma – 1.4 lentelė..	Yra dalis informacijos VM
<b>9 Papildomi funkciniai reikalavimai</b>	Aprašomi papildomi funkciniai reikalavimai, kurie nebuvo jau aprašyti 8bc skyriuje.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>10 Reikalavimai sistemos išvaizdai</b>			
<b>10a Išvaizdos reikalavimai</b>	Aprašomi reikalavimai susiję su sistemos išvaizdos dizainu. Kaip <i>sistemoje privalo būti matomas įmonės logotipas; sistemos pagrindinė spalva privalo būti mėlyna.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>10b Stiliaus reikalavimai</b>	Aprašomi reikalavimai susiję su sistemos pojūčiu ir stiliumi. Kaip <i>sistema turi atrodyti patikima.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11 Reikalavimai panaudojamumui</b>			



<b>11a Naudojimosi paprastumas</b>	Aprašomi reikalavimai susiję su sistemos panaudojimu: ar paprasta naudotis kuriamu produktu, ar lengva prisiminti, kuriose vietose yra funkcijos, ar lengva rasti reikalingus mygtukus, kokios gaunamos klaidos ir pan.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11b Personalizavimo ir kalbos konfigūravimo reikalavimai</b>	Nurodomi reikalavimai susiję su kalbų panaudojimu, klaidų tikrinimais, matavimo ir valiutų ženklais. Taip pat galimybės kiekvienam naudotojui koreguoti sistemos nustatymus pagal jo poreikius. Kaip <i>naudotojas galės pasirinkti iš 6 kalbų; naudotojas galės pasirinkti valiutą.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11c Mokymosi reikalavimai</b>	Reikalavimai, kurie specifikuoja kaip paprasta turėtų būti išmokti naudotis produktu. Kaip <i>sistema turi būti tokia paprasta, kad neprireiktų naudotojams mokymų.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11d Suprantamumo ir mandagumo reikalavimai</b>	Produktas turėtų būti suprantamas naudotojams, todėl jie nereikėtų mokintis papildomų terminų, kad galėtų produktu naudotis – tokius reikalavimus reikia surašyti 11d skyriuje. Kaip <i>sistema privalo naudoti medicininius terminus.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11e Prieinamumas neįgaliesiems</b>	Reikalavimai susiję su palengvinimais, kurie būti naudojami neįgaliųjų.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>11f Suteikiami patogumai</b>	Reikalavimai susiję su užduočių supaprastinimu, naudotojų darbo palengvinimu.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12 Efektyvumo ir našumo reikalavimai</b>			
<b>12a Reikalavimai užduočių vykdymo greičiui</b>	Reikalavimai aprašantys per kiek laiko sistema turi įvykdyti funkcijas. Taip pat galima aprašyti kiek laiko turėtų naudotojas užtrukti vykdydamas sistemos užduotis.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12b Darbo ir aplinkos saugos reikalavimai</b>	Kiekybiškai įvertinama žala žmogui, materialiam turtui ar aplinkai.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12c Reikalavimai tikslumui</b>	Kiekybinis produkto teikiamų rezultatų tikslumo įvertinimas.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM

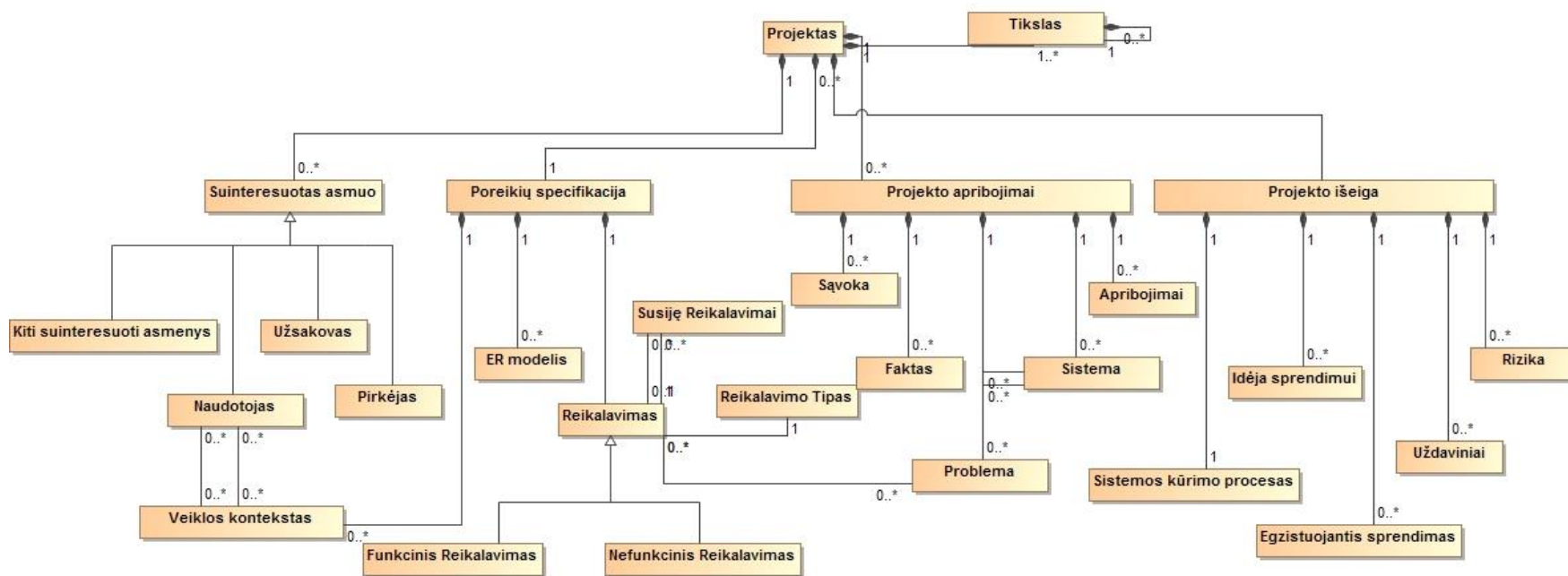
<b>12d Patikimumas ir pasiekiamumas</b>	Šiame skyriuje nurodomas būtinas gaminio patikimumas, kaip bendras leistinas gedimų skaičius. Taip pat nurodomas kuriamo produkto prieinamumas.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12e Atsparumas trukdžiams ir klaidoms</b>	Reikalavimai skirti nurodyti produkto veikimą neįprastomis aplinkybėmis. Kaip <i>sistemos reakcija prisijungus dideliame žmonių kiekiui.</i>	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12f Reikalavimai apdorojamų duomenų apimtims</b>	Reikalavimai skirti nurodyti apimtį, kurias privalo galėti apdoroti produktas, bei saugomų duomenų kiekis produkte.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12g Reikalavimai išplečiamumui</b>	Reikalavimai skirti nurodyti numatomo dydžio padidėjimo valdymą.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>12h Reikalavimai produkto ilgaamžiškumui</b>	Reikalavimai skirti apibrėžti produkto tarnavimo laiką.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>13 Reikalavimai veikimo sąlygoms</b>			
<b>13a Numatoma fizinė aplinka</b>	Reikalavimai skirti apibūdinti produkto naudojimo aplinką. Aprašomos sąlygos, kurioms gali prireikti specialių reikalavimų pasiruošimo ar mokymų metu.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>13b Aplinkosauginiai reikalavimai</b>	Visi reikalavimai susiję su ekologiškumu, išsaugojimu, perdirbimu. Reikalavimai padės apsvarstyti visus produkto aspektus, kurie gali turėti įtakos aplinkai.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>13c Reikalavimai darbui su gretimomis sistemomis</b>	Reikalavimai aprašantys bendraujančias sistemas. Šiuos reikalavimus būtina žinoti prieš kuriant produktą, kad būtų galima numatyti integracijos susisiekiavimo langus.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>13d Reikalavimai sist. platinimo/gamybos formatui</b>	Reikalavimai, kurie yra būtini, kad produktas taptų parduodamu daiktu.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>13e Reikalavimai leidybos procesui</b>	Reikalavimai aprašantys naujo produkto išleidimo ciklus ir formą. Šie reikalavimai padeda nustatyti kaip dažnai bus išleidžiami nauji produktai.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM

<b>13f Reikalavimai atvirkštiniam suderinamumui</b>	Reikalavimai skirti aprašyti produkto suderinamumą su senesne gaminio versija ar kitais produktais.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>14 Reikalavimai sistemos priežiūrai</b>			
<b>14a Sistemos aptarnavimas</b>	Reikalavimai skirti aprašyti kiekybinį laiko įvertinimą, kiek užtruktų atlikti produkto pakeitimus.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>14b Sistemos palaikymas</b>	Reikalavimai skirti nurodyti reikalingą produkto palaikymo lygį.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>14c Pritaikymo platformoms reikalavimai</b>	Reikalavimai skirti aprašyti perkėlimą į kitą platformą ar aplinką.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>15 Reikalavimai saugumui</b>			
<b>15a Prieigos reikalavimai (teisės)</b>	Reikalavimai, kurie nurodo kas turi teisį gauti prieigą prie produkto, bei kurioms dalims (funkcijoms) suteikiamos šios prieigos.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>15b Vientisumo (integralumo) reikalavimai</b>	Reikiamų duomenų bazių ir kitų failų bei paties produkto vientisumo specifikacija. Kokie veiksmai turi būti imama kai vyks puolimai iš išorės ar netinkami naudotojo veiksmai.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>15c Reikalavimai privatumui</b>	Specifikacija, kuri nurodo ką produktas turėtų daryti, kad būtų užtikrintas saugomos informacijos privatumas.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>15d Audito reikalavimai</b>	Reikalavimai, ką gaminsys turėtų atlikti, kad būtų galima vykdyti audito patikrinimus.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>15e Reikalavimai savisaugai nuo išorinių grėsmių</b>	Reikalavimai, kurie aprašo kaip produktas turi būti apsaugotas nuo neteisėtos ar nepageidaujamos PĮ.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>16(a) Kultūriniai reikalavimai</b>	Reikalavimai būdingi sociologiniams veiksniams, turintiems įtakos produkto priimtinumui. Ypač svarbu, jei produktas kuriamas užsienio rinkai.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>17 Atitikties reikalavimai</b>			

<b>17a Atitikties teisės aktams reikalavimai</b>	Teisiniai reikalavimai, kurie padės išspręsti vėlavimų, ieškinių ir teisinių mokesčių.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>17b Atitikties standartams reikalavimai</b>	Reikalavimai, nurodantys, kokius standartus reikia naudoti.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>18 Atviros problemos ir klausimai</b>	Veiksniai, kurie šiuo momentu nėra aiškūs iki galo, bet gali ženkliai pakeisti produkto kūrimą.	Lentelės forma aprašoma: 1. numeris; 2. sąsaja su produkto dalimi (funkcijos numeris, reikalavimo numeris ir pan.); 3. aprašymas; 4. susijusios suinteresuotos šalys; 5. veiksmai, kas galėtų įvykti; 6. problemos sprendimo būdas;	Nėra atitikmens VM
<b>19 Egzistuojantys sprendimai</b>		Sprendimai, kuriuos galima panaudoti, taip sutaupant laiko, kuriant savo produktą.	
<b>19a Prieinamos sistemos</b>	Egzistuojantys kiti produktai, kuriuos galima panaudoti, kuriant savo produktą.	Sąrašo forma, paminint papildomą informaciją apie produktą kaip: 1. kaina; 2. prieinamumas; 3. laiko intervalas per kurį galima sukurti reikalingą integraciją.	Nėra atitikmens VM
<b>19b Prieinami komponentai</b>	Galimi komponentai, kuriuos galima įsigyti arba jau buvo kurti įmonėje, kuriuos galima panaudoti.	Sąrašo forma, paminint papildomą informaciją apie produktą kaip: 1. kaina; 2. prieinamumas; 3. laiko intervalas per kurį galima sukurti reikalingą integraciją.	Nėra atitikmens VM
<b>19c Kopijuotini sprendimai</b>	Egzistuojantys sprendimai, kuriuos galima panaudoti, su minimaliomis korekcijomis.	Sąrašo forma, paminint papildomą informaciją apie produktą kaip: 1. kaina; 2. prieinamumas; 3. laiko intervalas per kurį galima sukurti reikalingą integraciją.	Nėra atitikmens VM
<b>20 Naujos problemos</b>			
<b>20a Poveikis diegimo aplinkai</b>	Aprašomas poveikis, kurį gali sukelti naujas produktas dabartinei diegimo aplinkai. Taip pat reikėtų paminėti, ko naujas produktas neturėtų daryti diegimo aplinkoje.	Laisva tekstine forma.	Nėra atitikmens VM
<b>20b Poveikis esamoms sistemoms</b>	Poveikis, kurį gali sukelti integracijos su nauju produktu.	Laisva tekstine forma.	Nėra atitikmens VM
<b>20c Probleminė naudotojų reakcija</b>	Detaliai aprašomos galimos naudotojų reakcijos į naują produktą.	Laisva tekstine forma.	Nėra atitikmens VM
<b>20d Apribojimai diegimo aplinkoje</b>	Aprašymai bet kokių galimų problemų, kurios yra susijusios sąryšiu tarp naujo produkto ir diegimo aplinkos.	Laisva tekstine forma.	Nėra atitikmens VM

<b>20e Kitos potencialios problemos</b>	Identifikuojamos situacijos, kurioms yra reikalingas pasiruošimas, norint jų išvengti.	Laisva tekstine forma.	Nėra atitikmens VM
<b>21 Uždaviniai</b>		Užduotys ir uždaviniai, kuriuos reikia įgyvendinti, norint sukurti produktą.	
<b>21a Sistemos kūrimo procesas</b>	Aprašomas kūrimo ciklas, kuris bus naudojamas.	Siūloma atvaizduoti aukšto lygio procesų diagrama. Taip pat pridėti aprašymą.	Nėra atitikmens VM
<b>21b Detalus kūrimo planas</b>	Aprašomos produkto kūrimo fazės ir jose esantys darbai.	Galima braižyti LEAN grafiką ir pridėti aprašymą, kuris nurodo: 1. fazės pavadinimas; 2. naudotojo nauda; 3. atlikimo data; 4. naudojami komponentai, funkciniai ir nefunkciniai reikalavimai.	Nėra atitikmens VM
<b>22 Migravimas į naują produktą</b>			
<b>22a Migravimo į naują produktą reikalavimai</b>	Reikalingi darbai, norint kad naujas produktas veiktų diegimo aplinkoje. Gali reikėti perkonvertuoti duomenų bazę, procedūras.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>22b Duomenų transformavimo reikalavimai</b>	Reikalingos duomenų transformacijos.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>23 Rizikų įvertinimas</b>	Aprašomos galimos rizikos ir galimybės, kurios gali įvykti produkto kūrimo metu.	Lentelės forma, nurodant: 1. pavadinimas; 2. aprašymas; 3. tikimybė, kad rizika įvyks; 4. poveikis ir jo paaiškinimas; 5. planas minimizuoti rizikos faktorių; 6. planas sumažinti rizikos poveikį;	Nėra atitikmens VM
<b>24 Kaštai</b>	Nurodomas kaštų skaičiavimo būdas ir gaunama kaštų suma.	Laisva forma.	Nėra atitikmens VM
<b>25 Naudotojo dokumentacija ir apmokymai</b>			
<b>25a Reikalavimai naudotojų dokumentacijai</b>	Aprašomi reikalavimai reikalingoms dokumentacijoms, kurios bus naudojamos mokymams. Net tik kokių jų reikia, bet ir kas turėtų būti jose būtinai paminėta	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM
<b>25b Reikalavimai naudotojų apmokymui</b>	Aprašomi reikalavimai naudotojų mokymams: kokių ir kiek jų reikės, kas jose turėtų būti dėstoma.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitikmens VM

<b>26 Perspektyviniai reikalavimai</b>	Reikalavimai, kurie šioje produkto versijoje, nebus įgyvendinami. Bet į kuriuos galima atsižvelgti, kuriant kitą produkto versiją.	Atvaizduojama lentelės forma – 1.3 lentelė.	Nėra atitiktens VM
<b>27 Idėjos sprendimams</b>	Papildomos idėjos kūrimo procesui, kurios nėra aprašomos kituose skyriuose.	Laisva forma.	Nėra atitiktens VM



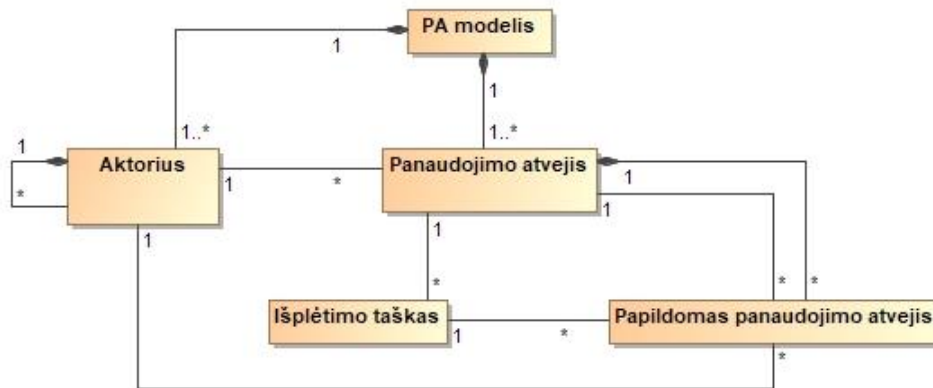
1.3 pav. Volere reikalavimų specifیکavimo šablono metamodelis

## 1.7. UML panaudojimo atvejų diagramos metamodelio analizė.

UML panaudojimo atvejų diagramos metamodelis susideda iš *Aktorių*, *Panaudojimo atvejų* bei ryšių tarp jų (galimi ryšiai: asociacijos (angl. *association*), apibendrinimo (angl. *generalization*), apimantys (angl. *include*), išplečiantys (angl. *extend*)). *Aktorius* gali būti naudotojas arba išorinė kita sistema. Jie galės atlikti sistemos funkcijas (*Panaudojimo atvejis* ir *Papildomas panaudojimo atvejis*). *Papildomi panaudojimo atvejai* gali būti trijų tipų: apimantys, apibendrinantys, išplečiantys.

Apimantys panaudojimo atvejai bus įvykdomi, kartu su jų pagrindiniu panaudojimo atveju, o išplečiantys panaudojimo atvejai, bus įvykdomi tik tuomet, kai sąlyga, nurodyta *Išplėtimo taškas* klasėje, bus teisinga.

Metamodelio aprašymas ir atvaizdavimas (1.4 pav.) sudaryti remiamas [10] ir [18] šaltiniais



1.4 pav. UML panaudojimo atvejų diagramos metamodelio schema

## 1.8. Metodo naudotojų analizė

Naujai sukurtas funkcionalumas – automatizuotas reikalavimų analizei reikalingų diagramų generavimas ir pusiau automatizuotas naudotojo reikalavimų specifikacijos sudarymas – yra skirtas dviejų tipų naudotojams:

- sistemų analitikams – pagrindinis darbas išgauti ir analizuoti sistemos reikalavimus. Naudotojas, kuris pildys informaciją apie dalykinę sritį;
- sistemų architektas – iš pateiktų reikalavimų generuos sistemos projektavimui reikalingas diagramas.

Metodo naudotojų analizė aprašyta – 1.12 lentelėje.

### 1.12 lentelė Metodo teikiami privalumai naudotojams.

Naudotojas	Galimos problemos	Funkcionalumo privalumai
Sistemų analitikas	Empiriškumo faktorius sudarant reikalavimų specifikaciją	Šabloninė reikalavimų užrašymo forma
	Žmogiškosios klaidos	
	Ilgas procesas	Sistemos galimybės dalį reikalavimams reikalingų punktų panaudoti pakartotinai žinių bazę, kuri sukurta naudojanti ŽGS veiklos modeliu
Sistemų architektas	Klaidos modeliuojant diagramą	Automatizuotas diagramų generavimas
	Ilgas procesas	
	Empiriškumo faktorius modeliuojant diagramas	

### 1.9. Darbo tikslas, uždaviniai ir siekiami privalumai

**Darbo tikslas** – pasiūlyti veiklos modeliu grindžiamą, kompiuterizuotą naudotojo reikalavimų specifikacijos kūrimo metodą, apimančią UML panaudojimo atvejų diagramų generavimo funkcionalumą, kuris sumažintų empiriškumo faktorių specifikuojant reikalavimus, taip užtikrinant tikslesnį ir pilnesnį reikalavimų specifikacijos turinį.

#### Darbo uždaviniai:

1. išanalizuoti reikalavimų specifikacijos įrankius ir šablonus bei žiniomis grindžiamų sistemų struktūrą;
2. ištirti UML panaudojimo atvejų diagramos ir reikalavimų specifikacijos sudarymo galimybes, naudojant veiklos modelį kaip pagrindinį dalykinės srities žinių šaltinį;
3. praturtinti veiklos metamodelio sudėtį integruojant trūkstamus reikalavimų specifikacijos sudarymui būtinus elementus;
4. sukurti metodo įgyvendinimui būtinus algoritmus, skirtus reikalavimų specifikacijos sudarymui ir UML panaudojimo atvejų diagramoms generuoti;
5. sukurti prototipą, ir patikrinti jį pasirinktos dalykinės srities testiniais duomenimis;
6. eksperimentiškai patikrinti sukurto metodo funkcionalumo efektyvumą.

#### Siekiami privalumai

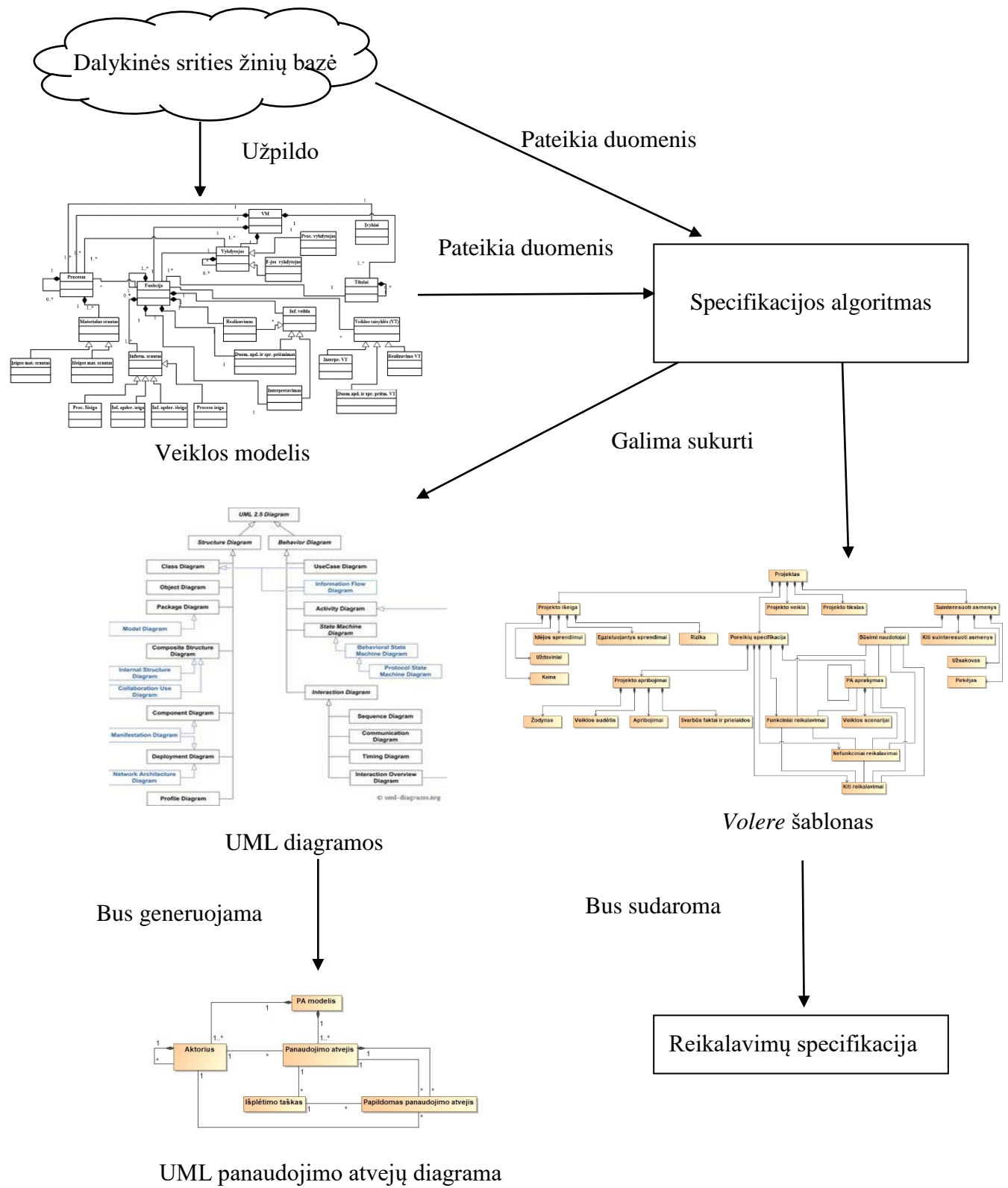
Siūlomas funkcionalumas ne tik sutaupys laiko specifikuojant reikalavimus, bet ir turėtų sumažinti klaidų tikimybę. Taip būtų dėl to, kad reikalavimus reikėtų fiksuoti struktūrizuotoje formoje. Taip pat, automatizuotas diagramų generavimas panaikins diagramų modeliavimo darbus bei klaidas, kurios galėtų atsirasti jas modeliuojant.



### **1.10. Siekiamo sprendimo apibrėžimas**

Kaip galima pritaikyti ŽGS veiklos modelį, pavaizduota laisvos formos diagramoje (1.5 pav.). Tokia forma pasirinkta, dėl aiškesnio suvokimo, koku būdu galima pritaikyti veiklos modelį. Algoritmas gauna duomenis iš veiklos modelio, kuris yra užpildomas informacija iš dalykinės srities — tai gali būti atlikta analitiko, kuris suveda duomenis, arba algoritmas pats save išmokina jį užpildyti. Likusi informacija reikalinga specifikacijai yra užpildoma rankiniu būdu.

Kai algoritmui yra pateikiami duomenys, galima generuoti UML diagramas ir projekto informaciją pagal *Volere* šabloną. Kuriamas metodas apsiribos tik UML panaudojimo atvejų diagramų ir naudotojo reikalavimų specifikacijos sudarymu.



1.5 pav. Veiklos modelio praktinis panaudojimas IS inžinerijoje

### 1.11. Analizės išvados

Atlikus analizę buvo padarytos šios išvados:

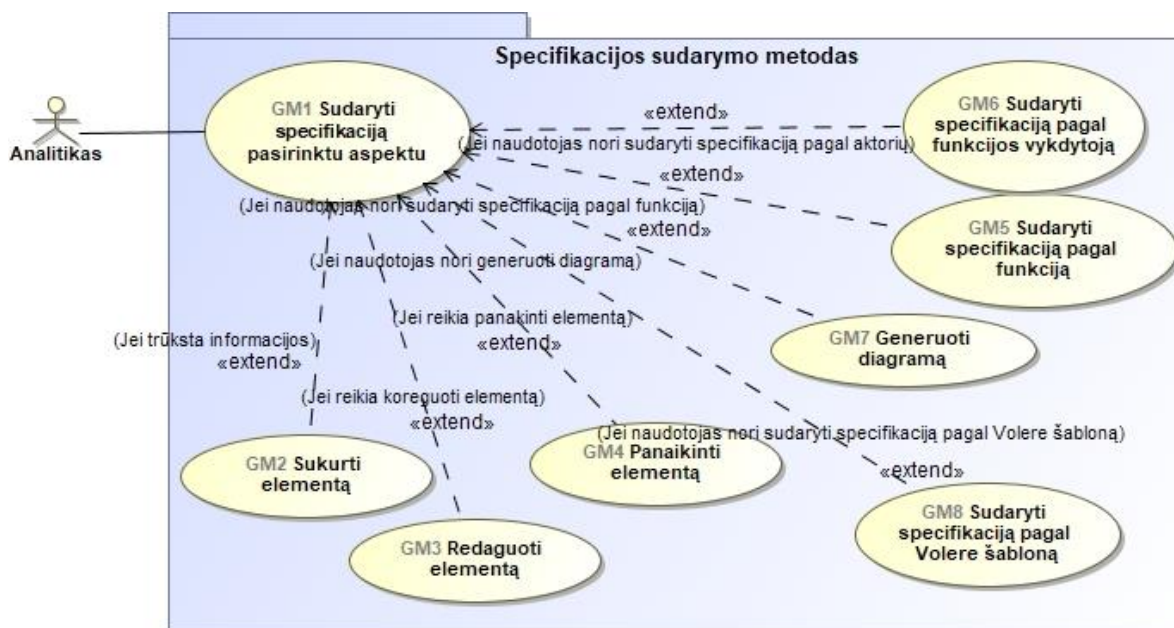
1. palyginus keturis reikalavimų specifikacijos įrankius (*Modern Requirements4TFS*, *IBM Rational DOORS*, *Visure*, *ReQtest*), nustatyta, kad nei vienas iš jų neturi funkcionalumo, skirto generuoti su naudotojo reikalavimų specifikavimo procesu susijusias UML diagramas, todėl tikslinga sukurti naudotojo reikalavimų specifikacijos sudarymo metodą;
2. įvertinus ŽGS veiklos modelį, nustatyta, kad jį papildžius naujais elementais arba pridėjus naujų atributų esamiems, naudotojo reikalavimų specifikavimo procesą galima praturtinti pusiau automatizuotu naudotojo reikalavimų specifikacijos (įskaitant UML projektines diagramas) sudarymo funkcionalumu;
3. palyginus keturis reikalavimų specifikacijos šablonus (*Volere*, *IEEE 830-1998*, *Use Case Approach*, *Karl Wiegner*), metodo kūrimui pasirinktas – *Volere* šablonas, todėl, kad kuriamojo metodo funkcionalumui (lyginant su kitais nagrinėtais šablonais) realizuoti leidžia naudoti plačiausią pasirinkimą struktūrizuotų specifikacijos skyrių aprašymo formų;
4. išanalizavus *Volere* šablono sudėtį ir sukūrus jo metamodelį, nustatytos trūkstamos vartotojo specifikavimo procesui būtinos klasės, kuriomis papildytas veiklos metamodelis, apibrėžiantis kuriamos sistemos funkcionalumui būtinas klases;
5. atlikus UML panaudojimo atvejų diagramos metamodelio analizę, nustatyta, kad lyginant su veiklos modeliu ir *Volere* šablono metamodeliu, trūksta tik išplečiančių (angl. *extend*) ir papildančių veiklų (angl. *include*) aprašančiųjų klasių, būtinų šios diagramos generavimui;
6. reikalavimų specifikacijos sudarymo metodas turėtų palengvinti darbą sumažindamas klaidų kiekius dviem naudotojų tipams: sistemų analitikams ir sistemų architektams.

## 2. Formalusis metodo reikalavimų specifikacijos ir projekto aprašas

Skyriuje aptariamas siūlomo metodo aprašas ir sumodeliuotos diagramos, kurios vaizduoja naudotojo reikalavimų specifikacijos sudarymo metodo veikimo principą.

### 2.1. Reikalavimų specifikacija

Metodo reikalavimų specifikaciją pradėdame nuo UML panaudojimo atvejų diagramos. Taip galime įsitikinti, kokios funkcijos yra reikalingos norint įgyvendinti reikalavimų specifikacijos sudarymo metodą, bei kokie naudotojų tipai galės vykdyti šį funkcionalumą.



2.1 pav. Reikalavimų specifikacijos sudarymo metodo UML panaudojimo atvejų diagrama

Kuriamas reikalavimų specifikacijos sudarymo metodas, kuris atvaizduotas 2.1 pav., susideda iš vienos pagrindinės funkcijos – *Sudaryti specifikaciją pasirinktu aspektu* – ir trijų išplečiančiųjų redagavimo funkcijų – *Sukurti elementą*, *Redaguoti elementą* ir *Panaikinti elementą*, ir keturių specifikacijos aspekto pasirinkimo išplečiančiųjų funkcijų – *Sudaryti specifikaciją pagal funkciją*, *Sudaryti specifikaciją pagal funkcijos vykdytoją*, *Generuoti diagramą*, *Sudaryti specifikaciją pagal Volere šabloną*.

Elementas – reikalavimų specifikacijos aprašomasis matas, kuriam priskiriamas *Volere* šablono skyrius. Skyrių informaciją galima prilyginti klasėms, iš kurių bus kuriama duomenų bazės struktūra.

Planuojama, kad šiuo metodu naudosis vienas aktorius – *Analitikas*.

### 2.1.1. Metodo veiklų specifikavimas.

Kiekviena metodo funkcija bus aprašoma pasitelkiant *Volere* šablono aprašomąją panaudojimo atvejų lentelę (1.4 lentelė) ir UML veiklos diagramomis (angl. *UML activity diagram*).

Metodo panaudojimo atvejai aprašomi *Volere* šablonu 2.1 lentelė – 2.4 lentelėse, UML veiklos diagramos atvaizduojamos 2.2 pav. – 2.5 pav.

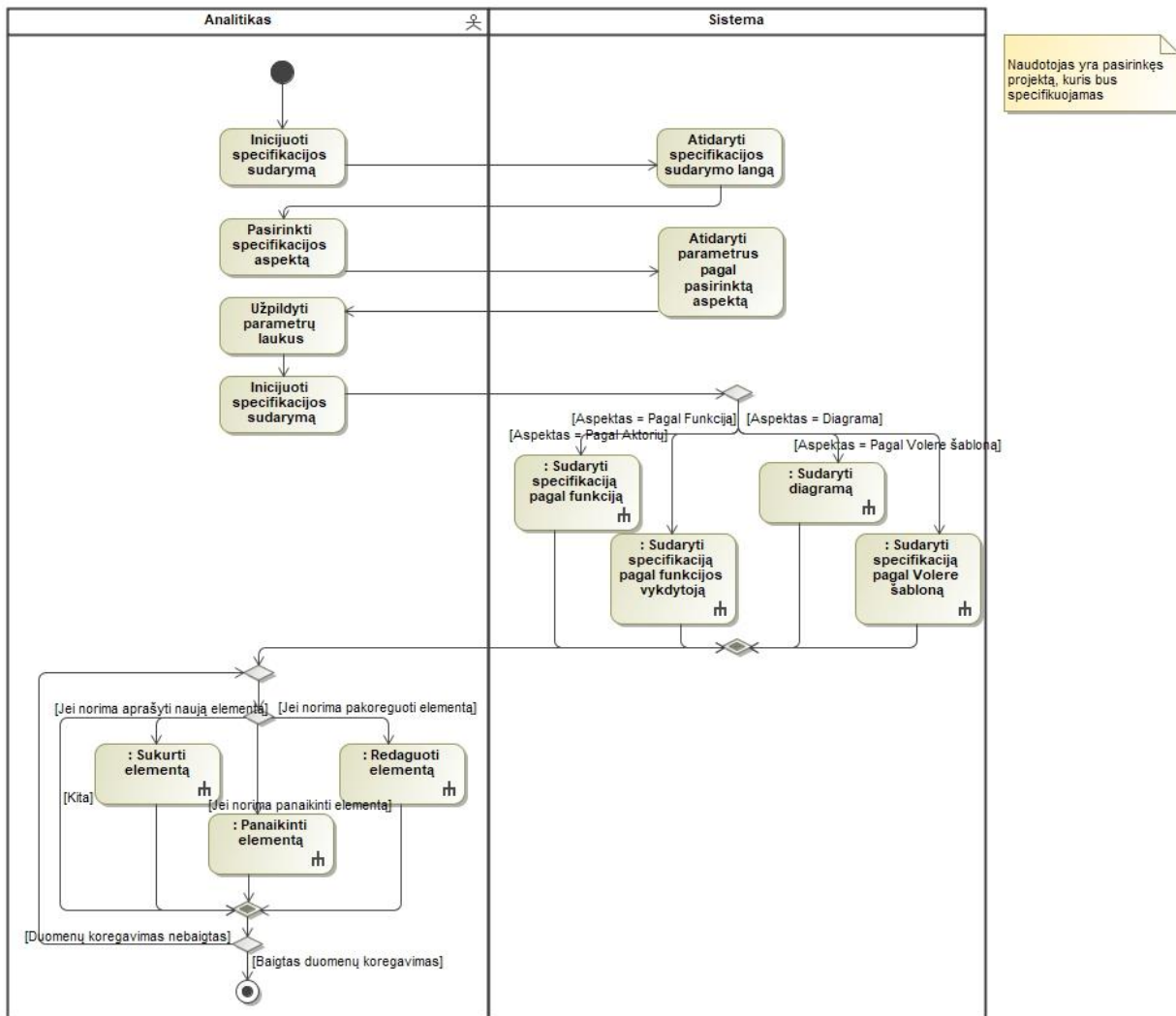
- PA Sudaryti specifikacijos dalį pasirinktu aspektu.

