
**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Vaidotas Drungilas**

# UML PANAUDOJIMO ATVEJŲ MODELIO IŠGAVIMAS IŠ INTERNETINIŲ INFORMACINIŲ SISTEMŲ

Baigiamasis magistro projektas

**Vadovė**
doc. dr. L. Čeponienė

**KAUNAS, 2018**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

# UML PANAUDOJIMO ATVEJŲ MODELIO IŠGAVIMAS IŠ INTERNETINIŲ INFORMACINIŲ SISTEMŲ

Baigiamasis magistro projektas

**Informacinių sistemų inžinerijos studijų programa (kodas 621E15001)**

**Vadovė**

doc. dr. L. Čeponienė

2018-05-21

**Recenzentas**

lekt. dr. A. Šukys

2018-05-21

**Projektą atliko**

Vaidotas Drungilas

2018-05-21

**KAUNAS, 2018**

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
(Fakultetas)

(Studento vardas, pavardė)

Informacinių sistemų inžinerijos studijų programa, 621E15001
(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „UML panaudojimo atvejų modelio išgavimas iš internetinių informacinių sistemų“

**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

20 ___ m. _____ ____ d.
Kaunas

Patvirtinu, kad mano, **Vaidoto Drungilo**, baigiamasis projektas tema „UML PANAUDOJIMO ATVEJŲ MODELIO IŠGAVIMAS IŠ INTERNETINIŲ INFORMACINIŲ SISTEMŲ“ 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)* |

Kaunas 2018

Drungilas, Vaidotas. *UML panaudojimo atvejų modelio išgavimas iš internetinių informacinių sistemų*. Magistro baigiamasis projektas / vadovė doc. dr. Lina Čeponienė; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatikos inžinerija, technologijos mokslai

Reikšminiai žodžiai: *UML, atvirkštinė inžinerija, interneto informacinės sistemos, panaudojimo atvejų diagrama, veiklos diagrama.*

Kaunas, 2018. 95 p.

## SANTRAUKA

Dėl didelio internetinių informacinių sistemų kūrimo tempo, dažnu atveju šios informacinės sistemos yra prastai dokumentuojamos, taip apsunkinant jų palaikymą ir tobulinimą. Tokia pat situacija susidaro, kai veikiančioje informacinėje sistemoje įvykdomi pakeitimai ar informacinės sistemos funkcionalumas plečiamas, o sistemos projektas nėra papildomas. Dėl to informacinės sistemos dokumentacija greitai atitrūksta nuo esamo funkcionalumo ir tampa ne taip gerai panaudojama informacinės sistemos plėtojimo ir palaikymo metu.

Šiame darbe apžvelgiamos galimybės sistemos UML modelius išgauti atgalinės inžinerijos būdu, siekiant palengvinti sistemos dokumentavimą. Išanalizavus rinkoje esančius sprendimus ir mokslinius tyrimus atliktus šioje srityje, nebuvo rastas sprendimas, kuris leistų neturint sistemos programinio kodo sugeneruoti panaudojimo atvejų modelį. Todėl nuspręsta kurti naują metodą ir įrankį, kuris, nepriklausomai nuo internetinei sistemai naudotos kūrimo technologijos, leistų išgauti jos UML panaudojimo atvejų modelį.

Šiame darbe pateikiamas siūlomas metodas, leidžiantis iš užregistruotų sistemos naudotojo veiksmų ir informacijos, išgautos iš grafinės sąsajos, sudaryti panaudojimo atvejų modelį. Taip pat pateikiamas metodo veikimui pademonstruoti sukurto įrankio projektas ir jo realizacijos aprašymas.

Eksperimento metu, siekiant patikrinti metodo ir įrankio veikimo teisingumą, sugeneruoti keturi panaudojimo atvejų modeliai. Du panaudojimo atvejų modeliai įvertinti, juose gautas diagramas ir jų elementus palyginus su analitikų suprojektuotais modeliais. Kiti du modeliai pateikti ekspertų vertinimui, siekiant išsiaiškinti sugeneruotų panaudojimo atvejų modelių kokybę. Įvertinus eksperimentų rezultatus, nustatyta, kad darbe sukurtas metodas yra tinkamas panaudojimo atvejų modelio generavimui ir jo kokybė, ekspertų yra vertinama aukštais balais

Drungilas, Vaidotas. *Reverse Engineering of UML Use Case Model from Website Usage Records*: Master‘s thesis in Information Systems Engineering / supervisor assoc. prof. Lina Čeponienė. Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics Engineering, Technology Science

Keywords: *UML; Reverse engineering; Website; Use Case diagram; Activity diagram*.

Kaunas, 2018. 95 p.

## SUMMARY

As websites exists in highly competitive environment, most of website developers tend to sacrifice quality of website documentation to spend more resources on website functionality to get competitive edge. As websites are poorly documented it is harder to maintain or introduce new functionality. The same kind of situation happens, when websites are updated without updating the corresponding documentation. This way websites’ documentation loses its value and is not as useful while maintaining or updating.

In this thesis, capabilities of reverse engineering websites to extract UML models are analyzed, in order to simplify website documenting. After the analysis of tools and research in the area, no tool or method was found, capable of generating Use Case model without source code analysis. It was concluded that new method and tool should be created, that would be able to generate UML Use Case model not depending on platform or programing language.

In this thesis the method, that provides ability to generate use case model from use interface data and recorded usage on websites, was developed, accompanied by the method prototype implementation.

To test the correctness of the created method and prototype tool, four Use Case models were generated. Two of them were evaluated by comparing them to Use Case models created by analysts. The other two were presented to UML modeling experts, to evaluate generated models’ quality. Experiment results indicate that method and tools presented in this thesis are capable of creating Use Case models, and experts evaluated the quality of generated models with the high scores.

**TURINYS**

**LENTELIŲ SĄRAŠAS**

**PAVEIKSLŲ SĄRAŠAS**

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

Darbe naudojami specifiniai terminai ir santrumpos bei jų paaiškinimas.

UML (angl. *Unified Modeling Language*) – unifikuota modeliavimo kalba.
HTTP (angl. *HyperText Transfer Protocol*) – hyperteksto perdavimo protokolas.
IS – informacinė sistema.
IIS – internetinė informacinė sistema.
HTML – (angl. *Hyper text Markup Language*) – hiperteksto žymėjimo kalba.
CSS – (angl. *Cascading Style Sheets*) – pakopiniai stilių šablonai.
MDD – (angl. *Model Driven Development*) – Modeliais grindžiamas programinės įrangos kūrimas.
MDA – (angl. *Model Driven Architecture*) – Modeliais grindžiamas programinės įrangos architektūra.
WEB2UML – naujai kuriamas panaudojimo atvejų modelio išgavimo iš internetinių informacinių sistemų įrankis.
IIS – internetinė informacinė sistema
CASE – (angl. *Computer-Aided Software Engineering*) programinės įrangos inžinerija naudojantis programine įranga.

# ĮVADAS

Kuriant mažo biudžeto informacines sistemas ar neturint tinkamų specialistų komandoje, specifikavimo ir dokumentavimo etapui dažniausiai skiriama mažai dėmesio. Taip naujai sukurta programinė įranga, pabaigus jos kūrimą, lieka be aiškios dokumentacijos ir funkcionalumo aprašymo. Tokias sistemas toliau tobulinant ar palaikant kyla problemų dėl dokumentacijos stokos.

Tokia pat situacija susidaro, kai veikiančioje informacinėje sistemoje įvykdomi pakeitimai ar informacinės sistemos funkcionalumas plečiamas, o sistemos projektas nėra papildomas. Dėl to, informacinės sistemos dokumentacija greitai atitrūksta nuo esamo funkcionalumo ir tampa ne taip gerai panaudojama informacinės sistemos plėtojimo ir palaikymo metu. Stengiantis suderinti šį neatitikimą, iššvaistoma daug pastangų ir lėšų, kurios galėtų būtų skiriamos sistemos tobulinimui.

Siekiant palengvinti informacinių sistemų modeliavimą, pasiūlyta daug sprendimų veikiančių informacinių sistemų modeliams rengti. Tam kuriami metodai ir programinė įranga [1] [2] [3], kuri atgalinės inžinerijos būdu sukuria visą ar dalį informacinės sistemos projekto. Tačiau nei vienas iš analizuotų sprendimų nėra pilnai pritaikytas išgauti panaudojimo atvejų modelį atgalinės inžinerijos būdu, neturint galimybės statiškai analizuoti programinio kodo. Nagrinėti metodai generuoja įvairias UML diagramas iš veikiančių sistemų, bet nebuvo rasta metodo ar įrankio, gebančio generuoti UML panaudojimo atvejų diagramą ir po vieną veiklos diagramą kiekvienam sugeneruotam panaudojimo atvejui nesinaudojant statine kodo analize. Panaudojimo atvejų diagramos ir veiklos diagramų kiekvienam panaudojimo atvejui rinkinys sudarytų panaudojimo atvejų modelį, kurį ir siekiama sugeneruoti šiame darbe.

Šio **darbo tikslas** – palengvinti UML modelių išgavimą iš informacinių internetinių sistemų grafinės naudotojo sąsajos. Darbe siekiama išgauti UML panaudojimo atvejų modelį (kuris apima panaudojimo atvejų ir veiklos diagramas) iš užregistruotų grafinės sąsajos naudotojo veiksmų internetinėje informacinėje sistemoje. Sukurtas metodas ir įrankis taip pat palengvina internetinių informacinių sistemų testavimą, suteikia galimybę informacinių sistemų savininkams lengviau atrasti skirtumus tarp jų internetinės informacinės sistemos ir konkurentų internetinių informacinių sistemų, lyginant modelius tarpusavyje.

Darbo tikslui pasiekti iškelti tokie **uždaviniai**:

1. išanalizuoti UML diagramų kūrimą internetinėms informacinėms sistemoms;
2. išanalizuoti naudotojo veiksmų registravimo įrankius;
3. išanalizuoti atvirkštinės inžinerijos metodus, naudojamus UML diagramų išgavimui;
4. suprojektuoti ir realizuoti UML diagramų atvirkštinės inžinerijos metodą ir jį realizuojantį įrankį;
5. metodą ir įrankį patikrinti eksperimentiškai, įvertinant sugeneruojamų diagramų kokybę;
6. apibendrinti tyrimo rezultatus.

Įgyvendinus šiuos uždavinius, naujai sukurtas metodas ir įrankis leidžia automatizuotai išgauti UML panaudojimo atvejų modelius iš internetinių informacinių sistemų. Įrankis suteikia galimybę analitikams vykdyti liktinių sistemų analizę, palyginti sistemų funkcionalumą tarpusavyje, ar patikrinti sistemos atitikimą parengtai sistemos specifikacijai.

Darbe realizuotas sprendimas – tai automatizuotas įrankis, leidžiantis generuoti UML diagramas iš internetinių informacinių sistemų grafinės naudotojo sąsajos bei sistemoje užregistruotų jos naudotojų veiksmų. Įrankiui sukurti buvo aprašytas metodas, apimantis naudotojo veiksmų registravimą ir registruotų veiksmų rinkinio transformavimą į UML diagramas. Įrankis realizuotas kaip internetinės naršyklės papildinys, leidžiantis registruoti naudojimąsi sistema. Šio registravimo rezultatai, pasinaudojant metode aprašytais algoritmais, transformuojami į XMI formato failus, kurie

gali būti importuoti į UML modeliavimo CASE įrankį. Modeliavimo įrankyje gautas panaudojimo atvejų modelis gali būti analizuojamas, papildomas ar naudojamas kaip kitų modelių dalis.

Magistro darbas susideda iš aštuonių skyrių. Pirmajame skyriuje analizuojama UML modeliavimo kalba ir jos taikymas internetinėms informacinėms sistemoms modeliuoti. Šiame skyriuje taip pat išanalizuoti UML modeliavimo įrankiai, specializuoti įrankiai palaikantys UML diagramų išgavimą iš sistemų atgalinės inžinerijos metodu, bei metodai, skirti atgalinės inžinerijos būdu išgauti UML diagramas iš egzistuojančių sistemų. Antrajame skyriuje pateikiama UML panaudojimo atvejų modelio išgavimo iš internetinių informacinių sistemų metodika ir jai įvykdyti reikalingi algoritmai. Trečiajame skyriuje detalizuojami darbe keliami reikalavimai sukurtai sistemai „WEB2UML“ bei šios sistemos projektas. Ketvirtajame skyriuje pateikiama informacija apie „WEB2UML“ įrankio realizaciją ir jos testavimą. Šiame skyriuje taip pat pateikiama dokumentacija sistemos naudotojui. Penktasis skyrius skirtas aprašyti eksperimento eigai, nustatytoms sistemos naudojimo rekomendacijoms. Septintajame skyriuje pateikiamos darbo išvados.

# 1. PROBLEMINĖS SRITIES ANALIZĖ

## 1.1. Analizės tikslas

Analizė vykdoma siekiant geriau suprasti, kaip internetinių informacinių sistemų grafinė naudotojų sąsaja ir naudojimo registravimas internetinėmis informacinėmis sistemomis gali būti transformuotas į UML diagramas. Išsiaiškinti gerąsias praktikas naudojamas generuojant UML diagramas atgalinės inžinerijos metodu. Siekiama detaliai išnagrinėti įrankius gebančius generuoti UML diagramas ar kitaip apdoroti internetinių informacinių sistemų grafinę sąsają ir naudojimąsi jomis. Taip pat ištirti jau egzistuojančių metodų ir jiems pritaikytų įrankių, gebančių sugeneruoti UML diagramas ar modelius, privalumus ir trūkumus.

## 1.2. Tyrimo objektas, sritis ir problema

Šiame darbe atliekamo **tyrimo objektas** – internetinių informacinių sistemų modeliavimas UML kalba. Kadangi darbe nagrinėjama galimybė išgauti UML diagramas iš jau veikiančios sistemos grafinės sąsajos, darbo **tyrimo sritis** apima internetinių informacinių sistemų UML modelių atvirkštinę inžineriją.

Pagrindinė šiame tyrime nustatyta **problema**: UML modelių atvirkštinė inžinerija iš nedokumentuotų internetinių sistemų vyksta neefektyviai. Neturint automatizuotų sprendimų, sistemų analitikai, projektuotojai, kūrėjai turi nagrinėti egzistuojančią nedokumentuotą sistemą ir rankiniu būdu kurti diagramas, aprašančias šios sistemos veikimą. Automatizavimas palengvintų šį procesą ir supaprastintų analitikų darbą.

## 1.3. UML modelių išgavimo iš internetinių informacinių sistemų galimybių analizė

### 1.3.1. UML ir jos taikymas internetinėms informacinėms sistemoms modeliuoti

Šiuolaikinių mažos ar vidutinės apimties interneto informacinių sistemų kūrimas dažnai vykdomas mažomis komandomis ir trumpomis projekto tobulinimo iteracijomis, kurių metu orientuojamasi į funkcionuojantį produktą, todėl mažiau dėmesio skiriama sistemos projektui ir dokumentacijai. Dėl to kuriant ar perėmus palaikyti tokį mažiau dokumentuotą projektą, daug laiko sugaištama aiškinantis jo funkcines ir architektūrines savybes. Šią problemą galima spręsti, dviem būdais: įtraukti projekto dokumentaciją ir architektūrinį projektą į privalomus reikalavimus arba sukurti įrankį, kuris sudarytų sąlygas automatizuotam dokumentacijos išgavimui iš jau minėtų nedokumentuotų sistemų. Šis naujas automatizuotas dokumentacijos išgavimo metodas, būtų ypač naudingas projektams kuriamiems *Agile* [4] tipo metodais.

Alternatyvą internetinių sistemų kūrimui didesnį dėmesį skiriant jų funkcionalumui pateikia modeliais grindžiamas kūrimas (angl. *Model Driven Development*) [5]. Modelis – modeliais grindžiamame programinės įrangos kūrime yra informacija, išskirtinai atvaizduojanti tam tikrą esybę ar procesą, remiantis apibrėžtais kriterijais. Modeliais grindžiamame kūrime pagrindinis dėmesys skiriamas modelių kūrimui, tobulinimui ir analizei. Šio tipo kūrimas leidžia atsiriboti nuo konkrečios technologinės platformos ar programavimo kalbos, kuriant abstrakčius modelius. Šie išsamūs modeliai tolesniuose programinės įrangos kūrimo etapuose, tokiuose kaip programinės įrangos kūrimas ar testavimas, yra naudojami kaip pamatinis šaltinis, kuriuo remiantis tikrinama ar sistema atitinka reikalavimus. Dažnu atveju modeliai yra transformuojami iš modelio į programinį kodą ar jo šabloną. Dėl galimybės automatiškai transformuoti modelį į programinį kodą, modeliais grįstos informacinės sistemos gali būti lengvai pritaikomos daugelyje platformų ar technologinių sprendimų.

Išsamiai modeliais grindžiamas programinės įrangos kūrimas pristatomas „Object Management Group“ korporacijos modeliais grindžiamoje architektūroje MDA (angl. *Model Driven Architecture*) [6] [7]. Modeliais grindžiamoje architektūroje aprašomas standartizuotas požiūris į sistemų kūrimą, kurio esmė – pasinaudojant modeliais perkelti sistemų projektavimą į aukštesnį abstrakcijos lygmenį. Remiantis MDA, siekiama pagerinti sistemų planavimą, kūrimą ir kitus

sistemos gyvavimo ciklo procesus siekiant aukštesnės galutinio produkto kokybės. MDA sudaro standartų rinkinys, nusakantis modelių atvaizdavimą, apsikeitimą modeliais tarp skirtingų modeliavimo kalbų, modelių transformaciją ir dokumentacijos sukūrimą.

Reikalavimai informacinėms sistemoms gali būti deklaruojami įvairiomis formomis. Pradedant nuo neformalaus pokalbio, užrašų ant lipnių lapelių ar formalaus dokumento, reikalavimų modelio. Neformalios reikalavimų specifikacijos dažniausiai taikomos smulkesniems projektams, kitaip nei projektams, kurie naudoja formalią specifikaciją. Formaliąją specifikaciją galime skirstyti į du pogrupius: rašytiniai – dažniausiai struktūrizuoti dokumentai ir modeliai. Dažniausiai dideliuose projektuose šie du reikalavimų tipai naudojami kartu. Neteisingai apibrėžus reikalavimus sistemos kūrimo pradžioje, jų pakeitimas ar pridėjimas vėlesniuose kūrimo etapuose, gali pareikalauti daug resursų.

Reikalavimų dokumento pavyzdžiu galime imti IEEE standartą [8] , kuris skirtas programinės įrangos reikalavimų inžinerijos procesams aprašyti. Jame galima rasti pavyzdinį dokumento sukūrimo šabloną, nors standarte ir pabrėžiama, kad viena struktūra nėra pritaikoma visiems projektams.

Informacinių sistemų reikalavimų specifikavimo metu specifikuojami du reikalavimų tipai. Funkciniai reikalavimai yra susiję su sistemos vykdomomis funkcijomis. Šie reikalavimai dažnu atveju atvaizduojami panaudojimo atvejų, veiklos ar klasių diagramomis. Nefunkciniai reikalavimai skirti aprašyti reikalavimus nesusijusius su sistemos funkcionalumu. Jiems atvaizduoti dažniausiai naudojamos klasių diagramos ar reikalavimų lentelės. Reikalavimus aprašant modeliais dažniausiai naudojamos UML elgsenos diagramos. Jos puikiai papildo surinktą tekstinę ir vizualine informacija.

Norint, kad MDA būtų panaudota sėkmingai modelių sudarymui reikia skirti daug laiko ir pastangų. Sudarant teisingus ir tinkamus modelius būtina aiškiai ir išsamiai išnagrinėti dalykinę sritį. Atsižvelgiant į internetinių informacinių sistemų kūrimo specifiką ir evoliucinę produkto raidą dažnai neįmanoma surinkti išsamių, pilnų reikalavimų. Tačiau modeliai vis vien yra puikiai pritaikomi internetinių informacinių sistemų kūrime, tik turi būti nuosekliai atnaujinami. Nuolatiniam modelių atnaujinimui dažnai nėra skiriama pakankamai dėmesio, todėl automatizuoti modelių kūrimo įrankiai yra reikalingi šiai problemai spręsti.

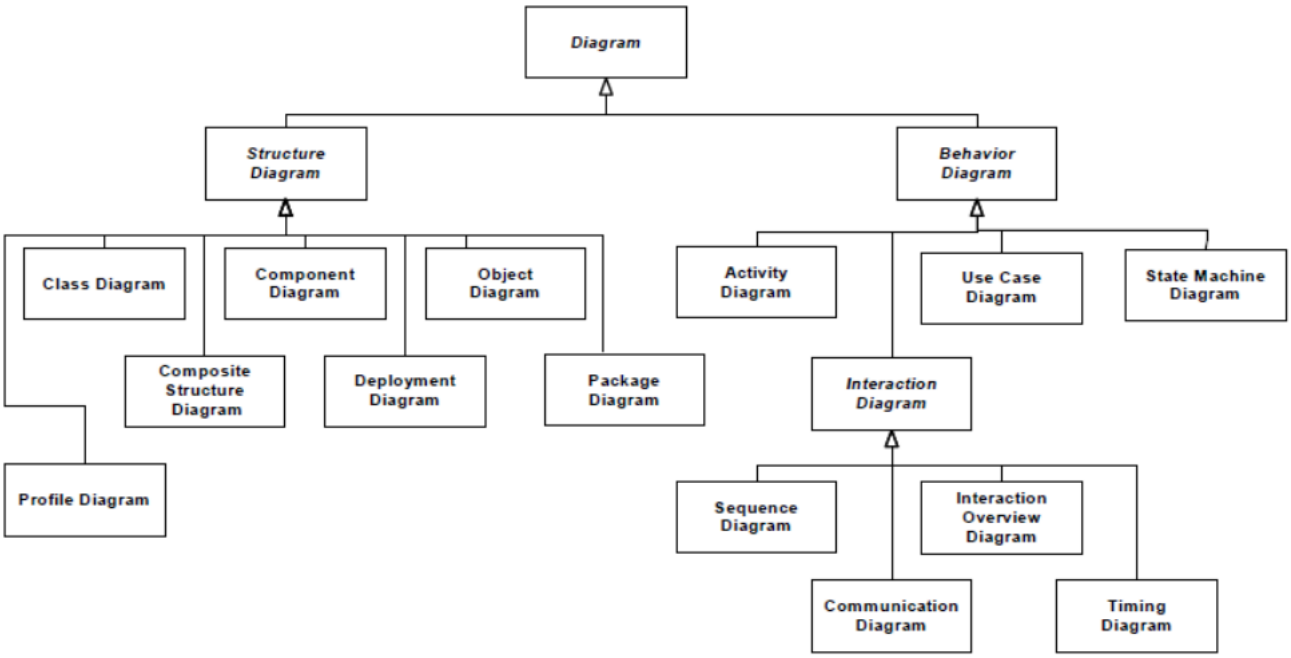
Modeliais grindžiama architektūra puikiai pritaikoma didelės apimties sistemomis, kurių kūrimas trunką ilgą laiką, dažnai pasinaudojant krioklio ar fontano programinės įrango kūrimo procesais. Tačiau MDD nėra tinkamas tuo atveju kai, modelio paruošimas daug kartų viršiją visas kitas programinės įrangos kūrimo proceso veiklas. Taip gali nutikti kai projekto apimtis ar funkcionalumas nėra didelis ir jis yra unikalus kitų dažniausiai kuriamų projektų atžvilgiu (toje pačioje įmonėje).

Viena iš plačiausiai naudojamų modeliavimo kalbų (taip pat ir MDA paradigmoje) yra vieninga modeliavimo kalba UML (angl. *Unified Modeling Language*). UML yra vaizdinė modeliavimo kalba, sukurta aprašyti modeliais:

- verslo procesus;
- sistemų analizę, projektavimą ir realizavimą;
- kitas ne programines sistemas ar procesus.

UML diagrama – tai grafinis kuriamos sistemos modelio ar jau sukurtos sistemos vaizdinys tam tikru pjūviu. UML projekto modelis susideda iš keleto skirtingų tipų diagramų. Modelius galima tikrinti, vykdant žingsnius iš eilės, tai palengvina klaidų aptikimą ankstyvose projekto stadijose. UML modeliai taip pat padeda sekti informacijos pokyčius, neatitikimus. UML diagramos internetinių informacinių sistemų kūrimo ir palaikymo metu leidžia lengviau perteikti informaciją ir sistemų kūrėjams tarpusavyje, ir verslo atstovams. Taip pat UML diagramos aprašo sistemos funkcionalumą ir architektūrą, kuria remiantis galima kurti, palaikyti ar plėtoti informacinę sistemą.

Šiame darbe naudojama naujausia UML 2.5 versija [9]. UML 2.5 diagramų tipai ir jų pagrindinės grupės pateikiamos 1.1 pav.




**1.1 pav. UML 2.5 diagramų tipai [9]**

Pagrindinės dvi diagramų grupės, į kurias skirstomos UML diagramos tai:

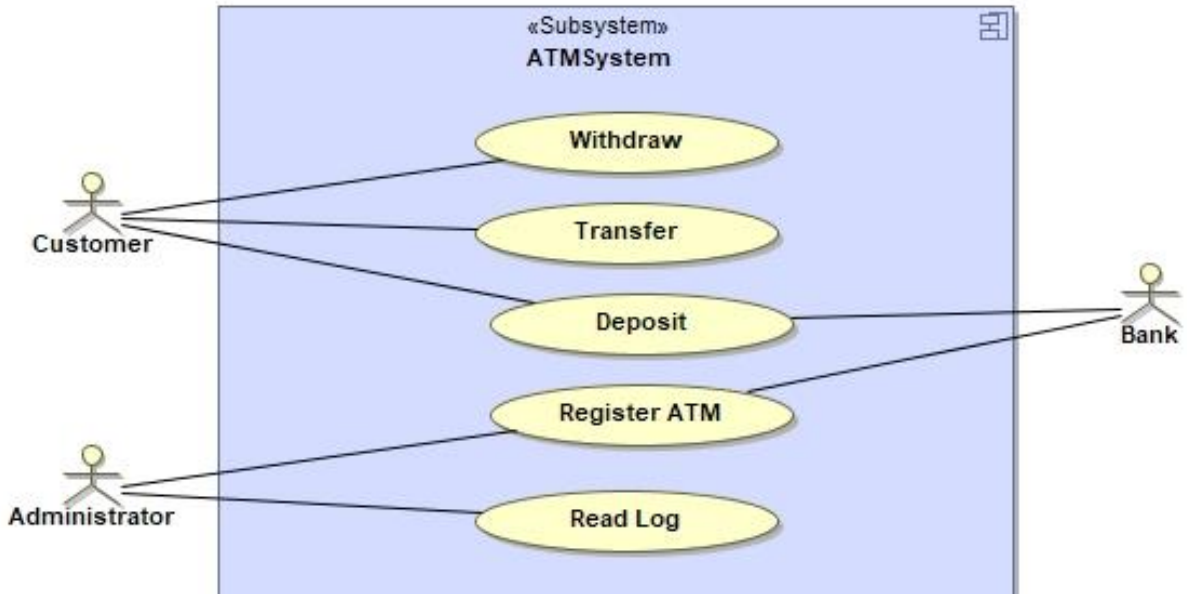
1. Struktūrinės diagramos, aprašančios sistemos statinį vaizdą ir pademonstruojančios, kaip sistemoje esantys komponentai siejasi tarpusavyje:
   1.1. klasių diagrama, kuri skirta nurodyti sistemos, posistemio ar komponento struktūrą susijusiomis klasėmis ir sąsajomis, su savo dalimis, apribojimais ir ryšiais;
   1.2. objektų diagrama;
   1.3. paketų diagrama, kuri aprašo paketus ir ryšius tarp jų;
   1.4. komponentų diagrama, kuri aprašo fizinius ar programinius komponentus ir priklausomybes tar šių komponentų;
   1.5. sudėtinės struktūros diagrama, kuri skirta aprašyta tam tikro objekto vidinę struktūrą arba kaip šis objektas bendradarbiauja su kitais objektais;
   1.6. diegimo diagrama, kuri skirta aprašyti sistemos architektūrą, nurodant kaip programinės įrangos objektai siejasi su techninės įrangos objektais;
   1.7. profilio diagrama, kuri leidžia atvaizduoti UML meta modelį skirtingoms platformoms ar skirtingoms dalykinėms sritims.

2. Elgsenos diagramos, aprašančios, kas turi vykti modeliuojamoje sistemoje:
   2.1. veiklos diagrama, kuri aprašo žemo lygio sistemos veiksmų sekas ir reikalavimus toms sekoms;
   2.2. būsenų diagrama, kuri skirta modeliuoti objektų būsenas ir metodus, kuriuos naudojant objektas pereina iš vienos būsenos į kitą;
   2.3. panaudojimo atvejų diagrama, kuri aprašo naudotojo veiksmus, kurios sistema ar jos dalis turi ar gali įvykdyti vienam ar keliems išoriniams naudotojams;
   2.4. sąveikų diagramos, kurios skirtos atvaizduoti objektų komunikavimą pranešimais;
   UML išskiria keturis sąveikų diagramų tipus:
      2.4.1. sekų diagrama, kuri aprašo kaip keičiamasi informacija tarp objektų perduodant žinutes;
      2.4.2. komunikacijos diagrama, kuri koncentruojasi į bendravimą tarp komponentų ir kaip jie reaguoja į gautas ar perduodamas žinutes;
      2.4.3. laiko diagrama, kuri atvaizduoja, kaip keičiasi objektų būsenos, kintant laikui;
      2.4.4. sąveikų apžvalgos diagrama, kuri nurodo komunikavimą per keletą veiklos diagramų, susikoncentruojant į valdymo srautus.

Atsižvelgiant į projekto dydį ir tipą, parenkamos naudojamos diagramos. Dažniausiai naudojamos – klasių, sekų, panaudojimo atvejų ir veiklos diagramos. UML modeliavimo kalba tinka modeliuoti internetines informacines sistemas ir dėl jos sąlyginio paprastumo ir naudojimo populiarumo.

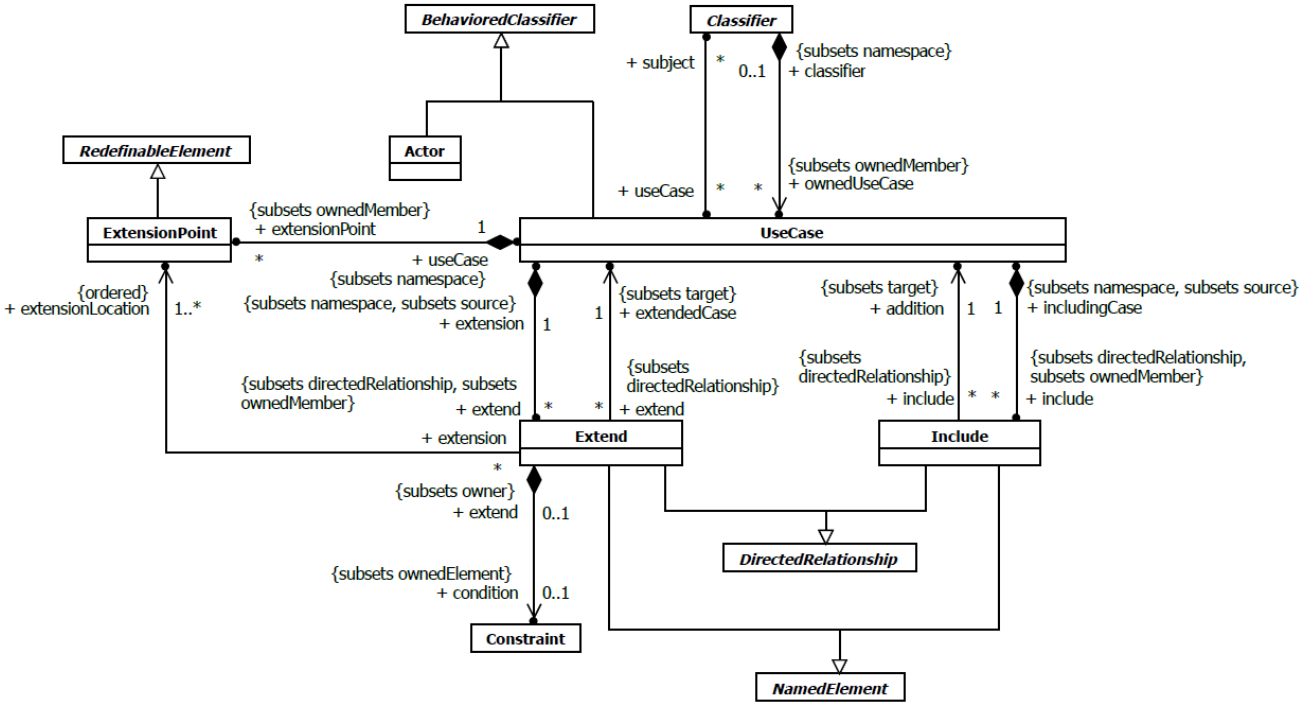
Šiame darbe siekiama iš veikiančios sistemos išgauti pagrindinius veiksmus, kuriuos naudotojais gali atlikti sistemoje – tam aprašyti tinka UML panaudojimo atvejų diagrama. Kiekvieną išgautą panaudojimo atvejį būtų naudinga detalizuoti, aprašant jo scenarijų – tam tinka UML veiklos diagrama. Taigi, toliau detaliau analizuosime šias dvi UML diagramas.

UML panaudojimo atvejų diagrama skirta aprašyti sistemos funkcinius reikalavimus. Panaudojimo atvejų diagramoje vaizduojama sistemos naudotojų sąveika su sistema ar sistemos sąveika su kitomis sistemomis. Pagrindiniai elementai šioje diagramoje yra panaudojimo atvejai, aktoriai ir ryšiai tarp jų. Panaudojimo atvejis aprašo sistemos funkcionalumą tam tikru pasirinktu detalumo lygmeniu. Aktorius panaudojimo atvejų diagramoje atvaizduoja sistemos naudotojo rolę ar kitą sistemą, kuri naudojasi tam tikru siūlomu funkcionalumu. Panaudojimo atvejai gali būti detalizuojami veiklos diagramomis, apibūdinančiomis panaudojimo atvejo funkcionalumą aukštu abstrakcijos lygmeniu. Daugelis kitų diagramų modeliuojant sistemos projektą panaudojimo atvejų diagramą naudoja kaip atskaitos tašką lyginant, ar teisingai sukurti kiti sistemos modeliai. 1.2 pav. pateikiamas UML panaudojimo atvejų diagramos pavyzdys [9].




**1.2 pav.** UML panaudojimo atvejų diagramos pavyzdys **[9]**

UML kalbą sudarantys elementai detalizuojami UML 2.5 metamodelyje [9]. Panaudojimo atvejų diagramos metamodelis pateikiamas 1.3 pav. Šio metamodelio detalius paaiškinimus galima rasti 1.1 lentelėje. Šioje lentelėje aprašyti pagrindiniai panaudojimo atvejų diagramos elementai, kurie dažniausiai naudojami, nedetalizuojant elementų naudojamų specializuotuose modeliavimo atvejuose.




**1.3 pav.** Panaudojimo atvejų diagramos metamodelis **[9]**

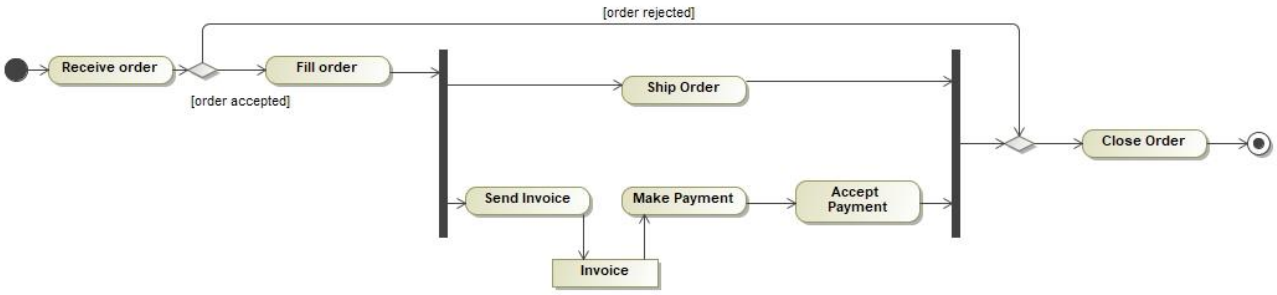
**1.1 lentelė Panaudojimo atvejų metamodelio elementų detalizavimas**

| **Elementas** | **Apibūdinimas** |
|---|---|
| *Usecase* | Elementas reprezentuojantis tam tikrą naudingą sistemos funkcionalumą. |
| *Actor* | Sistemos dalyvį ar kitą sistemą reprezentuojantis elementas. |
| *Extend* | Elementas, skirtas atvaizduoti ryšiui tarp dviejų panaudojimo atvejų, nurodant tam tikrą sąlygą, kuriai įvykus įvykdoma sujungto panaudojimo atvejo veikla. |
| *Include* | Elementas, skirtas atvaizduoti ryšiui tarp dviejų panaudojimo atvejų nurodant, kad į vieno panaudojimo atvejo veiklą įeina kito elemento veikla. |
| *Association* | Ryšys tarp aktoriaus ir panaudojimo atvejo. |
| *Generalization* | Ryšys, skirtas atvaizduoti paveldėjimą tarp elementų. |

UML panaudojimo atvejų scenarijai gali būti detalizuojami kelių tipų diagramomis, tačiau dažniausiai naudojamos veiklos ar sekų diagramos. Veiklos diagramos turi paprastesnę notaciją, todėl geriau tinka aprašyti panaudojimo atvejo veiklą aukštesniame abstrakcijos lygmenyje. Sekų diagramos dažnu atveju reikalauja tikslesnio modeliavimo, todėl geriau aprašo programinio kodo veikimo principus.

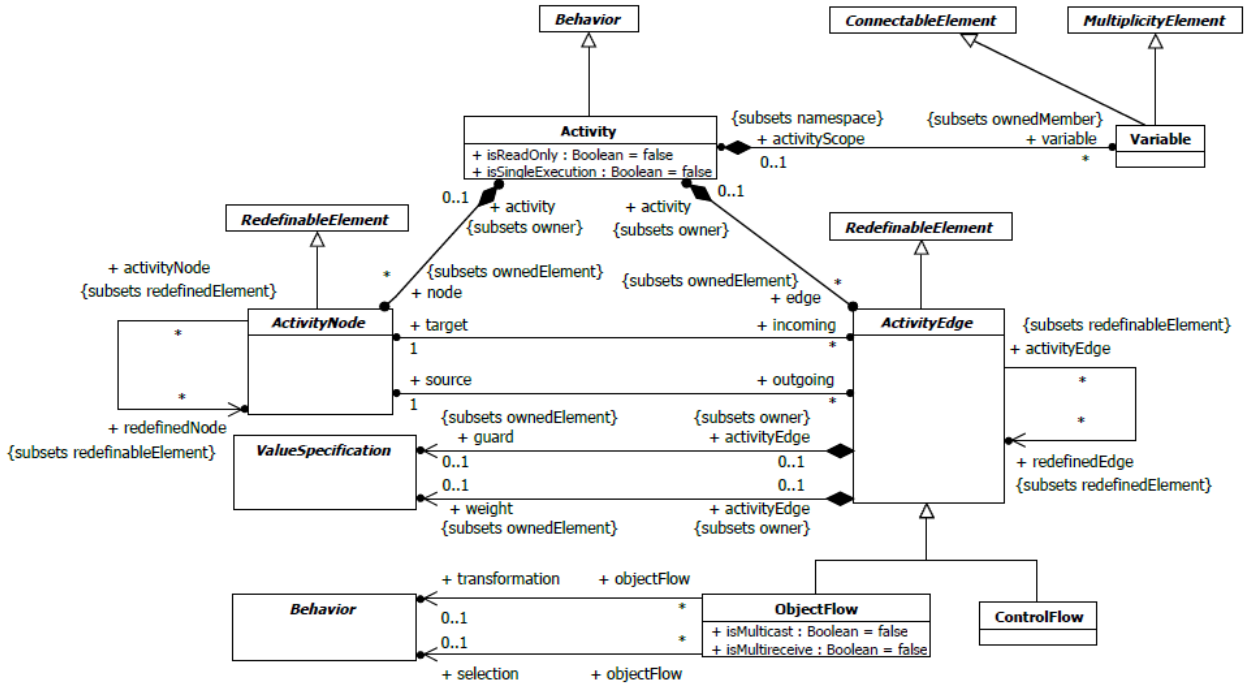
Veiklos diagramos skirtos aprašyti veiklos procesus. Šie procesai gali būti kompiuterizuoti arba ne. Veiklos diagramas dažniausiai sudaro veikla, kuri detalizuota veiksmų sekomis. Šie veiksmai gali būti organizacijoje vykdomos operacijos ar programinio kodo funkcijų iškvietimas. Aprašyti veiksmai jungiami į veiksmų sekas, kurie ir sudaro veiklos diagramą. Veiklos diagramoje taip pat galima atvaizduoti operavimą duomenimis ar objektais, kas padeda nustatyti, kada objektai ar duomenys yra reikalingi veikloje ir kada gali pasikeisti objekto ar duomenų būsenos. Veiklos diagramos taip pat gali būti naudojamos informacinių sistemų modeliavime aprašyti sisteminio lygio procesus [9].

Pavyzdinė veiklos diagrama pavaizduota 1.4 pav., šiame procese vaizduojamas užsakymo pateikimo procesas.



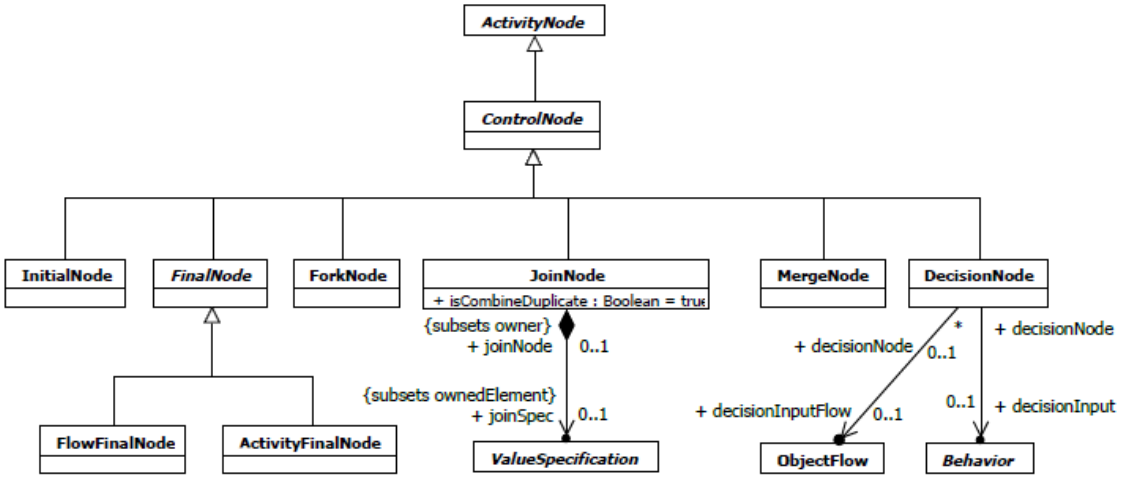

**1.4 pav.** UML veiklos diagramos pavyzdys **[9]**

UML veiklos diagramos susideda iš daugelio elementų. Visus šiuos elementus aprašantys metamodeliai pateikiami: 1.5 pav., 1.6 pav. ir 1.7 pav. Šiuose paveikslėliuose atvaizduojama elementų hierarchija parodant, kaip jie siejasi tarpusavyje. Dažniausiai veiklos diagramose naudojami elementai aprašyti 1.2 lentelėje.



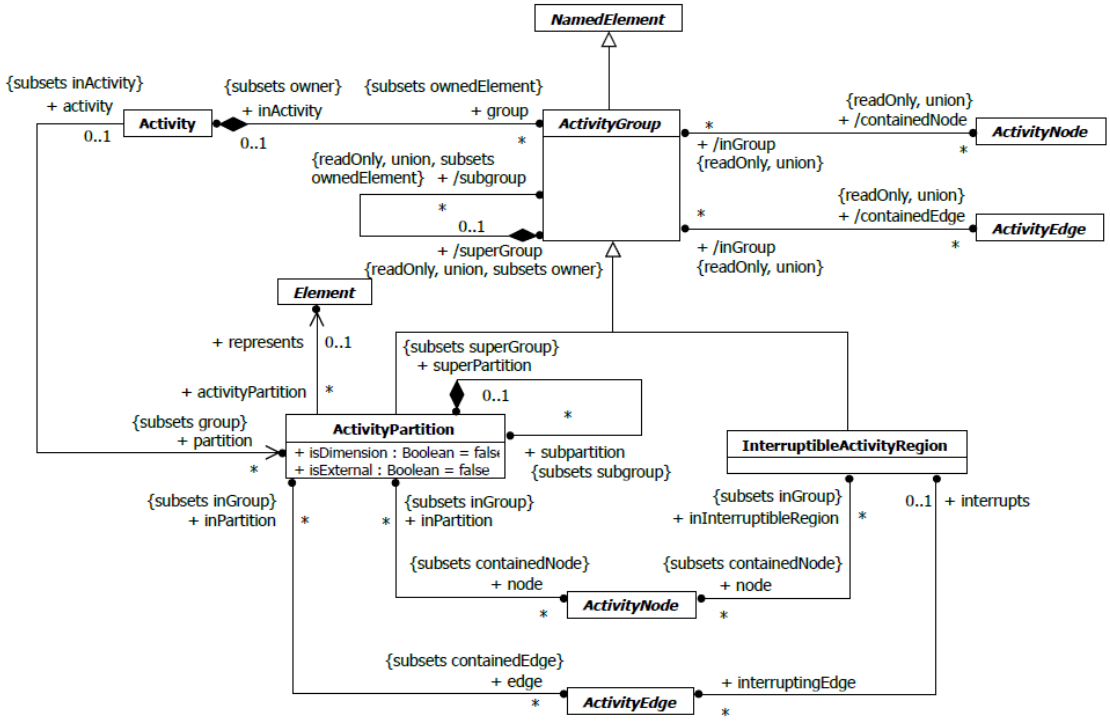

**1.5 pav. UML veiklos diagramos metamodelio fragmentas**

Abstrakčiausias veiklos diagramų metamodelis pateikiamas 1.5 pav. Jame matome, kaip veiklą aprašantis (*Activity)* elementas siejasi su valdymo mazgais, kurie išvedami detalizuojant *ActivityNode* elementą. Taip pat matome, kokiais ryšiai galima junti keletą *ActivityNode* tipo elementų – tai *objectFlow* ir *controlFlow* ryšiai. Valdymo elementus detalizuojanti diagrama atvaizduota 1.6 pav. joje matome pradžios, pabaigos sprendimo apjungimo ir kitus elementus.




**1.6 pav. UML veiklos diagramos valdymo mazgų metamodelis**

Veiklos diagramos veiklų grupių metamodelis pateikiamas 1.7 pav. Ši diagrama atvaizduoja, kaip veiklos elementai gali būti jungiami į grupes, kaip pavyzdžiui struktūrinis veiklos mazgas, skirtas veiklos procese detalizuoti kito proceso veiksmų seką.




**1.7 pav. UML veiklos diagramos veiklų grupių metamodelis**

**1.2 lentelė Veiklos diagramos elementų detalizavimas**

| **Elementas** | **Apibūdinimas** |
|---|---|
| *Activity* | Veikla, aprašanti tam tikrą procesą. |
| *Action* | Veiksmas, atliekamoje veikloje. |
| *Initial node* | Elementas, nurodantis veiklos pradžią. |
| *Activity final node* | Elementas, nurodantis veiklos pabaigą. |
| *Decision node* | Sąlygos mazgas, leidžiantis išskirstyti kontrolės srautą į du ar daugiau srautų, jiems pritaikant sąlygas. |
| *Merge node* | Suliejimo mazgas, leidžiantis apjungti keletą srautų. |
| *Control flow* | Valdymo srautas, sujungiantis keletą veiksmų elementų tarpusavyje. |

Veiklos diagramos panaudojimo atvejų modelyje leidžia suprasti panaudojimo atvejų modelio ir kiekvieno detalizuoto panaudojimo atvejo apimtį. Tai suteikia galimybę aptikti klaidas panaudojimo atvejų modelyje ir panaudojimo atvejus, kurių veiklos diagramos turi ypač daug veiksmų, išskaidyti į mažiau abstrakčius. Taip suvienodinama panaudojimo atvejų modelyje naudojamų panaudojimo atvejų apimtis ir pagerinama panaudojimo atvejų diagramos kokybė. Veiklos diagramos taip pat gali būti taikomos aprašyti internetinėse sistemose atliekamą veiklą.

### 1.3.2. Internetinių informacinių sistemų specifika

Tradicinės internetinės informacinės sistemos skirstomos į statinius puslapius, serverio – kliento architektūros sistemas ir internetinės aplikacijas. Internetinės informacinės sistemos savo kūrimo procesu, diegimo aplinka ir kūrimo greičiu skiriasi nuo tradicinių informacinių sistemų. Remiantis knygą aprašančia tyrimus atliktus internetinių sistemų inžinerijoje [10], esminiai aspektai kurie paverčia internetinių informacinių sistemų kūrimo procesą unikaliu yra:

- daugelis internetinių informacinių sistemų pasižymi evoliucine raida, kuri išsiskiria nuolatiniu turinio, funkcionalumo, navigacijos ir dizaino atnaujinimu. Internetinės informacinės sistemos ypatingai evoliucionuoja funkcionalumo ir reikalavimų srityse. Dažnu atveju, daugiausia pokyčių įgyvendinama po sistemos įdiegimo. Internetinių informacinių sistemų kitimo tempas yra daug didesnis nei tradicinių. Tam įtaką daro tai, kad projekto pradžioje neįmanoma surinkti visų reikiamų reikalavimų. Internetinio turinio pokyčių dažnis taip pat sąlyginai didelis, todėl sėkmingai valdyti, keisti ir atnaujinti reikalavimus yra didelis techninis, organizacinis ir vadybinis iššūkis;
- internetinės informacinės sistemos turi būti pritaikytos naudoti plačiai auditorijai, kuri turi skirtingus reikalavimus, išankstines viltis ir įgūdžius. Todėl internetinių informacinių sistemų grafinė naudotojo sąsaja turi tenkinti įvairią auditoriją;
- internetinėse informacinėse sistemose informacija pateikiama pasitelkiant daugelį įvairių medijų, tokių kaip tekstas, grafiniai elementai, vaizdinė ir garsinė medžiaga. Turinys taip pat gali būti dinamiškai apdorojamas;
- internetinės informacinės sistemos, ypač pritaikytos globaliems naudotojams, turi atsižvelgti į skirtingus kultūrinius aspektus ir nacionalinius standartus, įskaitant skirtingus matavimo vienetus ir skirtingas kalbas;
- internetinės informacinės sistemos turi būti panaudojamos skirtingo dydžio ir pajėgumo įrenginiuose, programose ir formatuose. Būtina atsižvelgti į ypač skirtingus naudojamus interneto greičius;
- internetinių svetainių evoliucinis tobulinimas nusako, kad internetinių informacinių sistemų kūrimo procesas taip pat turi būti laipsniškas;
- internetinės informacinės sistemos gali pasižymėti įvairia architektūra, tačiau viena populiariausių – kliento serverio architektūra. Serveris šioje vietoje tai paslaugos teikėjas, o klientas paslaugos gavėjas. Paslaugos teikėjas savo mašinoje saugo programinį kodą, kurio pagalba sukuriami dinaminiai puslapiai ir per duomenų perdavimo kanalą perduodami klientui, kur jie atvaizduojami ir naudojami. Taigi klientui niekada neperduodamas nesukompiliuotas programinis kodas, kurio pagalba būtų galima atgalinės inžinerijos būdų atkurti sistemos programinį kodą.

### 1.3.3. Internetinių svetainių grafinė naudotojo sąsaja ir panaudojimo atvejų modelis

Internetinės informacinės sistemos grafinė naudotojo sąsaja gali būti sudaroma naudojant daugelį įvairių kalbų ir metodų, tačiau populiariausia ir dažniausiai naudojama yra HTML, CSS ir *JavaScript* kalbų kombinacija. Kitos programinės kalbos dažniau naudojamos kuriant internetinius interaktyvius sprendimus.

Hiperteksto žymėjimo kalba HTML (angl. *Hyper Text Markup Language*) – tai žymų kalba, skirta pateikti turinį internete. HTML susideda iš elementų, jų atributų ir kitų mazgų, tokių kaip tekstiniai mazgai esantys tarp elementų. Elementai HTML kalboje turi vardą ir reikšmę. HTML elementas gali tūrėti atributą, kuris gali nurodyti elemento veikimą arba suteikti būtinas reikšmes

elemento veikimui. *Cascading Style Sheets* (CSS) arba pakopiniai stilių šablonai – tai kalba, skirta aprašyti kita struktūrine kalba aprašyto dokumento vaizdavimą. *JavaScript* – tai objektinė skriptų programavimo kalba, dažniausiai naudojama internetinių puslapių interaktyvumo realizacijai.

Atsižvelgiant į internetinių informacinių sistemų įvairovę, šiame darbe pasirinkta registruoti naudotojų veiksmus naudojantis sistema ir šias veiksmų sekas atgalinės inžinerijos metodu transformuoti į UML panaudojimo atveju diagramas, kurios detalizuojamos veiklos diagramomis.

### 1.3.4. Atvirkštinė inžinerija

Atvirkštinė inžinerija [11]– tai procesas, kurio metu išgaunama informacija apie sukurtą sistemą, siekiant atkurti šią sistemą, jos komponentą ar kitą informaciją apie nagrinėjamą sistemą. Daugiausiai pasiūlyta sprendimų gebančių atvirkštinės inžinerijos būdu išgaunančių klasių diagramas, atliekant statinę ar dinaminę programinio kodo analizę. Šie spendimai dažnai integruojami į programinio kodo kompiliatorius [12]. Atsižvelgiant į tai, kad dabartinės atvirkštinės inžinerijos metodu diagramas išgaunančios programos daugiausia siekia išgauti struktūrines diagramas, buvo nuspręsta sukurti naują įrankį, kuris būtų orientuotas į elgsenos tipo diagramas.

Internetinių sistemų informacinių sistemų atvirkštinė inžinerija gali būti vykdoma daugeliu būdų, tačiau dažniausiai naudojami yra statinės ir dinaminės kodo analizės metodai.

Statinės kodo analizės metu nuskaitomi programinio kodo failai, išgautą informaciją išsaugant tam tikroje struktūroje, kuri vėliau panaudojama generuoti atvirkštinės inžinerijos rezultatus.

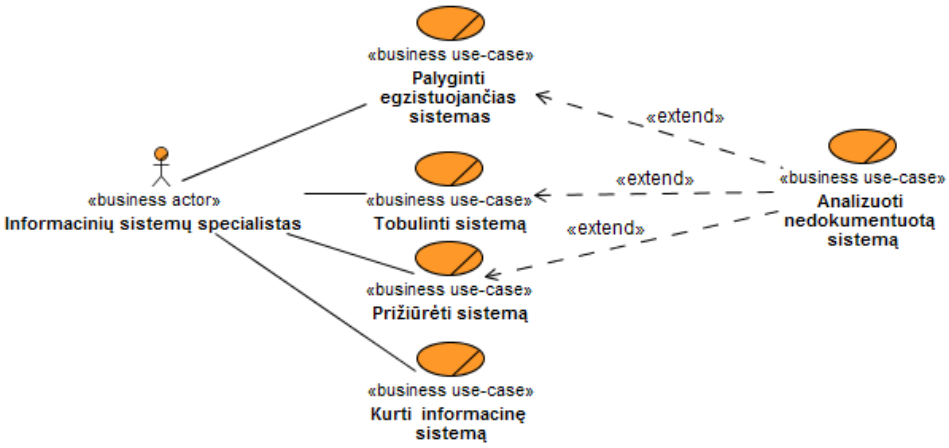
Dinaminės kodo analizės metu sekama veikianti programa, renkant informaciją apie programinio kodo pokyčius, juos registruojant ir vėliau panaudojant generuoti atvirkštinės inžinerijos rezultatus.

Naudojant šiuos du metodus kartu, pirmiausia įvykdomas statinės kodo analizės etapas, sugeneruojantis didžiąją dalį tikėtinų rezultatų. Tačiau dėl dinaminės puslapių prigimties šis analizės būdas gali neatvaizduoti visų galimų rezultatų. Todėl naudojama dinaminė kodo analizė, kuri siekia surasti dinaminius kodo pokyčius ir papildomai juos išanalizuoti. Atgalinės inžinerijos metu UML struktūrinių diagramų generavimui dažniausiai naudojama statinė kodo analizė arba statinės ir dinaminės kodo analizės mišinys.

Atgalinė inžinerija standartinei programinei įrangai dažnu atveju gali būti vykdoma analizuojant programinį kodą vienoje ar kitoje formoje. Šis atgalinės inžinerijos procesas išanalizuoja programinės įrangos programos kodą ir pateikia gautus rezultatus, kurie vėliau gali būti tobulinami ekspertų ar iš gautų rezultatų gaunamos tam tikros išvados apie sistemą. Tačiau, norint atlikti internetinių informacinių sistemų atgalinę inžineriją, programos kodą ne visais atvejais galima išgauti, dėl dažnai naudojamos serverio-kliento architektūros, kai klientui pateikiami tik programos sugeneruoti dinaminiai rezultatai.

## 1.4. Tyrimo objekto naudotojų analizė

Informacinių sistemų kūrimo metu UML modeliai pasitarnauja visose programinės įrangos kūrimo stadijose. Informacinių sistemų specialistai, naudodami UML diagramas, gali struktūriškai atvaizduoti sistemos architektūrą ir jos funkcionalumą (1.8 pav.). Informacinių sistemų specialistai UML modeliais gali pasinaudoti analizuodami liktinės ar panašios sistemos funkcionalumą ir architektūrą. Perėmus informacinės sistemos priežiūrą iš kitų informacinės sistemų kūrėjų ar trečiųjų asmenų, UML modeliai pasitarnauja kaip mokymosi medžiaga, leidžianti greitai ir efektyviai įsigilinti į sistemos funkcionalumą ir architektūrą. Projektuojant ir tobulinant sistemą UML modeliai yra pamatinis elementas, kuriuo remiantis priimami sprendimai. UML modeliai taip pat pasitarnauja tikrinant, ar sistema atitinka specifikaciją ir reikalavimus. UML modeliai taip pat leidžia priimti sprendimus, kokių funkcinių reikalavimų sistemoje vis dar trūksta. Turint kelių panašių sistemų specifikaciją, galima lyginti jas tarpusavyje, suprantant kuo informacinių sistemų funkcionalumas skiriasi tarpusavyje ir kur jas būtų galima patobulinti.



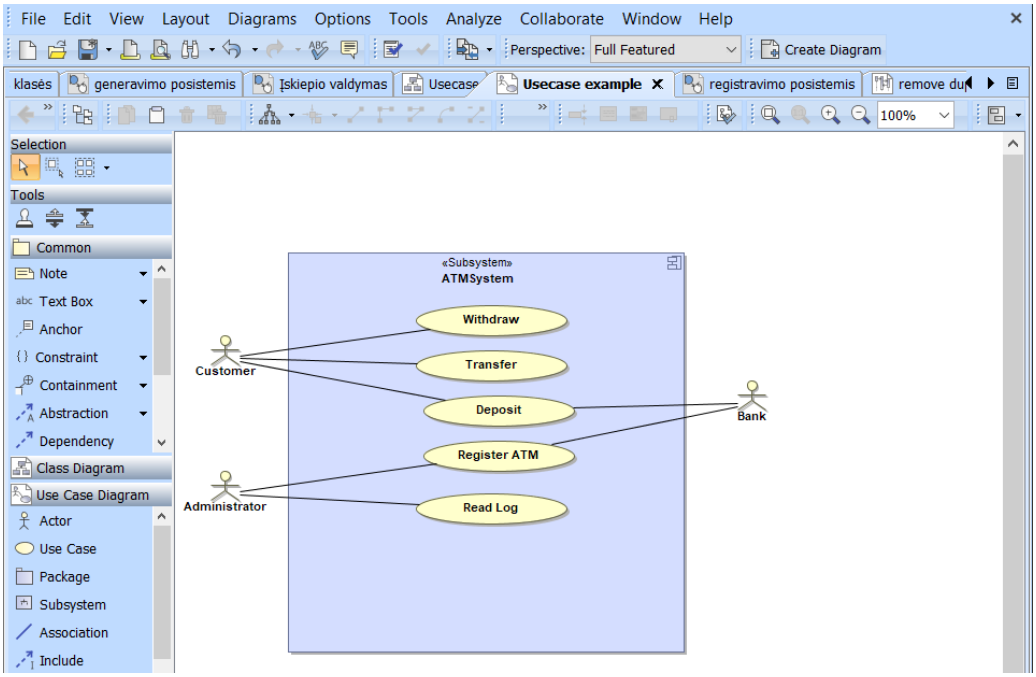

**1.8 pav. Pagrindinės informacinių sistemų specialisto funkcijos, kuriose naudojami UML modeliai**

## 1.5. Esamų problemos sprendimo metodų analizė

### 1.5.1. Įrankių, naudojamų UML modelių projektavimui, analizė

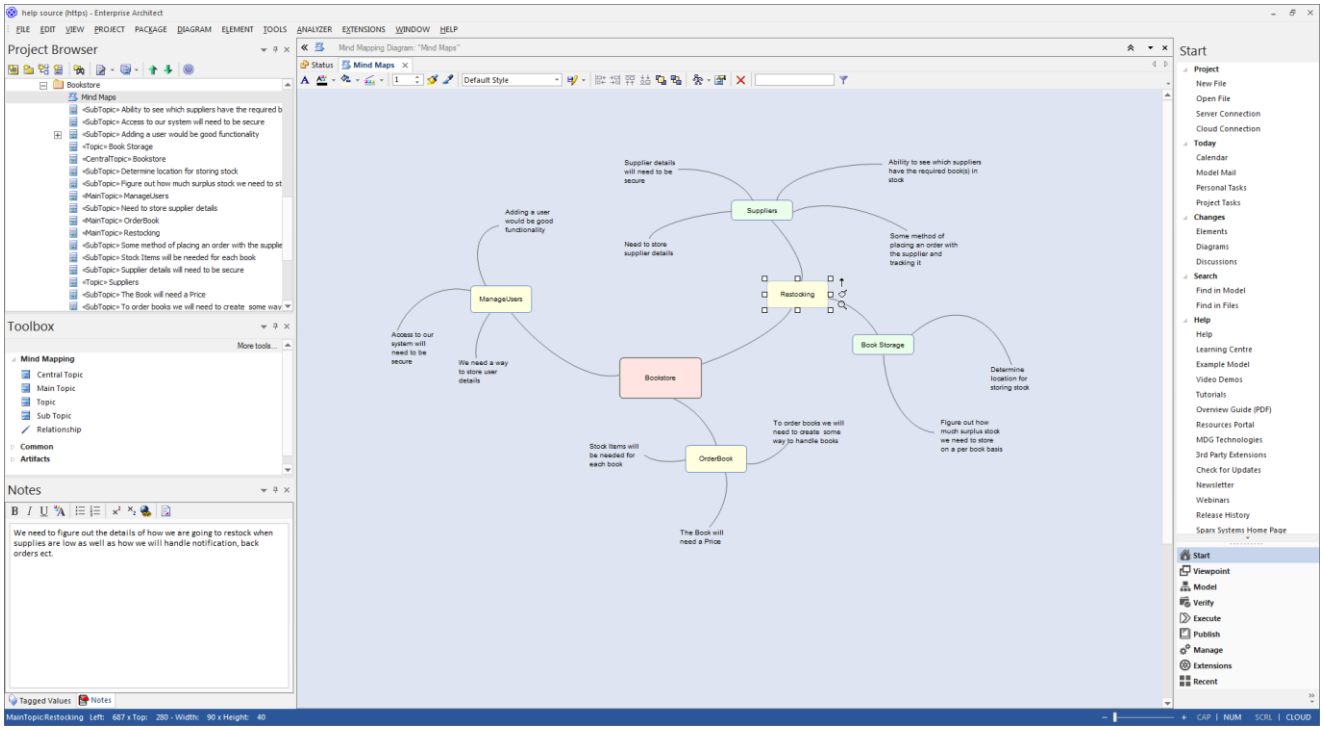
Šiame darbe detaliau analizuojami trys UML modeliavimo įrankiai: „MagicDraw“, „Enterprise Architect“ ir „Rational Software Architect“.

„MagicDraw“ [13] (1.9 pav.), tai kompanijos „No Magic“ sukurtas modeliavimo įrankis, palaikantis keletą modeliavimo kalbų ir karkasų. Įrankis naudojamas organizacijų architektūros, programinės įrangos ir sistemų modeliavimui. Šis įrankis turi daug įvairių paketų ir priedų taip pat leidžia atgalinės inžinerijos metodu generuoti modelius iš daugelio programavimo kalbų. Šis programinės įrangos paketas puikiai pritaikomas panaudojimo atvejų modelių kūrimui, kurie vėliau gali būti eksportuojami į kitus įrankius XMI formatu. Programa „MagicDraw“ taip pat palaiko naujausią UML 2.5 versiją ir yra suderinama su „Object Management Group“ (OMG) standartizuotais UML kalbos standartais. Nors programa ir palaiko daugelį atgalinės inžinerijos metodų, ji nepateikia pakankamai reikalingų funkcijų išgauti panaudojimo atvejų modelį iš internetinių informacinių sistemų, neturint pilno sistemos programinio kodo.



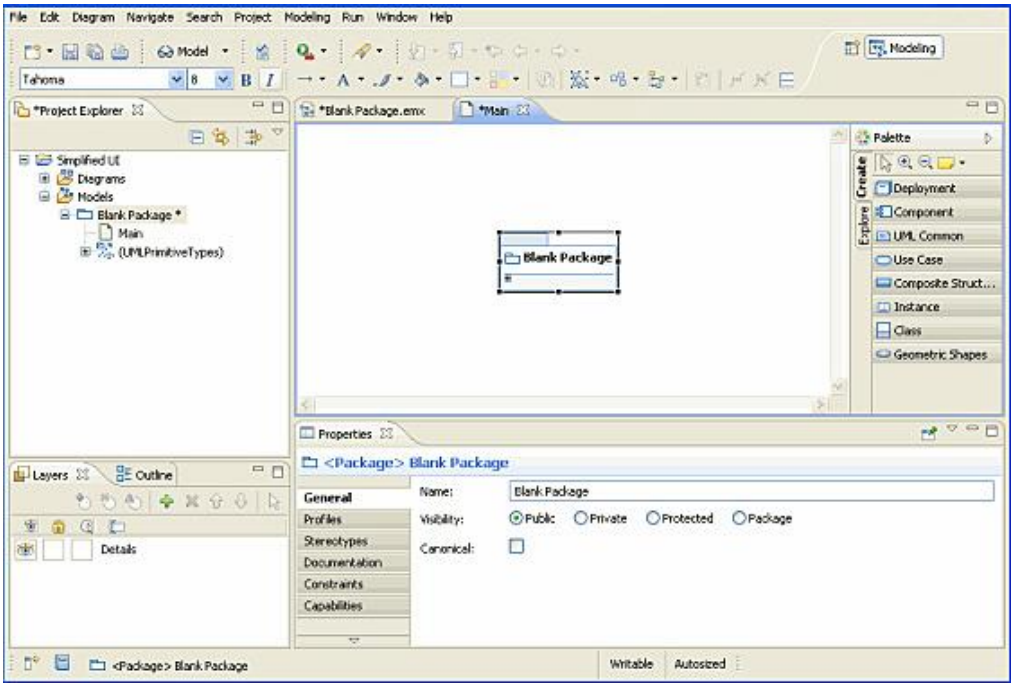

**1.9 pav.** „MagicDraw“ **[13]** programinės įrangos grafinės naudotojo sąsajos pavyzdys

„Enterprise Architect“ [14] (1.10 pav.) – kompanijos „Sparx Systems“ sukurtas modeliavimo įrankis, palaikantis daugelį modeliavimo kalbų, lengvai besiintegruojantis su kitais įrankiais ir turintis galimybę sukurti modelius iš daugelio programavimo kalbų. Šis įrankis taip pat pateikia galimybę išgauti UML modelius atgalinės inžinerijos būdu ir palaiko daugelį kalbų, tačiau neteikia galimybės generuoti panaudojimo atvejų diagramas iš šių programavimo kalbų kodo. Plačiausiai palaikomas atgalinės inžinerijos būdu išgaunamas diagramos tipas yra klasių diagrama. Šis įrankis, kaip ir „MagicDraw“, netenkina reikalavimų šiame darbe siekiamam rezultatui išgauti.




**1.10 pav.** „Enterprise Architect“ **[14]** programinės įrangos grafinės naudotojo sąsajos pavyzdys

„Rational Software Architect“ [15] (1.11 pav.) – kompanijos IBM įrankis, realizuotas naudojantis nemokamu atviro kodo programinės įrangos karkasu „Eclipse“, sukurtas programinės įrangos modeliavimui ir palaikantis atgalinės inžinerijos bei kodo generavimo funkcijas. Skirtingai nei „MagicDraw“ ir „Enterprise Architect“, „Rational Software Architect“ palaiko nedidelį kiekį programavimo kalbų iš kurių galima išgauti modelius atgalinės inžinerijos būdu.



**1.11 pav. „**Rational Software Architect“ **[15]** grafinės naudotojo sąsajos pavyzdys

**1.3 lentelė. Įrankių, naudojamų UML modelių projektavimui, palyginimas**

| **Įrankis** | **„MagicDraw“** | **„Rational Software Architect“** | **„Enterprise Architect“** |
|---|---|---|---|
| Sukūrusi kompanija | „No Magic“ | „IBM“ | „Sparx Systems“ |
| Atgalinė inžinerija | + | + | + |
| Kodo generavimas iš modelių | + | + | + |
| Programavimo kalbos, kurios atgalinės inžinerijos metodu gali būti transformuotos į modelius. | Java, C++, C#, CIL, CORBA IDL, DDL, EJB, XML Schema, WSDL | C++, Java, WSDL | ActionScript, C, C#, C++, Delphi, Java, PHP, Python, Visual Basic, Visual Basic .NET, DDL, XML Schema, WSDL |
| Integruojasi su kita programine įranga | + | + | + |
| Galimybė generuoti modelius iš sistemų grafinės naudotojų sąsajos ir užregistruotų sistemos veiksmų. | - | - | - |
| XMI formato palaikymas | XMI 2.4 | XMI 1.3 | XMI 2.1 |

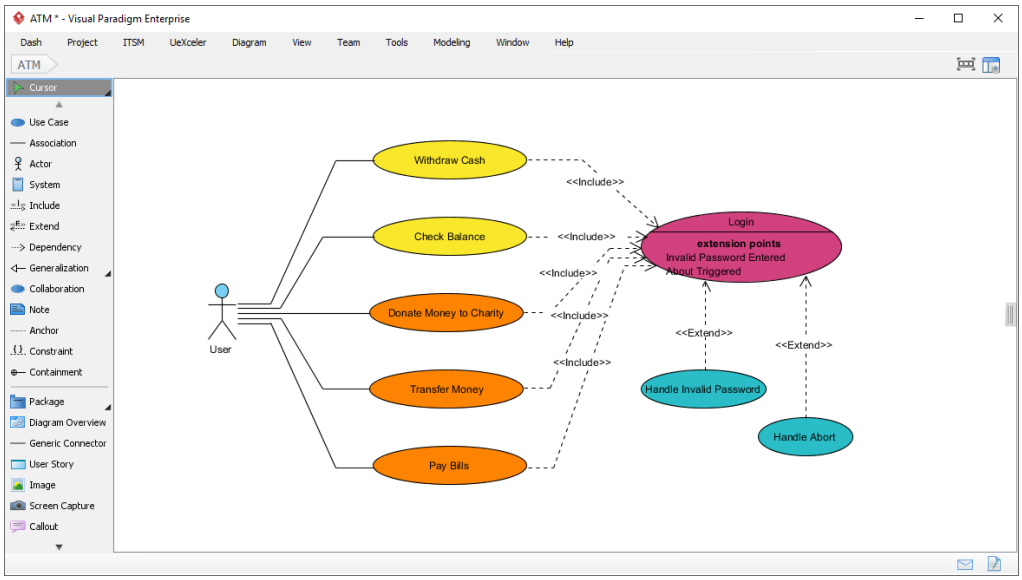
Nors aptarti įrankiai turi galimybę transformuoti programinį kodą iš daugelio programavimo kalbų, tačiau nei vienas įrankis nepalaiko modelių generavimo iš veikiančios internetinės informacinės sistemos, jei neturime programinio kodo. Kadangi siekiama sukurti tokį įrankį, kuris galėtų sugeneruoti sistemos panaudojimo atvejų modelį, nepriklausomai nuo to, ar turime programinės įrangos išeities kodą, ar ne, ši programinė įranga negali patenkinti mūsų poreikių. Tačiau ši programinė įranga puikiai tinka atvaizduoti panaudojimo atvejų modelius. Atlikus analizę, nuspręsta pasirinkti kompanijos „No Magic“ programinės įrangos paketą „MagicDraw“ teikiamą mokslo įstaigoms, dėl didelės ekspertinės patirties dirbant su įrankiu ir galimybės lengvai importuoti ir eksportuoti failus XMI formatu.

### 1.5.2. Specializuotų įrankių UML modeliams generuoti analizė

Išanalizavus UML modeliavimo įrankius ir nesuradus įrankio, kuris leistų sugeneruoti panaudojimo atvejų modelį pasinaudojant standartiniu funkcionalumu, buvo išanalizuoti specializuoti sprendimai, kurie gali būti pritaikomi internetinėms informacinėms sistemoms ar gali analizuoti IIS programinį kodą.

„WebUML“ – Carlo Bellettini, Alessandro Marchetto, Andrea Trentini [16] pasiūlytas sprendimas pritaikytas klasių ir būsenų diagramų generavimui. Šis įrankis diagramas generuoja naudodamas statinę ir dinaminę programinio kodo analizę. Statinė analizė vykdoma naudojant programinio kodo nuskaitymo metodus su nurodytų šablonų atpažinimo mechanizmu. Dinaminė kodo analizė vykdoma simuliuojant programinio kodo vykdymą. Išgauti modeliai šiek tiek perkrauti informacijos, todėl geriausiai tinka internetinių svetainių testavimui. Atsižvelgiant į tai, kad statinė kodo analizė buvo vykdoma pasitelkiant šablonų atpažinimo mechanizmą, galima daryti išvadą, kad šis sprendimas nėra pritaikomas dideliam skaičiui informacinių sistemų.

„Visual Paradigm“ (1.12 pav.) – naudojamas 230.000 [17] naudotojų atgalinės inžinerijos metodu diagramas generuojantis įrankis, palaikantis daugelį populiariausių programavimo kalbų ir galintis sugeneruoti bei sumodeliuoti UML klasių, sekų, komponentų ir kitas diagramas. Šis įrankis dažniausiai naudojamas kartu su integruotomis kūrimo aplinkomis (angl. *Integrated Development Environment* IDE), kurios leidžia programuoti ir kompiliuoti programinį kodą. Tačiau „Visual Paradigm“ atgalinės inžinerijos metodu generuojamoms diagramoms naudoja tik statinę kodo analizę, todėl jis labiau tinka generuojant struktūrines diagramas, tokias kaip klasių diagramos.




**1.12 pav.** „Visual Paradigm“ įrankio langas

„PHP2XMI“ [18] – įrankis, leidžiantis išgauti elgsenos tipo informaciją iš programinio kodo, naudojant statinės analizės metodus. Šis įrankis buvo skirtas palengvinti internetinių informacinių sistemų saugumo analizę ir PHP programavimo kalba sukurtų internetinių informacinių sistemų testavimą. „PHP2XMI“ veikia trimis etapais. Pirmasis etapas nuskaito programinį kodą. Šio etapo metu surenkama informacija apie puslapio URL adresus, puslapyje naudojamus *http* kintamuosius, sesijas ir slapukus (angl. *cookies*). Antrasis etapas įvykdo programinį kodą, sugeneruodamas vykdymo medžius, kurie vėliau išfiltruojami pašalinant perteklinę informaciją ir ją išsaugant SQL duomenų bazėje. Paskutinio etapo metu informacija iš SQL duomenų bazės transformuojama į UML 2.1 sekų diagramas, kurios gali būti atvaizduojamos visuose UML 2.1 palaikančiuose įrankiuose. Atsižvelgiant į tai, kad šiuo metu naujausia UML versija yra 2.5 šis įrankis yra kiek pasenęs, bet jame naudojamas metodas lengvai pritaikomas ir šiuolaikinėse internetinėse informacinės sistemose.

Šių trijų įrankių palyginimas pateiktas 1.4 lentelėje. Atsižvelgiant į lentelės duomenis matome, kad nei vienas įrankis neleidžia sugeneruoti panaudojimų atvejų diagramų ir tik vienas iš išnagrinėtų pavyzdžių diagramų generavimui naudoja grafinės naudotojo sąsajos analizę. Tarp įrankių dažniausiai pasitaikantis atvirkštinės inžinerijos taikymas – klasių diagramų generavimas naudojant statinę kodo analizę.

**1.4 lentelė. Įrankių UML modeliams generuoti atgalinės būdų palyginimas**

| Kriterijai | „WebUML“ | „Visual Paradigm“ | „PHP2XMI“ |
|---|---|---|---|
| Statinė kodo analizė | + | + | + |
| Dinaminė kodo analizė | + | - | + |
| Panaudojimo atvejų diagramų generavimas | - | - | - |
| Sekų diagramų generavimas | - | + | + |
| Klasių diagramų generavimas | + | + | - |
| Būsenų diagramų | + | - | - |
| Analizuojamas internetinių informacinių sistemų grafinės naudotojo sąsajos kodas | + | - | - |
| Analizuojamas internetinių informacinių sistemų programinis kodas | - | + | + |
| Metodas, naudojamas įrankyje, pritaikytas internetinėms informacinėms sistemoms | + | - | + |

### 1.5.3. Veiksmų registravimas

Naudotojų veiksmų registravimas internetinėse informacinėse sistemose plačiai naudojamas daugelyje sričių. Dažniausiai pasitaikantis veiksmų registravimo pavyzdys yra naudotojų veiksmų internetinėse sistemose sekimas, siekiant išgauti ir saugoti informaciją apie naudotoją marketingo tikslais. Tai atliekama siekiant sužinoti, kokia informacija domina naudotoją, ir taip rodyti tik jam pritaikytas reklamas. Tokias praktikas naudoja didžiosios korporacijos, tokios kaip „Google“ [19] ar „Facebook“. Taip pat veiksmų registravimas gali būti naudojamas, siekiant automatizuoti tam tikrą procesą. Tokių įrankių pavyzdžiais būtų galima laikyti „iMacros“ [20] ir „Selenium“ [21] įrankius. Pavyzdžiui, „Selenium“ įrankis leidžia įrašyti sistemos naudotojo įvedamas reikšmes ir internetinėje sistemoje vykdomus veiksmus, juos saugoti ir atkartoti siekiant ištestuoti sistemos veikimą.

Kitas veiksmų registravimo pavyzdys – tai „Google Analytics“ [19] įrankių rinkinys, skirtas stebėti ir analizuoti gautus duomenis apie internetinių sistemų naudotojus. Ši sistema leidžia nustatyti įvairius stebėjimo parametrus ir pagal juos stebėti veiksmus, atliktus svetainėse, kuriose įdiegti „Google Analytics“ reikalingi programinio kodo failai (angl. *script*).

Panašus į aptartus pavyzdžius naudotojų veiksmų registravimas internetinėse informacinėse sistemose gali būti panaudotas, išgaunant naudotojo veiksmų sekas, siekiant jas transformuoti į UML diagramas. Tačiau visi šie veiksmų registravimo metodai turi trūkumą – juose gautą informaciją sudėtinga saugoti struktūrizuotai todėl negalime tiesiogiai taikyti nei vieno iš šių įrankių teikiamo funkcionalumo tiesiogiai.

### 1.5.4. Metodų UML modeliams generuoti atvirkštinės inžinerijos būdu analizė

UML diagramų generatorius (angl. *UML Diagram Generator*) – tai įrankis ir metodas sukurtas Mohammad I. Muhairat [22], kuris leidžia generuoti klasių ir panaudojimų atvejų diagramas naudojant įvykių lenteles. Autorius apibrėžia tris įvykių tipus: išoriniai įvykiai, laikini įvykiai ir sąlyginiai įvykiai. Suvedus šiuos įvykius į struktūrizuotas įvykių lenteles sugeneruojamos diagramos.

Standartinis panaudojimo atvejų išgavimo metodas susideda iš penkių žingsnių:

1. Nustatyti aktorius kiekvienam įvykiui ar veiksmui iš šaltinių ir veiksmų galiniam gavėjui.
2. Nustatyti sąryšius tarp aktorių, jei jie egzistuoja.
3. Nustatyti panaudojimo atvejus (analitikas juos nustato iš veiksmo, kurį vykdo aktorius).
4. Nustatyti ryšius tarp panaudojimų atvejų.
5. Apjungti aktorius ir panaudojimo atvejus į vieną diagramą.

Generuodamas klasių diagramas, šis metodas pereina penkis žingsnius:

1. Kiekvienam įvykiui nustatoma klasė ar veiksmas iš veiksmų šaltinių ir objektų.
2. Nustatomi ryšiai tarp klasių, kurie standartiškai yra vienas su daug asociacijos ryšiai, nebent analitikas nustato kitaip.
3. Nustatomi klasių atributai, kuriuos analitikas išveda iš įvesties ir išvesties žinučių.
4. Nustatomos operacijos, kurios lengvai išvedamos iš veiksmų.
5. Rezultatai apjungiami.

Įvertinant metodo atliekamą nesudėtingą diagramų transformavimą, reiktų nepamiršti, kad šio metodo silpnybė slypi naudojamose įvykių lentelėse, kurios yra nesudėtingos struktūros, tačiau apimančios didelį kiekį duomenų, kuriuos surinkti užtrunka labai daug laiko.

Formų užpildymų metodu grįstas klasių diagramos generavimas [23] buvo pasiūlytas 2014 metais Akram Abdel Qader. Jis aprašo, kaip formomis grįstus dokumentus ar internetinius puslapius transformuoti į klasių diagramas. Metode aprašomi tokie keturi žingsniai:

1. Informacijos išgavimo procesas: šis procesas išsaugo suvedamus į formas dokumentus ir juos patalpina nenormalizuotose lentelėse.
2. Normalizavimo procesas: šis procesas nuskaito visus įrašus iš nenormalizuotos lentelės ir juos normalizuoja, taip pat sukurdamas ryšius, kurie saugomi naujoje normalizuotoje lentelėje.

3. Susiejimo procesas: šis procesas transformuoja normalizuotą lentelę į esybių ryšių diagramą.
4. Transformavimo procesas: šis procesas transformuoja esybių ryšių diagramą į klasių diagramą.

Analizuojamas metodas puikiai tinka pasirinktai sričiai, tačiau atsižvelgiant į duomenų išgavimo būdą ir sugeneruojamų diagramų tipus, pastebima, kad šis metodas siaurai pritaikomas.

„ReGUI“ [24] metodas ir įrankis pristatytas siekiant palengvinti kompiuterinių aplikacijų grafinės naudotojo sąsajos testavimą. „ReGUI“ įrankis generavimo metu išnagrinėja visus pasiekiamus meniu punktus, siekdamas nustatyti jų pradinę būseną. „ReGUI“ įrankis nustato ar buvo atidarytas naujas sisteminis langas bei registruoja, ar jis buvo uždarytas. Informacija apie šiuos įvykius ir elementų būsenas vėliau saugoma medžio tipo struktūrose, pašalinus pasikartojančius įrašus. Pasinaudojus šio projekto gerosiomis praktikomis galima sukurti įrankį, generuojantį UML diagramas ir pritaikyti jį internetinėms informacinės sistemoms.

**1.5 lentelė. UML modelių atvirkštinės inžinerijos metodų palyginimas**

| Metodas | **„UMLdg“** | **„Form fill document into UML class diagram“** | **„ReGUI“** |
|---|---|---|---|
| Statinė kodo analizė | + | + | + |
| Dinaminė kodo analizė | - | - | + |
| Sudaromos tarpinių rezultatų lentelės | + | + | - |
| Metodas leidžia sukurti sekų diagramas | - | + | - |
| Metodas leidžia sukurti klasių diagramas | + | + | - |
| Metodas leidžia sukurti panaudojimo atvejų diagramas | + | - | - |
| Metodas pritaikomas internetinėms informacinėms sistemoms | - | + | - |
| Analizuojamas informacinių sistemų grafinės naudotojo sąsajos kodas | + | + | + |
| Analizuojamas internetinių informacinių sistemų programinis kodas | - | + | - |
| Duomenys surenkami įvykiais pagrįstomis lentelėmis | + | + | - |

Palyginę šiuos metodus tarpusavyje galime pastebėti, kad visi metodai vykdo statinę kodo analizę, tačiau tik „ReGUI“ metodas vykdo dinaminę kodo analizę. Peržvelgiant UML diagramų generavimo galimybes „UMLdg“ metodas leidžia generuoti daugiausiai diagramų, tuo tarpu „Form fill document into UML class diagram“ yra skirtas generuoti tik klasių diagramas. Mažiausiai diagramų generuoja „ReGUI“, tačiau jis puikiai atvaizduoja, kaip atgalinės inžinerijos metodu išgauti sistemos grafinės sąsajos vaizdinį. Taip pat anksčiau aptartas „PHP2XMI“ įrankis realizuotas remiantis „ReGUI“ metodu, todėl pasirinkta išnagrinėti šį metodą. Išnagrinėjus visus metodus, buvo nustatyta, jog šiuo metu iškeltai problemai spręsti netinka vienas įrankis ar metodas, tačiau egzistuoja daug gerųjų praktikų, kurias galima pritaikyti šiame darbe siūlomame metode ir įrankyje.

### 1.5.5. Susiję tyrimai

IEEE Access žurnale išpublikuotame tyrime [25] pateikiama egzistuojančių modeliais grindžiamos atgalinės inžinerijos sprendimų analizę. Šioje apžvalgoje italų mokslininkai analizuoja modelius ir įrankius, naudojamus atgalinėje inžinerijoje. Taip pat analizuojami standartai ir sukurtų programų, kurios atgalinės inžinerijos būdu išgauna modelius ar programinį kodą, galimybės.

Tyrimo autoriai pabrėžia, kad modelių atgalinė inžinerija dar sąlyginai jauna sritis ir mokslinių tyrimų skaičius šioje srityje tik augs.

Taip pat naudinga nagrinėti net tik įrankius, kurie palaiko atgalinės inžinerijos metodus, bet ir metodus, sukurtus atgalinės inžinerijos būdu išgauti kitas elgsenos tipo diagramas. Puikus to pavyzdys būtų metodas, pasiūlytas prancūzų mokslininkų [26]. Šiame tyrime pateikiamas metodas skirtas dinamiškai generuoti UML sekų diagramas. Šis dinaminis metodas naudojamas analizuoti sistemas, kurioms negalime pritaikyti statinės kodo analizės. Tyrimas taip pat aprašo, kaip išgauti sistemos veikimo pėdsakus iš veikiančios sistemos. Šis tyrimas puikiai pritaikomas sistemoms, kurios sukurtas naudojant objektinį programavimą. Kadangi internetinės sistemos labai įvairios ir dažnai nėra sukurtos naudojant objektinį programavimą, šio tyrimo rezultatai tiesiogiai negali būti naudojami išgauti informacijos iš internetinių informacinių sistemų. Vis dėl to šis tyrimas suteikia naudingų įžvalgų apie UML diagramų išgavimą iš programinių sistemų atgalinės inžinerijos būdu.

Geruų praktikų taip pat galima rasti sprendime, tiesiogiai pritaikytame generuoti UML diagramas iš internetinių informacinių sistemų. Šį įrankį, pavadintą WARE, pasiūlė italų mokslininkai [27]. Įrankis skirtas atgalinės inžinerijos būdu generuoti UML diagramas. Jis leidžia generuoti panaudojimo atvejų diagramas, kuriose panaudojimo atvejai detalizuojami sekų diagramomis. Metodas ir jo pagrindu realizuotas įrankis pateikia puikius pavyzdžius, kaip sugeneruoti panaudojimo atvejų diagramą ir ją detalizuojančias diagramas bei būtų puikus įrankis išgauti modeliams iš internetinių informacinių sistemų, tačiau kaip savo įvestiems duomenis naudoja programos kodą. Programinio kodo, kaip įvesties duomenų, naudojimas apriboja svetainių analizę – galima analizuoti tik tokias internetines informacines sistemas, kurių programinį kodą galime išgauti. Taigi neturime galimybės pritaikyti šį metodą veikiančių internetinių sistemų analizei, kurių programinės įrangos kodas nepasiekiamas.

Išanalizavus tyrimus, vykdomus šioje srityje nustatyta, kad sprendžiama problema yra aktuali ir nagrinėjama daugelyje skirtingų tyrimų. Tinkamas sprendimas generuoti UML panaudojimo atvejų modelį internetinėms informacinėms sistemos, kurių programos kodas neprieinamas, dar nėra pasiūlytas. Analizuotų metodų, įrankių, sprendimų taikomos gerosios praktikos gali būti sėkmingai pritaikytos šiame darbe, kuriant metodą ir įrankį, gebantį generuoti UML panaudojimo atvejų modelį iš veikiančios internetinės informacinės sistemos, naudojant informaciją, gautą registruojant analizuojamos sistemos naudotojų veiksmus.

## 1.6. Analizės išvados

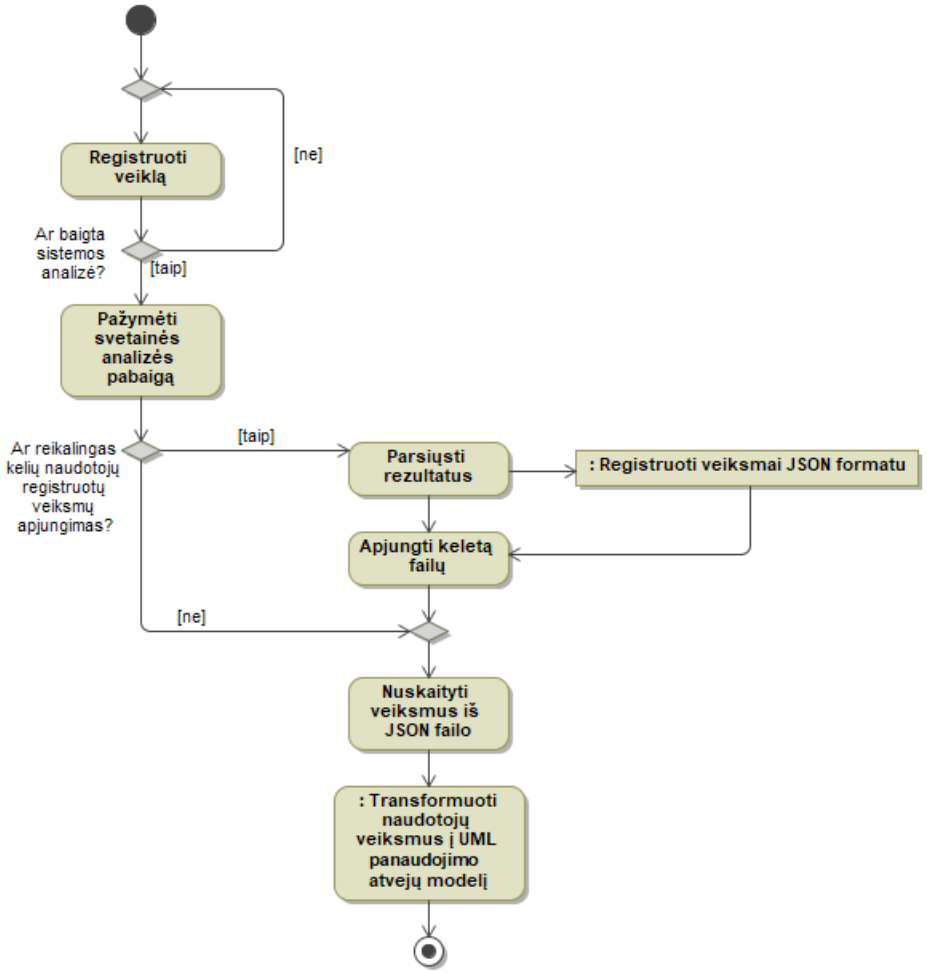
1. Išnagrinėjus internetinių informacinių sistemų specifiką nustatyta, kad metodai ir jų realizacijos geriausiai leidžiantys išnagrinėti IIS veikia naršyklėje, kuri suteikia galimybę nagrinėti grafinės naudotojo sąsajos elementus, neturint tiesioginio priėjimo prie programos kodo.
2. Išnagrinėjus atgalinės inžinerijos metodus nustatyta, kad tinkamiausias būdas UML diagramų generavimui iš internetinių informacinių sistemų – grafinės naudotojų sąsajos analizė, taikant statinę vartotojo sąsajos analizę ir naudotojo veiksmų registravimą.
3. Išnagrinėjus UML diagramas ir jų taikymą sistemų specifikavimui pastebėta, kad informaciją, išgautą registruojant naudojimąsi sistema, galima atvaizduoti UML panaudojimo atvejų ir veiklos diagramomis, kurios sudaro panaudojimo atvejų modelį.
4. Išanalizavus rinkoje egzistuojančius įrankius ir metodus nebuvo surastas sprendimas, tinkantis UML modelių generavimui iš internetinių informacinių sistemų grafinės naudotojo sąsajos ir veiksmų registravimo. Tačiau buvo surasta konkrečių pavyzdžių, kurie įrodo, kad naudojant veiksmų registravimą ir vartotojo grafinės sąsajos analizę, galima realizuoti UML panaudojimo atvejų modelio generavimo metodą ir įrankį.

## 2. PANAUDOJIMO ATVEJŲ MODELIO IŠGAVIMO IŠ INTERNETINIŲ INFORMACINIŲ SISTEMŲ METODIKA IR ALGORITMAS

Šiame darbe pateikiamas panaudojimo atvejų modelio išgavimo iš internetinių informacinių sistemų procesas, susidedantis iš dviejų pagrindinių etapų:

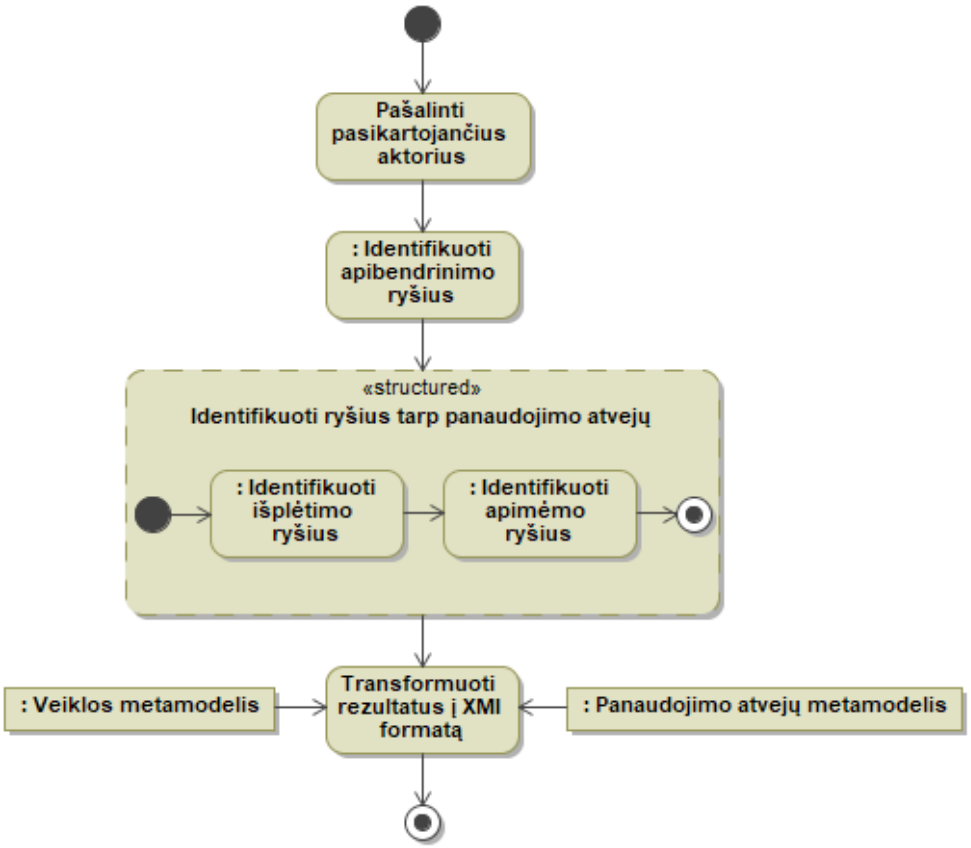
1. Darbo su analizuojama informacine sistema duomenų registravimo.
2. Gautų duomenų apdorojimo ir transformavimo.

Šie etapai aprašo informacijos išgavimo metodą, kuris atvaizduojamas 2.1 pav. Metodo metu naudotojas atveria norimą analizuoti sistemą, užregistruoja atliekamą rolę ir atliekamos veiklos pavadinimą. Toliau metodas užregistruoja naudotojo atliekamus veiksmus analizuojamoje sistemoje. Veiklos registravimas vykdomas, kol suregistruojami visi su naudotojo veikla susiję veiksmai. Veiksmų registravimas baigiasi tuo, kad naudotojas pažymi, jog veikla baigta. Tuomet naudotojas gali pasirinkti tęsti veiksmų registravimą, pradedant naujos veiklos registravimą ar baigti registravimo procesą. Veiklų registravimas kartojamas iki tol, kol užregistruojamos visos sistemoje egzistuojančios rolės arba kai užregistruota apimtis tenkina naudotoją. Kiekvienos rolės veiksmus gali vykdyti skirtingi asmenys ir vėliau pasinaudojant užregistruotų rezultatų eksportavimu ir jų apjungimu sukurti vieną didesnės apimties failą, reprezentuojantį sistemos veiklas. Sudarytas JSON formato failas apdorojamas antrajame etape, kurio metu sistema sugeneruojamas panaudojimo atvejų modelį.



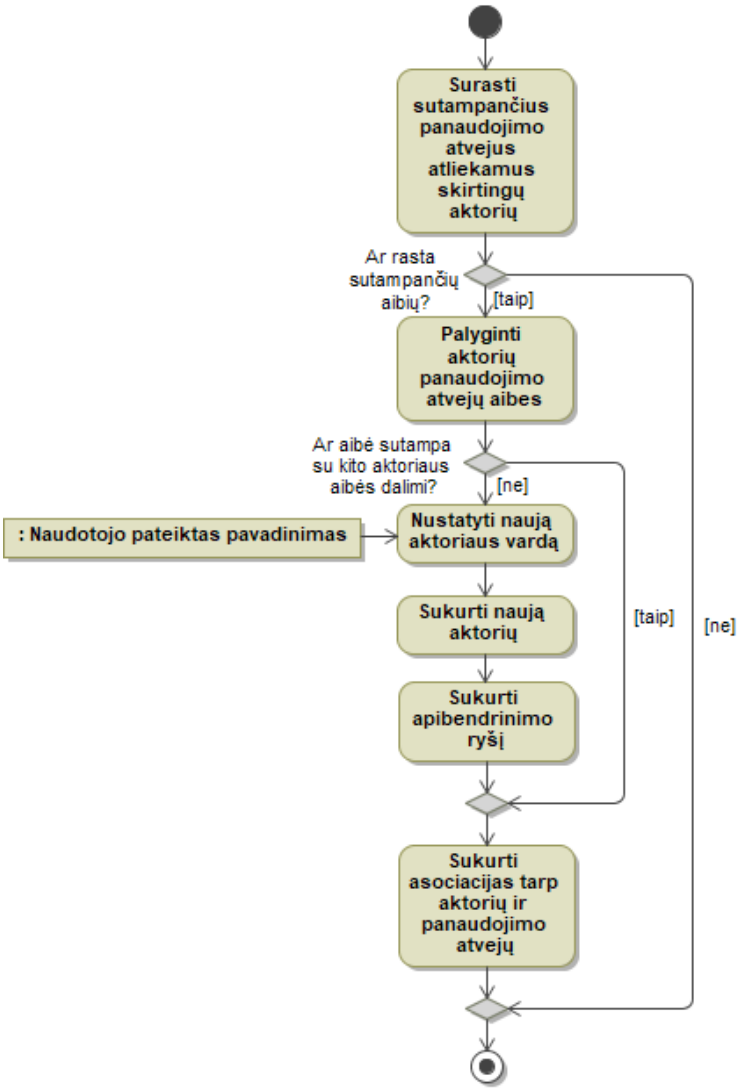

**2.1 pav.** Panaudojimo atvejų modelio generavimo algoritmo veiklos diagrama

Antrajame etape apdorojami pirmojo etapo metu gauti rezultatai siekiant juos transformuoti į XMI formatu atvaizduotą failą, kuris reprezentuoja panaudojimo atvejų modelį. Veiklos diagrama reprezentuojančią veiksmus reikalingus atlikti norint transformuoti naudotojo registruotus veiksmus į XMI formatu atvaizduotą panaudojimų atvejų modelį atvaizduoja 2.2 pav.



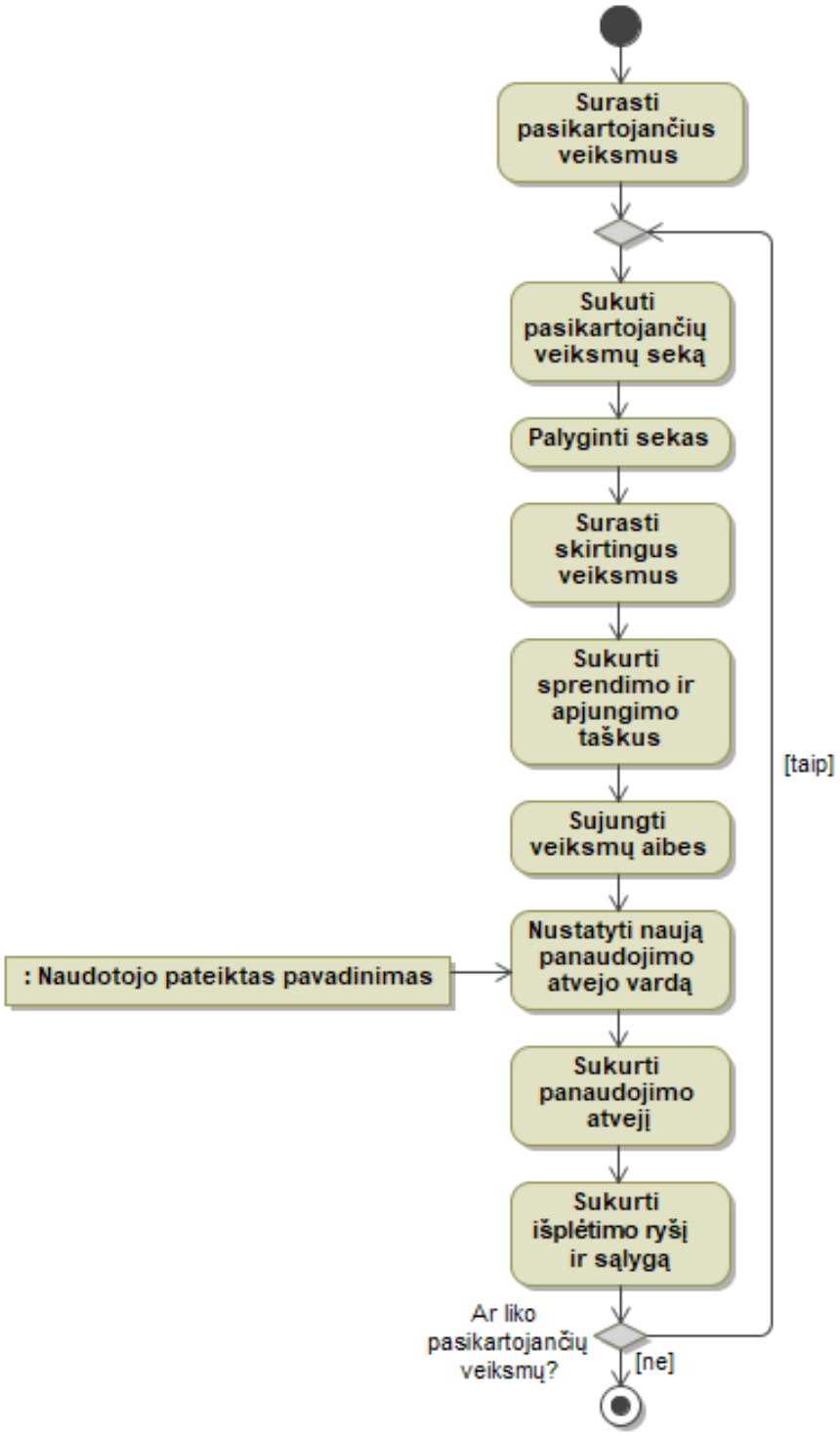

**2.2 pav.** Naudotojų veiksmų transformavimo į UML panaudojimo atvejų modelį veiklos diagrama

Sistema iš gauto rezultato failo pašalina pasikartojančius aktorius, kurie galėjo atsirasti dėl skirtingų sistemoje naudojamų dalykinės srities terminų ar dėl žmogiškosios klaidos. Pašalinus pasikartojimus tarp užregistruotų sistemos aktorių, nustatomi apibendrinimo ryšiai. Apibendrinimo ryšių nustatymas pradedamas skenuojant panaudojimo atvejus ir ieškant dviejų ar daugiau aktorių, kurie turi pilnai, ar dalinai sutampančius panaudojimo atvejus. Jei aktorių atliekami panaudojimo atvejai identiški - sukuriami panaudojimo atvejų ryšiai tarp aktorių. Visi pasikartojantys panaudojimo atvejai pašalinami. Jei nustatomas nepilnas panaudojimo atvejų sutapimas tarp aktorių, sukuriamas naujas apibendrinantis aktorius. Tarp aktorių, kuriems buvo rastas panaudojimo atvejų sutapimas ir naujai sukurto aktoriaus sukuriami apibendrinimo ryšiai. Detali diagrama, aprašanti apibendrinimo ryšių nustatymą, pateikiama 2.3 pav.



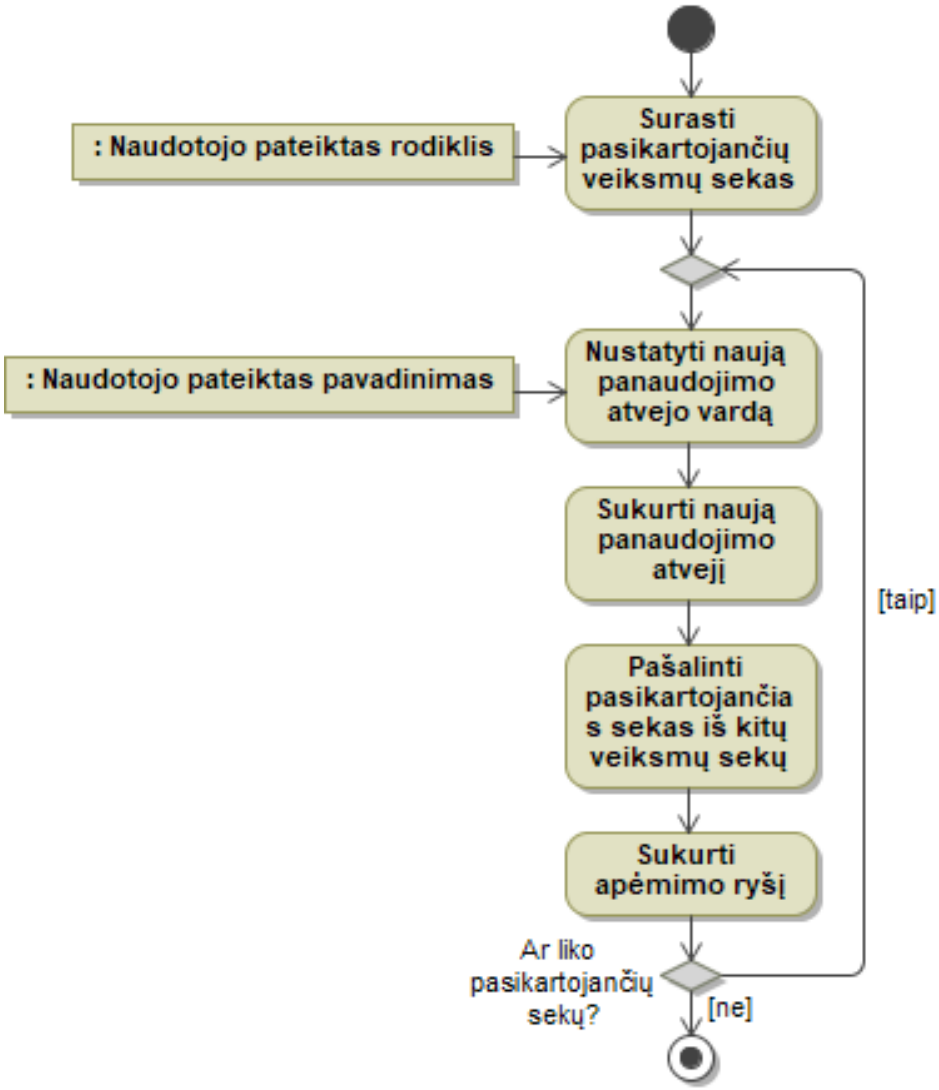

**2.3 pav.** Apibendrinimo ryšių nustatymo veiklos diagrama

Toliau vykdomi grupavimo veiksmai, kurie leidžia panaudojimų atvejų modelyje sukurti išplėtimo ir apėmimo ryšius tarp panaudojimo atvejų. Siūlomas sprendimas nustato išplėtimo ryšius taip papildydamas panaudojimo atvejų modelį papildomais ryšiais. Visi reikalingi veiksmai išplėtimo ryšių identifikavimo procese vaizduojami 2.4 pav. Išplėtimo ryšiai suteikia papildomą semantinę prasmę, jie aprašo alternatyvius panaudojimo atvejo scenarijaus veiksmus. Išplėtimo ryšių nustatymo procesas prasideda nuo surinktų analizuojamos sistemos veiksmų sekų patikrinimo, taip pat identifikuojami sekose dalinai pasikartojantys veiksmai. Iš nustatytų pasikartojančių veiksmų sekų, šias palyginus, ištraukiamos sutampančios sekų poros. Identifikuotos sekos pradžioje ir pabaigoje atitinkamai įterpiami sprendimo ir apjungimo elementai, o sekos sujungiamos, taip sukuriant naują veiklos diagramą. Kiekviena naujai sukurta sprendimo šaka gali būti aprašoma kaip naujas panaudojimo atvejis, kurio pavadinimas yra parenkamas naudotojo. Kito žingsnio metu tarp naujų ir originalių panaudojimo atvejų sukuriamas išplėtimo (angl. *extension*) ryšys, kuriam sugeneruojama išplėtimo sąlyga (angl. *extension point*).




**2.4 pav.** Išplėtimo ryšių nustatymo veiklos diagrama

Norint sumažinti pasikartojimų (angl. *redundancy*) skaičių veiklos diagramose, tarp panaudojimo atvejų gali būti naudojami apėmimo ryšiai (angl. *include*). Siūlomas metodas registruoja visus sistemoje atliekamus veiksmus, dėl to sudaroma labai plati pasikartojančių veiksmų aibė. Apėmimo ryšių nustatymas prasideda nuo visų veiksmų sekų nustatymo, kurių veiksmų skaičius didesnis, nei naudotojo nustatytas skaičius. Naudotojui nenurodžius šio skaičiaus jo reikšmė yra lygi dviems. Kiekvienai surastai sekai metodas sukuria naują panaudojimo atvejį, kurio pavadinimą pateikia naudotojas. Tuomet surastos sekos yra pašalinamos iš pirminių panaudojimo atvejų veiksmų sekų, vietoj jų įterpiamas naujai sukurtas panaudojimo atvejis. Galiausiai sukuriamas apėmimo ryšys tarp susijusių panaudojimo atvejų. Detalią veiklos diagramą aprašančią apėmimo ryšių nustatymą galima rasti 2.5 pav.




**2.5 pav.** Apėmimo ryšių nustatymo veiklos diagrama

XMI formato failo generavimo metu, gauti registracijos rezultatai, kartu su nustatytais ryšiais tarp aktorių ir panaudojimo atvejų transformuojami į diagramas. Visų pirma sukuriama panaudojimo atvejų diagrama, ir užpildoma aktoriais, panaudojimo atvejais ir ryšiais. Kiekvienam panaudojimo atvejui sukuriama jį detalizuojanti veiklos diagrama. Šioje veiklos diagramoje sukuriamos dvi juostos (angl. *swimlane*). Pirmosios juostos pavadinimas užpildomas aktoriumi, atliekančiu panaudojimo atvejį, antrosios juostos pavadinimui priskiriamas žodis „*System*“. Pirmojoje juostoje vaizduojami naudotojo atlikti veiksmai sistemoje, antrojoje – išvestiniai sistemos veiksmai. Naudotojo veiksmo pavadinimas sukuriamas laikantis tokios struktūros:

*Veiksmo pavadinimas = HTML elemento pavadinimas + <veiksmas atliktas su elementu> + <HTML elementą apibūdinanti informacija išgauta iš sistemos>;*

< > šioje struktūroje reiškia, kad dedamoji nėra privaloma.

Detalus HTML elementų ir galimų veiksmų su jais sąrašas pateikiamas 2.1 lentelėje. Šis sąrašas nėra baigtinis ir esant reikalui gali būti papildytas ar pakoreguotas, siekiant geriau pritaikyti įrankį analizuojamai sistemai.

Gautas XMI failas gali būti importuojamas į UML CASE įrankius. Šiuose įrankiuose sugeneruotas panaudojimo atvejų modelis gali būti tobulinamas, analizuojamas ir naudojamas, kaip reikalavimo ar kitų modelių dalis.

Šis naujai sukurtas modelis, taip pat buvo pristatytas konferencijoje IVUS 2018 ir jį galima rasti 9.1.2 priede.

**2.1 lentelė. Veiksmų pavadinimai**

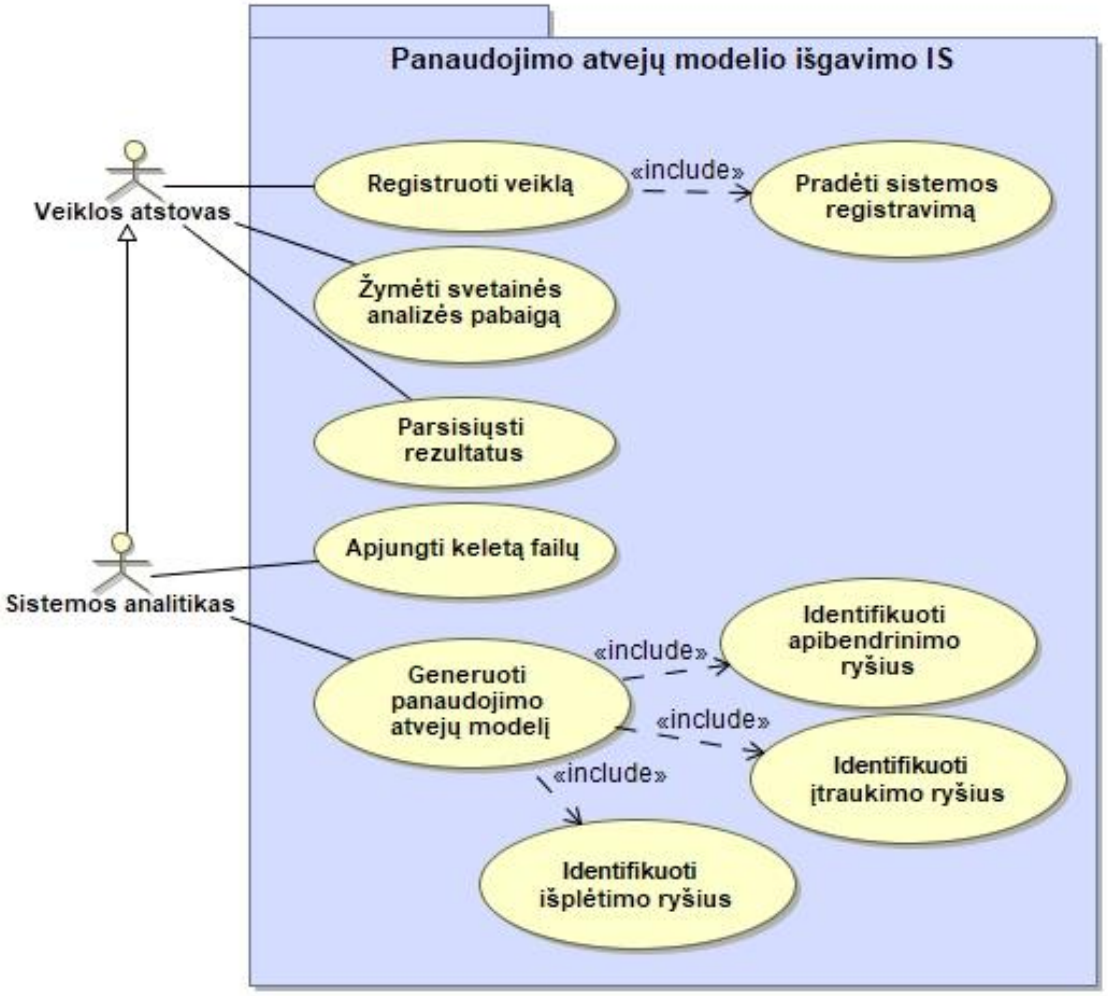
| Numeris | HTML elementas | Galimas atlikti veiksmas su elementu |
|---|---|---|
| 1 | Input | Change of input |
| 2 | Select | Change of input |
| 3 | A | Open |
| 4 | Form | Submit |
| 5 | Checkbox | Change of input |
| 6 | Text area | Change of input |
| 7 | Button | Click |
| 8 | Radio button | Change of input |
| 9 | Span | Click |
| 10 | Li | Click |

# 3. „WEB2UML“ SISTEMOS REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS

## 3.1. „WEB2UML“ sistemos reikalavimų specifikacija

### 3.1.1. „WEB2UML“ sistemos funkciniai reikalavimai

„WEB2UML“ sistemai taikomi funkciniai reikalavimai šiame skyriuje atvaizduojami panaudojimo atvejų diagrama (3.1 pav.) bei kiekvieną panaudojimo atvejį detalizuojančiomis veiklos diagramomis. Taip pat pateikiamos panaudojimo atvejų specifikacijos lentelėmis. Sistemoje egzistuoja dvi pagrindinės naudotojų rolės – veiklos atstovas ir sistemų analitikas. Sistemoje, nepriklausomai nuo to ar esi veiklos atstovas, ar sistemų analitikas gali registruoti sistemos veiklą. Norint užbaigti registravimą, reikia pažymėti sistemos analizės pabaigą, ir tuomet galima parsisiųsti rezultatus. Sistemos analitikas sistemoje gali apjungti keletą užregistruotų failų ir sugeneruoti panaudojimo atvejų modelį identifikuojant apibendrinimo (*generalization*), apėmimo (*include*) bei išplėtimo (*extend*) ryšius.

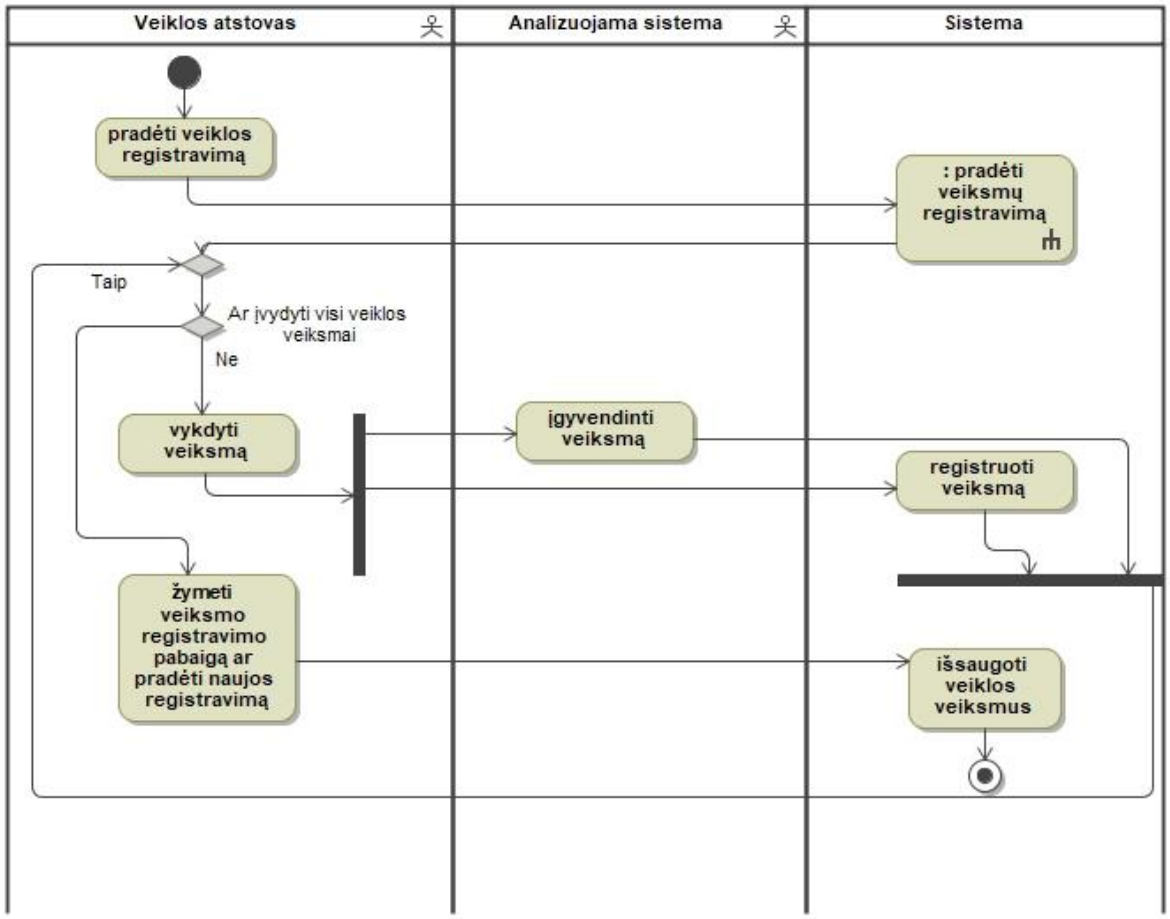



**3.1 pav.** UML modelių generavimo įrankio panaudojimo atvejų diagrama

Panaudojimo atvejo „Registruoti veiklą“ metu sistemos naudotojas užregistruoja veiksmus, atliekamus analizuojamoje sistemoje. 3.1 lentelėje pateikiama panaudojimo atvejo specifikacija, 3.2 pav. pateikta veiklos diagrama aprašo šio panaudojimo atvejo scenarijų. Prieš pradedant vykdyti šį panaudojimo atvejį privaloma įsidiegti įskiepį naršyklėje. Šio panaudojimo atvejo metu veiklos atstovas pateikia sistemoje atliekamo veiksmo pavadinimą ir jo atliekamą rolę. Toliau veiklos atstovas analizuojamoje sistemoje vykdo veiksmus. Taip kartojama, kol veiklos atstovas užregistruoja norimą kiekį veiklų sistemoje.

**3.1 lentelė. Panaudojimo atvejo „Registruoti veiklą“ specifikacija**

| | |
|---|---|
| **PA.** Registruoti veiklą | |
| **Tikslas.** Užregistruoti norimos veiklos parametrus skirtus analizei | |
| **Aprašymas.** Registruojami veiksmai analizuojamoje sistemoje | |
| **Prieš sąlyga** | Naudotojas įsidiegia įskiepį, atsidaro panelę ir pradedą įrašymą |
| **Aktorius** | Veiklos atstovas |
| **Sužadinimo sąlyga** | Sistemos naudotojas nuspaudžia registravimo pradžios mygtuką |
| **Apimami PA** | PA „Pradėti sistemos įrašymą“ |
| **Po sąlyga:** | Sistema išsaugo užregistruotos veiklos rezultatus. |

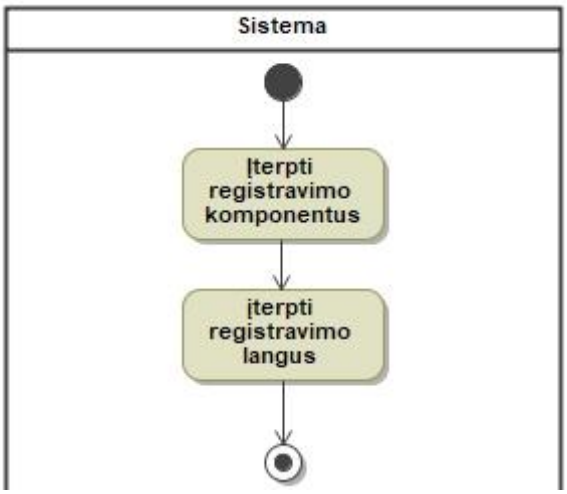



**3.2 pav.** Panaudojimo atvejo „Registruoti veiklą“ veiklos diagrama

„Pradėti sistemos registravimą“ metu sistema pasiruošia reikiamus resursus analizuojamos sistemos veiklos analizei. Žemiau esanti lentelė specifikuoja šį panaudojimo atvejį, kaip ir 3.3 pav. Šio panaudojimo atvejo metu į kiekvieną svetainėje esantį puslapį įterpiami sistemos veiksmų registravimo komponentai bei langai reikalingi įvesti sistemos naudotojo rolės ir veiklos pavadinimą.

**3.2 lentelė. Panaudojimo atvejo „Pradėti sistemos registravimą“ specifikacija**

| | |
|---|---|
| **PA.** Pradėti sistemos įrašymą | |
| **Tikslas.** Paruošti sistemą veiksmų registravimui | |
| **Aprašymas.** Į sistemą įtraukiami registravimo scenarijai | |
| **Prieš sąlyga** | Pradėta sistemos analizė |
| **Aktorius** | Veiklos atstovas |
| **Sužadinimo sąlyga** | Sistemos analizės pradžia |
| **Po sąlyga:** | Pradedama naujai registruoto aktoriaus veiksmų registracija. |

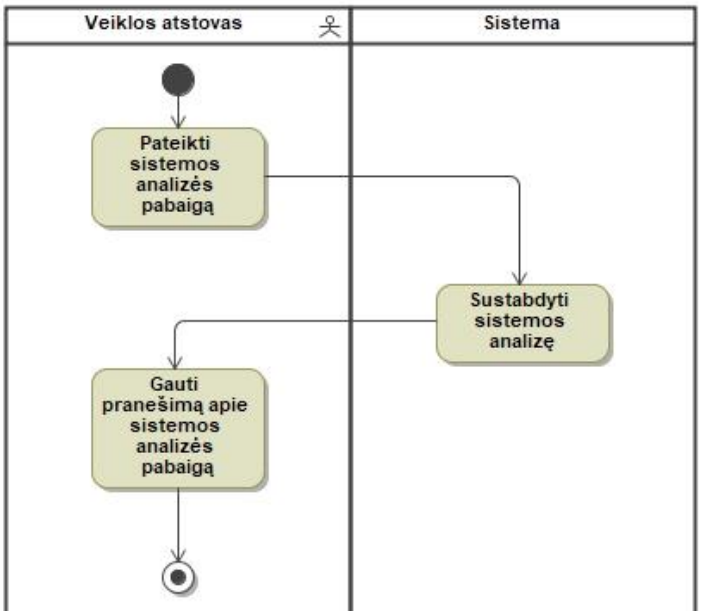



**3.3 pav.** Panaudojimo atvejo „Pradėti sistemos registravimą“ veiklos diagrama

Panaudojimo atvejis „Žymėti svetainės analizės pabaigą“ vykdomas tada kai nusprendžia, kad naudojimasis sistemoje yra pakankamai detaliai užregistruotas. Žemiau esanti lentelė apibūdina šį panaudojimo atvejį, o dar detaliau jį atvaizduoja 3.4 pav. vaizduojama veiklos diagrama. Šios veiklos metu veiklos atstovas pažymi, kad sistemos analizė yra baigta, sistema išsaugo rezultatus ir juos paruošia generavimo procesui.

**3.3 lentelė. Panaudojimo atvejo „Žymėti svetainės analizės pabaigą“ specifikacija**

| | |
|---|---|
| **PA.** Žymėti svetainės analizės pabaigą | |
| **Tikslas.** Sustabdyti analizuojamos svetainės veiksmų registravimą | |
| **Aprašymas.** Užregistruojami veiksmai, kuriuos rolės atstovas gali vykdyti analizuojamoje sistemoje. | |
| **Prieš sąlyga** | Sistemos registravimas buvo pradėtas |
| **Aktorius** | Dalykinės srities atstovas. |
| **Sužadinimo sąlyga** | Nuspaustas mygtukas *stop recording* |
| **Po sąlyga:** | Sistema nustoja registruoti veiksmus |



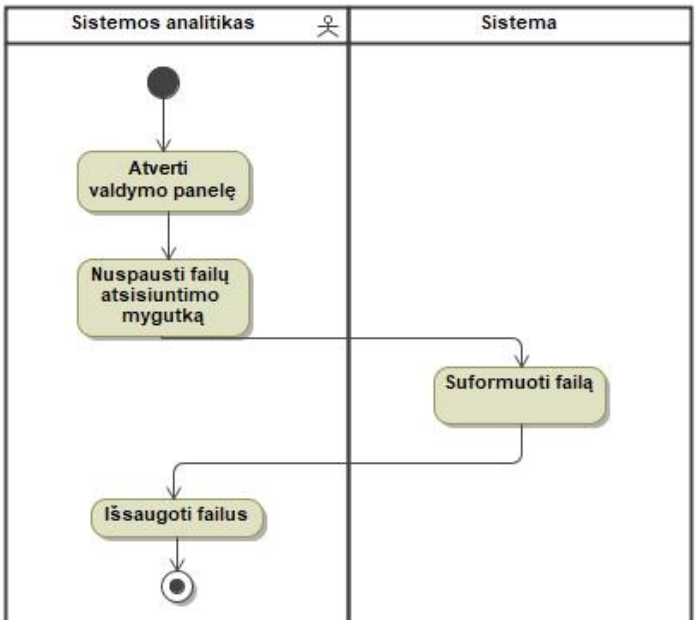

**3.4** pav. Panaudojimo atvejo „Žymėti svetainės analizės pabaigą“ veiklos diagrama

Panaudojimo atvejis parsisiųsti rezultatus vykdomas tada, kai norima išsaugoti arba perduoti užregistruotus analizuojamos sistemos veikimo rezultatus. Žemiau esanti lentelė ir 3.5 pav. vaizduojama diagrama detaliau specifikuoja šį panaudojimo atvejį. Šio panaudojimo atvejo metu

sistemos analitikas nuspaudžia mygtuką parsisiųsti failus ir sistema atsiunčia užregistruotą procesą JSON failą į jo kompiuterį.

**3.4 lentelė. Panaudojimo atvejo „Parsisiųsti rezultatus“ specifikacija**

| | |
|---|---|
| **PA.** Parsisiųsti rezultatus | |
| **Tikslas.** Išsaugoti duomenų failą savo kompiuteryje saugojimui ar duomenų apsikeitimui. | |
| **Aprašymas.** Šio PA metu vykdomi veiksmai su jau sukurtais aktoriais, arba jis sukūrimas. | |
| **Prieš sąlyga** | Naudotojas yra sistemos analitikas. |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Naudotojas pasirenka failų atsisiuntimo valdymo menių punktą. |
| **Po sąlyga:** | Išsaugojami veiksmai atlikti su aktoriais. |

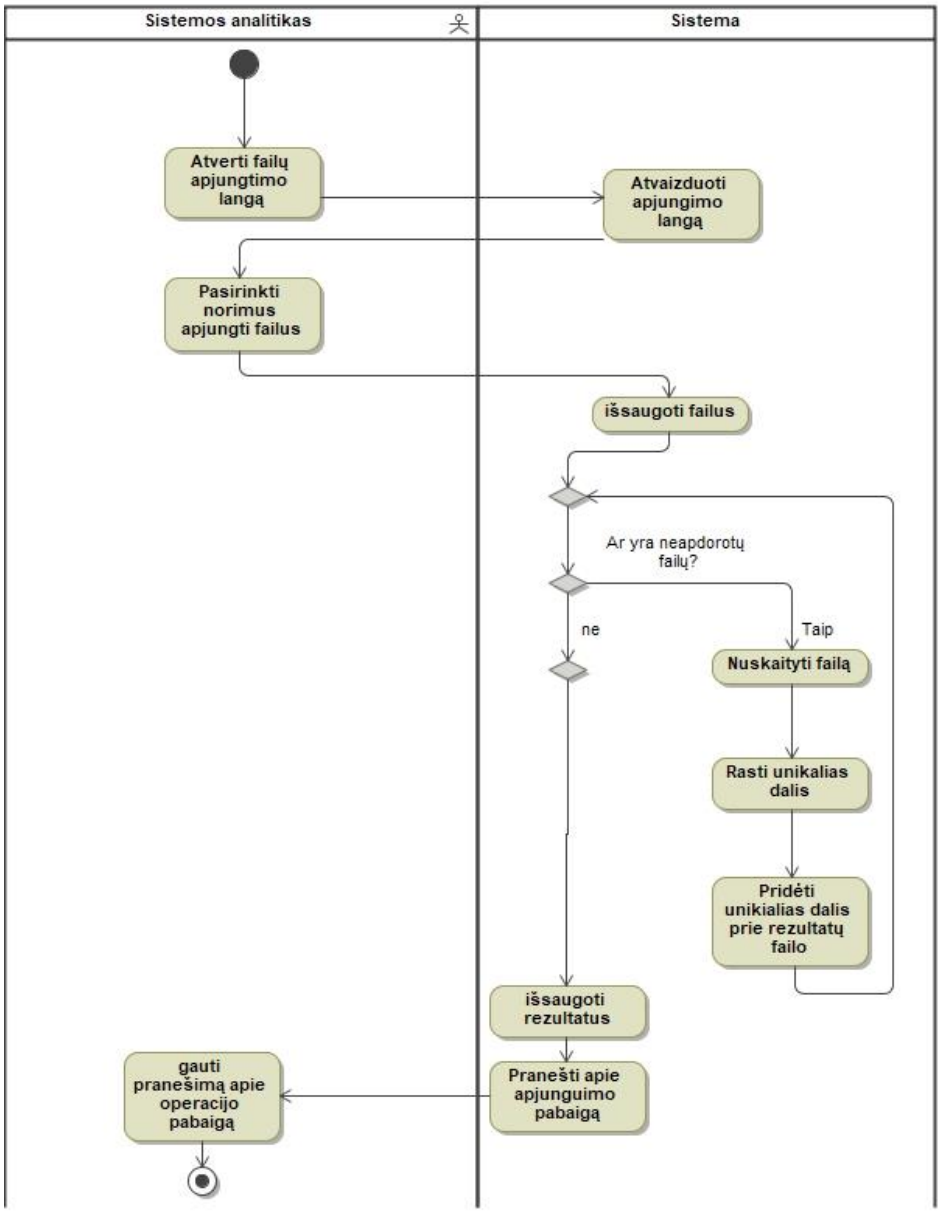



**3.5 pav.** Panaudojimo atvejo „Parsisiųsti rezultatus“ veiklos diagrama

Įgyvendinus panaudojimo atvejį „Parsisiųsti rezultatus“, gautus rezultatus galima perduoti kitam asmeniui ir juos apjungti pasinaudojant panaudojimo atveju „Apjungti keletą failų“, kurio veikimas aprašomas žemiau esančia lentele ir 3.6 pav. Kelių failų apjungimas vykdomas nustatant bendras failų dalis ir prie jų pridedant failų dalis, kurios skiriasi. Po du failus vienu metu galime sujungti tokį kiekį failų kiek tik pageidauja sistemos analitikas.

**3.5 lentelė. Panaudojimo atvejo „Apjungti keletą failų“ specifikacija**

| | |
|---|---|
| **PA.** Apjungti keletą failų | |
| **Tikslas.** Sujungti kelis failus iš saugotus per daug analizės kartų ar užregistruotus skirtingų naudotojų. | |
| **Aprašymas.** Atliekamas gautų registravimo duomenų apjungimas į vieną bendrą failą | |
| **Prieš sąlyga** | Naudotojas yra sistemos analitikas. |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Naudotojas pasirenka rezultatų apjungimo meniu punktą |
| **Po sąlyga:** | Apjungiami du ar daugiau naudojimusi sistema rezultatų failų. |

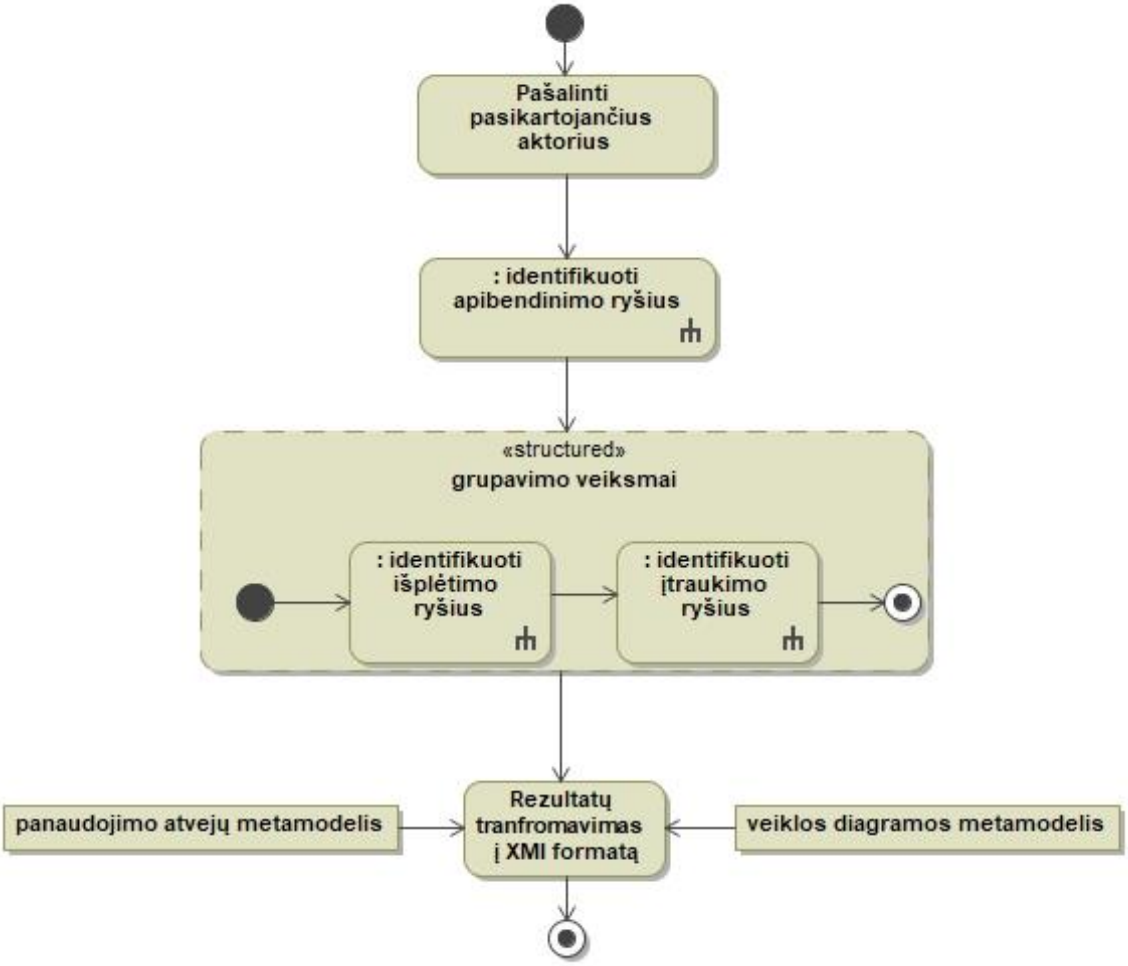



**3.6 pav.** Panaudojimo atvejo „Apjungti keletą failų“ veiklos diagrama

Panaudojimo atvejis „Generuoti panaudojimo atvejų modelį“ aprašo gautų registravimo rezultatų transformavimo į vieną failą procesą, kuris atliekamas prieš generuojant panaudojimo atvejų modelį. Detaliau šis panaudojimo atvejis specifikuojamas 3.6 lentelėje ir 3.7 pav. Generavimo metu pašalinami pasikartojantys aktoriai, identifikuojami apibendrini ryšiai, vykdomi grupavimo veiksmai, bei gauti rezultatai transformuojami į XMI formato failą.

**3.6 lentelė. Panaudojimo atvejo „Generuoti panaudojimo atvejų modelį“ specifikacija**

| | |
|---|---|
| **PA.** Generuoti panaudojimo atvejų modelį | |
| **Tikslas.** Inicijuoti panaudos atvejų generavimą | |
| **Aprašymas.** Šio PA metu sugeneruojamos panaudos atvejų diagramos ir atsiunčiamas naudotojui. | |
| **Prieš sąlyga** | Naudotojas yra sistemos analitikas. |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Naudotojas pasirenka gauti panaudos atvejų diagramas. |
| **Apimami PA** | PA „Identifikuoti apibendrinimo ryšius“, PA „ Identifikuoti išplėtimo ryšius“, PA „ Identifikuoti įtraukimo ryšius“ |
| **Po sąlyga:** | Sugeneruojamas ir parsiunčiamas XMI formato failas. |

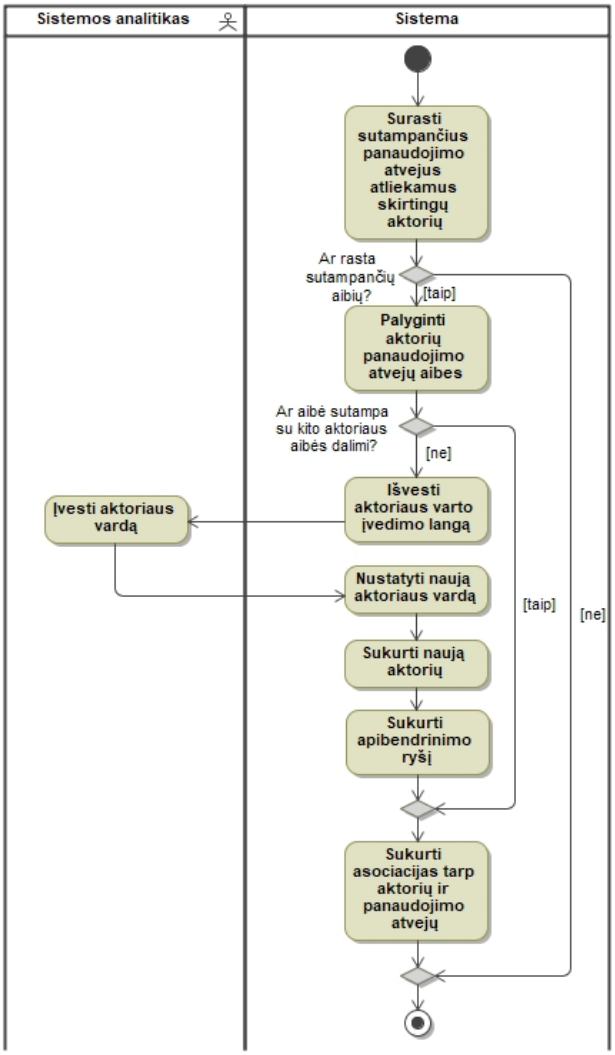



**3.7 pav.** Panaudojimo atvejo „Generuoti panaudojimo atvejų modelį“ veiklos diagrama

Panaudojimo atvejis „Identifikuoti apibendrinimo ryšius“ aprašo, kokie veiksmai turi būti atlikti, nustatant ir sukuriant visus reikiamus elementus detalizuojančius sistemą apibendrinimo ryšiais. Detaliau šis panaudojimo atvejis aprašomas lentelėje esančioje žemiau ir 3.8 pav. Šio panaudojimo atvejo metu sistema nustato peržiūri visus užregistruotus aktorius ieškant tokių aktorių, kurių sutampančių veiksmų skaičius būtų didesnis nei sistemoje nurodyta N skaičiaus riba. Jei tokių aktorių randama ir jų panaudojimo atvejai persidengia tarpusavyje pilnai, tiesiog sukuriami apibendrinimo ryšiai tarp jų ir iš visų išskyrus vieno aktorių pašalinami ryšiai su panaudojimo atvejais. Jei nustatomas nepilnas sutapimas tarp dviejų aktorių ir sutampančių panaudojimo atvejų skaičius yra didesnis nei N, tai N panaudojimo atvejų iškeliama ir susiejama su nauju aktoriumi, kitus panaudojimo atvejus paliekant sujungtus asociacijos ryšiais su originaliai užregistruotais aktoriais.

**3.7 lentelė. Panaudojimo atvejo „Identifikuoti apibendrinimo ryšius“ specifikacija**

| | |
|---|---|
| **PA.** Identifikuoti apibendrinimo ryšius | |
| **Tikslas.** Sugeneruoti apibendrinimo ryšius ir jiems atvaizduoti reikalingus papildomus elementus. | |
| **Aprašymas.** Transformuojant užregistruotą informaciją apie sistemos veiklą nustatyti ir suformuoti panaudojimo atvejus ir ryšius skirtus atvaizduoti panaudojimo atvejų diagramoje egzistuojantiems apibendrinimo ryšiams. | |
| **Prieš sąlyga** | |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Baigiama vykdyti veikla, skirta pašalinti pasikartojančius aktorius. |
| **Po sąlyga:** | Tarpiniuose rezultatuose išsaugoma informacija apie apibendrinimo ryšius |

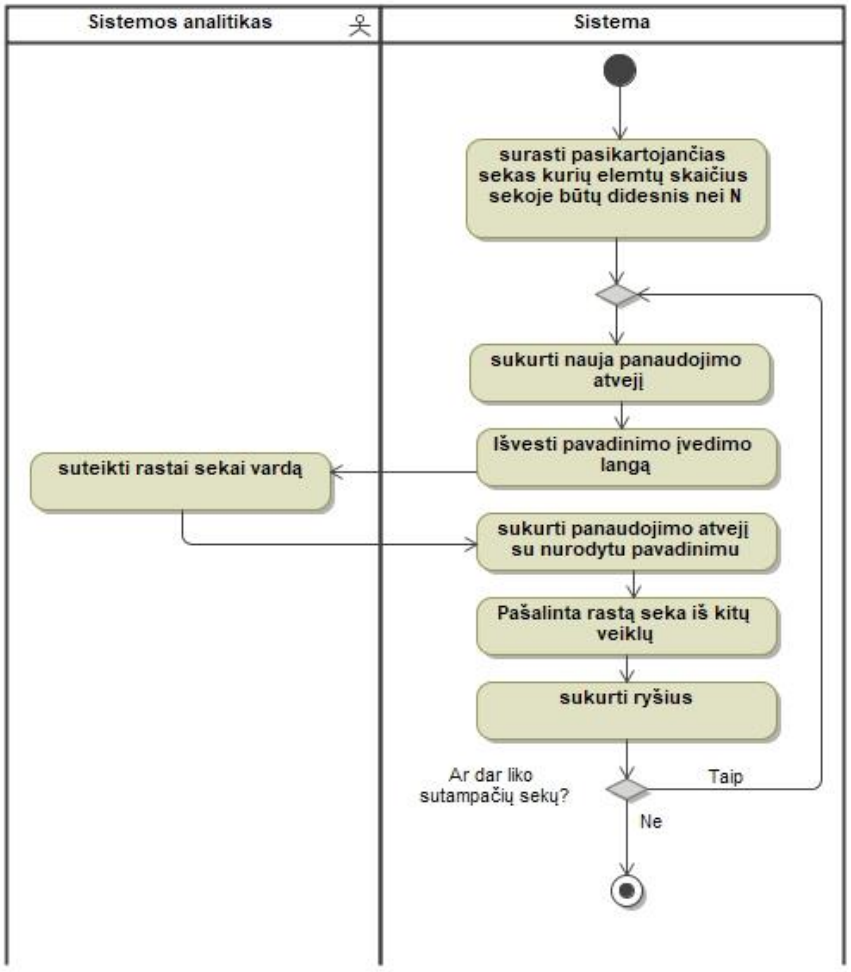



**3.8 pav.** Panaudojimo atvejo „Identifikuoti apibendrinimo ryšius“ veiklos diagrama

Panaudojimo atvejis „Identifikuoti įtraukimo ryšius“ aprašo, kaip sistemoje turi būti įgyvendinta registravimo rezultato analizė ir transformavimas, siekiant nustatyti įtraukimo ryšius tarp užregistruotų panaudojimo atvejų ar sukurti naujus panaudojimo atvejus, kurie praplėstų panaudojimo atvejų modelio detalumą. Išsamiau šis panaudojimo atvejis specifikuojamas žemiau esančioje lentelėje ir 3.9 pav. Įtraukimo (angl. *include*) ryšiai nustatomi ieškant pasikartojančių sekų tarp veiklas sistemoje aprašančių veiksmų sekų, kurios užregistruotos tik vieną kartą. Jei pavyksta rasti tokią seką, kuri yra ilgesnė nei N veiksmų, ji iškeliamą į naują panaudojimo atvejį ir pirminiai panaudojimo atvejai su naujai sukurtu panaudojimo atveju sujungiami <<*include*>> ryšiu. Sistemos naudotojas suteikia pavadinimą naujai sukurtam panaudojimo atvejui. Taip kartojama su visomis sekomis, kurios yra ilgesnės nei N ilgio ir pasikartoja bent dviejose veiklose.

**3.8 lentelė. Panaudojimo atvejo „Identifikuoti įtraukimo ryšius“ specifikacija**

| | |
|---|---|
| **PA.** Identifikuoti įtraukimo ryšius | |
| **Tikslas.** Sugeneruoti įtraukimo ryšius ir jiems atvaizduoti reikalingus papildomus elementus. | |
| **Aprašymas.** Transformuojant užregistruotą informaciją apie sistemos veiklą nustatyti ir suformuoti panaudojimo atvejus ir ryšius skirtus atvaizduoti panaudojimo atvejų diagramoje egzistuojantiems įtraukimo ryšiams. | |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Baigiama veikla „Identifikuoti apibendrinimo ryšius“ |
| **Po sąlyga:** | Į tarpinius rezultatus įrašyta informacija apie įtraukimo ryšius ir su jais susijusius panaudojimo atvejus. |

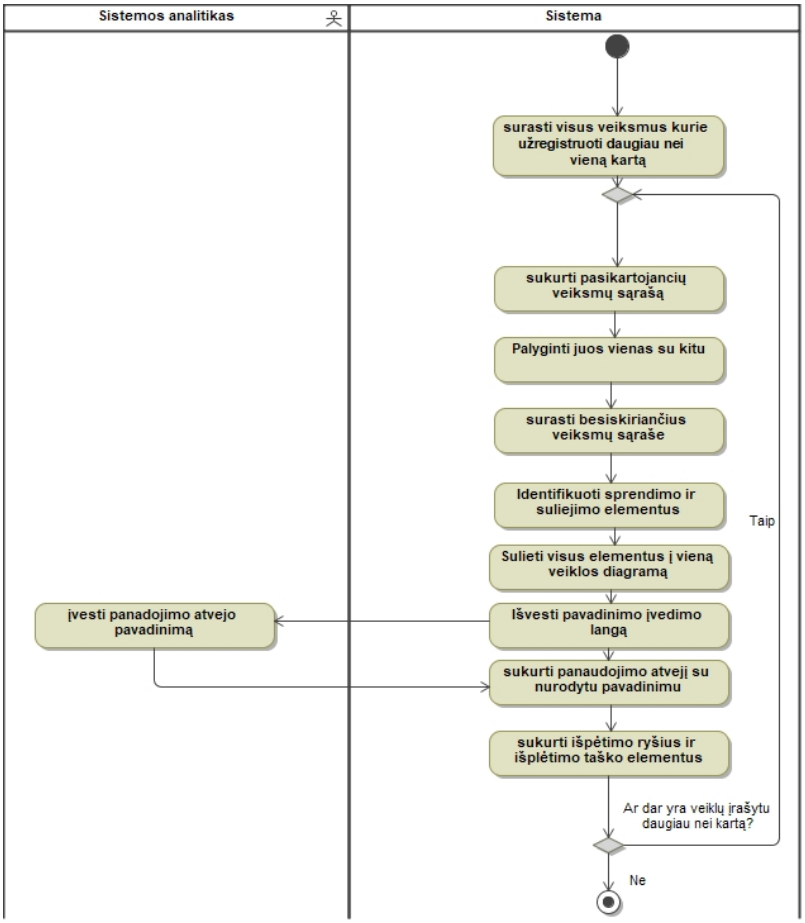



**3.9 pav.** Panaudojimo atvejo „Identifikuoti įtraukimo ryšius“ veiklos diagrama.

Panaudojimo atvejis „identifikuoti išplėtimo ryšius“ ,kaip ir panaudojimo atvejis „identifikuoti įtraukimo ryšius“ aprašo sistemos veiksmus reikalingus nustatyti ir sukurti reikiamus elementus bei ryšius siekiant kiek įmanoma išsamiau atvaizduoti panaudojimo atvejų modelį. Žemiau esanti lentelė ir 3.10 pav. detaliau specifikuoja šį panaudojimo atvejį. Išplėtimo ryšiai nustatomi analizuojant veiklas užregistruotas sistemoje daugiau nei vieną kartą. Nustatymas pradedamas surandant veiklas sistemoje užregistruotas tuo pat pavadinimu. Tarp šių veiklų surandamos pasikartojančios dalys, jos sujungiamos, o veiklų sekų dalys kurios skiriasi sujungiamos veiklos diagramoje sukuriant alternatyvų kelią. Jei nėra sutampančių dalių visos dalys tampa alternatyviaisiais keliais. Kiekvienam alternatyviam keliui sukuriamas naujas panaudojimo atvejis, kuris su pagrindiniu panaudojimo atveju, iš kurio buvo išgauta seka, sujungiamas <<*extend*>> ryšiu. Kadangi panaudojimai kuriami naudojantis tik sistemos naudotojo ir sistemoje egzistuojančia informacija išplėtimo taškas (angl. *extension point*) nesukuriamas.

**3.9 lentelė. Panaudojimo atvejo „Identifikuoti išplėtimo ryšius“ specifikacija**

| | |
|---|---|
| **PA.** Identifikuoti išplėtimo ryšius | |
| **Tikslas.** Sugeneruoti išplėtimo ryšius ir jiems atvaizduoti reikalingus papildomus elementus. | |
| **Aprašymas.** Transformuojant užregistruotą informaciją apie sistemos veiklą nustatyti ir suformuoti panaudojimo atvejus ir ryšius skirtus išplėtimo ryšių atvaizdavimui panaudojimo atvejų diagramoje | |
| **Aktorius** | Sistemos analitikas. |
| **Sužadinimo sąlyga** | Naudotojas pasirenka peržiūrėti panaudojimo atvejų tarpinius rezultatus |
| **Po sąlyga:** | Sugeneruojami ir parsiunčiami XMI failai. |

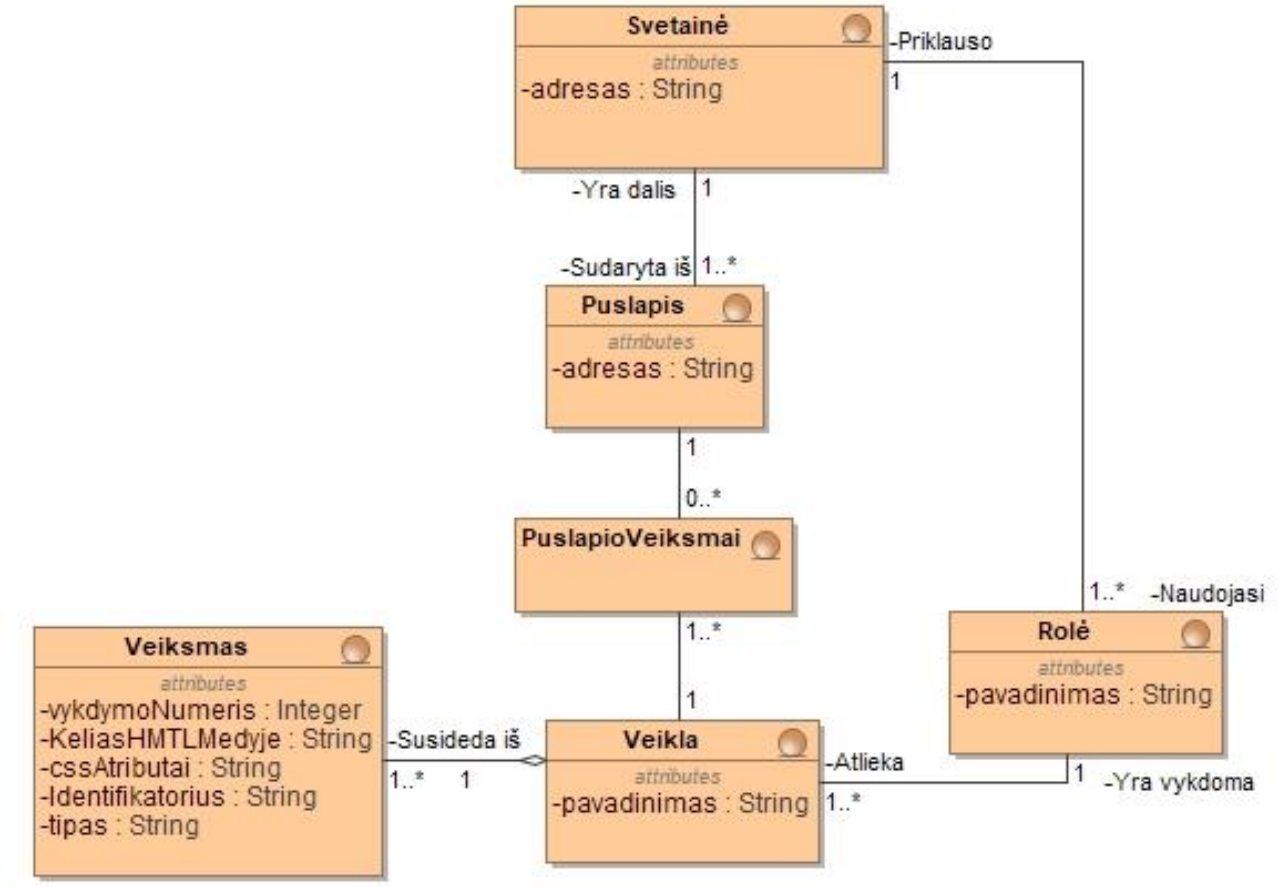



**3.10 pav.** Panaudojimo atvejo „Identifikuoti išplėtimo ryšius“ veiklos diagrama

### 3.1.2. „WEB2UML“ dalykinės srities modelis

„WEB2UML“ įrankiui reikia apibrėžti struktūrą, skirtą užregistruotiems analizuojamoje sistemoje veiksmams aprašyti. Taip pat svarbu detalizuoti struktūrą sugeneruotų diagramų saugojimui ir apsibrėžti elementus naudojamus sugeneruotuose diagramose.