**2.1 lentelė PA Sudaryti specifikaciją pasirinktu aspektu, *Volere* aprašymas**

<b>PA Sudaryti specifikaciją pasirinktu aspektu</b>	
<b>Tikslas/uždavinys.</b> Sudaryti naudotojo reikalavimų specifikaciją pagal pasirinktą aspektą. Taip pat, naudotojui prireikus koreguoti jau išsaugotą specifikaciją.	
<b>Aprašymas.</b> Naudotojas gali sukurti specifikaciją iš veiklos modelio ir iš papildomai pildomos <i>Volere</i> šablono duomenų bazės. Taip pat, turi būti galimybė sudaryti reikalavimų specifikaciją skirtingais aspektais: pagal pasirinktą aktorių, pasirinktą funkciją, pilną arba dalinį <i>Volere</i> šabloną ir norimą UML diagramą.	
<b>Sąlyga prieš</b>	Naudotojas turi būti atsidaręs projekto langą;
<b>Aktorius</b>	Analitikas
<b>Sužadinimo sąlyga</b>	Naudotojas nori sudaryti reikalavimų specifikaciją norimu aspektu; Naudotojas nori redaguoti specifikacijos elementų informaciją; Naudotojas nori papildyti specifikaciją; Naudotojas nori panaikinti esamus užpildytus specifikacijos elementus;
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b> PA Sukurti elementą; PA Redaguoti elementą; PA Panaikinti elementą; PA Sudaryti specifikaciją pagal funkciją; PA Sudaryti specifikaciją pagal funkcijos vykdytoją; PA Generuoti diagramą; PA Sudaryti specifikaciją pagal <i>Volere</i> šabloną;
	<b>Apimami PA</b>
	<b>Apibendrinti PA</b>
<b>Pagrindinis scenarijus – naudotojas nori sudaryti reikalavimų specifikaciją</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
1. Inicijuoti reikalavimų specifikacijos sudarymo lango atidarymą	2. Atidaryti reikalavimų specifikacijos sudarymo langą

3. Pasirinkti specifikacijos aspektą: <ul style="list-style-type: none"> <li>• Pagal Aktorių;</li> <li>• Pagal Funkciją;</li> <li>• Pagal <i>Volere</i> šabloną;</li> <li>• Diagrama;</li> </ul>	
4. Užpildyti laukus	
5. Inicijuoti reikalavimų specifikacijos sudarymą pagal pasirinktą aspektą	6. Pagal pasirinktą reikalavimų specifikacijos aspektą vykdyti atitinkamą išplėčiantį PA;
	7. Atvaizduoti atrinktą specifikaciją pagal aspektą
<b>Rezultatas</b>	Naudotojas matys sudarytą reikalavimų specifikaciją pagal pasirinktą aspektą.
<b>Alternatyvus scenarijus – naudotojas nori redaguoti reikalavimų specifikaciją</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
Naudotojas nori sukurti naują elementą	PA <i>Sukurti elementą</i>
Naudotojas nori redaguoti sukurto elemento informaciją	PA <i>Redaguoti elementą</i>
Naudotojas nori panaikinti elementą	PA <i>Panaikinti elementą</i>
<b>Rezultatas</b>	Duomenų bazės informacija bus pakoreguota.

Panaudojimo atvejo *Sudaryti specifikaciją pasirinktu aspektu* UML veiklos diagrama (2.2 pav.):



2.2 pav. PA Sudaryti specifikaciją pasirinktu aspektu UML veiklos diagrama

- PA Sudaryti specifikaciją pagal funkciją, PA Sudaryti specifikaciją pagal funkcijos vykdytoją, PA Generuoti diagramą, PA Sudaryti specifikaciją pagal *Volere* šablona.

Išplečiančių funkcijų: PA Sudaryti specifikaciją pagal funkciją, PA Sudaryti specifikaciją pagal funkcijos vykdytoją, PA Generuoti diagramą, PA Sudaryti specifikaciją pagal *Volere* šablona, veiksmų seka yra vienoda. Visos jos pagal pasirinktą specifikacijos aspekto tipą atrinka duomenis, kurie yra su juo susiję. Vykdamas PA Sudaryti specifikaciją pagal funkcijos vykdytoją, atrinks informaciją susijusią tik su sistemos naudotojais, kas yra naudotojų informacija, veiklos kontekstai, funkcijos. Jei vykdomas PA Sudaryti specifikaciją pagal *Volere* šablona, atrinkama visa specifikacijos informacija pagal *Volere* šablona.

– PA Sukurti elementą.

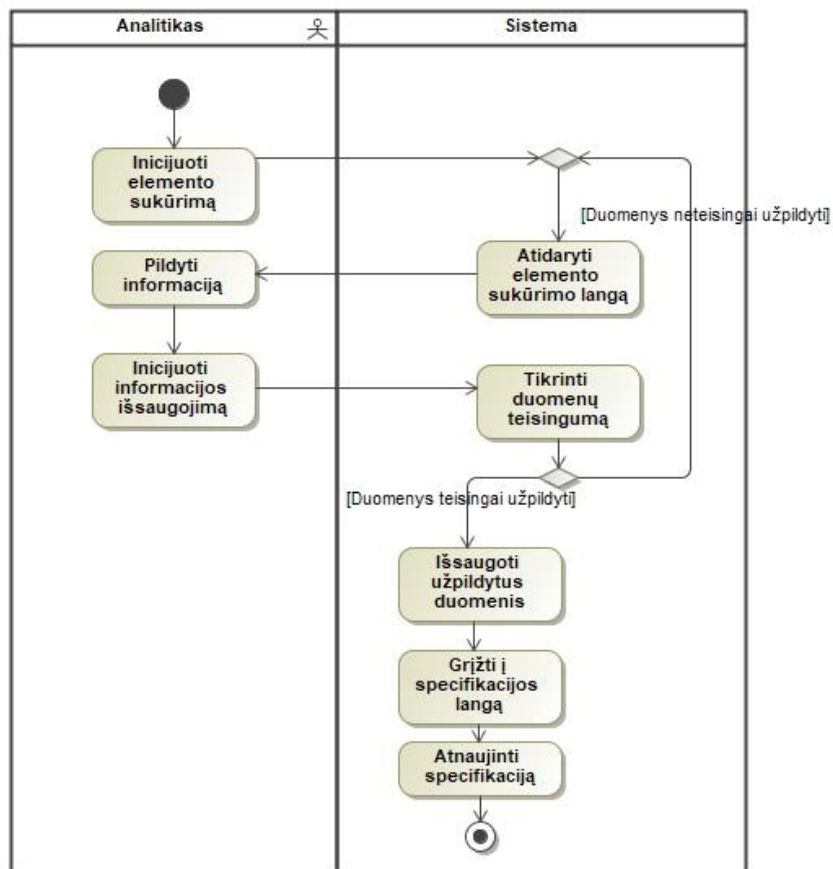
2.2 lentelė PA Sukurti elementą, *Volere* aprašymas

<b>PA Sukurti elementą</b>	
<b>Tikslas/uždavinys.</b> Sukurti naują specifikacijos elementą	
<b>Aprašymas.</b> Turi būti galimybė sukurti naują elementą, aprašomą ne tik <i>Volere</i> šablone, bet ir veiklos modelyje.	
<b>Sąlyga prieš</b>	Sukurta reikalavimų specifikacija
<b>Aktorius</b>	Analitikas
<b>Sužadinimo sąlyga</b>	Naudotojas nori sukurti naujus specifikacijos elementus;
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>
	<b>Apimami PA</b>
	<b>Apibendrinti PA</b>
<b>Pagrindinis scenarijus – elemento duomenys suvesti teisingai</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
1. Inicijuoti naujo elemento sukūrimą	2. Atidaryti elemento sukūrimo langą. Informacijos pildymo laukai pateikiami pagal elemento tipą
3. Pildyti elemento duomenis	
4. Išsaugoti duomenis	5. Tikrinti duomenis
	6. Išsaugoti atitinkamoje duomenų bazėje duomenis
	7. Uždaryti sukūrimo langą
	8. Atnaujinti specifikaciją
<b>Rezultatas</b>	Sukurtas naujas specifikacijos elementas
<b>Alternatyvus scenarijus – elemento duomenys suvesti neteisingai</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
	6a Išsiųsti įspėjimą apie neteisingai suvestus duomenis
	Kartoti 3 – 5o ka žingsnius kol bus teisingai suvesti duomenys
	7a. Išsaugoti duomenis atitinkamoje duomenų bazėje
	8a. Uždaryti sukūrimo langą



	9a. Atnaujinti specifikaciją
<b>Rezultatas</b>	Sukurtas naujas specifikacijos elementas

Panaudojimo atvejo *Sukurti elementą* UML veiklos diagrama (2.3 pav.):



2.3 pav. PA Sukurti elementą UML veiklos diagrama

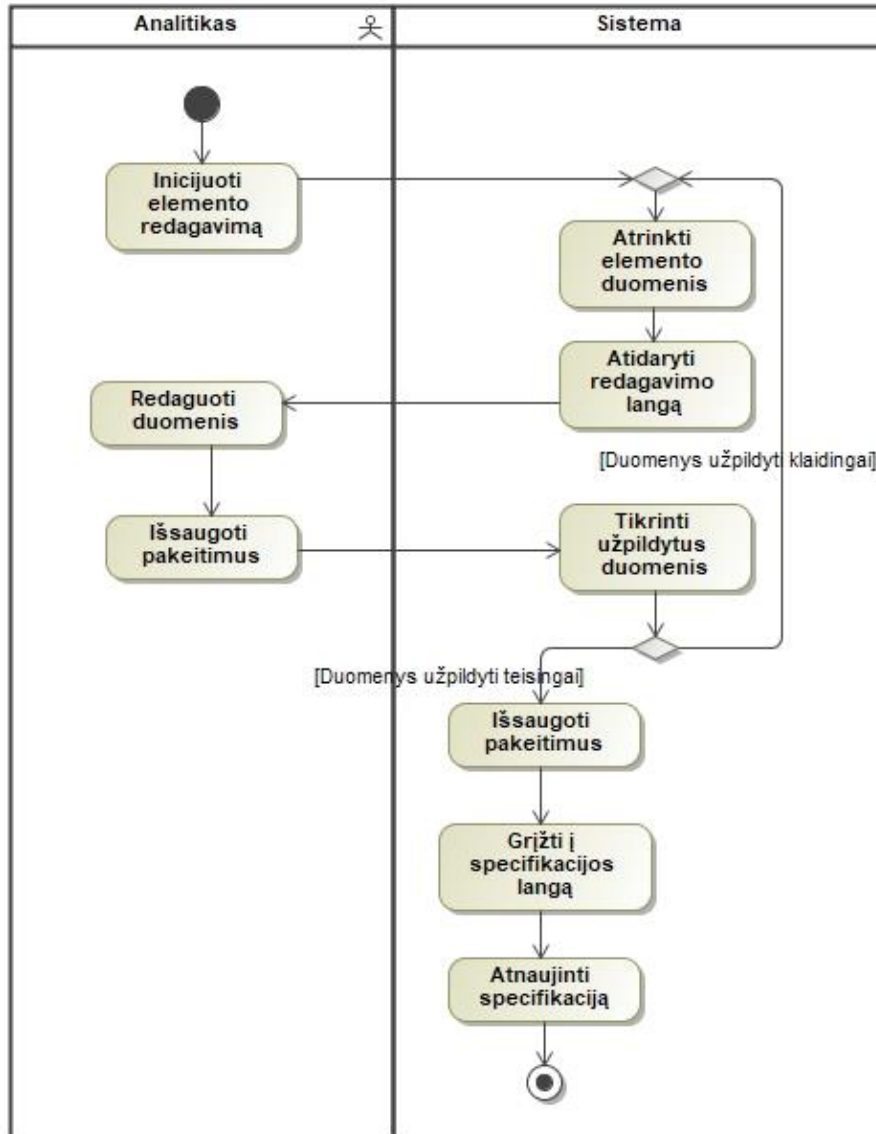
– PA Redaguoti elementą.

### 2.3 lentelė PA Redaguoti elementą, *Volere* aprašymas

<b>PA Redaguoti elementą</b>	
<b>Tikslas/uždavinys.</b> Redaguoti reikalavimų specifikacijos elemento informaciją.	
<b>Aprašymas.</b> Turi būti galimybė redaguoti visų specifikacijos elementų duomenis, kurie saugomi tiek veiklos modelyje, tiek <i>Volere</i> šablone.	
<b>Sąlyga prieš</b>	Turi būti sudaryta reikalavimų specifikacija
<b>Aktorius</b>	Analitikas
<b>Sužadinimo sąlyga</b>	Naudotojas nori redaguoti specifikacijos elementų informaciją;
<b>Susiję panaudojimo</b>	<b>Išplečiantys PA</b>

<b>atvejai</b>	<b>Apimami PA</b>	
	<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – elemento duomenys suvesti teisingai</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
1. Inicijuoti elemento duomenų redagavimą		2. Atidaryti elemento redagavimo langą. Informacijos redagavimo laukai pateikiami pagal elemento tipą
3. Redaguoti duomenis		
4. Išsaugoti duomenis		5. Tikrinti duomenis
		6. Duomenis išsaugoti atitinkamoje duomenų bazėje
		7. Uždaryti redagavimo langą
		8. Atnaujinti specifikaciją
<b>Rezultatas</b>	Elemento duomenys yra pakoreguoti.	
<b>Alternatyvus scenarijus – elemento duomenys suvesti neteisingai</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
		6a Išsiųsti išpėjimą pranešimą apie neteisingai suvestus duomenis
		Kartoti 3-5 žingsnius kol bus teisingai suvesti duomenys
		7a. Duomenis išsaugoti atitinkamoje duomenų bazėje
		8a. Uždaryti redagavimo langą
		9a. Atnaujinti specifikaciją
<b>Rezultatas</b>	Elemento duomenys yra pakoreguoti.	

Panaudojimo atvejo *Redaguoti elementą* UML veiklos diagrama (2.4 pav.):



2.4 pav. PA Redaguoti elementą UML veiklos diagrama

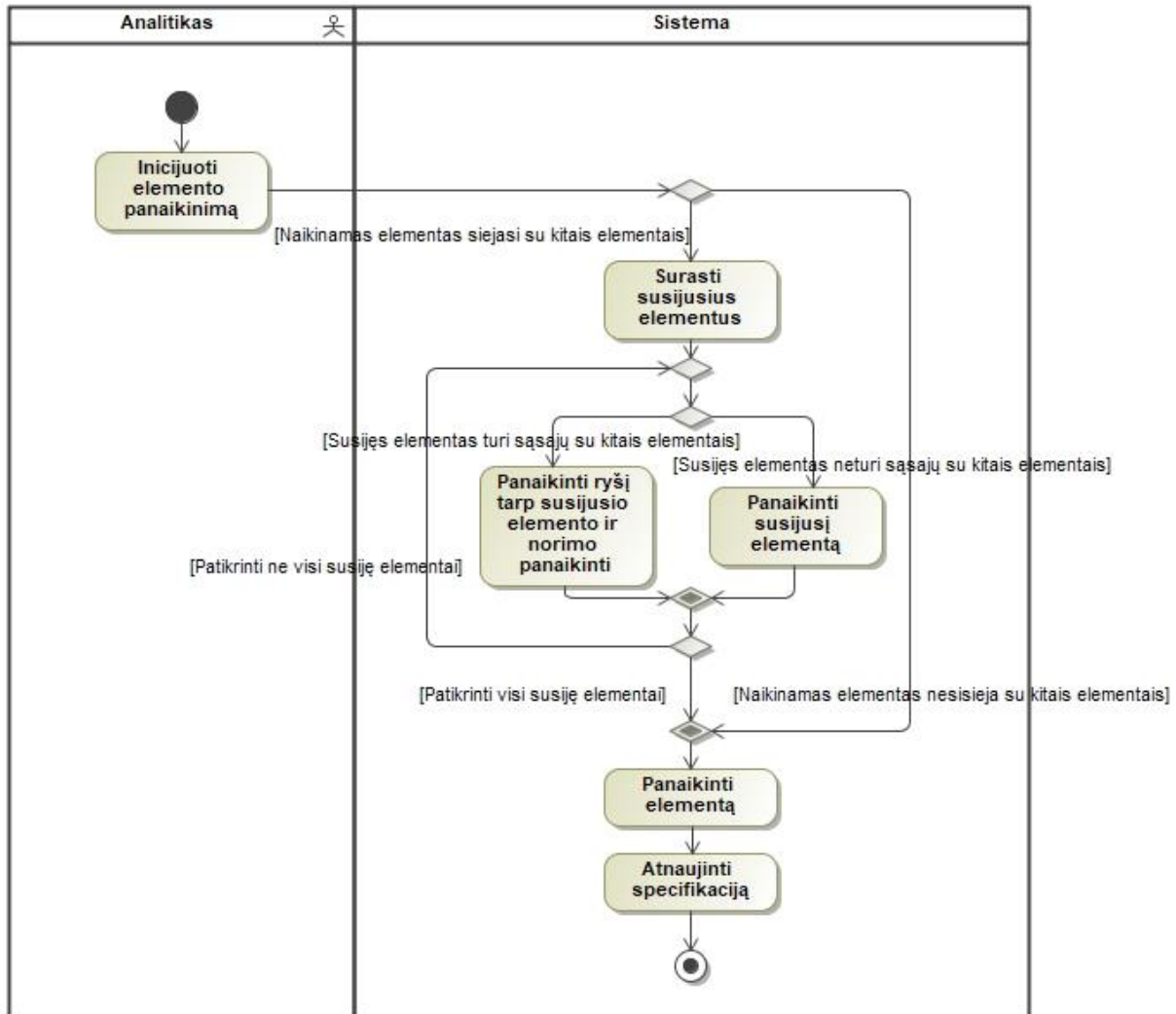
– PA Panaikinti elementą.

2.4 lentelė PA Panaikinti elementą, *Volere* aprašymas

<b>PA Panaikinti elementą</b>	
<b>Tikslas/uždavinys.</b> Panaikinti norimą specifikacijos elementą	
<b>Aprašymas.</b> Panaikinti ne tik norimą elementą, bet ir visus su juo susijusius elementus. Jei susijęs elementas turi sąsajų su kitais reikalavimų specifikacijos elementais, tokiu atveju reikia panaikinti ryšį tarp norimo panaikinti elemento ir susijusio elemento. Elementas iš tikrųjų nėra ištrinamas iš duomenų bazės, jis tik padaromas neaktyviu.	
<b>Sąlyga prieš</b>	Turi būti sudaryta reikalavimų specifikacija

<b>Aktorius</b>	Analitikas	
<b>Sužadavimo sąlyga</b>	Naudotojas nori panaikinti esamus specifikacijos elementus;	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	
	<b>Apimami PA</b>	
	<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – pasirinktas elementas nesisieja su kitais elementais</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
1. Inicijuoti elemento panaikinimą		2. Panaikinti elementą
		3. Atnaujinti specifikaciją
<b>Rezultatas</b>	Panaikinamas norimas elementas iš specifikacijos	
<b>Alternatyvus scenarijus – surasti susijusius elementus, kurie nesisieja su kitais elementais</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
		2a. Panaikinti susijusį elementą
		Kartoti 2a žingsnį, kol bus panaikinti visi susiję elementai
		3a. Panaikinti elementą
		4a. Atnaujinti specifikaciją
<b>Rezultatas</b>	Panaikintas norimas elementas ir su juo susiję elementai iš specifikacijos	
<b>Alternatyvus scenarijus – surasti susijusius elementus, kurie siejasi su kitais elementais</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
		2b. Panaikinti ryšį tarp elemento ir susijusio elemento
		Kartoti 2b žingsnį kol bus panaikinti visi susiję elementai
		3b. Panaikinti elementą
		4b. Atnaujinti specifikaciją
<b>Rezultatas</b>	Iš specifikacijos panaikintas norimas elementas ir ryšiai su susijusiais elementais	

Panaudojimo atvejo *Panaikinti elementą* UML veiklos diagrama (2.5 pav.):



2.5 pav. PA Panaikinti elementą UML veiklos diagrama

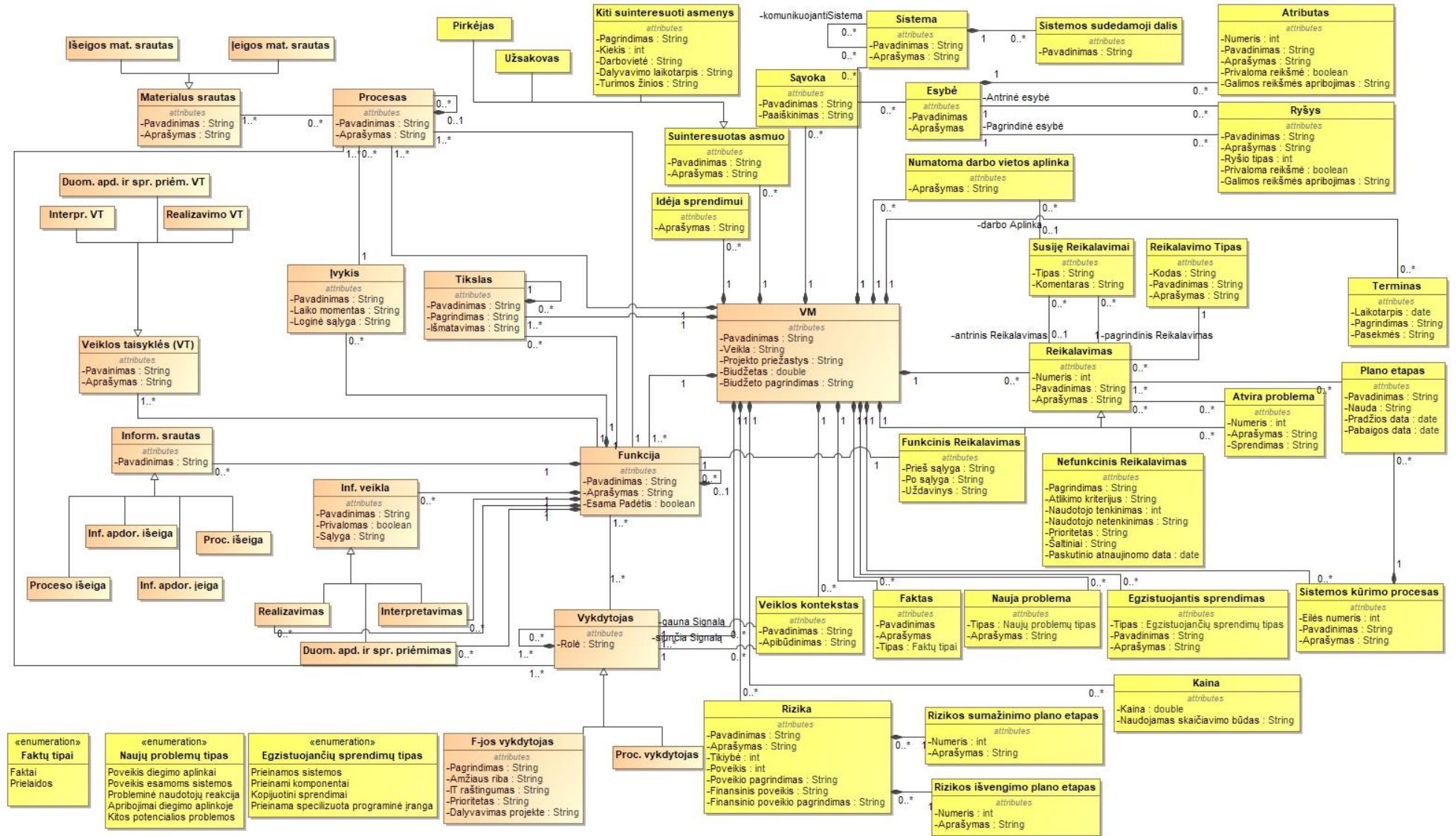
## 2.2. Dalykinės srities modelis

Dalykinės srities modelio kūrimas leidžia detalizuoti iš kokių elementų bus sudaroma reikalavimų specifikacija bei kokie duomenys aprašo kiekvieną elementą. Taip pat matosi, kurie elementai yra priklausomi nuo kurių ir kurie siejasi tarpusavyje.

Planuojama sistemos realizacijoje naudoti dvi atskiras duomenų bazines, kurios tarpusavyje bendrauja, bet norint lengviau parodyti šių duomenų bazių ryšį buvo sukurta bendra UML klasių diagrama. Diagramoje oranžine spalva atvaizduojamos klasės gaunamos iš veiklos modelio, o geltona spalva iš *Volere* šablono metamodelio.

Veiklos modelio klasės bus užpildomos atskirai, todėl sudarant specifikaciją, dalis informacijos bus gaunama automatiškai. O kita dalis, *Volere* šablono klasės, bus užpildomos rankiniu būdu.

Tiek veiklos modelio klasių, tiek *Volere* šablono metamodelio klasių išdėstymas diagramoje pakeistas dėl patogesnio suvokimo. Taip pat, esybės papildytos atributais pagal *Volere* šablono aprašą [14]. Dalykinės srities modelis – 2.6 pav.



2.6 pav. UML klasių diagrama aprašanti galimą duomenų bazės struktūrą

### 2.3. Metodo elementų komunikavimas

Sukuriamos reikalavimų specifikacijos elementų komunikavimą lengvai galima pavaizduoti naudojant UML sekų diagramas (angl. *UML sequence diagram*).

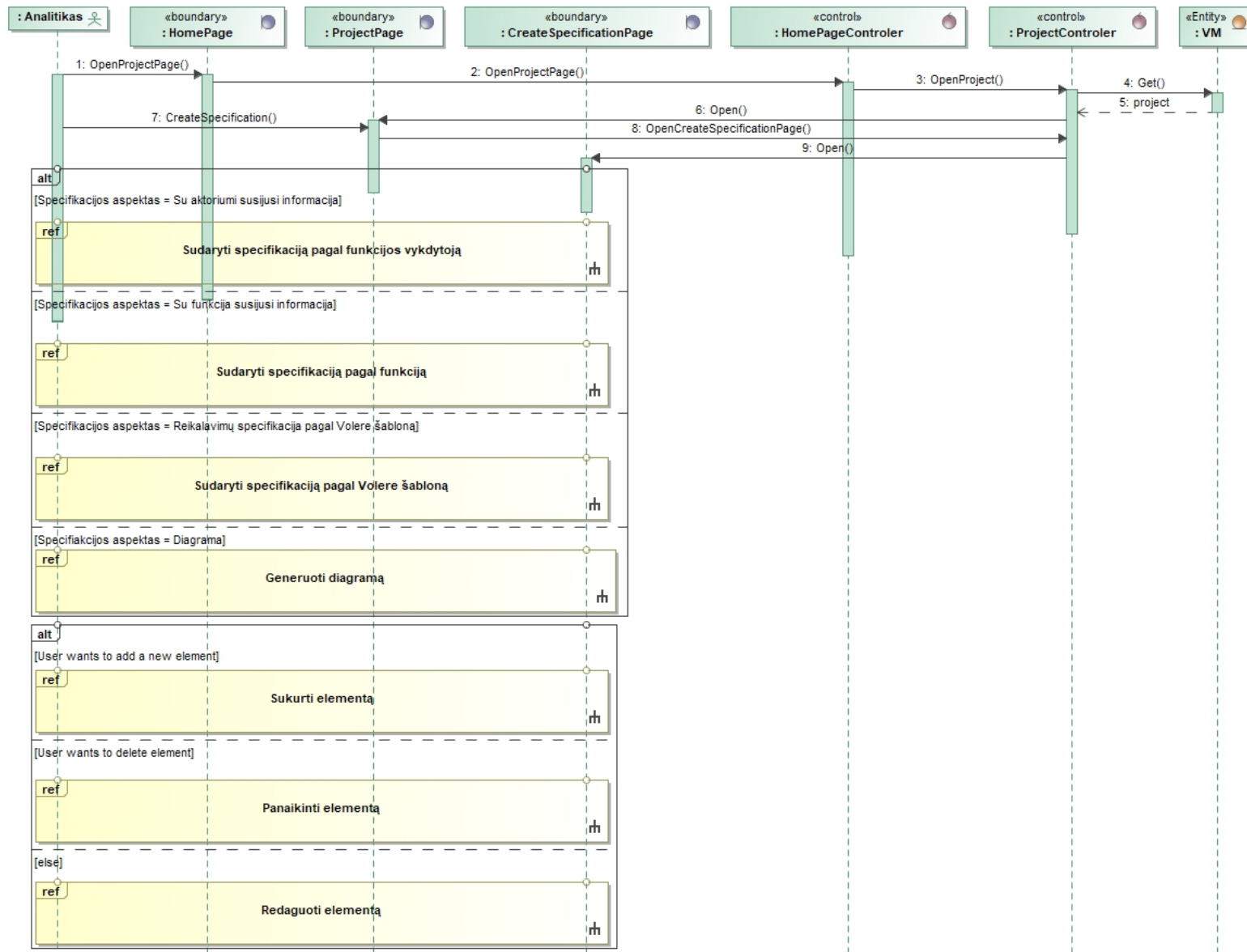
Pirmoji sekų diagrama yra apibendrinamoji – 2.7 pav., kuri susideda iš specifikacijos aspekto pasirinkimo, specifikacijos atvaizdavimo ir galimybės redaguoti specifikaciją.

Yra keturi specifikacijos aspektai, pagal kuriuos galima bus sudaryti reikalavimų specifikaciją:

1. *su aktoriumi susijusi informacija* – atvaizduojami duomenys pagal vieną pasirinktą aktorį;
2. *su funkcija susijusi informacija* – duomenys atvaizduojami pagal pasirinktą vieną funkciją;
3. *reikalavimų specifikacija pagal Volere šabloną* – galima pasirinkti vieną arba daugiau šablono skyrių, pagal kuriuos bus atvaizduojama informacija;
4. *diagramų generavimas* – pasirinkus vieną diagramą, ji yra sugeneruojama.

Aspektus, pagal kuriuos bus sudaroma reikalavimų specifikacija pasieksime atsidarę norimą projektą ir specifikuojimo langą.





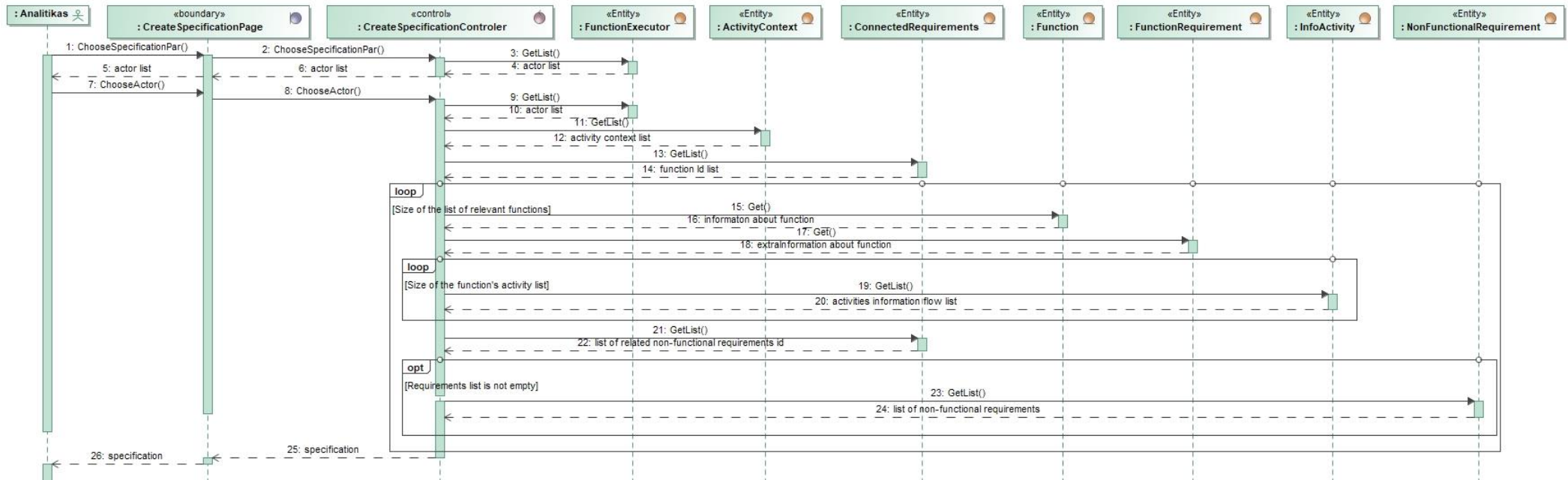
2.7 pav. Apibendrinamoji specifikacijos sudarymo metodo UML sekų diagrama

### 2.3.1. Specifikavimo metodo aspektų sekų diagramos

Žemiau esančios sekų diagramos atvaizduos kiekvieno aspekto detalesnį lygmenį – susijusių elementų atvaizdavimą.

Pirmas aprašomas aspektas – *Su aktoriumi susijusi informacija*, pagal kurį atrenkama visa susijusi informacija su pasirinktu aktoriumi (2.8 pav.). Atrenkama informacija:

- pasirinktas aktorius;
- susiję kiti aktoriai – aktoriai sujungti apibendrinamuoju ryšiu;
- susijusios funkcijos, kurias aktorius gali vykdyti. Taip pat, šių funkcijų veiklos ir susiję reikalavimai.
- veiklos kontekstas, kuriame dalyvauja pasirinktas aktorius.



2.8 pav. Aspekto Su aktoriumi susijusi informacija UML sekų diagrama

Antras aprašomas aspektas – *Su funkcija susijusi informacija* – pagal kurį atrenkama visa susijusi informacija su pasirinkta funkcija. Atrenkama informacija:

- pasirinktos funkcijos informacija;
- aktoriai, kurie gali vykdyti pasirinktą funkciją;
- kitos funkcijos – pasirenkamoji funkcija su kitomis funkcijomis siejasi ryšiais: apibendrinamasis, išplečiamasis ir apimamasis;
- dabartinė situacija – kaip vykdoma ši funkcija be IS;
- susiję nefunkciniai reikalavimai;
- atviros problemos, nuo kurių priklauso pasirinktoji funkcija;
- plano etapai, kuriuose yra nurodyti darbai susiję su pasirinkta funkcija.

Trečias aprašomas aspektas – *Specifikacija pagal Volere šabloną*, kuomet atrenkama reikalavimų specifikacija suskirstyta pagal *Volere* šablono skyrius. Sudarant skyrius naudotas *v18* versijos šablonas. Generuojant specifikaciją šiuo aspektu, galima pasirinkti, kurių šablono skyrių informaciją atrinkti.

Ketvirtas aprašomas aspektas – *Diagramos* – pagal kurį generuojamos UML diagramos. Šiuo metu negalima generuoti visų išvardintų diagramų, nes reikėtų papildyti dalykinės srities veiklos modelį. Šio darbo metu bus generuojama viena diagrama – UML panaudojimo atvejų diagrama. Galimos diagramos:

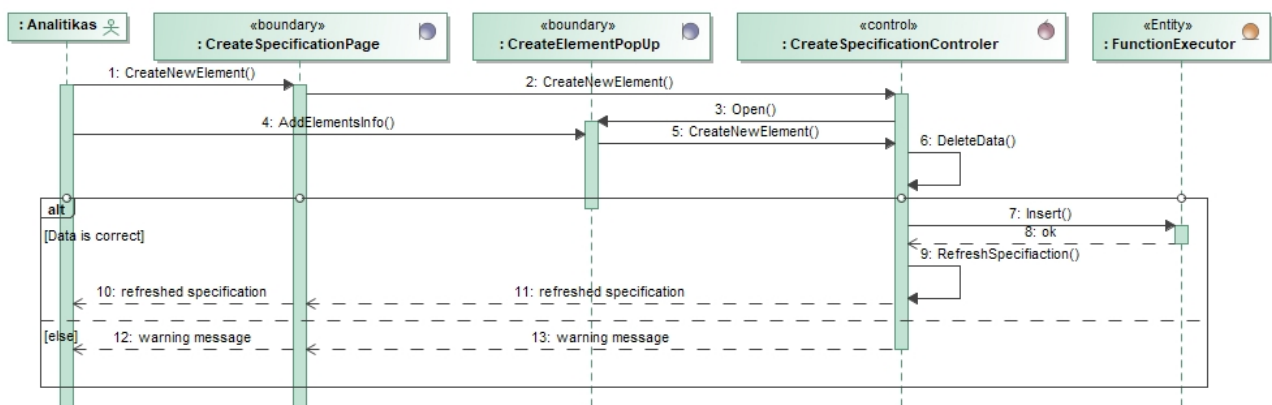
- UML panaudojimo atvejų diagrama;
- UML klasių diagrama;
- UML veiklos diagrama;
- UML diegimo diagrama;
- veiklos kontekstas.