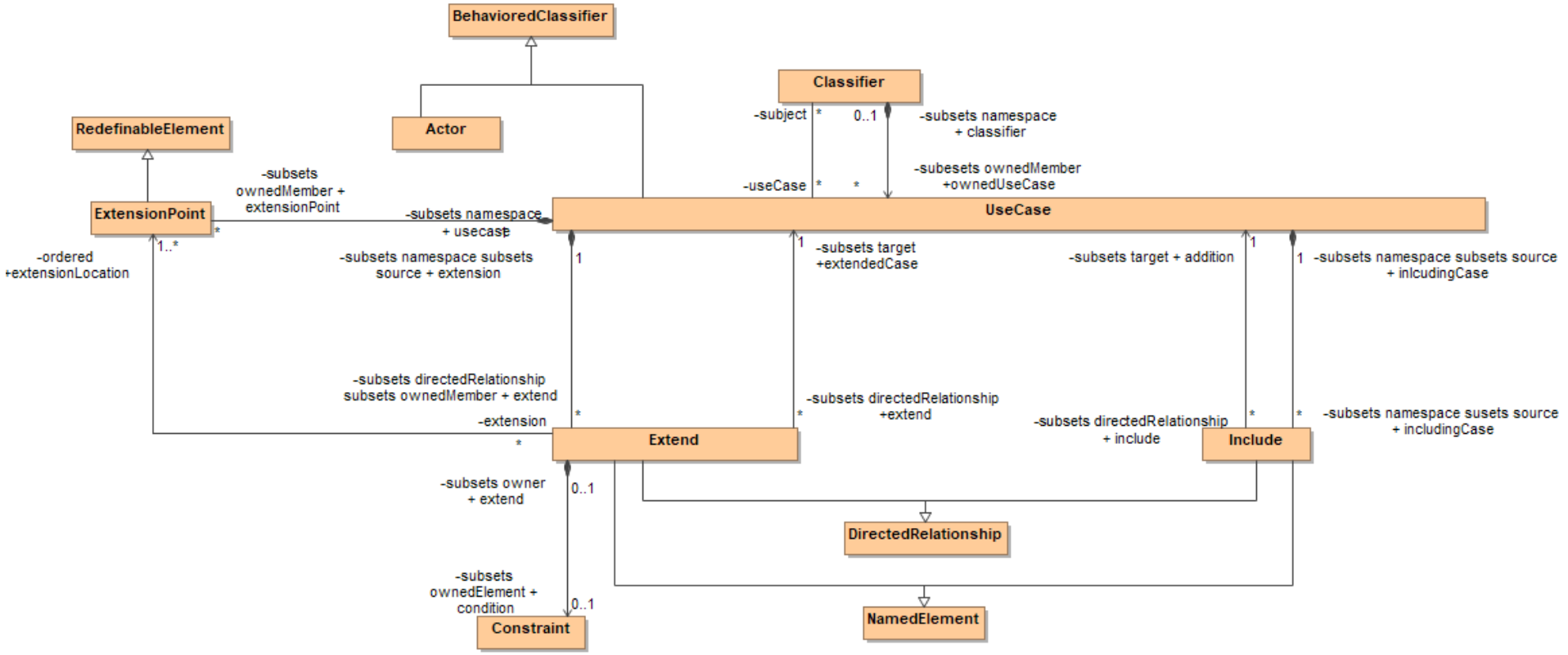
3.11 pav. pateikiamos sistemoje naudojamos esybės su joms priskirtais atributais ir struktūra. Šios esybės nurodo kokios struktūros turi būti failas, kuris kuriamas registruojant naudotojo veiksmus analizuojamoje sistemoje. Šiame faile registruojama informacija apie svetainę, svetainės adresas svetainėje egzistuojančio puslapio adresas ir sistemoje užregistruota naudojo rolė ar rolės. Viena naudotojo atliekama veikla gali tęstis per daugelį svetainių ir tose svetainėse esančių puslapių. Kiekvieną veiklą sudaro daug veiksmų




**3.11 pav.** „WEB2UML“ įrankio dalykinės srities esybių klasių diagrama.

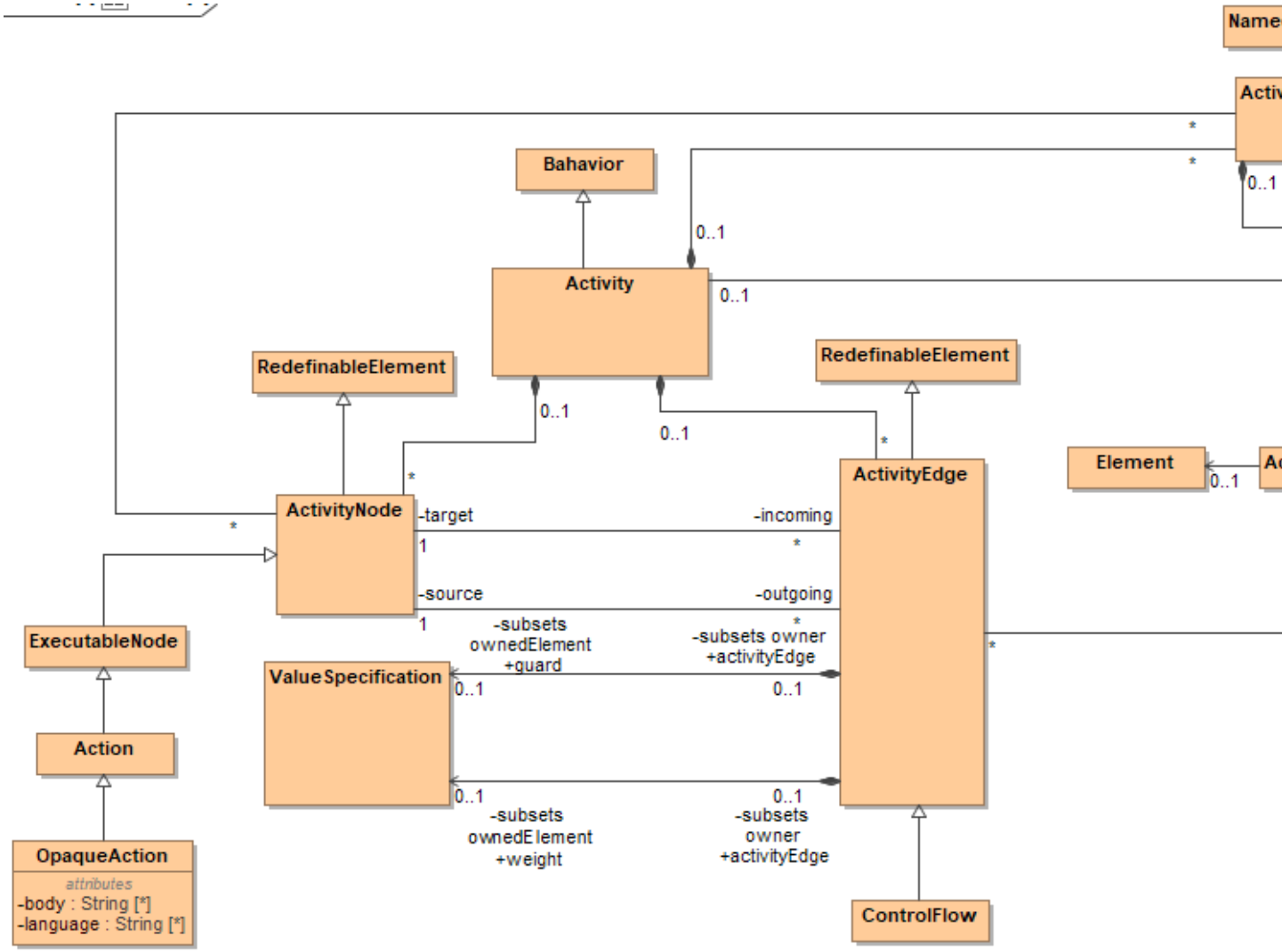
3.12 pav. pateikiamas UML 2.5 meta modelio fragmentas, kuris aprašo panaudojimo atvejų diagramos elementų struktūrą, kuri bus naudojama panaudojimo atvejų diagramai generuoti. 3.13 pav. ir 3.14 pav. pateikiami modifikuoti UML 2.5 versijos meta modelio fragmentai, detalizuojantys veiklos diagramų sudėtį it svarbiausius elementus, naudojamus šiame darbe generuojamose veiklos diagramose.

Pagrindiniai išrinkti panaudojimo atvejų diagramos elementai, tai panaudojimo atvejai, aktoriai, apėmimo, išplėtimo ir apibendrinimo ryšiai. Aktoriai su panaudojimo atvejais sujungiami asociacijos ryšiais.

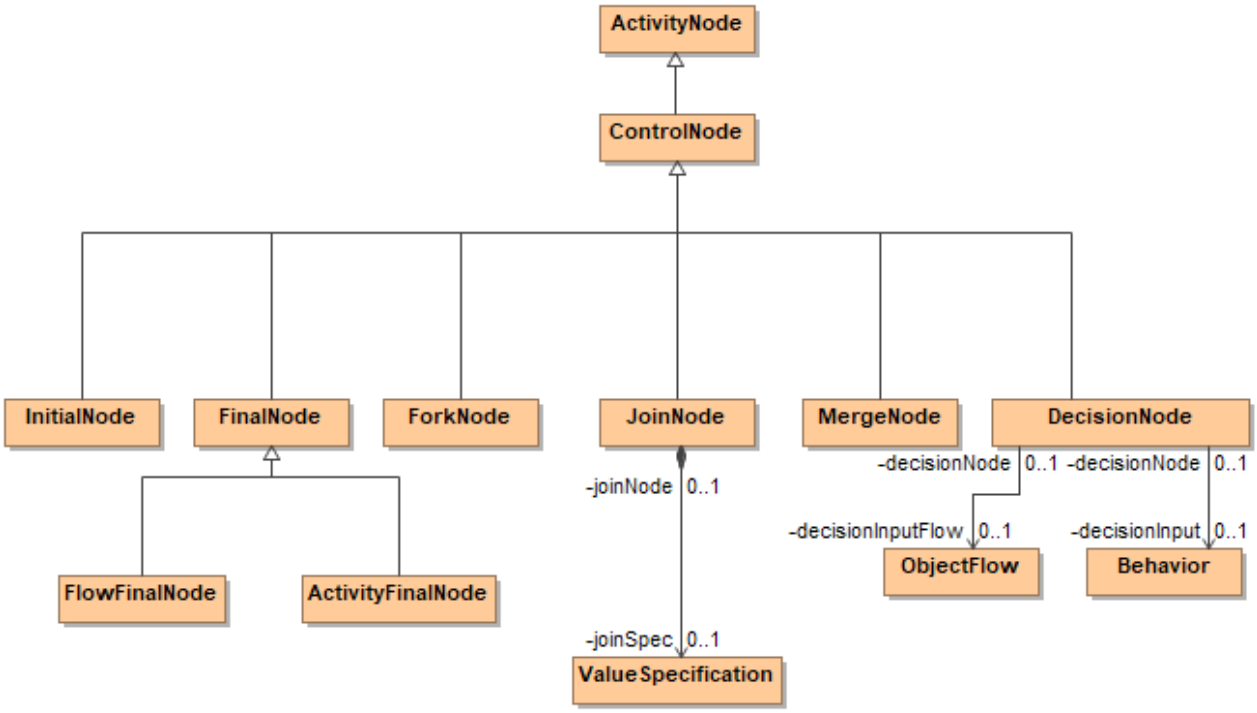



**3.12 pav.** Panaudojimo atvejų diagramos meta modelio fragmentas

Veiklos diagramų elementai susiaurinti iki pagrindinių veiksmų, valdymo srautų, veiklų juostų (*swimlane*) 3.13 pav. Taip veiklos diagramose kuriami valdymo mazgų elementai (3.14 pav.) tokie, kaip: pradžios ir pabaigos mazgai, sprendimo ir suliejimo mazgai, lygia gretinimo bei apjungimo mazgai. Šie valdymo mazgai įgalina aprašyti sudėtingesnius atvejus veiklos diagramose, bei pabrėžti, panaudojimo atvejų diagramose egzistuojančius <<*extend*>> ryšius.




**3.13 pav.** Veiklos diagramos meta modelio fragmentas




**3.14 pav.** Veiklos diagramos valdymo mazgų meta modelio fragmentas

### 3.1.3. „WEB2UML“ nefunkciniai reikalavimai

Sistemos nefunkciniai reikalavimai pirmiausia siejasi su priemonėmis veiksmų registravimui analizuojamoje sistemoje. Būtina apsibrėžti, kokius reikalavimus turi atitikti naršyklė, kurioje vyksta analizuojamos sistemos veiksmų registravimas. Taip pat svarbu nustatyti reikalavimus algoritmo naudojamų ir generuojamų duomenų formatams. Nefunkciniai reikalavimai detaliau aprašyti 3.10 lentelėje.

**3.10 lentelė. Nefunkciniai reikalavimai „WEB2UML“ sistemai**

| *Nr.* | **Reikalavimo tipas** | **Reikalavimas** | **Apibūdinimas** |
|---|---|---|---|
| *1* | Reikalavimas veikimo aplinkai | Sistema turi veikti Chromium variklį naudojančiose naršyklėse | Sistema turi veikti ant visų naršyklių naudojančių Chromium variklį, kurios palaiko naršyklės plėtinius. |
| *2* | Reikalavimas sistemos duomenų formatui | Sistema veiksmų registravimo rezultatus turi pateikti JSON formatu | Siekiant lengvesnio duomenų apsikeitimo tarp įrankių ir paprastesnio duomenų apdorojimo, jei tektų programą integruoti su kitomis, veiksmų registravimo rezultatai privalo būti atvaizduojami JSON formatu. |
| *3* | Reikalavimas sistemos duomenų formatui | Modelio rezultatus atvaizduoti XMI formatu | Atsižvelgiant į tai, kad XMI standartas kompanijos OMG yra patvirtintas, kaip standartinis duomenų apsikeitimo formatas tarp UML diagramas palaikančių įrankių, rezultatai privalo būti atvaizduojami XMI formatu siekiant palaikyti kuo daugiau UML diagramas galinčių atvaizduoti įrankių. |
| *4* | Reikalavimas sistemos veikimui | Veiksmų registravimas privalo netrikdyti analizuojamos sistemos veikimo | Siekiant užtikrinti tikslų veiksmų registravimą, naujai kuriama sistema privalo leisti nagrinėjamai sistemai funkcionuoti taip pat, kaip ir prieš paleidžiant veiksmų registravimą. |

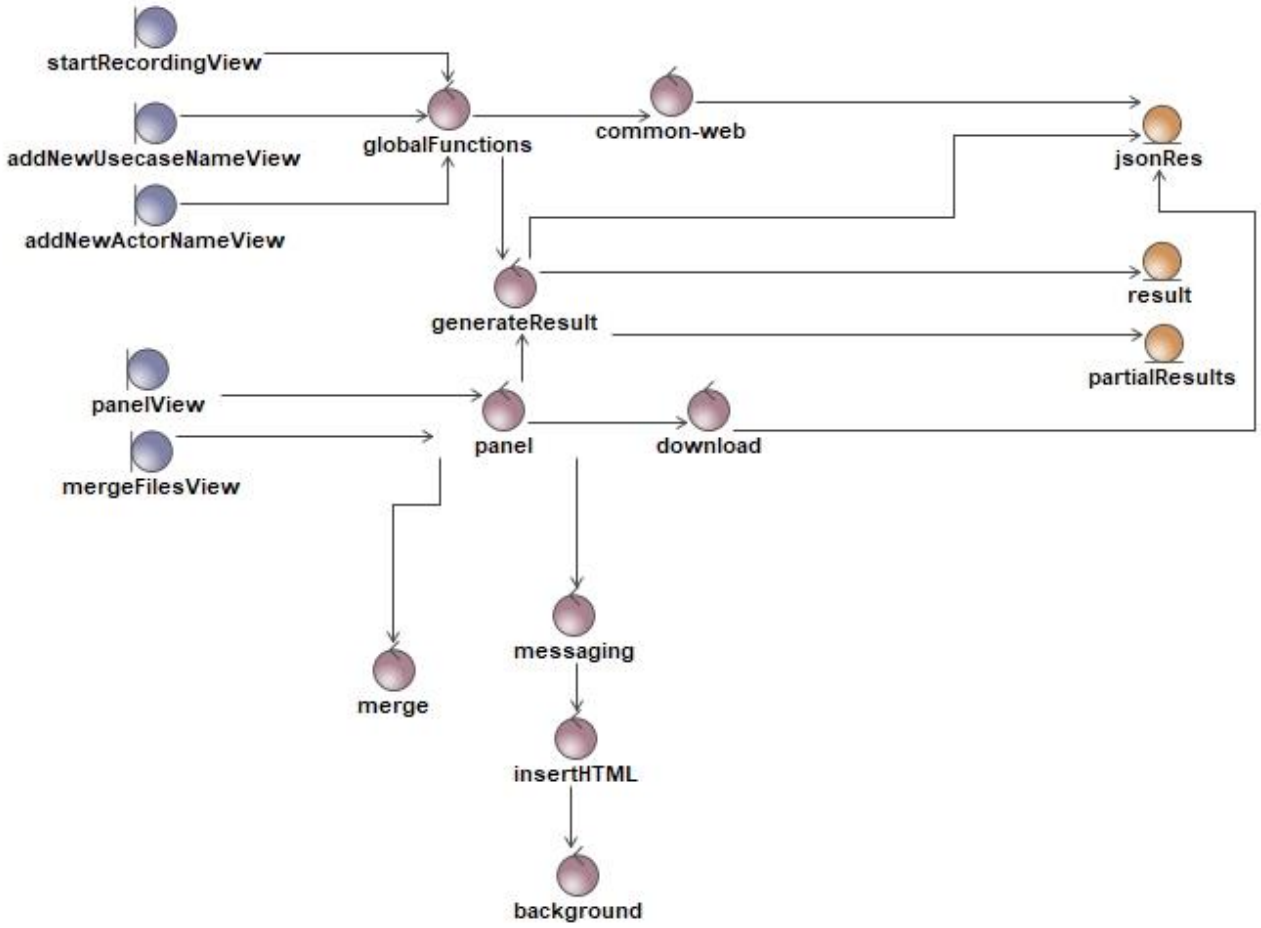
### 3.1.4. Reikalavimų apibendrinimas

Apibendrinant sistemos reikalavimus - naujai kuriama sistema turi leisti užregistruoti veiklą vykdomą internetinėse sistemose bei sugeneruoti panaudojimo atvejų modelį transformuojant šio registravimo rezultatus. Sistemos registravimo komponentų rezultatai turi grąžinti rezultatus atitinkančius 3.11 pav. atvaizduojamą struktūrą bei sugeneruoti diagramas turinčius elementus, kurie aprašyti 3.12 pav. , 3.13 pav. , 3.14 pav. Naujai sukurtas įrankis turi veikti, kaip *Chromium* varikliui pritaikytas įskiepis, tarpinius rezultatus saugoti JSON formatu, o generavimo rezultatą pateikti XMI formatu.

## 3.2. „WEB2UML“ sistemos projektas

### 3.2.1. Sistemos klasių modelis

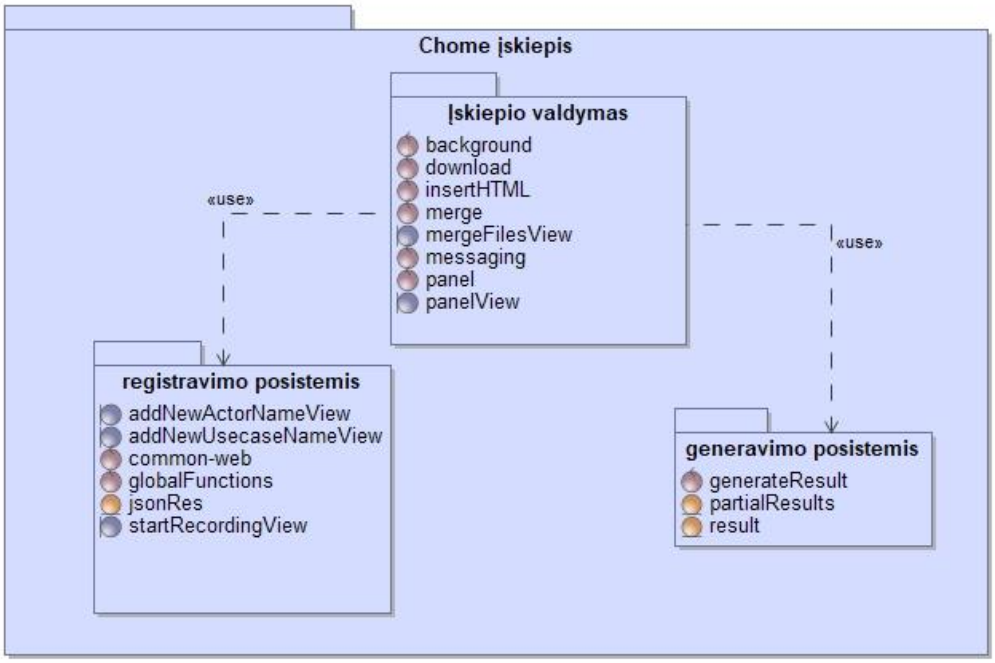
Panaudojimo atveju modelio generavimo sistemoje naudojamų klasių modelis atvaizduojamas 3.15 pav. Klasių modelyje matome, kad klasės skirstomos į grafinės naudotojo sąsajos klases, klases skirtas valdymui ir duomenų failus. Pagrindinės grafinės naudotojo sąsajos klasės, skirtos darbui su duomenų įvedimu į vartotojui išvedamus langus. Valdymo klasės galime suskirstyti į įskiepio valdymo klases rezultatų generavimo klases. Esybių klases galime suskirstyti į rezultatus ir įvesties duomenis.




**3.15 pav.** „WEB2UML“ sistemos klasių diagrama

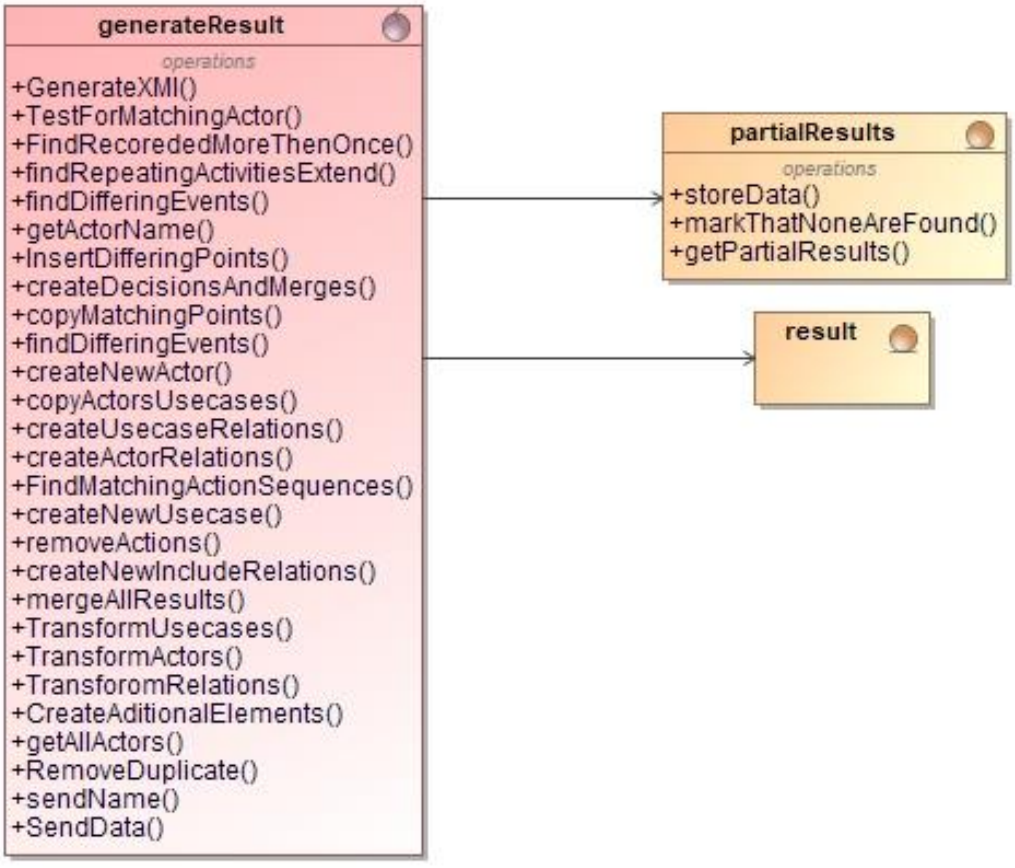
Paketų diagramoje 3.16 pav. vaizduojama sistemos loginė architektūra su jos dalis realizuojančiomis klasėmis. Įskiepis padalintas į tris posistemius (komponentus): įskiepio valdymo, registravimo ir generavimo. Registravimo posistemis atsakingas už vartotojo duomenų įvedimą ir

veiksmų atliekamų sistemoje registravimą. Įskiepio valdymo posistemis atsakingas už įskiepio komponentų valdymą. Generavimo posistemis apima rezultatų generavimo klases.



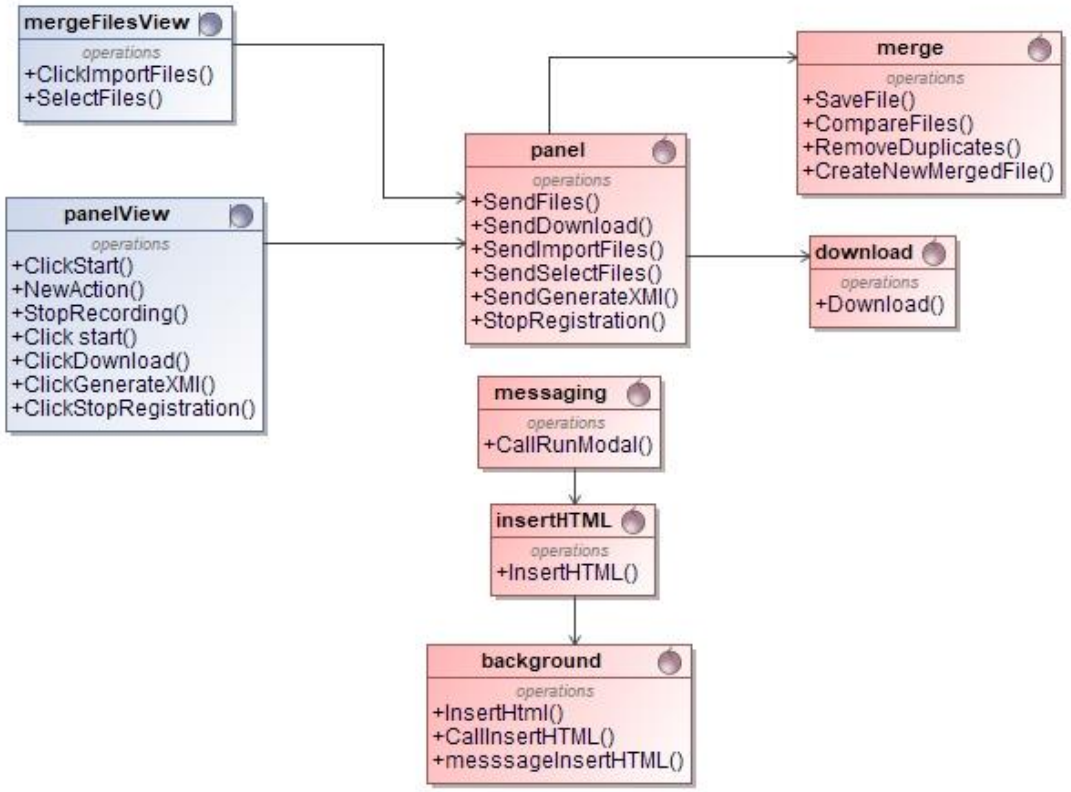

**3.16 pav.** „WEB2UML“ loginės architektūros diagrama

3.17 pav. pateikiamoje diagramoje detalizuojamos generavimo posistemio klasės, pateikiant jų operacijas ir tarpusavio ryšius.



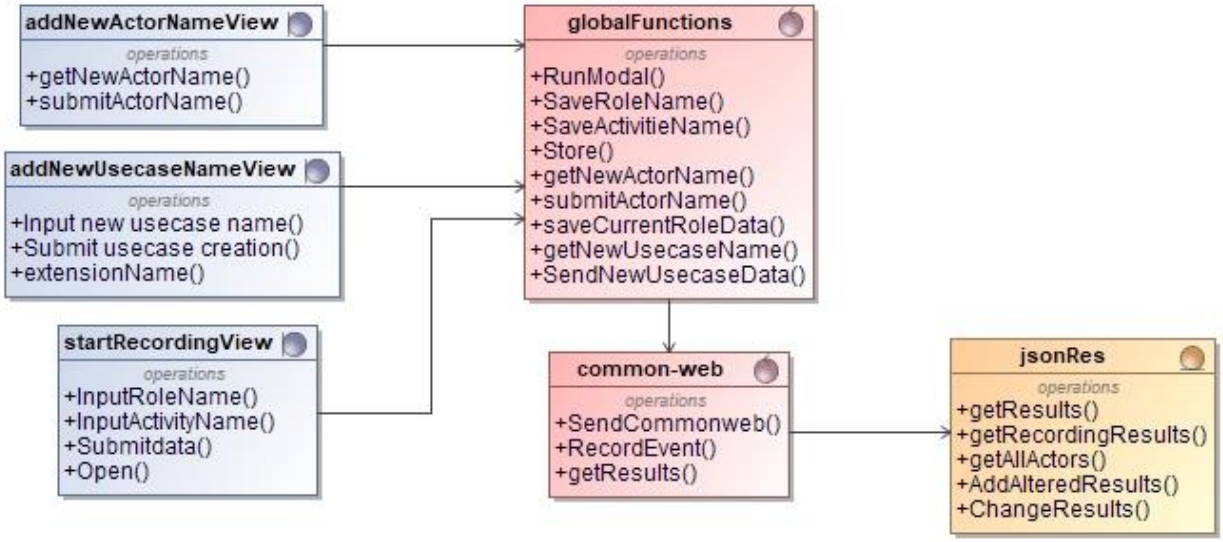

**3.17 pav.** Generavimo posistemio klasių diagrama

3.18 pav. vaizduojamoje diagramoje detalizuojamos įskiepio valdymo posistemio klasės. Šis posistemis apima valdiklių ir ribines klases, atsakingas už projektuojamo naršyklės įskiepio valdymą.




**3.18 pav.** Įskiepio valdymo posistemio klasių diagrama

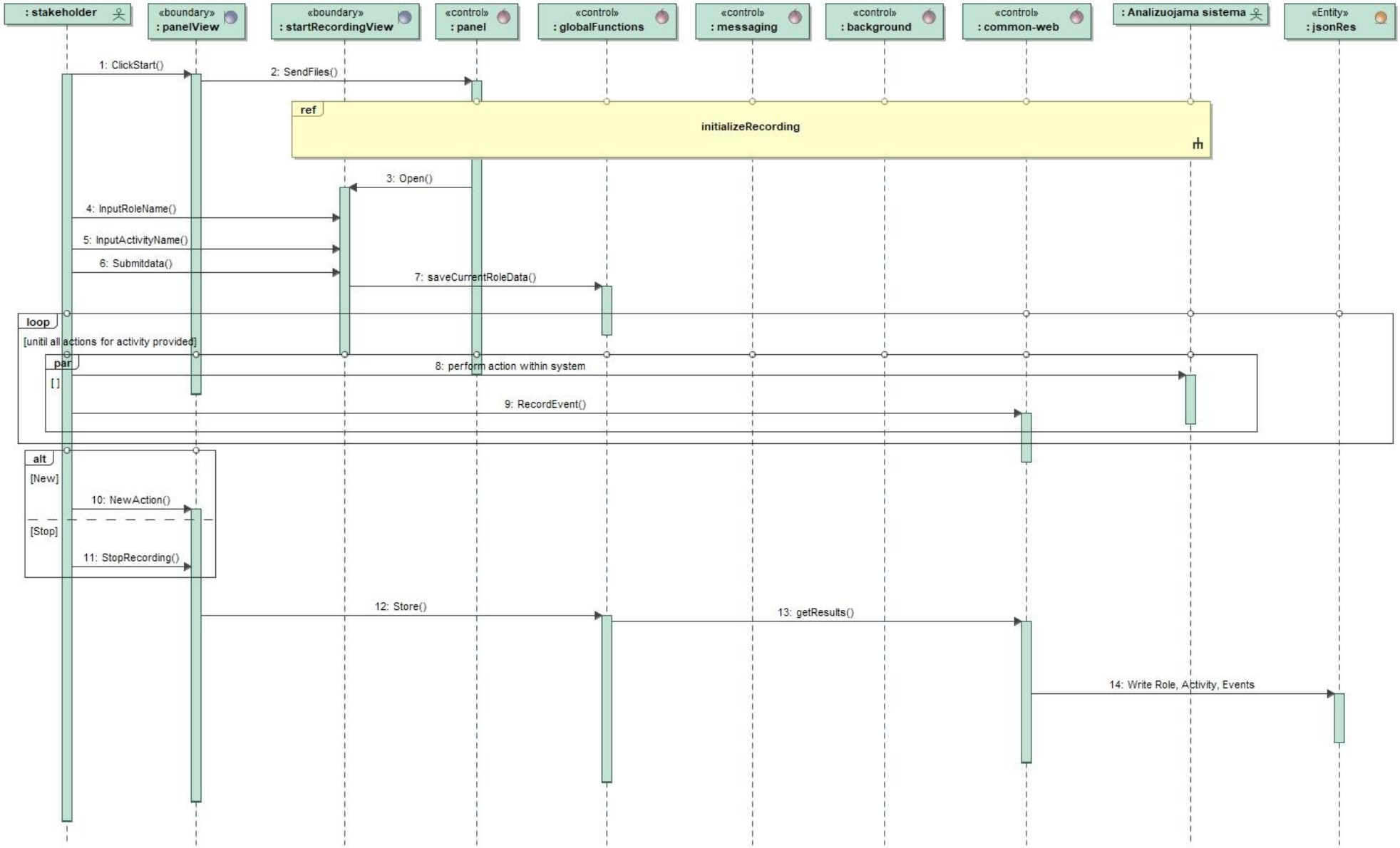
3.19 pav. vaizduojamoje diagramoje atvaizduojama registravimo posistemio klasių diagrama su atvaizduotomis klasių operacijomis ir ryšiais.



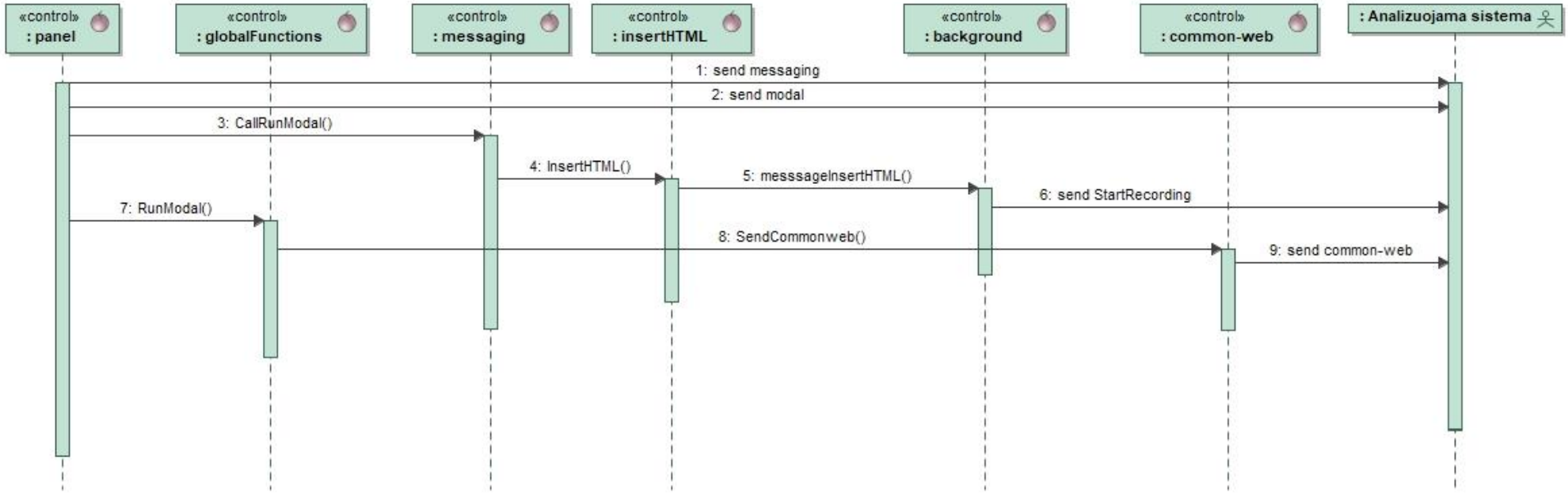

**3.19 pav.** Registravimo posistemio klasių diagrama

### 3.2.2. Kompiuterizuojamų panaudojimo atvejų sekų diagramos

Šiame skyriuje pateikiamos diagramos atvaizduojančios sistemą realizacijos klasėmis. 3.20 pav. atvaizduojama kaip sisteminių klasių lygmenyje vykdomas analizuojamos veiklos registravimas su nuorodą į išplečiantį panaudojimo atvejį pradėti sistemos registravimą, kuris atvaizduotas 3.21 pav.

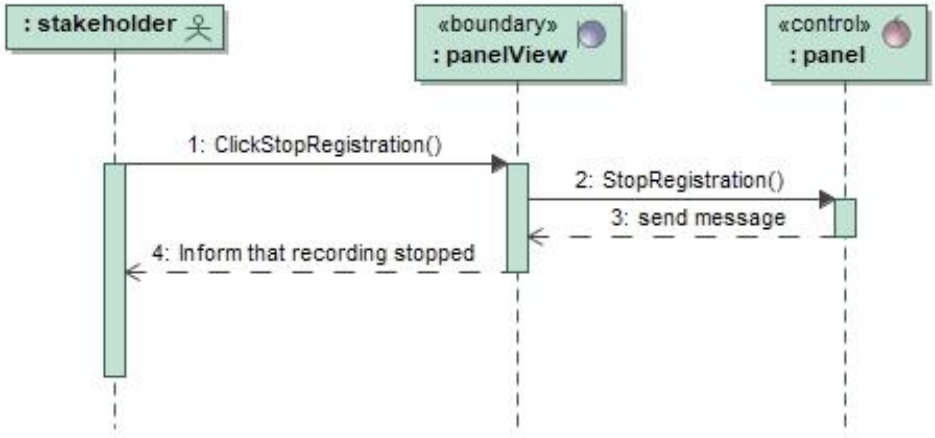



**3.20 pav.** Panaudojimo atvejo „Registruoti veiklą“ sekų diagrama

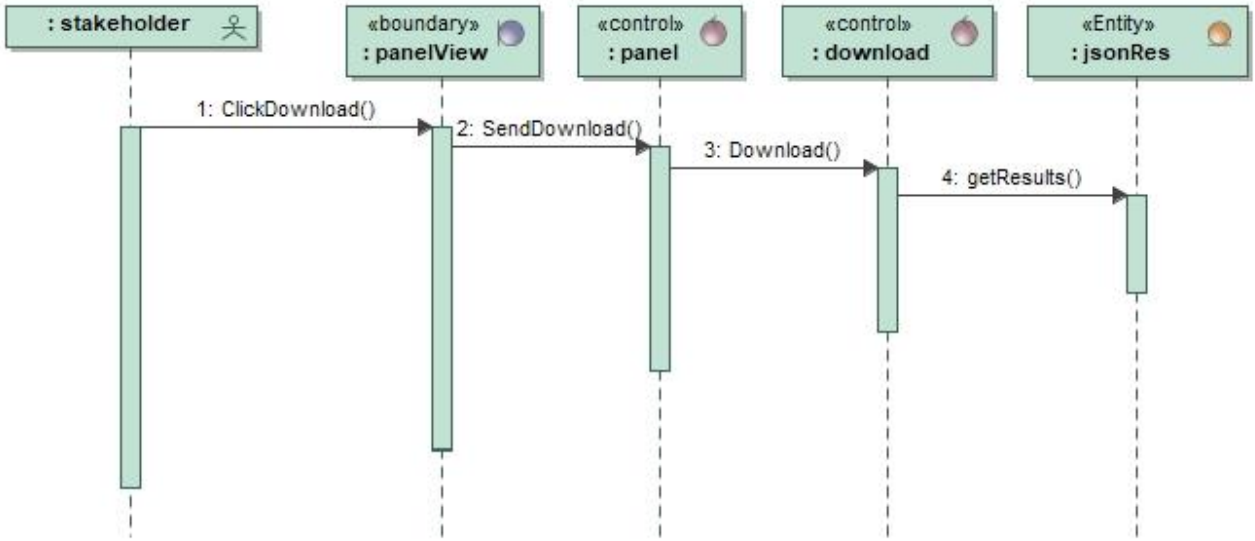



**3.21 pav.** Panaudojimo atvejo „Pradėti sistemos registravimą“ sekų diagrama

Šioje diagramoje 3.22 pav. atvaizduojamas procesas realizacijos klasėmis, nurodantis kokius veiksmus sistema vykdo, kai naudotojas sustabdo veiklos registravimą. Taip pat 3.23 pav. vaizduojama diagrama parodo kaip elgiasi sistema vykdant failo atsiuntimą naudotojui.

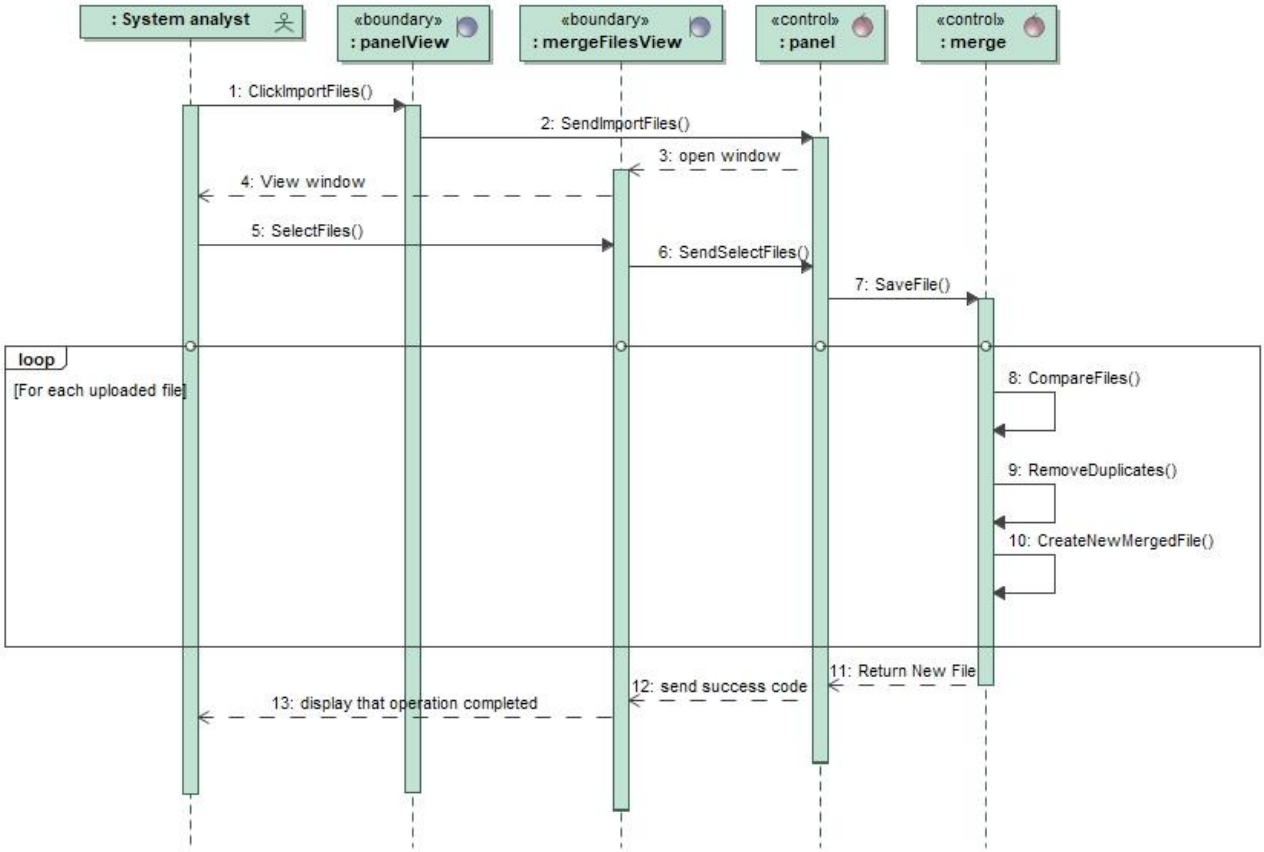



**3.22 pav.** Panaudojimo atvejo „Žymėti svetainės analizės pabaigą“ sekų diagrama



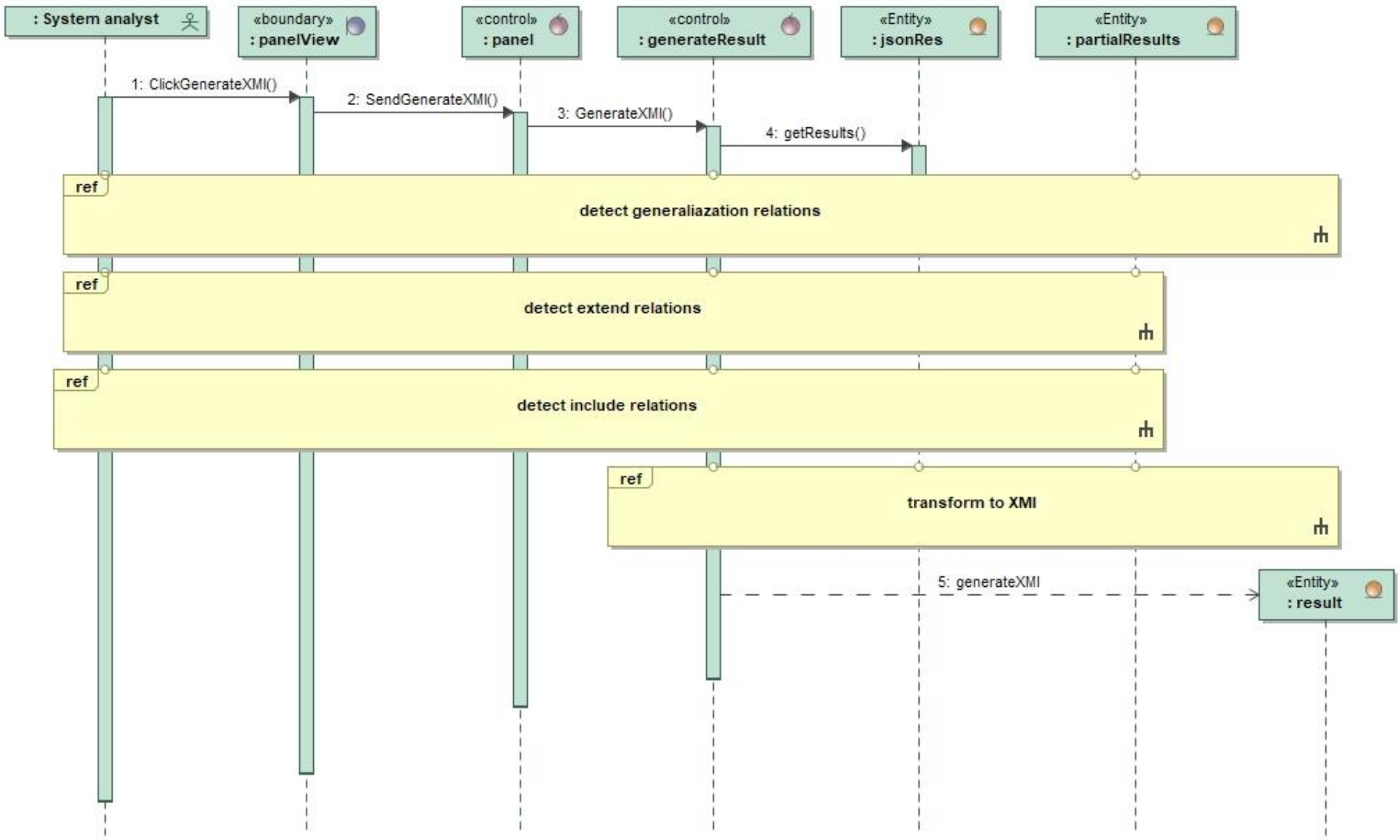

**3.23 pav.** Panaudojimo atvejo „Parsisiųsti rezultatus“ sekų diagrama

3.24 pav. atvaizduoja sistemos elgseną ir iškviečiama operacijas, siekiant apjungti keletą duomenų registravimo failų.



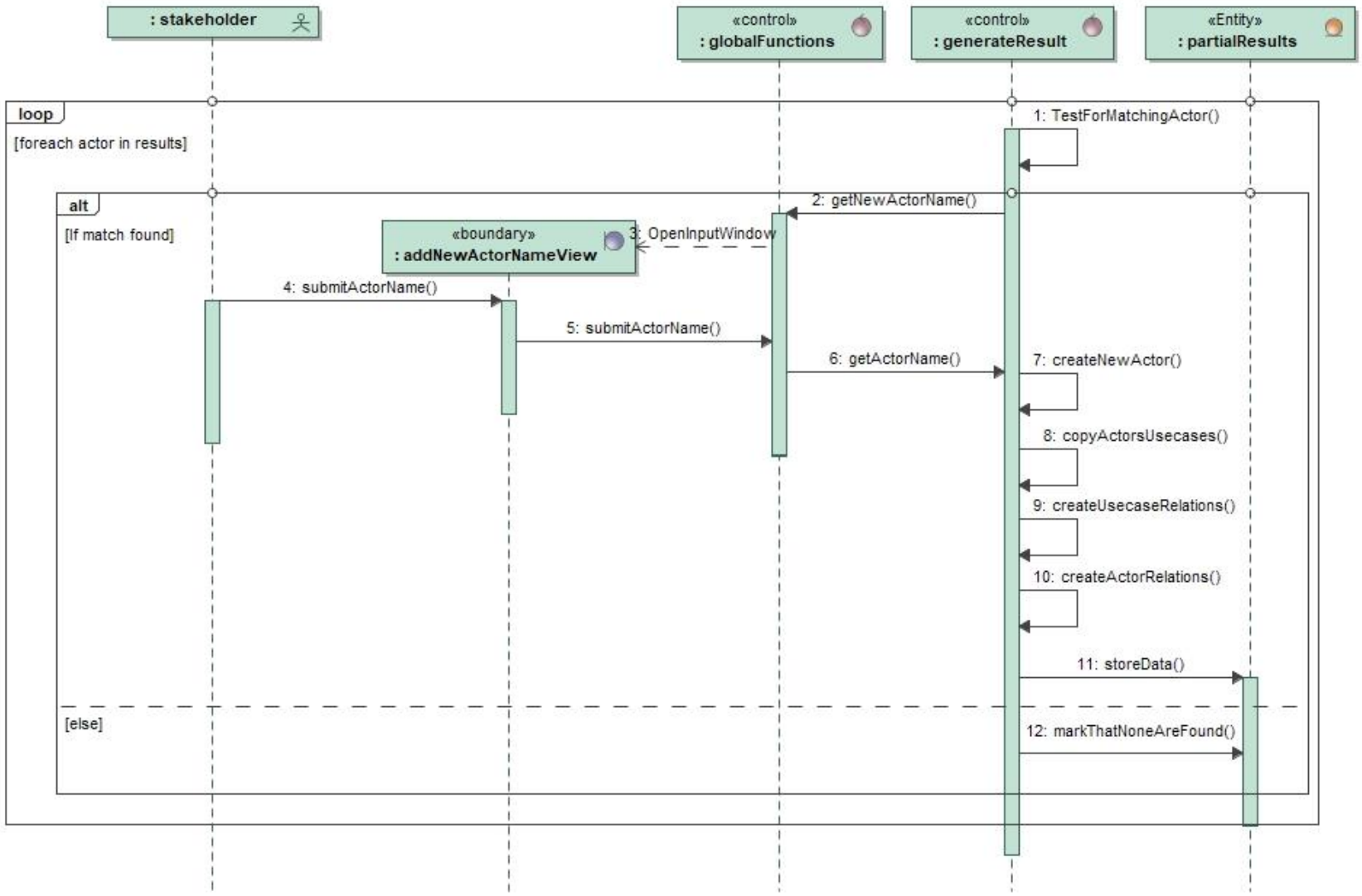

**3.24 pav.** Panaudojimo atvejo „Apjungti keletą failų“ sekų diagrama

3.25 pav. atvaizduoja procesą kurį vykdo sistema, kai naudotojas nusprendžia transformuoti sistemos analizės rezultatus į XMI formatu atvaizduotą panaudojimo atvejų modelį. Šioje diagramoje taip pat galime matyti įtrauktas veiklas, kurios vykdomos generavimo proceso metu.



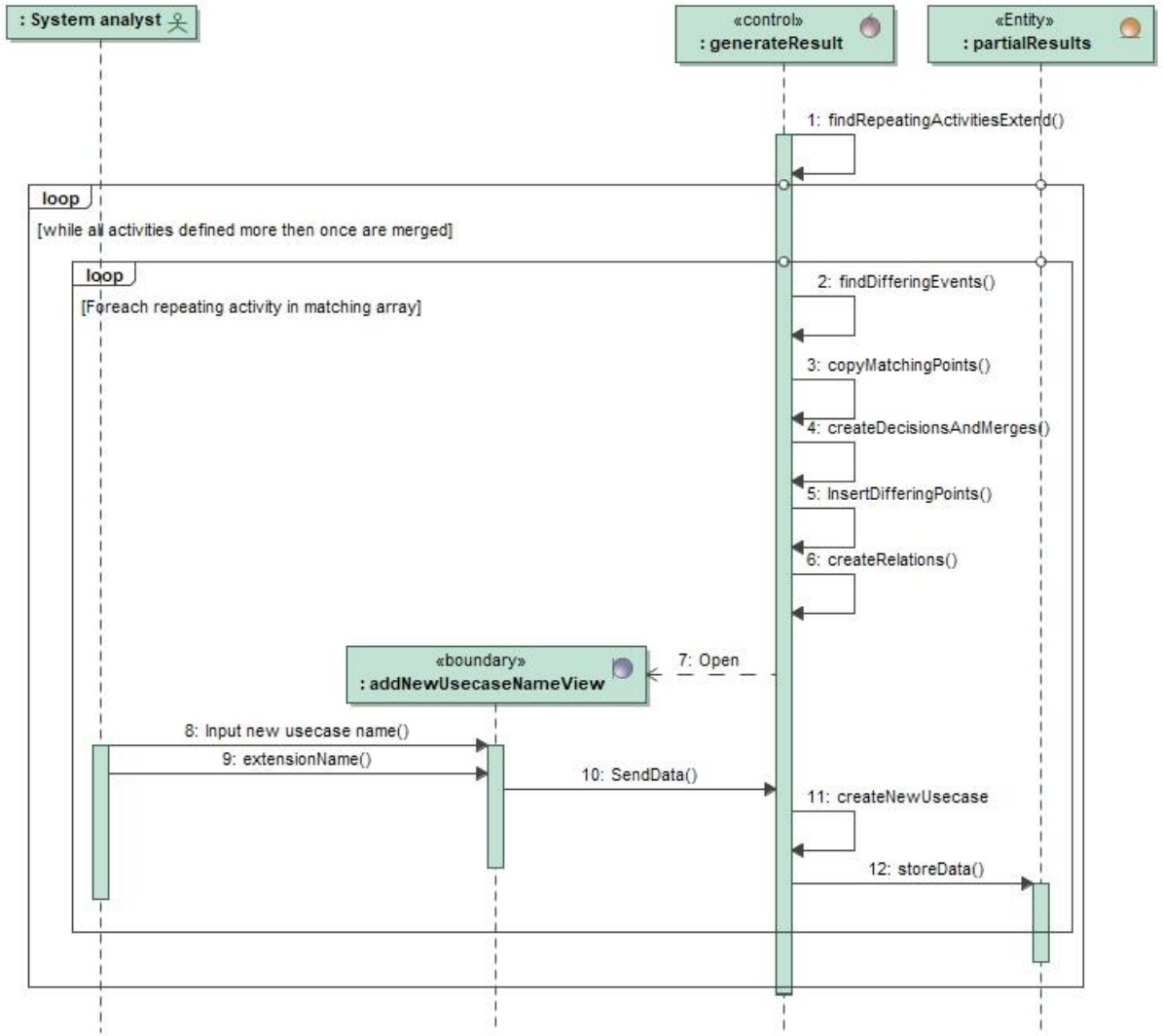

**3.25 pav.** Panaudojimo atvejo „Generuoti panaudojimo atvejų modelį“ sekų diagrama

Pirma nustatomi apibendrinimo ryšiai, kurių identifikavimas detaliai aprašytas 3.26 pav. diagramoje.



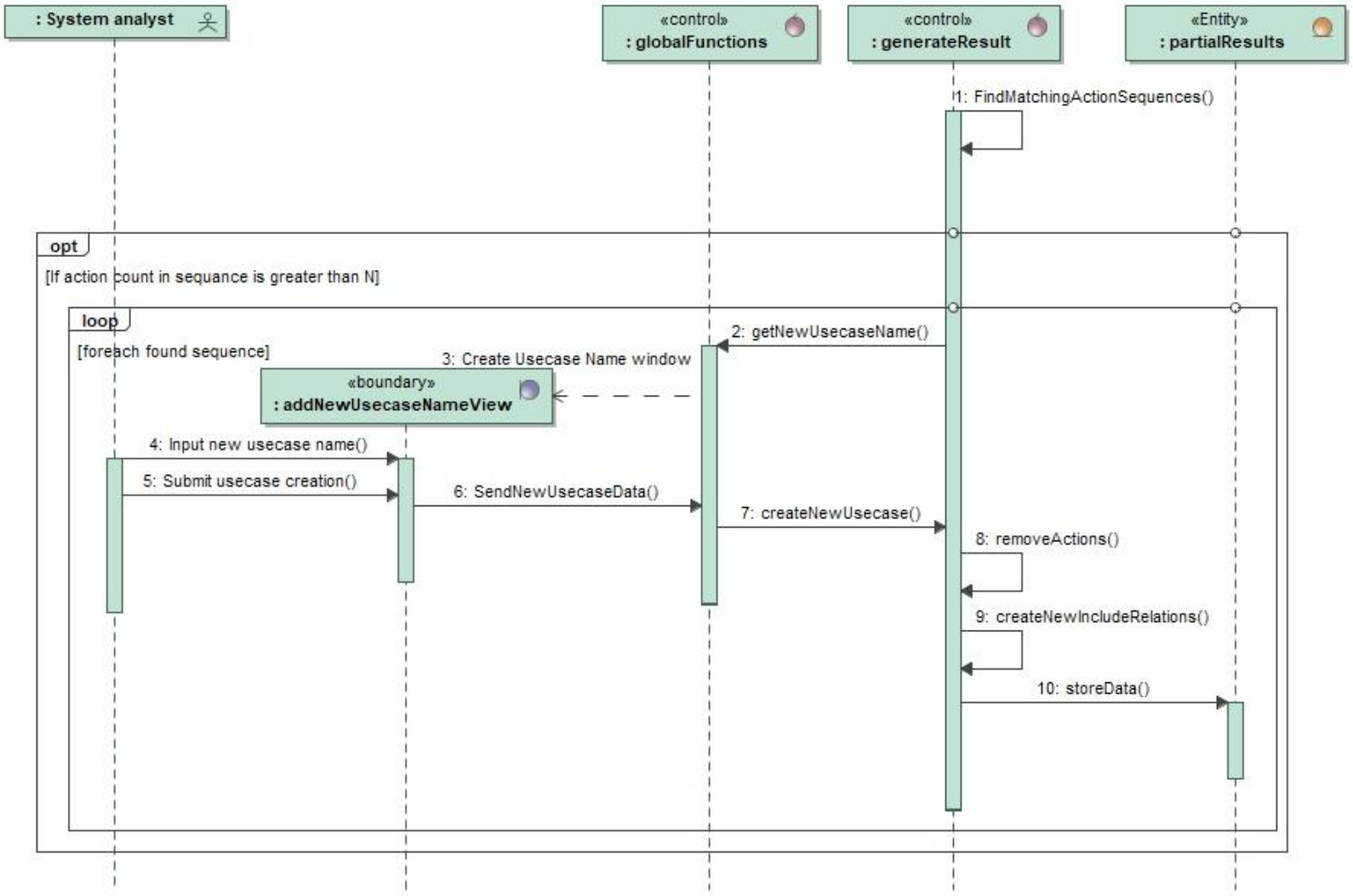

**3.26 pav.** Panaudojimo atvejo „Identifikuoti apibendrinimo ryšius“ sekų diagrama

Toliau identifikuojami išplėtimo ryšiai, surandant ir sukuriant visus reikiamus elementus kurių reikia siekiant atvaizduoti juos modelyje. Šis procesas detaliai atvaizduojamas 3.27 pav.



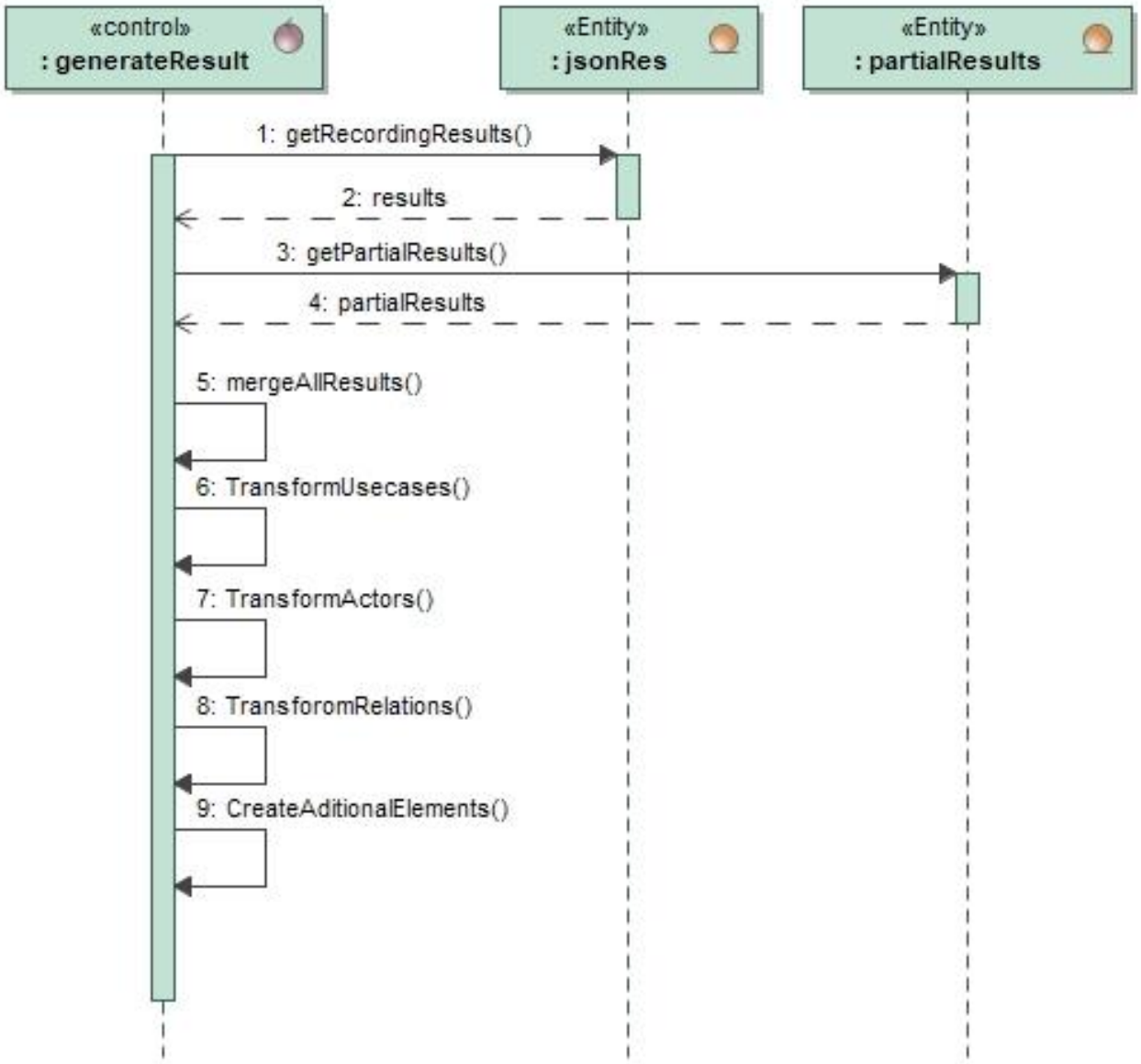

**3.27 pav.** Panaudojimo atvejo „Identifikuoti išplėtimo ryšius“ sekų diagrama

Priešpaskutinis generavimo veiklos etapas yra įtraukimo ryšių nustatymas ir rezultatų modelio sutvarkymas sukuriant ir įtraukiant visus reikalavimus elementus ir ryšius įtraukimo ryšiams atvaizduoti. Detalus šio etapo procesas atvaizduotas projekto klasėmis randamas 3.28 pav.




**3.28 pav.** Panaudojimo atvejo „Identifikuoti įtraukimo ryšius“ sekų diagrama

Paskutinis generavimo etapo veiksmas – tai rezultatų transformavimas, kurie buvo gauti praeituose etapuose į XMI formatą. Šis procesas detaliai atvaizduotas 3.29 pav.



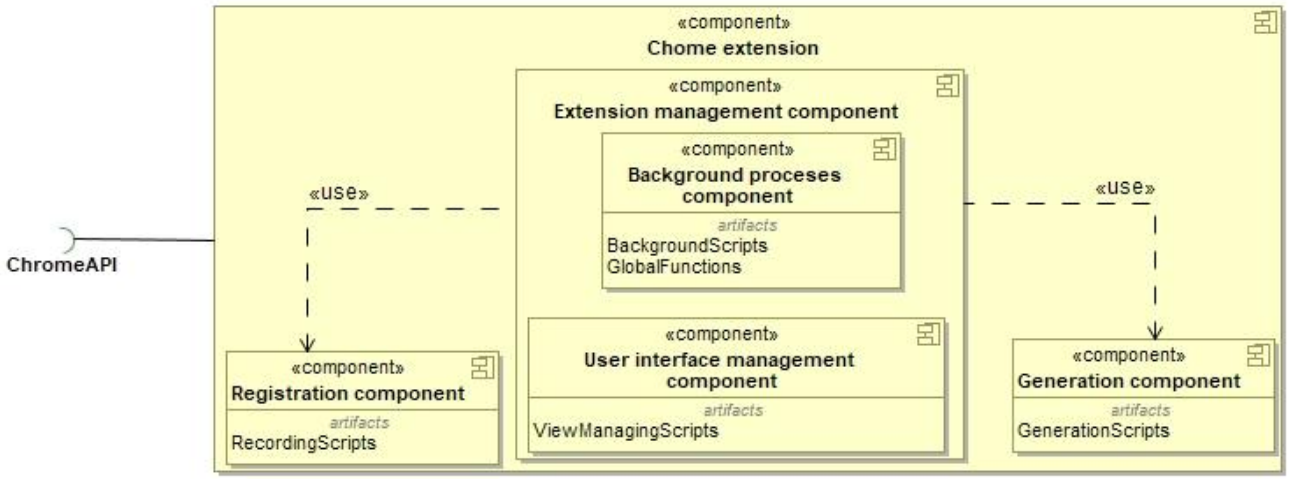

**3.29 pav.** Panaudojimo atvejo „Transformuoti rezultatų į XMI formatą“ sekų diagrama

Suprojektavus sistemą, realizacijos klasėmis, naujojo įrankio kūrimas, turės aiškų atskaitos ir tikrinimo tašką. Remiantis šiomis klasėmis bus sukuri sistemos komponentai ir klasės, kurios sudarys projekto programinio kodo rinkinį.

# 4. „WEB2UML“ SPRENDIMO REALIZACIJA IR TESTAVIMAS

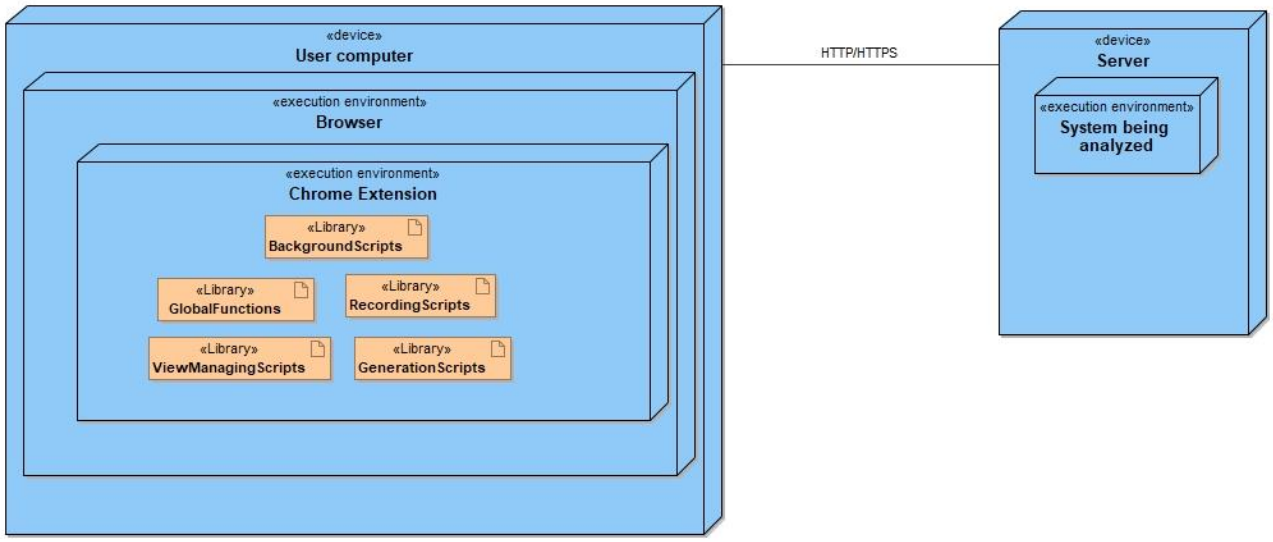
## 4.1. Sistemos komponentai

Sistemos komponentai ir komponentus realizuojantys artefaktai atvaizduojami 4.1 pav. Pagrindiniai komponentai skirti registravimo papildinio valdymui ir generavimui. Registravimo komponentas atsakingas už naudotojo veiksmų analizuojamoje sistemoje registravimą ir apdorojimą. Papildinio komponentas skirtas valdyti visus kitus komponentus ir atvaizduoti naudotojo grafinei sąsajai, kuria naudojantis naudotojas įveda duomenis į sistemą. Generavimo komponentas apdoroja registravimo komponento gautus rezultatus ir juos transformuoja į XMI tipo failą.




**4.1 pav.** „WEB2UML“ komponentų diagrama

Sistemos diegimo diagrama pateikiama 4.2 pav., joje matoma, kokiose aplinkose egzistuoja „WEB2UML“ sistema, iš kokių bibliotekų ji sudaryta, ir kaip ji bendrauja su analizuojama sistema.




**4.2 pav.** „WEB2UML“ diegimo diagrama

## 4.2. Sistemos testavimas

Sistemos testavimas buvo vykdomas, testuojant dažniausiai pasitaikančius sistemos naudojimo scenarijus. Scenarijus buvo sukurtas, siekiant patikrinti registravimo komponento veikimą, bei kiekvieną generavimo procese egzistuojantį ryšių nustatymo funkcionalumą.

**Pirmasis scenarijus**

Sistemoje įvedame aktorių pavadinimu A, kuris atlieka veiklą U1. Šios veiklos metu aktorius prisijungia prie sistemos, atveria pagrindinį sistemos langą, išfiltruoja duomenis pasirinktais kriterijais ir pateikia formą.

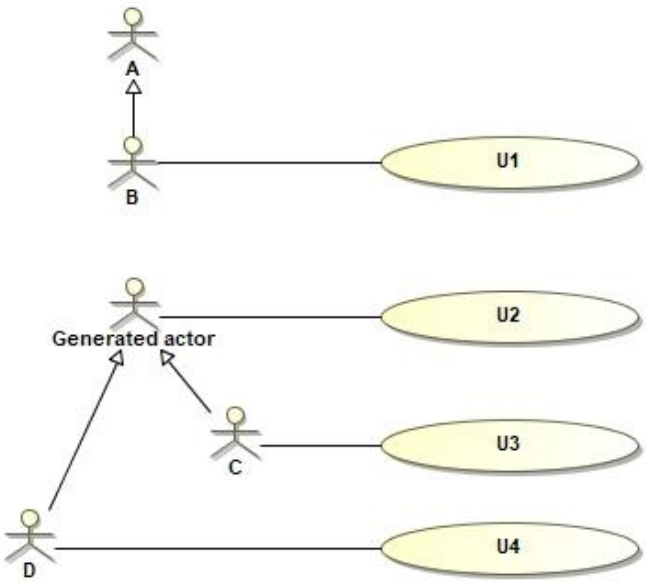
Siekiamas rezultatas. Sukuriama panaudojimo atvejų diagrama, kurioje egzistuoja aktorius A ir panaudojimo atveji U1 sujungti asociacijos ryšiais. Panaudojimo atvejų modelyje panaudos atvejis U1 detalizuojamas veiklos diagrama, kurioje atvaizduotas veiklos scenarijus su visais užregistruotais veiksmais.

**Antrasis scenarijus**

Sistemoje užregistruojame aktorių A, kuris vykdys veiklą U1. Užregistravęs veiklą, aktorius A pažymi, kad baigia savo veiklos registravimą. Aktorius B taip pat nurodo, kad vykdo veiklą U1 ir ją užregistruoja. Aktoriaus A ir B registruojamos veiklos sutampa identiškai. Siekiant patikrinti ir nepilną persidengimą tarp aktorių, aktorius C užregistruoja veiklą U2, kuri yra unikali lyginant su kitomis veiklomis. Aktorius C taip pat įgyvendina veiklą U3, kuri nesutampa su kitomis veiklomis. Aktorius D užregistruoja veiklą U2, kuri yra identiška aktoriaus C atliktai veiklai U2. Aktorius D taip pat įgyvendina veiklą U4.

Siekiamas rezultatas. Sukuriama panaudojimo atvejų diagrama, kurioje sukuriami aktoriai A, B, C ir D. Bei panaudojimo atvejai U1, U2, U3 ir U4. Tarp aktorių A ir B nustatomas apibendrinimo ryšys. Tarp aktorių C ir D nustatomas dalinis sutapimas, todėl sukuriamas naujas aktorius, kuris apibendrinimo (angl. *generalization*) ryšiu sujungiamas su naujai sukurtu aktoriumi. Aktoriai C ir D sujungiami apibendrinimo ryšiu su naujai sugeneruotu aktoriumi.

Iš pirmojo ir antrojo scenarijaus užregistruotų veiksmų sugeneruota panaudojimo atvejų diagrama pateikiama 4.3 pav. ji atitinka siekiamą rezultatą.



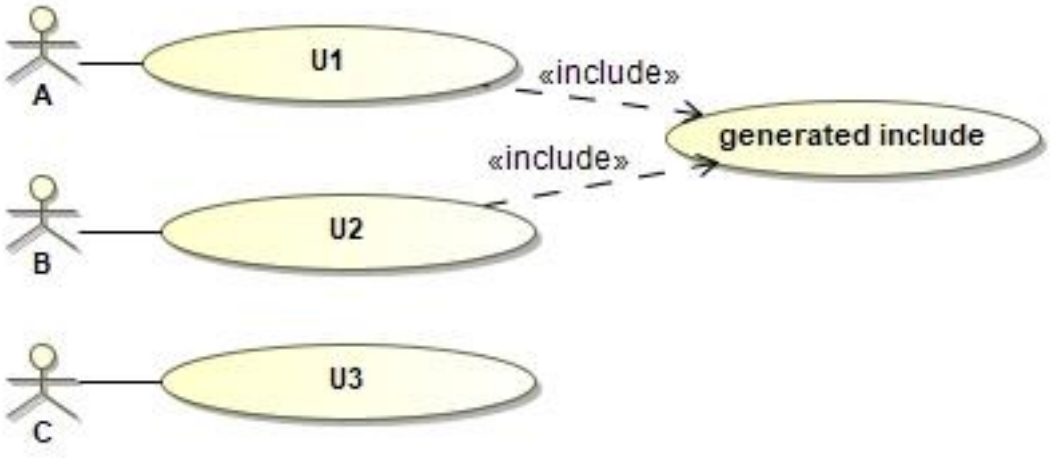

**4.3 pav. Apibendrinimo ryšių nustatymo testavimo rezultatai**

**Trečiasis scenarijus**

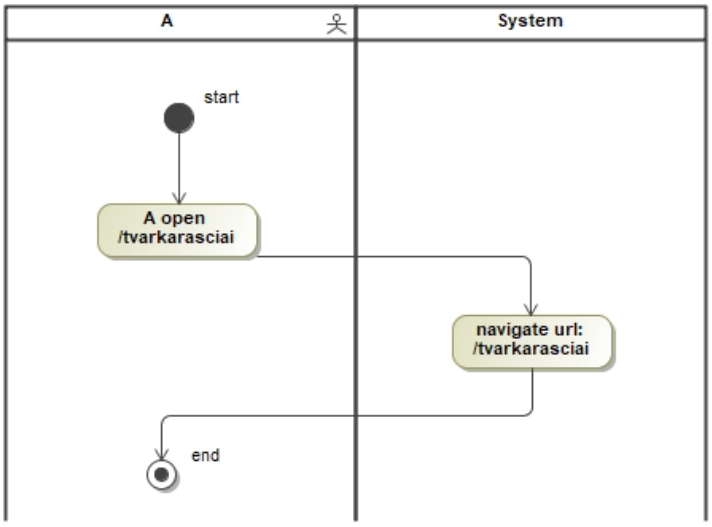
Trečiasis scenarijus skirtas apėmimo (*include*) ryšių nustatymui patikrinti. Šio scenarijaus metu aktorius A užregistruoja veiklą U1, kurios pradžioje reikia atverti svetainės puslapį */tvarkaraščiai*, šiame puslapyje aktorius A taip pat išfiltruoja norimos klasės tvarkaraštį, pasirinkdamas klasę. Aktorius B, vykdantis veiklą U2 atveria tą patį */tvarkaraščiai* puslapį ir išfiltruoja tvarkaraštį pasirinkdamas šios savaitės tvarkaraštį, nepasirinkdamas klasės. Galiausiai aktorius C sistemoje įvykdo veiklą U3, kurioje užregistruoja naują klasę.

Siekiamas rezultatas trečiojo scenarijaus atveju – nustatytas sutapimas tarp panaudojimo atvejų U1 ir U2 bei sukurtas naujas panaudojimo atvejis, kuris aprašo sutampančią veiklos sekos dalį. Tačiau tarp U3 ir likusių panaudojimo atvejų apėmimo ryšys nebus nustatytas.

Atlikus trečiojo scenarijaus veiksmus, sugeneruota panaudojimo atvejų diagrama, kuri pavaizduota 4.4 pav. Nustatyta sutampanti dalis tarp panaudojimo atvejų U1 ir U2 buvo iškelta į naują panaudojimo atvejį. Šį panaudojimo atvejį detalizuojanti diagrama atvaizduota 4.5 pav. Sistema trečiojo scenarijaus metu sugeneravo rezultatą, atitinkantį siekiamą.




**4.4 pav. Panaudojimo atvejų diagrama sugeneruota vykdant trečiąjį scenarijų.**



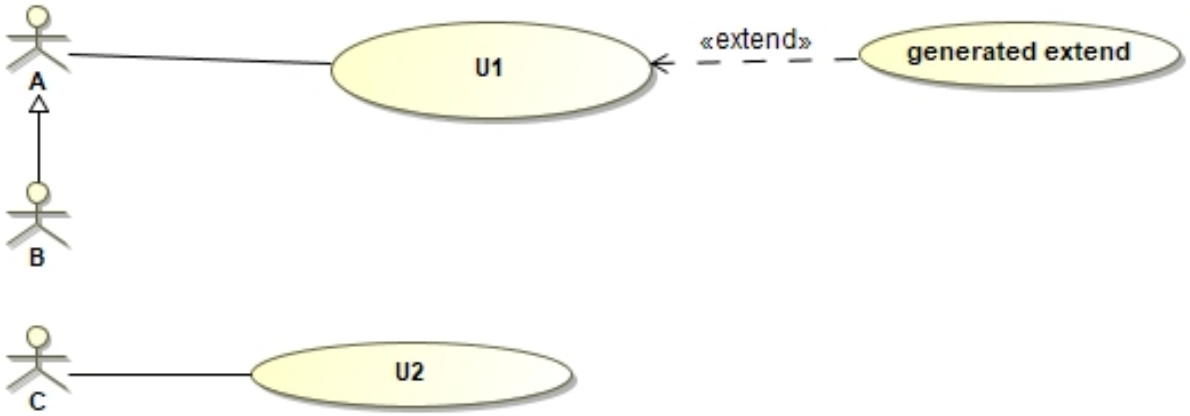

**4.5 pav. Veiklos diagrama detalizuojanti pasikartojančią dalį tarp veiklų U1 ir U2**

**Ketvirtasis scenarijus**

Ketvirtasis scenarijus skirtas patikrinti sistemos gebėjimą nustatyti išplėtimo ryšius. Ketvirtojo scenarijaus metu aktorius A užregistruoja veiklą U1. Aktorius B taip pat registruoja veiklą U1, tačiau neteisingai suvedęs adreso ir telefono lauką, turi grįžti ir užpildyti formą dar kartą, pataisant adreso lauką ir telefono numerį. Aktorius C įvykdo veiklą U2, kuri niekaip nesusijusi su veikla U1.

Siekiamas rezultatas, kad U1 panaudojimo atvejyje atsiras alternatyvus scenarijus, kurio metu ta pati veikla bus užregistruota ne tokia pat seka. Dėl to bus sukurtas išplečiantysis panaudojimo atvejis. Taip pat bus sukurtas paveldėjimas tarp aktorių A ir B. C aktoriaus veikla bus užregistruota.

Atlikus veiksmus, aprašomus ketvirtajame scenarijuje, gautas rezultatas atitiko rezultatus, buvo sukurtas naujas panaudojimo atvejis, kuris buvo sujungtas su U1 panaudojimo atveju išplėtimo ryšiu. Taip pat nustatytas apibendrinimo ryšys tarp aktorių A ir B, bei sugeneruotas aktorius C ir veikla U2. Rezultatai atitiko siekiamus rezultatus, gauta diagrama pateikiama 4.6 pav.




**4.6 pav. Panaudojimo atvejų diagrama įgyvendinus ketvirtąjį scenarijų.**

## 4.3. Realizacijos išvados

1. Realizuojant pastebėta, kad siekiant sukurti lanksčią sistemą, tarpinius rezultatus patogu saugoti JSON formatu, o siekiant pritaikyti sistemą naudojimui CASE įrankyje, rezultatus reikia atvaizduoti XMI formatu.

2. Ištestavus sistema nustatytą, kad sistema teisingai generuoja elementus ir ryšius tarp jų, nesant dideliam kiekiui *JavaScript* kalba generuojamo turinio, bei neregistruojant veiksmų <<*iframe*>> HTML kalbos žymoje.

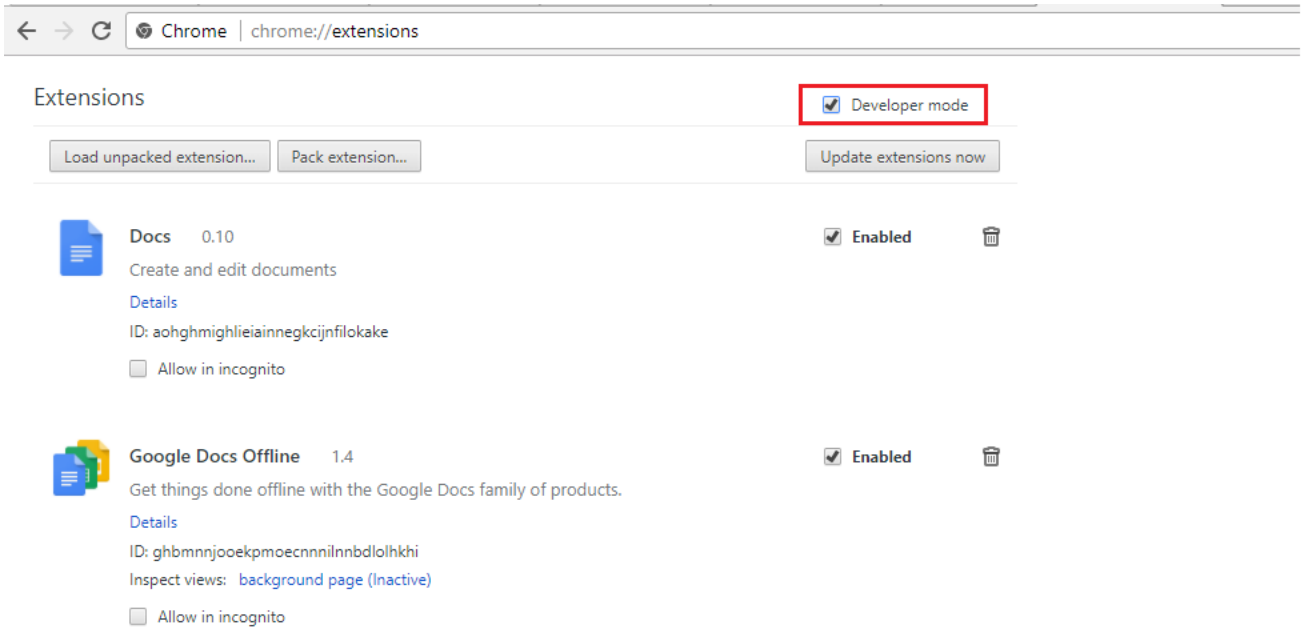
## 4.4. Dokumentacija naudotojui

### 4.4.1. Sistemos paskirtis, naudotojai, pagrindinės sistemos funkcijos

UML Panaudojimo atvejų modelio išgavimo iš internetinių sistemų sistema, skirta sugeneruoti panaudojimo atvejo modelį, registruojant sistemos naudotojo naudojimąsi sistema. Ši sistema realizuota kaip kompanijos „Google“ naršyklės „Chrome“ įskiepis. Pagrindiniai sistemos naudotojai, tai žmonės siekiantys geriau suprasti sistemos funkcionalumą išgaunant panaudojimo atvejų modelį. Kaip sistemos panaudojimo pavyzdį būtų galima įvardinti sistemos panaudojimo atvejų modelio išgavima iš lygtinės sistemos, siekiant sukurti naują sistemą senosios pagrindu, pagreitinant reikalavimų išgavimo etapą naujos sistemos kūrimui. Sistema taip pat galėtų pasitarnauti palyginant kelių sistemų funkcionalumą tarpusavyje bei lyginant panaudojimo atvejų modelius. Pagrindinės sistemos funkcijos tai, naudotojo veiklos registravimas, veiklą reprezentuojančių failų importavimas, eksportavimas ir registruotos veiklos ar kelių veiklų apjungtų į vieną transformavimas į XMI formato failą. Šie XMI formato failai vėliau gali būti importuojami į UML modeliavimo įrankius ir išsamiai nagrinėjami, tobulinami ar naudojami, kaip reikalavimų modelio dalis.

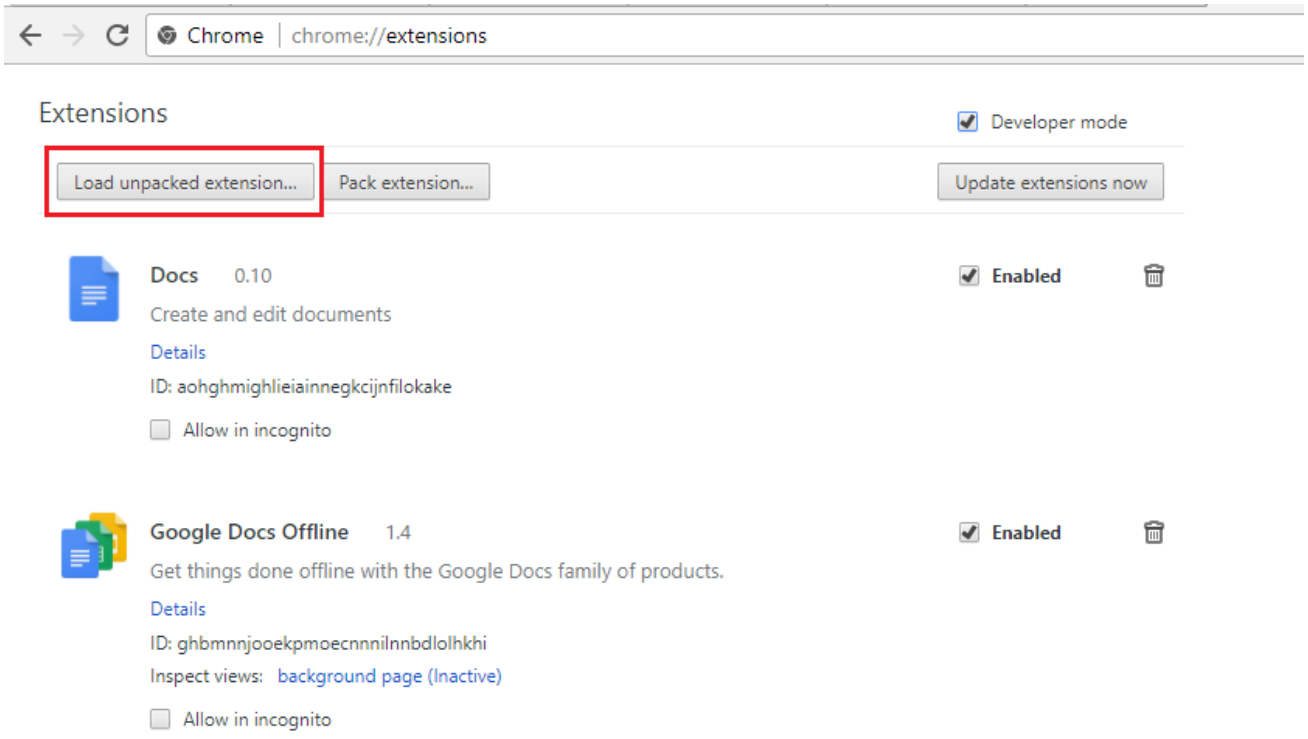
### 4.4.2. Sistemos diegimas ir paruošimas darbui

Prieš pradedant darbą su sistema būtina įsidiegti „Chrome“ naršyklę naudojamame kompiuteryje. Taip pat norint peržiūrėti rezultatus rekomenduojama įsidiegti „Magicdraw“ 18.5 sistemos versiją. „Chrome“ naršyklėje diegiama, parsisiunčiant instaliacinį aplanką jį išskleidžiant savo kompiuteryje. Toliau vykdomi paruošiamieji darbai prieš sistemos diegimą. Atveriame “Chrome“ naršyklę ir įvedame adresą: chrome://extensions/. Atsivėrusiame lange pažymime *Developer mode* varnelę, kaip pavaizduota 4.7 pav.



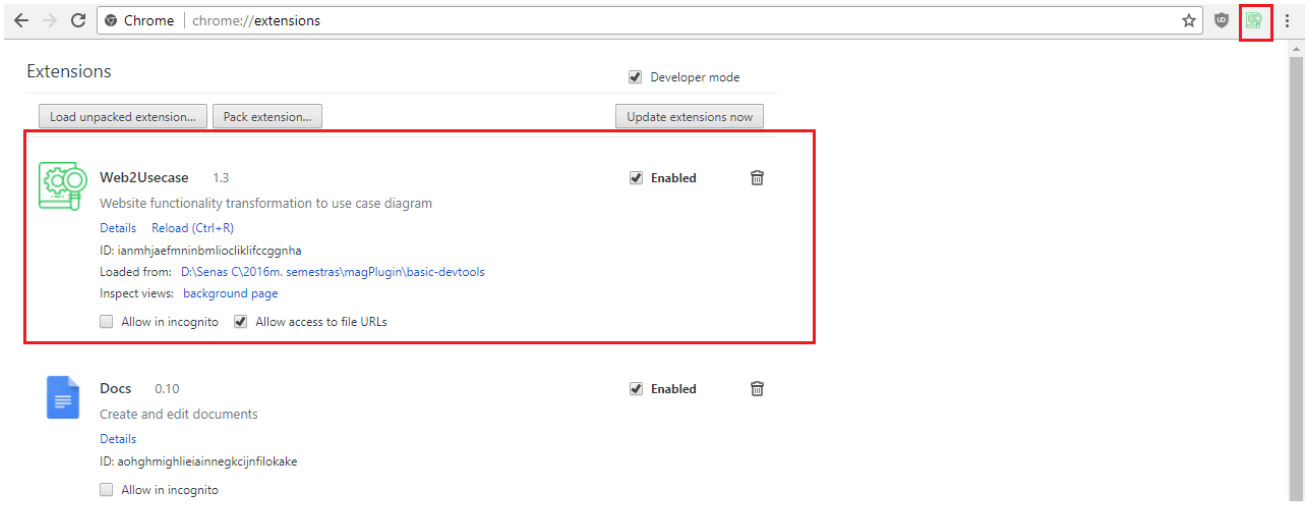

**4.7 pav.** Kūrėjo rėžimo įjungimas „Chrome“ naršyklėje

Atsiradusioje naujoje meniu juostoje spaudžiame mygtuką *Load unpacked extension,* kaip pavaizduota 4.8 pav. ir iššokusiame lange nurodome kelią iki iškleisto papildinio aplanko.




**4.8 pav.** Įskiepio įkėlimo mygtukas atvaizduotas kūrėjo menių juostoje

Sėkmingai įdiegus papildinį galime matyti naujai atsiradusį papildinį papildinių sąraše, kaip vaizduojama 4.9 **pav.**




**4.9** pav. Sėkmingai įdiegtas papildinys

### 4.4.3. Bendri sistemos naudojimo principai

Kiekvienas naudotojas besinaudojantis sistema gali naudotis pilnu jos funkcionalumu, tačiau siekiant aiškiau apibrėžti dažniausią sistemos panaudojimo būdą, bus aprašomi dviejų sistemos rolių standartiniai veiksmai.

### 4.4.4. Naudotojo vadovas kiekvienam naudotojo tipui

*4.4.4.1. Dalykinės srities atstovo naudotojo vadovas*

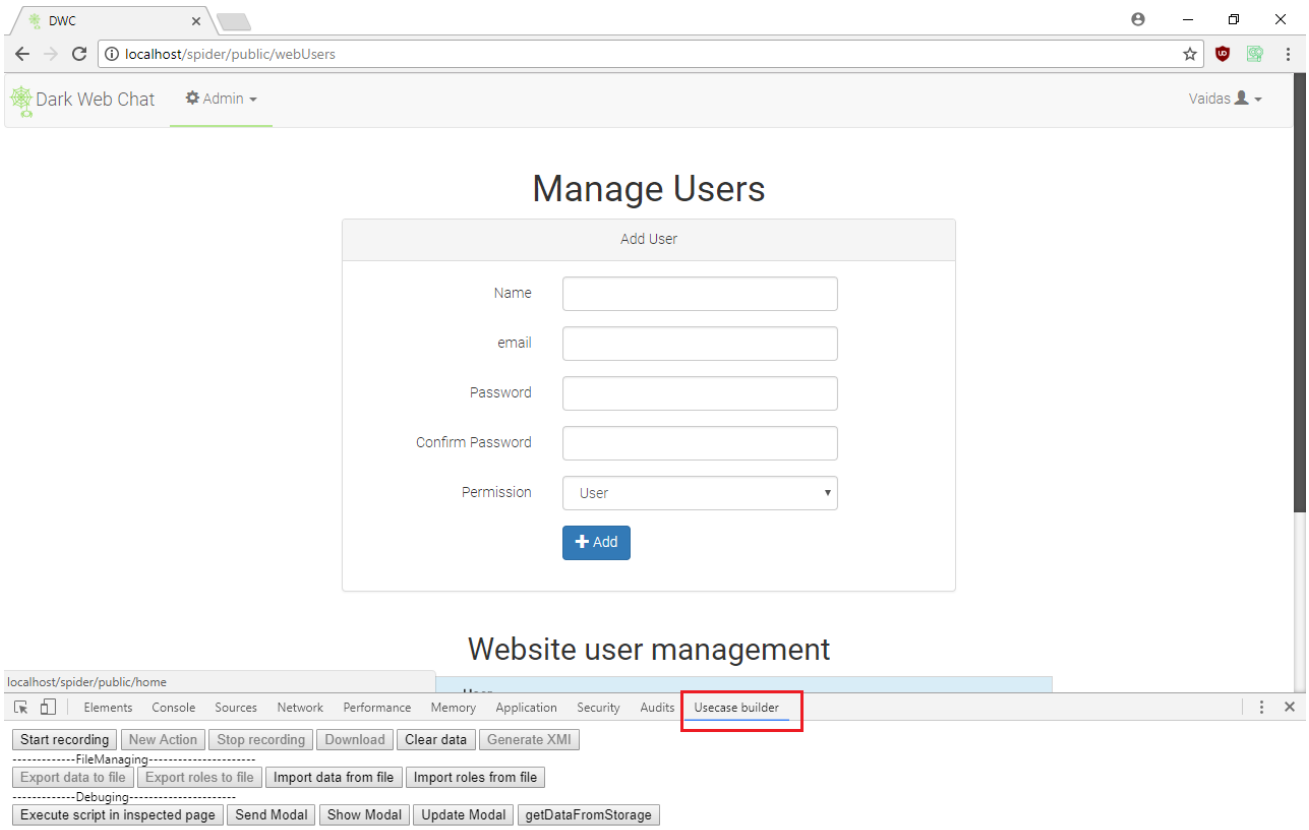
Dalykinės srities atstovo naudojimasis susideda iš dviejų pagrindinių etapų.

- Savo veiklos analizuojamoje sistemoje registravimo
- Rezultato failo sukūrimo ir perdavimo sistemos analitikui.

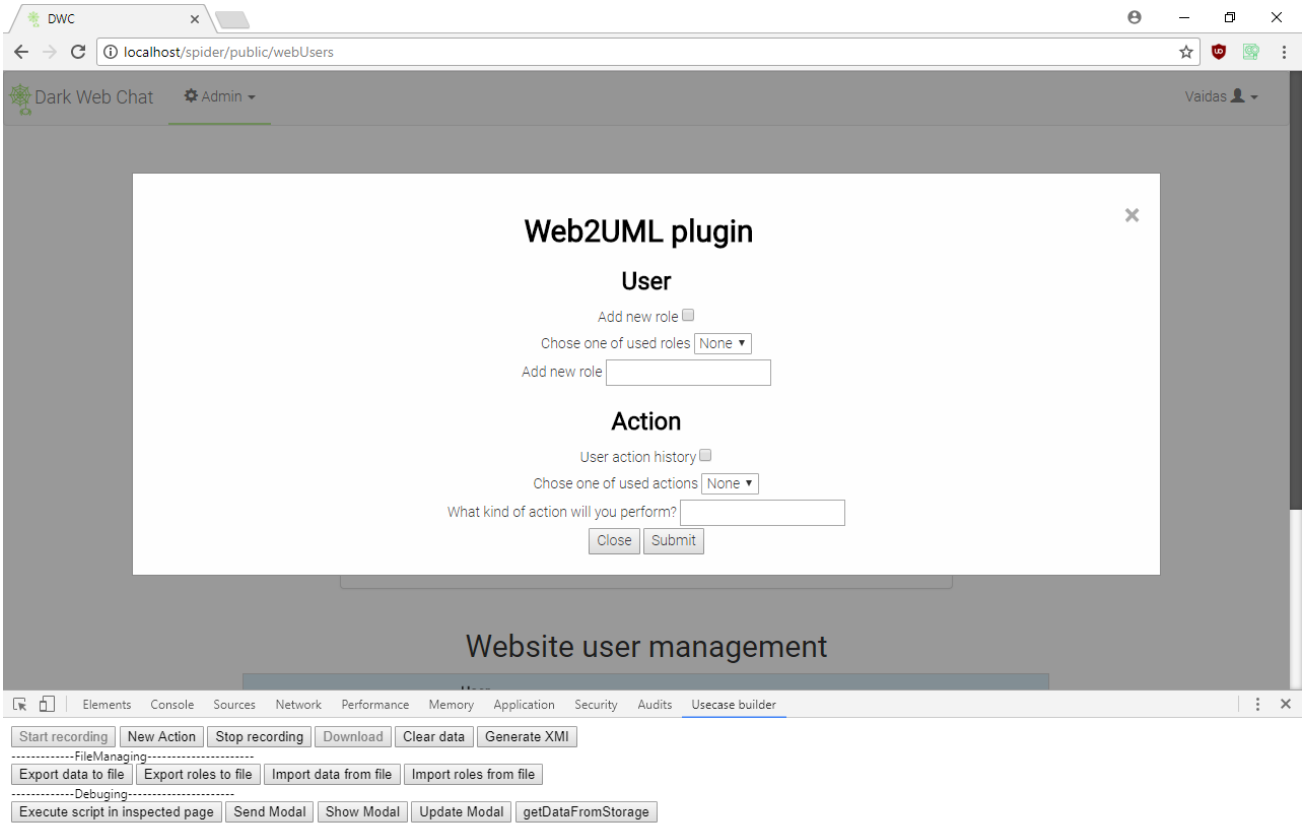
Savo veiklos registravimas analizuojamoje sistemoje pradedamas nuspaudžiant F12 ir apatinėje panelėje pasirenkant meniu punktą *Usecase builder*, kaip pavaizduota 4.10 pav. Atsivėrusioje panelėje nuspaudžiamas mygtukas *Start recording* ir naudotojui atvaizduojamas pradinių duomenų nustatymo langas. Šis langas atvaizduotas 4.11 pav. Jame matome naudotojo rolės įvedimo ir jo veiklos pavadinimo įvedimas. Nuspaudus mygtuką *Submit* uždaromas langas ir vykdoma veikla analizuojamoje sistemoje. Norint pridėti naują veiksmą spaudžiamas mygtukas *New action*. Norint baigti registruoti veiklas, spaudžiame mygtuką *Stop recording*.

*4.4.4.2. Analitiko naudotojo vadovas*

Sustabdę sistemos registravimą, galime atsisiųsti gautus rezultatus, nuspaudę mygtuką *Download* ir juos perduoti sistemos analitikui. Šis naudodamas mygtuką *Import data from file* gali apjungti daug registracijos failų ir nuspaudęs mygtuką *Generate XMI* sugeneruoti rezultatų failą, kuris parsiunčiami naudotojui į kompiuterį. Šį failą reiktų atverti su UML modeliavimui skirta programine įranga, kuri leidžia importuoti XMI formato failus.




**4.10 pav.** Plėtinio meniu atidarymo pavyzdys




**4.11 pav.** Sistemos naudotojo ir veiksmo įvedimo langas

# 5. EKSPERIMENTINIS „WEB2UML“ SISTEMOS TYRIMAS

## 5.1. Eksperimento planas

Siekiant patikrinti metodo ir jį realizuojančio įrankio „WEB2UML“ veikimą, buvo įgyvendinti du eksperimentai. Pirmuoju eksperimentu buvo siekiama kiekybiškai įvertinti metodą ir įrankio galimybę sugeneruoti tinkamas diagramas realiai veikiančioms sistemoms. Šios sistemos jau turi dokumentaciją ir joje pateiktos visos sistemą aprašančios diagramos taigi eksperimento metu buvo galimybė palyginti sugeneruotas ir sistemos kūrėjų sudarytas diagramas tarpusavyje. Antruoju eksperimentu buvo siekiama įvertinti sugeneruojamo panaudojimo atvejų modelio kokybę, remiantis ekspertų nuomone.

Pirmojo eksperimento metu buvo tiriama pasirinkta hipotezė, kurioje teigiame, kad panaudojimo atvejų modelis gali būti išgautas iš internetinių sistemų. Šios hipotezės patvirtinimui buvo pasirinkti tokie įvertinimo kriterijai:

1. Sugeneruota panaudojimo atvejų diagrama turi atitikti panaudojimo atvejų diagramą, aprašytą sistemos dokumentacijoje, nemažiau kaip 80 %.
2. Sugeneruotos veiklos diagramos turi atitikti veiklos diagramas, aprašytas sistemos dokumentacijoje, ne mažiau nei 50%.

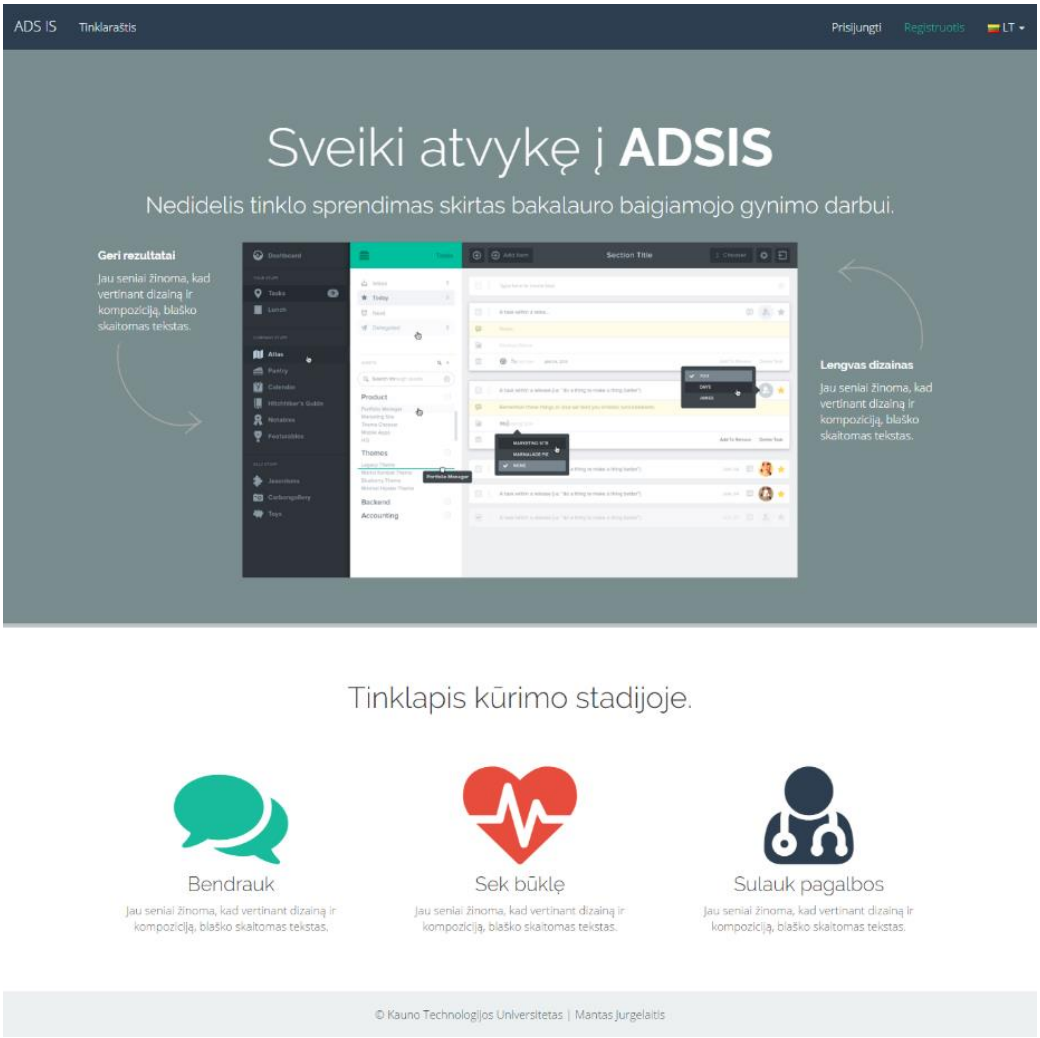
Panaudojimo atvejų diagramos atitikimas apskaičiuojamas palyginant sugeneruotos panaudojimo atvejų diagramos elementus, kaip pavyzdžiui aktoriai ir panaudojimo atvejai, su elementais originalioje diagramoje, ir, suskaičiavus sutapimą, išreiškiamas procentais. Įvertinant panaudojimo atvejų diagramos svarbą sistemos reikalavimams ir veikimui aprašyti, kaip kriterijus buvo pasirinktas 80 % atitikimas tarp originalios dokumentacijos ir sugeneruotos diagramos. 100% atitikimo tikėtis nėra verta, dėl galimybės panaudojimo atvejų diagramoje tą pačią dalykinę sritį aprašyti skirtingu abstrakcijos lygiu.

Atitikimas veiklos diagramose skaičiuojamas tik sistemos naudotojo veiksmams. Nors algoritmas ir geba sugeneruoti veiksmus, aprašančius sistemos veikimą, tačiau jie generuojami tik darant prielaidą, kad analizuojama sistema reaguoja į užregistruotą naudotojo veiksmą. Labai detalių veiklos diagramų, aprašančių sistemos veikimą, nesitikima sukurti vien tik analizuojant sistemos grafinę naudotojo sąsają ir registruojant veiksmus. Tam reikėtų papildomų duomenų, kurių darbe pateikiamas algoritmas šiuo metu negali įvertinti. Taip pat svarbu paminėti, kad veiklos diagramų neatitikimas galimas ir dėl skirtingo abstrakcijos lygio pasirinkimo. Todėl iš sugeneruotų veiklos diagramų tikimasi bent 50 % atitikimo.

## 5.2. Pirmasis eksperimentas – „WEB2UML“ kiekybinis vertinimas

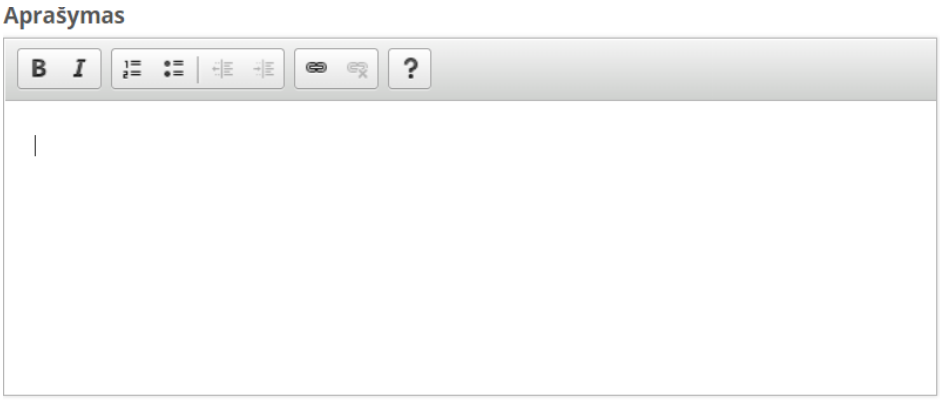
### 5.2.1. „ADSIS“ analizės rezultatai

Pirmasis eksperimentas buvo atliktas registruojant veiksmus, atliekamus „Informacinėje sistemoje sergantiems atopiniu dermatitu“ – „ADSIS“ 5.1 pav. Šioje sistemoje egzistuoja keturių tipų naudotojai: informacinės sistemos naudotojas, administratorius, gydytojas ir pacientas. Informacinės sistemos naudotojas gali prisiregistruoti sistemoje, prie jos prisijungti, bei peržiūrėti ir komentuoti tinklaraščio įrašus. Administratorius „ADSIS“ sistemoje atsakingas už naudotojų administravimą, sistemos elementų valdymą ir gali peržiūrėti statistiką. Didžioji dalis naudotojų šioje sistemoje egzistuoja su paciento ir gydytojo rolėmis. Paciento rolės naudotojai atsakingi už savo būklės informacijos valdymą, komentarų pateikimą apie ligos istoriją ir gydytojų pasirinkimą. Gydytojas atsakingas už pacientų diagnozių skyrimą, medicininių rekomendacijų teikimą ir gydymo plano valdymą. Taip pat gydytojai rašo tinklaraščio įrašus. Eksperimente naudotos ir sugeneruotos UML diagramos pateikiamos 9.1.1 priede




**5.1 pav.** Informacinės sistemos „ADSIS“ grafinė naudotojo sąsaja

Vykdant eksperimentą pastebėta, kad sistemoje neužregistruojamas duomenų įvedimas, naudojantis tekstiniu redaktoriumi, kuris realizuotas naudojantis <*iframe*> *HTML* žyma (pavyzdys – 5.2 pav.). Taip pat pastebėta nesutapimų tarp sistemos ir jos dokumentacijos, kai nebuvo realizuotas naudotojų informacijos keitimo funkcionalumas.




**5.2 pav.** Informacinės sistemos „ADSIS“ <*iframe*> elemento pavyzdys

Sugeneravus panaudojimo atvejų modelį „ADSIS“ buvo sugeneruoti 25 panaudojimo atvejai. Detalūs sugeneruotų panaudojimo atvejų rezultatai pateikiami 5.1 lentelėje, 5.3 lentelėje ir 5.3 pav. Nustatyta, kad panaudojimo atvejų diagramų atitikimo vidurkis yra 100%. Sugeneruotų

panaudojimo atvejų skaičius skiriasi nuo originalaus projekto dėl naujų panaudojimo atvejų sugeneravimo, nustatant <<*include*>> ir <<*extend*>> ryšius.

**5.1 lentelė. „ADSIS“ panaudojimo atvejų diagramos tyrimo rezultatai**

| **Kriterijai panaudojimo atvejų diagramų atitikimo vertinimui** | **Rezultatas** |
|---|---|
| Suprojektuotų panaudojimo atvejų skaičius | 16 |
| Sugeneruotų panaudojimo atvejų skaičius | 25 |
| Rastų sutampančių panaudojimo atvejų skaičius | 16 |
| Rastų sutampančių panaudojimo atvejų padengimas, procentais | 100% |
| Suprojektuotų aktorių skaičius | 4 |
| Sugeneruotų aktorių skaičius | 4 |
| Aktorių padengimas procentais | 100% |
| Suprojektuotų apibendrinimo (angl. *generalization*) ryšių tarp aktorių skaičius | 3 |
| Sugeneruotų apibendrinimo (angl. *generalization*) ryšių tarp aktorių skaičius | 3 |
| Sutampančių apibendrinimo ryšių skaičius, procentais | 100% |
| **Vidutinis panaudojimo atvejų diagramų atitikimas, procentais** | 100% |

**5.2 lentelė. „ADSIS“ panaudojimo atvejų tarpusavio ryšių tyrimo rezultatai**

| **Kriterijai panaudojimo atvejų tarpusavio ryšių atitikimo vertinimui** | **Rezultatas** |
|---|---|
| Suprojektuotų išplėtimo (angl. *extend*) ryšių skaičius | 4 |
| Sugeneruotų išplėtimo (angl. *extend*) ryšių skaičius | 3 |
| Suprojektuotų apėmimo (angl. *include*) ryšių skaičius | 1 |
| Sugeneruotų apėmimo (angl. *include*) ryšių skaičius | 9 |

Eksperimente sugeneruoti panaudojimo atvejai padengė visus originaliai suprojektuotus panaudojimo atvejus, kai kur net detalesniu lygmeniu. Tai, kas originalioje dokumentacijoje buvo aprašyta vienu panaudojimo atveju kai kuriais atvejais buvo sugeneruota išskaidant į keletą panaudojimo atvejų sujungtų <<*include*>> ar <<*extend*>> ryšiais. Sugeneruota panaudojimo atvejų diagrama taip pat turi identiškus aktorius ir aktorių apibendrinimo ryšius, lyginant su dokumentacijoje aprašytais.

Išplėtimo <<*extend*>> ir apėmimo <<*include*>> ryšių skaičiaus negalime vertinti tik palygindami juos originalioje ir sugeneruotoje diagramoje, dėl to, kad jų skaičius priklauso nuo panaudojimo atvejų diagramos detalumo lygio. Kaip matome, darbe sukurtas metodas ir įrankis nustatė daugiau apėmimo ryšių, nei išplėtimo ryšių.

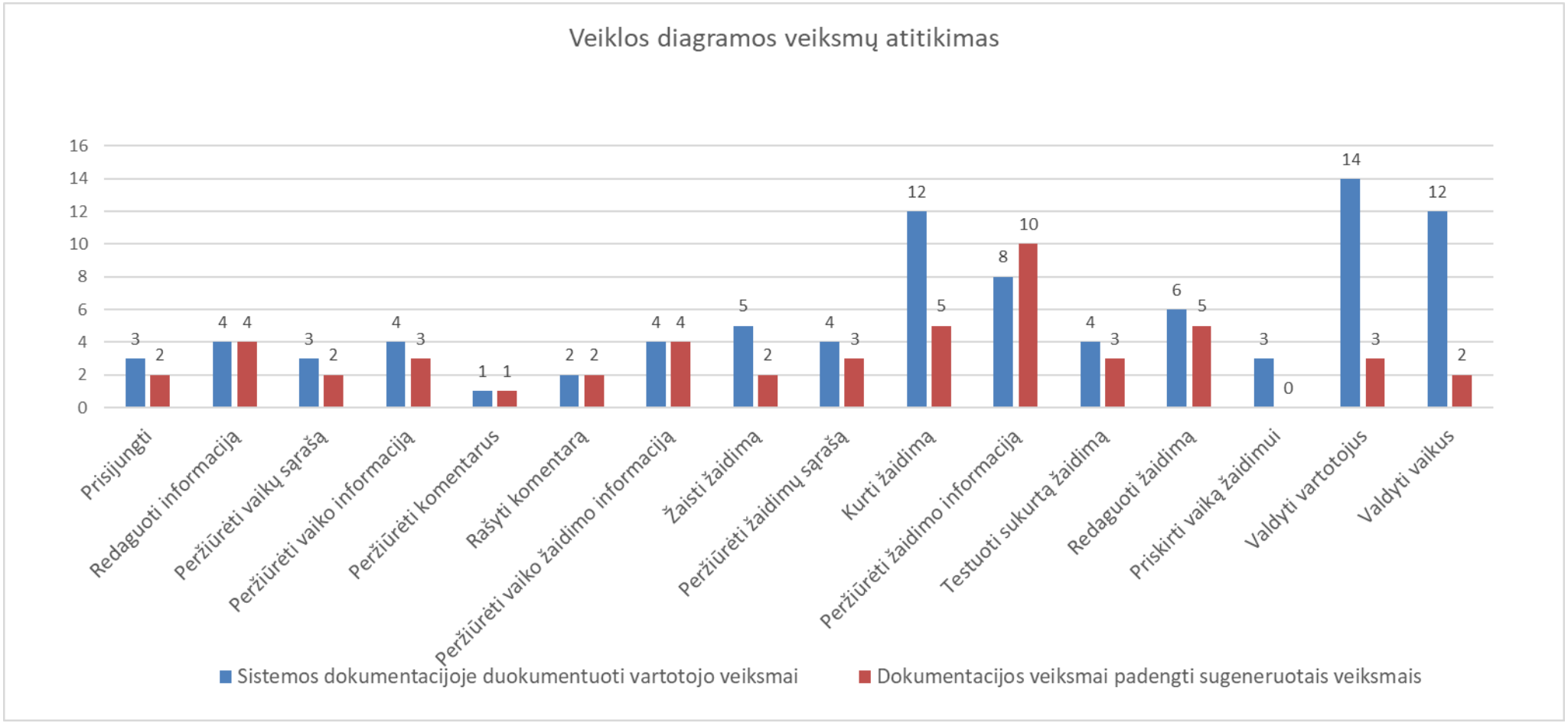

**5.3 pav. „**ADSIS“ panaudojimo atvejų diagramų elementų atitikimo rezultatai

Be panaudojimo atvejų diagramos, kiekvienas panaudojimo atvejis sugeneruotame rezultate buvo detalizuotas veiklos diagrama. Sugeneruotos veikos diagramos aprašo aktoriaus atliekančio panaudojimo atvejį veiklą ir nuspėjamą sistemos veiklą. Kiekviena veiklos diagrama buvo įvertinta ją palyginant su originaliąją panaudojimo atvejį detalizuojančią diagrama. Šio palyginimo rezultatai pateikti 5.3 lentelėje bei 5.4 pav. Sugeneruotų veiklos diagramų elementų sistemai „ADSIS“ ir jos originaliame modelyje egzistuojančių elementų skaičiaus palyginimas Susumavus rezultatus buvo nustatytas 75 % naudotojo veiksmų atitikimas ir 34% atitikimas sistemos pusėje. Didžioji dalis neatitikimų naudotojo dalyje, atsirado dėl veiksmų, kaip pavyzdžiui „Peržiūrėt tinklaraščio įrašus“, kurie nenurodo veiksmo sistemoje ir todėl negali būti užregistruoti. Trisdešimt keturių procentų atitikimas tarp sistemos veiksmų yra tenkinamas rezultatas žinant, kad nėra įmanoma nustatyti sistemos veikimo vien iš užregistruotų sistemos veiksmų ir sistemos grafinės naudotojo sąsajos. Dalis neatitikimų tarp sistemoje įvykdytų veiksmų ir sugeneruotų rezultatų atsirado dėl jau minėto <<*iframe*>> elemento naudojimo sistemoje. Taip pat sugeneruotos veiklos diagramos tiksliai nurodo kokius duomenis reikia įvesti pateikiant informaciją sistemai, ko nebuvo originalioje dokumentacijoje.

Apibendrindami galime teigti, kad sukurtas metodas ir sistema tenkina keliamus reikalavimus teisingam panaudojimo atvejų modelio generavimui ir padeda nustatyti neatitikimus tarp sistemos projekto ir veikiančios sistemos.

**5.3 lentelė. „ADSIS“ veiklos diagramų tyrimo rezultatai**

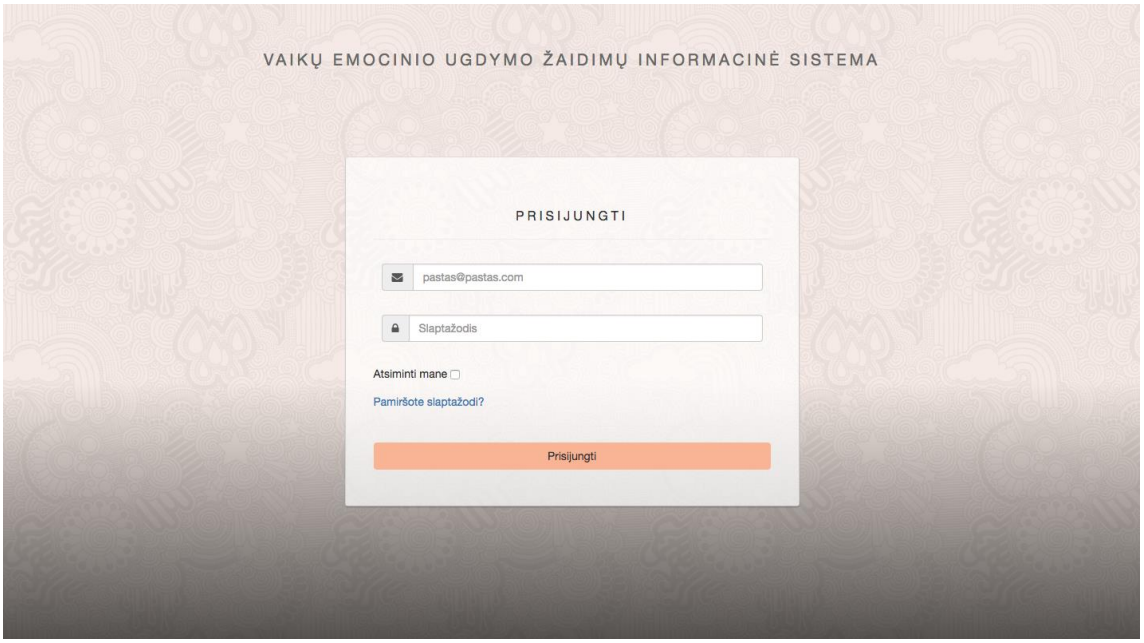
| Veiklos diagramų įvertinimas | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Naudotojo veiksmų vertinimas | | | | | Sistemos veiksmų vertinimas | | |
| **Panaudojimo atvejis** | **Nr.** | **Registracijos metu atlikti veiksmai** | **Dokumentacijoje aprašyti naudotojo veiksmai** | **Sugeneruoti naudotojo veiksmai** | **Dokumentacijoje aprašyti naudotojo veiksmai, turintys sugeneruotus atitikmenis** | **Dokumentacijoje aprašyti sistemos veiksmai** | **Sugeneruoti sistemos veiksmai** | **Dokumentacijoje aprašyti sistemos veiksmai turinys sugeneruotus atitikmenis** |
| Prisijungti | 1 | 5 | 2 | 5 | 2 | 5 | 2 | 2 |
| Registruotis | 2 | 6 | 2 | 6 | 2 | 2 | 2 | 2 |
| Peržiūrėti tinklaraščio įrašus | 3 | 2 | 10 | 2 | 8 | 8 | 6 | 5 |
| Valdyti komentarus | 4 | 6 | 11 | 6 | 8 | 5 | 4 | 2 |
| Administruoti naudotojus | 5 | 4 | 6 | 4 | 3 | 8 | 4 | 1 |
| Valdyti sistemos elementus | 6 | 30 | 7 | 28 | 7 | 8 | 8 | 2 |
| Peržiūrėti statistiką | 7 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| Peržiūrėti pacientų sąrašą | 8 | 1 | 4 | 1 | 1 | 2 | 1 | 1 |
| Peržiūrėti ligos istoriją | 9 | 30 | 25 | 29 | 20 | 1 | 1 | 1 |
| Valdyti diagnozes | 10 | 15 | 7 | 13 | 5 | 8 | 4 | 2 |
| Valdyti medicinines rekomendacijas | 11 | 9 | 7 | 9 | 5 | 8 | 4 | 2 |
| Valdyti gydymo planą | 12 | 19 | 7 | 17 | 5 | 8 | 4 | 2 |
| Valdyti tinklaraščio įrašus | 13 | 12 | 7 | 12 | 5 | 8 | 6 | 2 |
| Tvarkyti informaciją apie būklę | 14 | 23 | 7 | 23 | 5 | 8 | 8 | 2 |
| Valdyti ligos istorijos komentarus | 15 | 9 | 7 | 9 | 5 | 4 | 7 | 1 |
| Pridėti gydytoją prie gydytojų sąrašo. | 16 | 5 | 3 | 3 | 3 | 2 | 2 | 2 |
| **Viso:** | | **150** | **114** | **169** | **85** | **86** | **64** | **30** |
| **Naudotojo veiksmų atitikimo procentas** | **75%** | | | | | | | |
| **Sistemos veiksmų atitikimo procentas** | **34%** | | | | | | | |




**5.4 pav. Sugeneruotų veiklos diagramų elementų sistemai „ADSIS“ ir jos originaliame modelyje egzistuojančių elementų skaičiaus palyginimas**

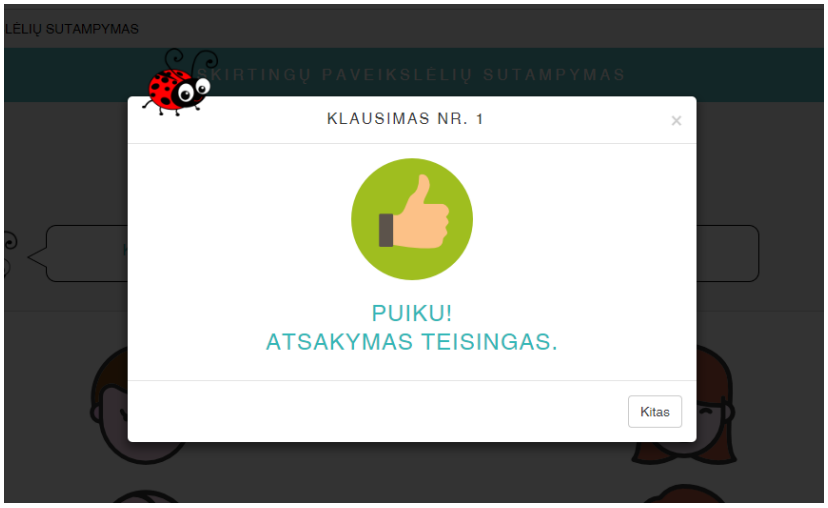
### 5.2.2. „VEUŽIS“ analizės rezultatai

Pirmojo eksperimento antroji dalis buvo atlikta registruojant veiksmus atliekamus „Vaikų emocinio ugdymo žaidimų informacinėje sistemoje“ – VEUŽIS 5.5 pav. Šioje sistemoje egzistuoja 5 tipų naudotojai: informacinės sistemos naudotojas, tėvas, vaikas, specialistas ir sistemos administratorius. Informacinės sistemos naudotojas gali prisijungti prie sistemos bei redaguoti profilio informaciją. Administratorius VEUŽIS sistemoje atsakingas už naudotojų bei vaikų administravimą. Specialistas šioje sistemoje gali valdyti žaidimus ir jiems priskirti vaikus. Tėvas šioje sistemoje gali peržiūrėti vaiko žaidimų informaciją, pateikti komentarus apie vaiko žaidimus. Galiausiai vaikas gali žaisti žaidimus sistemoje. Eksperimente naudotos ir sugeneruotos UML diagramos pateikiamos 9.1.2 priede




**5.5 pav.** Informacinės sistemos „VEUŽIS“ grafinė naudotojo sąsaja

Pradėjus registruoti veiksmus sistemoje, pastebėta, kad nepavyksta užregistruoti dinaminių turinio generuojamų *JavaScript*, todėl dalis sistemos funkcijų nebuvo pilnai registruotos (pvz. 5.6 pav.). Taip pat pastebėta, kad, remiantis originalia panaudojimo atvejų diagrama, administratoriui nėra galimybės prisijungti prie sistemos. Nors sistemoje prisijungti privalo visi naudotojai. Siūlomą šią klaidą ištaisyti. Tai dar kartą parodo, kad sistemos modelis ir veikianti sistema dažnu atveju nesutampa.




**5.6 pav.** Informacinės sistemos „VEUŽIS“ dinamiškai kuriamo turinio pavyzdys

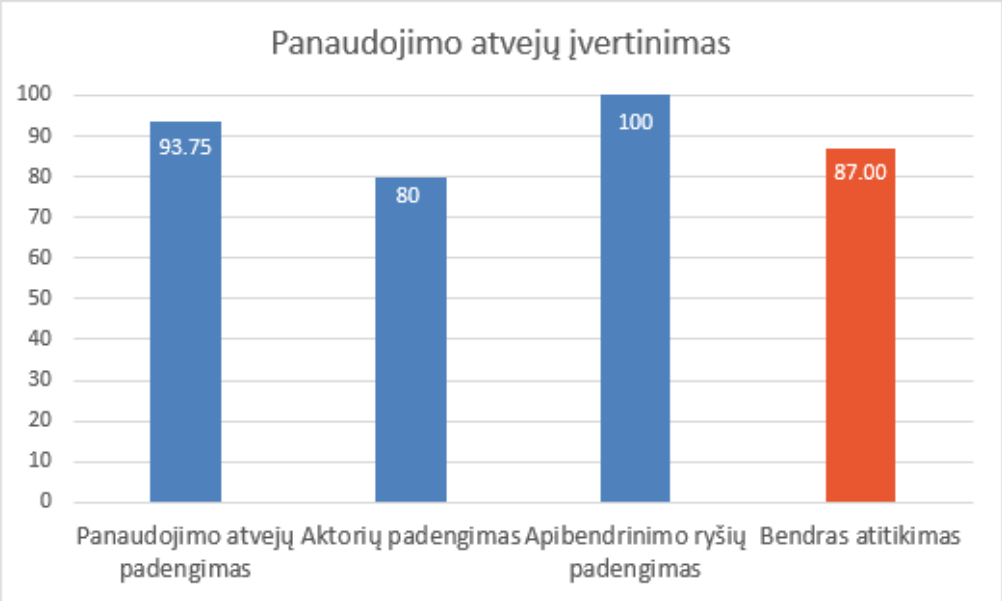
Sugeneravus panaudojimo atvejų modelį „VEUŽIS“ sistemai buvo nustatyta, kad net ir nesant galimybei užregistruoti visų veiksmų sistemoje sugeneruotos panaudojimo atvejų diagramos atitikimo procentas pasiekė 87%. Pilnus rezultatus galima peržvelgti 5.4 lentelė ir 5.5 lentelėse ir 5.7 pav. Sugeneruotų panaudojimo atvejų skaičius, tik nežymiai viršijo suprojektuotų panaudojimo atvejų skaičių. Tačiau, registravimo metu neužregistruota vaiko rolė sistemoje. Dėl ypatybės, kad norint įgyvendinti panaudojimo atvejį „Žaisti žaidimą“ jame turi dalyvauti du aktoriai, to įgyvendinti nepavyko, nes sistema leidžia užregistruoti tik vieno aktoriaus veiksmus vienu metu. Žinant šią sistemos ypatybę analitikas gavęs sugeneruotą panaudojimo atvejų diagramą lengvai galėtų ją pakoreguoti ir pridėti naują aktorių. Atsižvelgiant į klaidą originaliojoje diagramoje, apibendrinimo ryšių buvo sugeneruota daugiau nei buvo sukurta originalioje diagramoje. Tai tik dar kartą pabrėžia faktą, kad net ir prieš metus kurtoje sistemoje galima rasti neatitikimų tarp veikiančios sistemos ir jo modelio. Išplėtimo ir apėmimo ryšių skaičius sugeneruotas artimai atitinkant esamą sistemos dokumentaciją.

**5.4 lentelė. „VEUŽIS“ panaudojimo atvejų diagramos tyrimo rezultatai**

| **Panaudojimo atvejų diagramos parametras** | **Rezultatas** |
|---|---|
| Suprojektuotų panaudojimo atvejų skaičius | 16 |
| Sugeneruotų panaudojimo atvejų skaičius | 18 |
| Rastų sutampančių panaudojimo atvejų skaičius | 15 |
| Rastų sutampančių panaudojimo atvejų padengimas procentais | 94% |
| Suprojektuotų aktorių skaičius | 5 |
| Sugeneruotų aktorių skaičius | 4 |
| Aktorių padengimas procentais | 80% |
| Suprojektuotų apibendrinimo ryšių skaičius (angl. generalization) | 2 |
| Sugeneruotų apibendrinimo ryšių skaičius (angl. generalization) | 3 |
| Sutampančių apibendrinimo ryšių skaičius išreikštas procentais | 100% |
| **Vidutinis panaudojimo atvejų diagramų atitikimas, procentais** | 87% |

**5.5 lentelė „VEUŽIS“ panaudojimo atvejų tarpusavio ryšių tyrimo rezultatai**

| **Kriterijai panaudojimo atvejų tarpusavio ryšių atitikimo vertinimui** | **Rezultatas** |
|---|---|
| Suprojektuotų išplėtimo ryšių skaičius (angl. extend) | 7 |
| Sugeneruotų išplėtimo ryšių skaičius (angl. extend) | 6 |
| Suprojektuotų apėmimo ryšių skaičius (angl. include) | 2 |
| Sugeneruotų apėmimo ryšių skaičius (angl. include) | 3 |



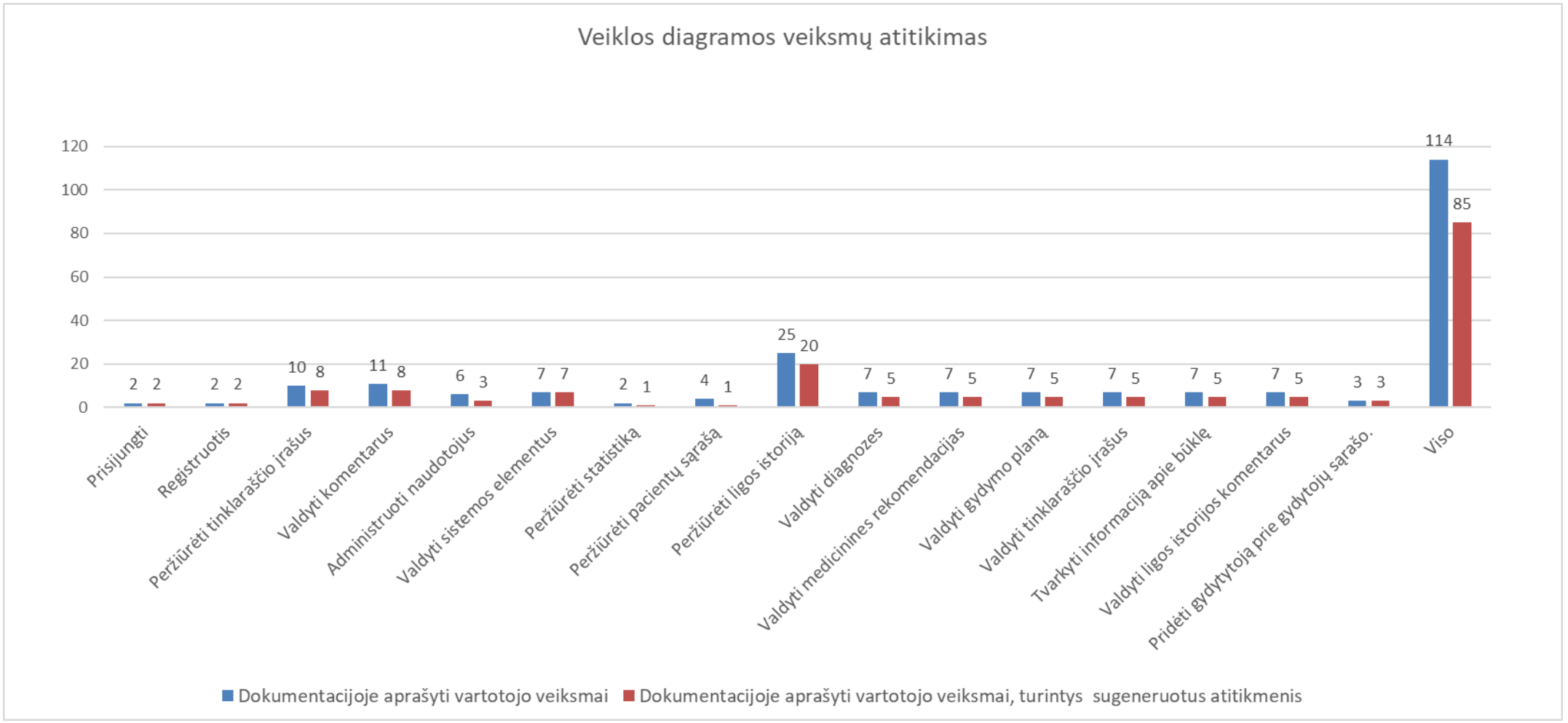

**5.7 pav.** „VEUŽIS“ panaudojimo atvejų diagramos elementų atitikimo rezultatas

Kaip ir pirmojo eksperimento metu, kiekvienas panaudojimo atvejis panaudojimo atvejų diagramoje buvo detalizuotas veiklos diagrama. Veiklos diagramos atitikimas originaliam modeliui pateikiamas 5.6 lentelėje. Galime pastebėti, kad naudotojo veiksmų sistemoje atitikimas siekia 57%, taip yra todėl, kad šios sistemos veiklos diagramos buvo aprašytos daug detaliau nei pirmosios sistemos metu. Taip pat tokį atitikimą lėmė sistemos registravimo komponento negebėjimas užregistruoti dinamiškai kuriamų sistemos elementų. Sistemos veiksmų nustatymo procentas lyginant su pirmąja sistema padidėjo iki 38%. Tai tikrai geras rezultatas žinant, kad nėra galimybės registruojant naudotojo veiksmus sistemoje nustatyti sistemos veikimo ypatybių.

Atlikus eksperimentą su „VEUŽIS“ sistema, pastebėta sistemos registravimo komponente egzistuojančių tobulintinų vietų, tačiau net ir su nepilnai užregistruotu sistemos naudojimosi scenarijumi buvo pasiekta užsibrėžtas reikalavimas, keliamas panaudojimo atvejų modelį aprašančioms veiklos diagramoms.

**5.6 lentelė „VEUŽIS“ veiklos diagramų tyrimo rezultatai**

| Veiklos diagramų įvertinimas | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Naudotojo veiksmų vertinimas | | | | Sistemos veiksmų vertinimas | | |
| Panaudojimo atvejis | Nr. | Sistemoje atlikti veiksmai | Sistemos dokumentacijoje dokumentuoti naudotojo veiksmai | Sugeneruoti naudotojo veiksmai | Dokumentacijos veiksmai padengti sugeneruotais veiksmais | Sistemoje dokumentuoti sistemos veiksmai | Sugeneruoti sistemos veiksmai | Sistemos veiksmai turinys atitikmenis |
| Prisijungti | 1 | 2 | 3 | 3 | 2 | 4 | 1 | 0 |
| Redaguoti informaciją | 2 | 10 | 4 | 10 | 4 | 4 | 3 | 3 |
| Peržiūrėti vaikų sąrašą | 3 | 2 | 3 | 2 | 2 | 1 | 2 | 1 |
| Peržiūrėti vaiko informaciją | 4 | 3 | 4 | 3 | 3 | 4 | 2 | 2 |
| Peržiūrėti komentarus | 5 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |
| Rašyti komentarą | 6 | 4 | 2 | 3 | 2 | 2 | 1 | 1 |
| Peržiūrėti vaiko žaidimo informaciją | 7 | 4 | 4 | 4 | 4 | 2 | 2 | 2 |
| Žaisti žaidimą | 8 | 11 | 5 | 5 | 2 | 5 | 4 | 2 |
| Peržiūrėti žaidimų sąrašą | 9 | 4 | 4 | 4 | 3 | 1 | 3 | 1 |
| Kurti žaidimą | 10 | 11 | 12 | 8 | 5 | 6 | 1 | 1 |
| Peržiūrėti žaidimo informaciją | 11 | 20 | 8 | 17 | 10 | 4 | 1 | 1 |
| Testuoti sukurtą žaidimą | 12 | 8 | 4 | 4 | 3 | 4 | 1 | 1 |
| Redaguoti žaidimą | 13 | 10 | 6 | 7 | 5 | 3 | 1 | 0 |
| Priskirti vaiką žaidimui | 14 | 6 | 3 | 2 | 0 | 3 | 2 | 2 |
| Valdyti naudotojus | 15 | 18 | 14 | 6 | 3 | 7 | 3 | 3 |
| Valdyti vaikus | 16 | 16 | 12 | 8 | 2 | 8 | 2 | 2 |
| **Viso** | | **131** | **89** | **88** | **51** | **59** | **30** | **23** |
| **Naudotojo veiksmų atitikimo procentas** | **57%** | | | | | | | |
| **Sistemos veiksmų atitikimo procentas** | **38%** | | | | | | | |




**5.8 pav. Sugeneruotų veiklos diagramų elementų sistemai „VEUŽIS“ ir jos originaliame modelyje egzistuojančių elementų skaičiaus palyginimas**

### 5.3. Antrasis eksperimentas – „WEB2UML“ kokybinis įvertinimas

Pirmojo eksperimento metu išsiaiškinome, kad naujai kuriamas metodas ir įrankis leidžia kurti panašios apimties ir atitikimo, kaip ir veikiančių sistemų, panaudojimo atvejų modelius. Tačiau nebuvo galimybės tiksliau įvertinti sugeneruojamo modelio kokybinių aspektų. Šiam kriterijui įvertinti buvo vykdomas antrasis eksperimentas. Jo metu sugeneruoti du pavyzdiniai panaudojimo atvejų modeliai, kurie buvo pateikti įvertinti ekspertams, turintiems UML modeliavimo patirties ir/arba OMG UML sertifikatus.

Pirmojoje antrojo eksperimento dalyje buvo tiriamas sistemos veiksmų registravimas *moodle.if.ktu.lt* sistemai. Šioje sistemoje buvo siekiama užregistruoti dažnai pasitaikančius studentų ir dėstytojų veiksmus viename kurse. Visos šioje eksperimento dalyje vertintos diagramos pateikiamos 9.1.3 priede. Eksperimento metu savo veiksmus sistemoje užregistravo dėstytojas ir studentas:

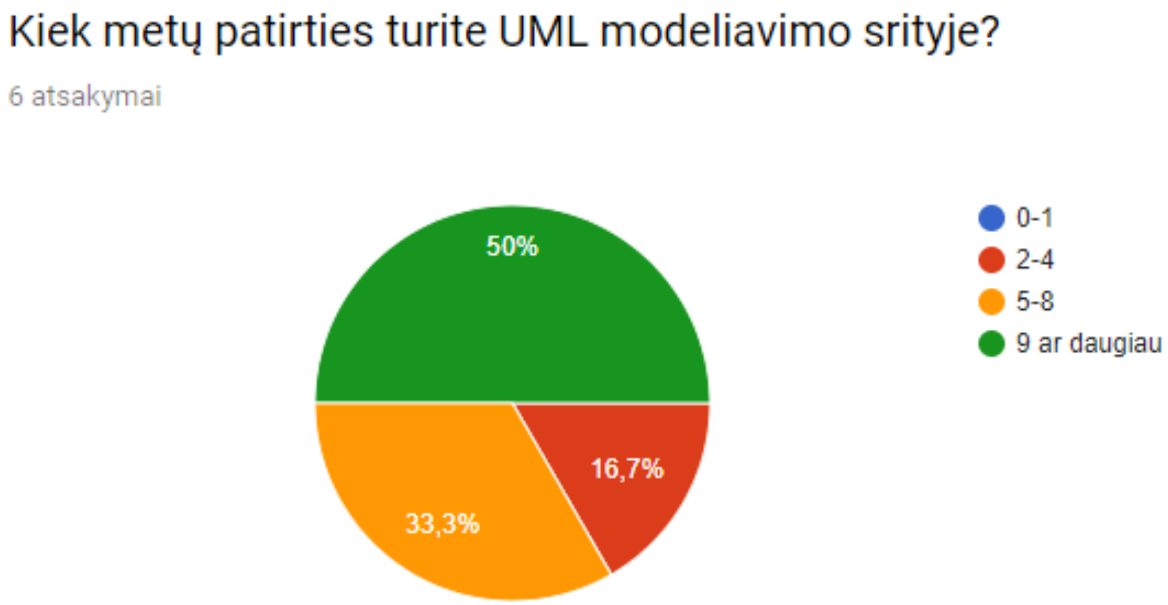
- Studentas prisijungė prie sistemos naudodamas bendrą universitetinę prisijungimo sistemą. Prisijungęs prie sistemos, studentas parsisiuntė modulio medžiagą. Taip pat jis išsprendė testą, kurį sudarė 10 klausimų. Pabaigęs pildyti testą, studentas pateikė atsiliepimą apie kursą.
- Dėstytojas eksperimento metu prisijungė prie sistemos, parsisiuntė kurso medžiagą, pateikė atsiliepimą apie kursą ir išsprendė tą patį testą, kaip ir studentas. Taip pat jis peržiūrėjo šio kurso statistiką. Dėstytojas taip pat pridėjo naują kursui reikalingą medžiagos failą, bei kitą nereikalingą failą pašalino iš sistemos. Prieš užbaigdamas darbą sistemoje, dėstytojas pakeitė vieno studento prisiregistravimo prie kurso datą.

Antrojoje antrojo eksperimento dalyje buvo tiriamas mokslininkams skirtas socialinis tinklas *researchgate.net*. Visos šioje eksperimento dalyje vertintos diagramos pateikiamos 9.1.4 priede. Eksperimento metu sistema naudojosi du mokslininkai: vienas jų vykdė tik peržiūros veiksmus (skaitytojas), o kitas – straipsnių įkėlimą (autorius):

- Skaitytojas prisijungė, ieškojo straipsnių pagal pasirinktą kriterijų ir peržiūrėjo gautą sąrašą bei peržiūrėjo siūlomus darbo skelbimus. Taip pat jis paprašė prieigos prie pasirinkto straipsnio pilnos medžiagos.
- Autorius įgyvendino visus skaitytojo veiksmus, pridėjo straipsnį ir jo failą bei pridėjo straipsnį be failo. Autorius vieną straipsnį pašalino. Taip pat jis pakeitė savo profilio nustatymus.

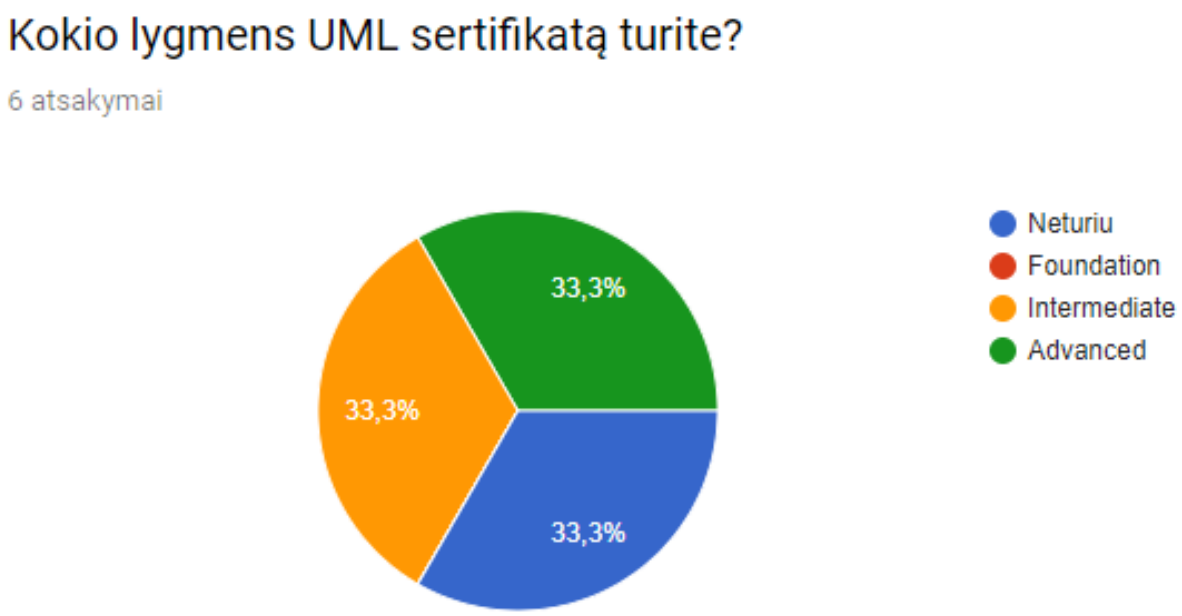
Šie scenarijai buvo atlikti siekiant užregistruoti naudotojų veiksmus *moodle.if.ktu.lt* ir *researchgate.net* sistemose. Iš gautų veiksmų registracijos duomenų buvo sugeneruoti panaudojimo atvejų modeliai šioms sistemoms ir pateikti įvertinti ekspertams.

Eksperimento metu šeši ekspertai užpildė apklausą, kurios metu atsakė į klausimus apie dviejų panaudojimo atvejų modelio kokybę. Pirmojoje apklausos dalyje ekspertai turėjo aprašyti savo turimą patirtį dirbant su UML ir nurodyti kokį turi ar neturi UML sertifikatą. Suskaičiavus rezultatus pastebėta, kad didžioji dalis apklausoje dalyvavusių dalyvių turėjo 5 ar daugiau metų patirtį. Dalyvių patirtis UML modeliavimo srityje atvaizduota 5.9 pav.




**5.9 pav.** Apklausos dalyvių pasiskirstymas pagal patirtį UML modeliavimo srityje

Daugiau nei pusė apklausos dalyvių buvo sertifikuoti UML ekspertai turinys OCUP 2 sertifikatus. Trečdalis apklausos dalyvių turėjo *Intermediate* lygmens sertifikatą, kitas trečdalis turėjo *advanced* lygio sertifikatą bei vienas trečdalis neturėjo jokio sertifikato. Kadangi sertifikatus išduoda kompanija OMG valdanti ir UML standartą galime teigti, kad šie specialistai yra aukšto kompetencijos lygio. Apklausos dalyvių pasiskirstymas pateiktas 5.10 pav.

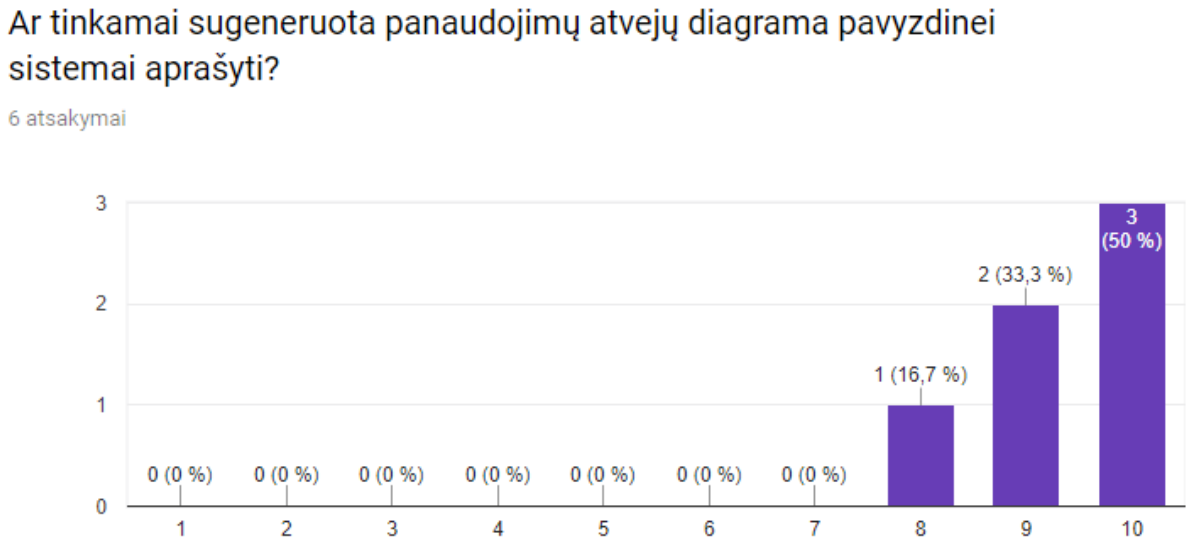



**5.10 pav.** Apklausos dalyvių pasiskirstymas pagal kompanijos OMG išduotus UML sertifikatus

### 5.3.1. Moodle.if.ktu.lt panaudojimo atvejų modelio vertinimas

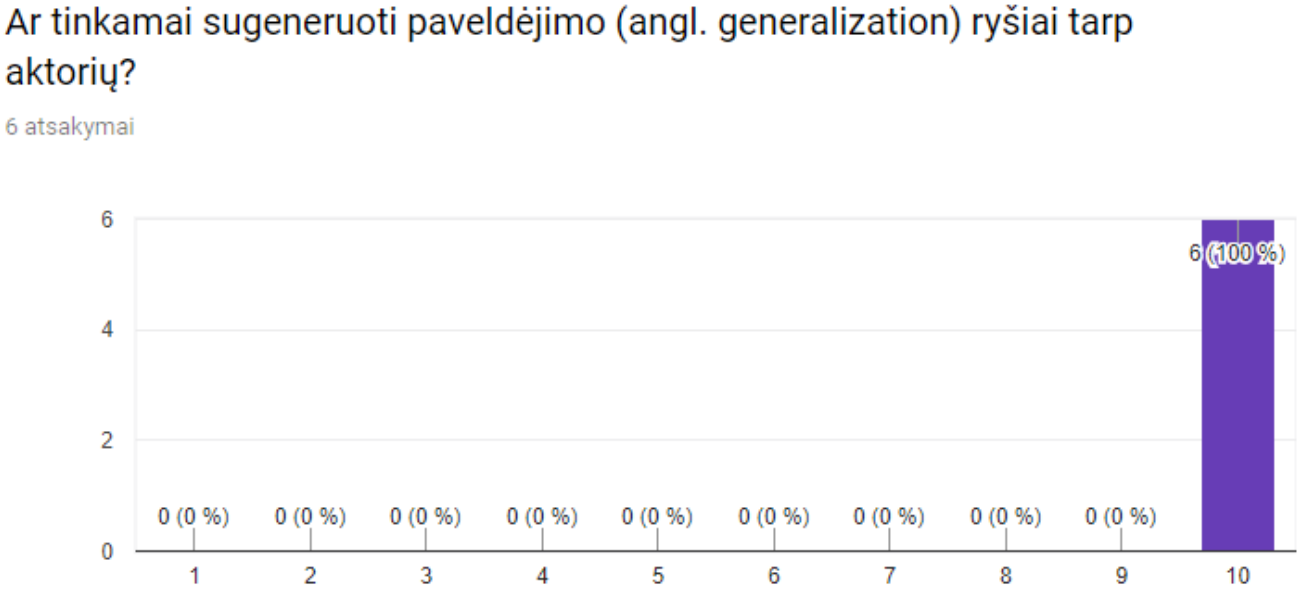
Žemiau esančiuose paveikslėliuose pateikiami apklausos dalyvių vertinimai apie įvairius panaudojimo atvejų modelio komponentus, kurie buvo sugeneruoti vykdant pavyzdinį scenarijų *moodle.if.ktu.lt* sistemoje. 5.11 pav. Atvaizduoja, kaip apklausos dalyviai vertina sugeneruotos

panaudojimo atvejų diagramos tinkamumą pavyzdinei sistemai aprašyti. Gauti rezultatai indikuoja, kad sugeneruota panaudojimo atvejų diagrama tinkamai dokumentuoja sistemos veikimą.



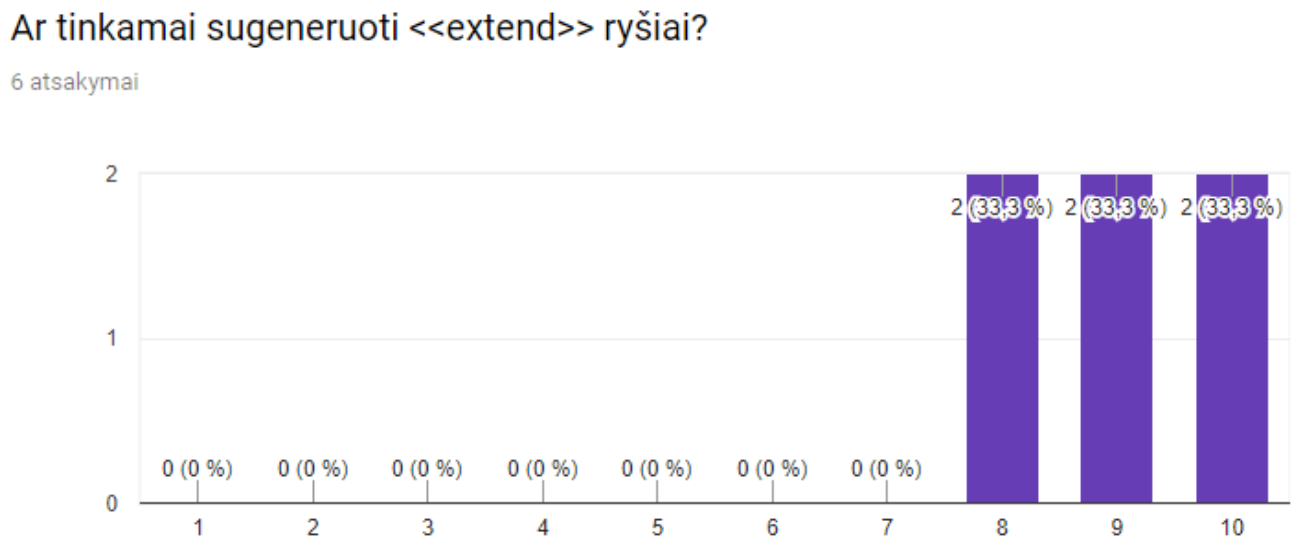

**5.11 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramos kokybę „moodle.if.ktu.lt“ sistemos atveju

Apklausos dalyviai taip pat puikiai vertina, įrankio galimybę nustatyti paveldėjimo ryšių tarp aktorių sugeneravimo funkcionalumą. Diagrama atvaizduojanti apklausos dalyvių atsakymų pasiskirstymas pateikiama 5.12 pav.



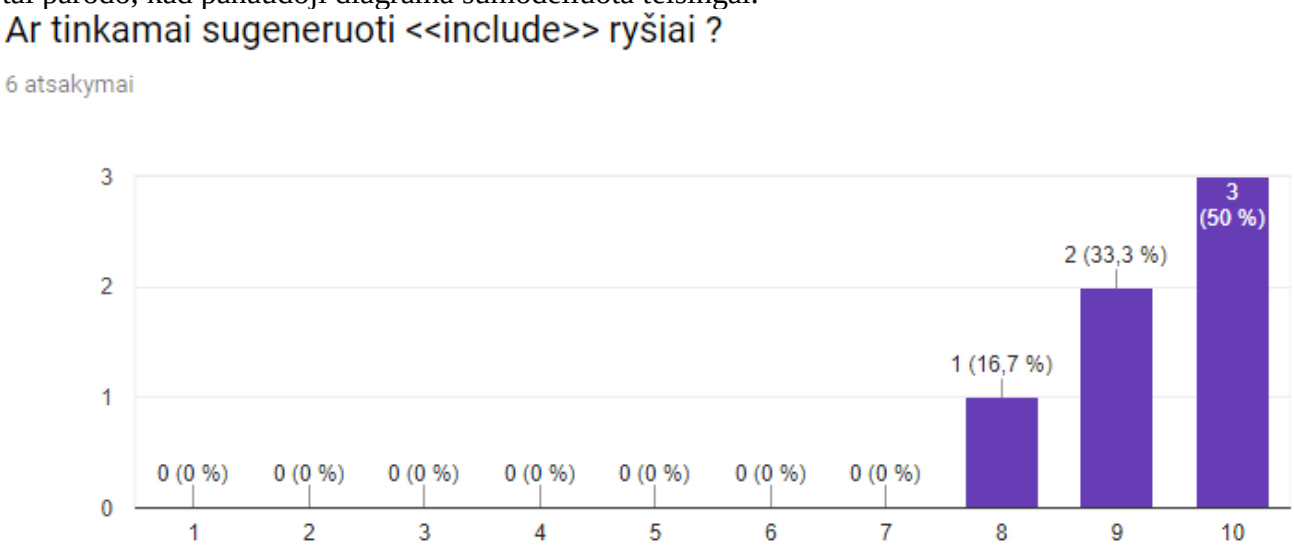

**5.12 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų paveldėjimo ryšių kokybę „moodle.if.ktu.lt“ sistemos atveju

Sistemos gebėjimą sukurti išplėtimo ryšius apklausos dalyviai vertina gerai. Diagrama atvaizduojanti dalyvių atsakymų pasiskirstymą pateikiama 5.13 pav.