### 2.3.2. Specifikacijos redagavimo sekų diagramos

Redaguoti sugeneruotą specifikaciją galima trimis būdais:

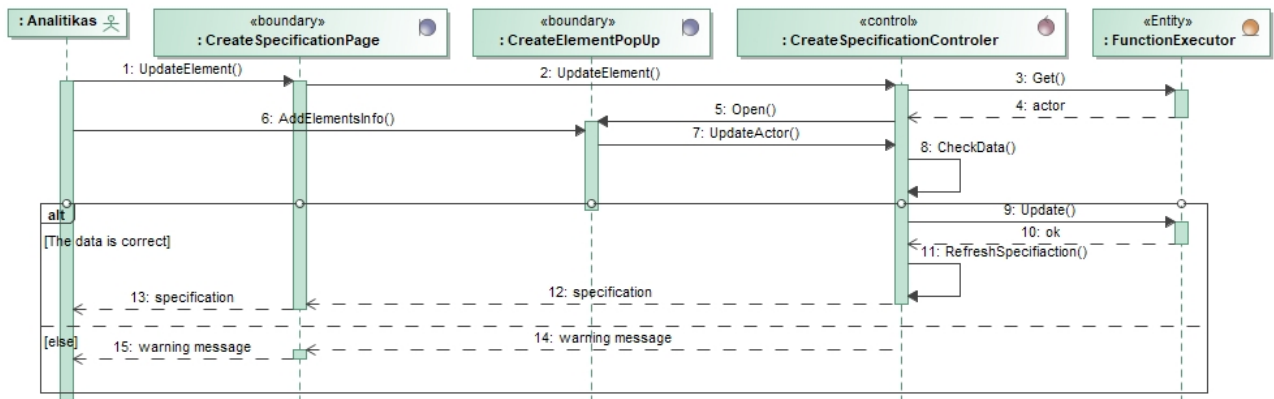
- sukurti naują elementą;
- redaguoti sukurto elemento duomenis;
- panaikinti elementą.

Naujo elemento sukūrimo funkciją galima pasiekti atliktus reikalavimų specifikacijos sudarymą. Pavyzdyje nurodomas naujo *Aktoriaus (f-jos vykdytojas)* sukūrimas (2.9 pav.)



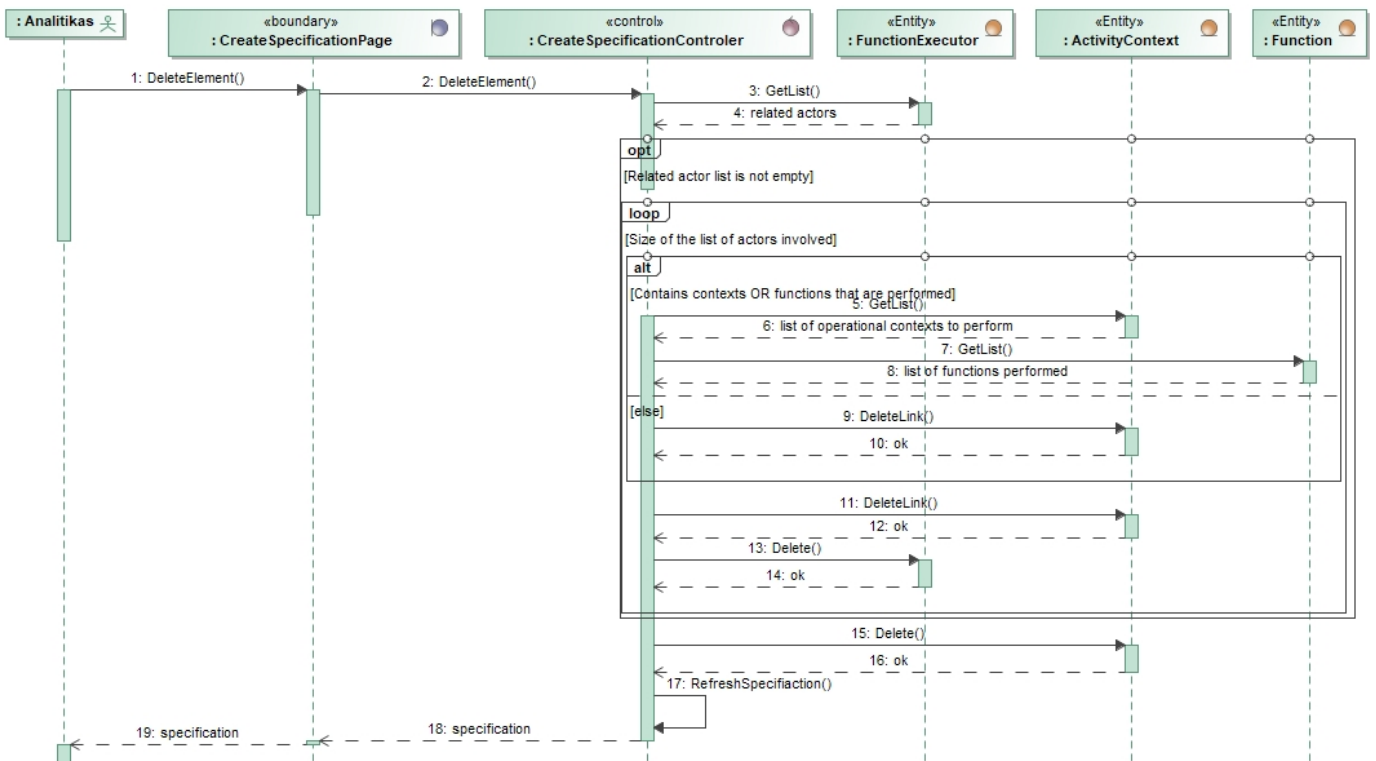
2.9 pav. Funkcijos Sukurti naują elementą UML sekų diagrama

Antroji duomenų redagavimo funkcija labai panaši į elemento sukūrimą, nes naudojamas tas pats informacijos įvedimo iššokantysis langas. Pavyzdyje atvaizduojamas *Aktoriaus (f-jos vykdytojo)* duomenų koregavimas (2.10 pav.).



2.10 pav. Funkcijos Redaguoti sukurto elemento duomenis UML sekų diagrama

Norint panaikinti jau sukurtą elementą iš pradžių reikia patikrinti ar elementas turi sąsajų su kitais elementais. Kuomet susiję elementai siejasi net tik su norimu ištrinti elementu, panaikinama sąsaja tik tarp normų panaikinti elementų, bet jei susiję elementai nesisieja su kitais, panaikinami ne tik elementai bet ir su jais susiję elementai. Pavyzdyje atvaizduojamas *Veiklos konteksto* panaikinimas (2.11 pav.).



2.11 pav. Funkcijos Panaikinti elementą UML sekų diagrama

## 2.4. Naudotojų sąsajos modelis

Naudotojų sąsajos modelio prototipai buvo kuriami naudojantis *MagicDraw* programa. Eskizams naudotos specialius diagramų tipas – naudotojo sąsajos modeliavimo diagrama (angl. *user interface*

modeling diagram). Dėl prototipų diagramų apimties, kai kurios iš jų bus skaidomos ir aprašomos atskirai.

Atsidariusiame specifikacijos lange iš pradžių matysis tik vienas laukas pavadinimu *Specifikacijos aspektas* su keturiais galimais pasirinkimais:

- su aktoriumi susijusi informacija;
- su funkcija susijusi informacija;
- pagal *Volere* šabloną;
- diagrama.

Kai specifikacija yra sudaryta, naudotojas ne tik gali peržiūrėti, bet ir redaguoti visą matomos specifikacijos informaciją.

Šiame skyriuje atvaizduoti laukai skirti aspektui *Volere* šablonas. Atvaizduojamas nepilnas šablonas, tik tie skyriai, kurių informacija atvaizduojama itin struktūrizuotomis lentelėmis.

Antrasis šablono skyrius – *Suinteresuotieji asmenys* – aprašo visus asmenis, kurie vienaip ar kitaip gali daryti įtaką sistemos kūrimui. Aprašomi užsakovai, pirkėjai, sistemos kūrėjai, būsimi naudotojai ir su jais susijusi informacija bei sistemos prižiūrėtojai (2.12 pav.).

V2c. Kiti suinteresuoti asmenys

Pavadinimas	Aprašymas	Dalyvaujančių projekte kiekis	Darbovietė	Dalyvavimo laikotarpis	Turimos žinios
Suinteresuotas asmuo 1	Aprašymas 1	Sveikasis skaičius	Darbovietė1	Tikslus laikotarpio aprašymas 1	Turimos žinios 1
Suinteresuotas asmuo 2	Aprašymas 2	Sveikasis skaičius	Darbovietė1	Tikslus laikotarpio aprašymas 2	Turimos žinios 1
Suinteresuotas asmuo 3	Aprašymas 3	Sveikasis skaičius	Darbovietė2	Tikslus laikotarpio aprašymas 3	Turimos žinios 1

V2d. Potencialūs produkto naudotojai

Pavadinimas	Pagrindimas	Amžiaus ribos	IT raštingumas
Aktorius 1	Pagrindimas 1	Sveikų skaičių laikotarpis	IT raštingumo aprašymas 1
Aktorius 2	Pagrindimas 2	Sveikų skaičių laikotarpis	IT raštingumo aprašymas 2
Aktorius 3	Pagrindimas 3	Sveikų skaičių laikotarpis	IT raštingumo aprašymas 3

2.12 pav. Naudotojo sąsajos prototipas. Suinteresuotieji asmenys

Šeštasis šablono skyrius – *Veiklos sudėtis* – aprašo procesus, kuriuos pakeis sistemos funkcijos. Esama padėtis atvaizduojama naudojantis UML veiklos diagramomis, o veiklos kontekstas – laisvos struktūros diagrama (2.13 pav.).

V6c. Veiklos suskaidymas

Duomenų srauto pavadinimas	Perduoda	Gauna	Apibūdinimas
Duomenų srautas	Aktorius 1	Procesas	Apibūdinimas
	Aktorius 3	Aktorius 1	
Duomenų srautas1	Aktorius 3	Procesas	Apibūdinimas 2
Duomenų srautas3	Procesas	Aktorius 2	Apibūdinimas 3

2.13 pav. Naudotojo sąsajos prototipas. Veiklos sudėtis

Septintasis šablono skyrius – *Duomenų modelis ir jo elementų žodynas* – atvaizduoja būsimos duomenų bazės esybes ir jų informaciją. Esybės ir jų ryšiai atvaizduojami naudojant UML klasių

diagramas, o likusi informacija aprašoma lentele, kurioje ne tik aprašomi esybių aprašymai, atributų pavadinimai ir tipai, bet taip pat ir galimos jų reikšmės apribojimai (2.14 pav.).

V7b. Duomenų žodynas (duomenų modelio specifikacija)

Esybė			
Esybė1		Aprašymas	
Atributai			
Nr.	Pavadinimas	Aprašymas	Galimos reikšmės apribojimai
1.	Atributas 1	Aprašymas 1	Tipas; Pradinės reikšmės;
2.	Atributas 2	Aprašymas 2	Tipas; Pradinės reikšmės;
Ryšiai			
Nr.	Tipas	Su kuria klase	Aprašymas
1.	Sąrašas	Esybė2	Aprašymas 2
2.	Asociacija	Esybė3	Aprašymas 1

2.14 pav. Naudotojo sąsajos prototipas. Duomenų modelis

Aštuntasis šablono skyrius – *Sistemos sudėtis* – atvaizduoja UML panaudojimo atvejų diagramą ir kiekvienos funkcijos aprašymą lentelės forma (1.4 lentelė). (2.15 pav.)

V8b Panaudojimo atvejai  
Kiekvienam panaudojimo atvejui užpildoma lentelė

PA		
Tikslas/uždavinys.		
Aprašymas.		
Prieš-sąlyga		
Aktorius		
Sužadinimo sąlyga		
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuoti PA	
Pagrindinis scenarijus Pavadinimas		
Naudotojo veiksmai		Sistemos veiksmai
Po-sąlyga		
Alternatyvūs scenarijai Pavadinimas		

2.15 pav. Naudotojo sąsajos prototipas. Sistemos sudėtis

Devintasis – septynioliktas, dvidešimt antras, dvidešimt penktas ir dvidešimt šablono skyriai aprašo skirtingų tipų sistemos reikalavimus. Visi reikalavimai aprašomi ta pačia forma naudojant – 1.3 lentelę. (2.16 pav.)

Reikalavimas #:	Reikalavimo tipas:	PA/FR#:
Aprašymas:		
Pagrindimas:		
Šaltinis:		
Atlikimo kriterijus:		
Naudotojo tenkinimas:	1-5	Naudotojo netenkinimas: 1-5
Prioritetas:		Konfliktai:
Šaltiniai:		
Paskutinio atnaujinimo data:		

2.16 pav. Naudotojo sąsajos prototipas. Reikalavimai

Aštuonioliktas šablono skyrius – *Atviros problemos ir klausimai* – aprašo problemas, kurios dar neturi sprendimo. Jos aprašomos, norint išvengti sistemos trikdžių (2.17 pav.).

<b>V18. Atviros problemos ir klausimai</b>	
Problemos numeris	
Paveikia	
Problemos santrauka	
Susiję suinteresuoti asmenys	
Sprendimas	

2.17 pav. Naudotojo sąsajos prototipas. Atviros problemos ir klausimai

Dvidešimt pirmas šablono skyrius – *Uždaviniai* – aprašo procesą, kurio bus laikomasi kuriant sistemą. Taip pat, nurodomas sistemos kūrimo tvarkaraštis (2.18 pav.).

<b>V21. Uždaviniai</b>		
<b>V21a. Sistemos kūrimo procesas</b>		
Eilės nr.	Pavadinimas	Aprašymas
1	Pavadinimas 1	Aprašymas 1
2	Pavadinimas 2	Aprašymas 2
3	Pavadinimas 3	Aprašymas 3
<b>V21b. Detalus kūrimo planas</b>		
Kūrimo procesas	Etapas	Laikotarpis
Pavadinimas 1	Etapas 1	
	Etapas 2	
Pavadinimas 2	Etapas 3	
	Etapas 4	
Pavadinimas 3	Etapas 5	

2.18 pav. Naudotojo sąsajos prototipas. Uždaviniai

Dvidešimt trečias šablono skyrius – *Rizikų įvertinimas* – aprašomos galimos rizikos bei planai, kaip išvengti jų ir kaip sumažinti jų poveikį (2.19 pav.).

<b>V23. Rizikų įvertinimas</b>	
Pavadinimas	
Rizikos faktoriaus aprašymas	
Rizikos faktoriaus tikimybė	
Rizikos faktoriaus poveikis (iki 10)	
Finansinis rizikos faktoriaus poveikis	
Planas išvengti poveikio arba minimizuoti rizikos faktorių (kai jis dar neįvyko)	1. 2. 3.
Planas, kaip sumažinti rizikos faktoriaus poveikį (kai jis jau įvyko)	1. 2.

2.19 pav. Naudotojo sąsajos prototipas. Rizikų įvertinimas

## 2.5. Formalus sprendimo aprašas

Pusiau automatinis reikalavimų specifikacijos sudarymo algoritmas susideda iš pagrindinių šių žingsnių:

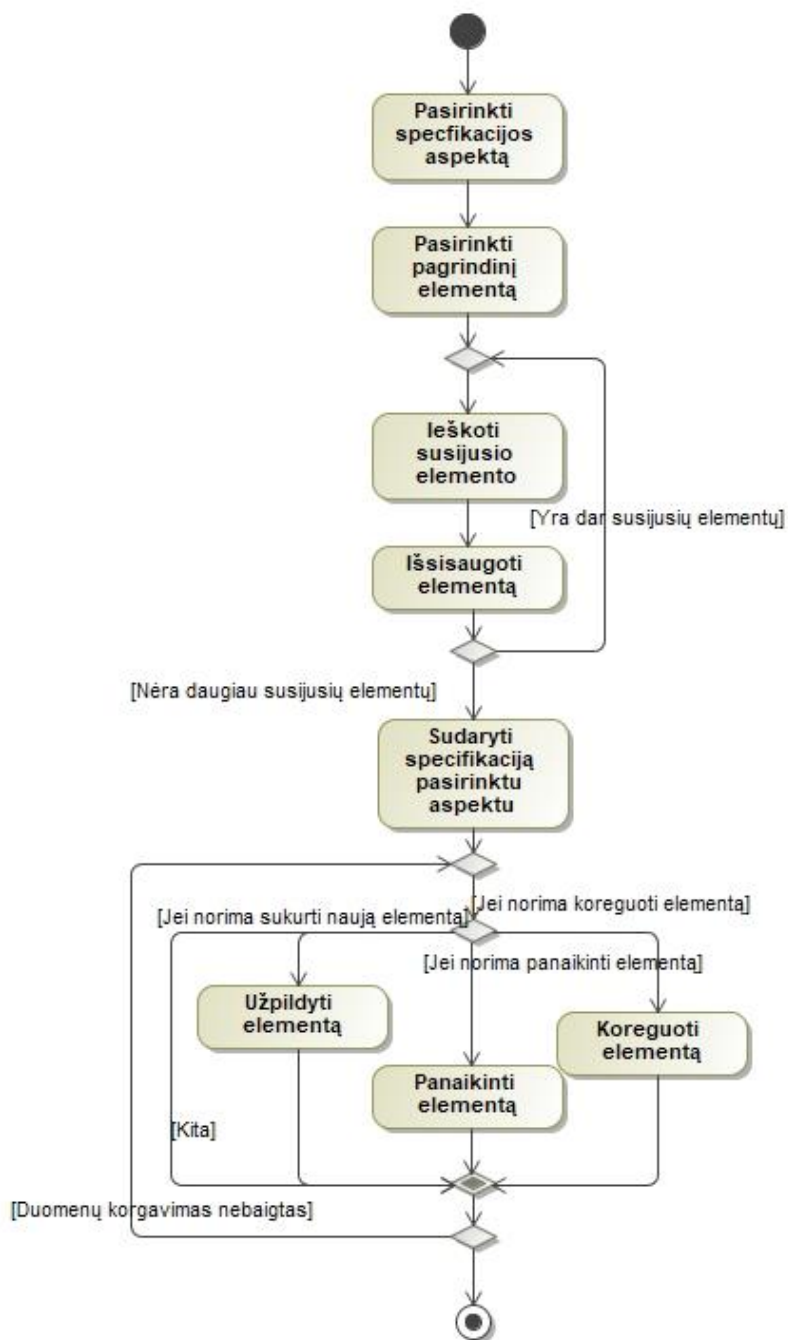
1. pasirinkti aspektą;



2. pasirinkti aspekto elementą – dažniausiai tipas prilyginamas pradiniam specifikacijos elementui;
3. ieškoti elementų pagal aspekto turimą susijusių elementų sąrašą;
4. atvaizduoti rastus elementus.

Metodas yra pusiau automatinis, nes dalis informacijos yra gaunama iš VM, dalis informacijos – diagramos – gali būti sugeneruota, o likusią dalį analitikas turi pats užpildyti. Taip pat yra galimybė sudarius specifikaciją, koreguoti elementų informaciją.

Specifikacijos sudarymo metodo makro-algoritmas – 2.20 pav.



2.20 pav. Specifikacijos metodo makro-algoritmas

Žemiau pateikiama lentelė, kurioje nurodomos VM priklausomybės *Volere* šablonui klasių lygmenyje.

### 2.5 lentelė VM klasių priklausomybė su *Volere* šablonu

Šablono sudėtis	VM klasės	<i>Volere</i> klasės
<b>1 Sistemos paskirtis</b>		
<b>1a Analizuojama veikla ir/ar projekto priežastys</b>	VM	Projektas
<b>1b Tikslai</b>	Tikslas	Tikslas
<b>2 Suinteresuoti asmenys</b>		
<b>2a Užsakovas</b>	-	Užsakovas
<b>2b Pirkėjas</b>	-	Pirkėjas
<b>2c Kiti suinteresuoti asmenys</b>	-	Kiti suinteresuoti asmenys
<b>2d Potencialūs produkto naudotojai</b>	Funkcijų valdytojas	Naudotojas
<b>2e Personazai</b>	-	Naudotojas
<b>2f Naudotojams priskirti prioritetai</b>	Funkcijų valdytojas	Naudotojas
<b>2g Naudotojų dalyvavimas projekte</b>	Funkcijų valdytojas	Naudotojas
<b>2h Sistemos aptarnaujantis personalas</b>	-	Kiti suinteresuoti asmenys
<b>3 Apribojimai</b>		
<b>3b Diegimo aplinka</b>	-	Sistema
<b>4 Sąvokos ir santrumpos</b>	-	Sąvoka
<b>5 Svarbūs faktai ir prielaidos</b>		
<b>5a Faktai</b>		Faktas
<b>5b Veiklos taisyklės</b>	Veiklos taisyklės	Faktas
<b>5c Prielaidos</b>	-	Faktas
<b>6 Veiklos sudėtis</b>		
<b>6a Esama padėtis</b>	Funkcija	Funkcinis reikalavimas
<b>6b Veiklos kontekstas</b>	-	Veiklos kontekstas
<b>6c Veiklos suskaidymas (įvykių specifikacija)</b>	-	Veiklos kontekstas
<b>7 Duomenų modelis ir jo elementų žodynas</b>		

<b>7a Duomenų modelis</b>	-	ER modelis
<b>7b Duomenų žodynas</b>	-	ER modelis
<b>8 Sistemos sudėtis (panaudojimo atvejų modelis)</b>		
<b>8a Sistemos ribos (panaudojimo atvejų diagrama)</b>	Funkcija, Funkcijos vykdytojas	Funkcinis reikalavimas, Naudotojas
<b>8b Panaudojimo atvejai</b>	Funkcija	Funkcinis reikalavimas
<b>9 Papildomi funkciniai reikalavimai</b>	-	Funkcinis reikalavimas
<b>10 – 17 Nefunkciniai reikalavimai</b>	-	Nefunkcinis reikalavimas
<b>18 Atviros problemos ir klausimai</b>	-	Problema
<b>19 Egzistuojantys sprendimai</b>	-	Egzistuojantys sprendimai
<b>20 Naujos problemos</b>	-	Problema
<b>21 Uždaviniai</b>	-	Uždaviniai, Sistemos kūrimo procesas
<b>22 Migravimas į naują produktą</b>	-	Nefunkcinis reikalavimas
<b>23 Rizikų įvertinimas</b>	-	Rizika
<b>24 Kaštai</b>	-	Kaštai
<b>25 Naudotojo dokumentacija ir apmokymai</b>	-	Nefunkcinis reikalavimas
<b>26 Perspektyviniai reikalavimai</b>	-	Nefunkcinis reikalavimas
<b>27 Idėjos sprendimams</b>	-	Idėja sprendimui

Buvo išskirtos 35 specifikacijos dalys, iš kurių 9 galima padengti VM. Vieni skyriai buvo labiau paskaidyti, kiti mažiau. Tai priklausė nuo kuriose duomenų bazės lentelėse yra saugojama *Volere* šablono skyriaus informacija.

## 2.6. Reikalavimų apibendrinimas

Atlikus kuriamo metodo reikalavimų specifikacijos aprašą prieita prie šių išvadų:

1. kuriamas vartotojo reikalavimų specifikacijos sudarymo metodas optimizuoja ir struktūrizuoja specifikacijos sudarymo procesą, taikant veiklos modelį kaip pradinį (pagrindinį) dalykinės srities žinių šaltinį;
2. atlikus veiklos ir *Volere* šablono metamodelių analizę nustatyta, kad pusiau automatinis vartotojo reikalavimų specifikacijos sudarymo metodas naudoja elementus, kurie gali būti keturių būsenų: a) papildomai panaudojant duomenis iš veiklos modelio, b) analitikas įveda informaciją neautomatizuotu būdu, c) dalis informacijos panaudojama iš VM, kita dalis įvedama analitiko, d) grafinių modelių generavimas;

### 3. Eksperimentinės realizacijos projektas

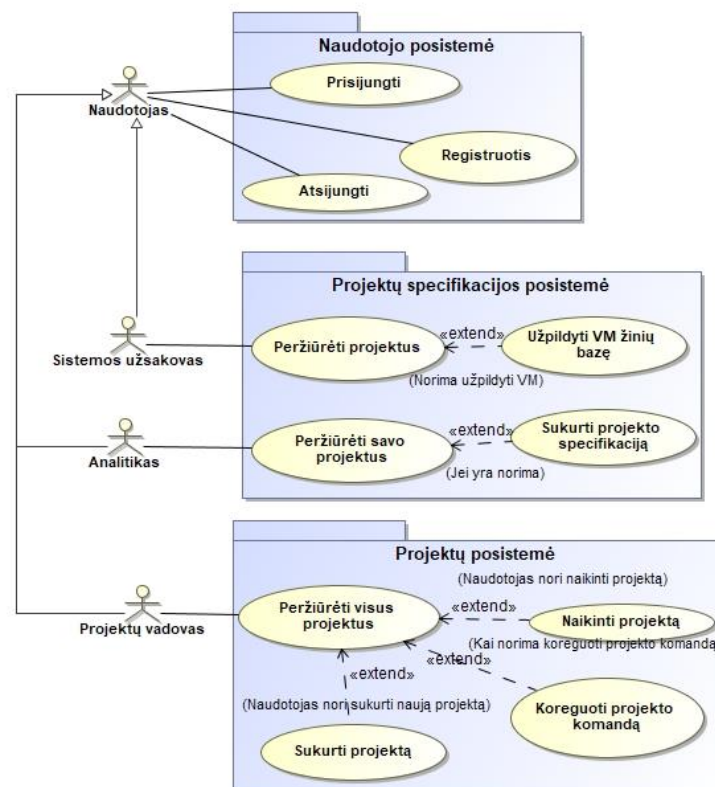
Norint įsitikinti, kad siūlomas naudotojo reikalavimų specifikacijos sudarymo metodas tinkamai veikia ir yra gaunami patenkinami rezultatai buvo sukurtas realizacijos projektas. Šiame skyriuje bus aptariamoms sukurtos sistemos duomenų bazės, funkcijos ir jų realizacija.

#### 3.1. Sistemos reikalavimų specifikacija

Sistemos reikalavimų specifikaciją pradėsime nuo UML panaudojimo atvejų (3.1 pav.), kuri sumodeliuota naudojant *MagicDraw* programa..

Diagramoje matome, kad sistema yra sudaryta iš trijų posistemų:

- naudotojo – bendrinės sistemos funkcijos, kurios apima prisijungimą prie sistemos ir savo naudotojo sukūrimą;
- projektų specifikacijos – pagrindinės sistemos funkcijos. Šioje posistemėje yra realizuojamas reikalavimų specifikacijos sudarymo metodas, prisijungusio naudotojo projektų atvaizdavimo funkcija ir veiklos modelio užpildymas;
- projektų – funkcijos skirtos projektų vadovui, su kuriomis galima sukurti naujus projektus, priskirti jiems darbuotojus.



3.1 pav. Eksperimentinės sistemos UML panaudojimo atvejų diagrama

#### 3.1.1. Realizuojamos sistemos veiklos.

Visos realizuojamos sistemos funkcijos šiame skyriuje bus aprašomos *Volere* šablono lentele (1.4 lentelė) (3.1 lentelė– 3.7 lentelė) ir UML veiklos diagrama (3.2 pav.– 3.8 pav.). Vienintelė funkcija,

kuri nebus papildomai aprašoma yra *Sukurti projekto specifikaciją*, nes šis panaudojimo atvejis atitinka specifikacijos sudarymo metodą, kuris buvo aprašomas 2 skyriuje.

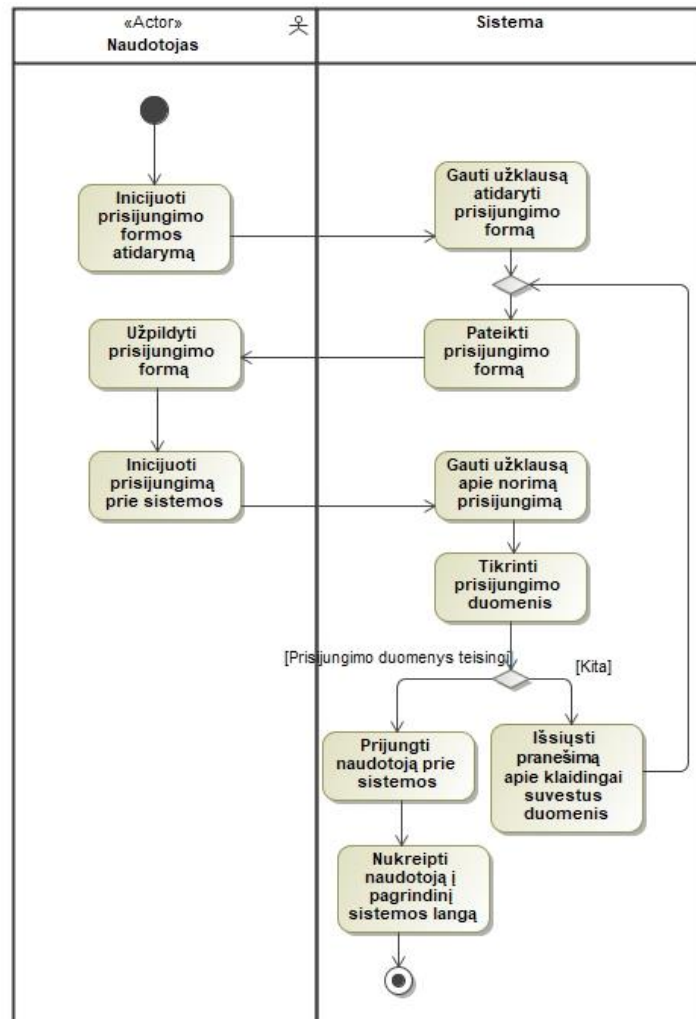
– PA Prisijungti

Prisijungimo prie sistemos funkcija aprašoma 3.1 lentelėje.

### 3.1 lentelė PA Prisijungti Volere aprašymas

<b>PA Prisijungti</b>	
<b>Tikslas/uždavinys.</b> Sukurti naudotojo sesiją sistemoje.	
<b>Aprašymas.</b> Naudotojas gali prisijungti prie sistemos ir vykdyti jos funkcijas turėdamas prisijungimo vardą ir slaptažodį.	
<b>Sąlyga prieš</b>	Naudotojas turi būti užsiregistravęs sistemoje.
<b>Aktorius</b>	Naudotojas.
<b>Sužadavimo sąlyga</b>	Naudotojas nori prisijungti prie sistemos.
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>
	<b>Apimami PA</b>
	<b>Apibendrinti PA</b>
<b>Pagrindinis scenarijus – naudotojas teisingai įveda duomenis</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
1. Inicijuoti prisijungimo formos atidarymą.	2. Atidaryti prisijungimo formą.
3. Užpildyti prisijungimo formą: <ul style="list-style-type: none"> <li>• Prisijungimo vardas;</li> <li>• Slaptažodis;</li> </ul>	
4. Inicijuoti prisijungimą.	5. Tikrinti prisijungimo duomenis.
	6. Prijungti naudotoją prie sistemos.
	2. Nukreipti naudotoją į pagrindinį sistemos puslapį.
<b>Rezultatas</b>	Naudotojas yra prijungiamas prie sistemos ir mato pagrindinį sistemos puslapį.
<b>Alternatyvus scenarijus – naudotojas klaidingai įveda duomenis</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
	6a. Išsiųsti pranešimą apie klaidingai įvestus duomenis.
	Kartoti žingsnius nuo anksčiau minėto 3 punkto, kol teisingai bus įvesti visi duomenys.
<b>Rezultatas</b>	Naudotojas yra prijungiamas prie sistemos ir mato pagrindinį sistemos puslapį.

UML veiklos diagrama atvaizduojanti prisijungimo funkcijos veiklas – 3.2 pav.



3.2 pav. Funkcijos Prisijungti UML veiklos diagrama

– PA Registruotis

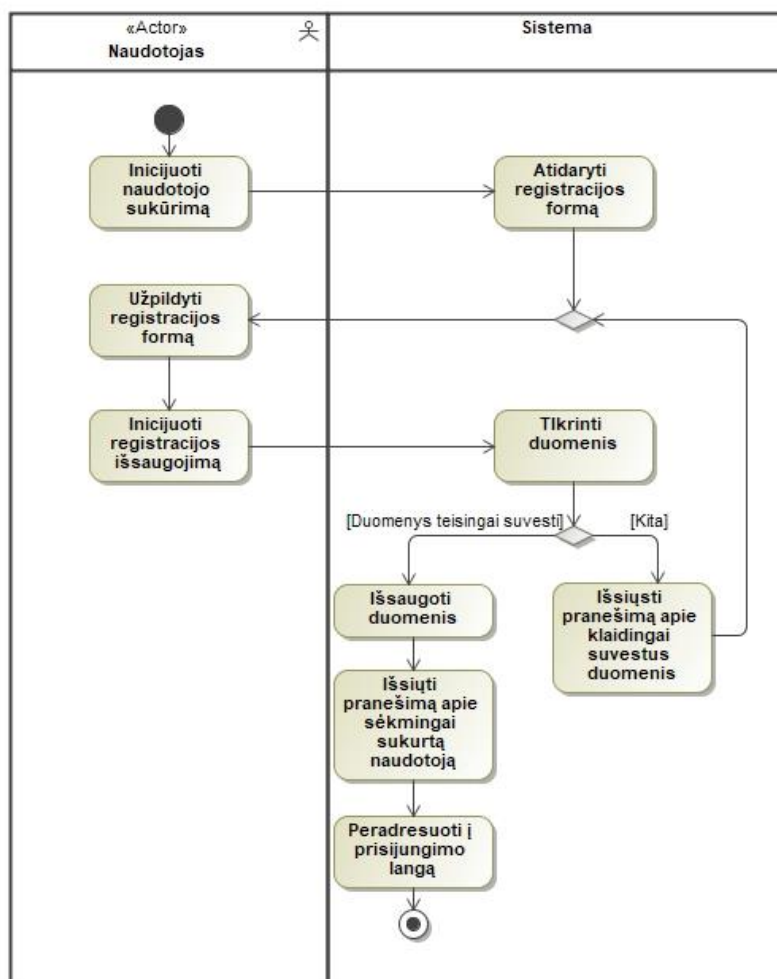
Norint naudoti sistemos funkcijas reikia galėti prisijungti prie sistemos. Šį rezultatą leidžia pasiekti funkcija *Registruotis*, kuri aprašoma 3.2 lentelėje. Sukurtoje sistemoje, kuri leis patikrinti metodo veikimą naudotojas turi pats sukurti savo naudotoją sistemoje.

3.2 lentelė PA Registruotis *Volere* aprašymas

<b>PA Registruotis</b>	
<b>Tikslas/uždavinys.</b> Sukurti naują sistemos naudotoją.	
<b>Aprašymas.</b> Naudotojas norėdamas prisijungti prie sistemos, visu pirma turi užsiregistruoti. Tai atliekama susigalvojus savo prisijungimo vardą ir slaptažodį ir juos išsaugojant sistemoje.	
<b>Sąlyga prieš</b>	Naudotojas dar neturi prisijungimo prie sistemos.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Naudotojas nori galėti prisijungti prie sistemos.
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>
	<b>Apimami PA</b>

		<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – naudotojas teisingai įvedė duomenis</b>			
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>	
1. Inicijuoti naudotojo sukūrimą.		2. Atidaryti registracijos formą.	
3. Užpildyti registracijos formą: <ul style="list-style-type: none"> <li>• Prisijungimo vardas;</li> <li>• Slaptažodis;</li> <li>• Pakartoti slaptažodį;</li> </ul>		4. Tikrinti duomenis.	
		5. Išsaugoti duomenis.	
		6. Išsiųsti pranešimą apie sėkmingai sukurtą naudotoją.	
		7. Peradresuoti naudotoją į prisijungimo langą.	
<b>Rezultatas</b>	Duomenų bazė atnaujinama nauju naudotoju.		
<b>Alternatyvus scenarijus – naudotojas klaidingai suvedė duomenis</b>			
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>	
		6a. Išsiųsti pranešimą apie klaidingai suvestus duomenis.	
		Kartoti žingsnius nuo anksčiau minėto 3 punkto, kol teisingai bus įvesti visi duomenys.	
<b>Rezultatas</b>	Duomenų bazė atnaujinama nauju naudotoju		

3.3 pav. atvaizduojama funkcijos *Registruotis* UML veiklos diagrama.



3.3 pav. Funkcijos Registruoti UML veiklos diagrama

- PA Peržiūrėti naudotojo projektų sąrašą.

Norint pasiekti kuriamą funkcionalumą, reikia įvykdyti *Peržiūrėti naudotojo projektų sąrašą* funkciją, kuri aprašoma 3.3 lentelėje. Įvykdžius funkciją naudotojas mato sąrašą projektų, kuriems jis gali sukurti specifikaciją.

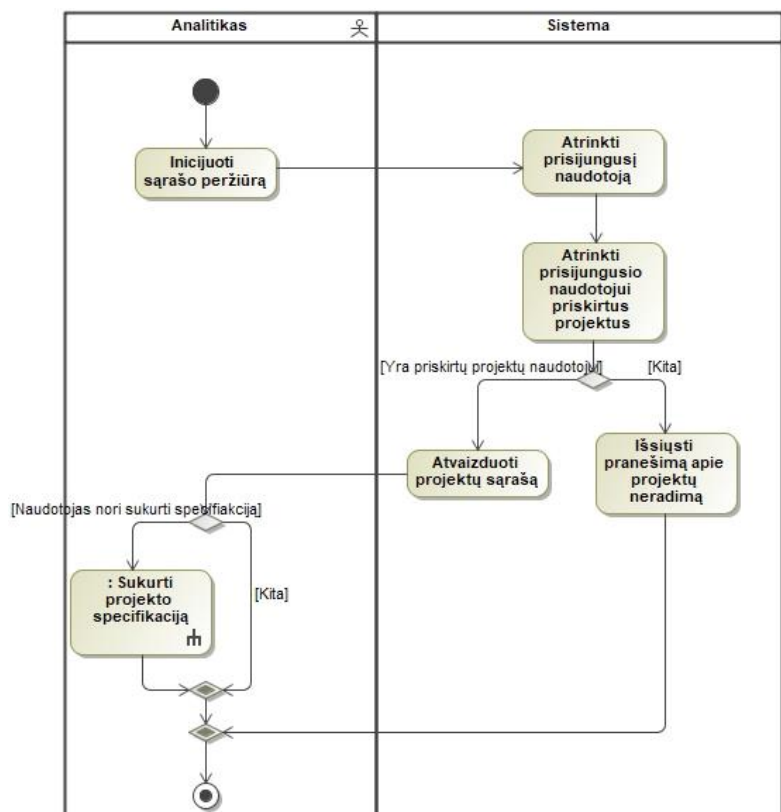
3.3 lentelė PA Peržiūrėti naudotojo projektų sąrašą *Volere* aprašymas

PA Peržiūrėti naudotojo projektų sąrašą		
<b>Tikslas/uždavinys.</b> Matyti prisijungusiam naudotojui priskirtų projektų sąrašą.		
<b>Aprašymas.</b> Naudotojas gali matyti visus jam priskirtus projektus, yra atmetami projektai, kurie yra nurodyti kaip nebeaktyvūs. Taip pat iš šio lango galima pasiekti projekto specifikacijos sukūrimo langą.		
<b>Sąlyga prieš</b>	Naudotojas yra prisijungęs prie sistemos.	
<b>Aktorius</b>	Analitikas.	
<b>Sužadinimo sąlyga</b>	Naudotojas nori matyti jam priskirtų projektų sąrašą. Naudotojas nori sukurti projektui specifikaciją.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	PA Sukurti projekto specifikaciją.
	<b>Apimami PA</b>	



<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – naudotojas nori matyti jam priskirtų projektų sąrašą</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
1. Inicijuoti projektų sąrašo atidarymą.	2. Atrinkti prisijungusį naudotoją.
	3. Atrinkti projektų sąrašą.
	4. Atidaryti projektų sąrašo langą.
5. Matyti projektų sąrašą.	
<b>Rezultatas</b>	Naudotojas mato visus jam priskirtus aktyvius projektus.
<b>Alternatyvus scenarijus – naudotojas nori sukurti specifikaciją</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
	6a. PA Sukurti projekto specifikaciją.
<b>Rezultatas</b>	Naudotojas yra nukreipiamas į specifikacijos sukūrimo langą.
<b>Alternatyvus scenarijus – nėra priskirtų projektų naudotojui</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>
	4b. Išsiųsti pranešimą apie projektų nebuvimą
<b>Rezultatas</b>	Naudotojas mato pranešimą.

3.4 pav. pavaizduota *Peržiūrėti naudotojo projektų sąrašą* funkcijos UML veiklos diagrama.



3.4 pav. Funkcijos *Peržiūrėti naudotojo projektų sąrašą* UML veiklos diagrama

- PA Peržiūrėti projektus ir PA Užpildyti VM žinių bazę

Šias abi funkcijas atlieka naudotojas, turintis rolę *Sistemos užsakovas*. Abi funkcijos savo veikimu primena kitas funkcijas, dėl to detalaus jų aptarimo nebus. PA *Peržiūrėti projektus*, veikimas atitinka PA *Peržiūrėti naudotojo projektų sąrašą*. O PA *Užpildyti VM žinių bazę* primena PA *Sukurti projekto specifikaciją*. Tik atrenkama informacija yra tik iš veiklos modelio informacijos.

- PA Peržiūrėti projektų sąrašą

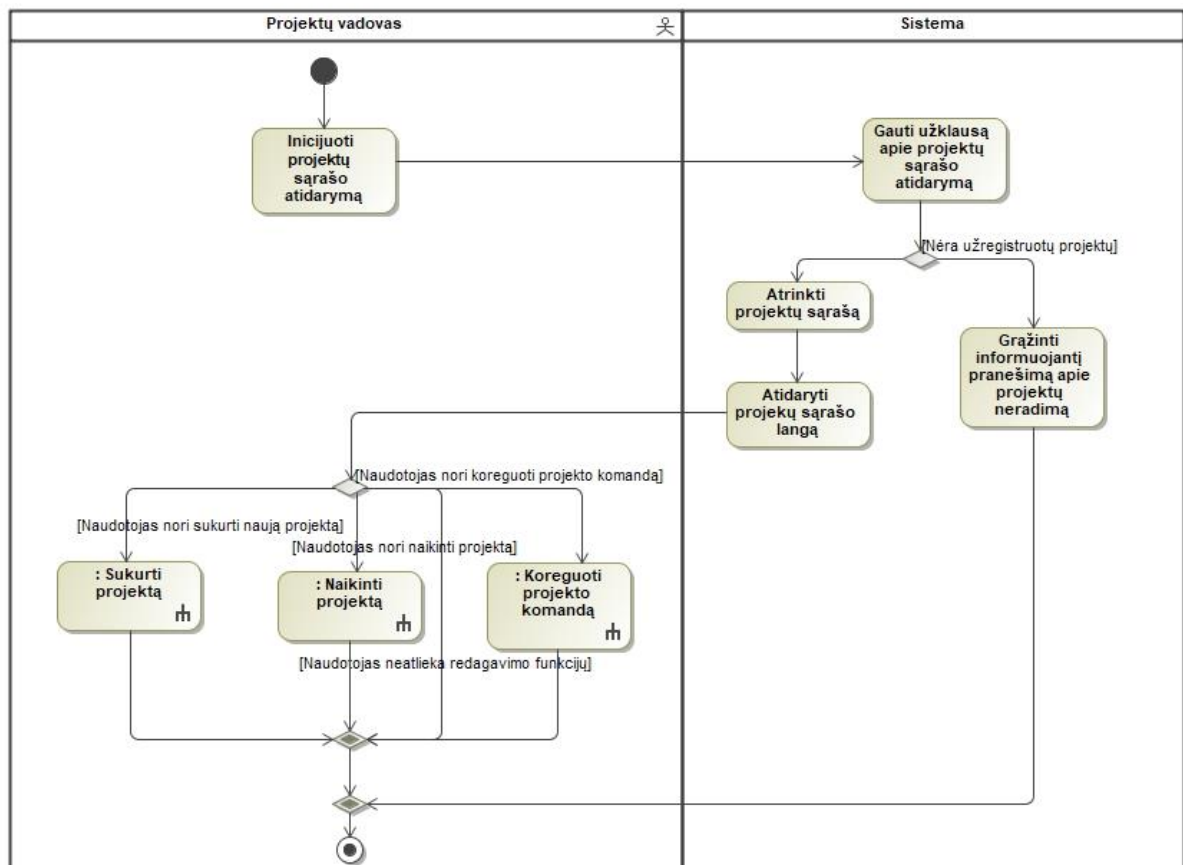
Administravimo funkcija – *Peržiūrėti projektų sąrašą* – aprašoma 3.4 lentelėje. Funkcijos tikslas — atvaizduoti visų aktyvių projektų sąrašą. Taip pat įvykdžius šią funkciją galima priskirti projektus naudotojams.

### 3.4 lentelė PA Peržiūrėti visus projektus *Volere* aprašymas.

PA Peržiūrėti visus projektus		
<b>Tikslas/uždavinys.</b> Matyti visų užregistruotų aktyvių projektų sąrašą		
<b>Aprašymas.</b> Naudotojas gali matyti visus užregistruotus projektus. Funkcija atmeta projektus, kurie yra nurodyti kaip nebeaktyvūs. Taip pat iš šio lango galima pasiekti naujo projekto sukūrimą ir sukurtų projektų „panaikinimą“.		
<b>Sąlyga prieš</b>	Naudotojas yra prisijungęs prie sistemos.	
<b>Aktorius</b>	Administratorius.	
<b>Sužadavimo sąlyga</b>	Naudotojas nori matyti projektų sąrašą. Naudotojas nori sukurti naują projektą. Naudotojas nori panaikinti projektą.	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	PA Naikinti projektą. PA Sukurti projektą. PA Koreguoti Projekto Komandą.
	<b>Apimami PA</b>	
	<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – yra aktyvių projektų</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
1. Inicijuoti projektų sąrašo atidarymą.		2. Atrinkti projektų sąrašą.
		3. Atidaryti projektų sąrašo langą.
4. Matyti projektų sąrašą.		
<b>Po-sąlyga</b>	Naudotojas mato visus aktyvius projektus.	
<b>Alternatyvus scenarijus – nėra aktyvių projektų</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
		3a. Išsiųsti pranešimą apie projektų nebuvimą.
<b>Rezultatas</b>	Naudotojas mato pranešimą.	
<b>Alternatyvus scenarijus – naudotojas nori redaguoti projektų sąrašą</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>

Naudotojas nori sukurti naują projektą	PA <i>Sukurti Projektą</i>
Naudotojas nori panaikinti projektą	PA <i>Panaikinti Projektą</i>
Naudotojas nori koreguoti projekto komandą	PA <i>Koreguoti Projekto Komandą</i>
<b>Rezultatas</b>	Duomenų bazės informacija pakoreguota.

3.5 pav. pavaizduota funkcijos *Peržiūrėti visus projektus* UML veiklos diagrama.



3.5 pav. Funkcijos *Peržiūrėti visus projektus* UML veiklos diagrama

- PA *Sukurti naują projektą*

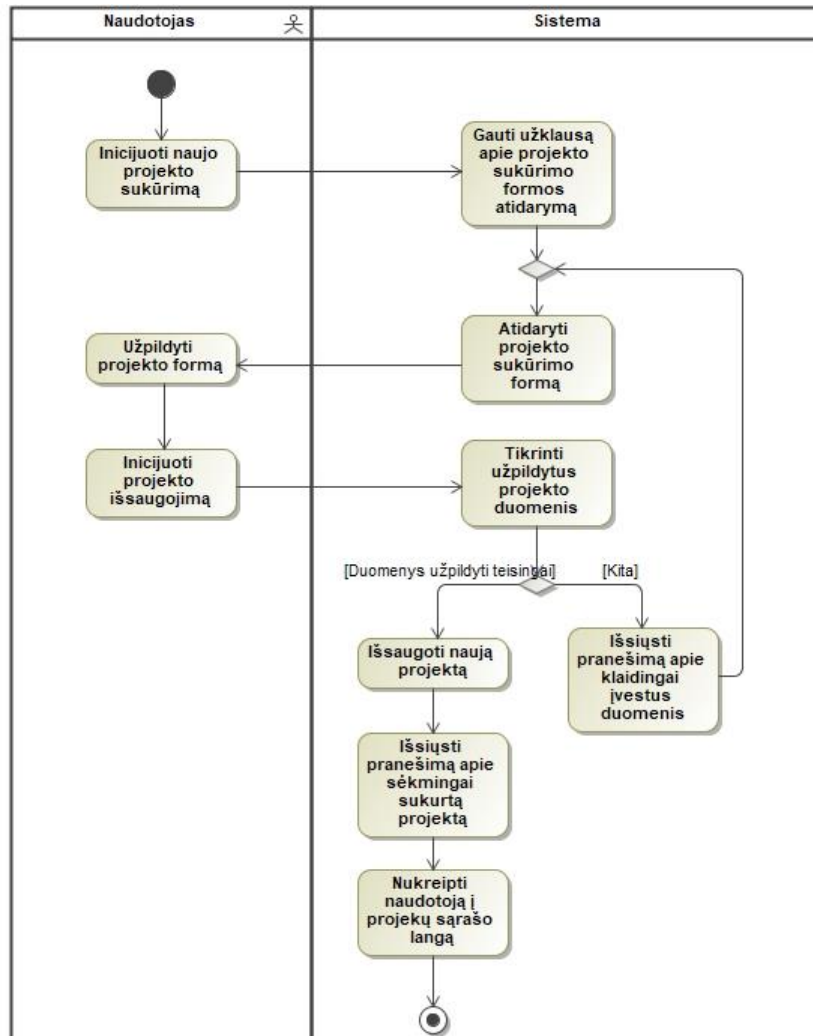
Funkcija – *Sukurti naują projektą* – aprašoma 3.5 lentelėje Sukuriant naują projektą reikia nurodyti jo pavadinimą, priežastis ir veiklos aprašymus.

3.5 lentelė PA *Sukurti projektą Volere* aprašymas

<b>PA Sukurti projektą</b>	
<b>Tikslas/uždavins.</b> Užregistruoti naują projektą sistemoje	
<b>Aprašymas.</b> Naudotojas gali užregistruoti naują projektą. Projekto informacija yra išsaugojama, kai visa privaloma informacija yra suvesta teisingai.	
<b>Sąlyga prieš</b>	Naudotojas yra atsidaręs projektų sąrašo langą
<b>Aktorius</b>	Administratorius
<b>Sužadinimo sąlyga</b>	Naudotojas nori sukurti naują projektą;
<b>Susiję panaudojimo</b>	<b>Išplečiantys PA</b>

atvejai	Apimami PA	
	Apibendrinti PA	
<b>Pagrindinis scenarijus – naudotojas teisingai įvedė duomenis</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
1. Inicijuoti naujo projekto sukūrimą		2. Atidaryti projekto sukūrimo formą
3. Užpildyti projekto formą		
4. Inicijuoti projekto išsaugojimą		5. Tikrinti užpildytus projekto duomenis
		6. Išsaugoti naują projektą
		7. Išsiųsti pranešimą apie sėkmingai įvykdytą išsaugojimą
		8. Nukreipti naudotoją į Projektų sąrašo langą
<b>Rezultatas</b>	Naujo projekto informacija išsaugojama, naudotojas mato atnaujintą projektų sąrašą	
<b>Alternatyvus scenarijus – yra neteisingai įvestų duomenų</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
		6a. Išsiųsti pranešimą apie klaidingai suvestus duomenis
		Kartoti žingsnius nuo anksčiau minėto 3 punkto, kol teisingai bus įvesti visi duomenys.
<b>Rezultatas</b>	Naujo projekto informacija išsaugoma ir naudotojas mato atnaujintą projektų sąrašą.	

3.6 pav. atvaizduojama funkcijos *Sukurti naują projektą* UML veiklos diagrama.



3.6 pav. Funkcijos Sukurti projektą UML veiklos diagrama

- PA Naikinti projektą

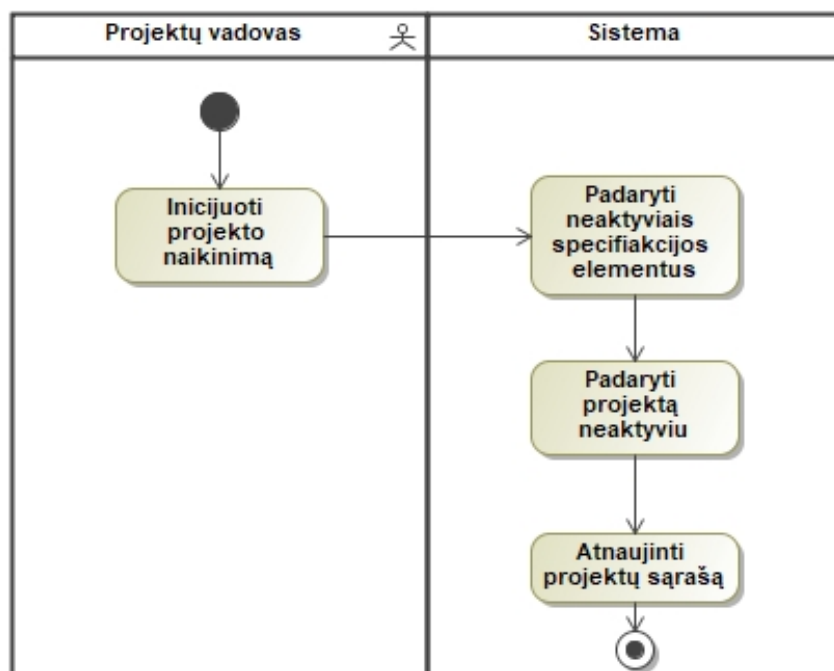
*Naikinti projektą* funkcija aprašoma 3.6 lentelėje. Funkcija projekto neištrina iš sistemos duomenų bazės, o uždeda žymą, kuri nurodo, kad projektas yra neaktyvus. Tai padarius projektas nefigūruoja bendruose projektų sąrašuose ir jam negalima sukurti specifikacijos.

3.6 lentelė PA Naikinti projektą *Volere* aprašymas

<b>PA Naikinti projektą</b>	
<b>Tikslas/uždavinys.</b> Padaryti projektą neaktyviu	
<b>Aprašymas.</b> Naudotojas gali naikinti projektą – pakeisti jo statusą į <i>Neaktyvus</i> . Tai padarius visi projekto specifikacijos elementai, automatiškai paverčiami neaktyviais. Taip pat, šis projektas nefigūruoja bendruose projektų sąrašuose.	
<b>Sąlyga prieš</b>	Naudotojas yra atsidaręs projektų sąrašo langą
<b>Aktorius</b>	Administratorius
<b>Sužadinimo sąlyga</b>	Naudotojas nori panaikinti projektą
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>
	<b>Apimami PA</b>

	<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus</b>		
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>	
1. Inicijuoti projekto sunaikinimą	2. Padaryti projekto specifikacijos elementus neaktyvius	
	3. Padaryti projektą neaktyviu	
	4. Atnaujinti projektų sąrašą	
<b>Rezultatas</b>	Projekto informacija ir specifikacijos elementai atnaujinami	

3.7 pav. atvaizduojama funkcijos *Naikinti projektą* UML veiklos diagrama.



3.7 pav. Funkcijos *Naikinti projektą* UML veiklos diagrama

– PA Koreguoti projekto komandą

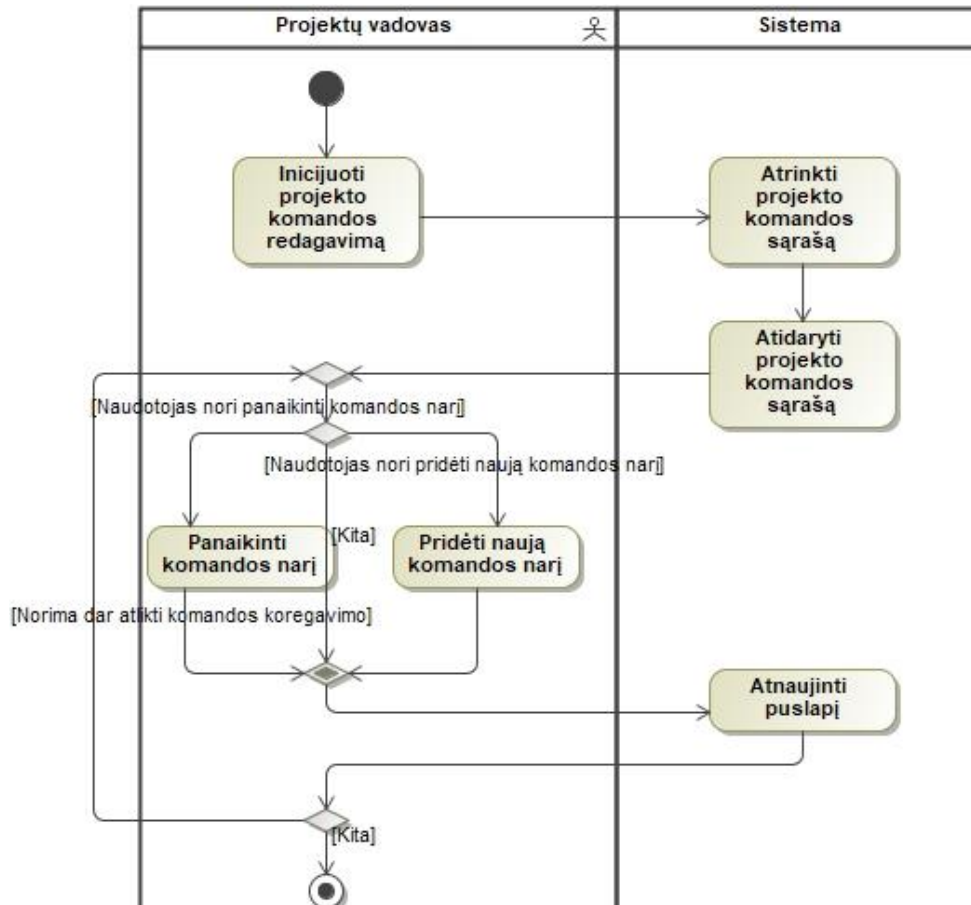
Vykdam funkciją *Koreguoti projekto komandą* galima arba pridėti naują komandos narį arba panaikinti esamą komandos narį. Funkcijos veikimas aprašomas – 3.7 lentelėje.

3.7 lentelė PA *Koreguoti projekto komandą* *Volere* aprašymas

<b>PA Koreguoti projekto komandą</b>	
<b>Tikslas/uždavinys.</b> Koreguoti projekto komandą	
<b>Aprašymas.</b> Naudotojas gali redaguoti komandos sudėtį – panaikinti arba pridėti naują komandos narį. Vienu kartu galima atlikti kelis koregavimus. Taip pat galima tik pasižiūrėti komandą, neatliekant jokių koregavimų.	
<b>Sąlyga prieš</b>	Naudotojas yra atsidaręs projektų sąrašo langą
<b>Aktorius</b>	Projektų vadovas

<b>Sužadavimo sąlyga</b>	Naudotojas nori pridėti naują komandos narį; Naudotojas nori peržiūrėti projekto komandą; Naudotojas nori panaikinti komandos narį;	
<b>Susiję panaudojimo atvejai</b>	<b>Išplečiantys PA</b>	
	<b>Apimami PA</b>	
	<b>Apibendrinti PA</b>	
<b>Pagrindinis scenarijus – pridėti naują komandos narį</b>		
<b>Naudotojo veiksmai</b>		<b>Sistemos veiksmai</b>
1. Inicijuoti projekto komandos sąrašo peržiūrą		2. Atrinkti projekto komandos sąrašą
		3. Atidaryti projekto komandos langą
4. Pridėti naują komandos narį		5. Atnaujinti puslapį
<b>Rezultatas</b>	Pridėtas naujas narys – išsaugotas įrašas duomenų bazėje	
<b>Alternatyvus scenarijus – panaikinti komandos narį</b>		
4a. Panaikinti komandos narį		
<b>Rezultatas</b>	Panaikintas komandos narys – panaikintas duomenų bazės įrašas	
<b>Alternatyvus scenarijus – atlikti keletą koregavimų</b>		
		6b. Kartoti žingsnius 4(4a) – 5, iki kol visi koregavimai bus atlikti

3.8 pav. atvaizduojama funkcijos *Koreguoti projekto komandą* UML veiklos diagrama.



3.8 pav. Funkcijos Koreguoti projekto komanda UML veiklos diagrama

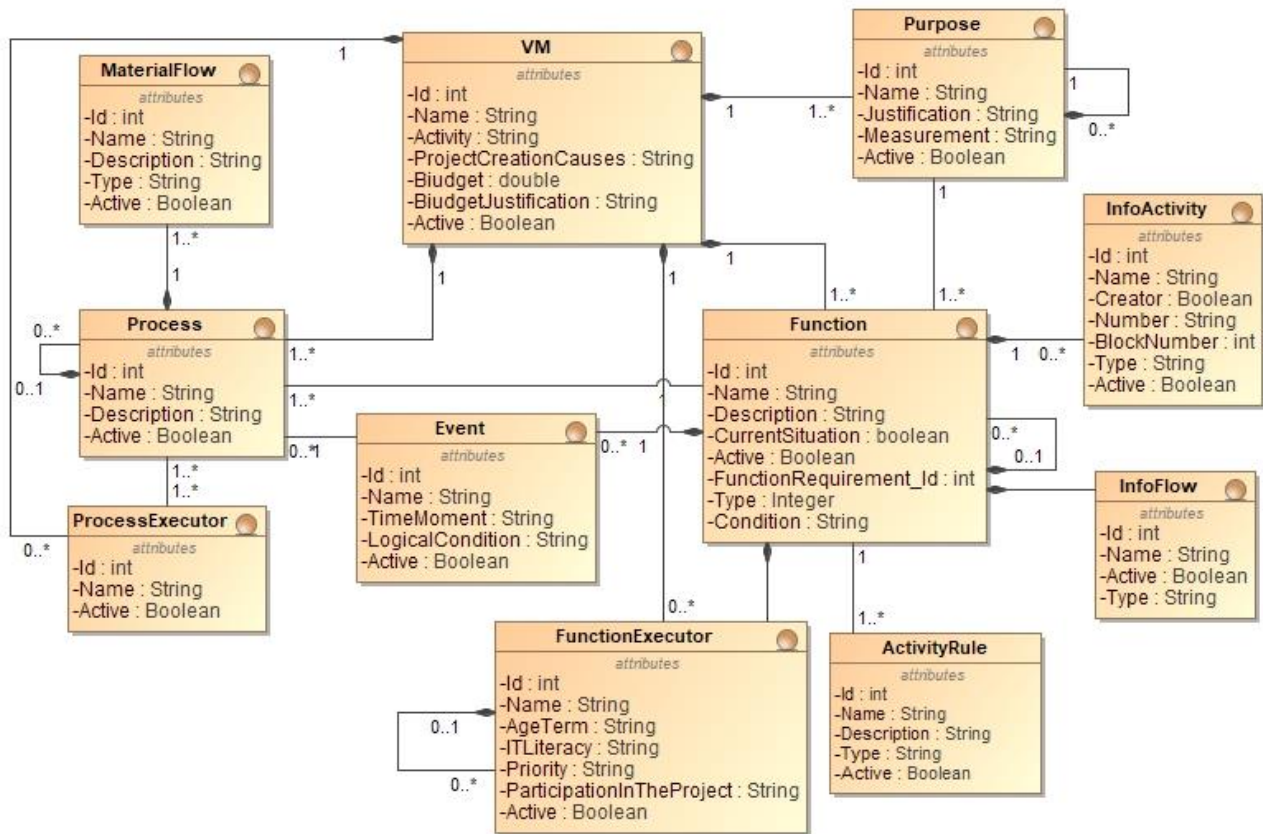
### 3.2. Dalykinės srities modelis

Dalykinės srities modelis susideda iš trijų dalių:

1. veiklos modelio klasių;
2. *Volere* šablono klasių;
3. naudotojo informacijos.

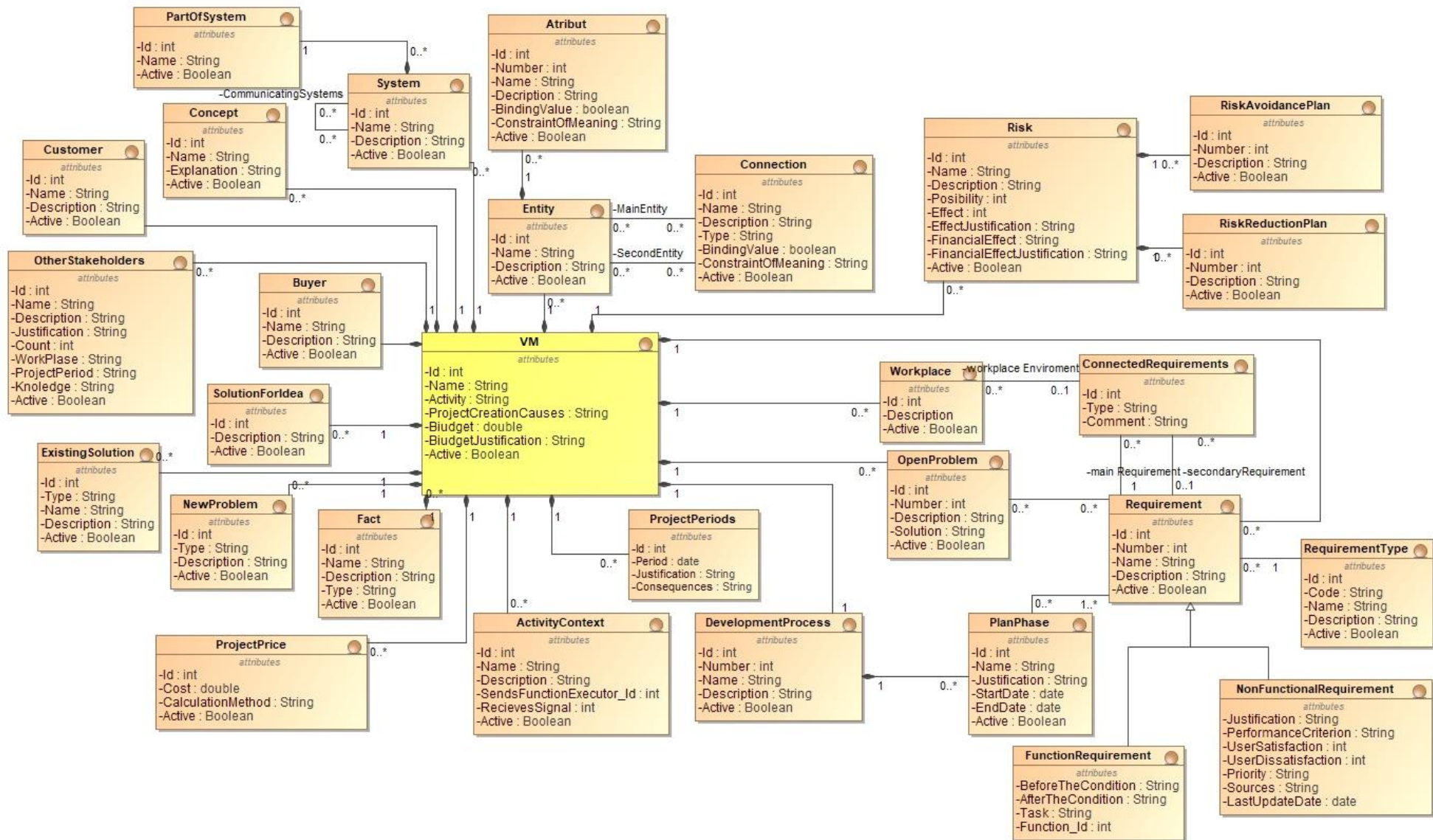
Veiklos modelio ir *Volere* šablono klasės paimamos iš 2.6 pav. ir sumodeliuotos UML klasių diagramomis. Jos yra suskaidytos į dvi diagramas – 3.9 pav. ir 3.10 pav. Tokiu atveju, jei kistų kuris nors iš dviejų modelių, būtų lengviau atlikti duomenų bazės koregavimą.





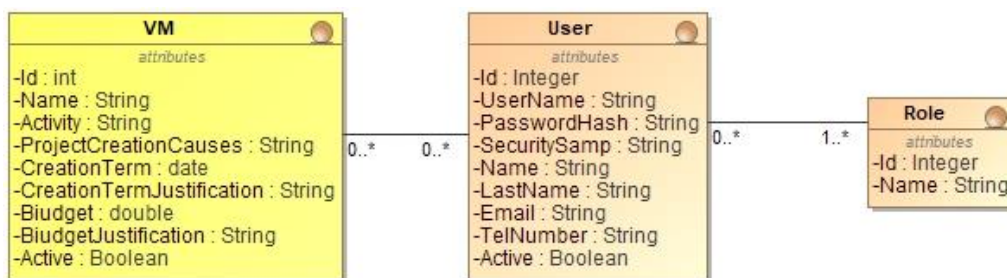
3.9 pav. VM klasių diagrama

Volere šablono duomenų bazės UML klasių diagramoje yra viena klasė – VM – pavaizduota geltona spalva, pabrėžiant ryšį tarp veiklos modelio duomenų bazės ir Volere šablono.



3.10 pav. Volere šablono UML klasių diagrama

Kaip ir *Volere* šablono klasių diagrama, taip ir naudotojo duomenų UML klasių diagrama (3.11 pav.) turi papildomą klasę – VM – dėl aiškumo. Ši duomenų bazė yra pati mažiausia, bet pakankama norint įvykdyti kuriamą metodą.



3.11 pav. Naudotojo duomenų UML klasių diagrama

### 3.3. Loginė architektūra.

Eksperimentinės sistemos architektūra, sumodeliuota naudojant UML paketų diagramą (angl. *UML package diagram*) (3.12 pav.), susideda iš trijų lygių:

1. naudotojo sąsajos – kurioje naudojami *HTML* tipo failai;
2. valdikliai – metodai, kurie ištraukia ir koreguoja duomenis esančius duomenų bazėse.
3. duomenų modelis – saugojamos visos trijų duomenų bazės klasės.



3.12 pav. Sistemos loginė architektūra

### 3.4. Panaudojimo atvejų sekų diagramos.

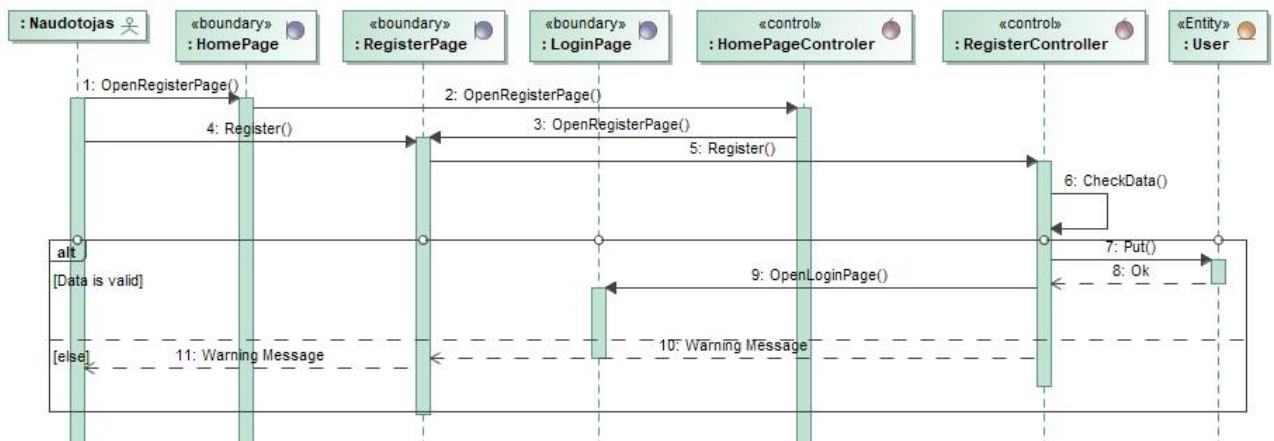
Norint suprasti kaip komunikuoja ekspertinės sistemos dalys kiekvienam panaudojimo atvejui buvo sukurtos UML sekų diagramos. Vienintelio panaudojimo atvejo PA *Sukurti projekto specifikaciją* diagrama nebuvo sumodeliuota, nes šios funkcijos veikimas aprašytas 2.3 skyriuje.

Diagramos suskirstytos pagal posistemas, 3.13 pav. – 3.16 pav.

- naudotojo posistemės sekų diagramos

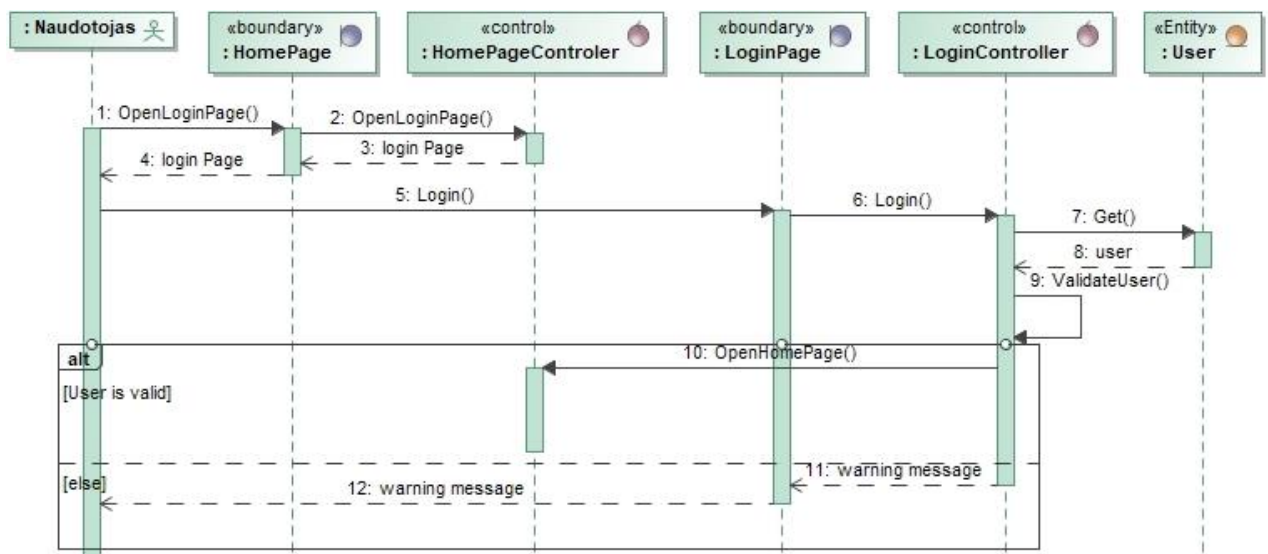
Naudotojo posistemė susideda iš trijų funkcijų: naudotojo registracijos, prisijungimo prie sistemos ir atsijungimo. Funkcijos *Atsijungti* sekų diagramos nėra, nes visi šios funkcijos žingsniai susideda iš sesijos nutraukimo. Kitų funkcijų sekų diagramos atvaizduotos 3.13 pav. – 3.14 pav.