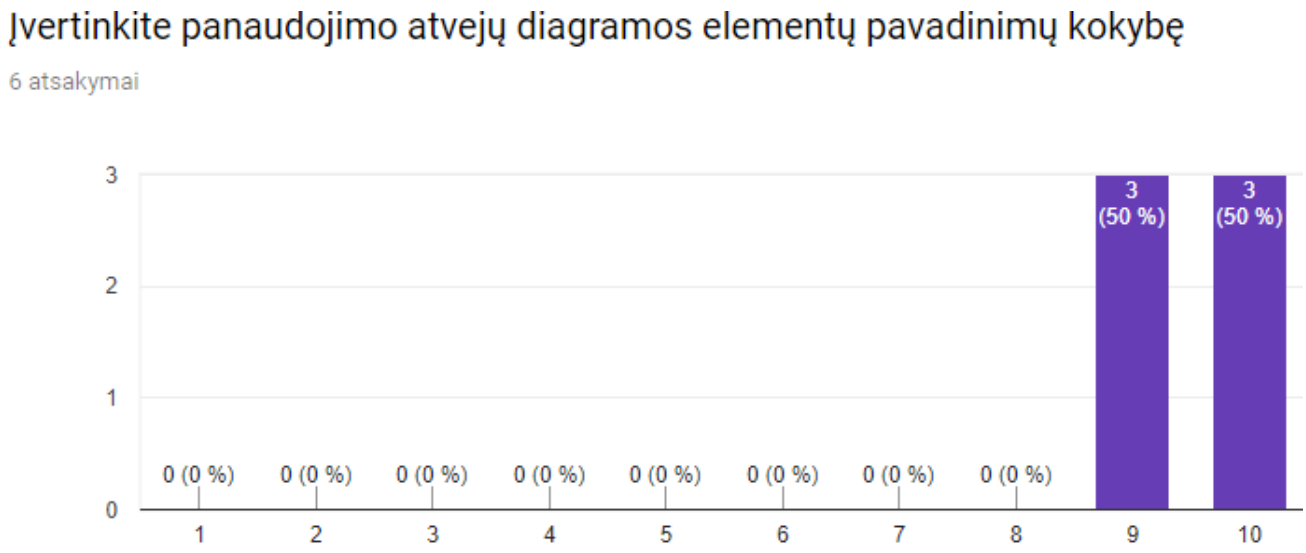

**5.13 pav. Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų išplėtimo ryšių kokybę „moodle.if.ktu.lt“ sistemos atveju.**

Kaip ir išplėtimo ryšius taip ir apėmimo ryšių generavimo galimybes apklausos dalyviai vertina nuo 8 iki 10 balų. Sugeneruotų apėmimo ryšių rezultatus galite rasti 5.14 pav. Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų apėmimo ryšių kokybę yra teigiami, tai parodo, kad panaudoji diagrama sumodeliuota teisingai.



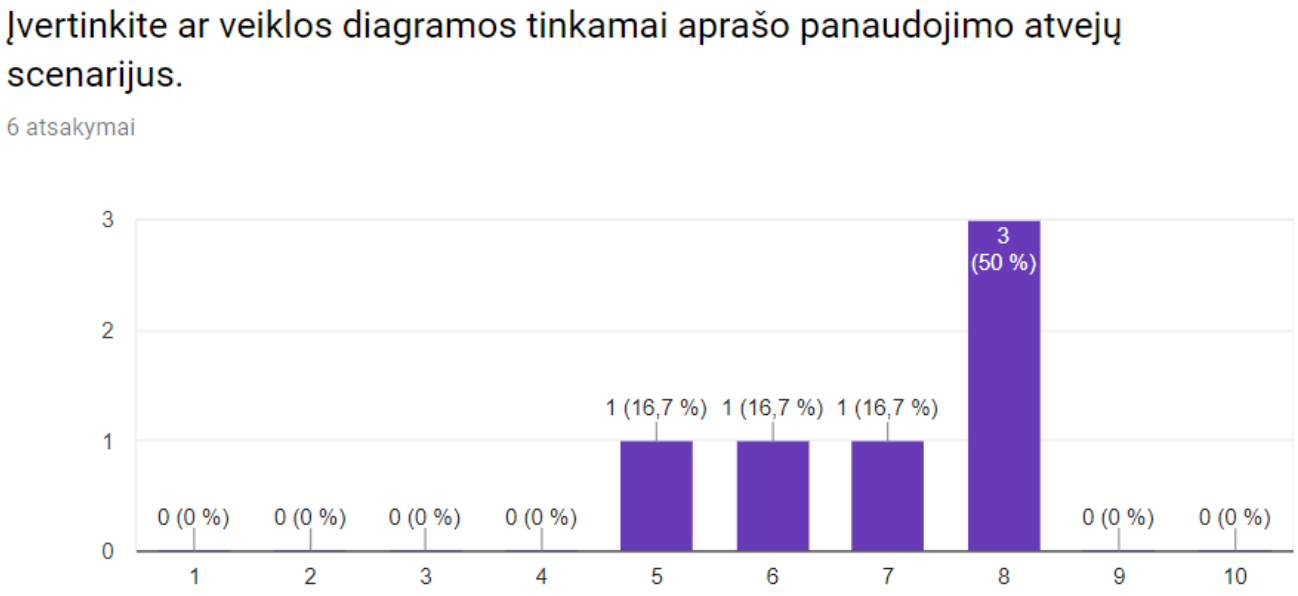

**5.14 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų apėmimo ryšių kokybę „moodle.if.ktu.lt“ sistemos atveju

Siekiant įvertinti panaudojimo modelyje esančios informacijos perteikimą analitikui, apklausos dalyvių buvo paprašyta įvertinti panaudojimo atvejų diagramoje esančius elementų pavadinimus. Šie juos, kaip ir kitus elementus įvertino gerai.



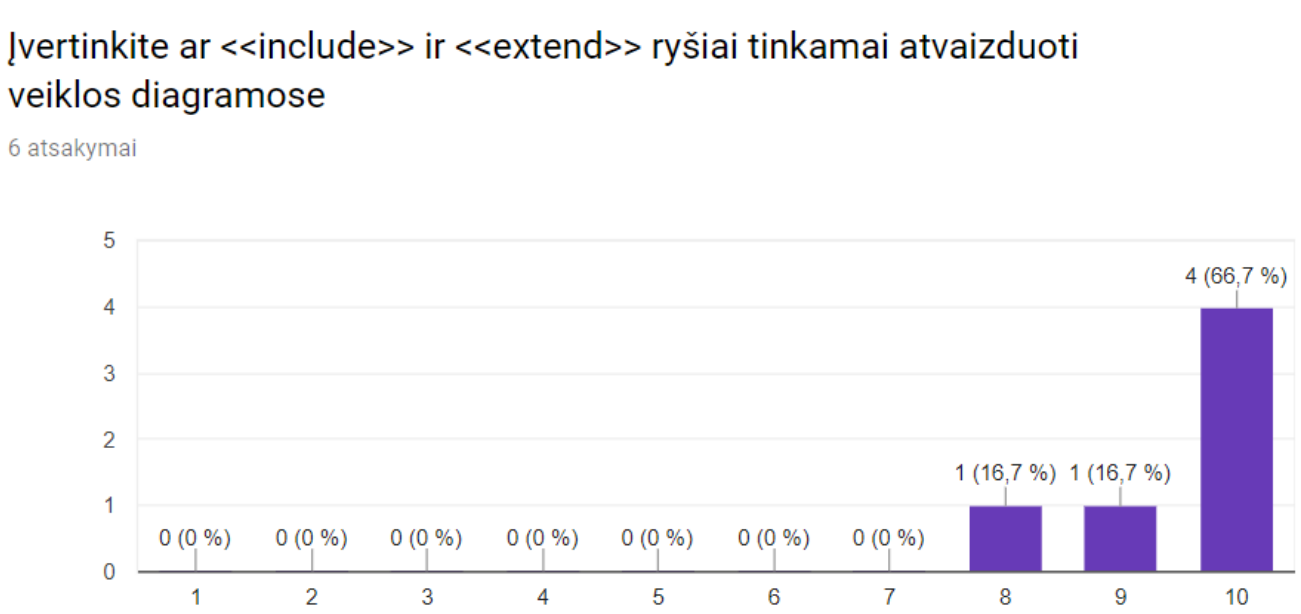

**5.15 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų elementų pavadinimų kokybę „moodle.if.ktu.lt“ sistemos atveju

Apibendrinant *moodle.if.ktu.lt* panaudojimo atvejų diagramos apklausos skyrių, galime teigti, kad apklausos dalyviai yra patenkinti sugeneruotos panaudojimo atvejų diagramos kokybe. Antrajame *moodle.if.ktu.lt* modelio įvertinimo skyriuje apklausos dalyviai įvertino veiklos diagramų detalizuojančių kiekvieną panaudojimo atvejį kokybę. Veiklos diagramose buvo prašoma įvertint bendrą teisingumą << extend>> ir <<include>> ryšių atvaizdavimą veiklos diagramose, bei veiklos diagramos elementų pavadinimų kokybę. Veiklos diagramos tinkamumo rezultatus panaudojimo atvejų diagramoms aprašyti galima rasti 5.16 pav. Kaip matome veiklos diagramų tinkamumą apibūdinti apklausos dalyviai vertina vidutiniškai.



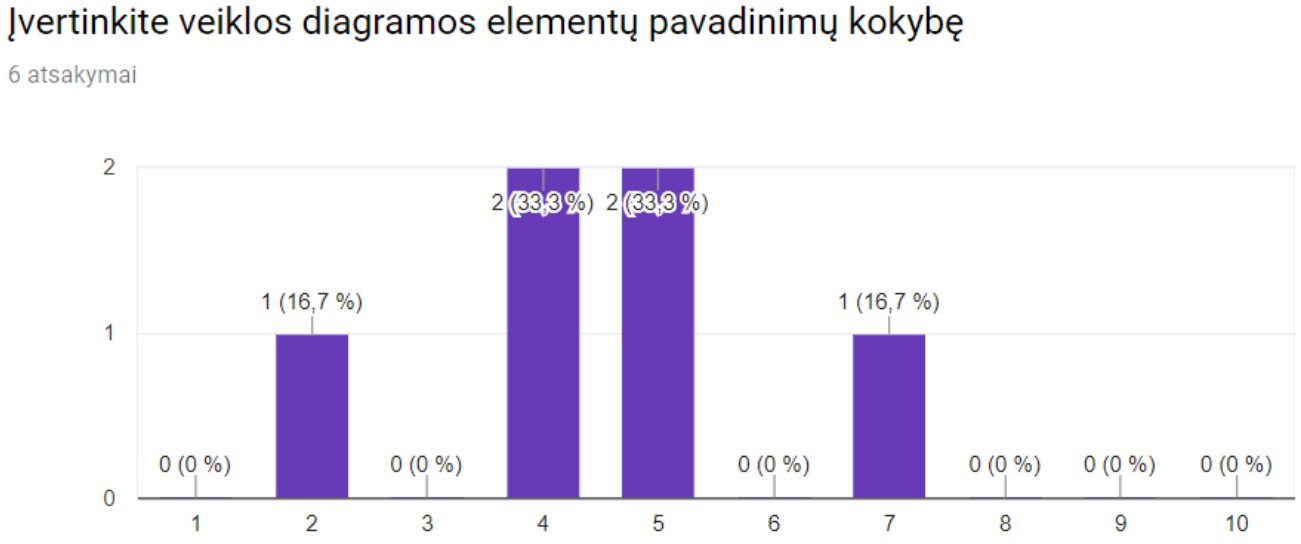

**5.16 pav.** Panaudojimo atvejus aprašančių veiklos diagramų apklausos dalyvių vertinimas „moodle.if.ktu.lt“ sistemos atveju

Atvirkščiai nei bendrą panaudojimo atvejų tinkamumą, apklausos dalyviai apėmimo ir išplėtimo ryšių atvaizdavimą veiklos diagramose vertina gerai. Detalę apklausos dalyvių atsakymų diagrama galima rasti 5.17 pav. Tai reiškia, kad naujai kuriamas įrankis ir metodas leidžia ne tik panaudojimo atvejų diagramoje sugeneruoti teisingus ryšius, bet ir juos gerai atvaizduoti veiklos diagramoje.




**5.17 pav.** Apėmimo ir išplėtimo ryšių atvaizdavimo teisingumo vertinimo rezultatai „moodle.if.ktu.lt“ sistemos atveju

Prasčiausiai iš visų apklausoje pateikiamų vertinti panaudojimo modelio elementų apklausos dalyviai įvertino veiklos diagramose esančių elementų kokybę 5.18 pav. Elementų kokybė gali būti pagerinama tobulinant veiksmų registravimo komponentą. Taip pat veiklos diagramos elementų kokybė priklauso ir nuo sistemos naudotojos sąsajos kodo – neteisingai sugeneruoti veiklos diagramos elementai gali indikuoti apie neinformatyvų ar net neteisingą sistemos grafinės naudotojos sąsajos kodą. Šiuo atveju sugeneruoti veiklos diagramos elementų pavadinimai sugeneruoti nekokybiškai, dėl sistemos registravimo komponento trūkumų.

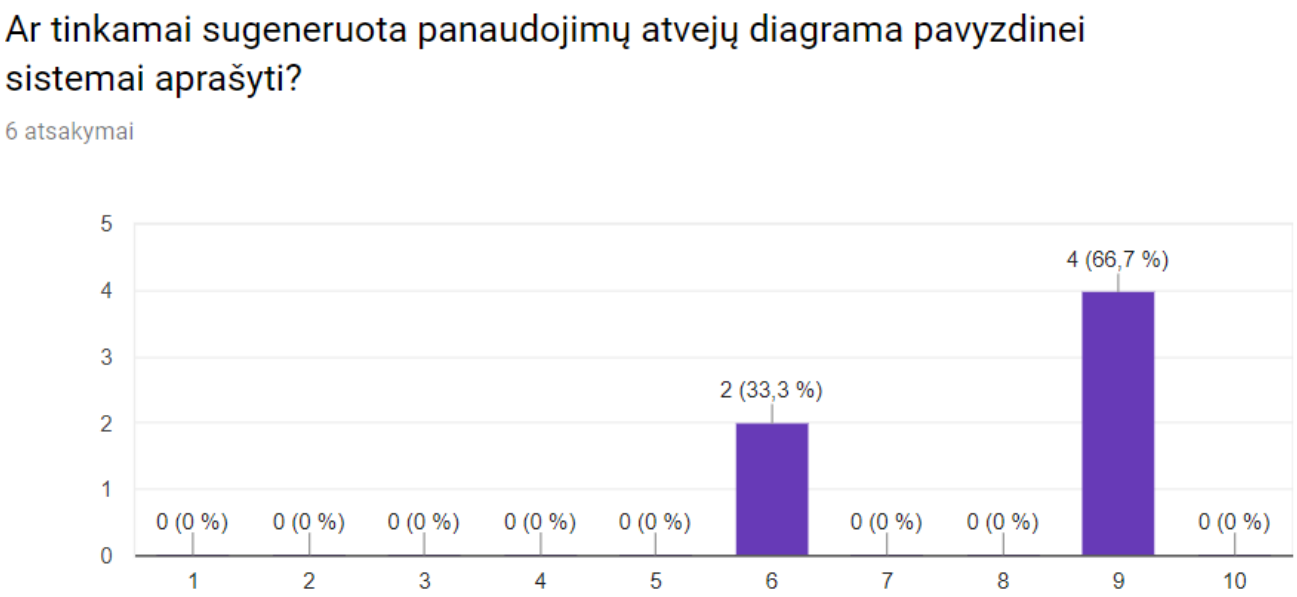



**5.18 pav.** Elementų pavadinimų esančių veiklos diagramose kokybės vertinimo rezultatai „moodle.if.ktu.lt“ sistemos atveju

### 5.3.2. ResearchGate.com panaudojimo atvejų modelio vertinimas

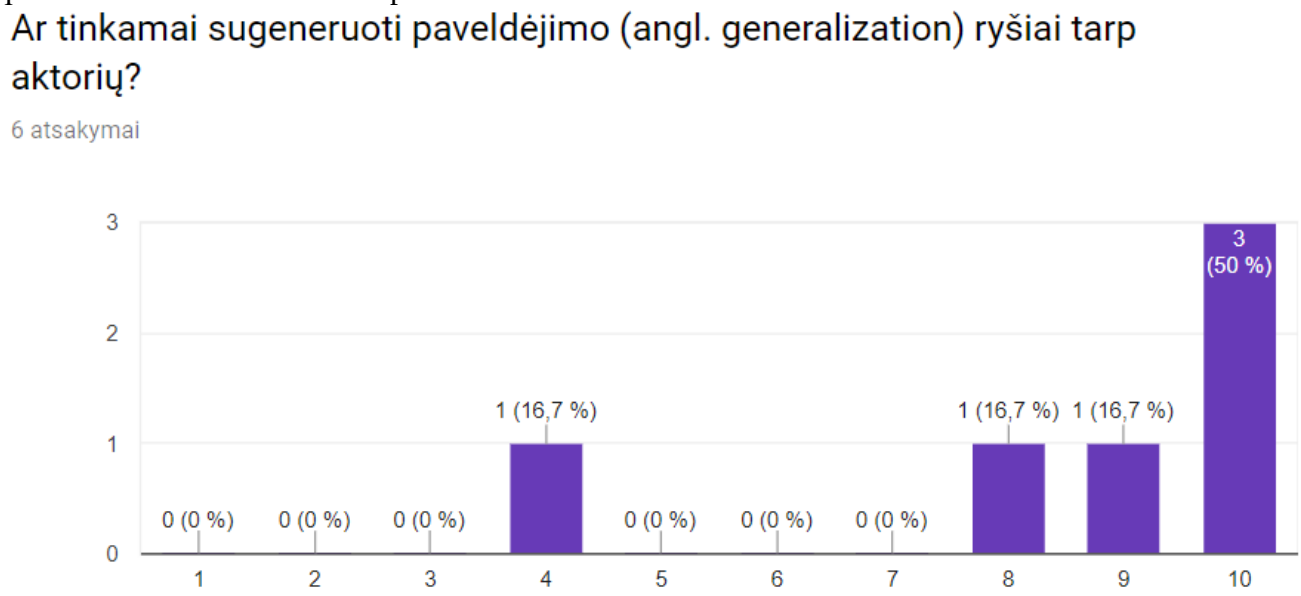
Antroji antrojo eksperimento dalis buvo atlikta analizuojant naujai sukurtu „WEB2UML“ įrankiu sugeneruotas panaudojimo atvejų modelis. Eksperimentams buvo pateiktas sistemos panaudojimo atvejų modelis ir jie atsakė į tuos pačius klausimus, kaip pirmojoje antrojo eksperimento dalyje.

Panaudojimo atvejų diagramos kokybiškumą du ekspertai įvertino vidutiniškai ir 4 - gerai. Detalus eksperto nuomonių pasiskirstymas pateikiamas 5.19 pav.



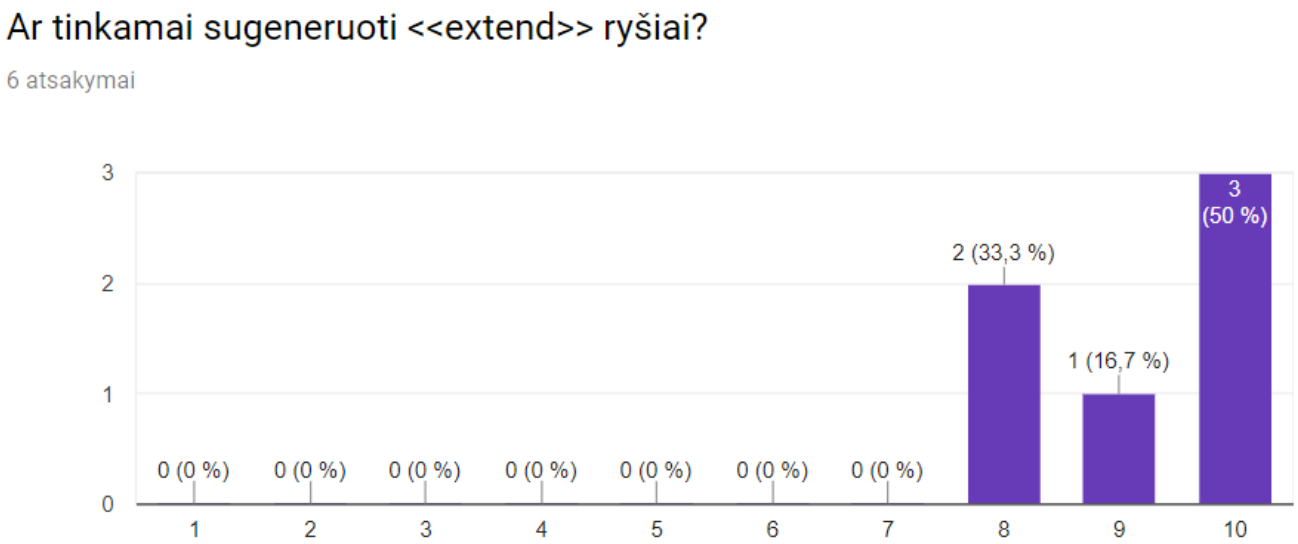

**5.19 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramos kokybę „ResearchGate“ sistemos atveju

Antrojo klausimo metu ekspertai turėjo atsakyti ar teisingai buvo sugeneruoti paveldėjimo ryšiai tarp aktorių sugeneruotoje panaudojimo atvejų diagramoje. Didžioji dalis ekspertų apibendrinimo ryšius vertino gerai, tik vienas ekspertas manė, kad paveldėjimo ryšiai sugeneruoti prasčiau nei vidutiniškai 5.20 pav.



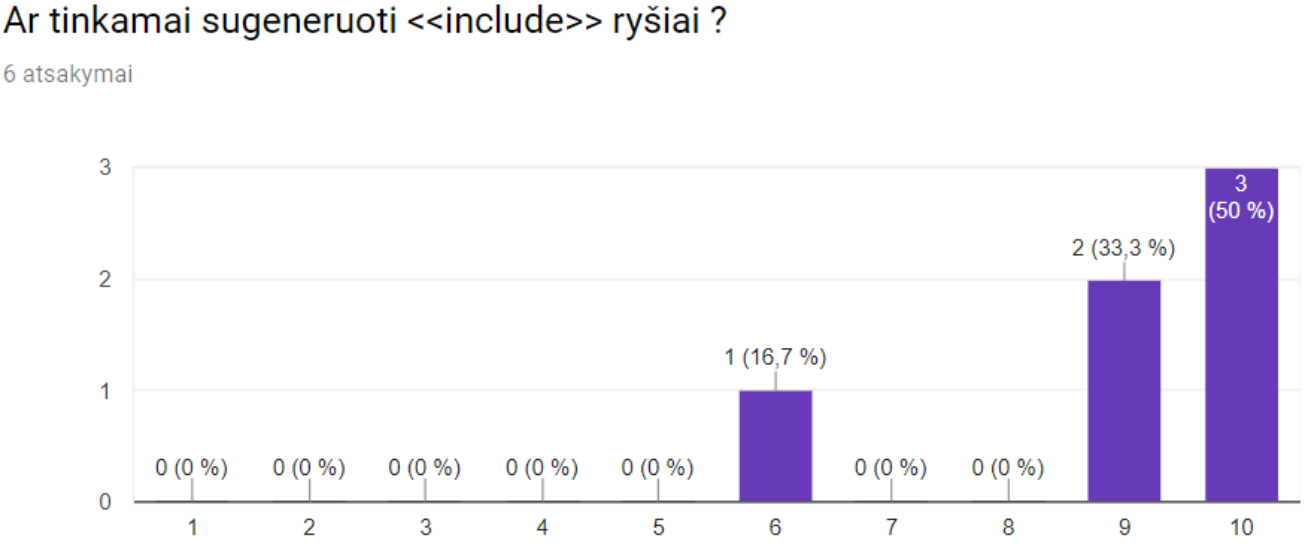

**5.20 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų paveldėjimo ryšių kokybę „ResearchGate“ sistemos atveju

Trečiojo klausimo metu apklausos dalyviai įvertino išplėtimo ryšių teisingumą sukurtoje panaudojimo atvejų diagramoje. Ekspertų nuomone išplėtimo ryšiai, šiame modelyje buvo sugeneruoti teisingai. Ekspertų nuomonių pasiskirstymas pateikimas 5.21 pav.



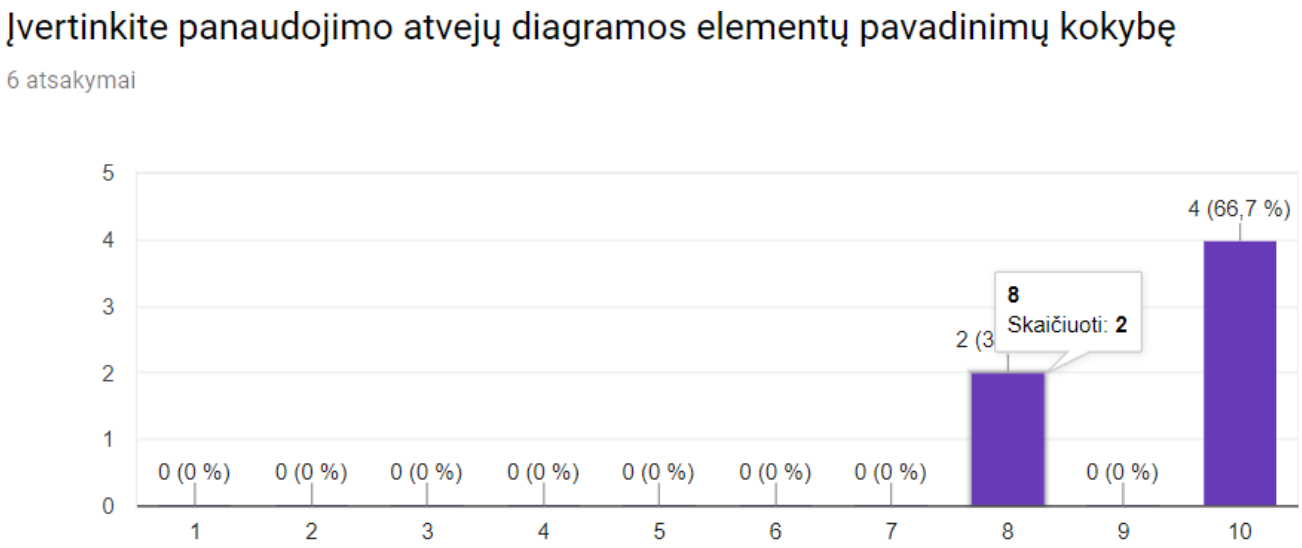

**5.21 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų išplėtimo ryšių kokybę „ResearchGate“ sistemos atveju

Ekspertai taip pat gerai vertino apėmimo ryšių sugeneravimo funkcionalumą naujai sukurtame įrankyje „WEB2UML“. Išsiskyrė tik vieno eksperto nuomonė, kuris apėmimo ryšių sukūrimą įvertino šiek tiek geriau nei vidutiniškai. Visus atsakymus galima peržiūrėti 5.22 pav.




**5.22 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų apėmimo ryšių kokybę „ResearchGate“ sistemos atveju

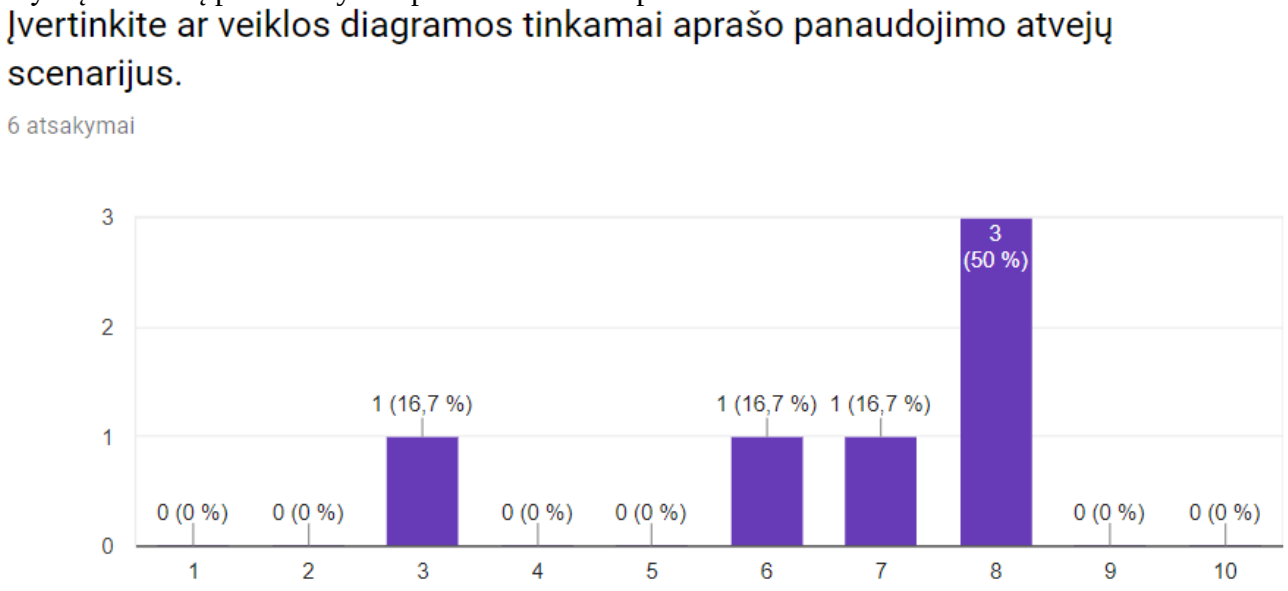
Ekspertų nuomone panaudojimo atvejų kokybė taip pat yra sugeneruota gerai. Apklausos dalyvių panaudojimu atvejų diagramos elementų kokybės vertinimai pateikti 5.23 pav.




**5.23 pav.** Apklausos rezultatai vertinant panaudojimo atvejų diagramoje sugeneruotų elementų pavadinimų kokybę „ResearchGate“ sistemos atveju

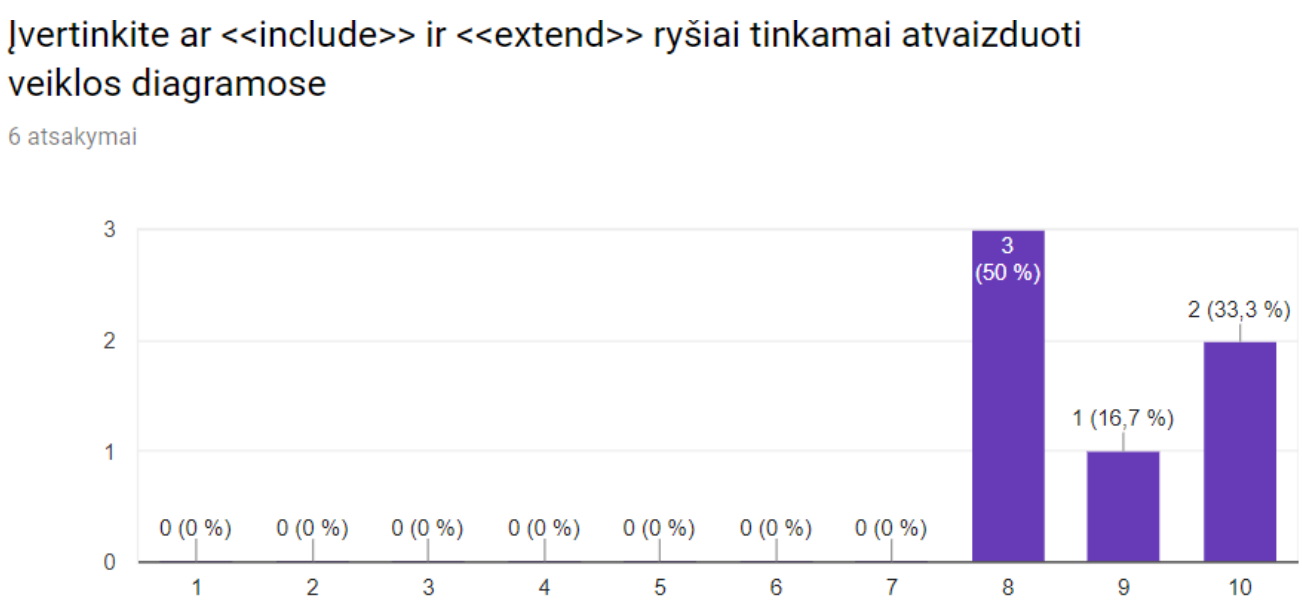
Kaip ir pirmojoje antrojo eksperimento dalyje, kiekvienas panaudojimo atvejis buvo detalizuotas veiklos diagrama. Veiksmų „ResearchGate“ registravimo metu pastebėta, kad puslapis HTML kodą generuoja dinamiškai remiantis JavaScript funkcijomis. Dėl šios priežasties dalis veiksmų veiklos diagramose nebuvo užregistruota, todėl tai atsiliepė veiklos diagramų kokybei. Nepaisant to Panaudojimo atvejų diagrama buvo sugeneruota kokybiškai ir kokybiškai nustatyti išvedamieji ryšiai tokie, kaip apibendrinimo apėmimo ir išplėtimo.

Nepaisant registravimo komponentų sukūrimo apklausos dalyviai veiklos diagramų galimybę aprašyti panaudojimo atvejų scenarijus įvertino geriau vidutiniškai. Tiesa vienas ekspertas manė, kad veiklos diagramos neteisingai aprašo panaudojimo atvejo scenarijų. Detalus apklausos dalyvių rezultatų pasiskirstymas pateikiamas 5.24 pav.



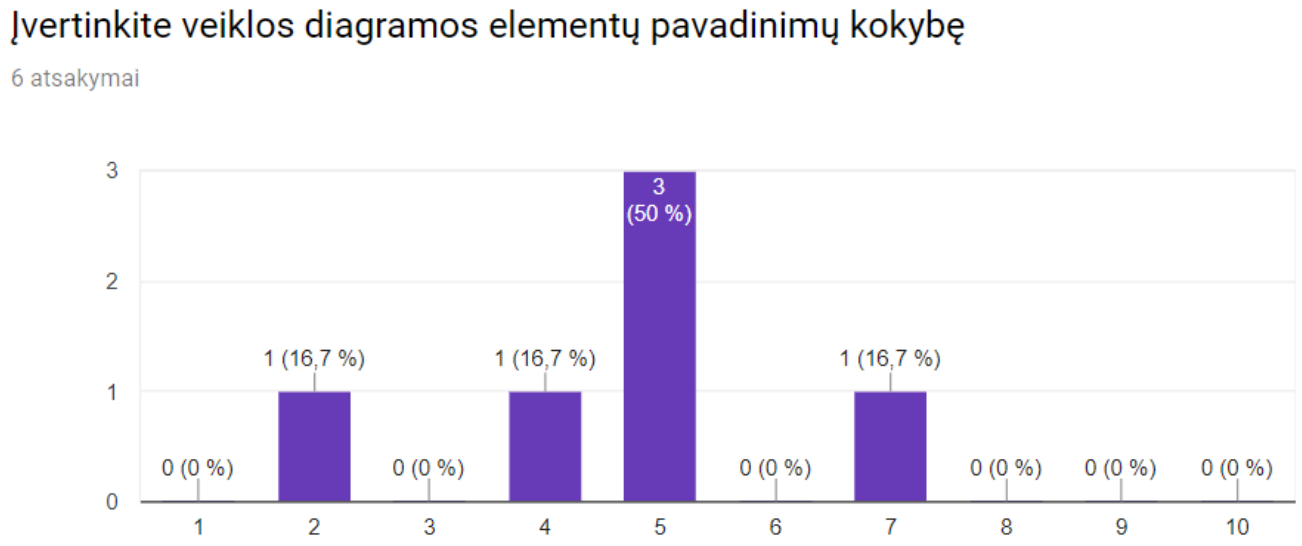

**5.24 pav.** Panaudojimo atvejus aprašančių veiklos diagramų apklausos dalyvių vertinimas „ResearchGate“ sistemos atveju

Kaip ir pirmojo eksperimento metu apklausos dalyviai įvertino, kad apėmimo ir išplėtimo ryšiai veiklos diagramose atvaizduoti teisingai, sukuriant reikiamus elementus. Tačiau šiuo atveju didžioji dalis ekspertų šį kriterijų vertino tik gerai. Visus apklausos dalyvių atsakymus galima peržiūrėti 5.25 pav.




**5.25 pav.** Apėmimo ir išplėtimo ryšių atvaizdavimo teisingumo vertinimo rezultatai „ResearchGate“ sistemos atveju

Kaip jau minėta „ResearchGate“ svetainė savo puslapių kūrimui naudoja sudėtingas *JavaScript* paremtas procedūras todėl nepavyko užregistruoti veiklos proceso taip, kaip jis buvo atliktas svetainėje. Dėl šios priežasties ekspertai įvertino sugeneruotus veiklos diagramos elementų pavadinimus prasčiau nei vidutiniškai 5.26 pav. Tai tik parodo, kad net ir iš dalinės registracijos galima gauti šiek tiek naudos aprašant visą veiklos procesą ir ekspertų nuomone toks veiksmų veiklos diagramoms išgavimo būdas yra bent jau vidutiniškai geras.




**5.26 pav.** Elementų pavadinimų esančių veiklos diagramose kokybės vertinimo rezultatai „ResearchGate“ sistemos atveju

Apibendrinant antrojo eksperimento antrąją dalį, matome, kad naujai sukurtas įrankis ekspertų nuomone gali būti pritaikomas kokybiškiems panaudojimo atvejo modeliams generuoti, tačiau veiklos diagramų elementų kokybė ir galimybė aprašyti panaudojimo atvejų scenarijus turi būti tobulintina. Šiam tikslui pasiekti reiktų sukurti, patobulintą veiksmų registravimo komponento versiją.

## 5.4. Eksperimento išvados

1. Pirmojo eksperimento metu sėkmingai sugeneruoti du skirtingi panaudojimo atvejų modeliai, kurie vėliau buvo įvertinti siekiant patikrinti modelių atitikimą tarp naujai sugeneruoto ir originaliai sumodeliuoto. Nors pastebėta, keletas trūkumų registravimo metu, panaudojimo atvejų modeliai buvo sugeneruoti sėkmingai ir atitiko jiems keliamus reikalavimus, taip patvirtinant iškeltą hipotezę, kad panaudojimo atvejų modelis gali būti išgautas iš internetinių informacinių sistemų.

2. Antrojo eksperimento metu įgyvendinus ekspertų apklausą, pastebėta, kad nepaisant tobulintinų panaudojimo atvejų scenarijų aprašytu veiklos diagramomis ekspertai gerai vertina sugeneruotų panaudojimo atvejų kokybę.

# 6. SPRENDIMO TAIKYMO REKOMENDACIJOS

## Taikymo apribojimai

Atlikus eksperimentą galime pastebėti, kad sistemos veiksmų registravimo komponentas nėra pritaikytas visoms sistemoms ir geriausiai veikia daug formų turinčiose sistemose. Registravimo klaidų gali pasitaikyti ir registruojant naujus puslapius, kurie savo turiniui atvaizduoti naudoja daug *JavaScript* kalba generuojamo turinio. Taip pat registravimo įrankis nėra pritaikytas užregistruoti veiksmus iš sistemų, kurios naudoja <<*iframe*>> HTML žymes todėl rekomenduojama trumpai peržvelgti sistemos struktūrą, prieš pradedant registraciją, siekiant išvengti galimų klaidų.

## Taikymo rekomendacijos

Naudojantis įrankiu rekomenduojama „Chrome“ įskiepį įsidiegti naujoje naršyklėje ir vienu metu registruoti tik vienos sistemos darbą. Rekomenduojama būti atsidarius tik vieną naršyklės skirtuką, nes, užsimiršus ir pradėjus vykdyti veiksmus kituose skirtukuose, jie taip pat bus užregistruojami. Jeigu nėra galimybės įdiegti naujos „Chrome“ naršyklės, rekomenduojama sukurti naują naudotoją „Chrome“ naršyklėje ir tada vykdyti registraciją. Siekiant apsaugoti konfidencialius duomenis rekomenduojami šį įskiepį įjungti tik registravimo metu, o visu kitu laiku laikyti išjungtą.

Veiksmų registravimo metu rekomenduojama sistemos naudotojams patiems leisti registruoti veiksmų vykdymą. Taip bus registruojamas pagrindinis naudotojų naudojamas scenarijus. Taip pat rekomenduojama užregistruoti kuo daugiau tos pačios rolės sistemos naudotojų naudojimosi sistema scenarijų, nuo alternatyviųjų scenarijų kiekio priklauso sistemos panaudojimo atvejų tikslumas.

Siekiant, užtikrinti, kad panaudojimo atvejų modelis būtų sugeneruotas teisingai, reikėtų sukurti dalykinės srities žodyną, kuriame būtų aprašyti sistemoje egzistuojančios rolės ir veiksmai. Panaudojimo atvejų generavimo procesas priklauso nuo šio semantinio žodyno, todėl kaip pavyzdžiui vienam aktoriui įvedus savo veiksmą pavadinimu prisijungti ir kitam įvedus tą patį veiksmą pavadinimu autentifikuotis. Tokia pati situacija gali susidaryti nustatant ryšius tarp aktorių, kai vieno aktoriaus pavadinimas nesutampą su kito aktoriaus pavadinimu, tačiau jie vykdo tas pačias veiklas sistemoje.

Taip pat siekiant teisingai nustatyti išplėtimo ryšius būtina užregistruoti norimą procesą daugiau nei vieną kartą. Kuo daugiau scenarijų bus užregistruota tuo pačiu vardu tuo tikslesnė panaudojimo atvejų ir tuo pačiu konkretų panaudojimo atvejį aprašanti veiklos diagrama bus tikslesnė.

Apėmimo ryšiai nustatomi tik tuose panaudojimo atvejuose, kurie buvo užregistruoti vieną kartą. Toks apribojamas yra taikomas, kad nepainioti apėmimo ir išplėtimo ryšių. Apėmimo ryšiai nustato bendras veiksmų sekos dalis, kurios yra ilgesnės nei N kintamasis įvestas sistemoje ir iškelia, jas į atskirą panaudojimo atvejį, kuris su panaudojimo atveju iš kurio buvo iškelta veiksmų seka sujungiamas apėmimo ryšiu. Taip panaudojimo atvejų diagramoje sumažėja dažnai pasikartojančių veiksmų ir ji tampa lengviau suprantama.

## 7. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

1. Išnagrinėjus internetinių informacinių sistemų (IIS) specifiką, nustatyta, kad metodai ir jų realizacijos, geriausiai leidžiantys išnagrinėti IIS, veikia naršyklėje, kuri suteikia galimybę nagrinėti grafinės naudotojo sąsajos elementus, neturint tiesioginio priėjimo prie programos kodo. Šis, nepriklausomas nuo programinio kodo, sprendimas suteikia galimybę analizuoti kuo įvairesnes IIS.
2. Išnagrinėjus atgalinės inžinerijos metodus, nustatyta, kad tinkamiausias būdas UML diagramų generavimui iš internetinių informacinių sistemų – grafinės naudotojų sąsajos analizė, taikant statinę kodo analizę ir naudotojo veiksmų registravimą.
3. Išnagrinėjus UML diagramas ir jų taikymą sistemų specifikavimui, pastebėta, kad informaciją, išgautą registruojant naudojimąsi sistema ir analizuojant sistemos grafinę vartotojo sąsają, galima atvaizduoti UML panaudojimo atvejų ir veiklos diagramomis, kurios sudaro panaudojimo atvejų modelį.
4. Išanalizavus rinkoje egzistuojančius įrankius ir metodus, nebuvo surastas sprendimas, tinkantis UML modelių generavimui iš internetinių informacinių sistemų grafinės naudotojo sąsajos ir veiksmų registravimo. Tačiau buvo surasta panašių pavyzdžių, kurie įrodo, kad, naudojant veiksmų registravimą ir vartotojo grafinės sąsajos analizę, galima realizuoti UML modelių generavimo metodą ir įrankį.
5. Realizuojant pasiūlytą panaudojimo atvejų modelio generavimo metodą, pastebėta, kad, siekiant sukurti lanksčią sistemą, tarpinius rezultatus patogu saugoti JSON formatu, o siekiant pritaikyti sistemą naudojimui CASE įrankiuose, rezultatus reikia atvaizduoti XMI formatu, kuris suteikia galimybę naudoti sugeneruotą modelį skirtinguose UML CASE įrankiuose.
6. Pirmojo eksperimento metu sėkmingai sugeneruoti du skirtingi panaudojimo atvejų modeliai, kurie vėliau buvo įvertinti, siekiant patikrinti modelių atitikimą tarp sugeneruoto ir originaliai sumodeliuoto. Pirmajam sugeneruotam panaudojimo atvejų modeliui, panaudojimo atvejų diagramos elementų vidutinis atitikimas siekė 100%, o naudotojų veiksmų atitikimas veiklos diagramose siekė 75%. Antrajam sugeneruotam panaudojimo atvejų modeliui, panaudojimo atvejų diagramos elementų vidutinis atitikimas siekė 87%, o naudotojų veiksmų atitikimas veiklos diagramose siekė 57%. Nors ir pastebėta keletas trūkumų registravimo metu, panaudojimo atvejų modeliai buvo sugeneruoti sėkmingai ir atitiko jiems keliamus reikalavimus, taip patvirtinant iškeltą hipotezę, kad panaudojimo atvejų modelis gali būti išgautas iš internetinių informacinių sistemų.
7. Antrojo eksperimento metu, atlikus ekspertų apklausą pastebėta, kad nepaisant tobulintinų panaudojimo atvejų scenarijų aprašymų veiklos diagramomis, ekspertai gerai vertina sugeneruotų panaudojimo atvejų modelių kokybę (ekspertų įvertinimų vidurkis – 86%) . Ekspertai taip pat gerai (vidurkis 92%) vertina abejuose pavyzdiniuose modeliuose nustatytų ryšių kokybę bei geriau nei vidutiniškai (vidurkis 68%) vertina sugeneruotų veiklos diagramų gebėjimą teisingai aprašyti panaudojimo atvejų scenarijus.
8. Pastebėta, kad darbe realizuotas įrankis negali pilnai išnaudoti savo galimybių tokiose sistemose, kuriose egzistuoja <<*iframe*>> elementai bei HTML kodo kūrimas sistemos atvaizdavimo metu ypač stipriai priklauso nuo *JavaScript* funkcijų. Siekiant teisingai išnaudoti įrankį, rekomenduojama jį naudoti daug formų turinčiose sistemose, kurios neperkrautos *JavaScript* funkcijomis, kuriančiomis dinaminį turinį.
9. Ateityje, siekiant skatinti UML modelių naudojimą programinės įrangos kūrime būtų naudinga pritaikyti sistemos pateikiamus rezultatus bent vienam atvirojo kodo UML modeliavimo įrankiui.

## 8. LITERATŪRA


[1] G. Antoniol, M. D. Penta ir G. D. Lucca, „An approach for reverse engineering of web-based applications," *Reverse Engineering, 2001. Proceedings,* 2001.

[2] C. D. Francescomarino, A. Marchetto ir P. Tonella, „Reverse Engineering of Business Processes exposed as Web Applications," *Software Maintenance and Reengineering, 2009. CSMR '09,* 2009.

[3] P. Tramontana, „Reverse engineering Web applications," *Software Maintenance, 2005. ICSM'05,* 2005.

[4] K. BECK ir e. al., *Manifesto for agile software development,* 2001.

[5] O. Pastor, S. España, J. I. Panach ir N. Aquino, „Model-Driven Development," įtraukta *Informatik-Spektrum,* 2008.

[6] Richard Soley and the OMG Staff Strategy Group, *Model Driven Architecture,* 2000 .

[7] S. J. M. Kendall, S. Axel ir U. D. Weise, „Model-Driven Architecture," įtraukta *International Conference on Object-Oriented Information Systems,* 2002.

[8] I. I. ISO, „ISO/IEC/IEEE 29148". USA 01 12 2011.

[9] „UML 2.5 Specification," 1 03 2015. [Tinkle]. Available: http://www.omg.org/spec/UML/2.5/PDF.

[10] G. Rossi, Ó. Pastor, D. Schwabe ir L. Olsina, Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag London, 2008.

[11] T. Cipresso ir M. Stamp, „Software reverse engineering," *Handbook of Information and Communication Security.Springer, Berlin, Heidelberg,* pp. 659-696, 2010.

[12] E. Korshunova, M. Petković, M. v. d. Brand ir M. Mousavi, „CPP2XMI: Reverse Engineering of UML Class, Sequence, and Activity Diagrams from C++ Source Code," įtraukta *Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06),* 2006.

[13] N. M. Inc, „MagicDraw," NoMagic, [Tinkle]. Available: https://www.nomagic.com/products/magicdraw.

[14] S. S. P. Ltd, „Enterprise Architect," [Tinkle]. Available: http://sparxsystems.com/products/ea/.

[15] IBM, „IBM Rational Software Architect," [Tinkle]. Available: https://www.ibm.com/developerworks/downloads/r/architect/index.html.

[16] C. Bellettini, A. Trentini ir A. Marchetto, „WebUML: Reverse engineering of web applications," įtraukta *Conference: Proceedings of the 2004 ACM Symposium on Applied Computing (SAC),* Nicosia, Cyprus, March, 2004.

[17] „https://www.visual-paradigm.com/," [Tinkle]. Available: https://www.visual-paradigm.com/.

[18] M. H. Alalfi, J. R. Cordy ir T. R. Dean, „Automated Reverse Engineering of UML Sequence Diagrams," įtraukta *IEEE International Conference on Software Testing Verification and Validation Workshops,* 2009.

[19] Google, „Google Analytics," [Tinkle]. Available: https://analytics.google.com/analytics/web/#/.

[20] Ipswitch, „iMacros," [Tinkle]. Available: https://imacros.net/about/.

[21] „SeleniumHQ Browser Automation," [Tinkle]. Available: https://www.seleniumhq.org/.

[22] M. I. Muhairat, R. E. Al-Qutaish ir A. A. Abdelqader, „UML Diagrams Generator: A New CASE Tool to Construct the Use-Case and Class Diagrams from an Event Table," *Journal of Computer Science 6,* pp. 253-260, 2010.

[23] M. I. Muhairat ir A. A. Qader, „A New Reverse Engineering Approach to Convert Form Fill Format Document into UML Class Diagram," *International Journal of Software Engineering,*

2014.

[24] I. C. Morgado, A. C. R. Paiva ir J. P. Faria, „Dynamic Reverse Engineering of Graphical User Interfaces,“ *International Journal on Advances in Software, vol 5,* 2012.

[25] C. Raibulet, F. A. Fontana ir M. Zanoni, „Model-Driven Reverse Engineering Approaches: A Systematic Literature Review,“ *IEEE Access,* 2017.

[26] T. Ziadi, M. A. A. d. Silva, L. M. Hillah ir M. Ziane, „A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams,“ įtraukta *16th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*, Las Vegas, United States, 2011.

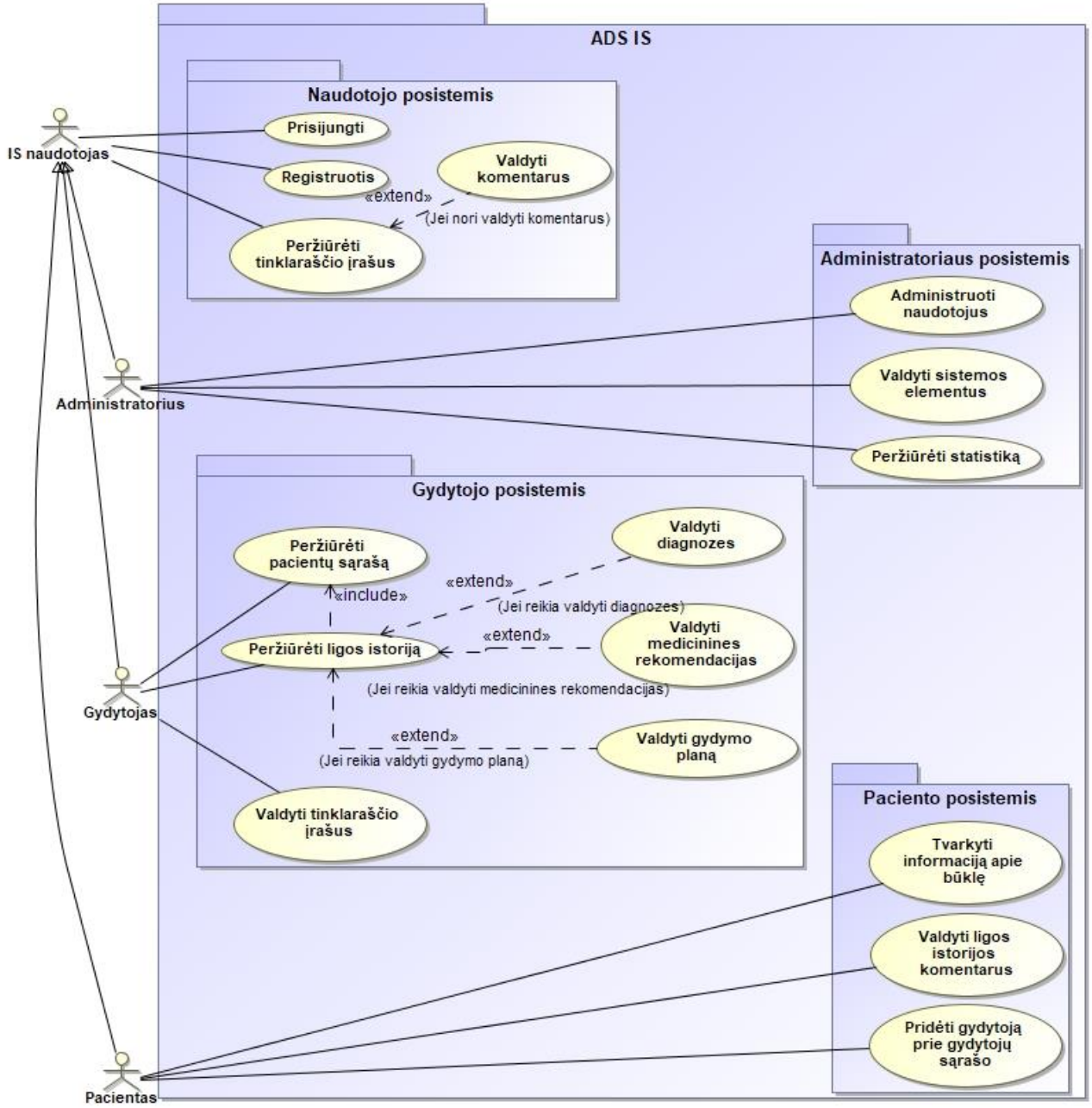
[27] G. A. D. Lucca, A. R. Fasolino ir P. Tramontana, „WARE: a tool for the Reverse Engineering of Web Applications,“ *Journal of Software Maintenance and Evolution: Research and Practice - Special issue: Web site evolution,* pp. 71-101, 2004.

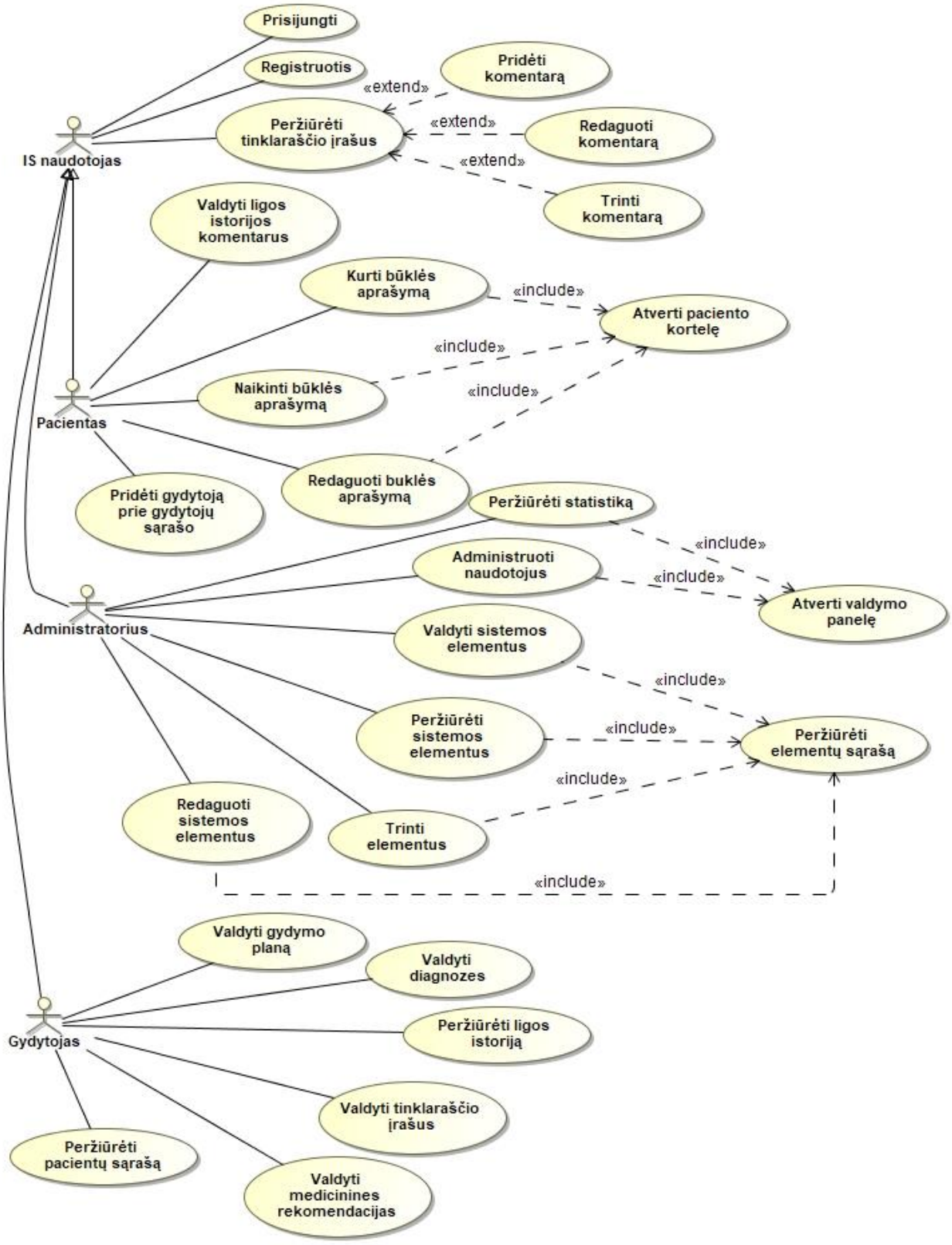
# 9. PRIEDAI

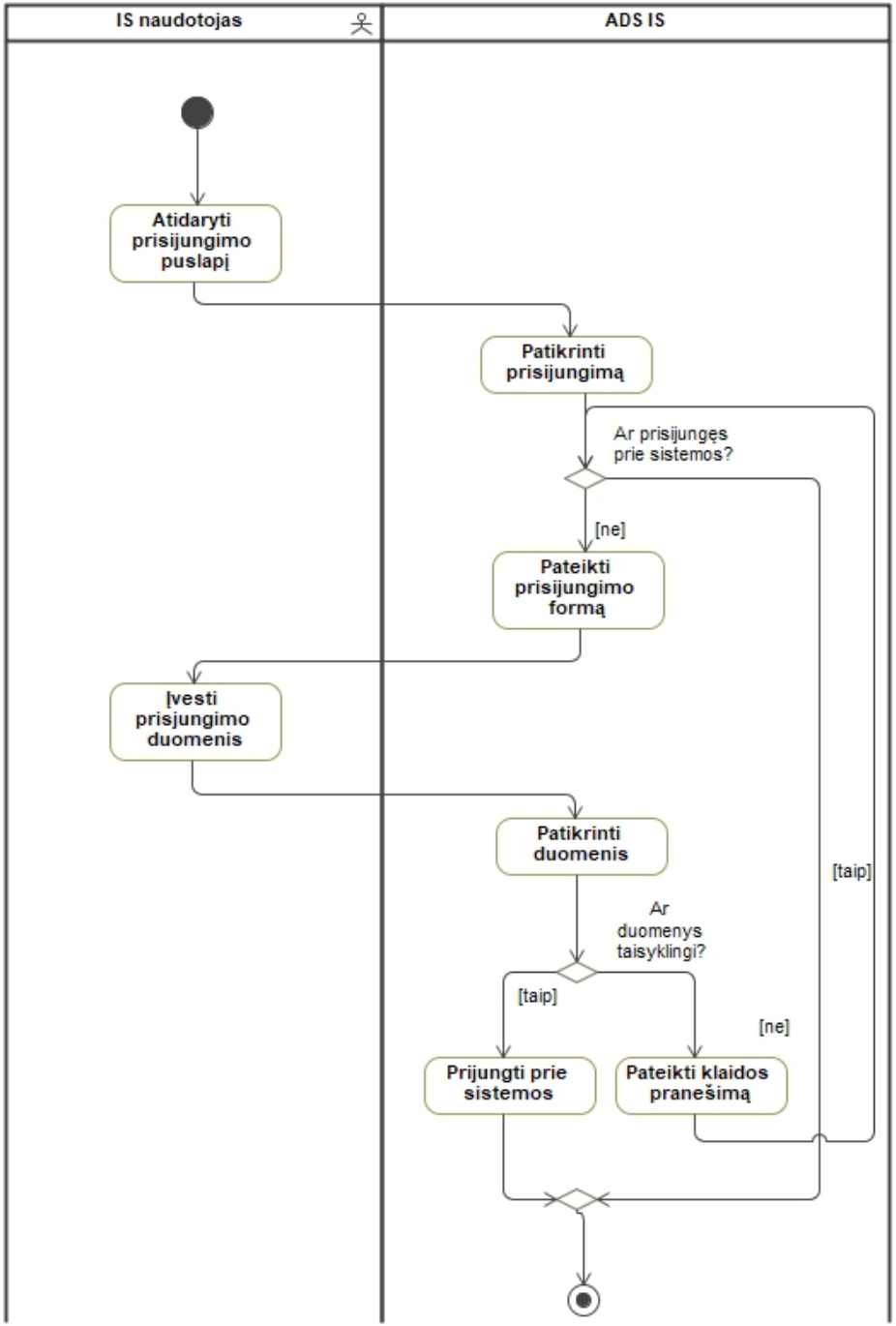
## 9.1. priedas. Sugeneruoti panaudojimo atvejų modeliai

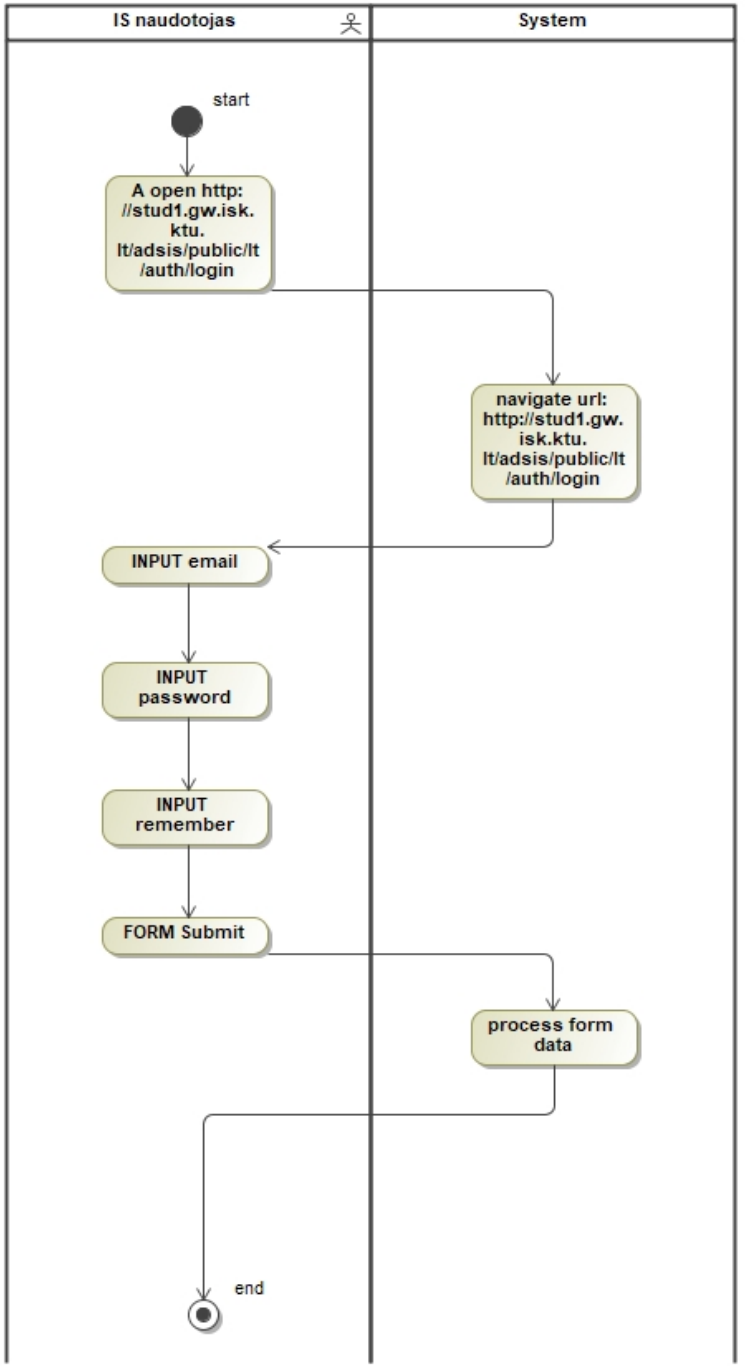
Pirmoji diagrama pateikiama atvaizduoja originaliąją sistemos dokumentaciją, antroji sugeneruotąją. Veiklos diagramų atvejų originalioji veiklos diagrama pateikiama pirma ir ją atitinkanti sugeneruota diagrama pateikiama iškarto po jos, jeigu neparašyta kitaip.

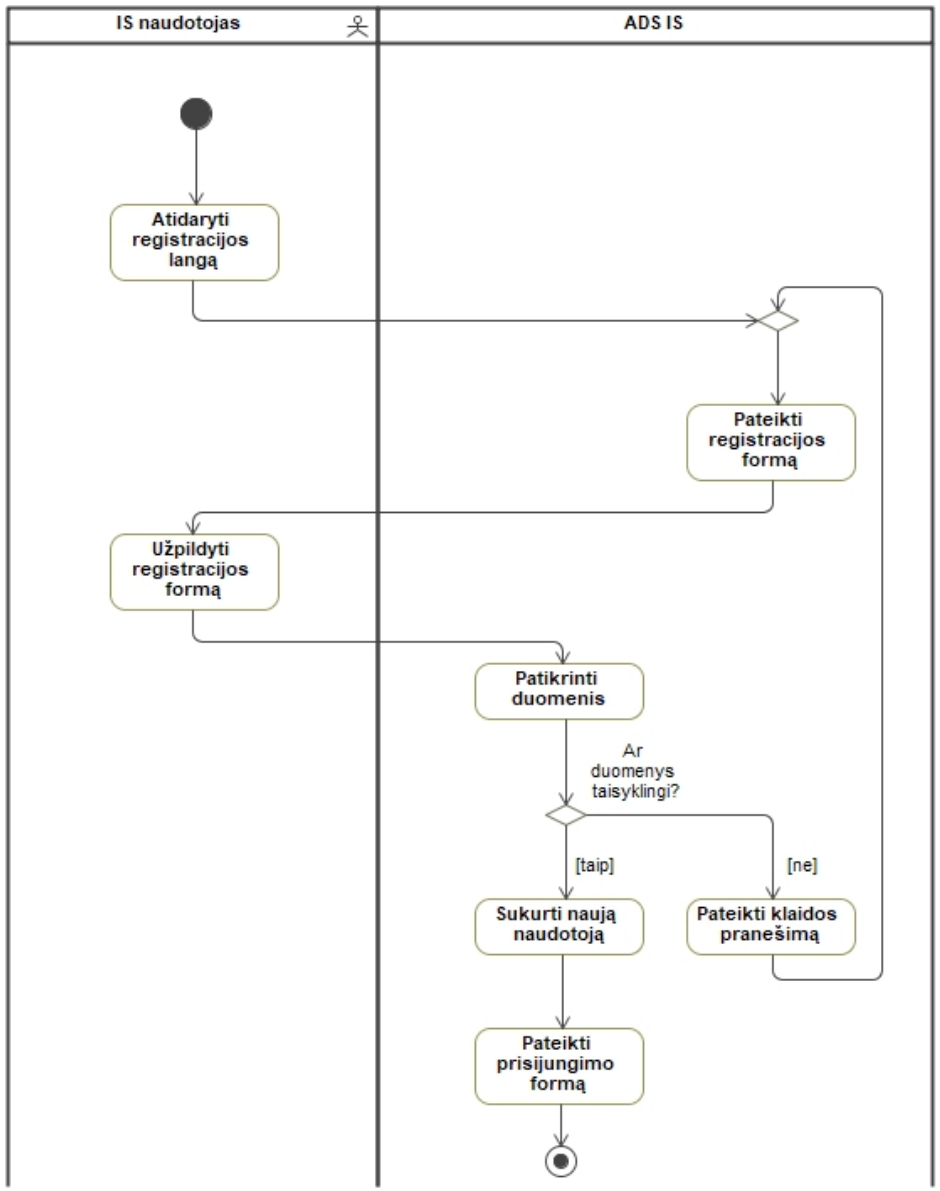
### 9.1.1. priedas „ADSIS“ panaudojimo atvejų modelis ir sugeneruotas panaudojimo atvejų modelis.