Registracijos forma susideda iš registracijos lango atidarymo, duomenų užpildymo ir jų patikrinimo. Jei informacija yra teisingai suvesta yra atidaromas prisijungimo langas (4.13 pav.).



3.13 pav. Funkcijos Registruotis UML sekų diagrama

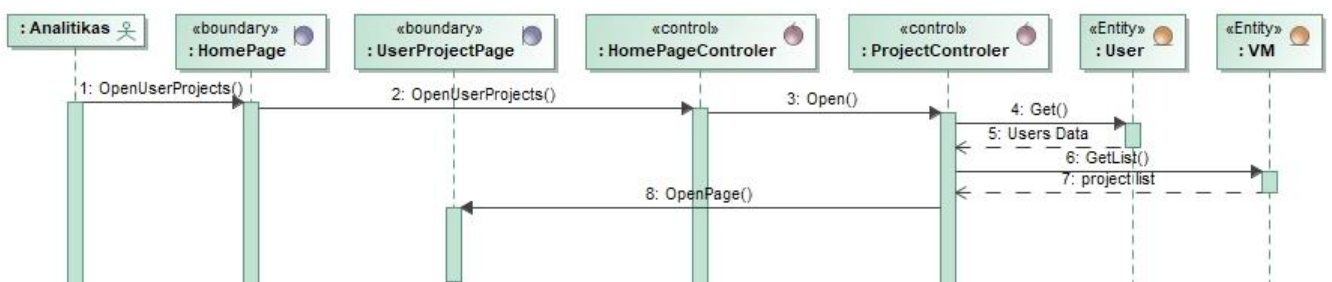
Prisijungimo funkcija susideda iš prisijungimo lango atsidarymo, duomenų užpildymo ir jų patikrinimo (3.14 pav.).



3.14 pav. Funkcijos Prisijungti UML sekų diagrama

- projektų specifikacijos posistemė

Funkcija *Peržiūrėti savo projektus* ir *Peržiūrėti projektus* (3.15 pav.) susideda iš informacijos apie prisijungusį naudotoją atrinkimo ir projektų, su kuriais naudotojas turi sąsają atrinkimo. Jei projektų sąrašas yra tuščias, grąžinamas informacinis pranešimas.

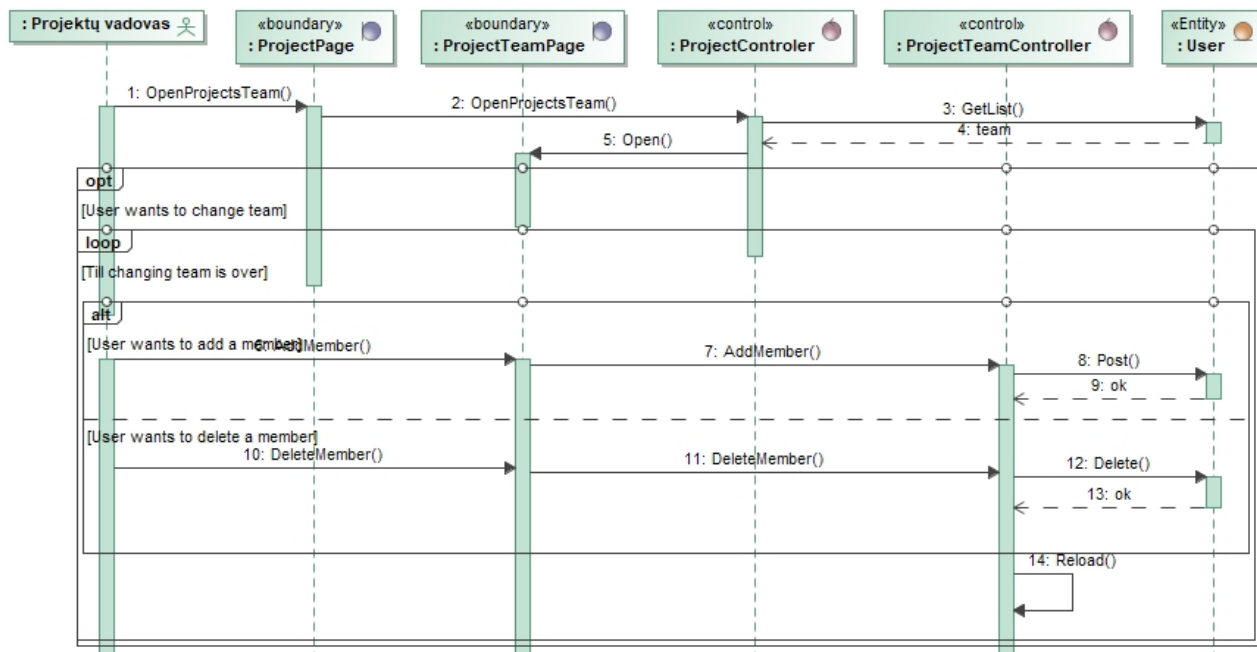


3.15 pav. Funkcijos Peržiūrėti savo projektus UML sekų diagrama

– projektų posistemė

Projektų posistemę sudaro trys funkcijos, kurias atlieka *Projektų vadovo* rolę turintys naudotojai: *Peržiūrėti projektų sąrašą*, *Sukurti projektą* ir *Naikinti projektą*. Diagramų veikimas atitinka kitas funkcijas, tik naudojami kiti valdikliai su kitomis duomenų bazės klasėmis. Dėl to jos nebus atvaizduojamos UML sekų diagramomis.

Funkcijos *Koreguoti projekto komandą* (3.16 pav.) susideda iš projekto komandos atrinkimo ir atidarymo naujame lange. Tai įvykdžius galima koreguoti daugiau nei vieną kartą – pridėti arba panaikinti komandos narį.



3.16 pav. Funkcijos *Koreguoti projekto komandą* UML sekų diagrama

## 4. Sprendimo realizacija ir testavimas

Šiame skyriuje bus aprašoma kokias technologijas naudojant buvo sukurta eksperimentinė sistema, naudojanti reikalavimų specifikavimo metodą. Taip pat aprašomas šios sistemos testavimas, kuris susideda iš dviejų dalių – serverio dalies kodo ir naudotojo sąsajos testavimo.

### 4.1. Sprendimo realizacija

Sprendimo realizaciją lengviausia suvokti pamačius sistemos fizinių išdėstymą. Tam tinkamiausia UML diegimo diagrama (4.1 pav.) (angl. *UML deployment diagram*). Diagramoje pavaizduota sistema susideda iš dviejų projektų – serverio dalies (*API*), kuri buvo suprogramuota *C#* programavimo kalba, naudojantis *.NET* karkasu, ir naudotojo sąsajos (*WEB API*), kuri buvo suprogramuota naudojantis *AngularJS* baziniu *JavaScript* kūrimo karkasu. Visas sistemos kodas buvo parašytas naudojantis *Visual Studio 2019 Enterprise* integruota programavimo aplinka.

Serverio dalis susideda iš trijų svarbių komponentų:

1. modeliai (*models*) – aprašomi objektai, kuriems priskiriamos reikšmės iš duomenų bazės;
2. valdikliai (*controllers*) – aprašomi metodai, kuriais yra ištraukiama informacija iš duomenų bazės bei koreguojami duomenys;
3. kontekstas (*context*) – aprašomi duomenų bazių modeliai.

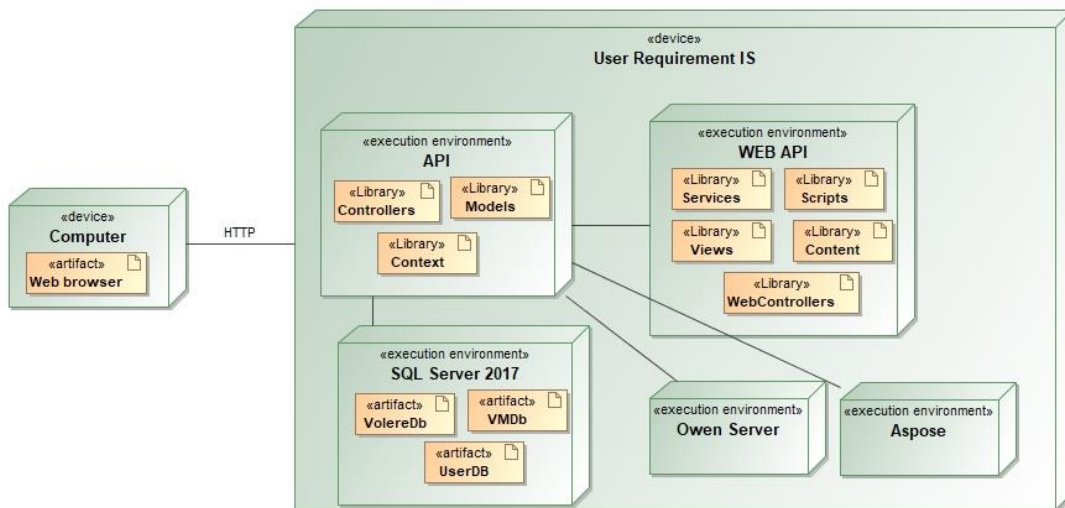
Naudotojo sąsaja susideda iš penkių svarbių komponentų:

1. paslaugų (*services*) – aprašomi metodai, kurie kreipiasi į serverio dalies valdiklius;
2. valdikliai (*web controllers*) – aprašomi metodai, kuriuose nurodomi naudotojo sąsajos mygtukų paspaudimų veiksmai;
3. naudotojo sąsajos puslapiai (*views*);
4. naudotojo sąsajos išvaizdos turinys (*content*);
5. naudotojo sąsajos papildomos funkcijos (*scripts*).

Taip pat matome, kad sistemoje yra naudojamas *SQL Server*, kuriame yra saugojamos trys duomenų bazės: naudotojo duomenų (*UserDb*), veiklos modelio duomenų (*VMDb*) ir *Volere* šablono (*VolereDb*).

Naudotojo duomenų autentifikavimui buvo naudojamas *Owen Server*, trečiųjų šalių paslauga, kuri naudoja *OAuth2* protokolą. Šis protokolas pasirinktas, nes jį naudojant slaptažodis yra apsaugojamas prisijungimo metu persiunčiant autentifikavimo žetoną, o ne prisijungimo duomenis.

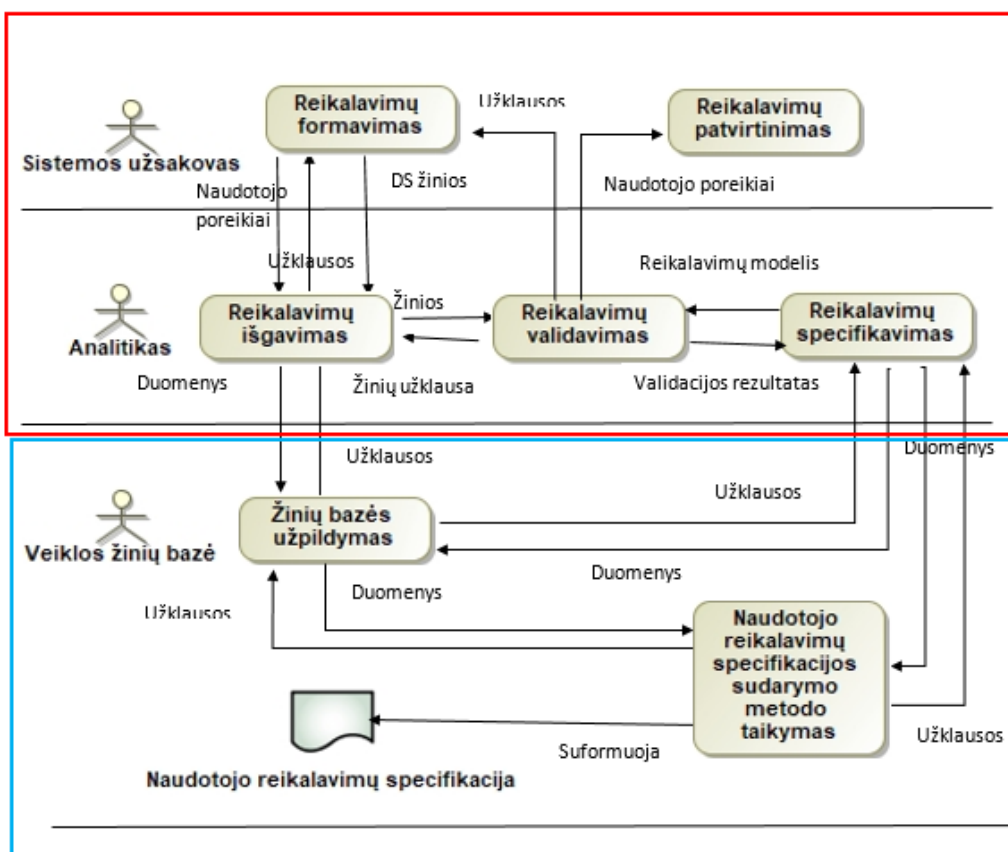
Diagramų generavimui buvo naudotas *Aspose* trečiųjų šalių programa, kurios pagalba galima suprojektuoti *MS Visio* failus.



4.1 pav. Eksperimentinės sistemos UML diegimo diagrama

#### 4.2. Sistemos veikimo aprašas

Svarbiausias sistemos funkcionalumas yra atrinkti užregistruoto projekto reikalavimų specifikaciją. Šis funkcionalumas pavaizduotas laisvos formos diagrama – 4.2 pav., kurios klasikinio reikalavimų surinkimo ir specifikavimo dalis sudaryta pagal [19] šaltinį.



4.2 pav. Ekspertinės sistemos funkcionalumas

Funkcionalumas susideda iš dviejų dalių:

1. sistemos poreikių formulavimas ir analizavimas – naudojami tradiciniais IS inžinerijos naudotojo poreikių surinkimo, analizės ir specifikavimo etapo pagrindiniai procesai, [19]. Šie procesai vyksta tarp sistemos užsakovo ir analitiko (pažymėta raudona spalva);
2. veiklos modelio užpildymas – dalykinės srities žiniomis užsakovas užpildo veiklos modelio žinių bazę. Ir projekto reikalavimų specifikacijos sudarymo algoritmo naudojimas – algoritmas naudojamas jau užpildytą veiklos modelio žinių bazę ir papildomą informaciją, kurią įveda analitikas, galima suformuoti reikalavimų specifikaciją ir sugeneruoti diagramas (pažymėta mėlyna spalva).

### 4.3. Testavimo modelis, duomenys, rezultatai

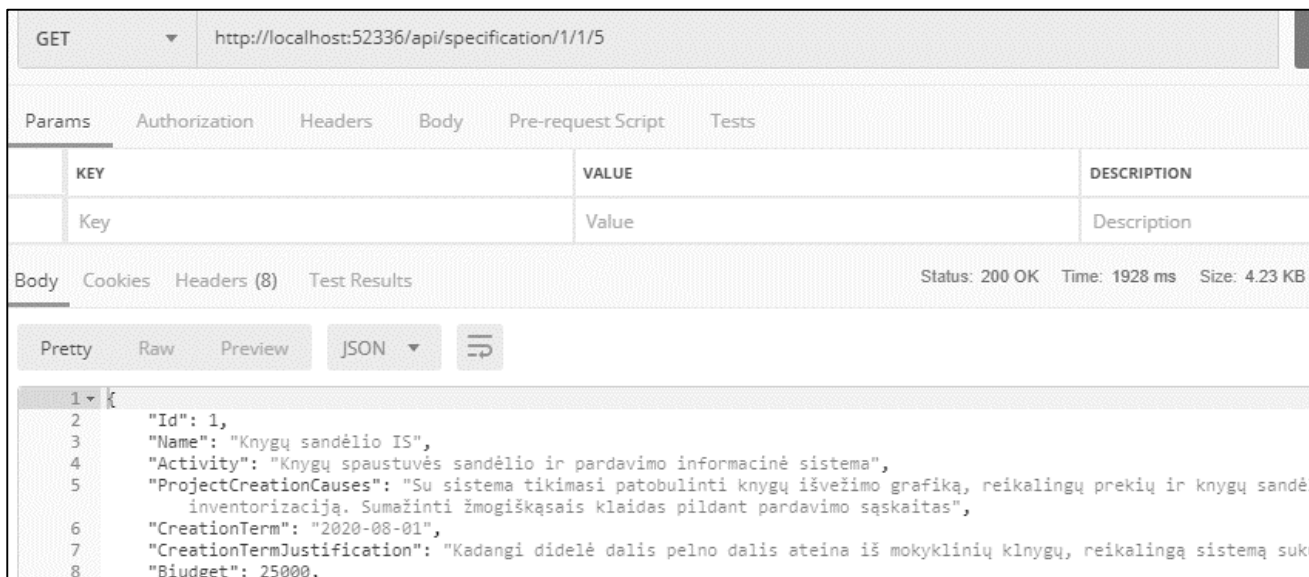
Kuriamas funkcionalumas buvo testuojamas dviem aspektais:

1. gaunami teisingi rezultatai, kuomet yra prašoma gražinti duomenis iš duomenų bazės ir koreguojant elementų informaciją.
2. atvaizduojama informacija atitinka *Volere* šablono turinį.

Pirmasis testavimo aspektas atliktas naudojantis *Postman* programa, kuri yra skirta testuoti *HTTP* užklausas, paduodamas kuriamai IS. Žemiau yra pateiktos kelios užklauskos (4.3 pav. – 4.11 pav.), kuriomis buvo testuojamas funkcionalumo veikimas.

Užklausų pavyzdžiai:

- projekto informacijos gražinimas, kai projektas egzistuoja. Nurodoma *GET HTTP* užklausa su papildomais parametrais – *Projekto id*, *Specifikacijos tipo id*, Aktoriaus pagal, kuri atrenkama informacija, *id*:



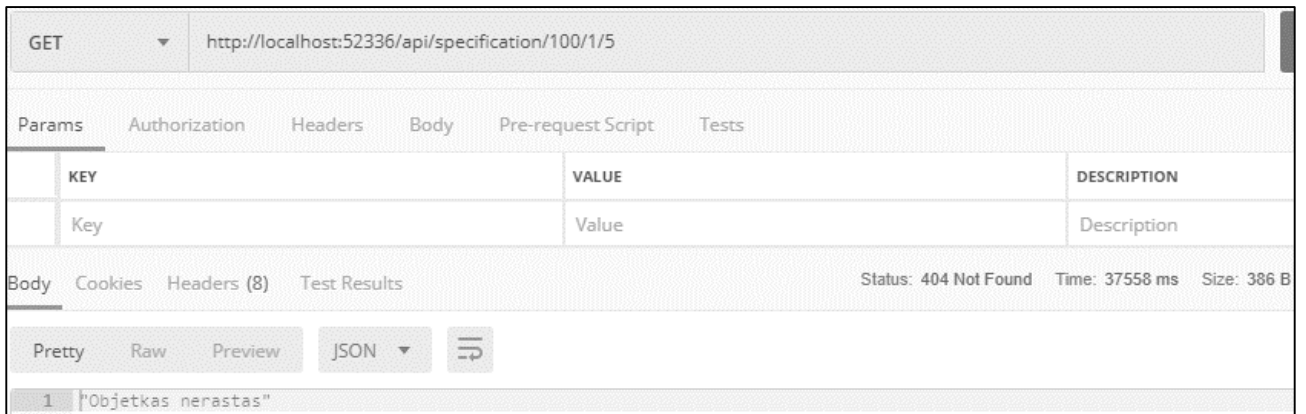
The screenshot shows a Postman interface with a GET request to `http://localhost:52336/api/specification/1/1/5`. The response is a JSON object with the following structure:

```
1 {
2   "Id": 1,
3   "Name": "Knygų sandėlio IS",
4   "Activity": "Knygų spaustuvės sandėlio ir pardavimo informacinė sistema",
5   "ProjectCreationCauses": "Su sistema tikimasi patobulinti knygų išvežimo grafiką, reikalingų prekių ir knygų sandėlio inventurizaciją. Sumažinti žmogiškaisiais klaidas pildant pardavimo sąskaitas",
6   "CreationTerm": "2020-08-01",
7   "CreationTermJustification": "Kadangi didelė dalis pelno dalis ateina iš mokyklinių knygų, reikalingą sistemą sukurti",
8   "Budget": 25000,
```

4.3 pav. Funkcionalumo testavimas. Projekto informacijos gražinimas

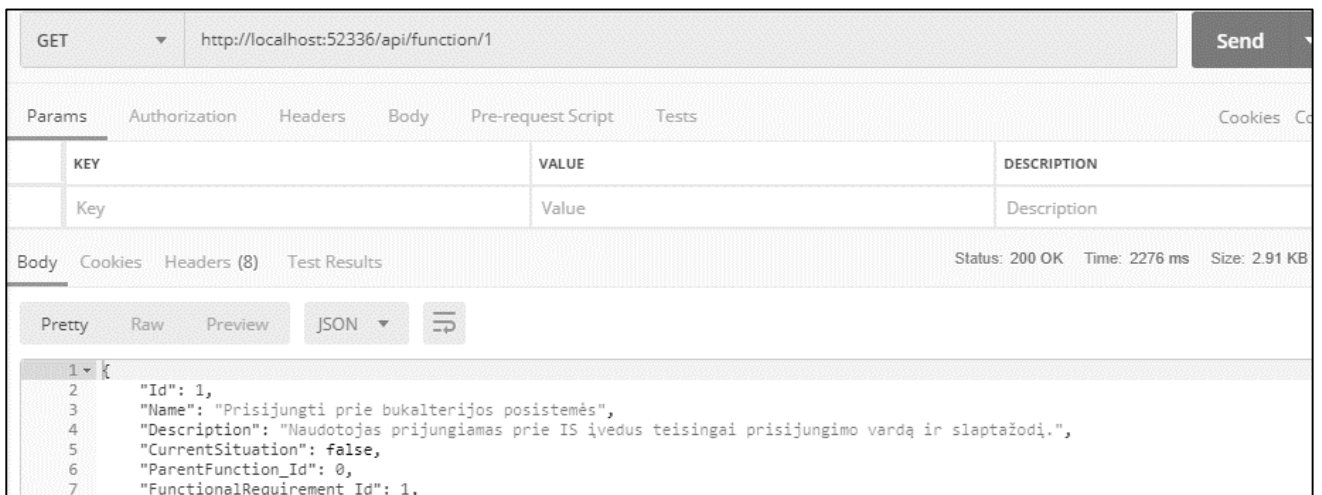


- projekto informacijos gražinimas, kai projektas neegzistuoja:



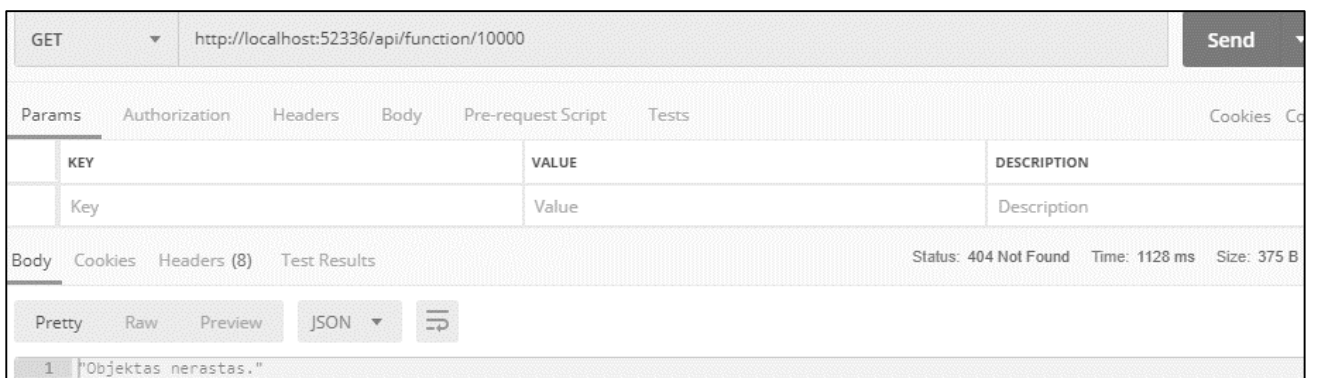
#### 4.4 pav. Funkcionalumo testavimas. Projekto informacijos gražinimas, projektas neegzistuoja

- projekto funkcijos papildomos informacijos gražinimas, kai funkcija egzistuoja. Nurodoma *GET HTTP* užklausa su papildomus parametru – *Funkcijos id*:



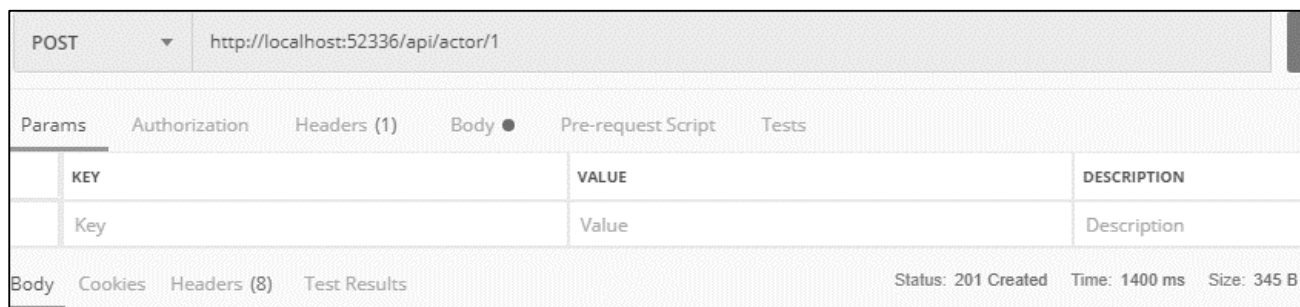
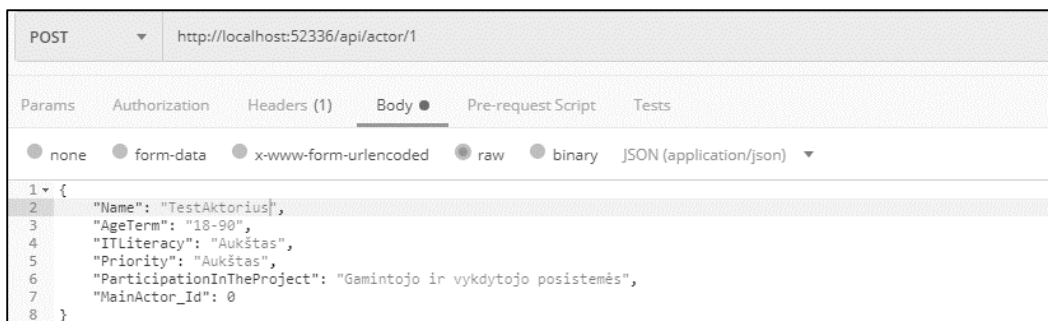
#### 4.5 pav. Funkcionalumo testavimas. Projekto funkcijos informacijos gražinimas

- projekto funkcijos papildomos informacijos gražinimas, kai funkcija neegzistuoja:



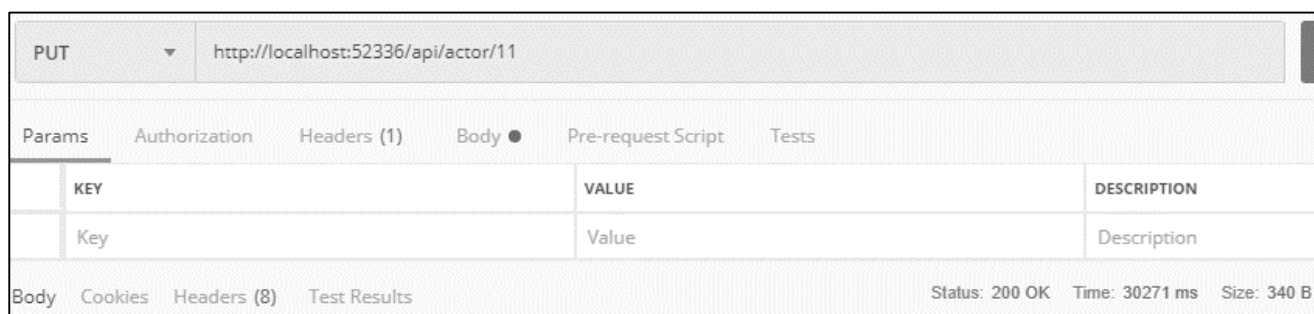
#### 4.6 pav. Funkcionalumo testavimas. Projekto funkcijos informacijos gražinimas, funkcija neegzistuoja

- naujo aktoriaus sukūrimas projekte. Nurodoma *POST HTTP* užklausa su papildomu parametru – *Aktoriaus id* ir *Body* dalyje nurodant naujo aktoriaus duomenis:



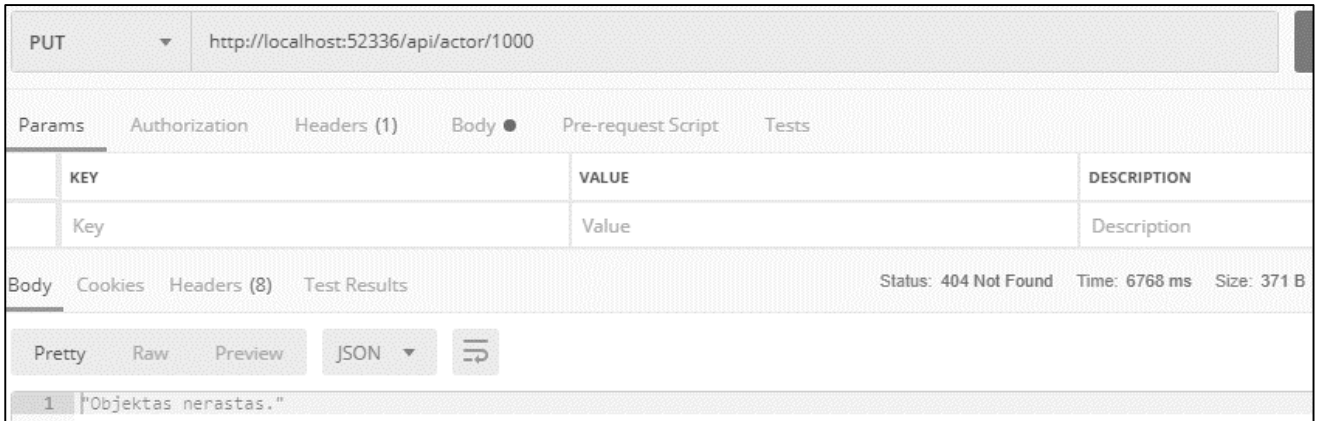
#### 4.7 pav. Funkcionalumo testavimas. Naujo aktoiaus sukūrimas

- aktoiaus informacijos koregavimas. Nurodoma *PUT HTTP* užklausa su papildomu parametru — *Aktoiaus id* ir *Body* dalyje nurodant aktoiaus naujus duomenis:



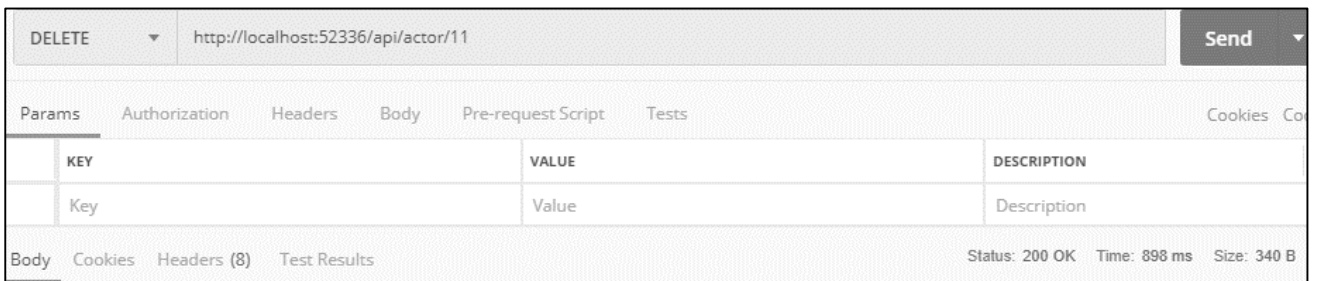
#### 4.8 pav. Funkcionalumo testavimas. Aktoiaus informacijos koregavimas

- aktoriaus informacijos koregavimas, kai aktorius neegzistuoja:



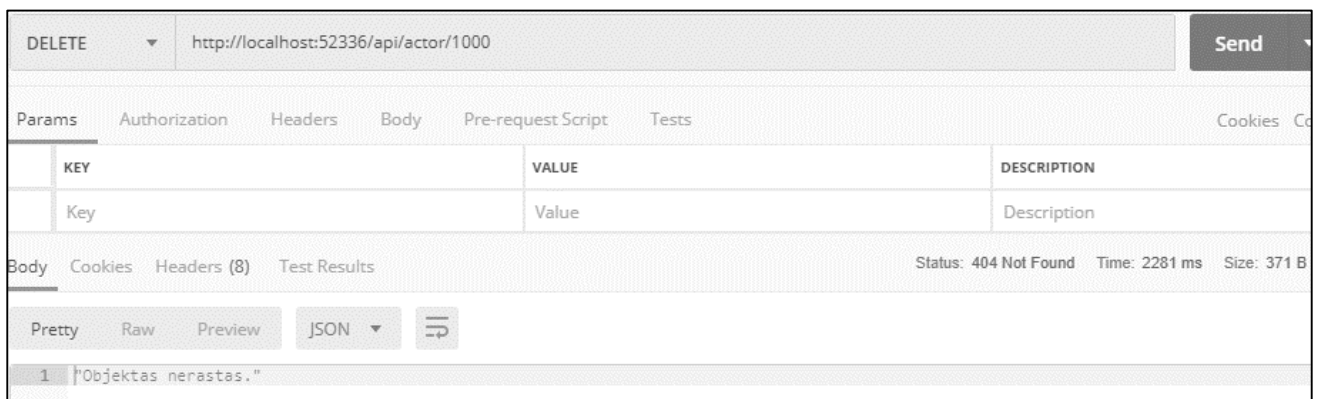
**4.9 pav. Funkcionalumo testavimas. Aktoriaus informacijos koregavimas, aktorius neegzistuoja**

- aktoriaus informacijos ištrynimasis. Nurodoma *DELETE HTTP* užklausa su papildomu parametru – *Aktoriaus id*:



**4.10 pav. Funkcionalumo testavimas. Aktoriaus ištrynimasis**

- aktoriaus informacijos ištrynimasis, kai aktorius neegzistuoja:



**4.11 pav. Funkcionalumo testavimas. Aktoriaus ištrynimasis, aktorius neegzistuoja**

Antras testavimo aspektas buvo atliktas, lyginant naudotojo sąsajoje atvaizduojamą informaciją su *Volere* šablono aprašymu (1.11 lentelė). Buvo lyginamos formos naudojamos *Specifikacija pagal pasirinktą aktorių* aspekto, nes šis aspektas naudoja vienas svarbiausių specifikacijos formas:

- projekto bendra informacija – pagal *Volere* šabloną ekspertinėje sistemoje projekto bendra informacija yra apjungiamas šiais skyriais: 1a. Analizuojama veikla ir priežastys – aprašoma laisva forma. Pasižiūrėjus į atvaizduojamą informaciją sistemoje (4.12 pav.) yra matoma visa anksčiau paminėta informacija.

**Pavadinimas**

Ligoninės IS

**Projekto aprašymas**





Ligoninės informacinės sistema, kurioje yra ambulatorijos, stacionaro, diagnostikos, operacijų, SPS, atask:

**Kūrimo priežastys**

Su sistema tikimasi pagreitinti žmonių darbą, išvengti žmogiškųjų klaidų

#### 4.12 pav. Projekto pagrindinės informacijos atvaizdavimas sistemoje

- funkcijų naudotojai – funkcijų naudotojai turi būti aprašomi lentelės forma, kurioje turi būti paminėta naudotojo rolė, pagrindinės atliekamos funkcijos, pagrindimas, amžiaus riba, IT raštingumas, ir kita reikšminga informacija. Visa ši informacija yra atvaizduojama sistemoje, kartu su informacija iš skyrių *2f Naudotojams priskirti prioritetai* ir *2g Naudotojų dalyvavimas projekte* (4.13 pav.) išskyrus *Kita reikšminga informacija*


Pavadinimas	Amžiaus riba	IT raštingumas	Dalyvavimas projekte	Pagrindimas	Prioritetas	
Gydytojas	25-64	Vidutinis	Gydytojų posistemė	Pildys paciento informaciją, tvirtins rezidentų įrašus	Pagrindinis naudotojas	 
Rezidentas	23-30	Vidutinis	Gydytojų posistemė	Pildys pacientų informaciją	Pagrindinis naudotojas	 


#### 4.13 pav. Funkcinių naudotojų informacijos atvaizdavimas sistemoje

- sistemos panaudojimo atvejai – pagal *Volere* šabloną informacija apie funkcinis reikalavimus, kurie yra panaudojimo atvejai, turėtų būti atvaizduojama lentelės forma. Lentelės užpildymo informacija pateikiama – 1.4 lentelėje. Visa informacija, kuri yra reikalinga norint aprašyti panaudojimo atvejį yra atvaizduojama sistemoje, bet ne lentelės principu. Sistemos panaudojimo atvejų vaizdas – 4.14 pav., kuris išskaidytas į keletą paveikslų, dėl aiškumo.

**Pavadinimas:**

Užpildyti anamnezę

 Tikslas

Tikslas	Pagrindimas	Pamatavimas	
Įvesti paciento informaciją	Reikalinga pildyti ligos istoriją kuo tiks	Anamnezė yra upildyta	

**Aprašymas:**

Gydytojas užpildo anamnezę: nusiskundimus, gyvenimo ir ligos pagrindinę informaciją

**Prieš-sąlyga:**

Naudotojas turi būti atsidaręs paciento kontekstą

+ Aktorius	
Aktorius	
Rezidentas	
Gdytojas	
+ Sužadinimo sąlyga	
Sužadinimo sąlyga	Loginė sąlyga
Naudotojas nori įvesti paciento informaciją	Naudotojas turi būti atsidaręs paciento kontekstą

Susiję panaudojimo atvejai	
+ PA	
Apimami PA	
Užsakyti diagnostinius tyrimus	

Veiklos scenarijai			
+ Scenarijus			
Pavadinimas	Duomenys teisingai suvesti ir pasirinkta pildyti naują anamnezę		
Naudotojo veiksmai	Sistemos veiksmai		
1	Atsidaryti anamnezės formą	2	Atidaryti formos pildymo pasirinkimą
3	Pasirinkti pildyti naują anamnezę	4	Atidaryti anamnezės formą
5	Užpildyti anamnezę		
6	Patvirtinti anamnezę	7	Išsaugoti anamnezės duomenis
		8	Atidaryti anamnezės peržiūros formą
Sąlyga po	Anamnezė išsaugojama		

#### 4.14 pav. Panaudojimo atvejų aprašomoji forma sistemoje

- nefunkciniai reikalavimai – pagal *Volere* šabloną informacija apie funkcinis reikalavimus, kurie yra panaudojimo atvejai, turėtų būti atvaizduojama lentelės forma. Lentelės užpildymo informacija pateikiama – 1.3 lentelėje. Visa informacija, reikalinga aprašyti reikalavimą yra atvaizduojama sistemoje, bet ne lentelės principu (4.15 pav.).

7

Tipas:

Pavadinimas:

Aprašymas:

Pagrindimas:

Atlikimo kriterijus:

Naudotojo tenkinimas:





Naudotojo netenkinimas:

Prioritetas:

Šaltiniai:

**4.15 pav. Nefunkcinių reikalavimų atvaizdavimas sistemoje**

- veiklos kontekstai – pagal *Volere* šabloną veiklos kontekstą aprašo lentelė, kurioje yra šie stulpeliai: pavadinimas, duomenų srauto siuntėjas, duomenų srauto gavėjas, duomenų srauto aprašymas. Tokia informacija taip pat yra atvaizduojama sistemoje (4.16 pav.).

Pavadinimas	Aprašymas	Siuntėjas	Gavėjas	
Įvedami paciento duomenys	Išsaugojami paciento ligos istorijos duomenys	Pacientas	Veikla	 
Įvedama paciento diagnozė	Gydytojas įveda paciento diagnozę	Gydytojas	Veikla	 

**4.16 pav. Veiklos kontekstų atvaizdavimas sistemoje**

## 5. Eksperimentinis reikalavimų specifikacijos sudarymo metodo tyrimas

Eksperimentinis tyrimas apima analizę, pateiktą reikalavimų aprašymo metode ir jų palyginimą su *Volere* šablono rekomendacijomis. Tai atlikus nustatyti kiekybiniai rodikliai, pagal kuriuos bus pateiktos rekomendacijos metodo tobulinimas.

### 5.1. Eksperimento planas

Eksperimentas, skirtas patikrinti reikalavimų specifikacijos sudarymo metodo, kuris naudojasi žinių baze kaip pirminių duomenų šaltiniu, tinkamumą susideda iš keturių žingsnių:

1. palyginti studentų parašytas specifikacijas su *Volere* šablono turiniu, kuris išanalizuotas 1.11 lentelėje;
2. išplėsti eksperimentinės sistemos, kuri aprašyta 3 skyriuje, funkcionalumą realizuojant specifikacijos aspektą – pagal *Volere* šabloną;
3. palyginti eksperimentinėje sistemoje naudojamų reikalavimų specifikavimo šabloninių formų sudėtį su *Volere* šablono turiniu, kuris išanalizuotas 1.11 lentelėje;
4. identifikuoti naudotojo reikalavimų specifikavimo būdus: a) panaudojant pakartotinai veiklos modelyje esančias dalykinės srities žinias, b) analitikui užpildžius *Volere* šabloną neautomatizuotu būdu, c) pakartotinai panaudojant analitiko neautomatizuotu būdu užpildytas specifikuotas žinias pagal *Volere* šabloną, generuojant grafinius UML modelius ir tekstinę informaciją.
5. Palyginti reikalavimų specifikacijos sudarymo metodo naudojamas specifikavimo formas su studentų sukurtomis specifikacijomis;
6. išanalizuoti gautus duomenis ir pateikti rekomendacijas.

### 5.2. Užpildytų specifikacijų palyginimas su *Volere* šablono rekomendacijomis

Buvo analizuojami trijų studijų programų – Didžiųjų verslo duomenų analizės, Informacinių sistemų inžinerijos, Programų sistemų inžinerijos – studentų darbai, kurie buvo ruošiami vienam iš dviejų modulių – T120M122 (Informacinių sistemų reikalavimų analizė ir specifikavimas) ir T120M015 (Reikalavimų programoms analizė). Visi darbai buvo paruošti 2018 metų rudens semestre. Iš viso buvo analizuojami 42 darbai, jų analizė lyginant su *Volere* šablono turiniu pateikiama – 5.1 lentelėje.

5.1 lentelė Užpildytų specifikacijų rezultatų palyginimas su Volere šablono turiniu

Šablono skyrius	Aprašymo struktūra pagal Volere	Aprašymo struktūra naudojama analitikų darbuose	Kiekis	Išvada
<b>1 Sistemos paskirtis</b>				
<b>1a Analizuojama veikla ir/ar projekto priežastys</b>	Laisva forma	Aprašymas	31	Visi 42 (100 %) analitikai aprašė teisingai, tik 11 (26 %) išskyrė papildomai veiklą ir projekto priežastis, taip suteikiant tikslumo ir aiškumo aprašymo skyriui.
		Laisva forma tekstas: išskiriant veiklą ir priežastis	11	
<b>1b Tikslai</b>	1.tikslas; 2. paaiškinimas; 3. tikslo rezultato išmatavimas.	1. tikslas; 2. Nauda; 3. matas (1. tikslas; 2. Aprašymas; 3. Pagrindimas; 4.tenkinimo kriterijus)	19	19 (45 %) analitikų aprašė tinkamai tikslus, o likusieji pateikė mažiau reikalingos informacijos
		Aprašymas	23	
<b>2 Suinteresuoti asmenys</b>				
<b>2a Užsakovas/2b Pirkėjas</b>	Pažymimas vardas (pavadinimas), sprendimai už kuriuos užsakovas yra atsakingas.	1. pavadinimas; 2. sprendimai už kuriuos yra atsakingas	17	24 (57 %) analitikai pažymėjo svarbiausias užsakovo/ pirkėjo duomenis – pavadinimas ir atsakingumą, projekte, o 4 (10 %) asmenys išskyrė informaciją apie užsakovo/ pirkėjo darbovietę.
		1. pavadinimas; 2. sprendimai už kuriuos yra atsakingas + papildoma informacija, kaip įmonės kodas, PVM mokėtojo kodas, adresas, pareigos, el. pašto adresas, tel. numeris, vardas pavardė.	7	
		Pateikiama informacija apie užsakovą, kaip 1. pavadinimas; 2. įmonės kodas; 3. PVM mokėtojo kodas; 4. vadovas; 5. adresas; 6. mob. telefonas; 7 el. pašto adresas;	4	
		Aprašymas	14	
<b>2c Kiti suinteresuoti asmenys</b>	1. asmens rolę; 2. pagrindimą; 3. dalyvaujančių projekte kiekį; 4. kur šis asmuo dirba 5. dalyvavimą projekte; 6. turimas žinias.	Diagrama (kaip yra šablone)	1	Svarbiausi duomenys (pavadinimas, sprendimai už kuriuos yra atsakingas) buvo aprašyti 25 (60 %) analitikų. 1 (2%) analitikas aprašė pilnai.
		1. pavadinimas; 2. kiekis; 3. aprašymas; 4. sprendimai už kuriuos yra atsakingas	1	
		1. pavadinimas; 2. kiekis; 3. darbo patirtis; 4. sprendimai už kuriuos yra atsakingas	2	



		1. pavadinimas; 2. kontaktinė informacija	1	
		1. pavadinimas; 2. pagrindimas; 3. kiekis; 4. kur dirba; 5. dalyvavimo laikotarpis	1	
		1. pavadinimą; 2. kiekį; 3. sprendimai už kuriuos yra atsakingas	2	
		1. pavadinimas; 2. pagrindimas; 3. kiekis; 4. kur dirba; 5. dalyvavimas projekte; 6. turimos žinios	1	
		1. pavadinimas; 2. kiekis; 4. darbovietė; 5. darbo patirtis; 6. tikslas	1	
		Išskiriami išoriniai ir vidiniai darbuotojai (tik pavadinimai)	1	
		1. pavadinimas; 2. aprašymas (už ką atsakingas)	19	
		Aprašymas	9	
		Neužpildytas skyrius	2	
<b>2d Potencialūs produkto naudotojai</b>	1. naudotojo rolę; 2. pagrindines atliekamas funkcijas (dalyvavimą projekte); 3. pagrindimą; 4. amžiaus ribas; 5. IT raštingumą; 5. kitą reikšmingą informaciją.	1. pavadinimas; 3. verslo suvokimo lygis; 4. IT raštingumas; 5. amžiaus grupės; 6. atliekamos funkcijos, tikslai	8	27 (64 %) analitikai aprašė naudotojus pagal rekomendacijas, vieni pridėjo papildomos informacijos kaip dalykinės srities suvokimas.
		1. pavadinimas; 2. tikslas; 3. rolės; 4. dalykinės srities išmanymas; 5. IT raštingumas; 6. prioritetas; 7. piniginiai ištekliai; 8 dalyvavimas projekte.	1	
		1. pavadinimas; 2. aprašymas; 3. prioritetas; 4. išsilavinimas; 5. atliekamos funkcijos	3	
		1. pavadinimas; 2. tikslai; 3. charakteristikos; 4. amžiaus grupė; 5. naudojami įrenginiai	1	
		1. pavadinimas; 2. atsakomybės	7	
		1. pavadinimas; 2. prioritetas; 3. funkcijos	1	
		1. pavadinimas; 2. atliekamos funkcijos, atsakomybės; 3 patirtis dalykinėje srityje; 4. IT	12	

		raštingumas; 5. kita; 6. prioritetas		
		1. pavadinimas; 2 IT raštingumas; 3. dalykinės srities suvokimas	2	
		1. pavadinimas; 2. sprendžiami uždaviniai; 3. patirtis sportinėje veikloje; 4. naudojamos kalbos; 5. papildomos charakteristikos; 6. prioritetas	1	
		1. pavadinimas; 2. užsakomi automobiliai; 3. vairavimo patirtis; 4. rizikos grupė; 5 IT raštingumas; 6. tipinės kelionės; 7. prioritetas	1	
		Aprašymas	5	
<b>2e Personažai</b>	1. amžius; 2. darbovietė; 3. šeima; 4. laisvalaikio praleidimo būdai; 5. gyvenamoji vieta; 6. mėgstami/ nemėgstami dalykai; 7. požiūris į pinigus/ technologijas; 8. kita svarbi informacija, kuri gali paveikti požiūrį į kuriamą produktą.	1. vardas; 2. amžius; 3. šeimyninė padėtis; 4. išsilavinimas; 5. pareigos; 6. pajamos; 7. IT raštingumas; 8. pomėgiai; 9. savybės; 10. ryšiai su bėgimu	1	15 (36 %) analitikų užpildė personažų skyrių, 2 (5 %) analitikai pateikė detalesnius personažus, išskiriant savybes kaip profesija, pomėgiai.
		1. vardas pavardė; 2. Amžius; 3. Profesija; 4. šeimyninė padėtis; 5. gyvenamoji vieta; 6. pomėgiai ir savybės	1	
		Aprašymas	13	
		Neužpildytas skyrius	27	
<b>2f Naudotojams priskirti prioritetai</b>	Nurodomas vienas iš prioritetų naudotojui: 1. pagrindiniai naudotojai; 2. antraeiliai naudotojai; 3. nesvarbūs naudotojai	Laisvas tekstas	1	20 (47 %) analitikų užpildė informaciją, 17 (40 %) pateikia prie 2d skyriaus
		Išskirta naudotojai prie tipų	2	
		Prie 2d	17	
		Neužpildytas skyrius	22	
<b>2g Naudotojų dalyvavimas projekte</b>	Lentelės forma	Prie 2d	15	17 (40 %) analitikų pateikė informaciją, 2 (5 %) pateikia prie 2d skyriaus
		Laisvas tekstas	2	
		Neužpildytas skyrius	25	

<b>2h Sistemos aptarnaujantis personalas</b>	Lentelės forma	Laisvas tekstas	1	10 (24%) analitikai užpildė informaciją, 4 (10 %) pateikė prie 2c ir 2d skyrių.
		Pateikta prie 2d ir 2c	4	
		1. pavadinimas; 2. už ką atsakingas	5	
		Neužpildytas skyrius	32	
<b>3 Apribojimai</b>				
<b>3a Apribojimai sprendimui/</b>	Aprašoma reikalavimų kortele (1.3 lentelė)	1. aprašas; 2. pagrindimas; 3. tinkamumas	14	Nei vienas analitikas neaprašė apribojimų sprendimui pagal rekomendacijos, bet 17 (40 %) struktūriškai aprašė apribojimus.
		1. apribojimas; 2. priežastis	3	
		Aprašymas	23	
		Neužpildytas skyrius	2	
<b>3b Diegimo aplinka</b>	Patogiausia atvaizduoti UML diegimo diagrama.	Aprašymas	7	15 (36 %) aprašė skyrių pagal rekomendacijas bei 15 (36 %) pateikė laisvos formos diagramą vietoj UML diegimo.
		UML diegimo diagrama	7	
		UML diegimo diagrama + aprašymas	8	
		Laisvos formos diagrama	8	
		Laisvos formos diagrama + aprašymas	7	
		1. pavadinimas; 2. Aprašymas; 3. pagrindimas	4	
		Neužpildytas skyrius	1	
<b>3c Komunikuojančios sistemos</b>	Patogiausia atvaizduoti UML diegimo diagrama.	1. pavadinimas; 2. aprašymas; 3 pagrindimas; 4. papildoma informacija	2	Nei vienas analitikas komunikujančių sistemų neatvaizdavo kaip diagrama, bet 12 (29 %) aprašė struktūriškai šias sistemas.
		1. aprašymas; 2. pagrindimas	10	
		Laisvas tekstas	22	
		Neužpildytas skyrius	8	
<b>3d Prieinama specializuota programinė įranga (PI)</b>	PI dokumentacijos, taip pat nurodoma kokias funkcijas, reikalavimus apima PI.	1. pavadinimas; 2. aprašymas	5	2 (5 %) pateikė PI aprašymus pagal rekomendacijas
		1. apribojimas; 2. pagrindimas; 3. tenkinimo kriterijus	2	

		Aprašymas	21	
		Neužpildytas skyrius	14	
<b>3e Numatoma darbo vietos aplinka</b>	Darbo vietos aprašymas, galima pridėti nuotraukų ar vaizdo įrašų.	Aprašymas	32	37 (88 %) analitikai pateikė darbo vietos aprašymus pagal rekomendacijas, 3 (7 %) papildomai pridėjo ir pagrindimą.
		Aprašymas + nuotrauka	2	
		1. pavadinimas; 2. aprašymas; 3. pagrindimas	3	
		Neužpildytas skyrius	5	
<b>3f Sistemos kūrimo terminas</b>	Aprašomas rašytine forma iš trijų dalių: terminas, pagrindimas terminui, pasekmės neįvykdžius darbų iki termino pabaigos.	1. pavadinimas; 2. terminas iki kada reikia įvykdyti	31	35 (83 %) nurodė terminą ir pagrindimą, kiek užtruktų įvykdyti projektą, o 3 (7 %) nurodė papildomai ir pasekmes, neįvykdžius darbų iki termino pabaigos.
		1. terminas; 2. loginis pagrindimas; 3. termino praleidimo pasekmės	3	
		Nurodoma kiek darbo valandų užtruks projektas	1	
		Laisvas tekstas	6	
		Neužpildytas skyrius	1	
<b>3g Sistemos kūrimo biudžeto apribojimai</b>	Aprašomas rašyti forma iš dviejų dalių: pinigine išraiška, šaltinis iš kur gaunamas biudžetas.	1. biudžetas; 2. aprašymas; 3. pagrindimas (šaltinis)	29	29 (69 %) analitikai apribojimą aprašė pagal rekomendacijas
		Aprašymas	8	
		Neužpildytas skyrius	5	
<b>3h Organizacijos/politiniai apribojimai</b>	Aprašoma reikalavimų kortele (1.3 lentelė)	Laisva forma	14	Nei vienas analitikas neaprašė apribojimų pagal rekomendacijas
		Neužpildytas skyrius	23	
<b>4(a) Sąvokos ir santrumpos</b>	Sąrašo forma, nurodant terminą ir jo reikšmę.	Terminas/ paaiškinimas	32	32 (76 %) analitikai aprašė sąvokas pagal rekomendacijas
		Neužpildytas skyrius	3	
<b>5 Svarbūs faktai ir prielaidos</b>				
<b>5a Faktai</b>	Laisva forma.	Laisva forma	32	32 (76 %) aprašė pagal rekomendacijas faktus.
		Neužpildytas skyrius	10	
<b>5b Veiklos taisyklės</b>	Laisva forma, kuri būtinai aprašo	Laisva forma	31	Nei vienas analitikas nepaminėjo

	veiklos taisyklę, priežastį taisyklei ir kas įgalino šią taisyklę.	Neužpildytas skyrius	11	priežasčių taisyklei, tik 31 (74 %) aprašė pačias taisykles.
<b>5c Prielaidos</b>	Laisva forma, kuri būtinai turi aprašyti prielaidą ir pasekmes, jei prielaida išsipildytu.	Laisva forma	27	3 (7 %) analiktai prie prielaidos aprašymo pridėjo ir pasekmes, kurių būtų prielaidai įvykus.
		1. prielaida; 2. Aprašymas; 3. pasekmės	3	
		Neužpildytas skyrius	12	
<b>6 Veiklos sudėtis</b>				
<b>6a Esama padėtis</b>	Siūloma naudoti UML veiklos diagramas, BPMN veiklos procesų diagramas.	Aprašymas	24	14 (33 %) analitikų nurodytą diagramą.
		UML veiklos diagrama + aprašymas	2	
		UML sekų diagrama	1	
		BPMN diagramos	1	
		BPMN diagrama + aprašymas	11	
		laisvos formos diagrama + aprašymas	2	
		Neužpildytas skyrius	1	
<b>6b Veiklos kontekstas</b>	Siūloma braižyti laisvos formos diagramą, kuri susideda iš duomenų srautų tarp suinteresuotų šalių ir galimo produkto.	Laisvos formos diagrama	41	41 (98 %) analitikų pateikė nurodytą diagramą.
		Laisvos formos tekstas	1	
<b>6c Veiklos suskaidymas (įvykių specifikacija)</b>	Lentelės pavidalu, nurodant: 1. pavadinimą; 2. duomenų srauto siuntėją; 3. duomenų srauto gavėją; 4. duomenų srauto aprašymą.	1. pavadinimas; 2. įeinantys/išeinantys informacijos srautai; 3. panaudojimo atvejo charakteristika	27	30 (71 %) analitikų pateikė svarbiausią informaciją apie veiklą: pavadinimą, informacija apie srautą.
		1. pavadinimas; 2. aprašymas	1	
		1. pavadinimas; 2. įeinantys/išeinantys informacijos srautai	2	
		1. srauto pavadinimas; 2. veikėjas kuris gauna srautą; 3. srauto siuntėjas; 4. apibūdinimas	1	
		Aprašymas	7	
		Neužpildytas skyrius	4	

<b>7 Duomenų modelis ir jo elementų žodynas</b>				
<b>7a Duomenų modelis</b>	Rekomenduojama naudoti UML klasių modelį, subjektų ryšių diagrama (angl. <i>Entity Relationship diagram</i> ).	Laisvos formos diagrama	3	39 (93 %) analitikų pateikė nurodytą diagramą.
		UML klasių diagrama + aprašymas	8	
		UML klasių diagrama	27	
		Subjektų ryšių diagrama	4	
<b>7b Duomenų žodynas</b>	Lentelės forma su informacija: 1. klasės pavadinimas; 2. atributai, jų aprašymai ir galimų reikšmių apribojimai; 3. klasės ryšiai su aprašymais.	1. klasės pavadinimas, aprašymas; 2. atributo pavadinimas, aprašymas, ar atributas yra raktas, duomenų tipas, duomenų ilgis	1	36 (86 %) analitikas paminėjo svarbią informaciją apie esybes ir atributus: esybės pavadinimas, aprašymą, o apie atributus, pavadinimą, apribojimus reikšmei. 1 (2 %) analitikas dar paminėjo ryšių pavadinimus, aprašymus ir jų apribojimus.
		1. klasės pavadinimas, aprašymas; 2. ryšio/ atributo pavadinimas, aprašymas, apribojimai, ar reikšmė privaloma	1	
		Klasės pavadinimas, aprašymas	4	
		1. klasės pavadinimas, aprašymas; 2. atributo pavadinimas, apribojimai	16	
		1. klasės pavadinimas, aprašymas; 2. atributo pavadinimas, aprašymas, apribojimai, ar reikšmė privaloma	18	
		Neužpildytas skyrius	2	
<b>8 Sistemos sudėtis (panaudojimo atvejų modelis)</b>				
<b>8a Sistemos ribos (panaudojimo atvejų diagrama)</b>	Atvaizduojama UML panaudojimo atvejų diagrama.	UML panaudojimo atvejų diagrama	42	Visi analitikai teisingai užpildė skyrių
<b>8bc Panaudojimo atvejai</b>	Atvaizduojama lentelės forma – 1.4 lentelė.	Kaip PA lentelė (1.4 lentelė)	36	36 (86 %) analitikai teisingai aprašė PA skyrių, 4 (10 %) tik dalinai aprašė siūlomą lentelę (1.4 lentelė)
		1. pavadinimas; 2. Aprašymas; 3. prieš sąlyga; 4. po sąlyga; 5. scenarijai; 6. susiję PA	2	
		1. pavadinimas; 2. Aprašymas; 3. prieš sąlyga; 4. po sąlyga; 5. sužadavimo sąlyga; 6. susiję PA; 7. tikslas	2	
		Aprašymas + BPMN diagramos	1	

		Neužpildytas skyrius	1	
<b>9 Papildomi funkciniai reikalavimai – 17 Atitikties reikalavimai;</b>	Aprašoma reikalavimų kortele (1.3 lentelė)	Kaip reikalavimų kortelė (1.3 lentelė)	34	34 (81 %) analitikai aprašė teisingai šiuos skyrius, 5 (12 %) analitikai aprašė dalinai teisingai
		1. pavadinimas; 2. aprašymas; 3. pagrindimas; 4. tenkinimo kriterijus	1	
		Dalis reikalavimų kaip reikalavimų kortelė (1.3 lentelė), kiti kaip aprašymas	4	
		Aprašymas	3	
<b>18 Atviros problemos ir klausimai</b>	Lentelės forma aprašoma: 1. numeris; 2. sąsaja su produkto dalimi (funkcijos numeris, reikalavimo numeris ir pan.); 3. aprašymas; 4. susijusios suinteresuotos šalys; 5. veiksmai, kas galėtų įvykti; 6. problemos sprendimo būdas;	1. numeris; 2. problema; 3. komentaras; 4. suinteresuoti asmenys; 5. sprendimas	3	Nei vienas analitikas neaprašė atvirų problemų ir klausimų pagal rekomendacijas. 3 (7 %) analitikai beveik aprašė teisingai, bet pamiršo paminėti problemos sąsają su reikalavimais
		1. problema; 2. paaiškinimas; 3. sprendimas	1	
		Aprašymas	20	
		Neužpildytas skyrius	18	
<b>19 Egzistuojantys sprendimai</b>	Sąrašo forma, paminint papildomą informaciją apie produktą kaip: 1. kaina; 2. prieinamumas; 3. laiko intervalas per kurį galima sukurti reikalingą integraciją.	1. pavadinimas; 2. aprašymas; 3. nuoroda; 4. padengia kuriuos funkcinis reikalavimus; 5. kaina	2	Nei vienas analitikas neaprašė egzistuojančių sprendimų pagal rekomendacijas
		1. pavadinimas; 2. aprašymas	2	
		1. pavadinimas; 2. aprašymas; 3. kaina	1	
		Laisvas tekstas	37	
<b>20 Naujos problemos</b>	Laisva tekstine forma.	1. problema; 2. komentaras; 3. suinteresuoti asmenys; 4. Priemonės; 5. sprendimas	1	37 (88 %) analitikai teisingai aprašė naujas problemas, bet 3 (7 %) analitikai aprašinėjo labiau struktūrizuota forma nei paprastas tekstinis aprašymas
		1. pavadinimas; 2. poveikis	1	
		1. pavadinimas; 2. poveikis; 3. sprendimas	1	
		Aprašymas	34	
		Neužpildytas skyrius	5	
<b>21 Uždaviniai</b>				
<b>21a Sistemos kūrimo procesas</b>	Siūloma atvaizduoti aukšto lygio	Diagrama	8	15 (36 %) analitikų pateikė sistemos

	procesų diagrama. Taip pat pridėti aprašymą.	Diagrama su aprašymu	7	kūrimo procesų diagramą, bet 27 (64 %) pateikė proceso aprašymą.
		Išvardinti etapai	6	
		Proceso aprašymas ir išvardinti proceso etapai	5	
		Aprašymas	15	
		Neužpildytas skyrius	1	
<b>21b Detalus kūrimo planas</b>	Galima braižyti LEAN grafiką ir pridėti aprašymą, kuris nurodo: 1. fazės pavadinimas; 2. naudotojo nauda; 3. atlikimo data; 4. naudojami komponentai, funkciniai ir nefunkciniai reikalavimai.	LEAN grafikas	24	Teisingiausiai aprašė 8 (19 %) analitikai (1. pavadinimas; 2. tikslas; 3. data; 4. susiję reikalavimai), bet kitą dalį informacijos LEAN grafiką pateikė 28 (67 %) analitikai, kas suteikia aišku ir paprastumo duomenims.
		LEAN grafikas + aprašymas (laisvas tekstas)	4	
		1. etapas; 2. aprašymas; 3. trukmė	3	
		1. pavadinimas; 2. tikslas; 3. data; 4. susiję reikalavimai	8	
		Aprašymas	1	
		Neužpildytas skyrius	2	
<b>22 Migravimas į naują produktą</b>	Kaip reikalavimų kortelė (1.3 lentelė)	1. aprašymas; 2. pagrindimas; 3. tenkinimo kriterijus	2	3 (7 %) analitikai teisingai aprašė sistemos migravimo reikalavimus, kaip reikalavimų kortelę (). Bet dar 2 (5%) naudojos apribojimų forma: aprašymas, pagrindimas ir tenkinimo kriterijus.
		Kaip reikalavimų kortelė (1.3 lentelė)	3	
		Aprašymas	26	
		Neužpildytas skyrius	11	
<b>23 Rizikų įvertinimas</b>	Lentelės forma, nurodant: 1. pavadinimas; 2. aprašymas; 3. tikimybė, kad rizika įvyks; 4. poveikis ir jo paaiškinimas; 5. planas minimizuoti rizikos faktorių; 6. planas sumažinti rizikos poveikį;	1. rizika; 2. aprašymas; 3. tikimybė; 4. poveikis.	9	16 (38 %) analitikų aprašė riziką svarbiausiais kriterijais: pavadinimu, aprašymu, tikimybe ir sprendimo būdu. 4 (10 %) analitikų sprendimo būdą išskaidė į mažinimo rizikos faktoriaus ir rizikos minimizavimo, bei paskaidė poveikį į finansinį ir bendrą poveikius.
		1. kategorija; 2. subkategorija; 3. rizikos veiksniai; 4. galimas padarinys; 5. tikimybė; 6. kada nustatyta; 7. atsakingas asmuo; 8. rizikos reitingas; 9. galimas sprendimas	1	
		1. rizika; 2. tikimybė	2	
		1. pavadinimas; 2. pasekmės; 3. galimas sprendimas	1	







		1. pavadinimas; 2. sukėlimas; 3. kategorija; 4. galimas padarinys; 5. tikimybė; 6. poveikis	1	
		1. pavadinimas; 2. tikimybė; 3. sprendimo būdas	5	
		1. pavadinimas; 2. aprašymas; 3. tikimybė; 4. poveikis; 5. finansinis poveikis; 6. planas minimizuoti riziką; 7. planas sumažinti rizikos faktorių	4	
		1. pavadinimas; 2. sprendimo būdas	4	
		Aprašymas	12	
		Neužpildytas skyrius	2	
<b>24 Kaštai</b>	Laisva forma.	Kaštai paskirstyti reikalingai įrangai, kad projektas būtų įgyvendintas	3	40 (95 %) analitikų teisingai aprašė kaštų skyrių, kadangi nebuvo iškeltų specifinių reikalavimų šiam skyriui. Bet 26 (41 %) analitikas pagrindė reikalingų kaštų kiekį.
		Kaštai paskirstyti pagal sistemos dalis (reikalavimų apribojimai, funkciniai reikalavimai ir pan)	2	
		Kaštai paskirstyti etapais	10	
		Kaštai paskirstyti pagal reikalingų darbuotojų valandas	11	
		Aprašymas	14	
		Neužpildytas skyrius	2	
<b>25 Naudotojo dokumentacija ir apmokymai;</b>	Kaip reikalavimų kortelė (1.3 lentelė)	1. pavadinimas; 2. organizatorius; 3. trukmė; 4. dalyvių kiekis; 5. planuojamų sesijų kiekis	1	5 (12 %) analitikai aprašė naudotojo dokumentacijos ir apmokymų skyrelį kaip reikalavimų kortelę (1.3 lentelė). O 5 (12 %) analitikai paminėjo šias savybes – mokymų pavadinimas, naudotojai, kurie dalyvaus ir atsakingas asmuo už mokymus.
		1. pavadinimas; 2. Pagrindimas; 3. tenkinimo kriterijus	2	
		1. pavadinimas; 2. dokumento paskirtis; 3. naudotojai kurie dalyvaus; 4. atsakingas asmuo už mokymus	3	
		Kaip reikalavimų kortelė (1.3 lentelė)	5	

		1. aprašymas; 2. naudotojai kurie dalyvaus; 3. atsakingas asmuo už mokymus	1	
		Aprašymas	26	
		Neužpildytas skyrius	4	
<b>26 Perspektyviniai reikalavimai</b>	Kaip reikalavimų kortelė (1.3 lentelė)	1. aprašymas; 2. pagrindimas; 3. tenkinimo kriterijus	1	3 (7 %) analitikai aprašė kaip nurodyta yra Volere reikalavimų kortelėje (1.3 lentelė)
		Reikalavimų kortelė (1.3 lentelė)	3	
		Aprašymas	20	
		Neužpildytas skyrius	18	
<b>27 Idėjos sprendimams</b>	Laisva forma.	Laisva forma	18	18 (43 %) analitikų pateikė duomenis nurodyta forma duomenis, kadangi likusieji nepildė šio skyriaus.
		Neužpildytas skyrius	24	

### 5.3. Volere šabono atvaizdavimo rezultatai

Šiame skyriuje bus aprašomos sudėtingesnės reikalavimų specifikavimo šabloninės formos, kurios nėra tik vienas laisvo teksto įrašymo laukas. Pilnesnę specifikaciją galima peržiūrėti 2 priede esančioje metodo sudarytoje specifikacijoje. Šiame dokumente nėra pasikartojančių reikalavimų lentelių.

1b skyrius – Tikslai – aprašo sistemos tikslus, kurie bus įgyvendinti naudojantis siūlomu produktu. Tikslai aprašomi trimis laisvos formos laukais: pavadinimas, pagrindimas ir pamatavimo būdas (5.1 pav.).

Tikslas	Pagrindimas	Pamatavimo būdas	
Supaprastinti informacijos vedimą	Norima sutaupyti laiką rašant paciento informaciją raštu	Pildymas tampa bent 10 proc. greitesnis sistemoje, nei pildant ranka	 
Įvesti informacijos pildymo šabloną	Norima utikrinti visos reikalingos informacijos įrašymą	Išrašuose yra visa reikalinga informacija	 

5.1 pav. Tikslų aprašymas

2c skyrius – Kiti suinteresuoti asmenys – aprašomi visi darbuotojų tipai, kurie dirbs prie produkto, tiek įmonėje, kuri atliks vykdymo darbus, tiek įmonėje, kurioje bus naudojamas produktas. Suinteresuoto asmens aprašymas susideda iš septynių laukų – šeši iš jų yra laisvos formos laukai (pavadinimas, aprašymas, pagrindimas, darbovietė, dalyvavimas projekte ir turimos žinios) ir vienas skaičiaus įvedimas (suinteresuotų asmenų kiekis) (5.2 pav.).

Suinteresuoto asmens vaidmuo	Aprašymas	Pagrindimas	Dalyvaujančių projekte kiekis	Kur dirba darbuotojas?	Dalyvavimas projekte	Turimos žinios	
IT specialistas	Ligoninės IT specialistas, kuris ves mokymus	Darbuotojas, turi suvokti visą sistemą, kad galėtų prvesti mokymus	3	Ligoninėje	Visas projektas	Ligoninėje vykstantys procesai	 
Projekto vadovas	Projektų vadovas, vadovauja projektų sričių vadovams	Reikalingas asmuo, kad planuotų projekto darbus, sektų jų atlikimą.	1	Įmonėje, kuri atlieka sistemos kūrimą	Visas projektas	Tikslai, darbų tvarkaraštis	 



5.2 pav. Kitų suinteresuotų asmenų aprašymas

2d skyrius – Potencialūs produkto naudotojai – aprašo asmenis, kurie naudosis produktu. Šis skyrius atitinka VM funkcijų vykdytojus. Naudotoją aprašo šeši laukai: trys iš jų tekstiniai laukai, vienas – skaičių ribos, du – pasirinkimai iš klasifikatorių (5.3 pav.)

Pavadinimas	Pagrindimas	Dalyvavimas projekte	Amžiaus riba	IT raštingumas	Prioritetas	
Administratorius	Kurs gydytojų kabinetų grafikus, kabinetų talonus	Administratoriaus funkcijos	23-64	Vidutinis	Nesvarbus naudotojas	 
IT administratorius	Administruos naudotojus, kabinetus ir skyrius	Visa sistema	23-64	Aukštas	Nesvarbus naudotojas	 
Gydytojas	Pildys paciento informaciją, tvirtins rezidentų įrašus	Gydytojų posistemė	25-64	Vidutinis	Pagrindinis naudotojas	 





5.3 pav. Potencialių produkto naudotojų aprašymas

Reikalavimų aprašymo struktūrą (1.3 lentelė) naudoja 3a Apribojimai sprendimui, 3 h Organizacijos/politiniai apribojimai, 9 Papildomi funkciniai reikalavimai – 17 Atitikties reikalavimai, 22 Migravimas į naują produktą, 25 Naudotojo dokumentacija ir apmokymai, 26 Perspektyviniai reikalavimų skyriai (5.4 pav.). Nurodomi laukai su pasirinkimais iš klasifikatoriaus: reikalavimo tipas, PA/FR#, naudotojo tenkinimas ir netenkinimas, prioritetas.