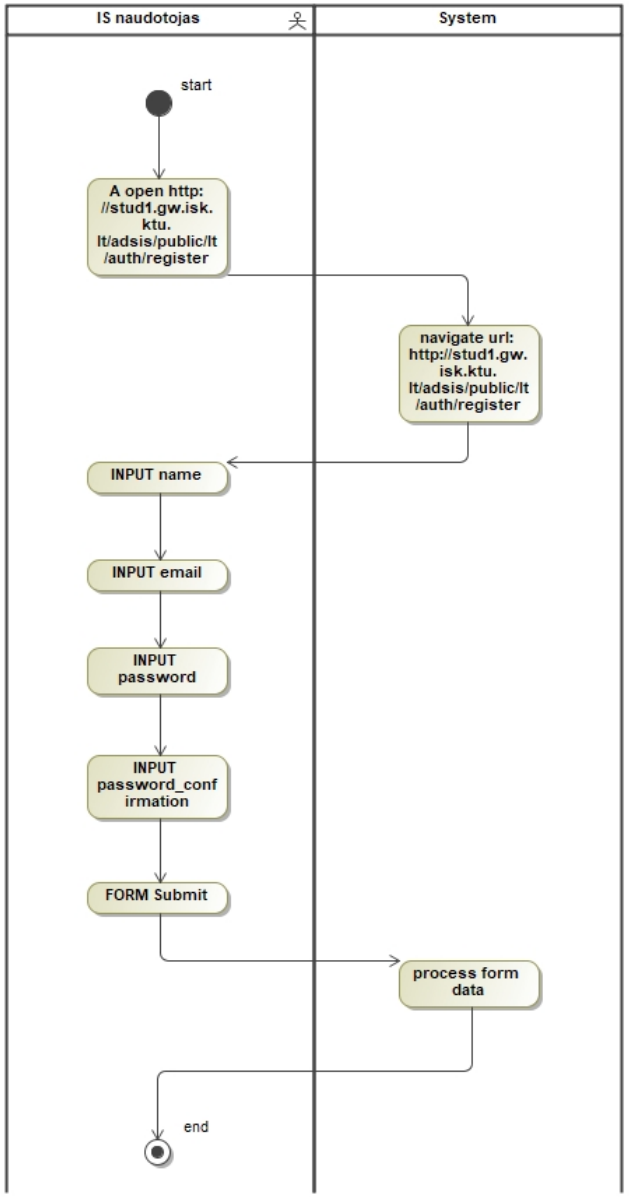


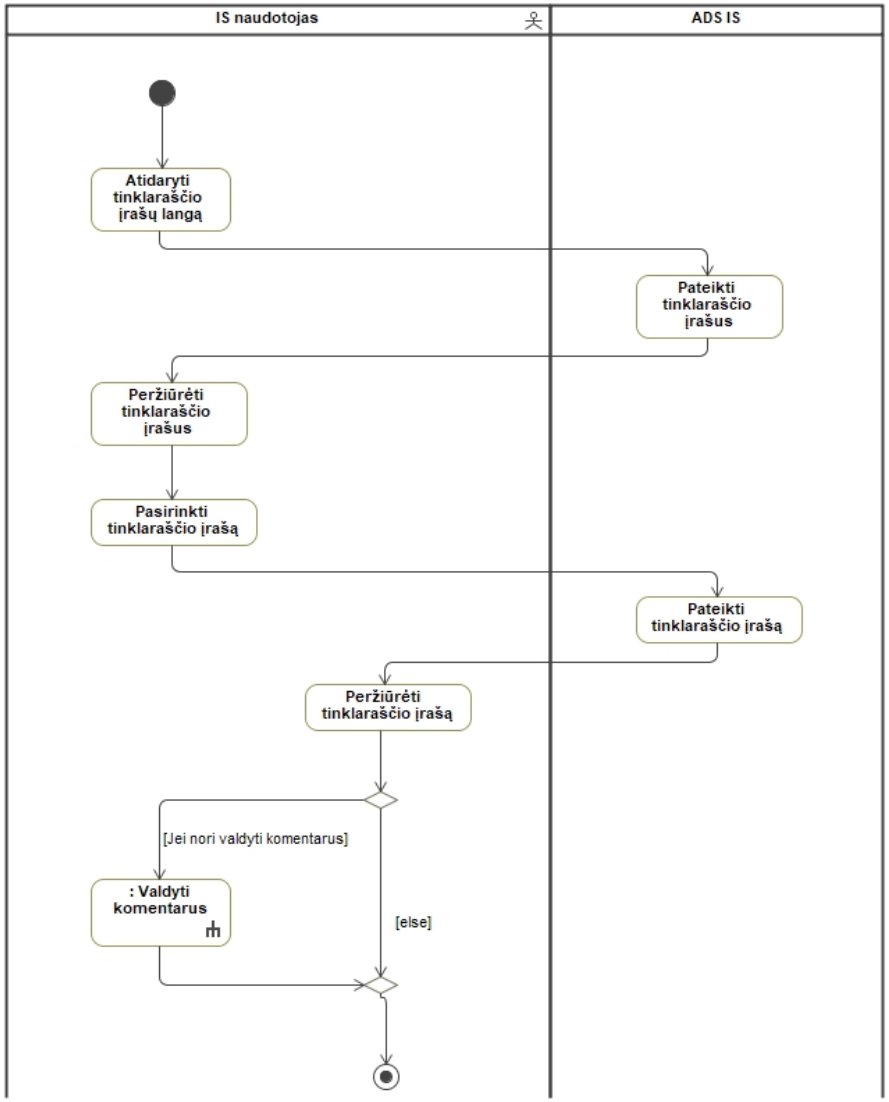

Prisijungti
Registruotis
Pridėti komentarą
«extend»
Peržiūrėti tinklaraščio įrašus
«extend»
Redaguoti komentarą
«extend»
Trinti komentarą
IS naudotojas
Valdyti ligos istorijos komentarus
Kurti būklės aprašymą
«include»
Atverti paciento kortelę
«include»
Naikinti būklės aprašymą
«include»
Pacientas
Redaguoti buklės aprašymą
Pridėti gydytoją prie gydytojų sąrašo
Peržiūrėti statistiką
«include»
Administruoti naudotojus
«include»
Atverti valdymo panelę
Administratorius
Valdyti sistemos elementus
«include»
Peržiūrėti sistemos elementus
«include»
Peržiūrėti elementų sąrašą
«include»
Redaguoti sistemos elementus
Trinti elementus
«include»
Valdyti gydymo planą
Valdyti diagnozes
Peržiūrėti ligos istoriją
Gydytojas
Valdyti tinklaraščio įrašus
Peržiūrėti pacientų sąrašą
Valdyti medicinines rekomendacijas

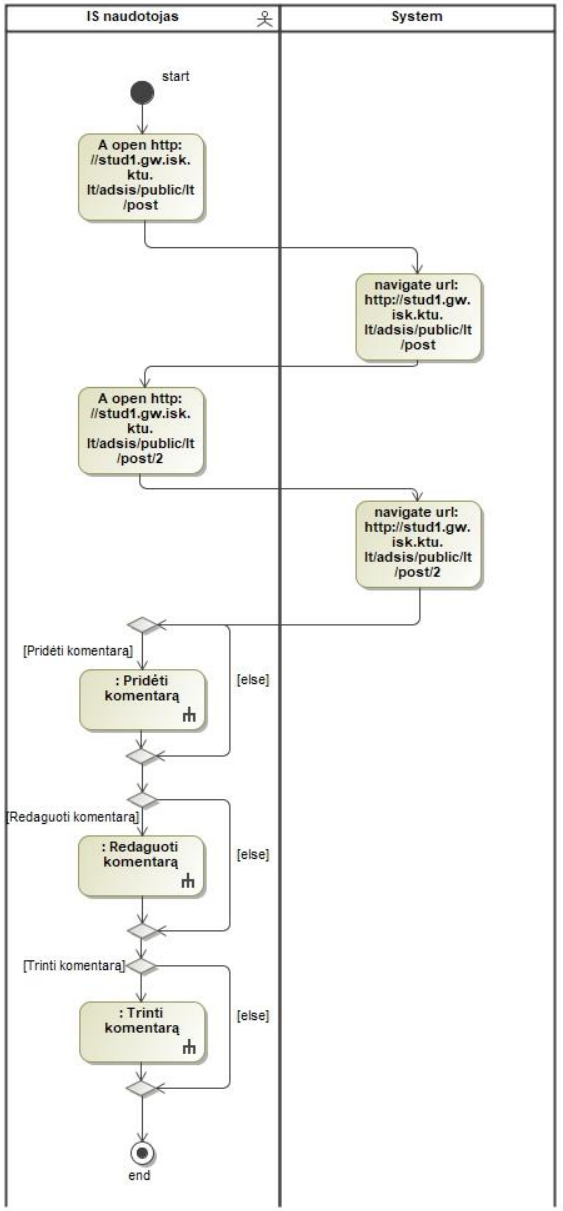

IS naudotojas
ADS IS
Atidaryti prisijungimo puslapį
Patikrinti prisijungimą
Ar prisijungęs prie sistemos?
[ne]
Pateikti prisijungimo formą
Įvesti prisjungimo duomenis
Patikrinti duomenis
Ar duomenys taisyklingi?
[taip]
[ne]
[taip]
Prijungti prie sistemos
Pateikti klaidos pranešimą

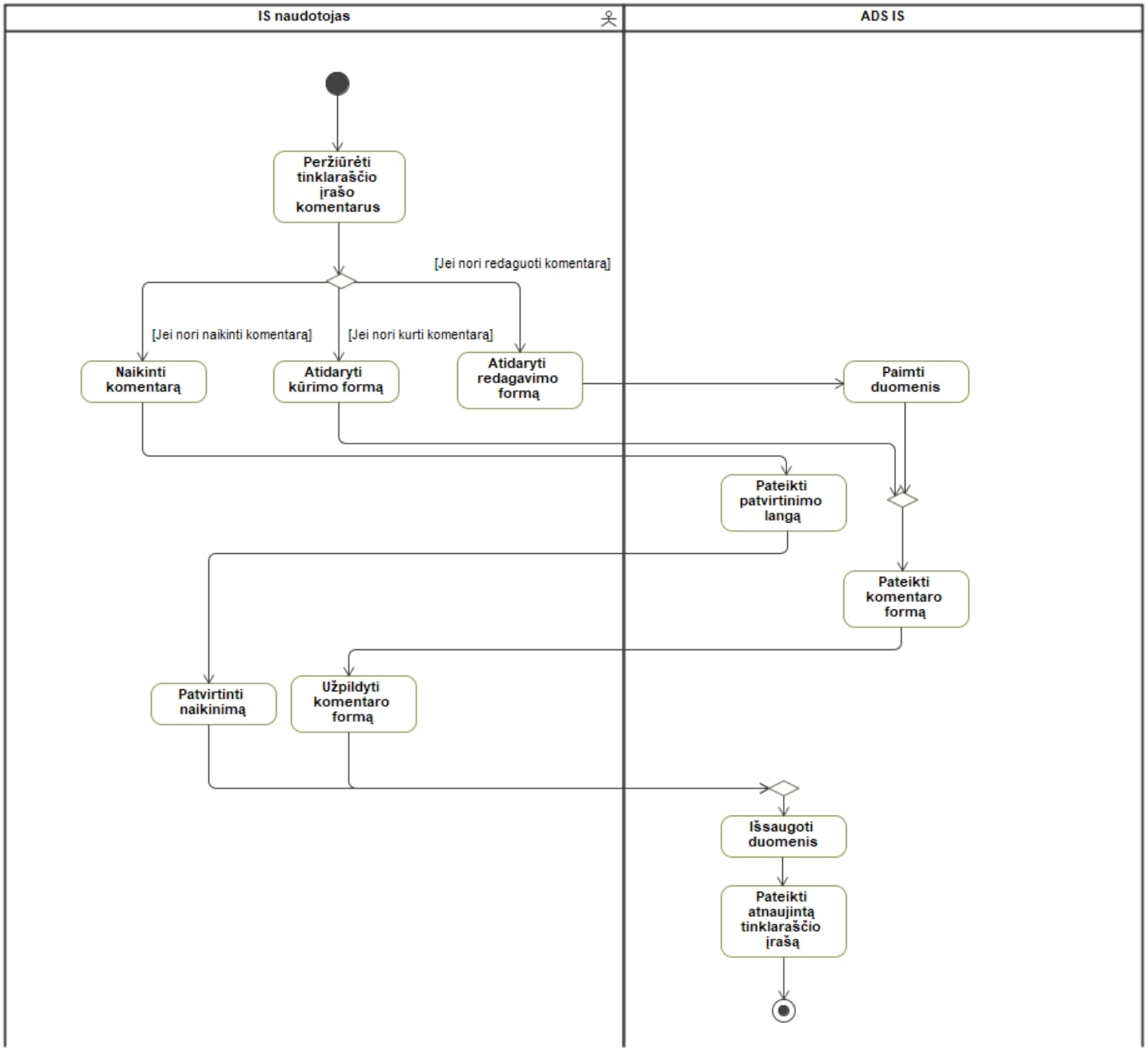

IS naudotojas
System
start
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/auth/login
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/auth/login
INPUT email
INPUT password
INPUT remember
FORM Submit
process form data
end

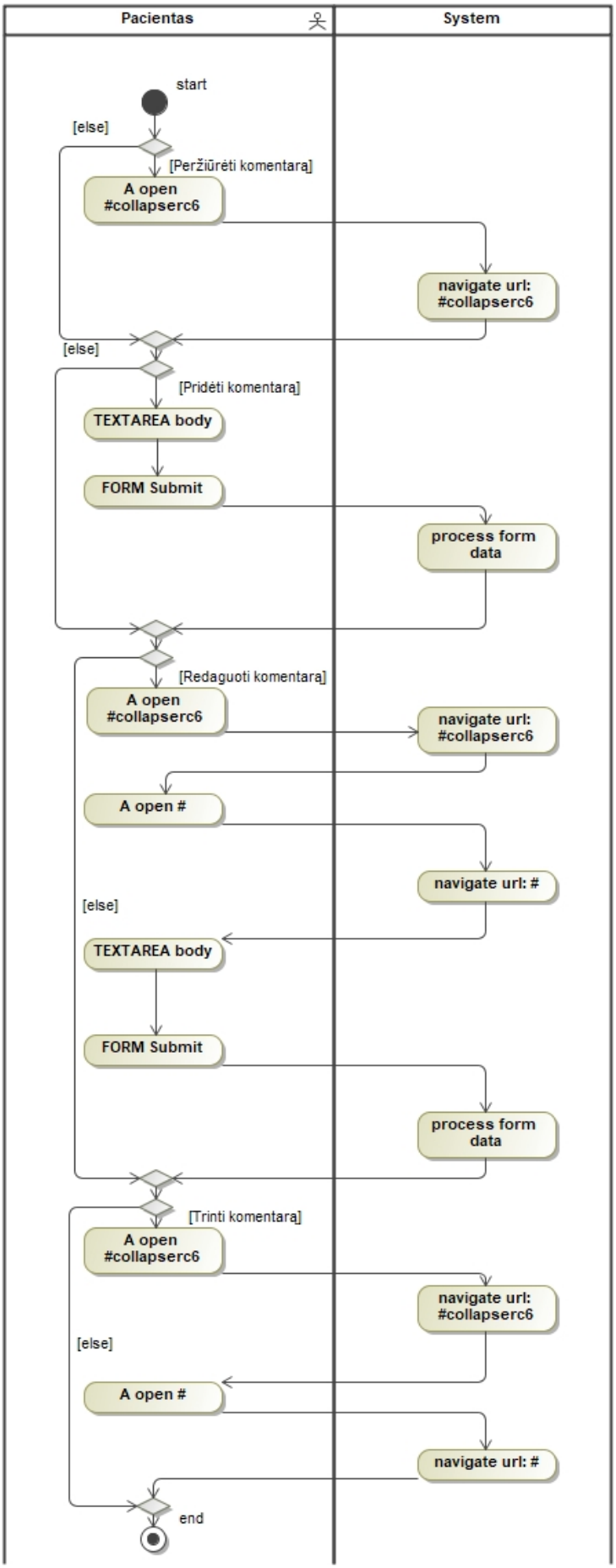

IS naudotojas
ADS IS
Atidaryti registracijos langą
Pateikti registracijos formą
Užpildyti registracijos formą
Patikrinti duomenis
Ar duomenys taisyklingi?
[taip]
[ne]
Sukurti naują naudotoją
Pateikti klaidos pranešimą
Pateikti prisijungimo formą

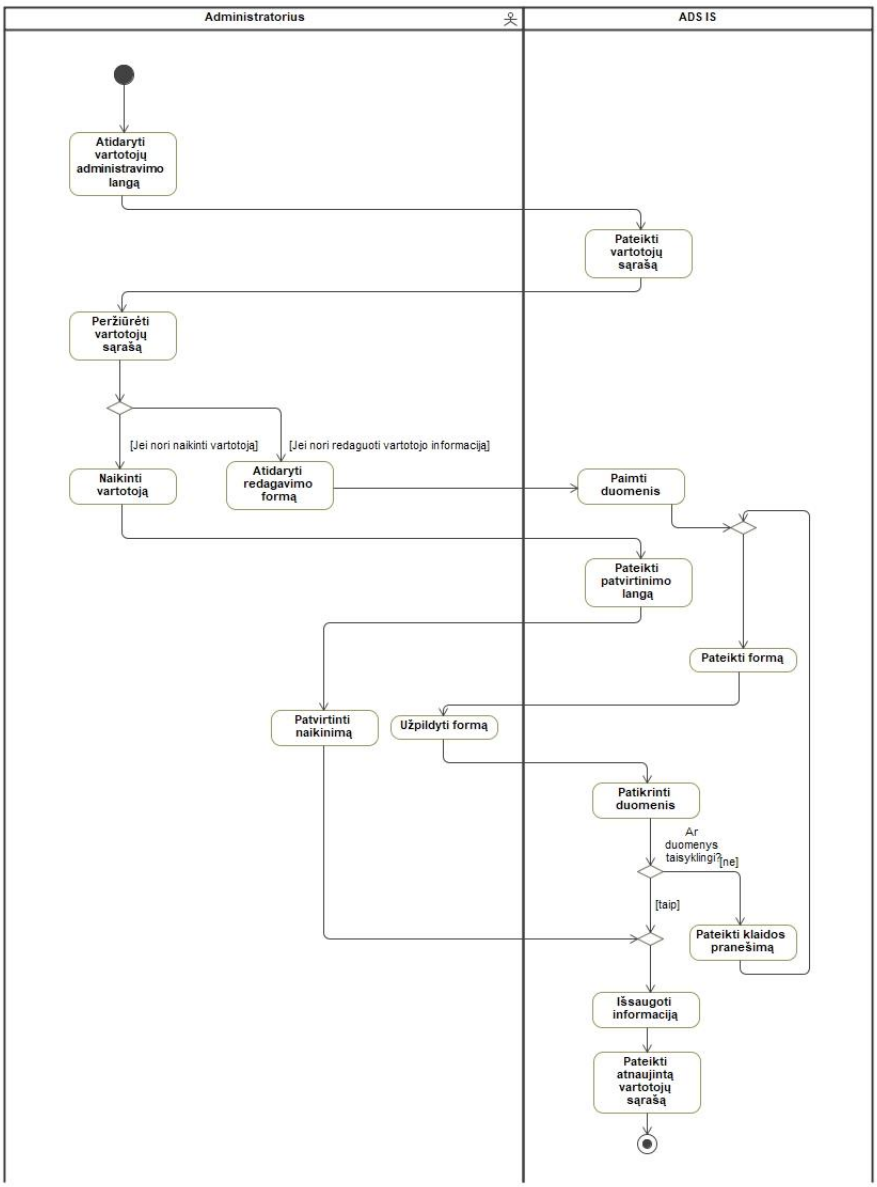

IS naudotojas
System
start
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/auth/register
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/auth/register
INPUT name
INPUT email
INPUT password
INPUT password_confirmation
FORM Submit
process form data
end

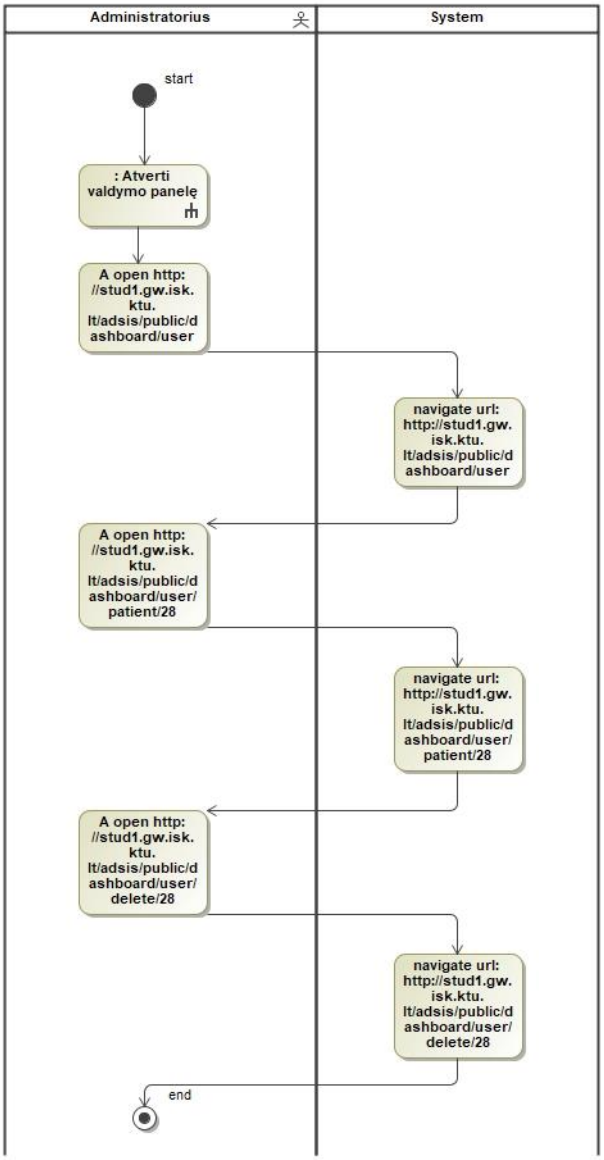

IS naudotojas
ADS IS
Atidaryti tinklaraščio įrašų langą
Pateikti tinklaraščio įrašus
Peržiūrėti tinklaraščio įrašus
Pasirinkti tinklaraščio įrašą
Pateikti tinklaraščio įrašą
Peržiūrėti tinklaraščio įrašą
[Jei nori valdyti komentarus]
: Valdyti komentarus
[else]

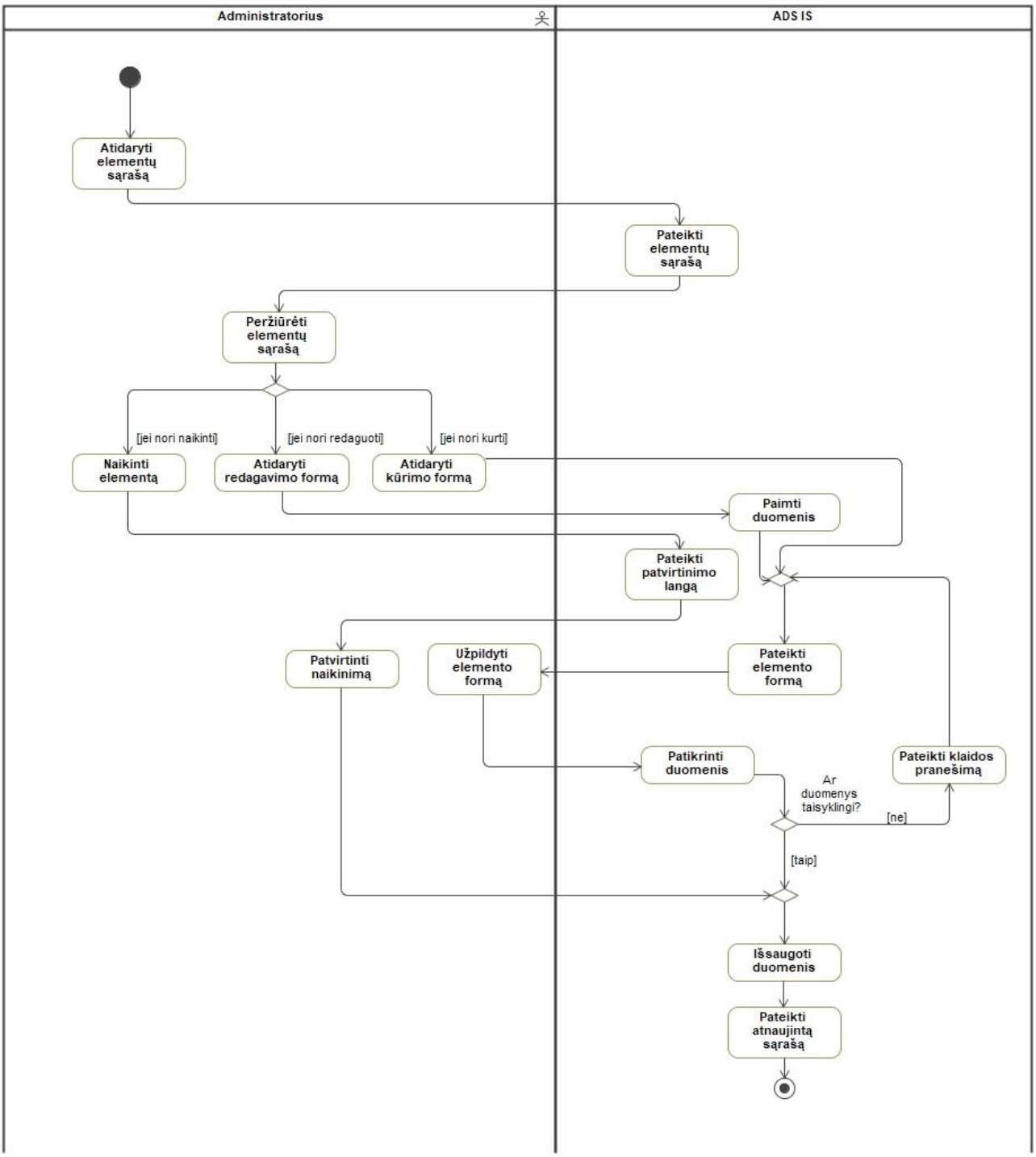

IS naudotojas
System
start
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/post
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/post
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/post/2
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/post/2
[Pridėti komentarą]
: Pridėti komentarą
[else]
[Redaguoti komentarą]
: Redaguoti komentarą
[else]
[Trinti komentarą]
: Trinti komentarą
[else]
end

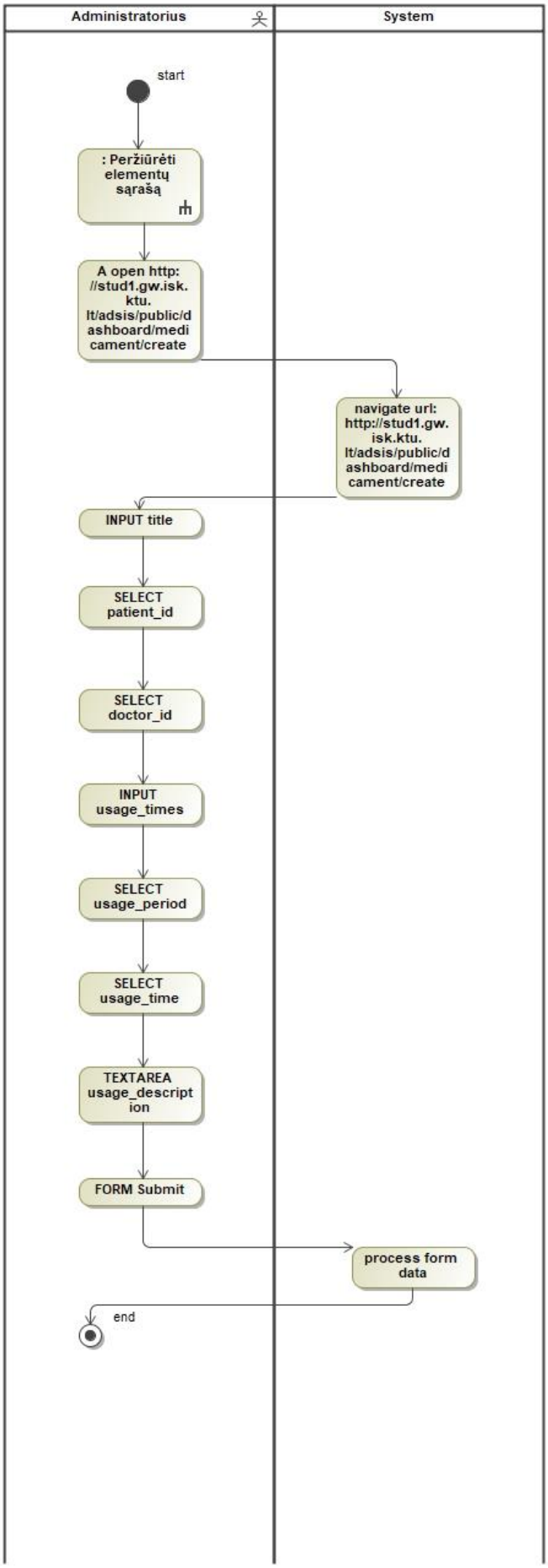

IS naudotojas
ADS IS
Peržiūrėti tinklaraščio įrašo komentarus
[Jei nori redaguoti komentarą]
[Jei nori naikinti komentarą]
[Jei nori kurti komentarą]
Naikinti komentarą
Atidaryti kūrimo formą
Atidaryti redagavimo formą
Paimti duomenis
Pateikti patvirtinimo langą
Pateikti komentaro formą
Patvirtinti naikinimą
Užpildyti komentaro formą
Išsaugoti duomenis
Pateikti atnaujintą tinklaraščio įrašą

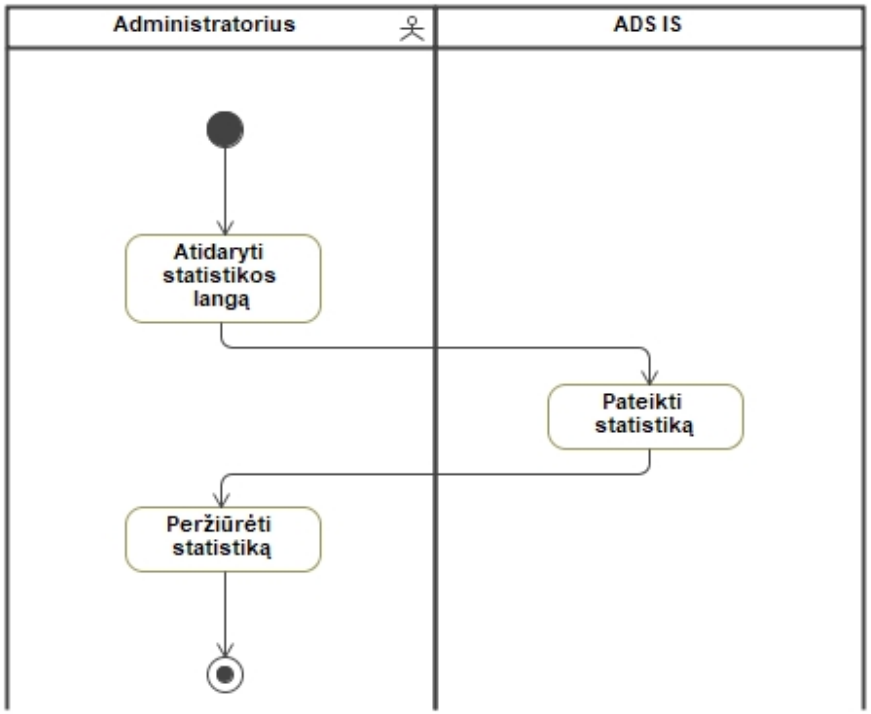

Pacientas
System
start
[else]
[Peržiūrėti komentarą]
A open #collapserc6
navigate url: #collapserc6
[else]
[Pridėti komentarą]
TEXTAREA body
FORM Submit
process form data
[Redaguoti komentarą]
A open #collapserc6
navigate url: #collapserc6
A open #
navigate url: #
[else]
TEXTAREA body
FORM Submit
process form data
[Trinti komentarą]
A open #collapserc6
navigate url: #collapserc6
[else]
A open #
navigate url: #
end

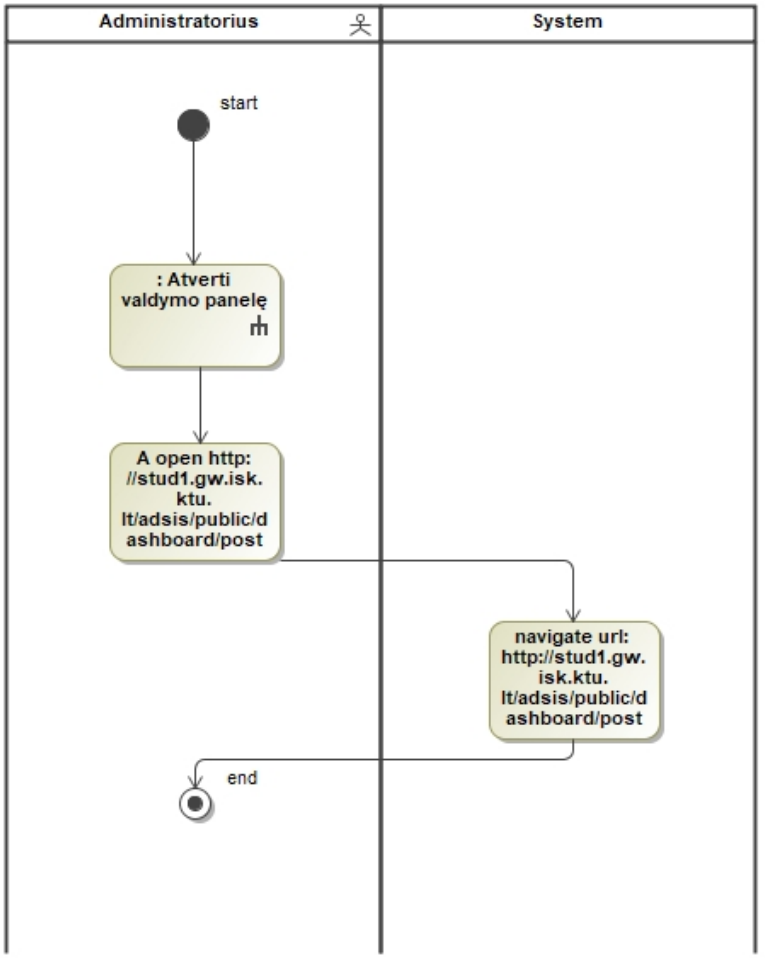

Administratorius
ADS IS
Atidaryti vartotojų administravimo langą
Pateikti vartotojų sąrašą
Peržiūrėti vartotojų sąrašą
[Jei nori naikinti vartotoją]
[Jei nori redaguoti vartotojo informaciją]
Naikinti vartotoją
Atidaryti redagavimo formą
Paimti duomenis
Pateikti patvirtinimo langą
Pateikti formą
Patvirtinti naikinimą
Užpildyti formą
Patikrinti duomenis
Ar duomenys taisyklingi?
[ne]
[taip]
Pateikti klaidos pranešimą
Išsaugoti informaciją
Pateikti atnaujintą vartotojų sąrašą

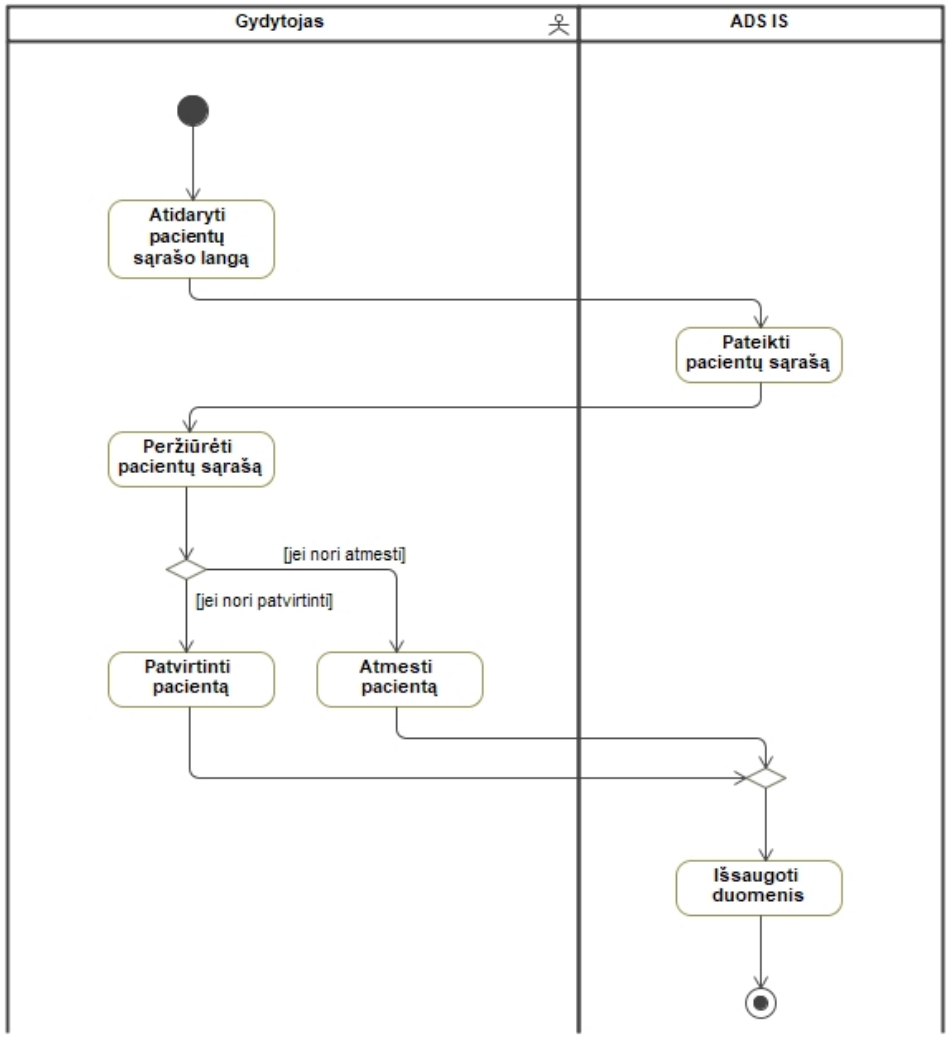

Administratorius
System
start
: Atverti valdymo panelę
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user/patient/28
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user/patient/28
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user/delete/28
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/user/delete/28
end

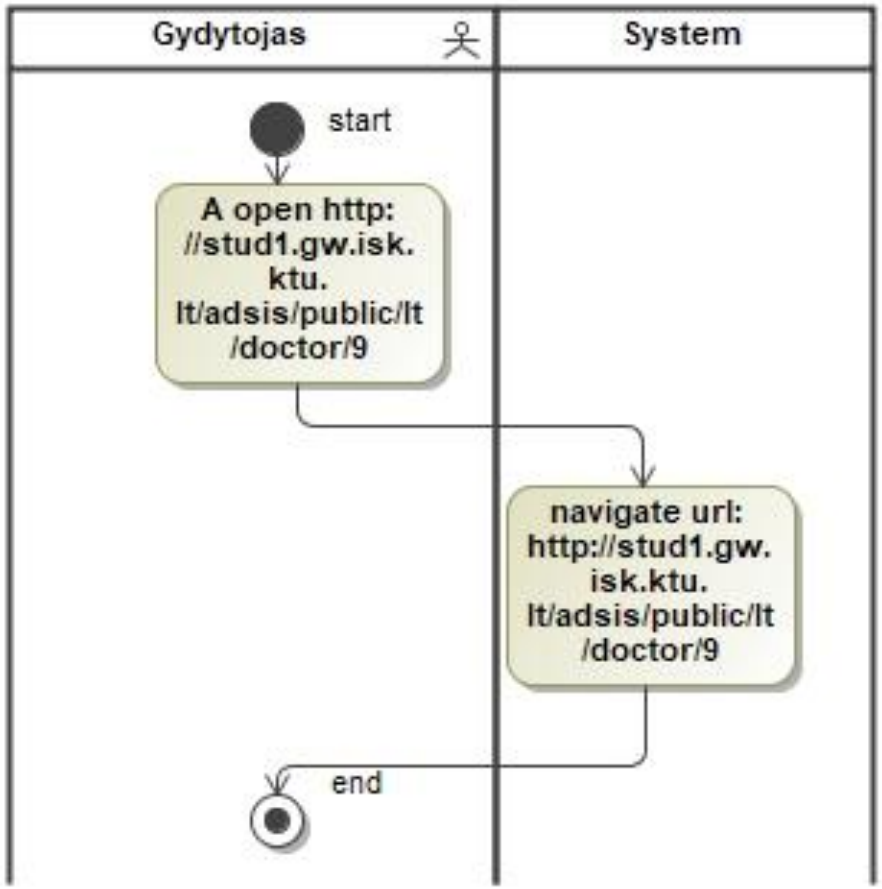

Administratorius
ADS IS
Atidaryti elementų sąrašą
Pateikti elementų sąrašą
Peržiūrėti elementų sąrašą
[jei nori naikinti]
[jei nori redaguoti]
[jei nori kurti]
Naikinti elementą
Atidaryti redagavimo formą
Atidaryti kūrimo formą
Paimti duomenis
Pateikti patvirtinimo langą
Patvirtinti naikinimą
Užpildyti elemento formą
Pateikti elemento formą
Patikrinti duomenis
Pateikti klaidos pranešimą
Ar duomenys taisyklingi?
[ne]
[taip]
Išsaugoti duomenis
Pateikti atnaujintą sąrašą

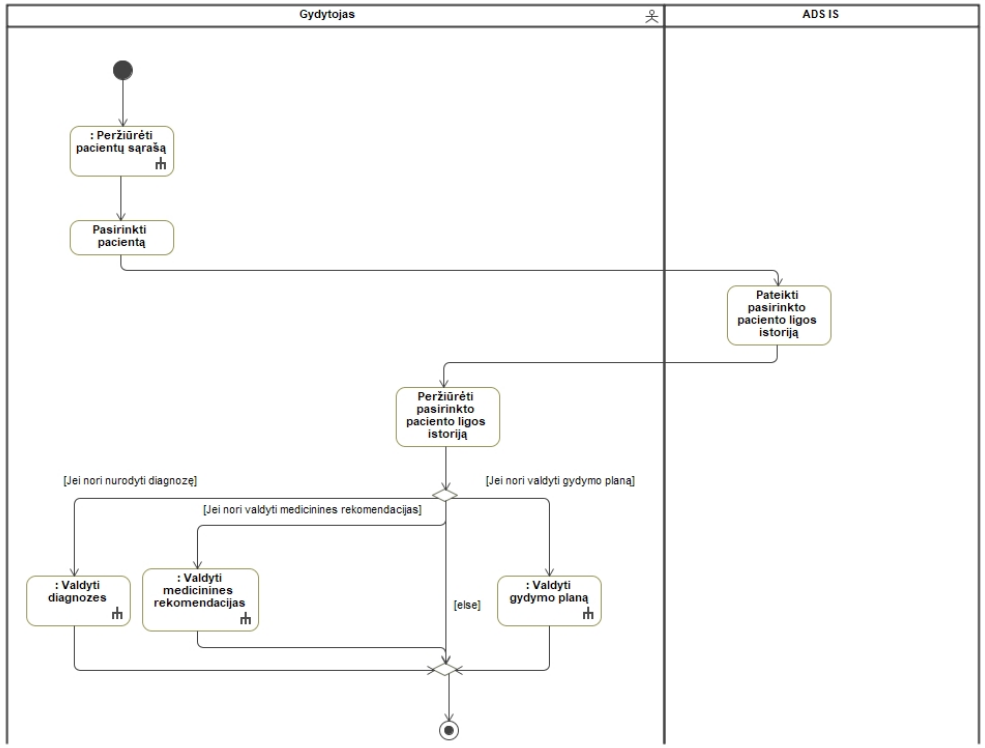

Administratorius
System
start
: Peržiūrėti elementų sąrašą
A open http: //stud1.gw.isk. ktu. lt/adsis/public/d ashboard/medi cament/create
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/d ashboard/medi cament/create
INPUT title
SELECT patient_id
SELECT doctor_id
INPUT usage_times
SELECT usage_period
SELECT usage_time
TEXTAREA usage_descript ion
FORM Submit
process form data
end

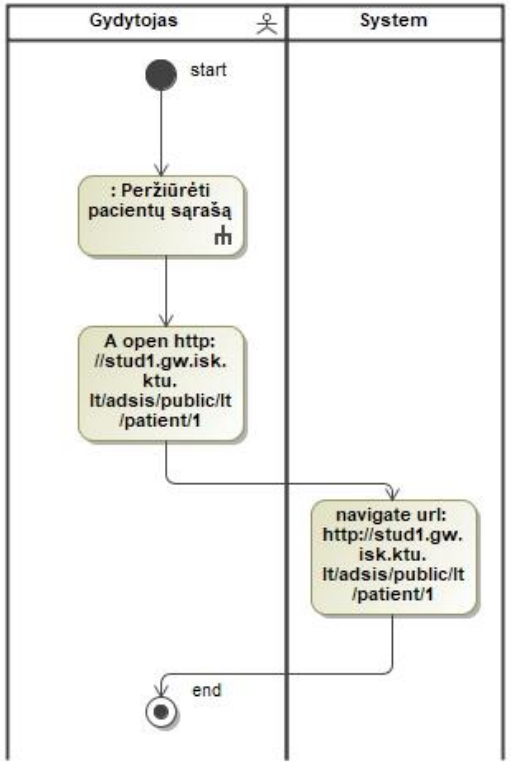

Administratorius
ADS IS
Atidaryti statistikos langą
Pateikti statistiką
Peržiūrėti statistiką

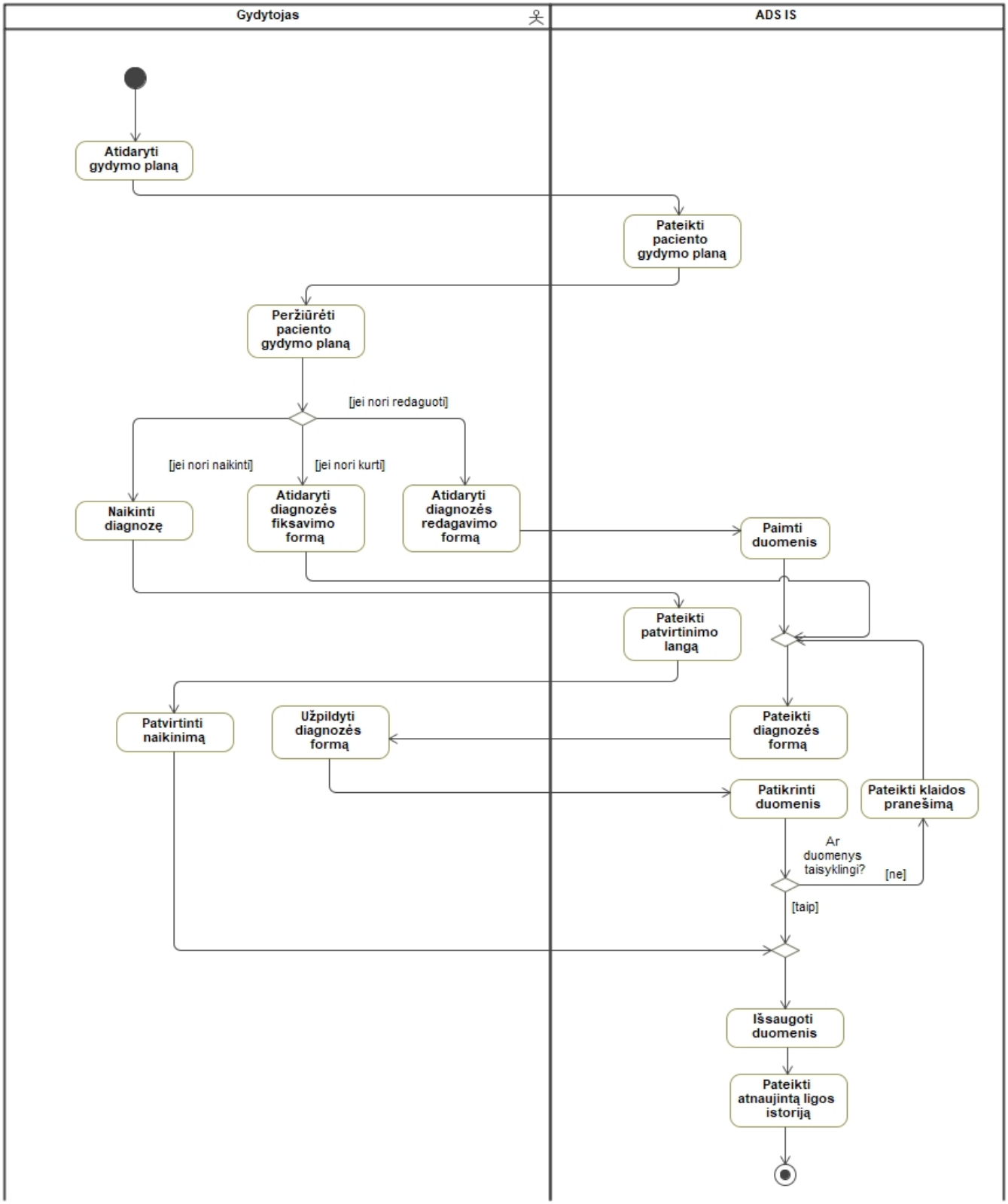

Administratorius
System
start
: Atverti valdymo panelę
A open http: //stud1.gw.isk. ktu. lt/adsis/public/d ashboard/post
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/d ashboard/post
end

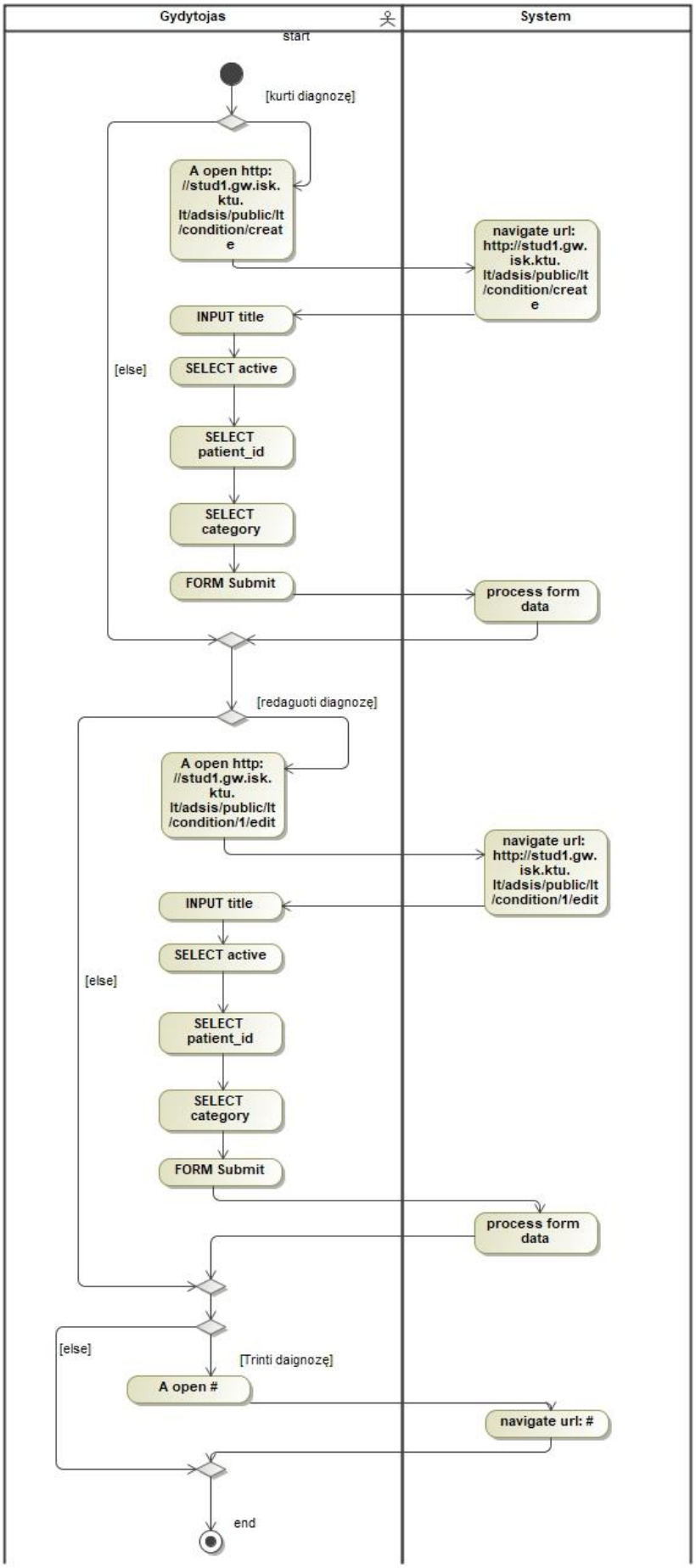

Gydytojas
ADS IS
Atidaryti pacientų sąrašo langą
Pateikti pacientų sąrašą
Peržiūrėti pacientų sąrašą
[jei nori atmesti]
[jei nori patvirtinti]
Patvirtinti pacientą
Atmesti pacientą
Išsaugoti duomenis

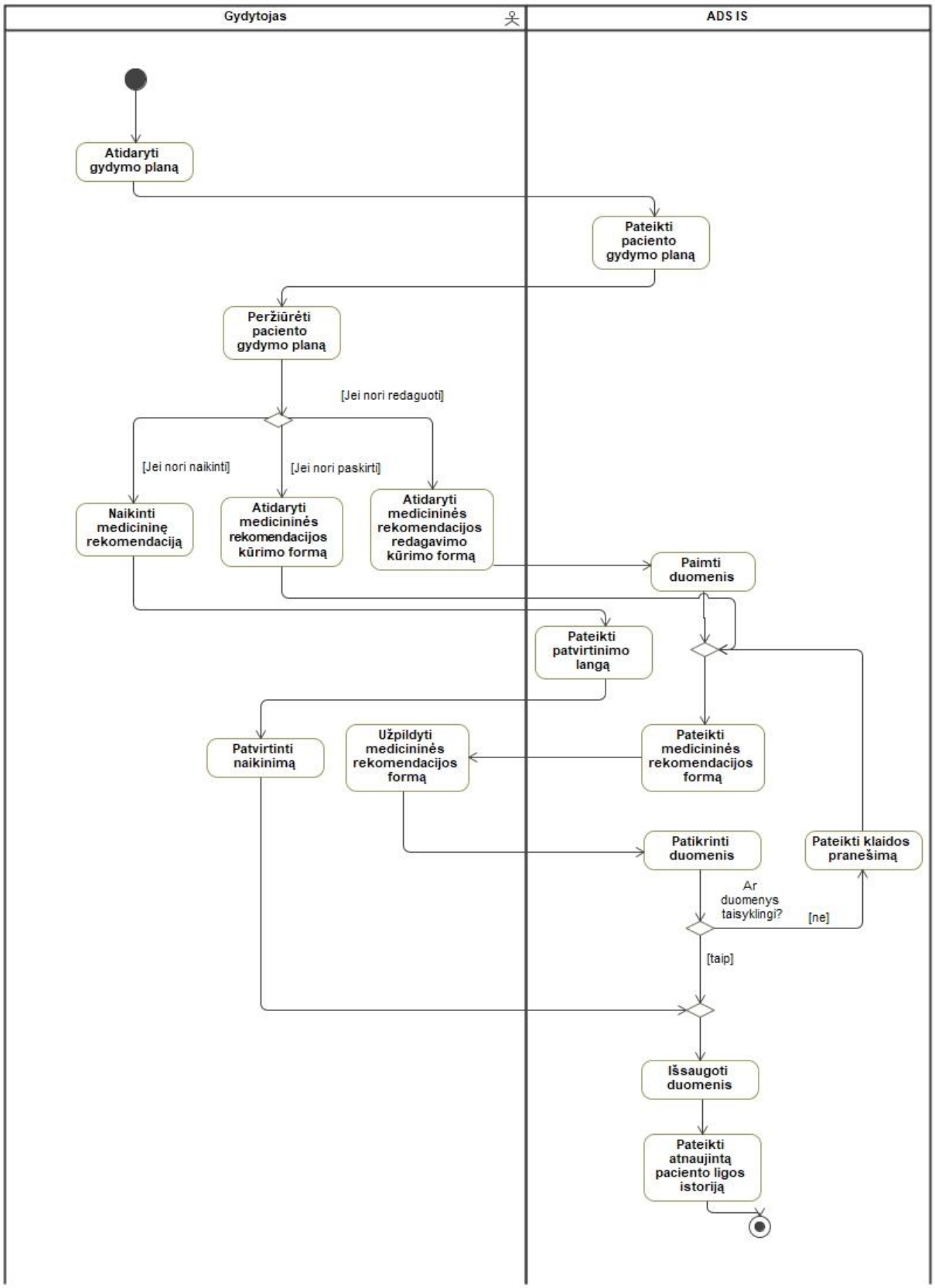

Gydytojas
System
start
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /doctor/9
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /doctor/9
end

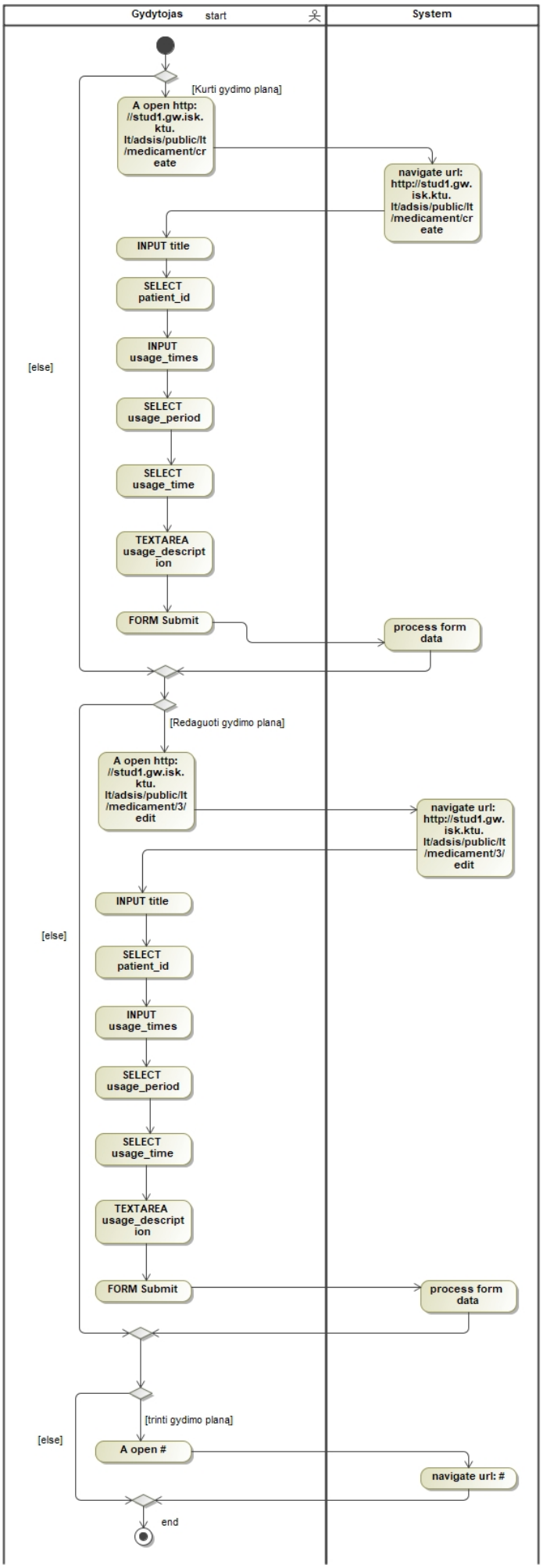

Gydytojas
ADS IS
: Peržiūrėti pacientų sąrašą
Pasirinkti pacientą
Pateikti pasirinkto paciento ligos istoriją
Peržiūrėti pasirinkto paciento ligos istoriją
[Jei nori nurodyti diagnozę]
[Jei nori valdyti gydymo planą]
[Jei nori valdyti medicinines rekomendacijas]
: Valdyti diagnozes
: Valdyti medicinines rekomendacijas
[else]
: Valdyti gydymo planą

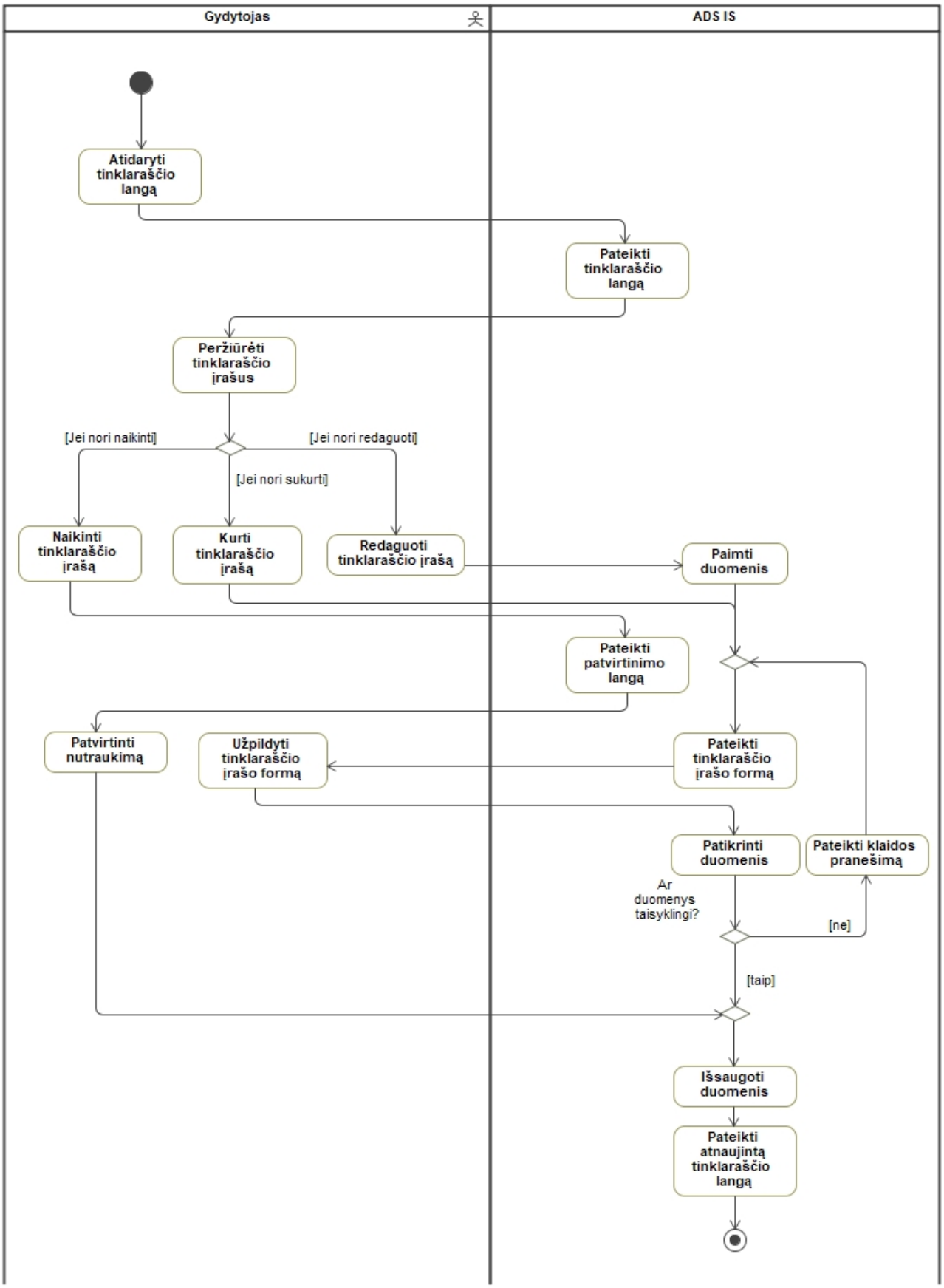

Gydytojas
System
start
: Peržiūrėti pacientų sąrašą
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/patient/1
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/patient/1
end

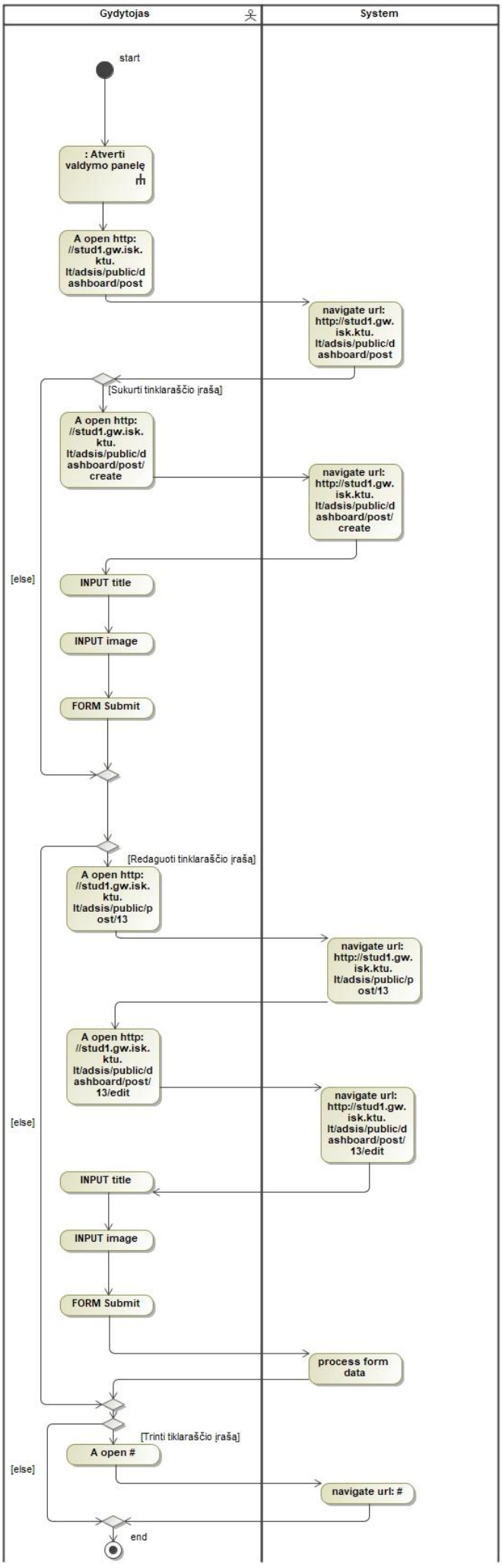

Gydytojas
ADS IS
Atidaryti gydymo planą
Pateikti paciento gydymo planą
Peržiūrėti paciento gydymo planą
[jei nori redaguoti]
[jei nori naikinti]
[jei nori kurti]
Naikinti diagnozę
Atidaryti diagnozės fiksavimo formą
Atidaryti diagnozės redagavimo formą
Paimti duomenis
Pateikti patvirtinimo langą
Patvirtinti naikinimą
Užpildyti diagnozės formą
Pateikti diagnozės formą
Patikrinti duomenis
Pateikti klaidos pranešimą
Ar duomenys taisyklingi?
[ne]
[taip]
Išsaugoti duomenis
Pateikti atnaujintą ligos istoriją

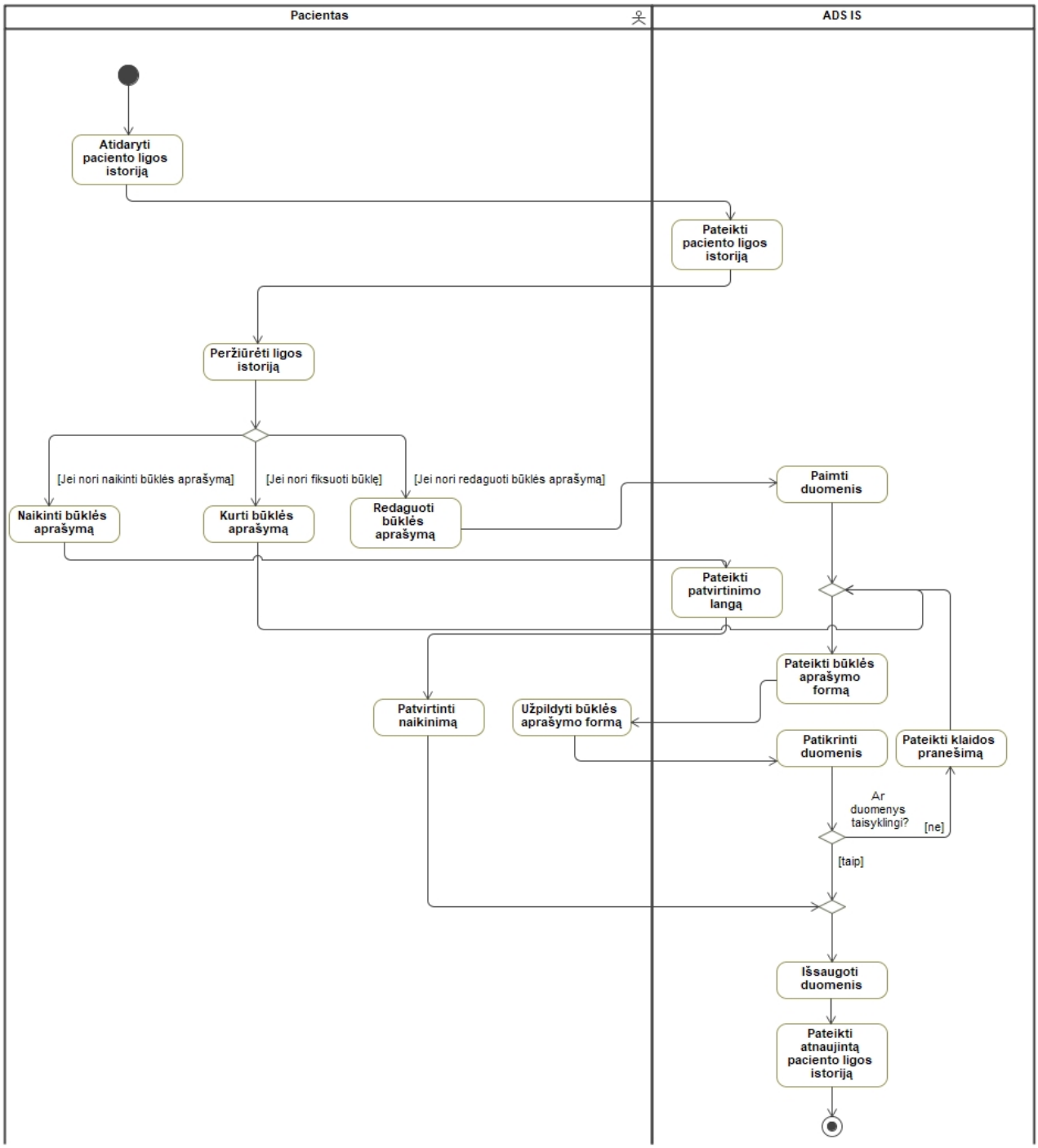

Gydytojas
System
start
[kurti diagnozę]
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/condition/create
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/condition/create
INPUT title
[else]
SELECT active
SELECT patient_id
SELECT category
FORM Submit
process form data
[redaguoti diagnozę]
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/condition/1/edit
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/condition/1/edit
INPUT title
[else]
SELECT active
SELECT patient_id
SELECT category
FORM Submit
process form data
[else]
[Trinti daignozę]
A open #
navigate url: #
end


Gydytojas
ADS IS
Atidaryti gydymo planą
Pateikti paciento gydymo planą
Peržiūrėti paciento gydymo planą
[Jei nori redaguoti]
[Jei nori naikinti]
[Jei nori paskirti]
Naikinti medicininę rekomendaciją
Atidaryti medicininės rekomendacijos kūrimo formą
Atidaryti medicininės rekomendacijos redagavimo kūrimo formą
Paimti duomenis
Pateikti patvirtinimo langą
Patvirtinti naikinimą
Užpildyti medicininės rekomendacijos formą
Pateikti medicininės rekomendacijos formą
Patikrinti duomenis
Pateikti klaidos pranešimą
Ar duomenys taisyklingi?
[ne]
[taip]
Išsaugoti duomenis
Pateikti atnaujintą paciento ligos istoriją

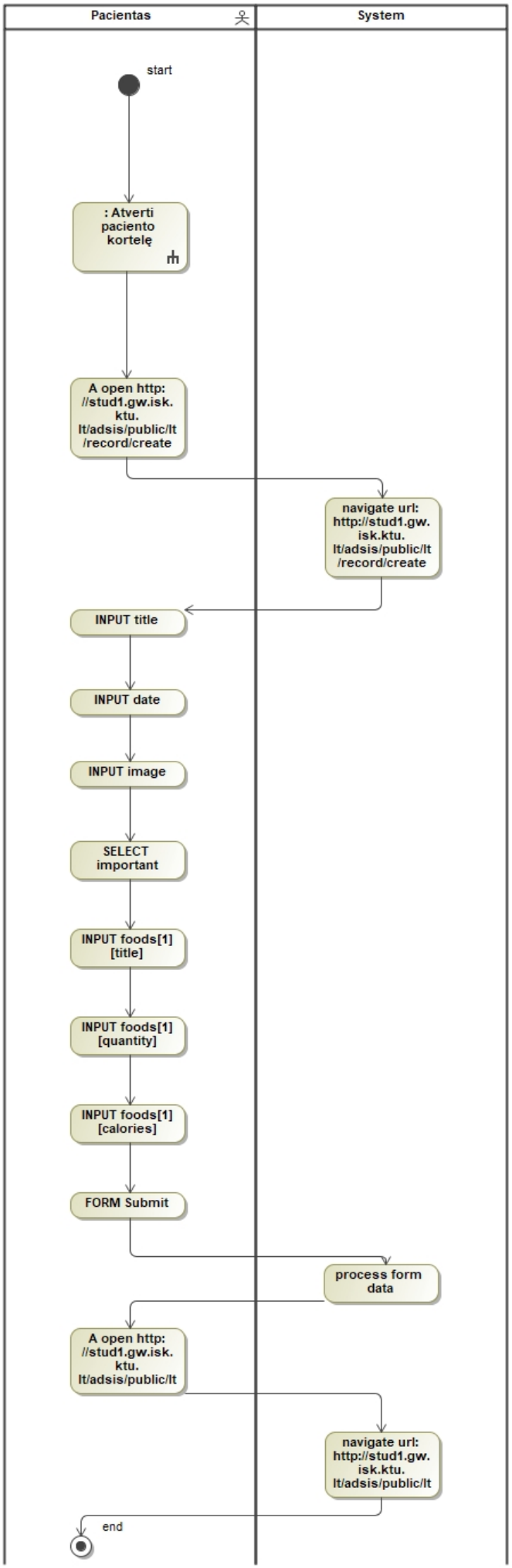

Gydytojas
start
System
[Kurti gydimo planą]
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/medicament/create
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/medicament/create
INPUT title
SELECT patient_id
INPUT usage_times
SELECT usage_period
SELECT usage_time
TEXTAREA usage_description
FORM Submit
process form data
[else]
[Redaguoti gydimo planą]
A open http://stud1.gw.isk.ktu.lt/adsis/public/lt/medicament/3/edit
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/lt/medicament/3/edit
INPUT title
SELECT patient_id
INPUT usage_times
SELECT usage_period
SELECT usage_time
TEXTAREA usage_description
FORM Submit
process form data
[else]
[trinti gydimo planą]
A open #
navigate url: #
[else]
end

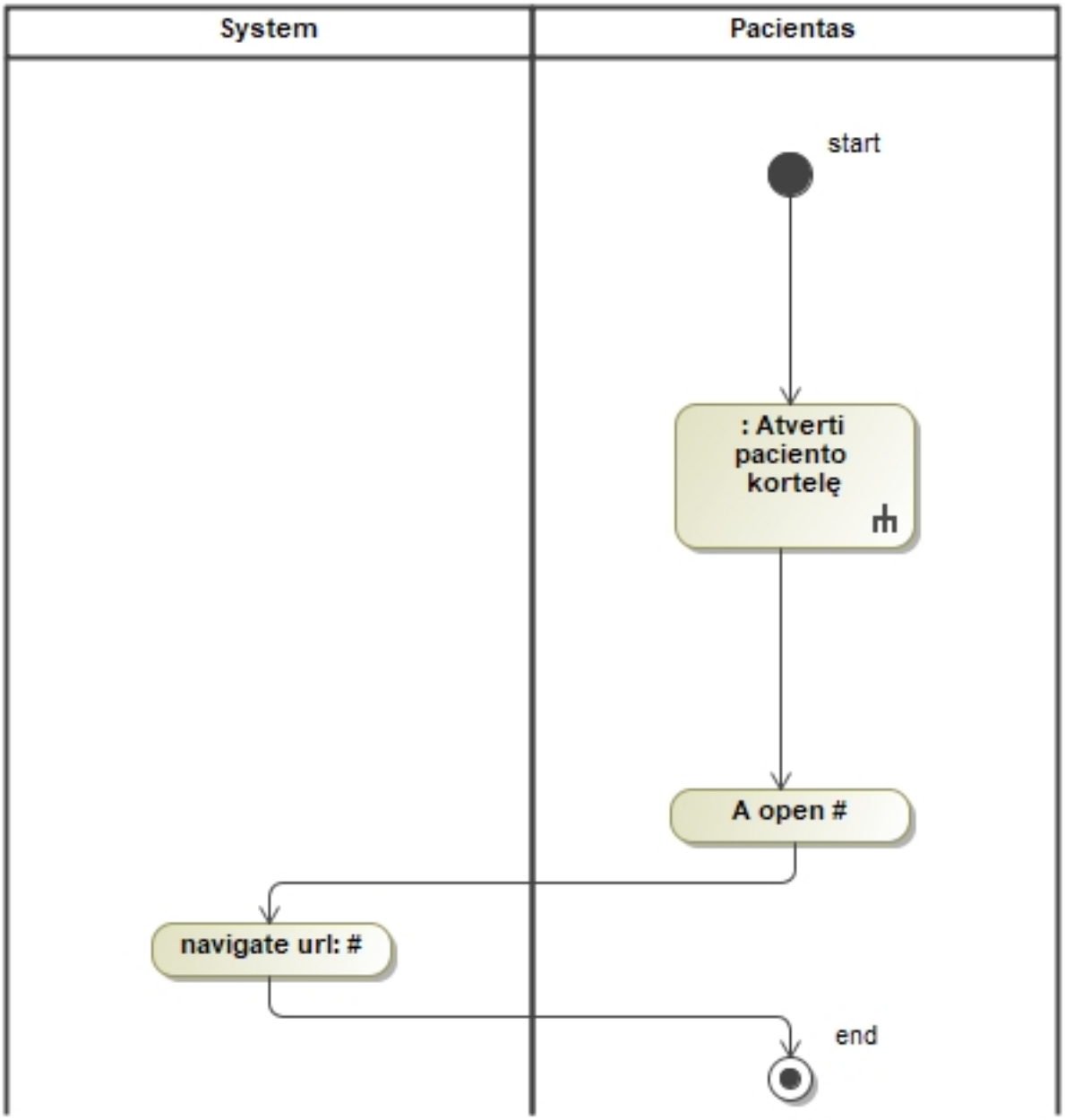

Gydytojas
ADS IS
Atidaryti tinklaraščio langą
Pateikti tinklaraščio langą
Peržiūrėti tinklaraščio įrašus
[Jei nori naikinti]
[Jei nori sukurti]
[Jei nori redaguoti]
Naikinti tinklaraščio įrašą
Kurti tinklaraščio įrašą
Redaguoti tinklaraščio įrašą
Paimti duomenis
Pateikti patvirtinimo langą
Patvirtinti nutraukimą
Užpildyti tinklaraščio įrašo formą
Pateikti tinklaraščio įrašo formą
Patikrinti duomenis
Pateikti klaidos pranešimą
Ar duomenys taisyklingi?
[ne]
[taip]
Išsaugoti duomenis
Pateikti atnaujintą tinklaraščio langą

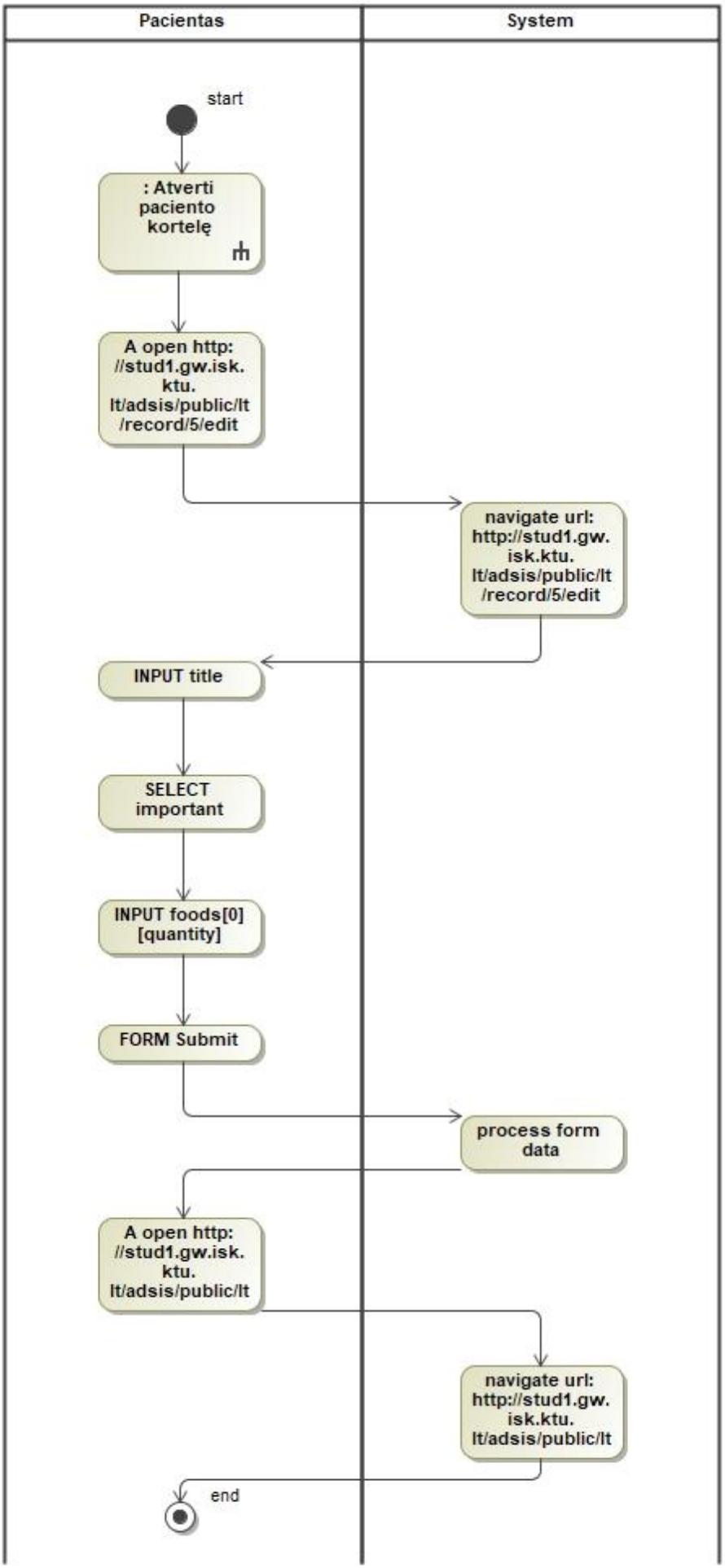

Gydytojas
System
start
: Atverti valdymo panelę
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post
[Sukurti tinklaraščio įrašą]
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post/create
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post/create
[else]
INPUT title
INPUT image
FORM Submit
[Redaguoti tinklaraščio įrašą]
A open http://stud1.gw.isk.ktu.lt/adsis/public/post/13
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/post/13
A open http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post/13/edit
navigate url: http://stud1.gw.isk.ktu.lt/adsis/public/dashboard/post/13/edit
[else]
INPUT title
INPUT image
FORM Submit
process form data
[Trinti tiklaraščio įrašą]
A open #
navigate url: #
[else]
end

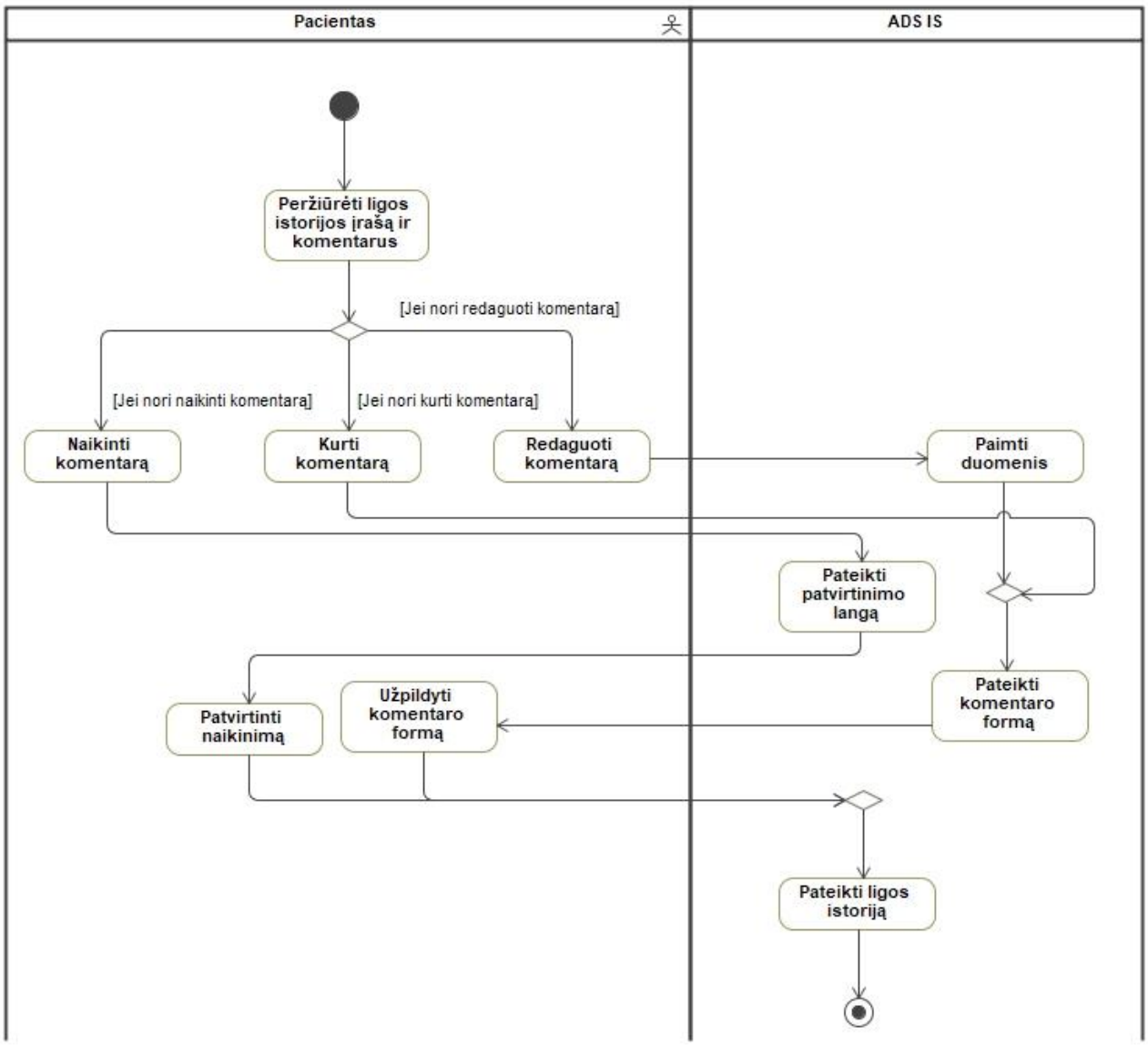


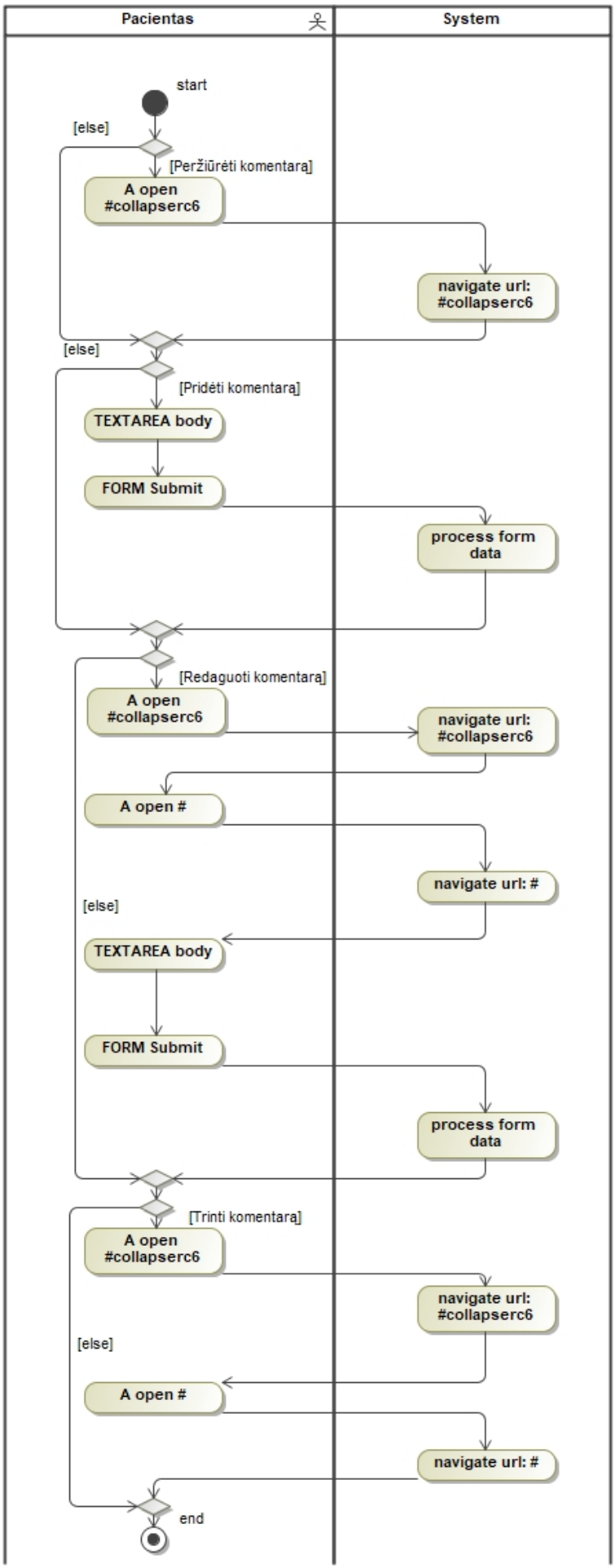

Šią diagramą sugeneruotame modelyje atitiko trys skirtingi panaudojimo atvejai.