Reikalavimas #	1	Reikalavimo tipas	3a	PA/FR#	Visa sistema	 
Aprašymas	Norint suteikti lanktumo ir pasiekiamumo sistemai, ji privalo būti pasiekama per interneto naršyklę					
Pagrindimas	Užtikrinti lengvą sistemos pasiekimą					
Atlikimo kriterijus	Sistemą galima pasiekti per vieną iš šių naršyklių: Google Chrome, Mozilla FireFox, Edge					
Naudotojo tenkinimas:	4	Naudotojo netenkinimas	5			
Prioritetas	Aukštas					
Šaltiniai						
Paskutinio atnaujinimo data	2020-02-18T00:00:00					



5.4 pav. Reikalavimų aprašymas

5b skyrius – Veiklos taisyklės – aprašomos trimis laisvos formos laukais – taisyklė, aprašymas ir pagrindimas (5.5 pav.)

Taisyklė	Aprašymas	Pagrindimas	
8-8 para	Para skaičiuojama nuo 8 ryto iki 8 ryto	Kadangi darbuotoju pamaina keičiasi 8 ryto.	 
SPS amb vizitas	Skubios pagalbos skyriuje vizitai formuojami kaip ambulatoriniai, nors pacientai yra guldomi	Kadangi SPS vizitai yra vienos paros.	 





5.5 pav. Veiklų taisyklių aprašymas

5c skyrius – Prielaidos – aprašomas trimis laisvos formos laukais: prielaida, aprašymas ir pasekmės, jei aprašyta prielaida įvyks (5.6 pav.).

Prielaida	Aprašymas	Pasekmės	
Pacientų registracijos	Yra numatoma koreguoti nacionalinę pacientų išankstinę registraciją	Nepakoregus integracijos su išankstiniu registraciniu sistema, nebebus gaunamos registracijos i IS	 

5.6 pav. Prielaidų aprašymas

6c skyrius – Veiklos suskaidymas – aprašomas keturiais laisvos formos laukais: veiklos pavadinimas, jos aprašymas, duomenų siuntėjas ir jų gavėjas (5.7 pav.)

Pavadinimas	Aprašymas	Siuntėjas	Gavėjas	
Įvedami paciento duomenys	Išsaugojami paciento ligos istorijos duomenys	Pacientas	Veikla	 
Įvedama paciento diagnozė	Gdytojas įveda paciento diagnozę	Gdytojas	Veikla	 

5.7 pav. Veiklos suskaidymo aprašymas

7b skyrius – Duomenų žodynas – aprašomas trejomis esybėmis: aprašoma esybė su dviem laikais: pavadinimo ir aprašymo. Esybės atributai ir ryšiai aprašomi vienodais laukais: pavadinimas, aprašymas ir galimos reikšmės apribojimai – laisvos formos laukais arba privaloma reikšmė – loginė sąlyga taip arba ne (5.8 pav.)

Esybė				
Darbuotojas		Aprašomi visi darbuotojai, kurie gali naudotis sistema, tiek visas med. personalas, tiek administracinis		
Atributai				
Nr.	Pavadinimas	Aprašymas	Galimos reikšmės apribojimai	Ar reikšmė privaloma?
1	Id	identifikatorius	susideda iš skaičių	Reikšmė privaloma
2	Darbuotojo_Kodas	Darbuotojo spaudos numeris	Spaudos numeris dažniausiai susideda iš skaičių, bet yra specialybių kurių spaudos nr turi ir raidžių junginius	Reikšmė privaloma
Ryšiai				
Nr.	Pavadinimas	Aprašymas	Galimos reikšmės apribojimai	Ar reikšmė privaloma?
1	Zmogus_Id	Zmogaus identifikatoriaus priskyrimas darbuotojui	Skaičius	Reikšmė privaloma



5.8 pav. Duomenų žodyno aprašymas

8bs skyrius – Panaudojimo atvejai – aprašomas pagal *Volere* funkcinių reikalavimų lentelę – 1.4 lentelė (5.9 pav.). Nurodomi laukai su pasirinkimais iš klasifikatoriaus: aktorius, susiję panaudojimo atvejai.

PA	Užpildyti anamnezę	
Tikslas	Pagrindimas	Pamatavimo būdas
Įvesti paciento informaciją	Reikalinga pildyti ligos istoriją kuo tiksliau	Anamnezė yra upildyta
Aprašymas	Gdytojas užpildo anamnezę: nusiskundimus, gyvenimo ir ligos pagrindinę informaciją	
Prieš sąlyga	Naudotojas turi būti atsidares paciento kontekstą	
Aktorius	Rezidentas;	
Sužadavimo sąlyga	Naudotojas nori įvesti paciento informaciją	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	Užsakyti diagnostinius tyrimus;
	Apibendrinti PA	

5.9 pav. Panaudojimo atvejų aprašymas

18 skyrius – Atviros problemos ir klausimai – aprašomi keturiais laisvos formos laukais: aprašymas, veiksmai, kas galėtų įvykti, problemos sprendimo būdas, o sąsaja su produkto dalimi – pasirenkamas laukas su klasifikatoriumi (5.10 pav.).

Numeris	Sąsaja su produkto dalimi	Aprašymas	Veiksmai, kas galėtų įvykti	Problemos sprendimo būdas	
1	2;	Gali pasikeisti ESPBI IS struktūra	1. Gali reikėti persiusti daugiau informacijos; 2. Gali pasikeisti informacijos laukai; 3. Gali pasikeisti vizitu struktūra	1. Sudaryti kuo paprastesnę formą perduoti duomenis iš Ligoninės IS į ESPBI IS. 2. Pasamdyti programuotojus kuo greičiau pakeisti persiuntimo formas; 3. Naudoti kuo daugiau ESPBI IS langų, o ne persiuntinėti duomenis	 



### 5.10 pav. Atvirų problemų ir klausimų aprašymas

21b skyrius – Detalus kūrimo planas – aprašomas dviejų dalių pagrindinis etapas: pavadinimas – laisvas tekstas, pradžios ir pabaigos datos – datos formatu. Antroji dalis – vidiniai etapai, kurie aprašomi: apvadinimu, pagrindimu – laisvos formos tekstas, pradžia ir pabaiga – datos formatas, susiję reikalavimai – pasirinkimas iš sąrašo (5.11 pav.)

Pavadinimas	Pradžia	Pabaiga		
Gydytojų posistemės funkcijos	2020-02-01T00:00:00	2020-06-06T00:00:00		 
Pavadinimas	Pagrindimas	Pradžia	Pabaiga	Susiję reikalavimai
1 Surinkti reikalavimus funkcijoms	Išsiaiškinti visus funkcijų poreikius	2020-02-05T00:00:00	2020-03-03T00:00:00	
2 Suprogramuoti LA funkcijas	Suprogramuoti pagrindines ligos atvejo funkcijas	2020-03-03T00:00:00	2020-04-04T00:00:00	1;2;
3 Suprogramuoti diagnostikos modulį	Suprogramuoti diagnostinių siuntimų ir atsakymo rašymo modulį	2020-03-03T00:00:00	2020-03-20T00:00:00	2;1;

### 5.11 pav. Detalaus kūrimo plano aprašymas

23 skyrius – Rizikų vertinimas – aprašomas trimis dalimis: rizikos aprašymas, planas minimizuoti riziką ir planas sumažinti rizikos poveikį. Pati rizika aprašoma: pavadinimu, aprašymu, tikimybe (skaičiaus įvedimas), skaitiniu ir finansiniu poveikiu (skaičiaus įvedimas), jų paaiškinimu. Abu planai susideda iš etapų, kurie aprašomi laisva forma. (5.12 pav.)

Pavadinimas	Sistema nebus patogiu naudotis	 
Aprašymas	Sistema bus nepatogi, sunku joje susigaudyti, paprasti veiksmai reikalaus daug mygtukų paspaudimų	
Tikimybė, kad rizika įvyks	40	
Poveikis	8	
Poveikio paaiškinimas	Sistema prailgins darbus lygintan su popierinės dokumentacijos pildymu	
Finansinis poveikis	60000	
Finansinio poveikio paaiškinimas	pinignė suma, kuri būtų skita sistemos dizaino ir procesų keitimui	
<b>Planas minimizuoti rizikos faktorių</b>		
1	panašių sistemų analizė	
2	išsamūs panaudojamumo testavimai, kuriuose dalyvauja med. personalas	
<b>Planas sumažinti rizikos poveikį</b>		
1	dizaino keitimas naudojant papildomą personalą	

### 5.12 pav. Rizikų įvertinimo aprašymas

#### **5.4. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas**

Palyginamoji lentelė (5.2 lentelė) tarp sukurtame reikalavimų specifikacijos sudarymo metode naudojamų reikalavimų aprašomųjų formų laukų ir *Volere* šablono skyrių turinio.

## 5.2 lentelė Palyginimas tarp sukurto reikalavimų specifikavimo metodo ir *Volere* šablono

Šablono skyrius	Aprašymo struktūra pagal <i>Volere</i>	Aprašymo struktūra metode	Informacija gaunama	Išvada
<b>1 Sistemos paskirtis</b>				
<b>1a Analizuojama veikla ir/ar projekto priežastys</b>	Laisva forma	Teksto įrašymo laukai – 1. Pavadinimas; 2. Projekto aprašymas; 3. Kūrimo priežastys.	Pakartotinai panaudojant VM žinias	Sukurta metode išskiriami 3 aprašymo laukai.
<b>1b Tikslai</b>	1.tikslas; 2. paaiškinimas; 3. tikslo rezultato išmatavimas.	Teksto įrašymo laukai – 1. tikslas; 2. pagrindimas; 3. pamatavimo būdas.	Pakartotinai panaudojant VM žinias	Sukurto metodo forma atitinka <i>Volere</i> skyrių.
<b>2 Suinteresuoti asmenys</b>				
<b>2a Užsakovas/2b Pirkėjas</b>	Pažymimas vardas (pavadinimas), sprendimai už kuriuos užsakovas yra atsakingas.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. dalyvavimas projekte.	Analitikas įveda informaciją	Sukurta metodas turi papildomą aprašymo lauką.
<b>2c Kiti suinteresuoti asmenys</b>	1. asmens rolę; 2. pagrindimą; 3. dalyvaujančių projekte kiekį; 4. kur šis asmuo dirba 5. dalyvavimą projekte; 6. turimas žinias.	Teksto įrašymo laukai: 1. suinteresuoto asmens vaidmuo; 2. aprašymas; 3. pagrindimas; 4. kur dirba darbuotojas?; 5. dalyvavimas projekte; 6. turimos žinios; Skaičiaus įrašymo laukai: 1. dalyvaujančių projekte kiekis.	Analitikas įveda informaciją	Sukurta metodas turi papildomą aprašymo lauką ir yra apribojamas laukui „Dalyvaujančių asmenų projekte kiekis“ – galima įvesti tik skaičių.
<b>2d Potencialūs produkto naudotojai</b>	1. naudotojo rolę; 2. pagrindines atliekamas funkcijas (dalyvavimą projekte); 3. pagrindimą; 4. amžiaus ribas; 5. IT raštingumą; 5. kitą reikšmingą informaciją.	Teksto įrašymo laukai: 1. pavadinimas; 2. pagrindimas; 3. dalyvavimas projekte; 4. amžiaus riba. laukai su klasifikatoriais: 1. IT raštingumas (pasirinkimas: aukštas, vidutinis, žemas); 2. prioritetas (pasirinkimas: pagrindinis naudotojas; antraeilis naudotojas; nesvarbus naudotojas).	Pakartotinai panaudojant VM žinias	Sukurta metodas turi papildomą lauką Prioritetas su pasirinkimo variantas ir laukui „IT raštingumas“ pridėti pasirinkimo variantai.



<b>2e Personāžai</b>	1. amžius; 2. darbovietē; 3. šēma; 4. laisvalaikio praleidimo būdai; 5. gyvenamoji vieta; 6. mēgstami/nemēgstami dalykai; 7. požiūris ī pinigus/ tehnologijas; 8. kita svarbi informacija, kuri gali paveikti požiūrī ī kuriamā produktā.	Informacija nēra aprāšoma metode.		Nusprēsta nepildyti metode šio skyriaus informacijos, kadangi turinys yra pakankamai abstraktus ir skyrius retai pildomas.
<b>2f Naudotojams priskirti prioritetai</b>	Nurodomas vienas īs prioritētū naudotojui: 1. pagrindiniai naudotojai; 2. antraeiliai naudotojai; 3. nesvarbūs naudotojai.	Atvaizduojama lentelēs īs duomenū suvestū 2d skyriuje. Informacija atvaizduojama lentelēs forma: 1. prioritetas; 2. funkcijū vykdytojų sārāšas.	Īs 2d skyriaus	Sukurto metodo forma atitinka Volere šablono turinī.
<b>2g Naudotojų dalyvavimas projekte</b>	Lentelēs forma.	Atvaizduojama lentelēs īs duomenū suvestū 2d skyriuje. Informacija atvaizduojama lentelēs forma: 1. pavadinimas (funkcijos vykdytojo); 2. dalyvavimas projekte.	Īs 2d skyriaus	Sukurto metodo forma atitinka Volere šablono turinī.
<b>2h Sistemos aptarnaujantis personalas</b>	Lentelēs forma.	Atvaizduojama lentelēs īs duomenū suvestū 2c skyriuje. Informacija atvaizduojama lentelēs forma: 1. suinteresuoto asmens vaidmuo; 2. dalyvaujančių projekte kiekis; 3. kur dirba darbuotojas?; 4. dalyvavimas projekte.	Īs 2c skyriaus	Sukurto metodo forma atitinka Volere šablono turinī.
<b>3 Apribojimai</b>				

<b>3a Apribojimai sprendimui/ 3 h Organizacijos/politiniai apribojimai; 9 Papildomi funkciniai reikalavimai – 17 Atitikties reikalavimai; 22 Migravimas į naują produktą; 25 Naudotojo dokumentacija ir apmokymai; 26 Perspektyviniai reikalavimai</b>	Aprašoma reikalavimų kortele (1.3 lentelė).	Skaičiaus įrašymo laukai: 1. numeris # Teksto įrašymo laukai: 1. aprašymas; 2. pagrindimas; 3. atlikimo kriterijus; 4. Šaltiniai. Laukai su klasifikatoriais: 1. reikalavimo tipas (pasirinkimas iš sąrašo – <i>Volere</i> skyriaus numeris); 2. PA/FR# (PA ir reikalavimų sąrašai); 3. naudojimo tenkinimas (pasirinkimas 1 – 5); 4. naudotojo netenkinimas (pasirinkimas 1 – 5); 5. prioritetas (pasirinkimas: aukštas, vidutinis, žemas). Datos įrašymo laukai: 1. paskutinio redagavimo data (automatiškai paimamas išsaugojimo laikas).	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį. Papildomai yra penki laukai su pasirinkimais.
<b>3b Diegimo aplinka</b>	Patogiausia atvaizduoti UML diegimo diagrama.	Generuojama diagrama iš 3c skyriaus.	Diagrama	Sukurtame metode nerealizuotas diagramos generavimas
<b>3c Komunikuojančios sistemos</b>	Patogiausia atvaizduoti UML diegimo diagrama.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. sistemos sudedamosios dalys: 3.1. pavadinimas; 3.2. aprašymas.	Analitikas įveda informaciją	Sukurtame metode informacija atvaizduojame tekstu, <i>Volere</i> šablonas rekomenduoja naudoti diagramą.
<b>3d Prieinama specializuota programinė įranga (PI)</b>	PĮ dokumentacijos, taip pat nurodoma kokias funkcijas, reikalavimus apima PĮ.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas.	Analitikas įveda informaciją	Sukurtame metode nėra išskirti laukai kokias funkcijas ir reikalavimus apima PĮ, bet šią informaciją galima įvesti į lauką „Aprašymas“.
<b>3e Numatoma darbo vietos aplinka</b>	Darbo vietos aprašymas, galima pridėti nuotraukų ar vaizdo įrašų.	Teksto įrašymo laukai: 1. aprašymas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>3f Sistemos kūrimo terminas</b>	Aprašomas rašytine forma iš trijų dalių: terminas, pagrindimas terminui, pasekmės neįvykdžius darbų iki termino pabaigos.	Teksto įrašymo laukai: 1. pagrindimas; 2. pasekmės. Datos įrašymo laukai: 1. data.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>3g Sistemos kūrimo biudžeto apribojimai</b>	Aprašomas rašyti forma iš dviejų dalių: piniginė išraiška, šaltinis iš kur gaunamas biudžetas.	Teksto įrašymo laukai: 1. paaiškinimas Skaičiaus įrašymo laukai: 1. biudžetas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>4(a) Sąvokos ir santrumpos</b>	Sąrašo forma, nurodant terminą ir jo reikšmę.	Teksto įrašymo laukai: 1. sąvoka; 2. paaiškinimas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.

<b>5 Svarbūs faktai ir prielaidos</b>				
<b>5a Faktai</b>	Laisva forma.	Teksto įrašymo laukai: 1. faktas; 2. aprašymas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>5b Veiklos taisyklės</b>	Laisva forma, kuri būtinai aprašo veiklos taisyklę, priežastį taisyklei ir kas įgalino šią taisyklę.	Teksto įrašymo laukai: 1. taisyklė; 2. aprašymas; 3. pagrindimas.	Pakartotinai panaudojant VM žinias	Sukurto metodo forma atitinka Volere šablono turinį.
<b>5c Prielaidos</b>	Laisva forma, kuri būtinai turi aprašyti prielaidą ir pasekmes, jei prielaida išsipildytu. Priklausomai nuo prielaidos sudėtingumo, gali reikėti pateikti papildomų šaltinių, kurie suteiktų daugiau informacijos.	Teksto įrašymo laukai: 1. taisyklė; 2. aprašymas; 3. pasekmės.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>6 Veiklos sudėtis</b>				
<b>6a Esama padėtis</b>	Siūloma naudoti UML veiklos diagramas, BPMN veiklos procesų diagramas.	Sugeneruojamos diagramos iš VM Procesų ir papildomos informacijos.	Pakartotinai panaudojant VM žinias + diagrama	Sukurta metode nerealizuotas diagramos generavimas.
<b>6b Veiklos kontekstas</b>	Siūloma braižyti laisvos formos diagramą, kuri susideda iš duomenų srautų tarp suinteresuotų šalių ir galimo produkto.	Iš užpildytos informacijos iš 6c skyriaus.	Diagrama	Sukurta metode nerealizuotas diagramos generavimas.
<b>6c Veiklos suskaidymas (įvykių specifikacija)</b>	Lentelės pavidalu, nurodant: 1. pavadinimą; 2. duomenų srauto siuntėją; 3. duomenų srauto gavėją; 4. duomenų srauto aprašymą.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. siuntėjas; 4. gavėjas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>7 Duomenų modelis ir jo elementų žodynas</b>				
<b>7a Duomenų modelis</b>	Rekomenduojama naudoti UML klasių modelį, dubjektų ryšių diagrama (angl. <i>Entity Relationship diagram</i> ).	Iš užpildytų duomenų iš 7b skyriaus.	Diagrama	Sukurta metode nerealizuotas diagramos generavimas.

<b>7b Duomenų žodynas</b>	Lentelės forma su informacija: 1. klasės pavadinimas; 2. atributai, jų aprašymai ir galimų reikšmių apribojimai; 3. klasės ryšiai su aprašymais.	Teksto įrašymo laukai: 1. esybės pavadinimas; 2. esybės aprašymas; 3. atributai: 3.1. pavadinimas; 3.2. aprašymas; 3.3. galimos reikšmės apribojimai; 3.4. ar privaloma reikšmė? (loginė sąlyga); 4. ryšiai: 4.1. pavadinimas; 4.2. aprašymas; 4.3. galimos reikšmės apribojimai; 4.4. ar privaloma reikšmė? (loginė sąlyga).	Analitikas įveda informaciją	Sukurtame metode papildomai yra nurodoma ar atributas/ ryšys yra privalomas.
<b>8 Sistemos sudėtis (panaudojimo atvejų modelis)</b>				
<b>8a Sistemos ribos (panaudojimo atvejų diagrama)</b>	Atvaizduojama UML panaudojimo atvejų diagrama.	Iš užpildytų duomenų iš 8bc skyriaus.	Diagrama	Metode generuojamoje diagramoje panaudojimo atvejai nėra sugrupuoti į posistemas.
<b>8bc Panaudojimo atvejai</b>	Atvaizduojama lentelės forma – 1.4 lentelė.	Teksto įrašymo laukai: 1. PA; 2. tikslas: 2.1. pavadinimas; 2.2. pagrindimas; 2.3. pamatavimo būdas; 3. aprašymas; 4. prieš sąlyga; 5. sužadavimo sąlyga; 6. scenarijai: 6.1. pavadinimas; 6.2. naudotojo veiksmai (matomi PA informacijoje) 6.3. sistemos veiksmai (matomi PA informacijoje); 6.4. rezultatas. Laukai su klasifikatoriais: 1. aktorius (pasirinkimas iš užregistruotų funkcijos vykdytojų); 2. susiję panaudojimo atvejai (pasirinkimas iš užregistruotų funkcijų).	Pakartotinai panaudojant VM žinias + analitikas įveda informaciją	Skurto metodo forma atitinka <i>Volere</i> šablono turinį. Papildomai metode yra naudojami du laukai su pasirinkimais.
<b>18 Atviros problemos ir klausimai</b>	Lentelės forma aprašoma: 1. numeris; 2. sąsaja su produkto dalimi (funkcijos numeris, reikalavimo numeris ir pan.); 3. aprašymas; 4. susijusios suinteresuotos šalys; 5. veiksmai, kas galėtų įvykti; 6. problemos sprendimo būdas.	Teksto įrašymo laukai: 1. aprašymas; 2. veiksmai, kas galėtų įvykti; 3. Problemos sprendimo būdas. Laukai su klasifikatoriais: 1. sąsaja su produkto dalimi (reikalavimų ir PA sąrašas); 2. suinteresuotos šalys (suinteresuotų šalių pasirinkimo sąrašai).	Analitikas įveda informaciją	Skurto metodo forma atitinka <i>Volere</i> šablono turinį. Papildomai yra du laukai su pasirinkimais.
<b>19 Egzistuojantys sprendimai</b>	Sąrašo forma, paminint papildomą informaciją apie produktą kaip: 1. kaina; 2. prieinamumas; 3. laiko intervalas per kurį galima sukurti reikalingą integraciją.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. reikalingi tobulinimai; 4. užtruktų laiko pritaikyti projekte. Skaičių įrašymo laukai: 1. kaina.	Analitikas įveda informaciją	Skurto metodo forma atitinka <i>Volere</i> šablono turinį.

<b>20 Naujos problemos</b>	Laisva tekstine forma.	Teksto įrašymo laukai: 1. aprašymas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>21 Uždaviniai</b>				
<b>21a Sistemos kūrimo procesas</b>	Siūloma atvaizduoti aukšto lygio procesų diagrama. Taip pat pridėti aprašymą.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. fazės: 3.1. pavadinimas; 3.2. aprašymas.	Analitikas įveda informaciją	Sukurtame metode nėra atvaizduojama aukšto lygio procesų diagrama, bet ši informacija nurodoma tekstiniu pavidalu.
<b>21b Detalus kūrimo planas</b>	Galima braižyti LEAN grafiką ir pridėti aprašymą, kuris nurodo: 1. fazės pavadinimas; 2. naudotojo nauda; 3. atlikimo data; 4. naudojami komponentai, funkciniai ir nefunkciniai reikalavimai.	Teksto įrašymo laukai: 1. fazės pavadinimas; 2. Pagrindimas. Datos įrašymo laukai: 1. pradžios data; 2. pabaigos data. Laukai su klasifikatoriais: 1. susiję reikalavimai (reikalavimų sąrašas) Iš šios informacijos būtų galima generuoti diagramą.	Analitikas įveda informaciją + diagrama	Sukurto metodo forma atitinka Volere šablono turinį, bet diagramos generavimas nėra realizuotas.
<b>23 Rizikų įvertinimas</b>	Lentelės forma, nurodant: 1. pavadinimas; 2. aprašymas; 3. tikimybė, kad rizika įvyks; 4. poveikis ir jo paaiškinimas; 5. planas minimizuoti rizikos faktorių; 6. planas sumažinti rizikos poveikį.	Teksto įrašymo laukai: 1. pavadinimas; 2. aprašymas; 3. poveikis ir jo paaiškinimas; finansinis poveikio paaiškinimas; 4. planas minimizuoti rizikos faktorių; 5. planas sumažinti rizikos poveikį. Skaičiaus įrašymo laukai: 1. tikimybė, kad rizika įvyks; 2. finansinis poveikis.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.
<b>24 Kaštai</b>	Laisva forma.	Teksto įrašymo laukai: 1. skaičiavimo būdas. Skaičiaus įrašymo laukai: 1. kaštai.	Analitikas įveda informaciją	Sukurtame metode yra išskiriami du laukai, o ne vienas aprašymo laukas.
<b>27 Idėjos sprendimams</b>	Laisva forma.	Teksto įrašymo laukai: 1 aprašymas.	Analitikas įveda informaciją	Sukurto metodo forma atitinka Volere šablono turinį.

Užpildžius palyginimo lentelę ir ją išanalizavus, nustatyta:

1. aštuonios skirtingos duomenų gavimo būsenos:

- analitikas įveda informaciją;
- pakartotinai panaudojant VM žinias;
- analitikas įveda informaciją ir galima sugeneruoti diagramą;
- pakartotinai panaudojant VM žinias ir galima sugeneruoti diagramą;
- pakartotinai panaudojant VM žinias ir analitikui įvedus papildomą informaciją;
- generuojama informacija tekstiniu pavidalu – duomenys skirti lentelei užpildyti, duomenys yra gaunami iš jau užpildytų skyrių;
- generuojama diagrama;
- duomenys nepildomi metode.

*Volere* šablonas suskirstytas į 35 elementus, pagal šablono skyrių aprašymo rekomendacijas.

Kiekybinis elementų būsenų įvertinimas:

- analitikas įveda informaciją neautomatizuotu būdu – 57 % (20);
- pakartotinai panaudojant VM žinias – 11 % (4);
- analitikas įveda informaciją ir galima sugeneruoti diagramą – 3 % (1);
- pakartotinai panaudojant VM žinias ir galima sugeneruoti diagramą – 3 % (1);
- pakartotinai panaudojant VM žinias ir analitikui įvedus papildomą informaciją – 3 % (1);
- generuojama informacija tekstiniu pavidalu – 9 % (3);
- grafinių modelių generavimas – 11 % (4);
- duomenys nepildomi metode – 3 % (1).

2. siekiant sumažinti empiriškumo faktorių reikėtų, kad kuo daugiau elementų aprašomųjų laukų turėtų apribojimus įvedamoms reikšmėms arba pasirinkimus iš klasifikatoriaus. Iš 103 galimų elementų laukų:

- skaičiaus įvedimo laukai – 6 % (6);
- loginės sąlygos laukai – 1% (1);
- datos įvedimo laukai – 4 % (4);
- pasirinkimas iš klasifikatoriaus – 11 % (11).

3. siekiant sumažinti empiriškumo faktorių reikėtų turėti kuo mažiau elementų, kurie būtų aprašomi tik vienu ar dviem (išskiriant pavadinimą ir aprašymą) laisvo teksto laukais. Iš 35 elementų buvo 8 (23 %), kurių aprašymas susidėjo iš vieno ar dviejų tekstinių laukų.

### **5.5. Metodo sudaromos reikalavimų specifikacijos palyginimas su studentų parašytomis specifikacijomis**

Atlikus tiek reikalavimų specifikacijos sudarymo metode naudojamų formų palyginimą su *Volere* šablono turiniu (5.2 lentelė), tiek palyginimą tarp studentų parašytų specifikacijų ir *Volere* šablono turinio (5.1 lentelė), galima palyginti situaciją be metodo (studentų parašytos specifikacijos) ir situaciją naudojant metodą. Palyginimas aprašomas 5.3 lentelėje.

### 5.3 lentelė Palyginimas situacijos be metodo ir naudojant metodą

Palyginimo kriterijus	Be metodo	Naudojant metodą
Galima aprašyti visus Volere šablono skyrius	Taip	Visus, išskyrus <i>2e Personažai</i>
Galima pridėti papildomų laukų, kurie būtų susiję su projekto dalykine sritimi	Taip	Ne
Visiems skyriams yra pateikiamos struktūrizuotos pildymo formos	Ne	Taip
Galima įkelti sumodeliuotus diagramų atvaizdus	Taip, bet reikia analitikui sumodeliuoti	Taip, metodas, įgalina sugeneruoti diagramas iš veiklos modelyje saugomų dalykinės srities

Atlikus palyginimą, nustatyta, kad analitikui pačiam rašant reikalavimų specifikaciją, galima pridėti papildomų aprašomųjų laukų, nes visas formas analitikas pats turi pasidaryti pagal pateiktas šablono rekomendacijas, dėl to didelė dalis skyrių yra aprašomos tik vienu aprašymo lauku arba *Volere* šablono neatitinkančia forma. O taikant metodą, visos rekomenduojamos formos yra pateikiamos prie skyrių bei galima generuoti UML notacijos diagramas iš jau užpildytų duomenų, kas įgalina sumažinti reikalavimų specifikacijos sudarymo empiriškumo faktorių, sumažinti laiko sąnaudas ir žmogiškųjų klaidų faktorių.

### 5.6. Sprendimo taikymo rekomendacijos

Atlikus metode naudojamų *Volere* šablono skyrių aprašymui skirtų laukų analizę ir įvardinus kiekybinius metodo kriterijus, pateikiamos šios rekomendacijos:

1. metodo duomenų bazę papildžius naujomis klasėmis arba esamais atributais būtų galima pusiau automatizuotu būdu sudaryti naudotojo reikalavimų specifikaciją, padengiančią ~ 97 % *Volere* šablono v.18 turinio. Dabartinė metodo versija gali padengti ~ 86 % *Volere* šablono v.18 versiją, nes 11 % sudaro generuojamos diagramos, išskyrus UML panaudojimo atvejų diagramos, kurios nebuvo integruotos šioje metodo versijoje;
2. išplėtus ir papildžius aprašomuosius elementų laukus ar jiems suteikus apribojimus, galima būtų dar labiau sumažinti empiriškumo faktorių. Dabartinėje metodo versijoje 22 % visų elementų laukų yra ne teksto įvedimo laukai, tai reiškia, kad šie laukai bus užpildyti šabloniškai. Bet 23 % aprašomųjų elementų aprašomi tik vienu ar dviem tekstiniais laukais, tai reiškia, kad taip yra didinamas empiriškumo faktorius, nes šiuos elementus kiekvienas analitikas užpildys su tiek informacijos kiek jam atrodys pakankama.

## Išvados

1. palyginus keturis reikalavimų specifikacijos įrankius (*Modern Requirements4TFS*, *IBM Rational DOORS*, *Visure*, *ReQtest*), nustatyta, kad nei vienas iš jų neturi funkcionalumo, skirto generuoti su naudotojo reikalavimų specifikavimo procesu susijusias UML diagramas, todėl tikslinga sukurti naudotojo reikalavimų specifikacijos sudarymo metodą;
2. įvertinus ŽGS veiklos modelį, nustatyta, kad jį papildžius naujais elementais arba pridėjus naujų atributų esamiems, naudotojo reikalavimų specifikavimo procesą galima praturtinti pusiau automatizuotu naudotojo reikalavimų specifikacijos (įskaitant UML projektines diagramas) sudarymo funkcionalumu;
3. palyginus keturis reikalavimų specifikacijos šablonus (*Volere*, *IEEE 830-1998*, *Use Case Approach*, *Karl Wieger*), metodo kūrimui pasirinktas – *Volere* šablonas, todėl, kad kuriamojo metodo funkcionalumui (lyginant su kitais nagrinėtais šablonais) realizuoti leidžia naudoti plačiausią pasirinkimą struktūrizuotų specifikacijos skyrių aprašymo formų;
4. išanalizavus *Volere* šablono sudėtį ir sukūrus jo metamodelį, nustatytos trūkstamos vartotojo specifikavimo procesui būtinos klasės, kuriomis papildytas veiklos metamodelis, apibrėžiantis kuriamos sistemos funkcionalumui būtinas klases;
5. atlikus UML panaudojimo atvejų diagramos metamodelio analizę, nustatyta, kad lyginant su veiklos modeliu ir *Volere* šablono metamodeliu, trūksta tik išplečiančių (angl. *extend*) ir papildančių veiklų (angl. *include*) aprašančiųjų klasių, būtinų šios diagramos generavimui;
6. atlikus veiklos ir *Volere* šablono metamodelių analizę nustatyta, kad pusiau automatinis vartotojo reikalavimų specifikacijos sudarymo metodas naudoja elementus, kurie gali būti keturių būsenų: a) papildomai panaudojant duomenis iš veiklos modelio, b) analitikas įveda informaciją neautomatizuotu būdu, c) dalis informacijos panaudojama iš VM, kita dalis įvedama analitiko, d) grafinių modelių generavimas;
7. kuriamas vartotojo reikalavimų specifikacijos sudarymo metodas optimizuoja ir struktūrizuoja specifikacijos sudarymo procesą, taikant veiklos modelį kaip pradinį (pagrindinį) dalykinės srities žinių šaltinį;
8. analizės dalyje atlikto tyrimo pagrindu sukurta informacinė sistema, apimanti vartotojo reikalavimų specifikacijos sudarymo funkcionalumą, naudojant veiklos modelio žinių bazę kaip pagrindinį kompiuterizuojamos dalykinės srities žinių šaltinį;
9. IS realizacijos testavimas atliktas dviem aspektais: rezultatų teisingumo ir atitikimo *Volere* šablono sudėčiai. Atlikus testavimą nustatyta, kad sistemos funkcionalumas tenkina lūkesčius, todėl, kad metodo duomenų bazę papildžius naujomis klasėmis arba esamais atributais galima pusiau automatizuotu būdu sudaryti naudotojo reikalavimų specifikaciją, padengiančią ~97 % *Volere* šablono v.18 turinio. Dabartinė metodo versija gali padengti ~ 86 % *Volere* šablono v.18 versiją.



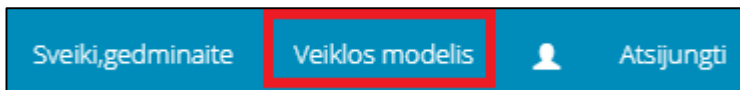
## Literatūros sąrašas

- [1] „Guru99,“ [Tinkle]. Available: <https://www.guru99.com/requirement-management-tools.html>. [Kreiptasi 9 Gruodžio 2018].
- [2] „Modern Requirements - Products - Modern Requirements4TFS,“ [Tinkle]. Available: <https://www.modernrequirements.com/modern-requirements-suite4tfs-3/>. [Kreiptasi 14 gruodžio 2018].
- [3] „IBM - Overview of Rational DOORS,“ [Tinkle]. Available: [https://www.ibm.com/support/knowledgecenter/SSYQBZ\\_9.6.0/com.ibm.doors.requirements.doc/topics/c\\_welcome.html](https://www.ibm.com/support/knowledgecenter/SSYQBZ_9.6.0/com.ibm.doors.requirements.doc/topics/c_welcome.html). [Kreiptasi 10 Gruodžio 2018].
- [4] „Visures Solutions,“ [Tinkle]. Available: <https://visuresolutions.com>. [Kreiptasi 10 Gruodžio 2018].
- [5] „ReQtest,“ [Tinkle]. Available: <https://reqtest.com>. [Kreiptasi 10 Gruodžio 2018].
- [6] S. Gudas, A. Lopata ir T. Skersys, „Approach to Enterprise modelling of Information Systems engineering,“ *INFORMATICA*, 2005.
- [7] G. Schreiber, B. Wielinga ir J. Breuker, *KADS. A Principled Approach to Knowled-Based System Development*, 1993.
- [8] S. Gudas ir A. Lopata, *Žiniomis grindžiama sistemų inžinerija*, 2011.
- [9] M. Ambraziūnas, „Veiklos Žinių Baze Išplėstos Modeliais Grindžiamos Architektūros Taikymo Informacijos Sistemų Inžinerijoje Metodas,“ Vilnius, 2014.
- [10] A. Lopata, „Enterprise Model Based Computerized Specification Method of User Functional Requirements,“ Kaunas, 2004.
- [11] S. Gudas, *Enterprise Knowledge Modelling: Domains and Aspects*, 2009.
- [12] G. Weichhart, C. Stary ir F. Vernadat, „Enterprise modelling for interoperable and knowledge-based enterprises,“ *International Journal of Production Research*, pp. 2818-2840, 2018.
- [13] D. J. Dörr, „Requirements Specifications & Standards,“ 2016/2017.
- [14] J. Robertson ir S. Robertson, *Volere Requirements Specification Template*, 2007.
- [15] IEEE Computer Society, „IEEE Recommended Practice for Software Requirements Specifications,“ 1998.
- [16] D. Leffingwell ir D. Widrig, *Managing Software Requirements; A Use Case Approach*, 1999.
- [17] K. Wiegers ir J. Beatty, *Software Requirements Third Edition*, 2013.
- [18] C. EL Hog, R. B. Djemaa ir I. Amous, „AWS-WSDL: a WSDL extension to support adaptive web service,“ *ACM International Conference Proceeding Series*, pp. 477-480, 2011.
- [19] A. Batool, Y. Motla, B. Hamid, S. Asghar, M. Mukhtar ir M. Ahmed, „Comparative study of traditional requirement engineering and Agile requirement engineering,“ *International Conference on Advanced Communication Technology, ICACT*, pp. 1006-1014, 2013.