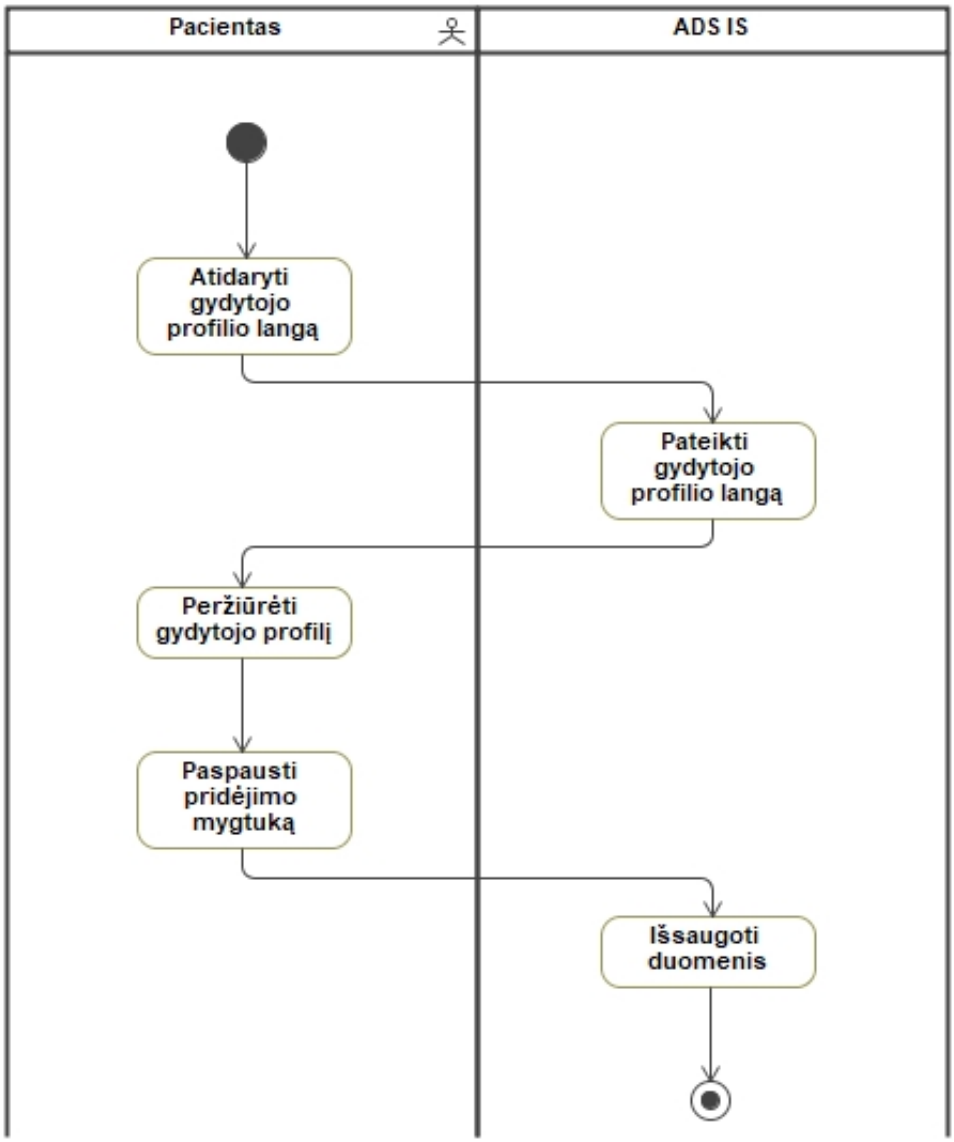

Pacientas
System
start
: Atverti paciento kortelę
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /record/create
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /record/create
INPUT title
INPUT date
INPUT image
SELECT important
INPUT foods[1] [title]
INPUT foods[1] [quantity]
INPUT foods[1] [calories]
FORM Submit
process form data
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt
end

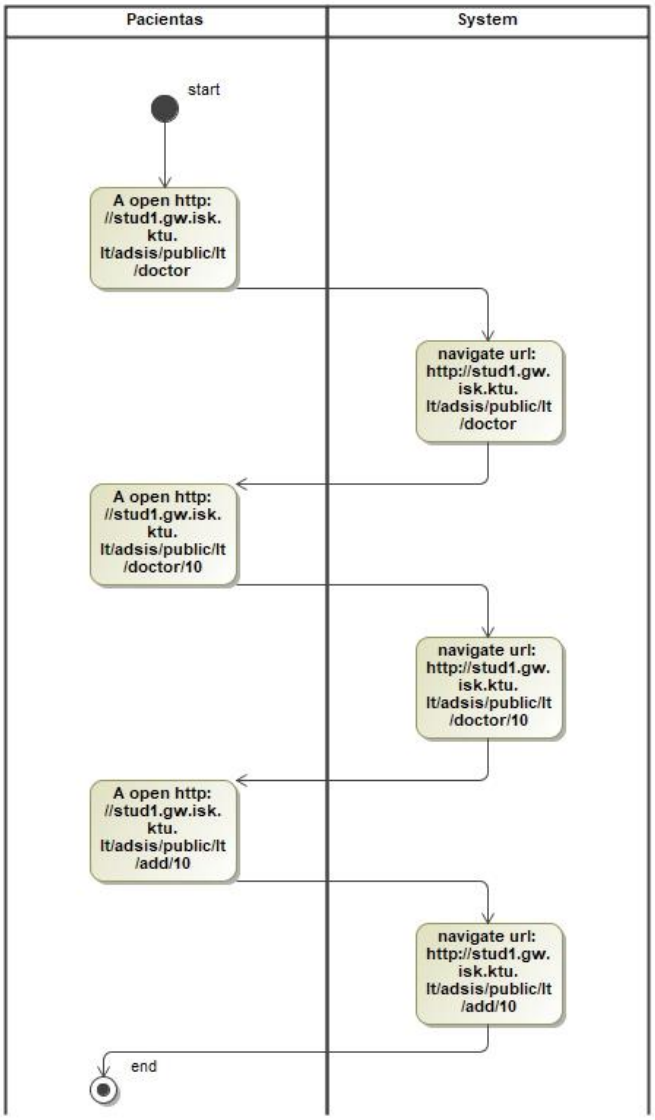

System
Pacientas
start
: Atverti paciento kortelę
A open #
navigate url: #
end


Pacientas
System
start
: Atverti paciento kortelę
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /record/5/edit
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /record/5/edit
INPUT title
SELECT important
INPUT foods[0] [quantity]
FORM Submit
process form data
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt
end


Pacientas
ADS IS
Peržiūrėti ligos istorijos įrašą ir komentarus
[Jei nori redaguoti komentarą]
[Jei nori naikinti komentarą]
[Jei nori kurti komentarą]
Naikinti komentarą
Kurti komentarą
Redaguoti komentarą
Paimti duomenis
Pateikti patvirtinimo langą
Pateikti komentaro formą
Patvirtinti naikinimą
Užpildyti komentaro formą
Pateikti ligos istoriją

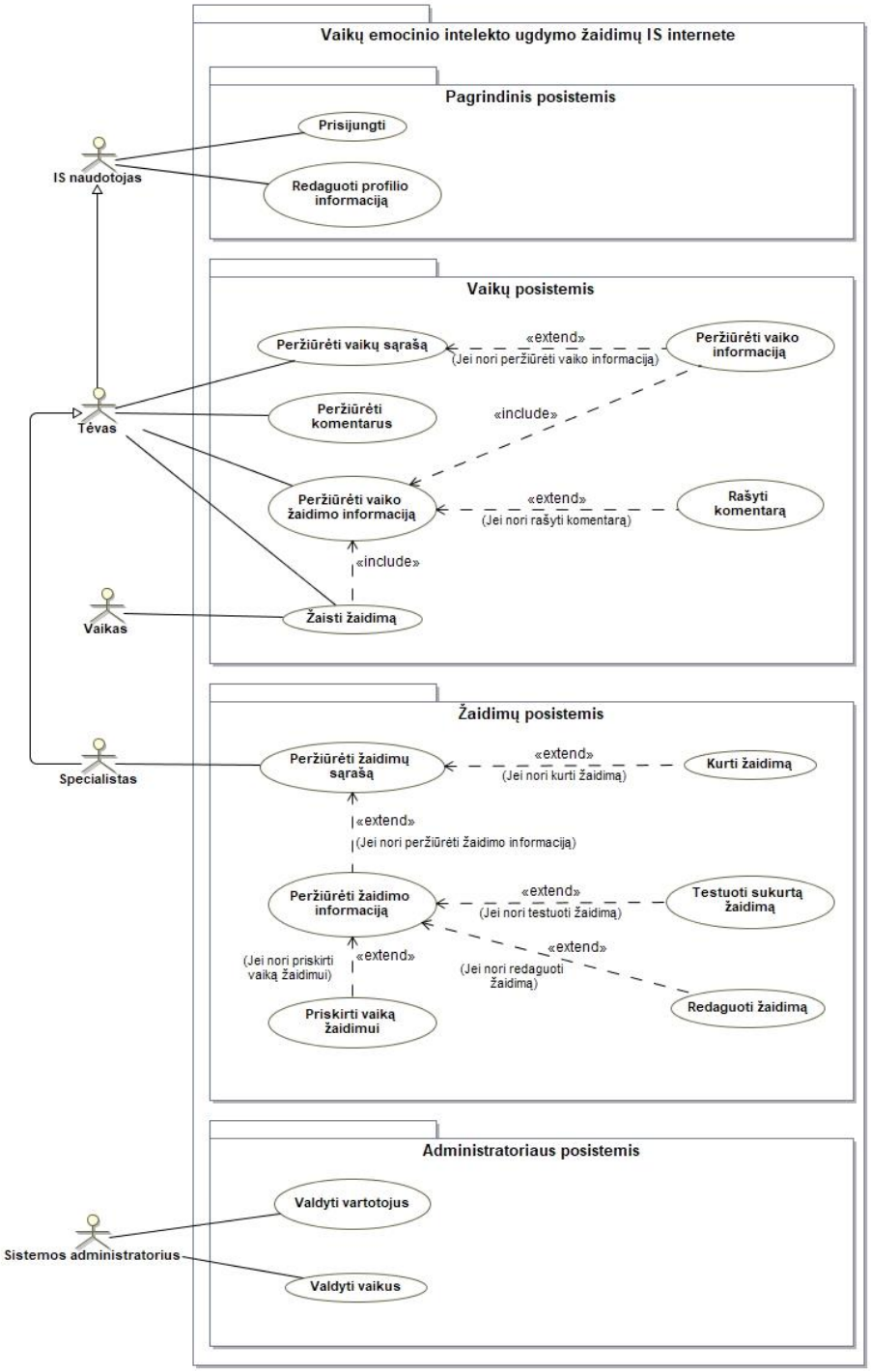

Pacientas
System
start
[else]
[Peržiūrėti komentarą]
A open #collapserc6
navigate url: #collapserc6
[else]
[Pridėti komentarą]
TEXTAREA body
FORM Submit
process form data
[Redaguoti komentarą]
A open #collapserc6
navigate url: #collapserc6
A open #
navigate url: #
[else]
TEXTAREA body
FORM Submit
process form data
[Trinti komentarą]
A open #collapserc6
navigate url: #collapserc6
[else]
A open #
navigate url: #
end


Pacientas
ADS IS
Atidaryti gydytojo profilio langą
Pateikti gydytojo profilio langą
Peržiūrėti gydytojo profilį
Paspausti pridėjimo mygtuką
Išsaugoti duomenis

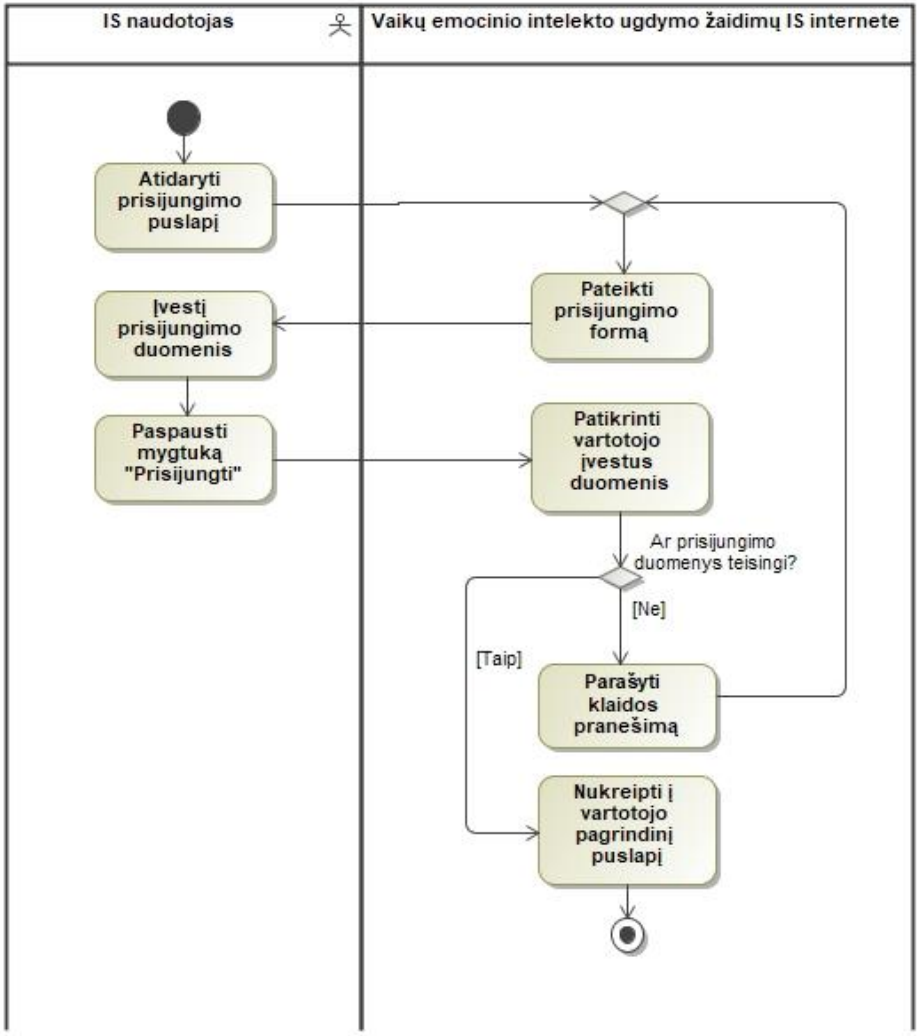

Pacientas
System
start
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /doctor
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /doctor
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /doctor/10
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /doctor/10
A open http: //stud1.gw.isk. ktu. lt/adsis/public/lt /add/10
navigate url: http://stud1.gw. isk.ktu. lt/adsis/public/lt /add/10
end

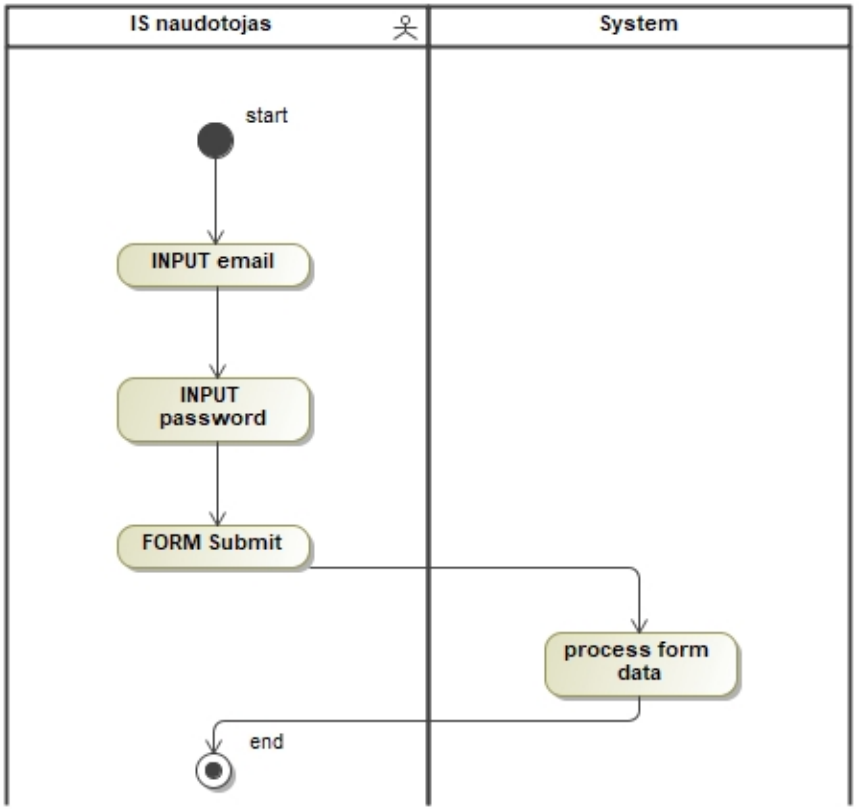
### 9.1.2. priedas „VEUŽIS“ panaudojimo atvejų modelis ir sugeneruotas panaudojimo atvejų modelis

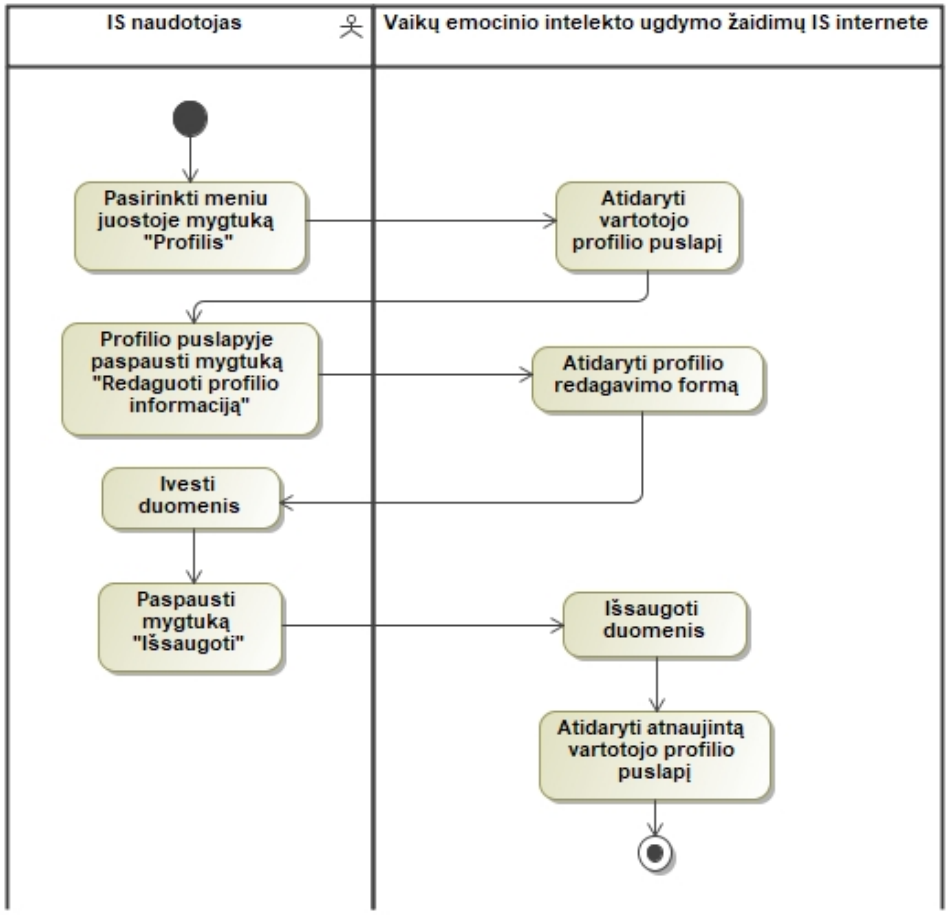
Šiame priede pirma pateikiama originalioji diagrama iškarto po jos viena sugeneruota diagrama, jeigu neprarašyta, kad vieną diagramą sugeneruotoje diagramoje atvaizduoja keletas diagramų.

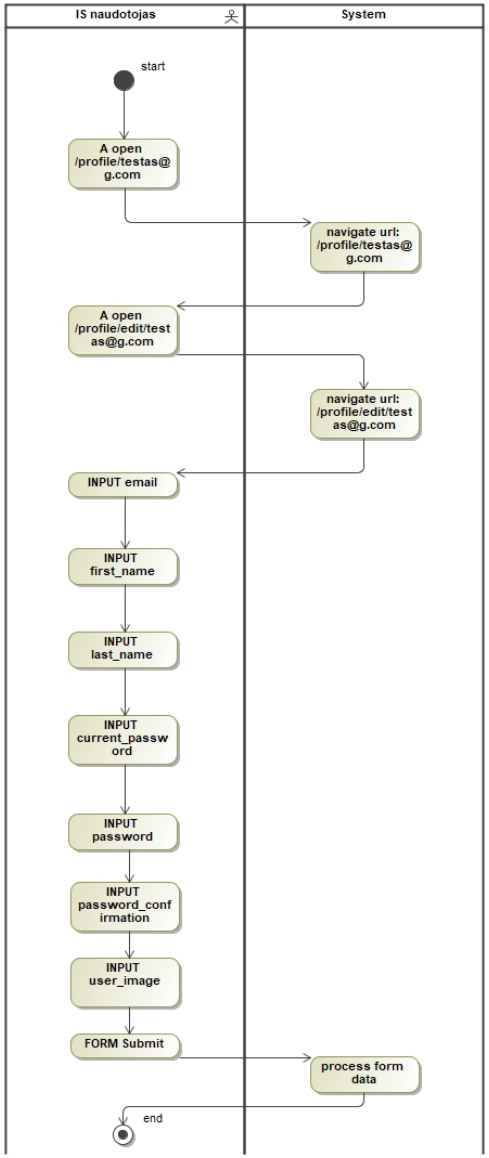


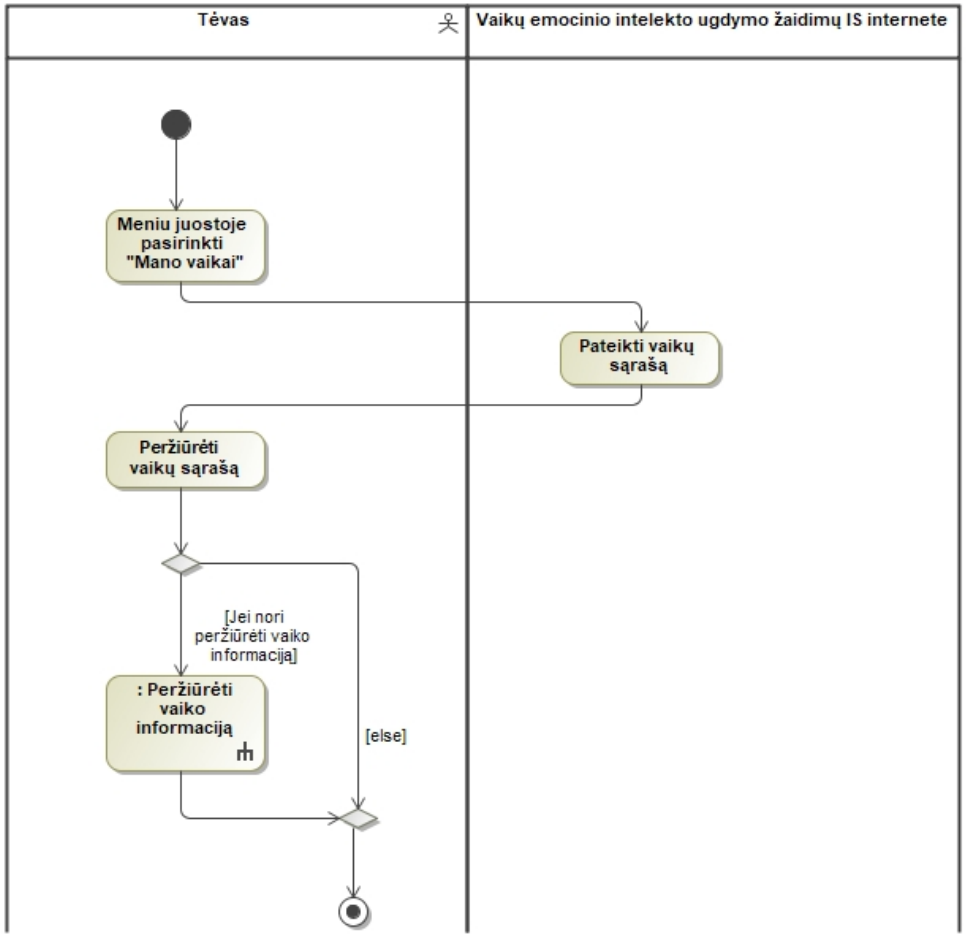

Prisijungti
IS naudotojas
Redaguoti profilio informaciją
Peržiūrėti vaiko žaidimo informaciją
Žaisti žaidimą
«extend»
«include»
«include»
Rašyti komentarą
Peržiūrėti vaikų sąrašą
«include»
Tėvas
Peržiūrėti vaiko informaciją
Peržiūrėti komentarus
Specialistas
Peržiūrėti žaidimo informaciją
Sistemos administratorius
Priskirti vaiką žaidimui
«extend»
Pridėti žaidimą
Peržiūrėti žaidimų sąrašą
«extend»
Testuoti žaidimą
«extend»
Redaguoti žaidimą
Valdyti vaikus
Valdyti vartotojus
«extend»
Valdyti vartotojus
«extend»
Pridėti vaiką

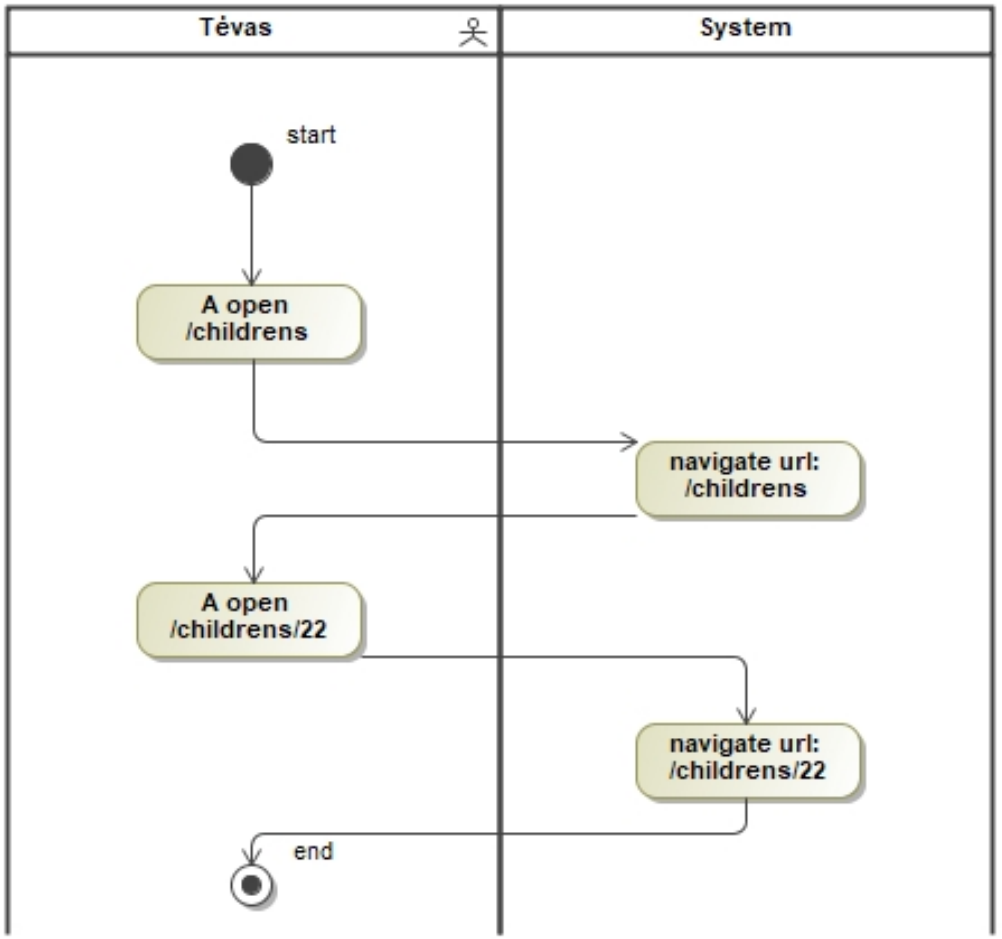

IS naudotojas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Atidaryti prisijungimo puslapį
Įvesti prisijungimo duomenis
Paspausti mygtuką "Prisijungti"
Pateikti prisijungimo formą
Patikrinti vartotojo įvestus duomenis
Ar prisijungimo duomenys teisingi?
[Ne]
[Taip]
Parašyti klaidos pranešimą
Nukreipti į vartotojo pagrindinį puslapį

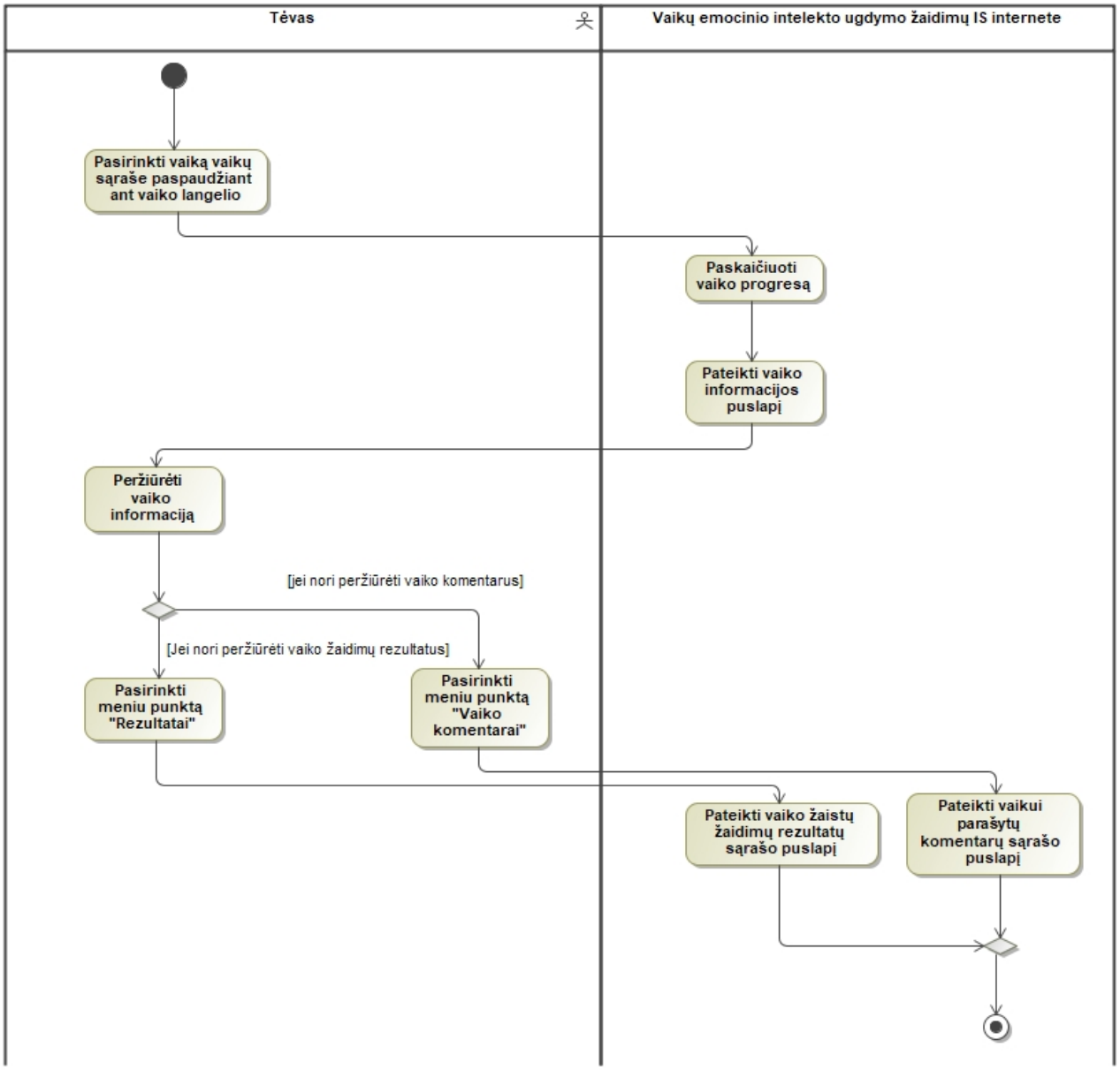

IS naudotojas
System
start
INPUT email
INPUT password
FORM Submit
process form data
end

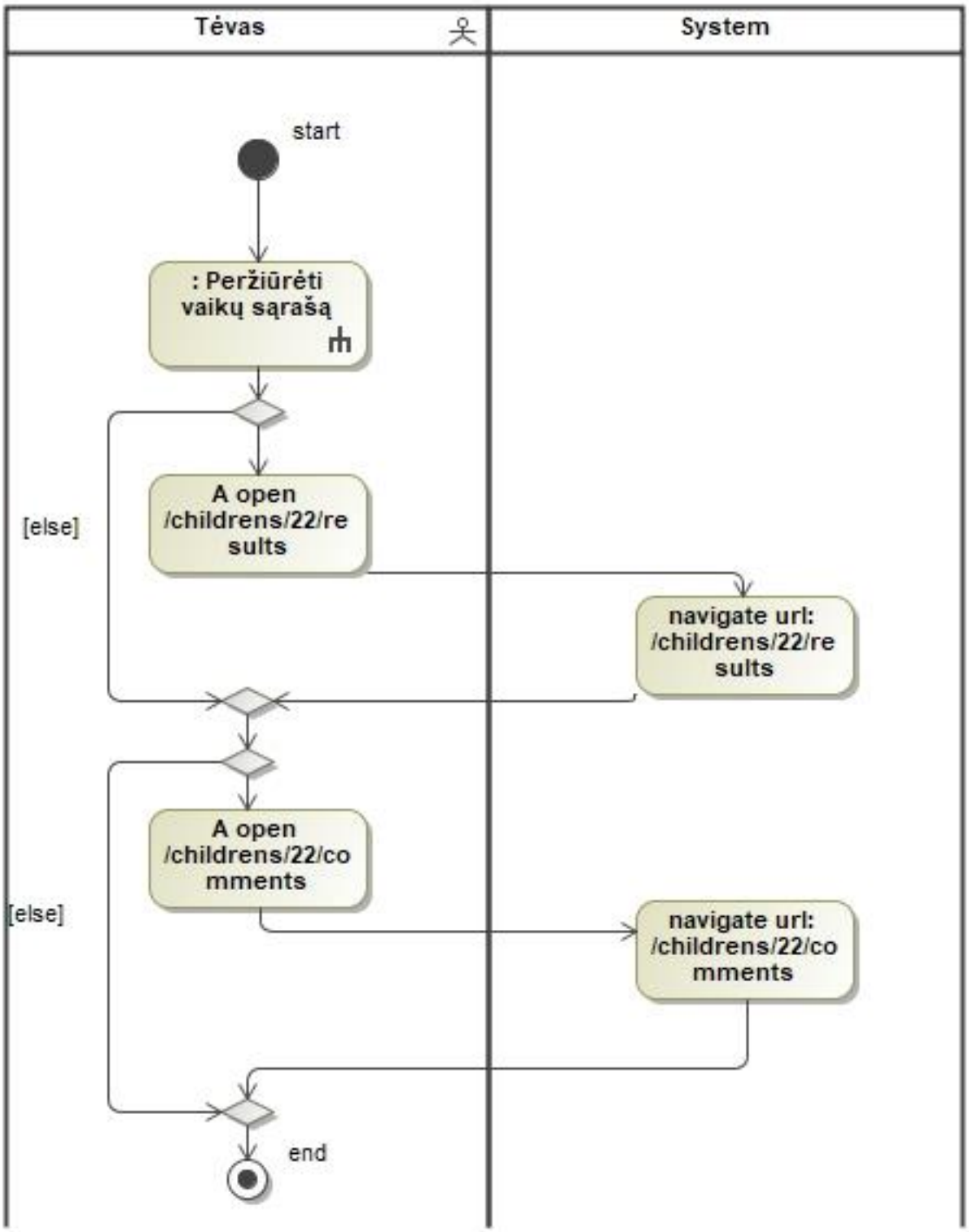

IS naudotojas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Pasirinkti meniu juostoje mygtuką "Profilis"
Atidaryti vartotojo profilio puslapį
Profilio puslapyje paspausti mygtuką "Redaguoti profilio informaciją"
Atidaryti profilio redagavimo formą
Ivesti duomenis
Paspausti mygtuką "Išsaugoti"
Išsaugoti duomenis
Atidaryti atnaujintą vartotojo profilio puslapį

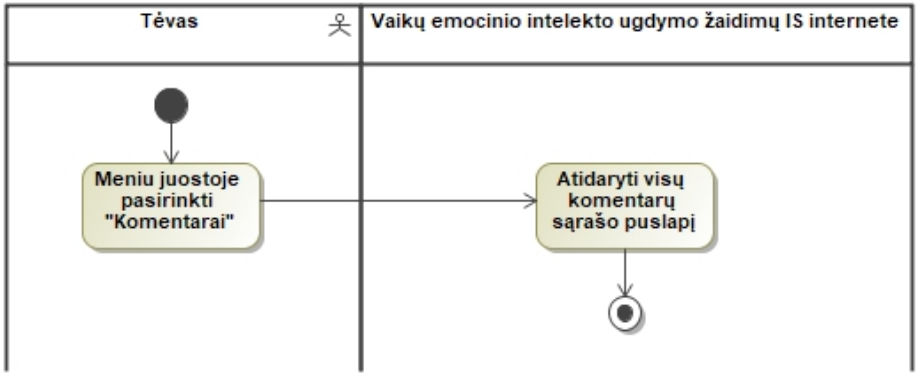

IS naudotojas
System
start
A open /profile/testas@g.com
navigate url: /profile/testas@g.com
A open /profile/edit/testas@g.com
navigate url: /profile/edit/testas@g.com
INPUT email
INPUT first_name
INPUT last_name
INPUT current_password
INPUT password
INPUT password_confirmation
INPUT user_image
FORM Submit
process form data
end

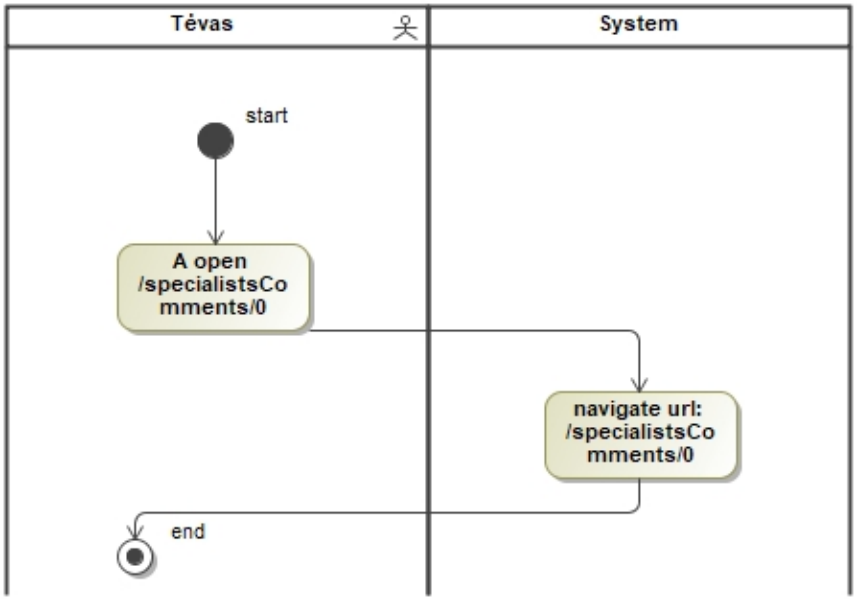

Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Meniu juostoje pasirinkti "Mano vaikai"
Pateikti vaikų sąrašą
Peržiūrėti vaikų sąrašą
[Jei nori peržiūrėti vaiko informaciją]
: Peržiūrėti vaiko informaciją
[else]

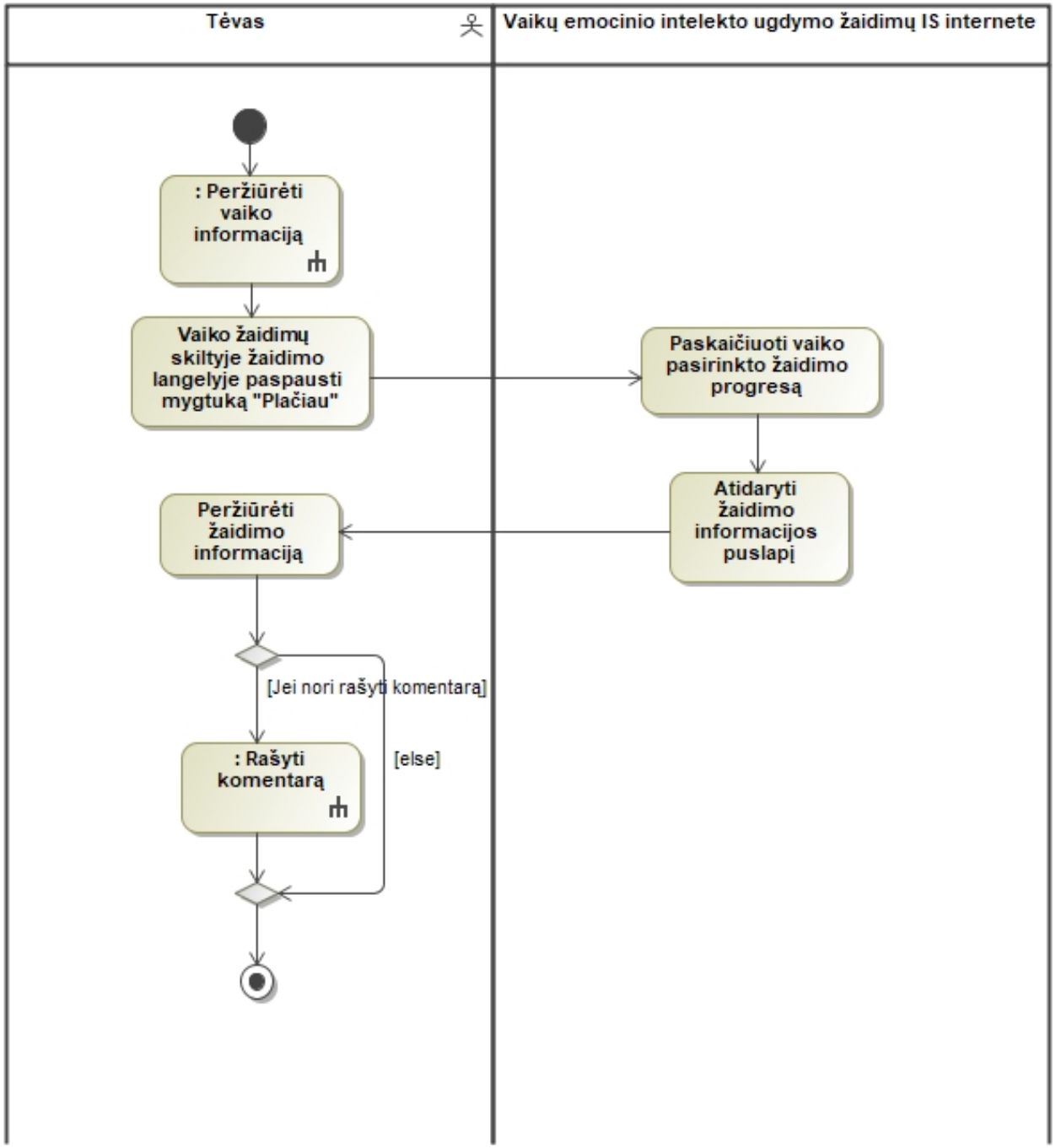

Tėvas
System
start
A open /childrens
navigate url: /childrens
A open /childrens/22
navigate url: /childrens/22
end

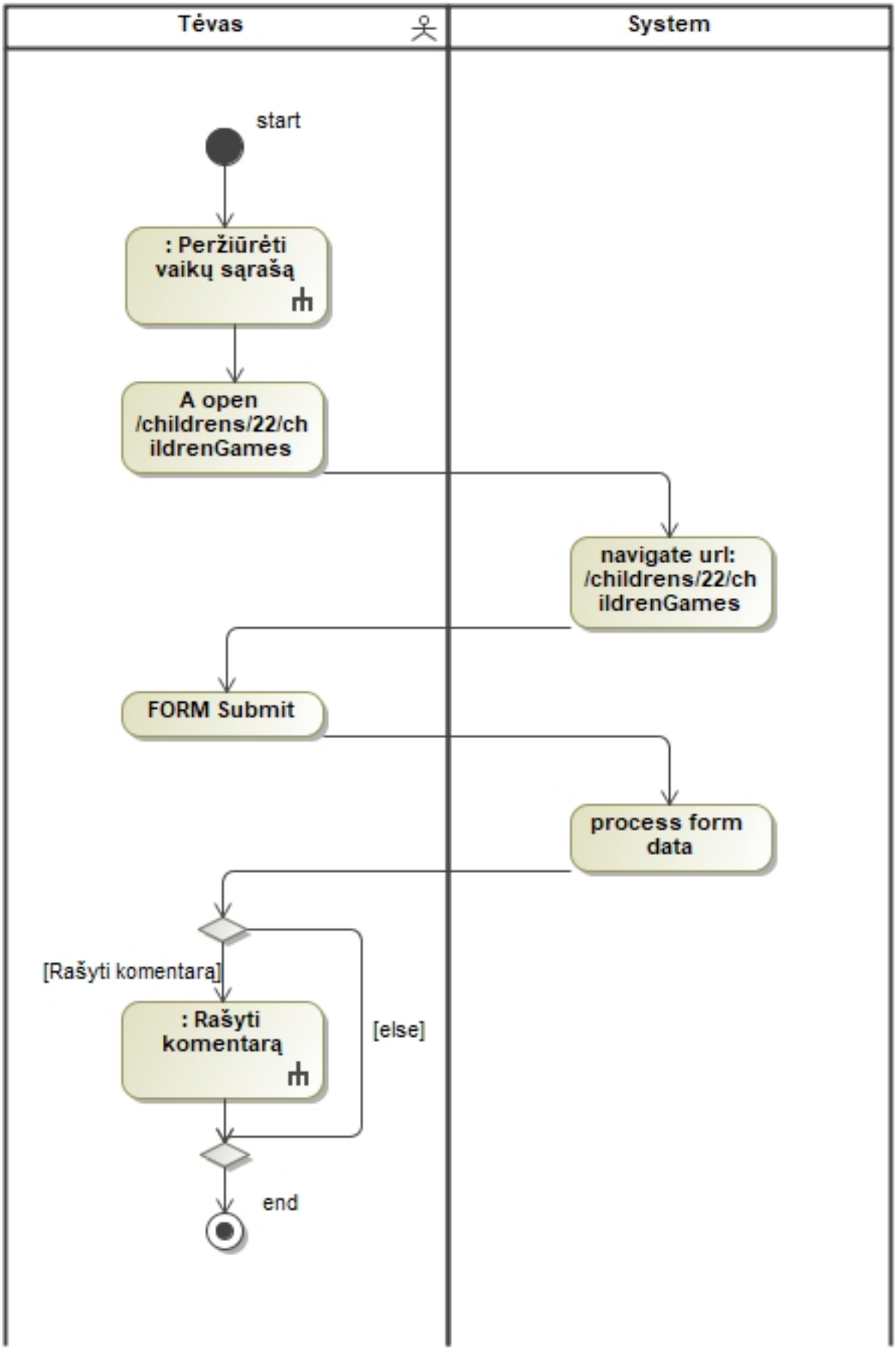

Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Pasirinkti vaiką vaikų sąraše paspaudžiant ant vaiko langelio
Paskaičiuoti vaiko progresą
Pateikti vaiko informacijos puslapį
Peržiūrėti vaiko informaciją
[jei nori peržiūrėti vaiko komentarus]
[Jei nori peržiūrėti vaiko žaidimų rezultatus]
Pasirinkti meniu punktą "Rezultatai"
Pasirinkti meniu punktą "Vaiko komentarai"
Pateikti vaiko žaistų žaidimų rezultatų sąrašo puslapį
Pateikti vaikui parašytų komentarų sąrašo puslapį

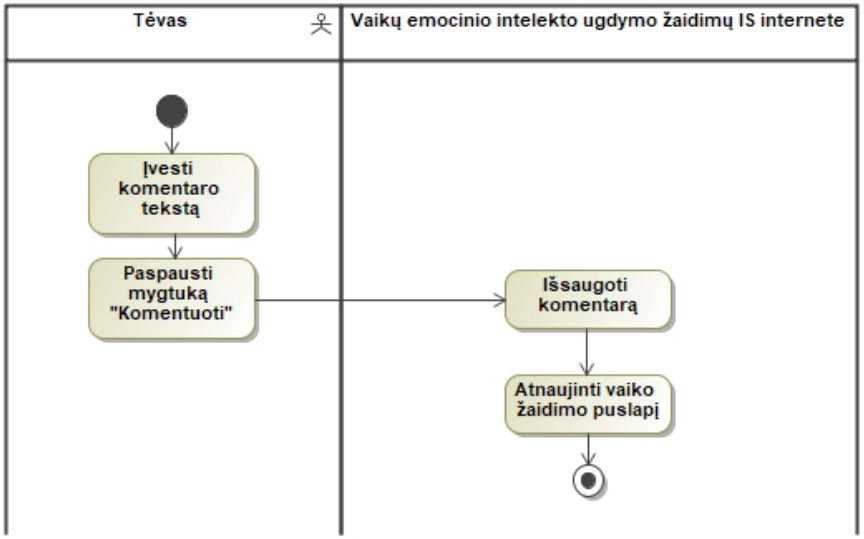

Tėvas
System
start
: Peržiūrėti vaikų sąrašą
A open /childrens/22/results
[else]
navigate url: /childrens/22/results
A open /childrens/22/comments
[else]
navigate url: /childrens/22/comments
end

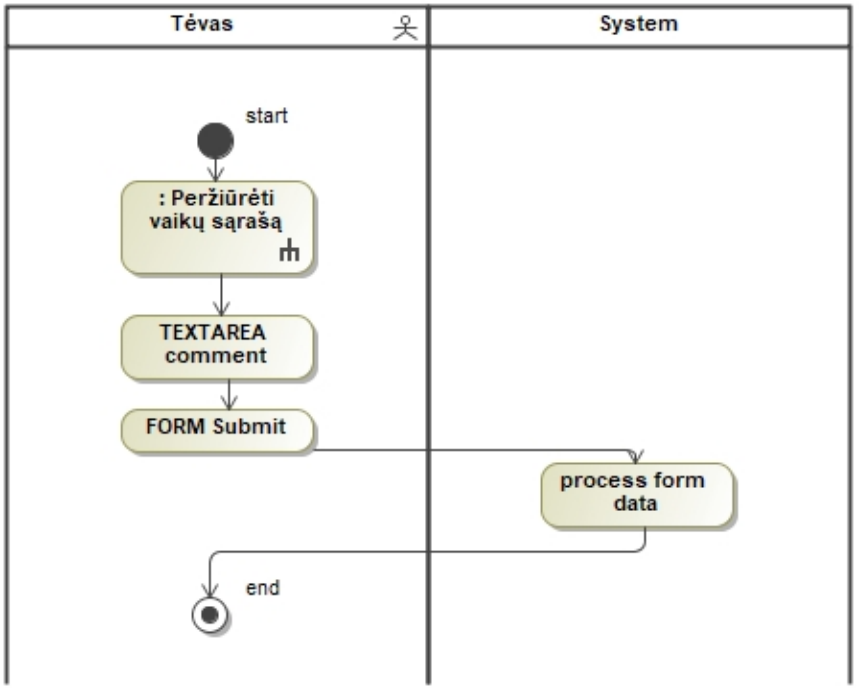

Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Meniu juostoje pasirinkti "Komentarai"
Atidaryti visų komentarų sąrašo puslapį

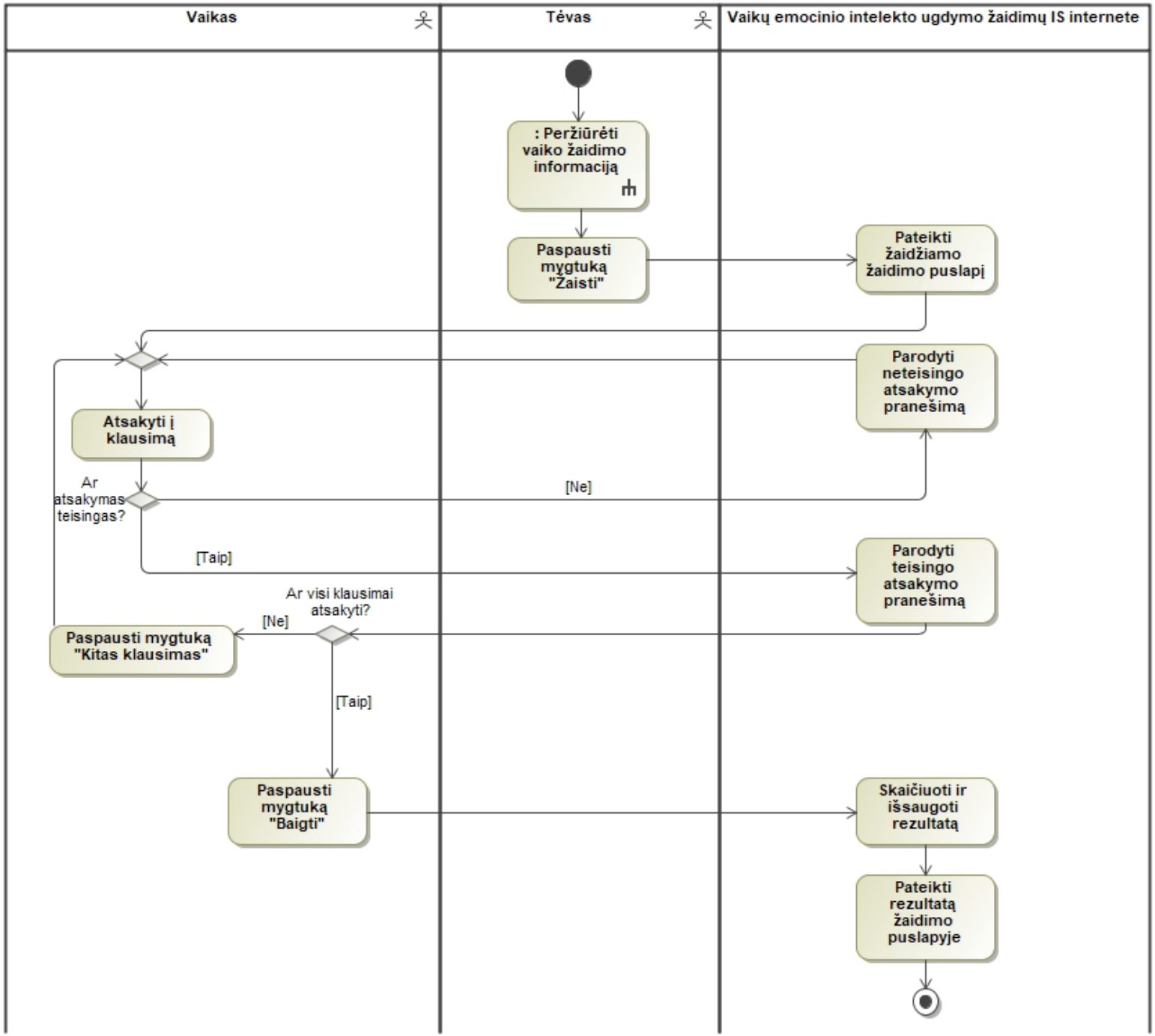

Tėvas
System
start
A open /specialistsComments/0
navigate url: /specialistsComments/0
end

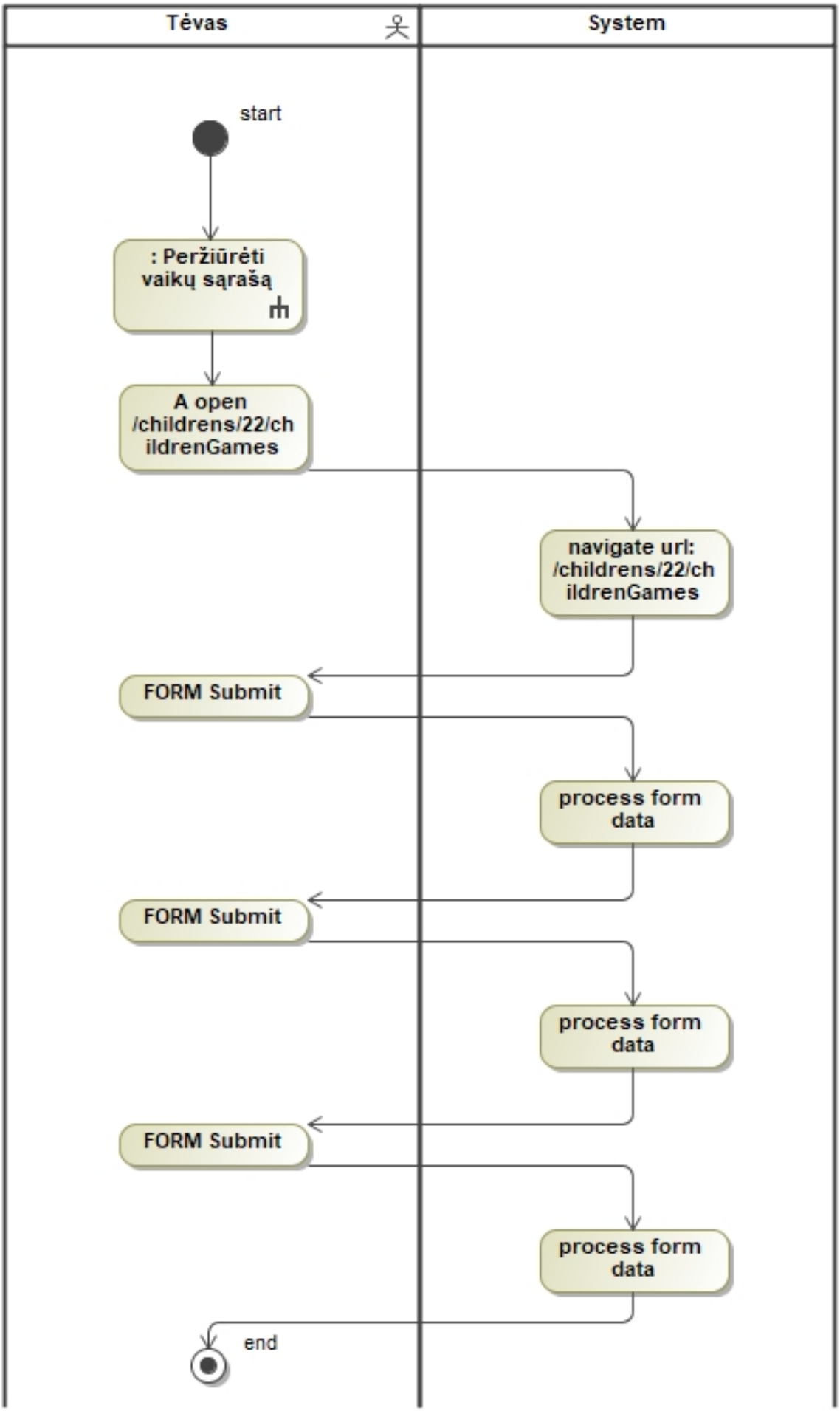

Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
: Peržiūrėti vaiko informaciją
Vaiko žaidimų skiltyje žaidimo langelyje paspausti mygtuką "Plačiau"
Paskaičiuoti vaiko pasirinkto žaidimo progresą
Atidaryti žaidimo informacijos puslapį
Peržiūrėti žaidimo informaciją
[Jei nori rašyti komentarą]
[else]
: Rašyti komentarą

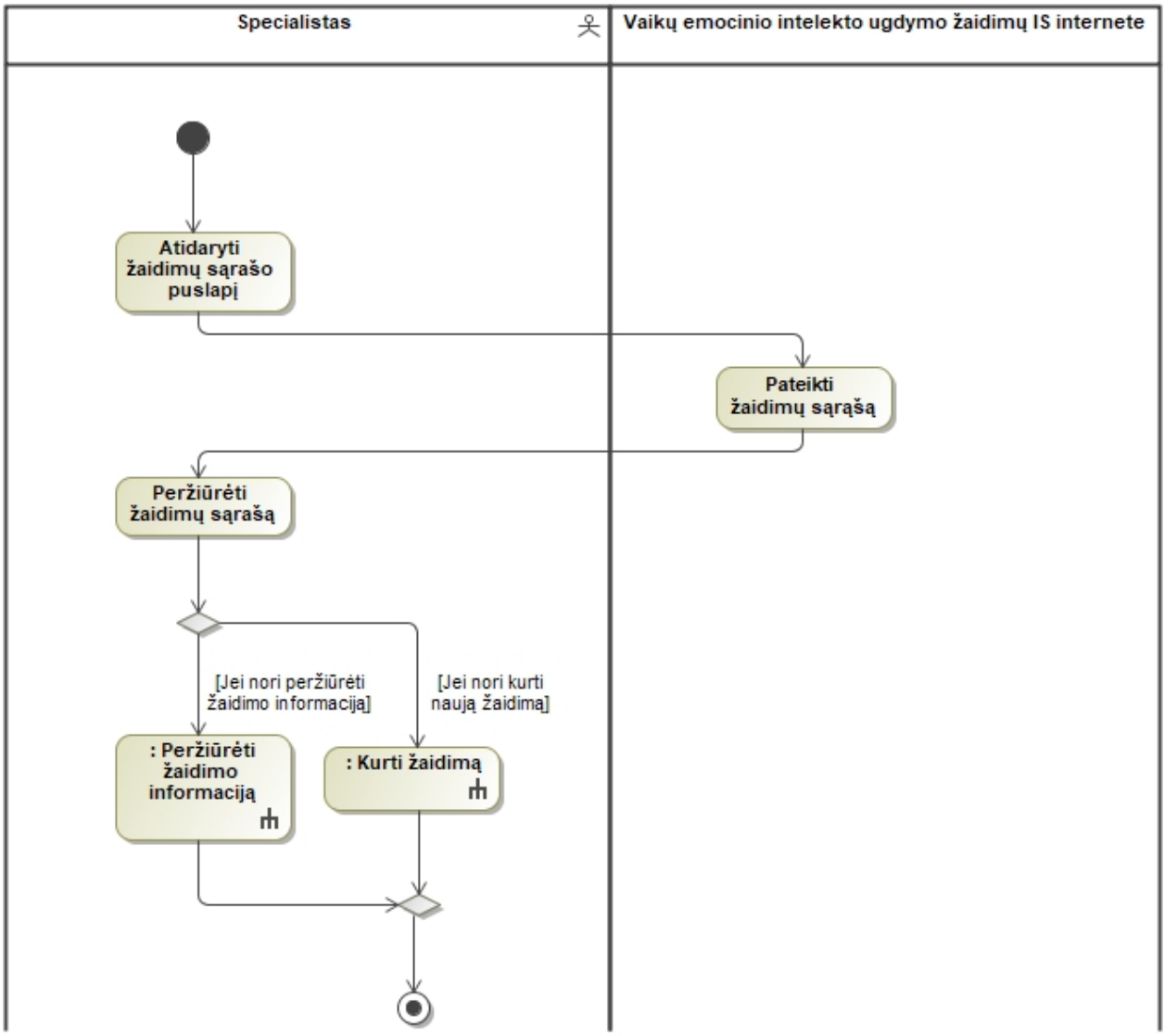

Tėvas
System
start
: Peržiūrėti vaikų sąrašą
A open /childrens/22/childrenGames
navigate url: /childrens/22/childrenGames
FORM Submit
process form data
[Rašyti komentarą]
: Rašyti komentarą
[else]
end

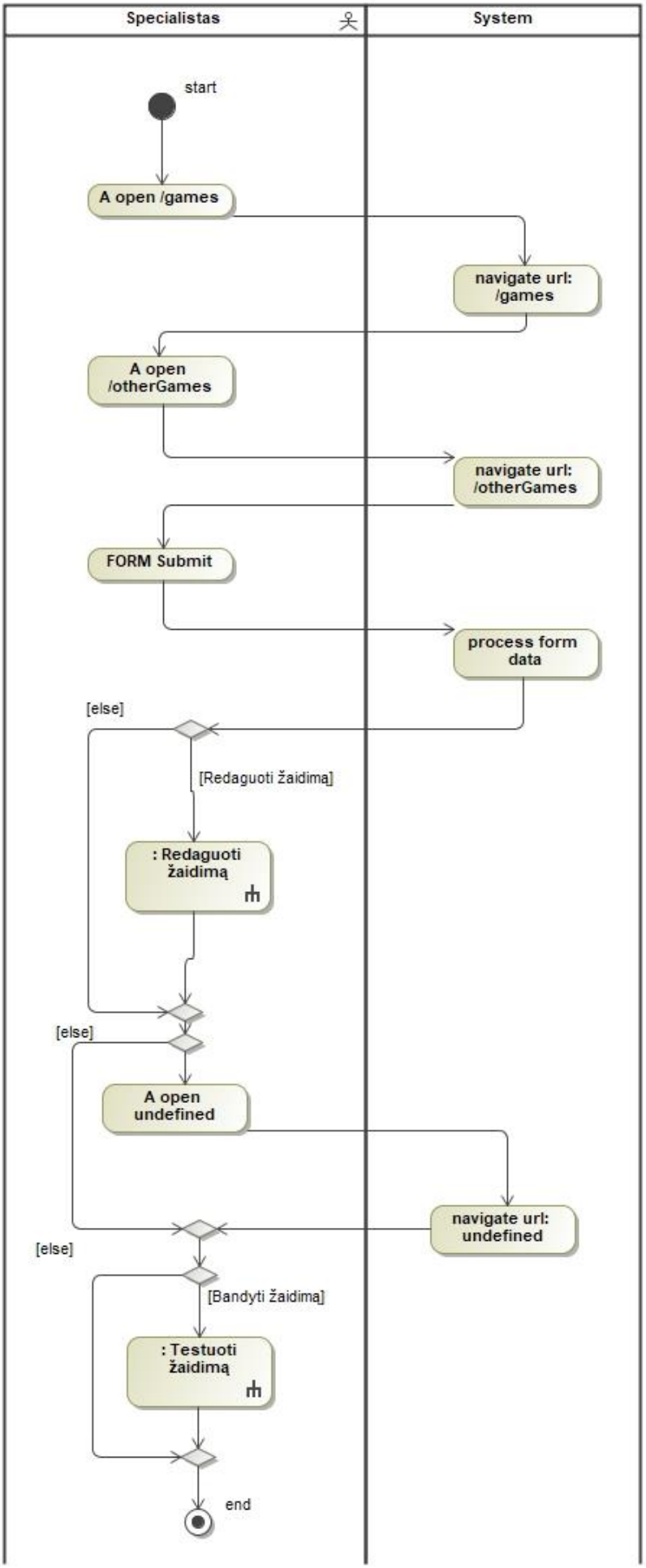

Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Įvesti komentaro tekstą
Paspausti mygtuką "Komentuoti"
Išsaugoti komentarą
Atnaujinti vaiko žaidimo puslapį

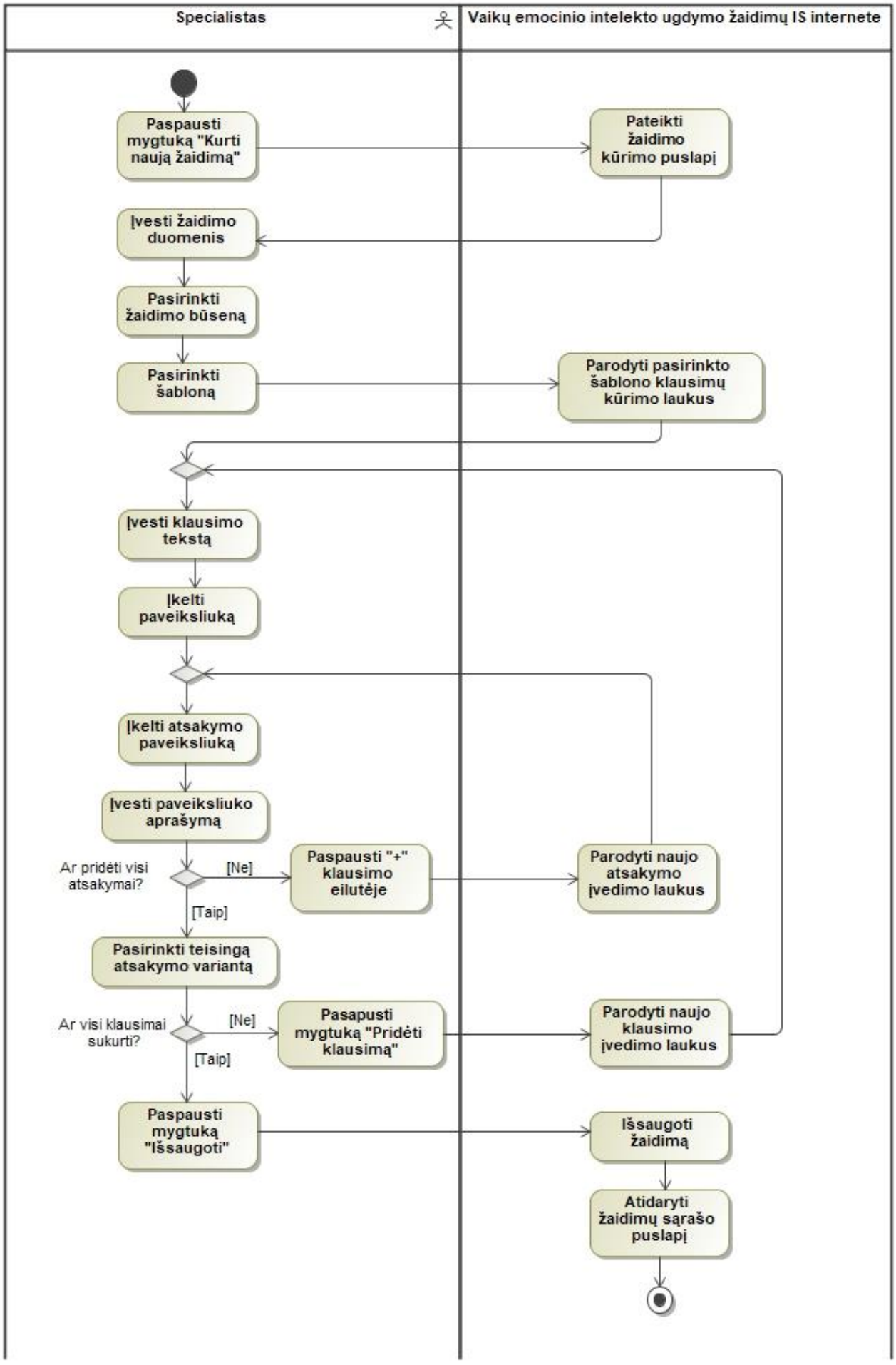

Tėvas
System
start
: Peržiūrėti vaikų sąrašą
TEXTAREA comment
FORM Submit
process form data
end

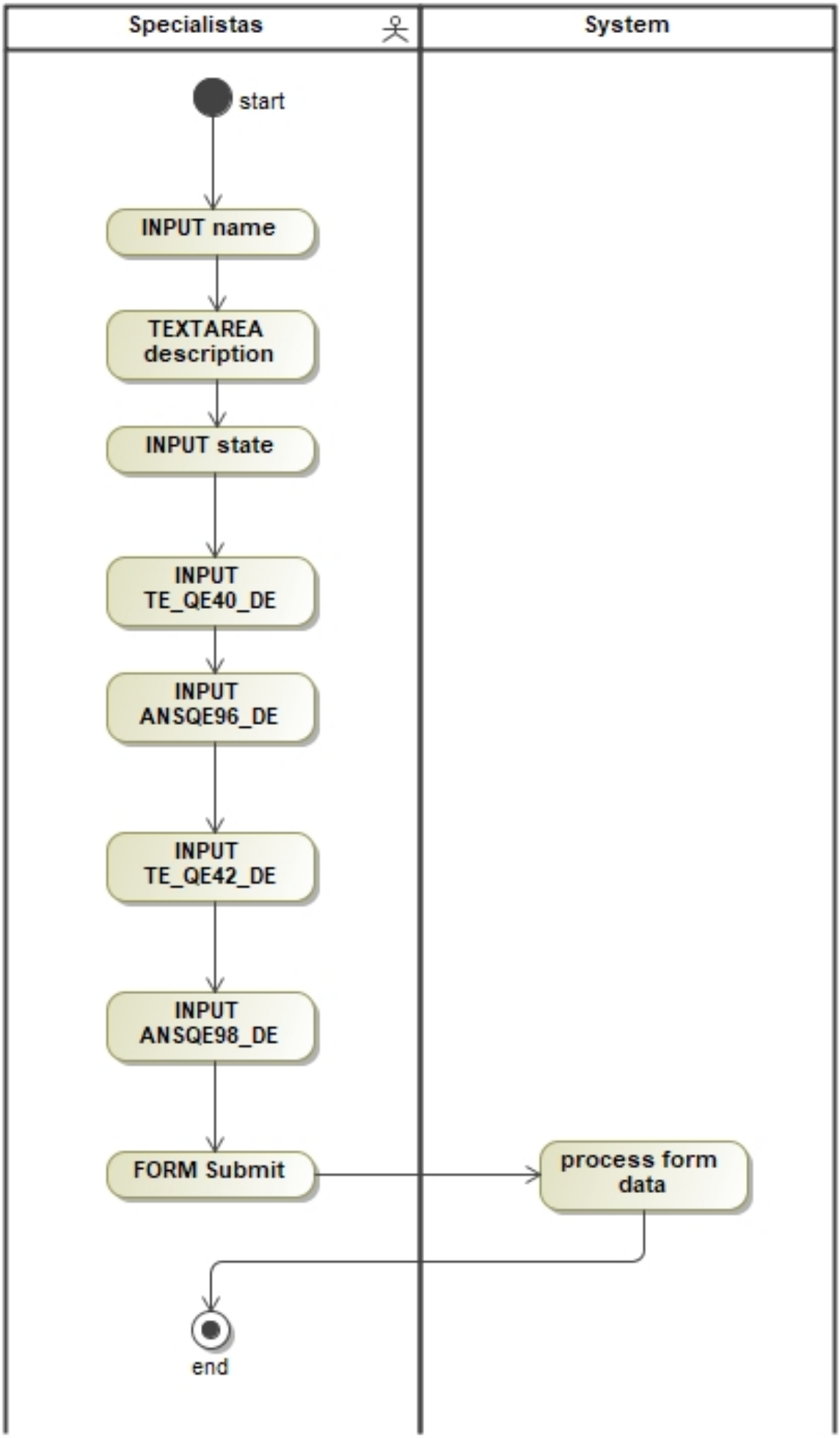

Vaikas
Tėvas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
: Peržiūrėti vaiko žaidimo informaciją
Paspausti mygtuką "Žaisti"
Pateikti žaidžiamo žaidimo puslapį
Parodyti neteisingo atsakymo pranešimą
Atsakyti į klausimą
Ar atsakymas teisingas?
[Ne]
[Taip]
Parodyti teisingo atsakymo pranešimą
Ar visi klausimai atsakyti?
[Ne]
Paspausti mygtuką "Kitas klausimas"
[Taip]
Paspausti mygtuką "Baigti"
Skaičiuoti ir išsaugoti rezultatą
Pateikti rezultatą žaidimo puslapyje

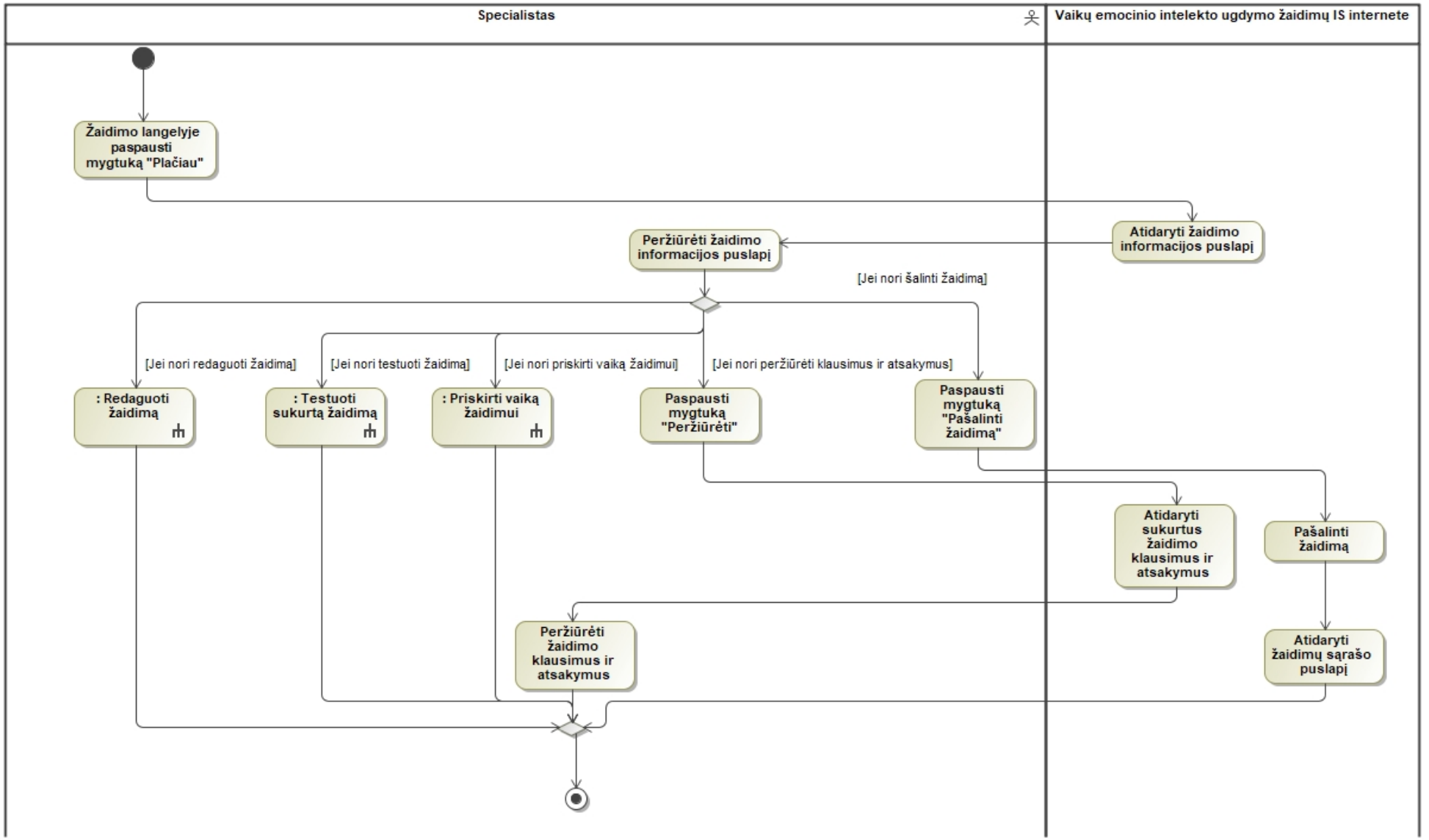

Tėvas
System
start
: Peržiūrėti vaikų sąrašą
A open /childrens/22/childrenGames
navigate url: /childrens/22/childrenGames
FORM Submit
process form data
FORM Submit
process form data
FORM Submit
process form data
end