## Priedai

### 1 priedas. Naudotojo instrukcija




Prisijungus naudotojui, kuriam yra priskirta *Sistemos Užsakovo* rolė, galima užpildyti veiklos modelio žinių bazę. Pirmas žingsnis – pasirinkti meniu punktą *Veiklos modelis* (1 pav.).



1 pav. Naudotojo vadovas. *Veiklos modelio* meniu punkto pasirinkimas

Tai įvykdžius atidaromas naudotojui priskirtų projektų sąrašas (2 pav.). Paspaudus mėlyną peržiūros piktogramą bus atidaromas veiklos modelio peržiūros langas (3 pav., dėl aiškumo lango paveikslas suskaidytas). Šiame lange ne tik galima peržiūrėti jau užpildytą veiklos modelio informaciją, bet ir pridėti naujų projekto elementų, juos koreguoti bei naikinti.

Projektų sąrašas

Pavadinimas	Aprašymas	
Knygų sandėlio IS	Knygų spaustuvės sandėlio ir pardavimo informacinė sistema	
Ligoninės IS	Ligoninės informacinės sistema, kurioje yra ambulatorijos, stacionaro, diagnostikos, operacijų, SPS, ataskaitų moduliai	
Kepyklos IS	Kepyklos IS leis atlikti turimų ingredientų apskaitą, registruoti užsakymus ir atlikti apskaitą	

2 pav. Naudotojo vadovas. Projektų sąrašo langas

Projekto specifikacija Ligoninės IS





Pavadinimas

Projekto aprašymas

Kūrimo priežastys









## Projekto tikslai

+ Tikslas

Pavadinimas	Aprašymas	Išmatavimas	
Supaprastinti informacijos vedimą	Norima sutaupyti laiką rašant paciento informaciją raštu	Pildymas tampa bent 10 proc. greitesnis sistemoje, nei pildant ranka	 
Įvesti informacijos pildymo šabloną	Norima utikrinti visos reikalingos informacijos įrašymą	Išrašuose yra visa reikalinga informacija	 





## Projekto funkcijų vykdytojai (aktoriai)

+ Funkcijos vykdytojas

Pavadinimas	Amžiaus riba	IT raštingumas	Dalyvavimas projekte	Pagrindimas	Prioritetas	
Administratorius	23-64	Vidutinis	Administratoriaus funkcijos	Kurs gydytojų kabinetų grafikus, kabinetų talonus	Nesvarbus naudotojas	 
IT administratorius	23-64	Aukštas	Visa sistema	Administruos naudotojus, kabinetus ir skyrius	Nesvarbus naudotojas	 
Gydytojas	25-64	Vidutinis	Gydytojų posistemė	Pildys paciento informaciją, tvirtins rezidentų įrašus	Pagrindinis naudotojas	 
Naudotojas	23-60	Žemas	Bendros funkcijos	Prisijungimas prie sistemos, savo nustatymų koregavimas	Nesvarbus naudotojas	 

## Projekto tikslai

+ Tikslas

Pavadinimas	Aprašymas	Išmatavimas	
Supaprastinti informacijos vedimą	Norima sutaupyti laiką rašant paciento informaciją raštu	Pildymas tampa bent 10 proc. greitesnis sistemoje, nei pildant ranka	 
Įvesti informacijos pildymo šabloną	Norima utikrinti visos reikalingos informacijos įrašymą	Išrašuose yra visa reikalinga informacija	 





## Procesų vykdytojai

+ Procesų vykdytojai

Pavadinimas	
Gydytojas	 
Slaugytojas	 
Rezidentas	 





Projekto procesai

+ Procesas

Pavadinimas	Aprašymas	
Aprašyti ligos istoriją	Aprašyti paciento anamnezę, nusiskundimus, pirminę apžiūrą	 
Užpildyti popierinį siuntimą tyrimui	Užpildyti popierinę siuntimo formą	 

Sistemos funkcijos

+ Sistemos funkcija

Pavadinimas	Aprašymas	
Užpildyti anamnezę	Gdytojas užpildo anamnezę: nusiskundimus, gyvenimo ir ligos pagrindinę informaciją	 
Užsakyti diagnostinius tyrimus	Gdytojas užsako diagnostinius tyrimus, jei apžiūra indikuoja, kad reikia juos atlikti	 

### 3 pav. Naudotojo vadovas. Veiklos modelio informacijos peržiūros langas

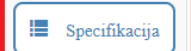


Prisijungus naudotojui, kuriam yra priskirta Analitiko rolė, galima užpildyti likusią specifikacijos informaciją pagal *Volere* šabloną. Pirmas žingsnis – pasirinkti meniu punktą *Mano projektai* (4 pav.):



### 4 pav. Naudotojo vadovas. *Mano projektai* meniu punkto pasirinkimas

Atsidariusiame lange (5 pav.) atvaizduojami projektai, kurių informaciją naudotojas gali koreguoti. Paspaudus mygtuką *Specifikacija* bus atidaromas langas (6 pav.) su pagrindine projekto informacija bei specifikacijos aspekto pasirinkimu.

Mano Projektų Sąrašas

Pavadinimas	Aprašymas	
Knygų sandėlio IS	Knygų spaustuvės sandėlio ir pardavimo informacinė sistema	
Ligoninės IS	Ligoninės informacinės sistema, kurioje yra ambulatorijos, stacionaro, diagnostikos, operacijų, SPS, ataskaitų moduliai	
Kepyklos IS	Kepyklos IS leis atlikti turimų ingredientų apskaitą, registruoti užsakymus ir atlikti apskaitą	

### 5 pav. Naudotojo vadovas. Analitikui priskirti projektai

Pavadinimas

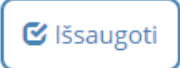
Ligoninės IS

Projekto aprašymas

Ligoninės informacinės sistema, kurioje yra ambulatorijos, stacionaro, diagnostikos, operacijų, SPS, ataska

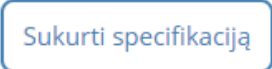
Kūrimo priežastys

Su sistema tikimasi pagreitinti žmonių darbą, išvengti žmogiškųjų klaidų



Specifikacijos aspektas:

Funckijos vykdytojų sąrašas:




6 pav. Naudotojo vadovas. Specifikacijos tipo pasirinkimo langas




Pasirinkus specifikacijos aspektą atidaromas specifikacijos koregavimo langas (7 pav., dėl aiškumo lango paveikslas suskaidytas), kuriame esanti informacija yra atvaizduojama iš veiklos modelio. Šiame lange pildomi laukai priklauso nuo pasirinkto specifikacijos aspekto. Lange galima ne tik pridėti naujus projekto elementus, bet ir pakoreguoti jau užpildytus.

Projekto specifikacija Ligoninės IS







Projekto funkcijų vykdytojai (aktoriai)



Pavadinimas	Amžiaus riba	IT raštingumas	Dalyvavimas projekte	Pagrindimas	Prioritetas	
Gydytojas	25-64	Vidutinis	Gydytojų posistemė	Pildys paciento informaciją, tvirtins rezidentų įrašus	Pagrindinis naudotojas	 
Rezidentas	23-30	Vidutinis	Gydytojų posistemė	Pildys pacientų informaciją	Pagrindinis naudotojas	 





Pasirinkto funkcijos vykdytojo veiklos kontekstai

[+ Veiklos kontekstas](#)

Pavadinimas	Aprašymas	Siuntėjas	Gavėjas	
Įvedami paciento duomenys	Išsaugojami paciento ligos istorijos duomenys	Pacientas	Veikla	 
Įvedama paciento diagnozė	Gydytojas įveda paciento diagnozę	Gydytojas	Veikla	 

Pasirinkto funkcijos vykdytojo atliekamos sistemos funkcijos

[+ Sistemos funkcija](#)

Pavadinimas	Aprašymas	
Užpildyti anamnezę	Gydytojas užpildo anamnezę: nusiskundimus, gyvenimo ir ligos pagrindinę informaciją	 
Užsakyti diagnostinius tyrimus	Gydytojas užsako diagnostinius tyrimus, jei apžiūra indikuoja, kad reikia juos atlikti	 

[Specifikacija](#)


### 7 pav. Naudotojo vadovas. Reikalavimų specifikacijos pildymo langas

Papildomą informaciją apie projekto elementą galima peržiūrėti paspaudus redagavimo mygtuką. Tokiu atveju atidaromas langas su išplėstine informacija. Pavyzdyje atvaizduojama *Funkcijos* papildoma informacija (8 pav., dėl aiškumo lango paveikslas suskaidytas).

Funckijos informacijos peržiūra

Pavadinimas:

[+ Tikslas](#)

Tikslas	Pagrindimas	Pamatavimas	
<input type="text" value="Įvesti paciento inforamciją"/>	<input type="text" value="Reikalinga pildyti ligos istoriją kuo tiks"/>	<input type="text" value="Anamnezė yra upildyta"/>	

Aprašymas:

Prieš-sąlyga:

**+ Aktorius**

<b>Aktorius</b>	
Rezidentas	
Gdytojas	

**+ Sužadavimo sąlyga**

<b>Sužadavimo sąlyga</b>	<b>Loginė sąlyga</b>	
Naudotojas nori įvesti paciento informaciją	Naudotojas turi būti atsidaręs paciento kontekstą	

Susiję panaudojimo atvejai

**+ PA**

<b>Apimami PA</b>	
Užsakyti diagnostinius tyrimus	

**Veiklos scenarijai**

**+ Scenarijus**

<b>Pavadinimas</b>	Duomenys teisingai suvesti ir pasirinkta pildyti naują anamnezę		
<b>Naudotojo veiksmai</b>	<b>Sistemos veiksmai</b>		
1	Atsidaryti anamnezės formą	2	Atidaryti formos pildymo pasirinkimą
3	Pasirinkti pildyti naują anamnezę	4	Atidaryti anamnezės formą
5	Užpildyti anamnezę		
6	Patvirtinti anamnezę	7	Išsaugoti anamnezės duomenis
		8	Atidaryti anamnezės peržiūros formą
<b>Sąlyga po</b>	Anamnezė išsaugojama		

**Susiję reikalavimai**

**+ Reikalavimas**

<b>Reikalavimas</b>	
Paslaugos įvedimas	

8 pav. Naudotojo vadovas. Funkcijos informacijos peržiūra.

Paspaudus reikalavimo informacijos peržiūros piktogramą, galima matyti visą reikalavimo aprašymo informaciją, pagal 1.3 lentelę (9 pav).

Nefunkcinio reikalavimo informacijos peržiūra

7

Tipas:

Pavadinimas:

Aprašymas:

Pagrindimas:

Atlikimo kriterijus:

Naudotojo tenkinimas:

Naudotojo netenkinimas:

Prioritetas:

Šaltiniai:

9 pav. Naudotojo vadovas. Reikalavimo informacijos peržiūra.

Užpildytą specifikaciją galima atsispausdinti paspaudus *Specifikacija* mygtuką. Taip bus sukurtas *Word* tipo failas, kuris aprašomas 4 priede.

## 2 priedas. Sudaryta specifikacija

# Ligoninės IS specifikacija pagal Volere šablona

Atspausdino Dovilė Gedminaitė

Specifikacijos versija: 01

Atspausdinimo data: 4/5/2020



## 1. Sistemos paskirtis

### 1a. Analizuojama veikla ir/ar projekto priežastys

Projekto pavadinimas: Ligoninės IS

Projekto aprašymas: Ligoninės informacinės sistema, kurioje yra ambulatorijos, stacionaro, diagnostikos, operacijų, SPS, ataskaitų moduliai

Projekto kūrimo priežastys: Su sistema tikimasi pagreitinti žmonių darbą, išvengti žmogiškųjų klaidų

### 1b Tikslai

Kuriamos sistemos tikslai:

Tikslas	Pagrindimas	Pamatavimas
Supaprastinti informacijos vedimą	Norima sutaupyti laiką rašant paciento informaciją raštu	Pildymas tampa bent 10 proc. greitesnis sistemoje, nei pildant ranka
Įvesti informacijos pildymo šabloną	Norima užtikrinti visos reikalingos informacijos įrašymą	Išrašuose yra visa reikalinga informacija

## 2 Suinteresuoti asmenys

### 2a Užsakovas

Užsakovo pavadinimas: LSMUL KK

Aprašymas: Tiek užsakovas, tiek pirkėjas yra tas pati organizacija. Pirkėją atstovauja įstaigos administracija, buhalterijos skyrius, IT tarnyba

Dalyvavimas projekte: Pirminės specifikacijos sudarymas, sistemos testavimas.

### 2b Pirkėjas

Pirkėjo pavadinimas: LSMUL KK

Aprašymas: Tiek užsakovas, tiek pirkėjas yra tas pati organizacija. Pirkėją atstovauja įstaigos administracija, buhalterijos skyrius, IT tarnyba

Dalyvavimas projekte: Pirminės specifikacijos sudarymas, sistemos testavimas.

### 2c Kiti suinteresuoti asmenys

Kiti suinteresuoti asmenys, kurie dalyvauja sistemos kūrime:

Suinteresuotų asmenų vaidmuo	Aprašymas	Pagrindimas	Dalyvaujančių projekte kiekis	Kur dirba darbuotojas ?	Dalyvavimas projekte	Turimos žinios
IT specialistas	Ligoninės IT specialistas, kuris ves mokymus	Darbuotojas, turi suvokti visą sistemą, kad galėtų praveisti mokymus	3	Ligoninėje	Visas projektas	Ligoninėje vykstantys procesai
Projekto vadovas	Projektų vadovas, vadovauja projektų sričių vadovams	Reikalingas asmuo, kad planuotų projekto darbus, sektų jų atlikimą.	1	Įmonėje, kuri atlieka sistemos kūrimą	Visas projektas	Tikslai, darbų tvarkaraštis

### 2d. Potencialūs produkto naudotojai

Galimi sistemos naudotojai:

Pavadinimas	Pagrindimas	Dalyvavimas projekte	Amžiaus ribos	IT raštingumas	Prioritetas
Gydytojas	Pildys paciento informaciją,	Gydytojų posistemė	25-64	Vidutinis	Pagrindinis naudotojas

	tvirtins rezidentų įrašus				
Slaugytoja	Pildys pacientų informaciją, užsakinės tyrimus	Slaugytojų posistemė	23-64	Vidutinis	Pagrindinis naudotojas
Vyr. slaugytoja	Formuos ataskaitas	Slaugytojų posistemė	25-64	Vidutinis	Antraeilis naudotojas

## 2f Naudotojams priskirti prioritetai

Galimi sistemos naudotojai, suskirstyti pagal jiems duotą prioritetą:

Prioritetas	Naudotojai
1. Pagrindiniai naudotojai	Gydytojas; Rezidentas; Slaugytoja;
2. Antraeiliai naudotojai	Vyr. slaugytoja;
3. Nesvarbūs naudotojai	Administratorius; IT administratorius; Naudotojas;

## 2g Naudotojų dalyvavimas projekte

Pavadinimas	Dalyvavimas projekte
Gydytojas	Gydytojų posistemė
Naudotojas	Bendros funkcijos
Rezidentas	Gydytojų posistemė
Slaugytoja	Slaugytojų posistemė
Vyr. slaugytoja	Slaugytojų posistemė

## 2h Sistemos aptarnaujantis personalas

Sistemos aptarnaujantis personalas:

Suinteresuotų asmenų vaidmuo	Dalyvaujančių projekte kiekis	Kur dirba darbuotojas?	Dalyvavimas projekte
IT specialistas	3	Ligoninėje	Visas projektas
Administratorius	3	Ligoninėje	Visas projektas

## 3 Apribojimai

### 3a Apribojimai sprendimui

Apribojimas Sistema privalo būti pasiekiamas per interneto naršyklę

Reikalavimas #:	1	Reikalavimo tipas:	3a	PA/FR#:	Visa sistema
Aprašymas:	Norint suteikti lankstumo ir pasiekiamumo sistemai, ji privalo būti pasiekiamas per interneto naršyklę				
Pagrindimas:	Užtikrinti lengvą sistemos pasiekimą				
Atlikimo kriterijus:	Sistemą galima pasiekti per vieną iš šių naršyklių: Google Chrome, Mozilla FireFox, Edge				
Naudotojo tenkinimas:	4	Naudotojo netenkinimas:	5		
Šaltiniai:					
Paskutinio atnaujinimo data:	2/18/2020				

### Apribojimas Sistema privalo galėti turėti integraciją su ESPBI IS

Reikalavimas #:	2	Reikalavimo tipas:	3a	PA/FR#:	Visa sistema
Aprašymas:	Vidinė ligoninės IS turi turėti sąsają su Lietuvos nacionaline sveikatos sistema				
Pagrindimas:	Užtikrinti paciento duomenų pasiekiamumą visoje Lietuvoje				
Atlikimo kriterijus:	Visi svarbūs paciento duomenys yra pasiekiami iš bet kurios ligoninės				
Naudotojo tenkinimas:	5	Naudotojo netenkinimas:	5		
Šaltiniai:	ESPBI IS dokumentacija				
Paskutinio atnaujinimo data:	2/18/2020				

### 3b Diegimo aplinka

#### 3c Komunikuojančios sistemos

##### Komunikuojanti sistema LabData

Pavadinimas:	Aprašymas	
LabData	Laboratorijos sistema	
1.	Genetinių tyrimų modulis	Atskiras aprašomasis genetinių modulis

##### Komunikuojanti sistema ESPBI IS

Pavadinimas:	Aprašymas	
ESPBI IS	Nacionalinė Lietuvos sveikatos sistema	
1.	NGN IS	Nėščiujų ir gimdyvių registravimo sistema

### 3d Prieinama specializuota programinė įranga (PI)

Pavadinimas	Aprašymas
Active Directory	Naudoti AD naudotojų sukūrimui

### 3e Numatoma darbo vietos aplinka

<b>Aprašymas</b>
Triukšminga darbo vieta
Nėra daug spausdintuvų
Didelės erdvės, nevisos vietos turi stalus

### 3f Sistemos kūrimo terminas

Data	Paiškinimas	Pasekmės, neįvykdžius numatytų darbų
2020-09-01	Reikalinga naują sistemą iki naujų mokslo metų, prieš ateinant naujai studentų ir rezidentų kartai	Reikės naujus žmones išmokinti kaip dirbti nauja sistema, kai jie jau bus padirbę su sena sistema

### 3g Sistemos kūrimo biudžeto apribojimai

Biudžetas: 45000

Paiškinimas: Galimos lėšos. Bet galima bandyti gauti papildomą finansavimą

### 3 h Organizacijos/politiniai apribojimai

#### 4. Sąvokos ir santrumpos

Sąvoka	Paaškinimas
ESPBI IS	Lietuvos nacionalinė sveikatos sistema
Anamnezė	Paciento ligos istorija

#### 5 Svarbūs faktai ir prielaidos

##### 5a Faktai

Faktas	Aprašymas
Rezidentai gali dirbti už gydytoją	Rezidentai gali rašyti paciento įrašus vietoj gydytojo

##### 5b Veiklos taisyklės

Taisyklė	Aprašymas	Pagrindimas
8-8 para	Para skaičiuojama nuo 8 ryto iki 8 ryto	Kadangi darbuotoju pamaina keičiasi 8 ryto.
SPS amb vizitas	Skubios pagalbos skyriuje vizitai formuojami kaip ambulatoriniai, nors pacientai yra guldomi	Kadangi SPS vizitai yra vienos paros.

##### 5c Prielaidos

Prielaida	Aprašymas	Pasekmės
Pacientų registracijos	Yra numatoma koreguoti nacionalinę pacientų išankstinę registraciją	Nepakoreguos integracijos su išankstiniu registracijų sistema, nebebus gaunamos registracijos i IS

#### 6 Veiklos sudėtis

##### 6a Esama padėtis

##### 6b Veiklos kontekstas

##### 6c Veiklos suskaidymas (įvykių specifikacija)

Pavadinimas	Aprašymas	Signalą siunčia	Signalą gauna
Įvedami paciento duomenys	Išsaugojami paciento ligos istorijos duomenys	Pacientas	Veikla
Įvedama paciento diagnozė	Gydytojas įveda paciento diagnozę	Gydytojas	Veikla

#### 7 Duomenų modelis ir jo elementų žodynas

##### 7a Duomenų modelis

##### 7b Duomenų žodynas

Esybė Darbuotojas

Esybė				
Darbuotojas	Aprašomi visi darbuotojai, kurie gali naudotis sistema, tiek visas med. personalas, tiek administracinis			
Atributai				
Nr.	Pavadinimas	Aprašymas	Galimos reikšmės apribojimai	Ar reikšmė privaloma
1	Id	identifikatorius	susideda iš skaičių	Reikšmė privaloma
2	Darbuotojo_Kodas	Darbuotojo spaudo numeris	Spaudo numeris dažniausiai susideda iš skaičių, bet yra specialybių kurių spaudo nr	Reikšmė privaloma

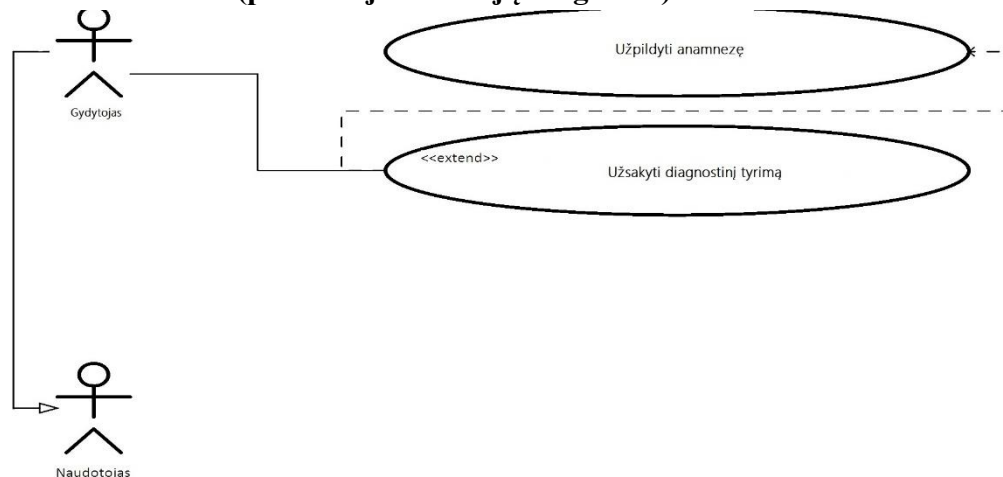
			turi ir raidžių junginius	
Ryšiai				
1	Žmogus_Id	Žmogaus identifikatoriaus priskyrimas darbuotojui	Skaičius	Reikšmė privaloma

### Esybė Žmogus

Esybė				
Žmogus	Aprašoma informacija apie žmogų			
Atributai				
Nr.	Pavadinimas	Aprašymas	Galimos reikšmės apribojimai	Ar reikšmė privaloma
1	Id	Identifikatorius	Susideda iš skaičių	Reikšmė privaloma
2	Vardas	Žmogaus pavardė ir vardas,	pavardė nuo vardo atskirti kableliu	Reikšmė privaloma

## 8 Sistemos sudėtis (panaudojimo atvejų modelis)

### 8a Sistemos ribos (panaudojimo atvejų diagrama)



### 8bc Panaudojimo atvejai

#### Panaudojimo atvejis Užpildyti anamnezę

PA	Užpildyti anamnezę		
Tikslas			
Pavadinimas: Įvesti paciento informaciją	Pagrindimas: Reikalinga pildyti ligos istoriją kuo tiksliau	Pamatavimas: Anamnezė yra užpildyta	
Aprašymas:	Gydytojas užpildo anamnezę: nusiskundimus, gyvenimo ir ligos pagrindinę informaciją		
Prieš sąlyga:	Naudotojas turi būti atsidaręs paciento kontekstą		
Aktorius:	Rezidentas; Gydytojas;		
Sužadinimo sąlyga:			
Pavadinimas	Naudotojas nori įvesti paciento informaciją		
Susiję panaudojimo atvejai	Išplečiantys PA		
	Apimami PA	Užsakyti diagnostinius tyrimus;	
	Apibendrinti PA		
Scenarijai			
Pagrindinis scenarijus	Duomenys teisingai suvesti ir pasirinkta pildyti naują anamnezę		
Naudotojo veiksmai	Sistemos veiksmai		

1 Atsidaryti anamnezės formą	2 Atidaryti formos pildymo pasirinkimą
3 Pasirinkti pildyti naują anamnezę	4 Atidaryti anamnezės formą
5 Užpildyti anamnezę	
6 Patvirtinti anamnezę	7 Išsaugoti anamnezės duomenis
	8 Atidaryti anamnezės peržiūros formą
Rezultatas	Anamnezė išsaugojama

#### Panaudojimo atvejis Užsakyti diagnostinius tyrimus

<b>PA</b>	<b>Užsakyti diagnostinius tyrimus</b>	
Tikslas		
Pavadinimas: Įvesti teisingai diagnostinio tyrimo siuntimą	Pagrindimas: Reikalinga kuo tiksliau įvesti diagnostinio tyrimo siuntimą	Pamatavimas: Teisingai užpildytas siuntimas
Aprašymas:	Gydytojas užsako diagnostinius tyrimus, jei apžiūra nustato, kad reikia juos atlikti	
Prieš sąlyga:	Naudotojas turi būti atsidaręs paciento anamnezę	
Aktorius:	Rezidentas; Gydytojas;	
Sužadavimo sąlyga:		
Pavadinimas	Naudotojas nori įvesti diagnostinio tyrimo siuntimą	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Apibendrinti PA	
Scenarijai		

### 9 Papildomi funkciniai reikalavimai

#### 10 Reikalavimai sistemos išvaizdai

##### 10a Išvaizdos reikalavimai

##### 10b Stiliumo reikalavimai

#### 11 Reikalavimai panaudojamumui

##### 11a Naudojimosi paprastumas

Reikalavimas #:	7	Reikalavimo tipas:	11a	PA/FR#:	---
Aprašymas:	Naudotojas negali padaryti daugiau nei 5 žingsnių užsakant paslaugą				
Pagrindimas:	Užtikrinti greitą ir patogų paslaugų įvedimą				
Atlikimo kriterijus:	Naudotojas gali užregistruoti paslaugą neįvykdęs 6 žingsnių				
Naudotojo tenkinimas:	5	Naudotojo netenkinimas:	5		
Šaltiniai:					
Paskutinio atnaujinimo data:	2/28/2020				

##### 11b Personalizavimo ir kalbos konfigūravimo reikalavimai

##### 11c Mokymosi reikalavimai

##### 11d Suprantamumo ir mandagumo reikalavimai

##### 11e Prieinamumas neįgaliesiems

##### 11f Suteikiami patogumai R

#### 12 Efektyvumo ir našumo reikalavimai

##### 12a Reikalavimai užduočių vykdymo greičiui

##### 12b Darbo ir aplinkos saugos reikalavimai

##### 12c Reikalavimai tikslumui

- 12d Patikimumas ir pasiekiamumas
- 12e Atsparumas trukdžiams ir klaidoms
- 12f Reikalavimai apdorojamų duomenų apimtims
- 12g Reikalavimai išplečiamumui
- 12h Reikalavimai produkto ilgaamžiškumui
- 13 Reikalavimai veikimo sąlygoms
- 13a Numatoma fizinė aplinka
- 13b Aplinkosauginiai reikalavimai
- 13c Reikalavimai darbui su gretimomis sistemomis
- 13d Reikalavimai sist. Platinimo/gamybos formatui
- 13e Reikalavimai leidybos procesui
- 13f Reikalavimai atvirkštiniam suderinamumui
- 14 Reikalavimai sistemos priežiūrai
- 14a Sistemos aptarnavimas
- 14b Sistemos palaikymas
- 14c Pritaikymo platformoms reikalavimai
- 15 Reikalavimai saugumui
- 15a Prieigos reikalavimai (teisės)
- 15b Vientisumo (integralumo) reikalavimai
- 15c Reikalavimai privatumui
- 15d Audito reikalavimai
- 15e Reikalavimai savisaugai nuo išorinių grėsmių
- 16 Kultūriniai reikalavimai
- 17 Atitikties reikalavimai
- 17a Atitikties teisės aktams reikalavimai
- 17b Atitikties standartams reikalavimai
- 18 Atviros problemos ir klausimai

Numeris	Sąsaja su produkto dalimi	Aprašymas	Veiksmai, kas galėtų įvykti	Problemos sprendimo būdas
1	2;	Gali pasikeisti ESPBI IS struktūra	<ol style="list-style-type: none"> <li>1. Gali reikėti persiusti daugiau informacijos;</li> <li>2. Gali pasikeisti informacijos laukai;</li> <li>3. Gali pasikeisti vizitu struktūra</li> </ol>	<ol style="list-style-type: none"> <li>1. Sudaryti kuo paprastesnę formą perduoti duomenis iš Ligoninės IS į ESPBI IS.</li> <li>2. Pasamdyti programuotojus kuo greičiau pakeisti persiuntimo formas;</li> <li>3. Naudoti kuo daugiau ESPBI IS langų, o ne persiuntinėti duomenis</li> </ol>

## 19 Egzistuojantys sprendimai

### 19a Prieinamos sistemos

Pavadinimas	Aprašymas	Reikalingi tobulinimai	Kaina	Užtruktų laiko pritaikyti projekte
AD	Direktoriijoje, kurioje yra saugojami visų	Sukurti integraciją tarp sistemos ir AD	0	20 val.

	darbuotojų duomenys			
--	---------------------	--	--	--

### 19b Prieinami komponentai

Pavadinimas	Aprašymas	Reikalingi tobulinimai	Kaina	Užtruktų laiko pritaikyti projekte
NGN IS	Galima panaudoti NGN IS kaip nėščių ir gimdyvių registravimo modulį	Sukurti integraciją tarp dviejų sistemų, perduodant abejomis kryptimis informaciją	0	120 val.

### 19c Kopijuotini sprendimai

Pavadinimas	Aprašymas	Reikalingi tobulinimai	Kaina	Užtruktų laiko pritaikyti projekte
SPS registravimo programa	Galima panaudoti tą pačią logiką kuriant Priėmimo modulį, kuris buvo panaudotas, kuriant SPS registrą	Atitinkamai pakoreguoti sprendimą, kad atitiktų sistemos logiką	0	500 val.

### 20 Naujos problemos

#### 20a Poveikis diegimo aplinkai

	Aprašymas
1.	Sistemoje bus naudojami el. parašai, kurie privalo veikti interneto naršyklėje

#### 20b Poveikis esamoms sistemoms

	Aprašymas
1.	Reikės atsisakyti dalies programų

#### 20c Probleminė naudotojų reakcija

	Aprašymas
1.	Naudotojai nenorės naudotis nauja sistema

#### 20d Apribojimai diegimo aplinkoje

	Aprašymas
1.	Sistemos naudojimas kompiuteriuose su senesnės versijos OS

#### 20e Kitos potencialios problemos

	Aprašymas
1.	Didelis personalo kiekis ir jo kaita

### 21 Uždaviniai

#### 21a Sistemos kūrimo procesas

Pavadinimas: Iteracinis modelis

Aprašymas: modelio veikimo principas yra klasikinio krioklio modelio iteracijų sujungimas. Kiekviena iteracija – yra veikianti projekto dalis. Dažniausiai tokio proceso pirmoji iteracija būna bazinis produktas, atsižvelgiant į pagrindinius reikalavimus. Kiekviena sekanti iteracija yra bazinio produkto plėtimas arba atskiro komponento sukūrimas.

	Fazė	
1.	Pagrindinių reikalavimų surinkimas	Surenkami visos sistemos reikalavimai
2.	Projektavimas	Suprojektuojamos visos sistemos funkcijos



3.	Programavimas	Suprogramuoja pirminė versija funkcijų
4.	Testuojamos funkcijos	Testuojama tiek sistemos dizainas, tiek pritaikymas darbovietėje, tiek funkcijų teisingumas
5.	Sistemos koregavimas	Koreguojami reikalavimai, funkcijos ,dizainas
6.	Mokymai	Apmokomi darbuotojai dirbti sistema

### 21b Detalus kūrimo planas

Plano fazė Naudotojo funkcijų posistemė

Pavadinimas	Pradžia	Pabaiga
Naudotojo funkcijų posistemė	2/1/2020	5/5/2020

Plano fazė Gydytojų posistemės funkcijos

Pavadinimas		Pradžia	Pabaiga		
Gydytojų posistemės funkcijos		2/1/2020	6/6/2020		
	Pavadinimas	Pagrindimas	Pradžia	Pabaiga	Susiję reikalavimai
1.	Surinkti reikalavimus funkcijoms	Išsiaiškinti visus funkcijų poreikius	2/5/2020	3/3/2020	
2.	Suprogramuoti LA funkcijas	Suprogramuoti pagrindines ligos atvejo funkcijas	3/3/2020	4/4/2020	1;2;
3.	Suprogramuoti diagnostikos modulį	Suprogramuoti diagnostinių siuntimų ir atsakymo rašymo modulį	3/3/2020	3/20/2020	2;1;

### 22 Migravimas į naują produktą

22a Migravimo į naują produktą reikalavimai

22b Duomenų transformavimo reikalavimai

### 23 Rizikų įvertinimas

Rizika Sistema neatitinka visų poreikių

<b>Pavadinimas</b>	Sistema neatitinka visų poreikių
<b>Aprašymas</b>	Sistemos reikalavimai neapima visų reikalingų funkcijų.
<b>Tikimybė, kad rizika įvyks</b>	60
<b>Poveikis</b>	7
<b>Poveikio paaiškinimas</b>	Nes jei nebus įgyvendinti visi poreikiai, sistema trukdys darbą
<b>Finansinis poveikis</b>	60000
<b>Finansinio poveikio paaiškinimas</b>	pinginė suma, kuria bus taisomos paliktos klaidos
<b>Planas minimizuoti rizikos faktorių</b>	
1.	Išsamūs reikalavimai
2.	Med. personalas dalyvauja testavimuose, naudojant realias situacijas

3.	Panašių sistemų analizė
4.	Dabartinių procesų analizė
<b>Planas sumažinti rizikos poveikį</b>	
1.	Naujų poreikių analizė, esamos sistemos kontekste
2.	Naujų funkcijų įdiegimas, naudojant papildomą personalą
3.	Kitų reikalavimų peržiūra

#### Rizika Sistema nebus patogiu naudotis

<b>Pavadinimas</b>	Sistema nebus patogiu naudotis
<b>Aprašymas</b>	Sistema bus nepatogi, sunku joje susigaudyti, paprasti veiksmai reikalaus daug mygtukų paspaudimų
<b>Tikimybė, kad rizika įvyks</b>	40
Poveikis	8
<b>Poveikio paaiškinimas</b>	Sistema prailgins darbus lyginant su popierinės dokumentacijos pildymu
<b>Finansinis poveikis</b>	60000
<b>Finansinio poveikio paaiškinimas</b>	pinginė suma, kuri būtų skirta sistemos dizaino ir procesų keitimui

#### Planas minimizuoti rizikos faktorių

1.	Panašių sistemų analizė
2.	Išsamūs panaudojamumo testavimai, kuriuose dalyvauja med. personalas

#### Planas sumažinti rizikos poveikį

1.	Dizaino keitimas naudojant papildomą personalą
----	--

## 24 Kaštai

	Kaina	Skaičiavimo būdas
1.	70000	Skaičiuojant funkcijas
2.	85000	Skaičiuojant laiką skirtą kiekvienai funkcijai

## 25 Naudotojo dokumentacija ir apmokymai

### 25a Reikalavimai naudotojų dokumentacijai

### 25b Reikalavimai naudotojų apmokymui

## 26 Perspektyviniai reikalavimai

## 27 Idėjos sprendimams

	Idėja
1.	Pridėti atskirą ataskaitų modulį
2.	Sisteminių klasifikatorių naudojimas dinaminuose dokumentuose