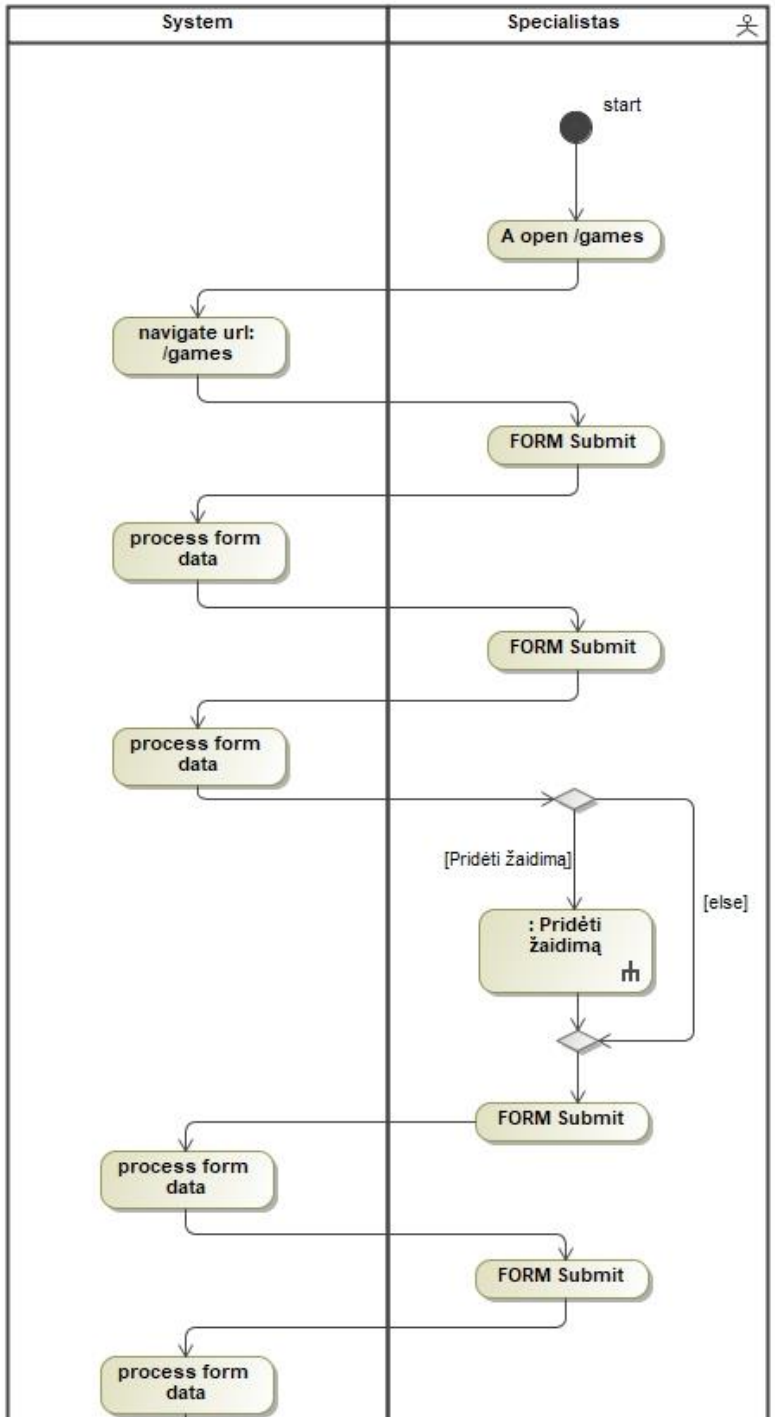

Specialistas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Atidaryti žaidimų sąrašo puslapį
Pateikti žaidimų sąrąšą
Peržiūrėti žaidimų sąrašą
[Jei nori peržiūrėti žaidimo informaciją]
[Jei nori kurti naują žaidimą]
: Peržiūrėti žaidimo informaciją
: Kurti žaidimą

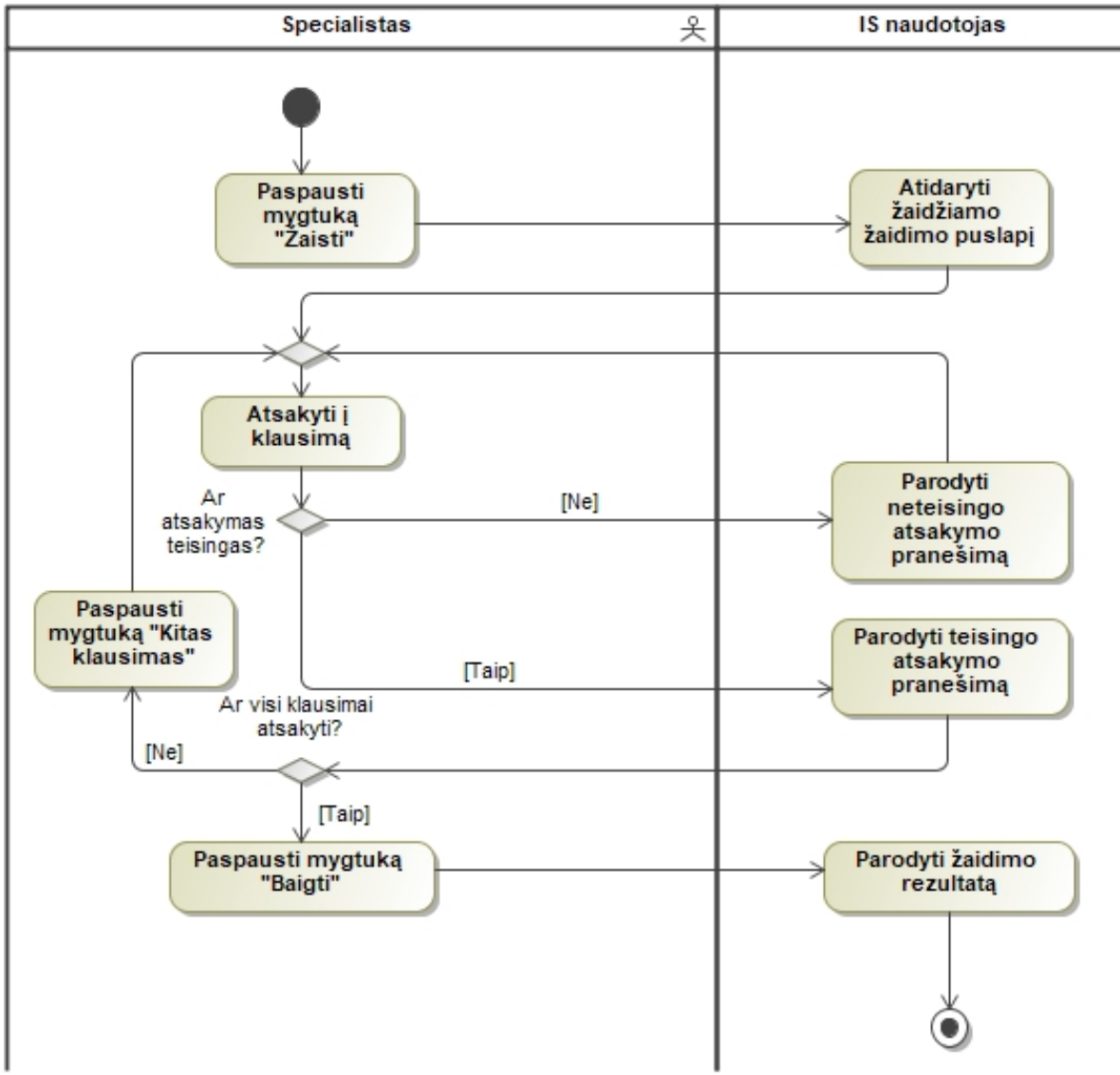

Specialistas
System
start
A open /games
navigate url: /games
A open /otherGames
navigate url: /otherGames
FORM Submit
process form data
[else]
[Redaguoti žaidimą]
: Redaguoti žaidimą
[else]
A open undefined
navigate url: undefined
[else]
[Bandyti žaidimą]
: Testuoti žaidimą
end

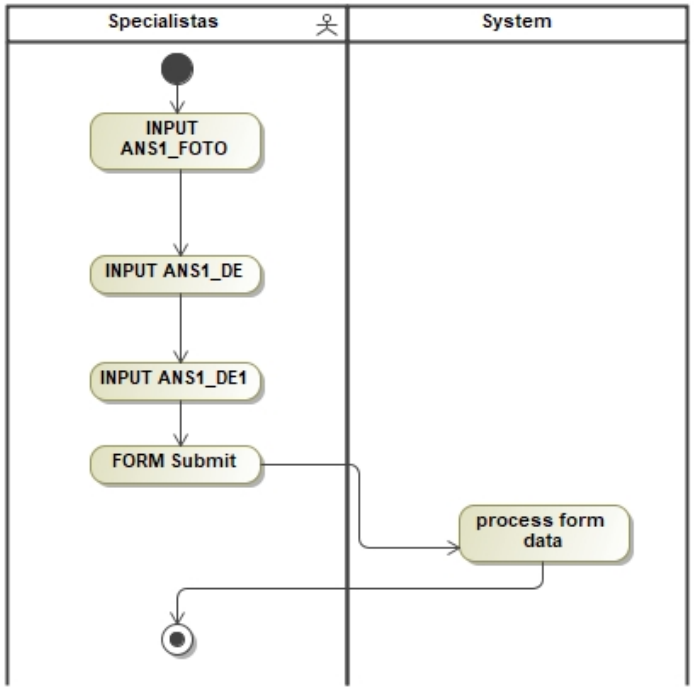

Specialistas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Paspausti mygtuką "Kurti naują žaidimą"
Pateikti žaidimo kūrimo puslapį
Įvesti žaidimo duomenis
Pasirinkti žaidimo būseną
Pasirinkti šabloną
Parodyti pasirinkto šablono klausimų kūrimo laukus
Įvesti klausimo tekstą
Įkelti paveiksliuką
Įkelti atsakymo paveiksliuką
Įvesti paveiksliuko aprašymą
Ar pridėti visi atsakymai?
[Ne]
Paspausti "+" klausimo eilutėje
Parodyti naujo atsakymo įvedimo laukus
[Taip]
Pasirinkti teisingą atsakymo variantą
Ar visi klausimai sukurti?
[Ne]
Pasapusti mygtuką "Pridėti klausimą"
Parodyti naujo klausimo įvedimo laukus
[Taip]
Paspausti mygtuką "Išsaugoti"
Išsaugoti žaidimą
Atidaryti žaidimų sąrašo puslapį

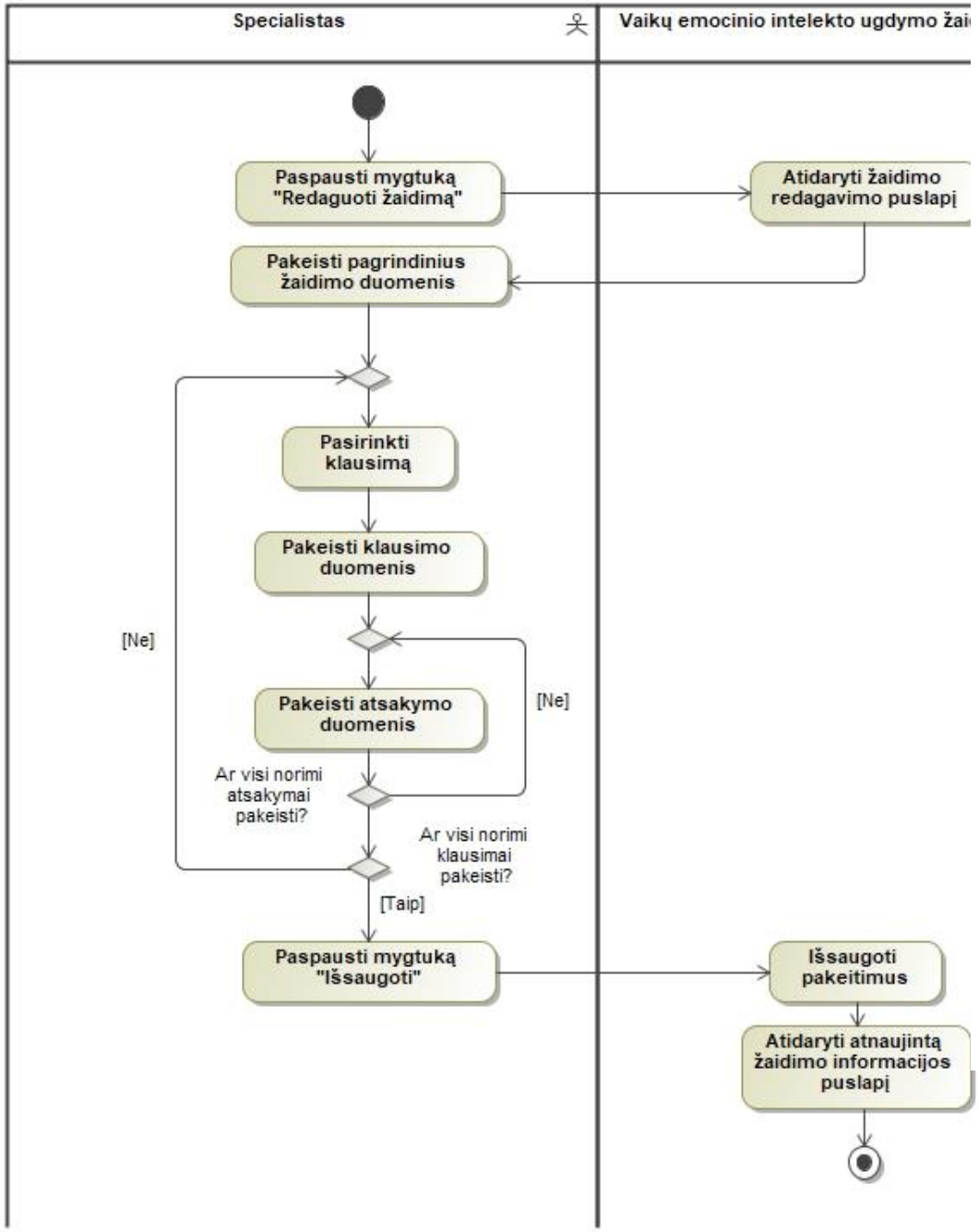

Specialistas
System
start
INPUT name
TEXTAREA description
INPUT state
INPUT TE_QE40_DE
INPUT ANSQE96_DE
INPUT TE_QE42_DE
INPUT ANSQE98_DE
FORM Submit
process form data
end

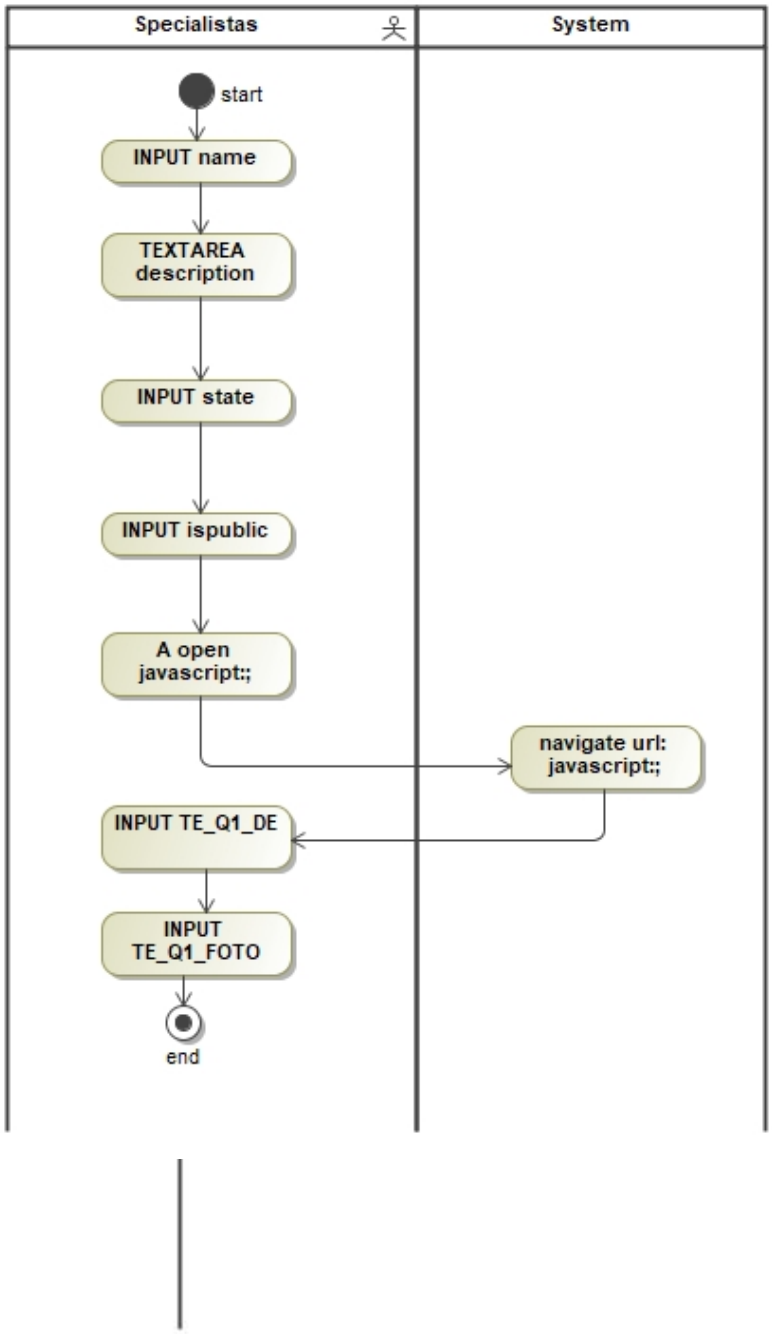

Specialistas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Žaidimo langelyje paspausti mygtuką "Plačiau"
Atidaryti žaidimo informacijos puslapį
Peržiūrėti žaidimo informacijos puslapį
[Jei nori šalinti žaidimą]
[Jei nori redaguoti žaidimą]
[Jei nori testuoti žaidimą]
[Jei nori priskirti vaiką žaidimui]
[Jei nori peržiūrėti klausimus ir atsakymus]
: Redaguoti žaidimą
: Testuoti sukurtą žaidimą
: Priskirti vaiką žaidimui
Paspausti mygtuką "Peržiūrėti"
Paspausti mygtuką "Pašalinti žaidimą"
Atidaryti sukurtus žaidimo klausimus ir atsakymus
Pašalinti žaidimą
Peržiūrėti žaidimo klausimus ir atsakymus
Atidaryti žaidimų sąrašo puslapį

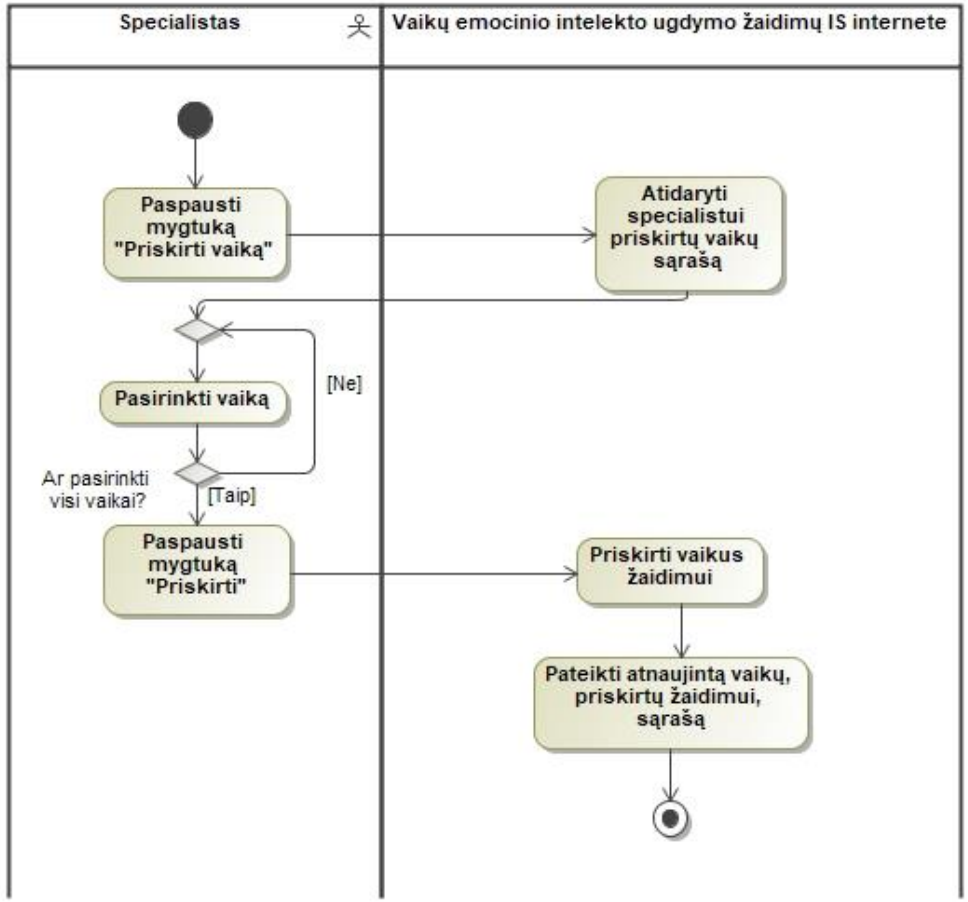

System
Specialistas
start
A open /games
navigate url: /games
FORM Submit
process form data
FORM Submit
process form data
[Pridėti žaidimą]
[else]
: Pridėti žaidimą
FORM Submit
process form data
FORM Submit
process form data

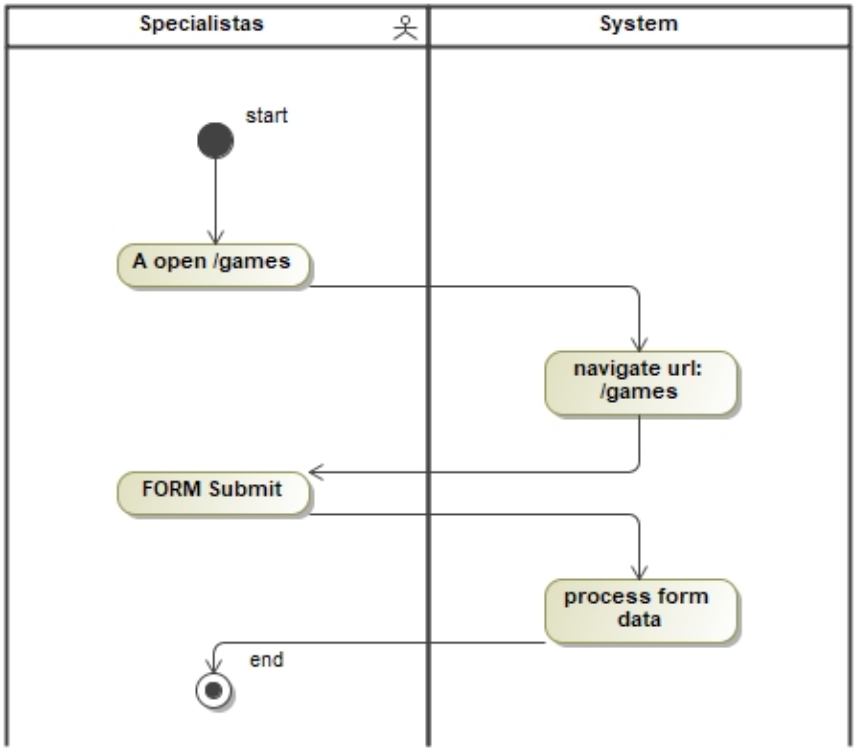

Specialistas
IS naudotojas
Paspausti mygtuką "Žaisti"
Atidaryti žaidžiamo žaidimo puslapį
Atsakyti į klausimą
Ar atsakymas teisingas?
[Ne]
Parodyti neteisingo atsakymo pranešimą
Paspausti mygtuką "Kitas klausimas"
[Taip]
Parodyti teisingo atsakymo pranešimą
Ar visi klausimai atsakyti?
[Ne]
[Taip]
Paspausti mygtuką "Baigti"
Parodyti žaidimo rezultatą

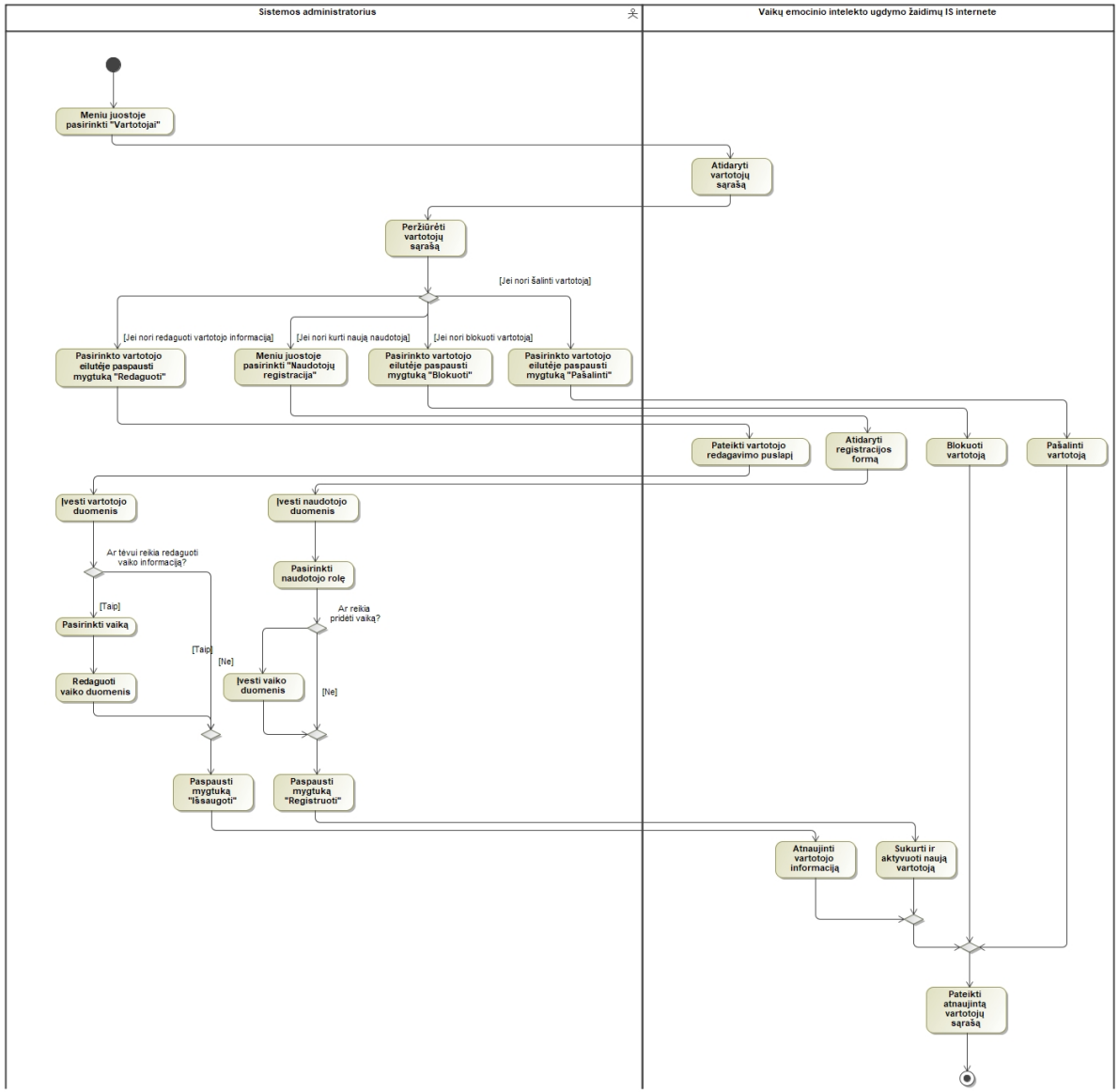

Specialistas
System
INPUT ANS1_FOTO
INPUT ANS1_DE
INPUT ANS1_DE1
FORM Submit
process form data


Specialistas
Vaikų emocinio intelekto ugdymo žai
Paspausti mygtuką "Redaguoti žaidimą"
Atidaryti žaidimo redagavimo puslapį
Pakeisti pagrindinius žaidimo duomenis
Pasirinkti klausimą
Pakeisti klausimo duomenis
[Ne]
Pakeisti atsakymo duomenis
[Ne]
Ar visi norimi atsakymai pakeisti?
Ar visi norimi klausimai pakeisti?
[Taip]
Paspausti mygtuką "Išsaugoti"
Išsaugoti pakeitimus
Atidaryti atnaujintą žaidimo informacijos puslapį

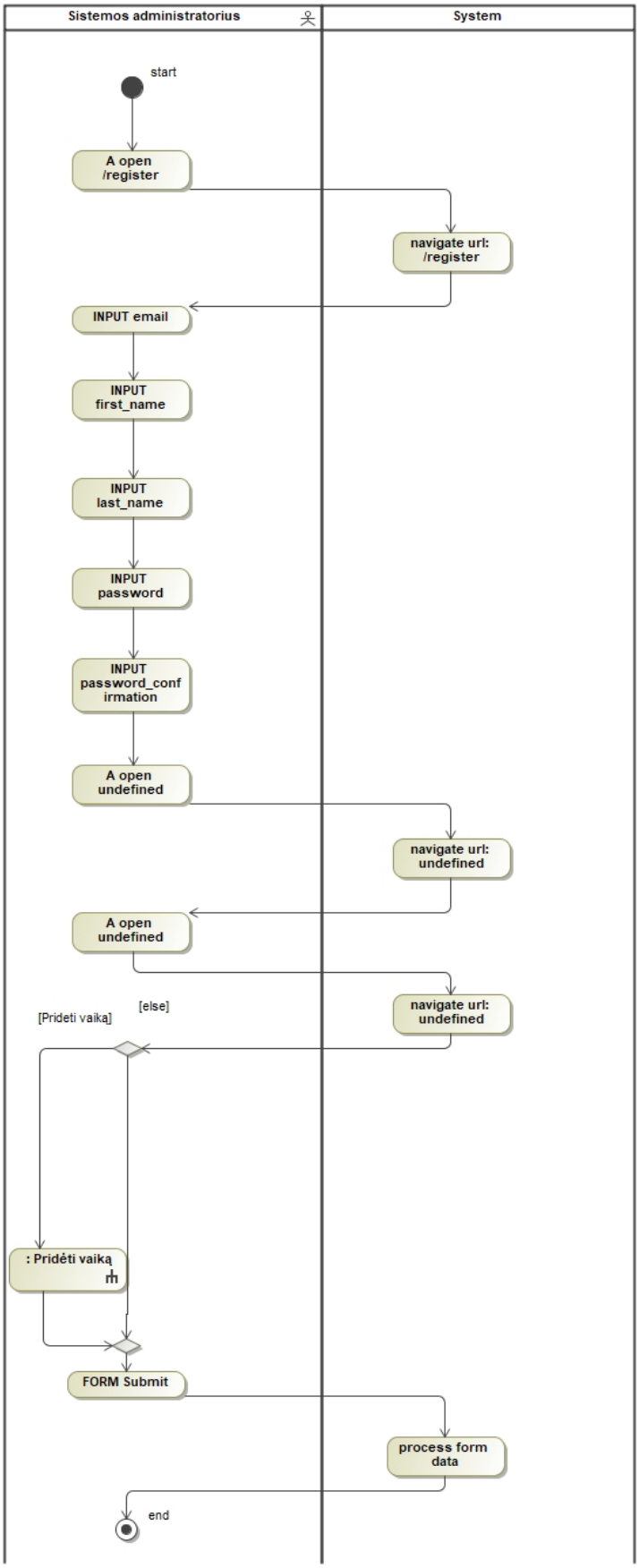

Specialistas
System
start
INPUT name
TEXTAREA description
INPUT state
INPUT ispublic
A open javascript:;
navigate url: javascript:;
INPUT TE_Q1_DE
INPUT TE_Q1_FOTO
end

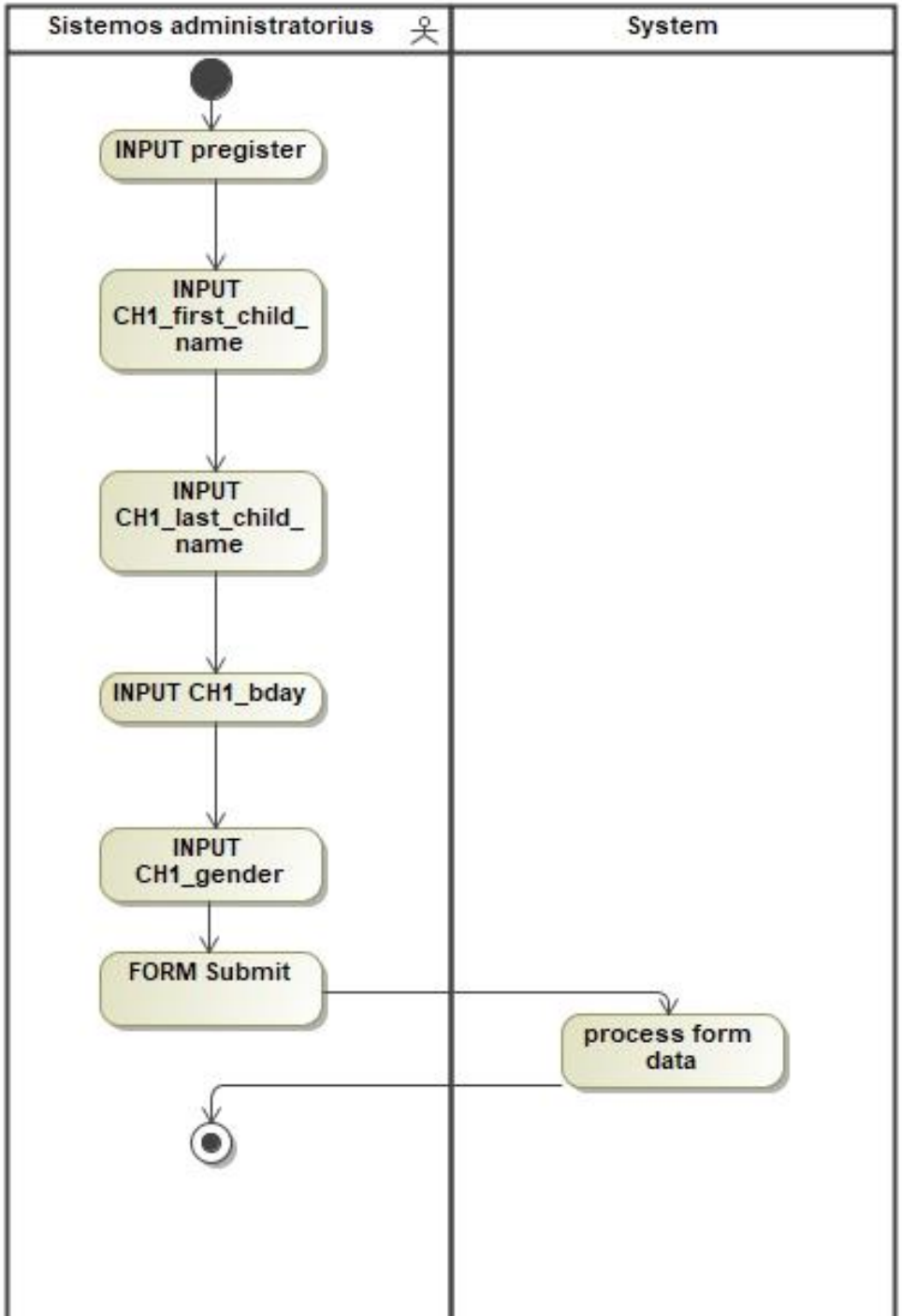

Specialistas
Vaikų emocinio intelekto ugdymo žaidimų IS internete
Paspausti mygtuką "Priskirti vaiką"
Atidaryti specialistui priskirtų vaikų sąrašą
Pasirinkti vaiką
[Ne]
Ar pasirinkti visi vaikai?
[Taip]
Paspausti mygtuką "Priskirti"
Priskirti vaikus žaidimui
Pateikti atnaujintą vaikų, priskirtų žaidimui, sąrašą

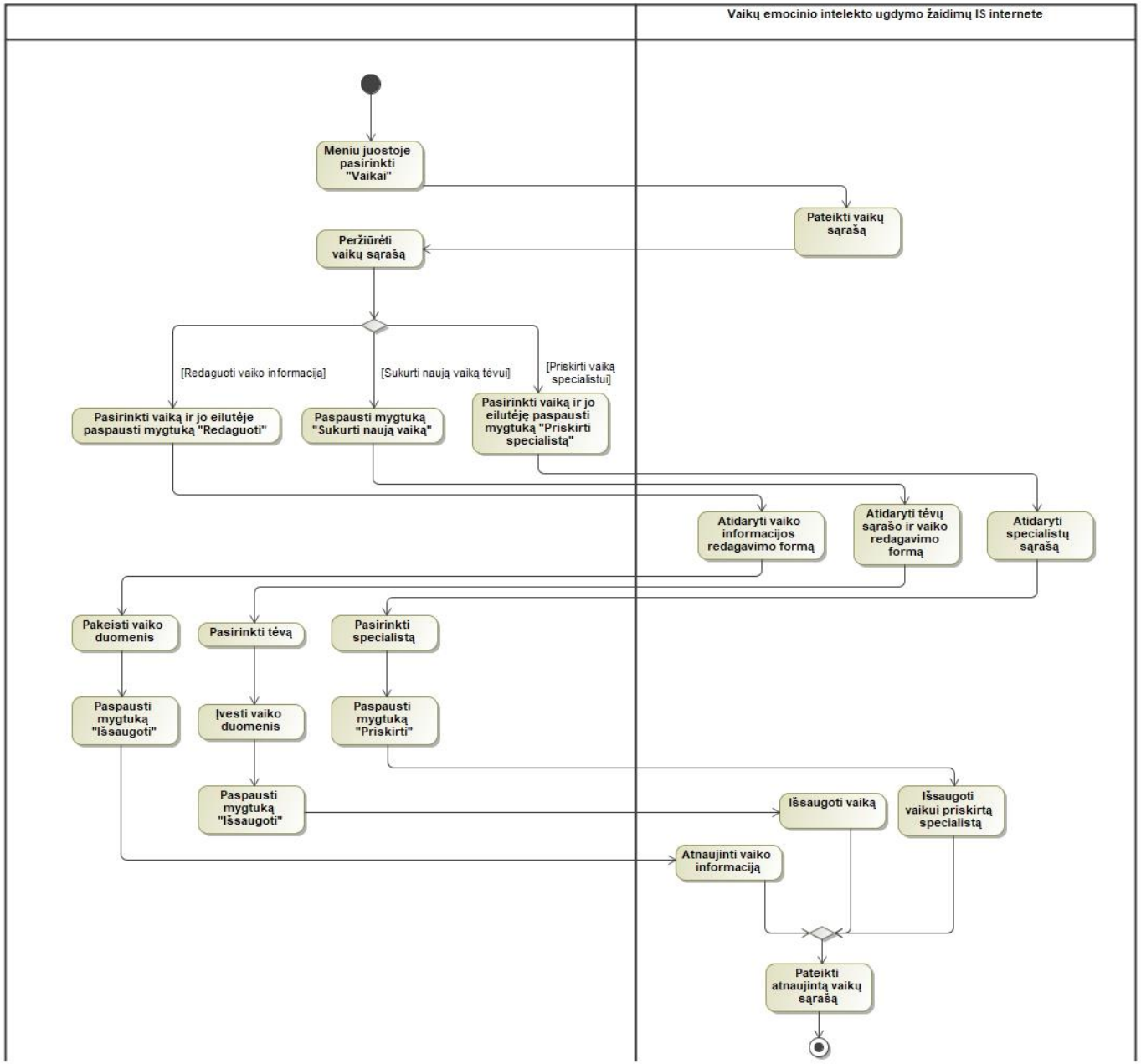

Specialistas
System
start
A open /games
navigate url: /games
FORM Submit
process form data
end

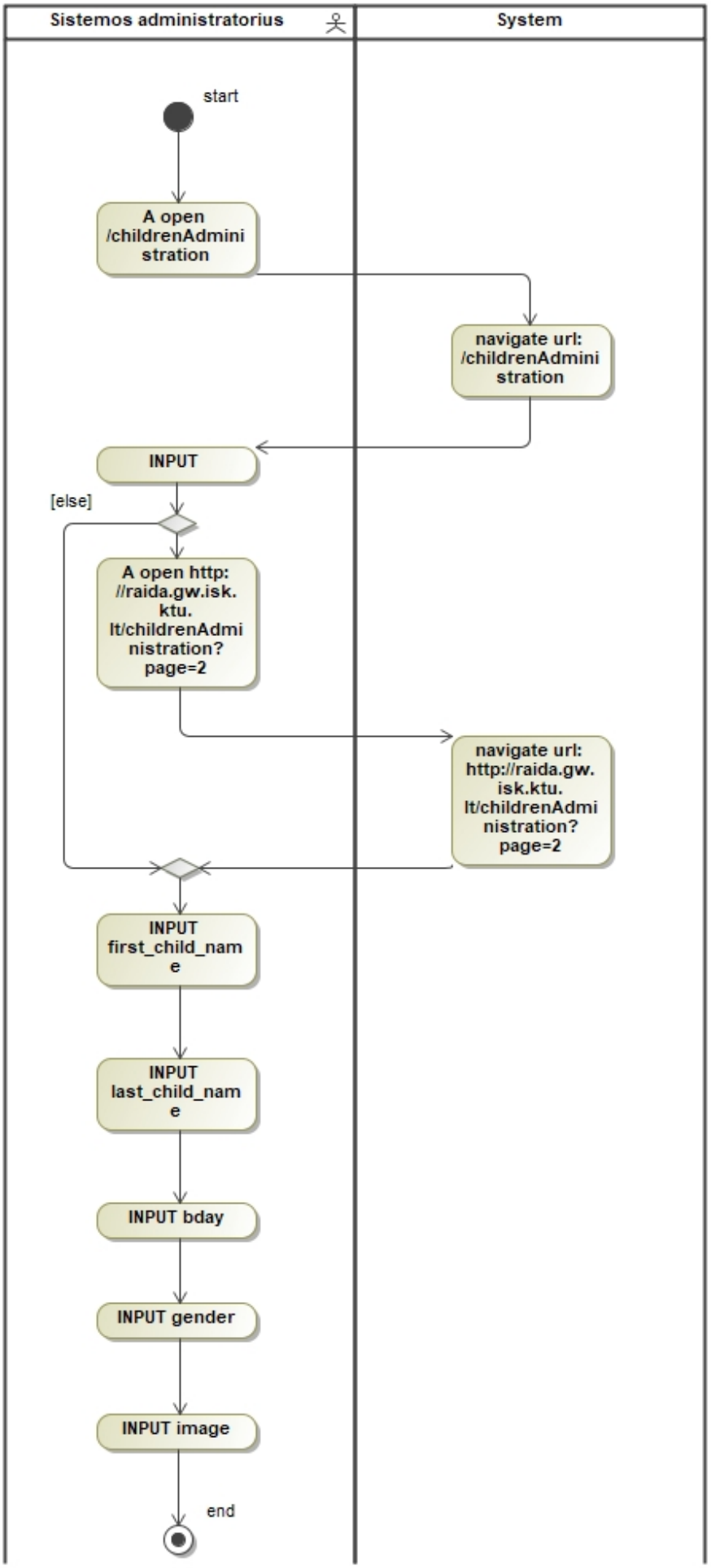



**Ši diagrama aprašoma dvejomis sugeneruotomis diagramomis**


Sistemos administratorius
System
start
A open /register
navigate url: /register
INPUT email
INPUT first_name
INPUT last_name
INPUT password
INPUT password_confirmation
A open undefined
navigate url: undefined
A open undefined
navigate url: undefined
[Prideti vaika]
[else]
: Pridėti vaiką
FORM Submit
process form data
end

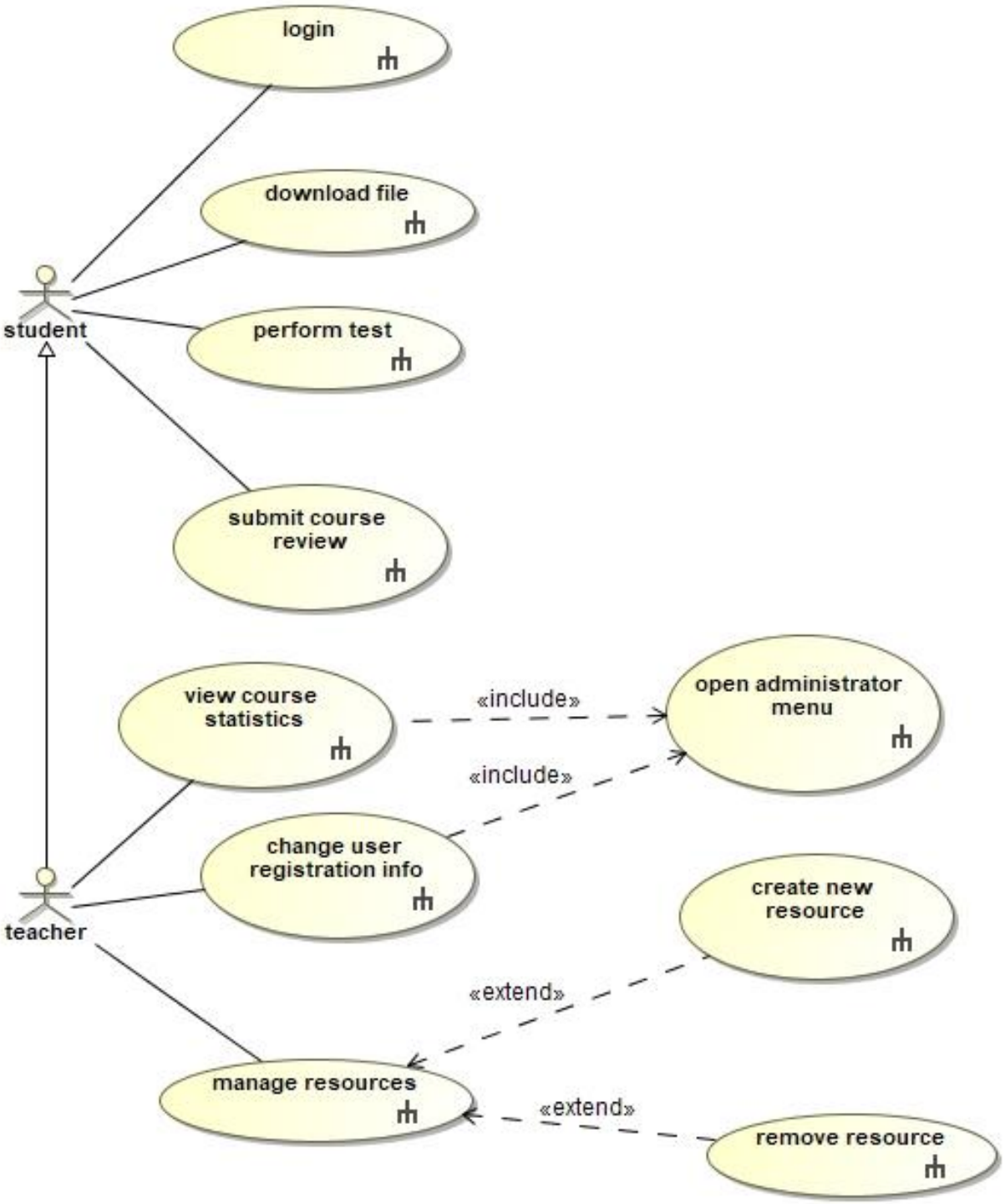

Sistemos administratorius
System
INPUT pregister
INPUT CH1_first_child_ name
INPUT CH1_last_child_ name
INPUT CH1_bday
INPUT CH1_gender
FORM Submit
process form data

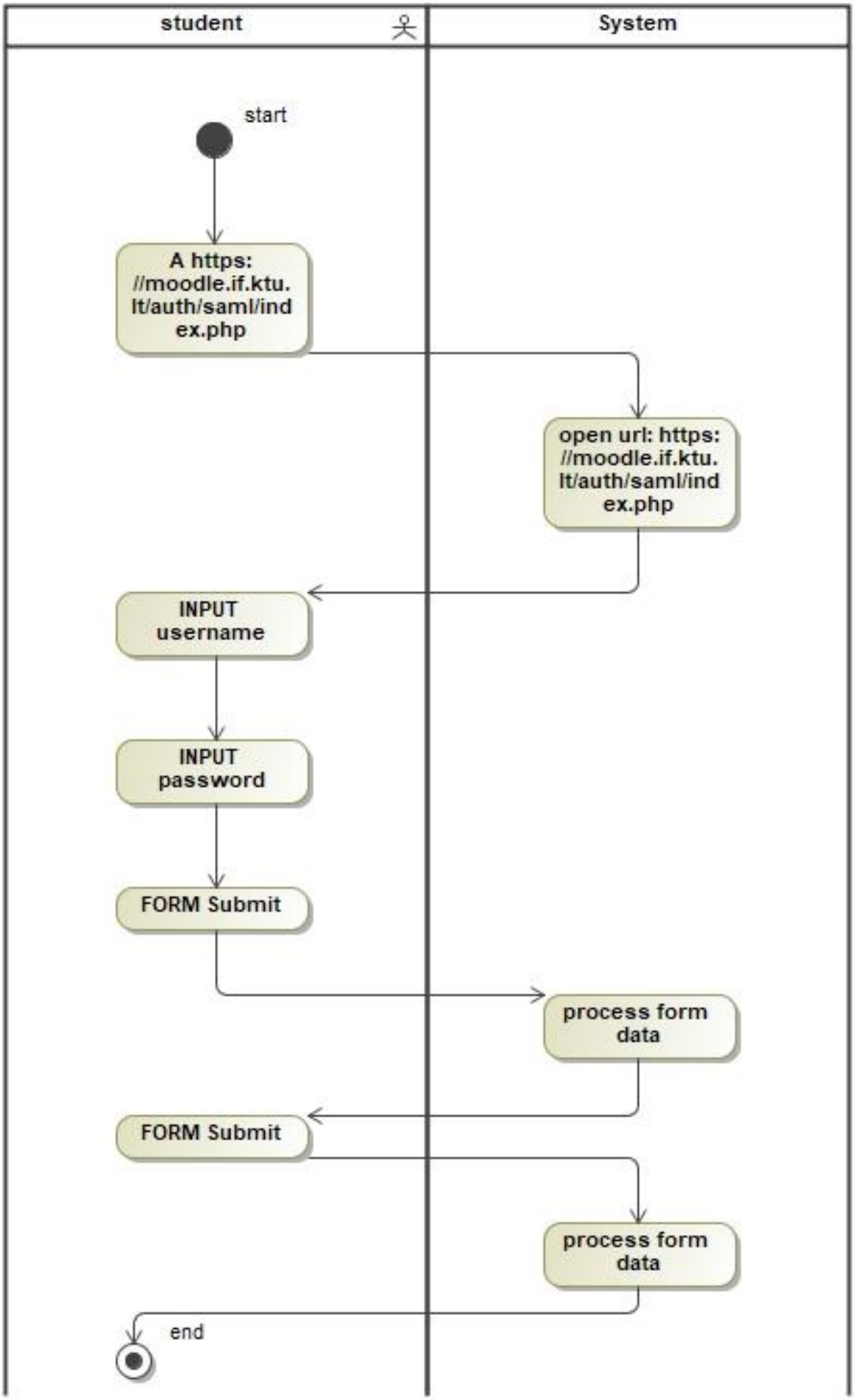

Vaikų emocinio intelekto ugdymo žaidimų IS internete
Meniu juostoje pasirinkti "Vaikai"
Pateikti vaikų sąrašą
Peržiūrėti vaikų sąrašą
[Redaguoti vaiko informaciją]
[Sukurti naują vaiką tėvui]
[Priskirti vaiką specialistui]
Pasirinkti vaiką ir jo eilutėje paspausti mygtuką "Redaguoti"
Paspausti mygtuką "Sukurti naują vaiką"
Pasirinkti vaiką ir jo eilutėje paspausti mygtuką "Priskirti specialistą"
Atidaryti vaiko informacijos redagavimo formą
Atidaryti tėvų sąrašo ir vaiko redagavimo formą
Atidaryti specialistų sąrašą
Pakeisti vaiko duomenis
Pasirinkti tėvą
Pasirinkti specialistą
Paspausti mygtuką "Išsaugoti"
Įvesti vaiko duomenis
Paspausti mygtuką "Priskirti"
Paspausti mygtuką "Išsaugoti"
Išsaugoti vaiką
Išsaugoti vaikui priskirtą specialistą
Atnaujinti vaiko informaciją
Pateikti atnaujintą vaikų sąrašą

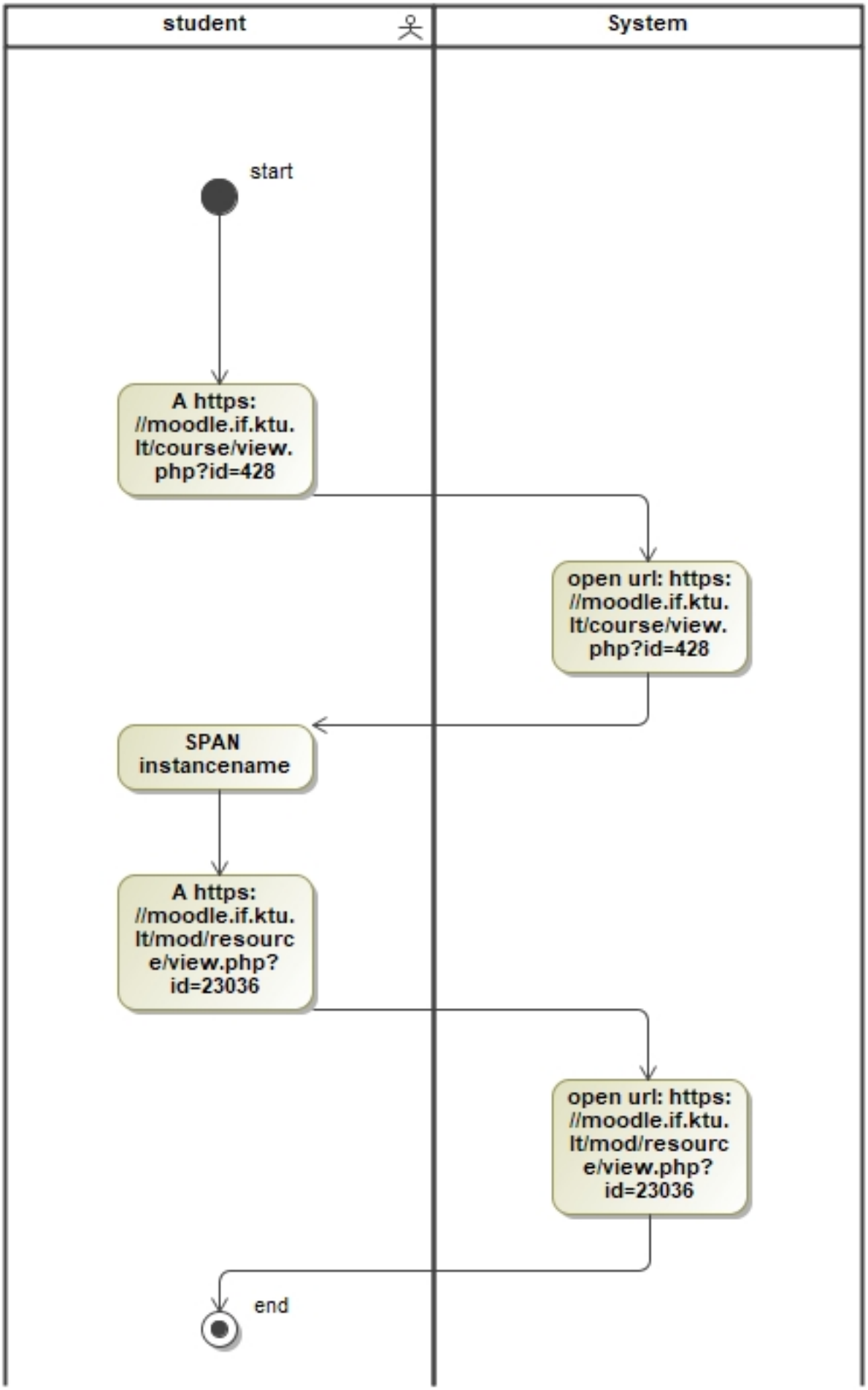

Sistemos administratorius
System
start
A open /childrenAdministration
navigate url: /childrenAdministration
INPUT
[else]
A open http://raida.gw.isk.ktu.lt/childrenAdministration?page=2
navigate url: http://raida.gw.isk.ktu.lt/childrenAdministration?page=2
INPUT first_child_name
INPUT last_child_name
INPUT bday
INPUT gender
INPUT image
end

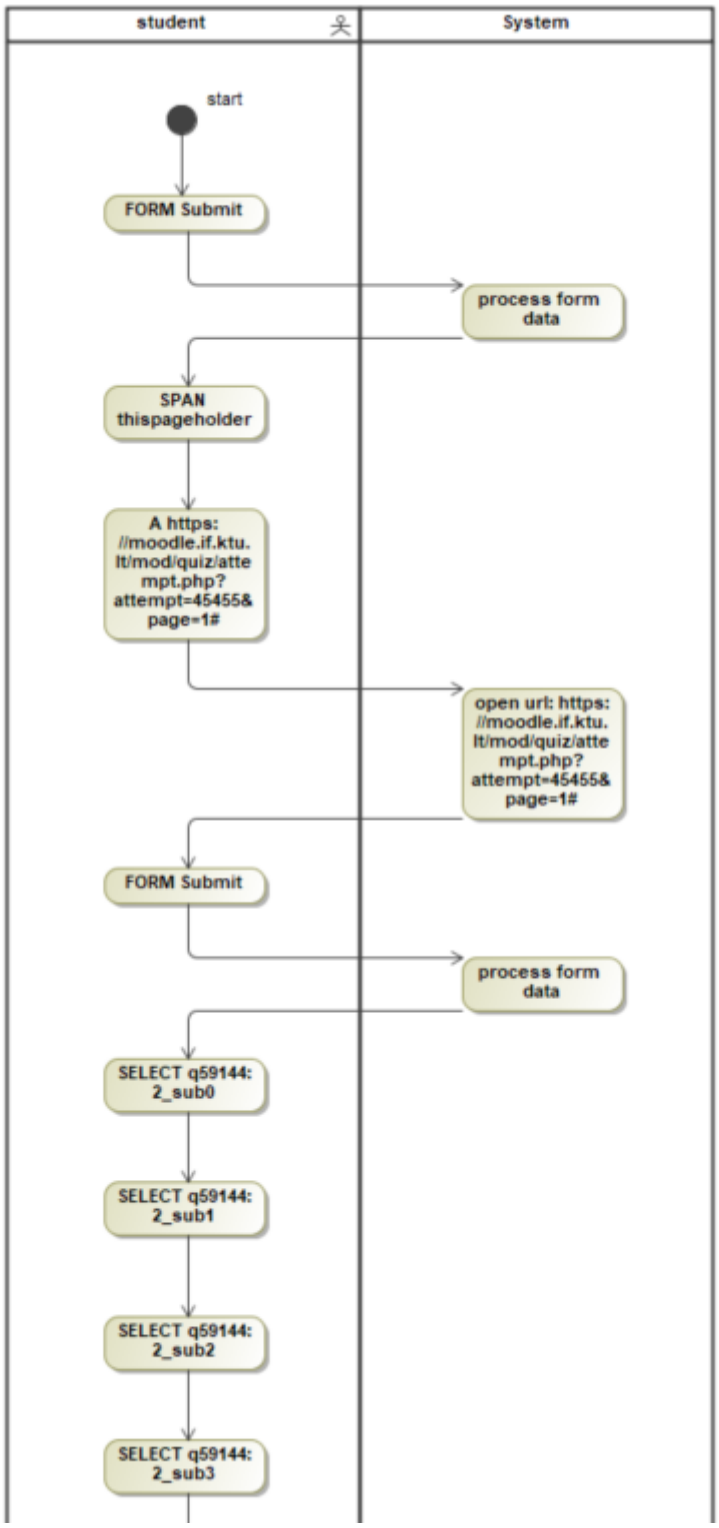
### 9.1.3. priedas. *moodle.if.ktu.lt* scenarijui sugeneruotas panaudojimo atvejų modelis.

Šiame skyriuje pateikiamos panaudojimo atvejų ir veiklos diagramos, kurios buvo vertintos ekspertų.

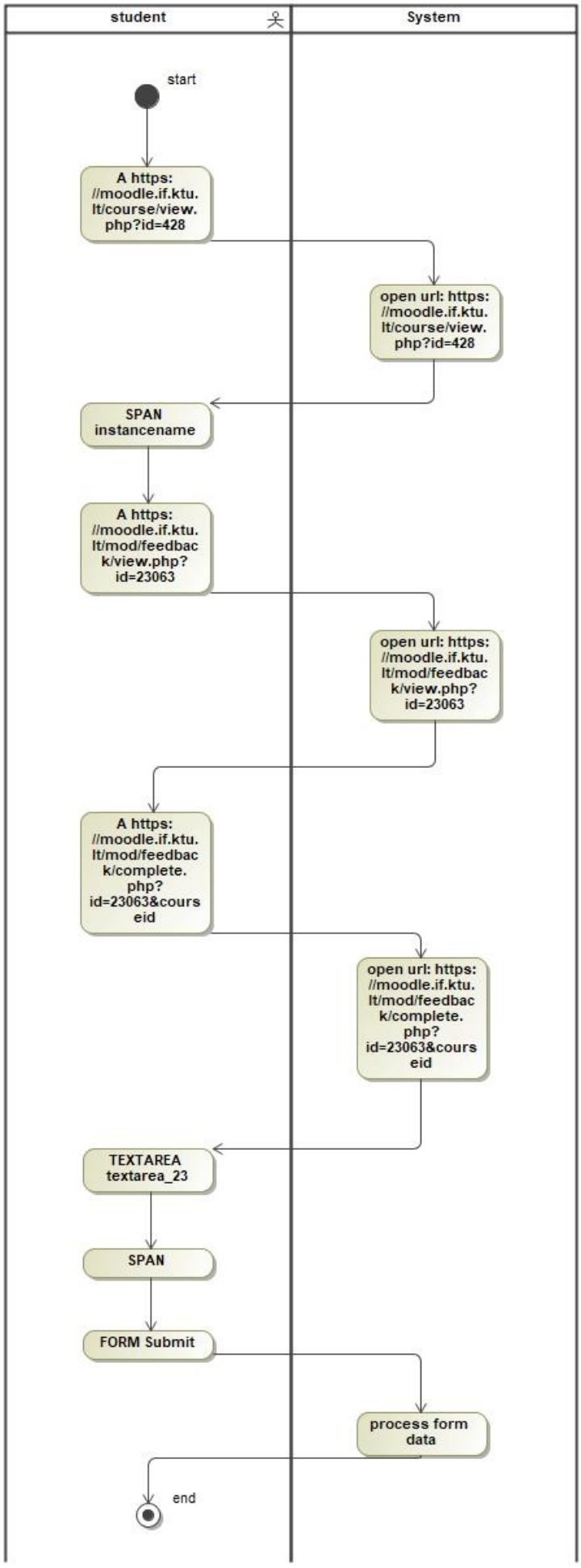


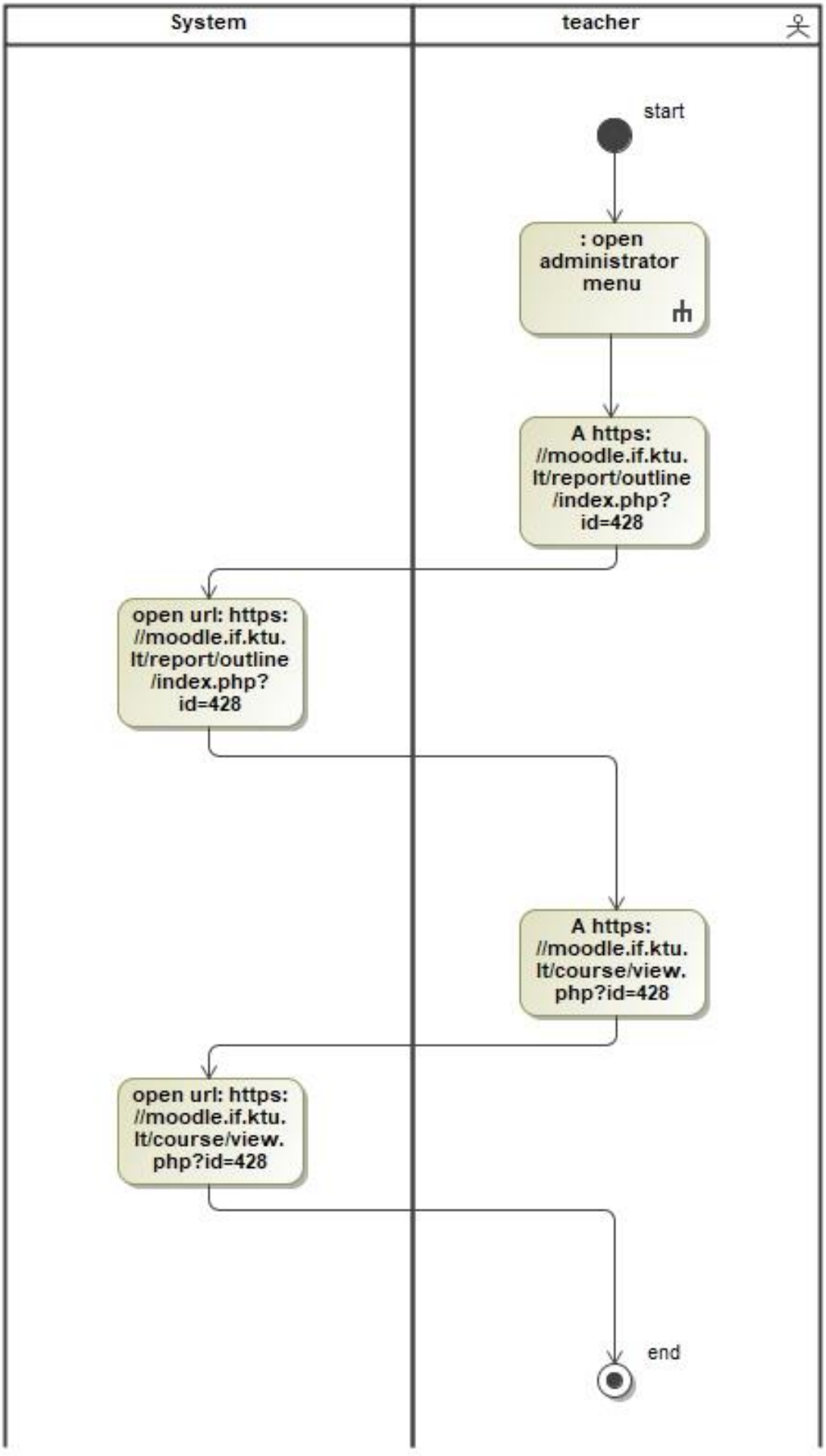

student
System
start
A https: //moodle.if.ktu. lt/auth/saml/ind ex.php
open url: https: //moodle.if.ktu. lt/auth/saml/ind ex.php
INPUT username
INPUT password
FORM Submit
process form data
FORM Submit
process form data
end

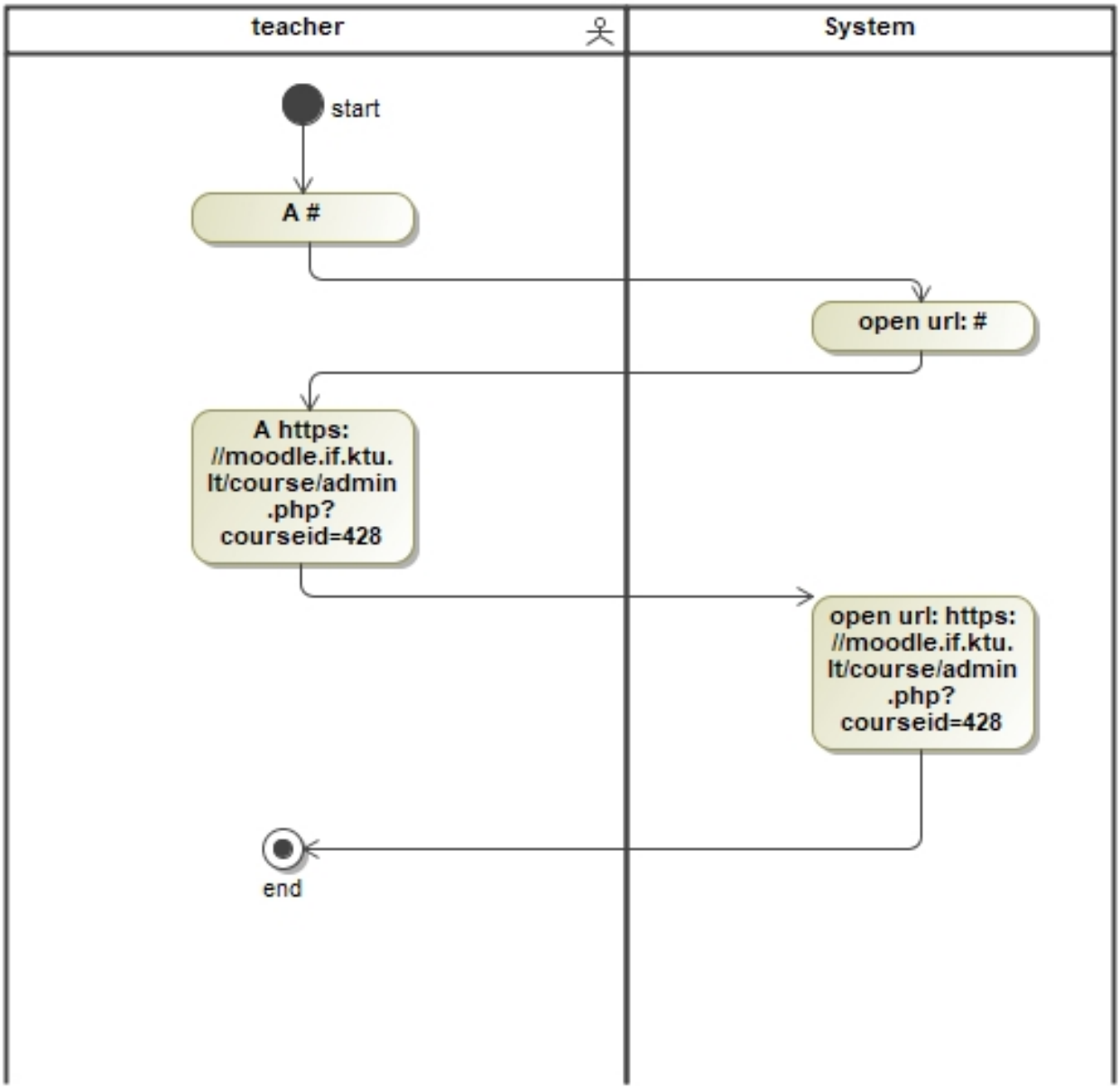

student
System
start
A https: //moodle.if.ktu. lt/course/view. php?id=428
open url: https: //moodle.if.ktu. lt/course/view. php?id=428
SPAN instancename
A https: //moodle.if.ktu. lt/mod/resourc e/view.php? id=23036
open url: https: //moodle.if.ktu. lt/mod/resourc e/view.php? id=23036
end

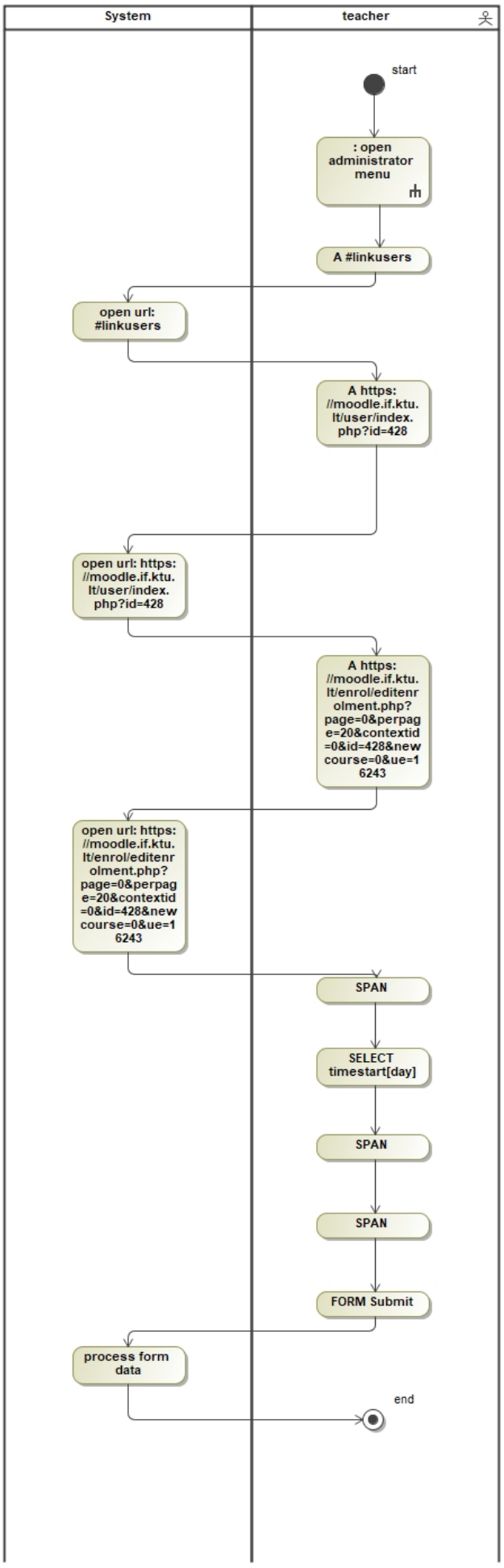


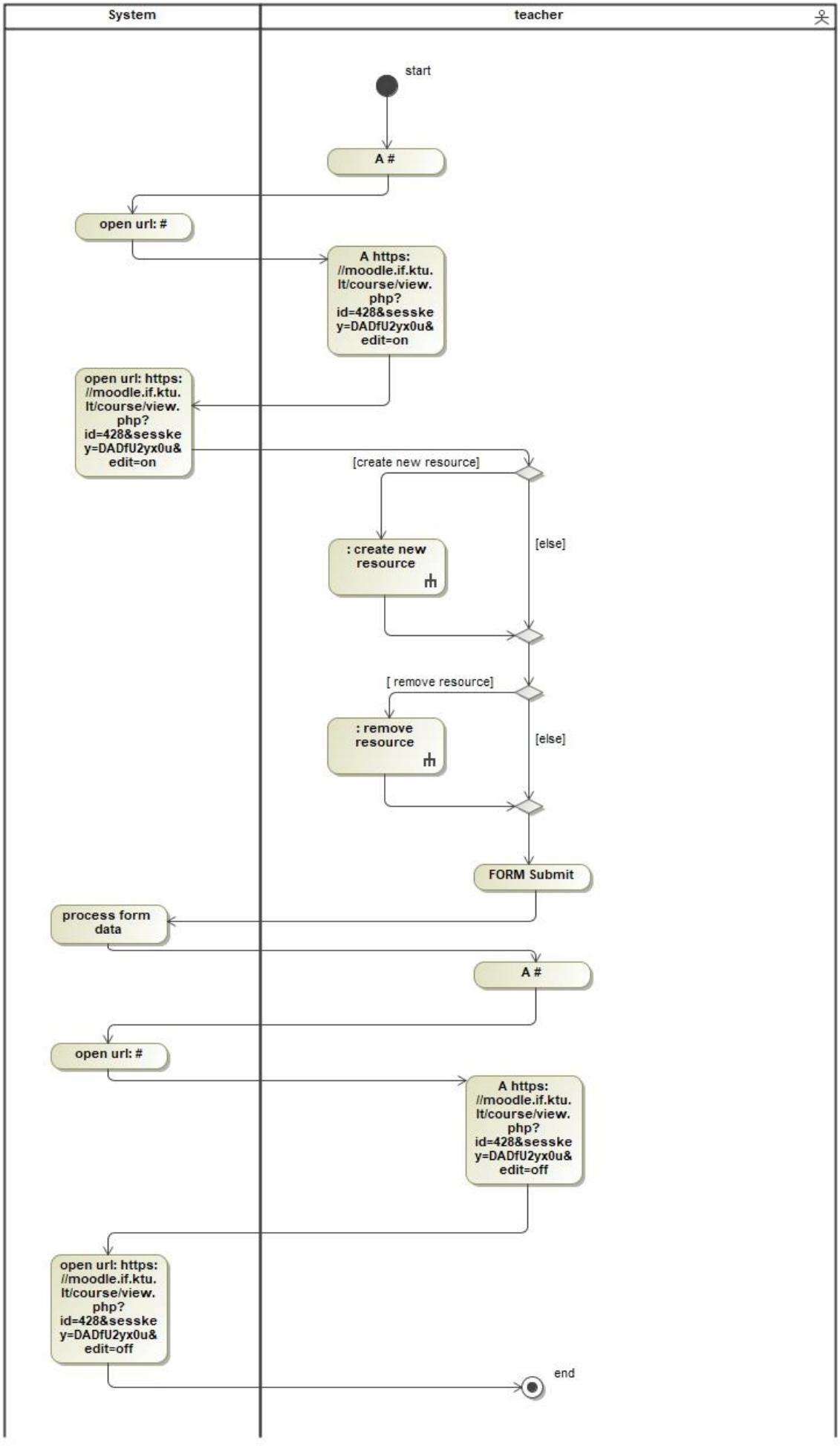

Dėl diagramos dydžio pateikiamas tik jos fragmentas.

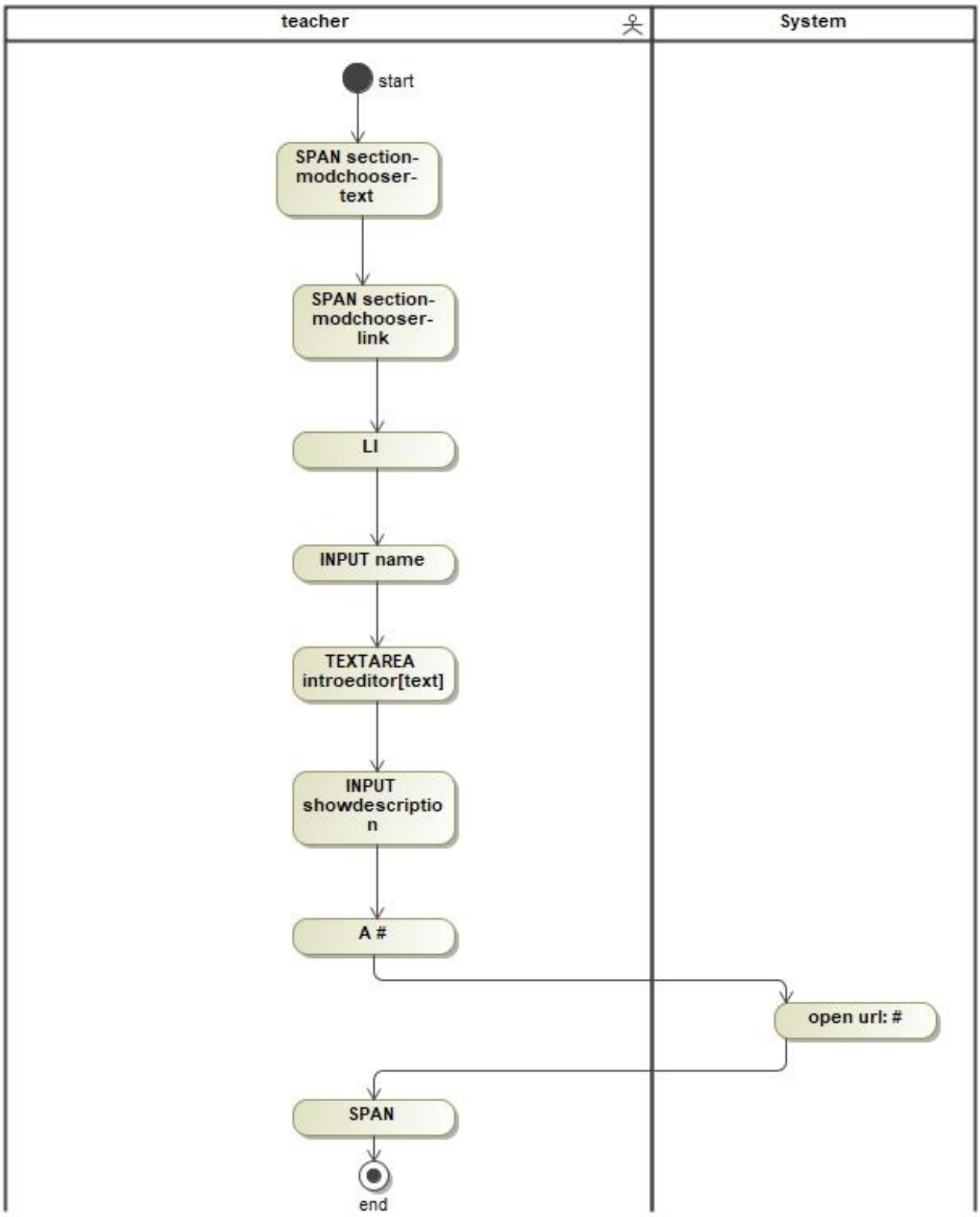

student
System
start
A https: //moodle.if.ktu. lt/course/view. php?id=428
open url: https: //moodle.if.ktu. lt/course/view. php?id=428
SPAN instancename
A https: //moodle.if.ktu. lt/mod/feedbac k/view.php? id=23063
open url: https: //moodle.if.ktu. lt/mod/feedbac k/view.php? id=23063
A https: //moodle.if.ktu. lt/mod/feedbac k/complete. php? id=23063&cours eid
open url: https: //moodle.if.ktu. lt/mod/feedbac k/complete. php? id=23063&cours eid
TEXTAREA textarea_23
SPAN
FORM Submit
process form data
end

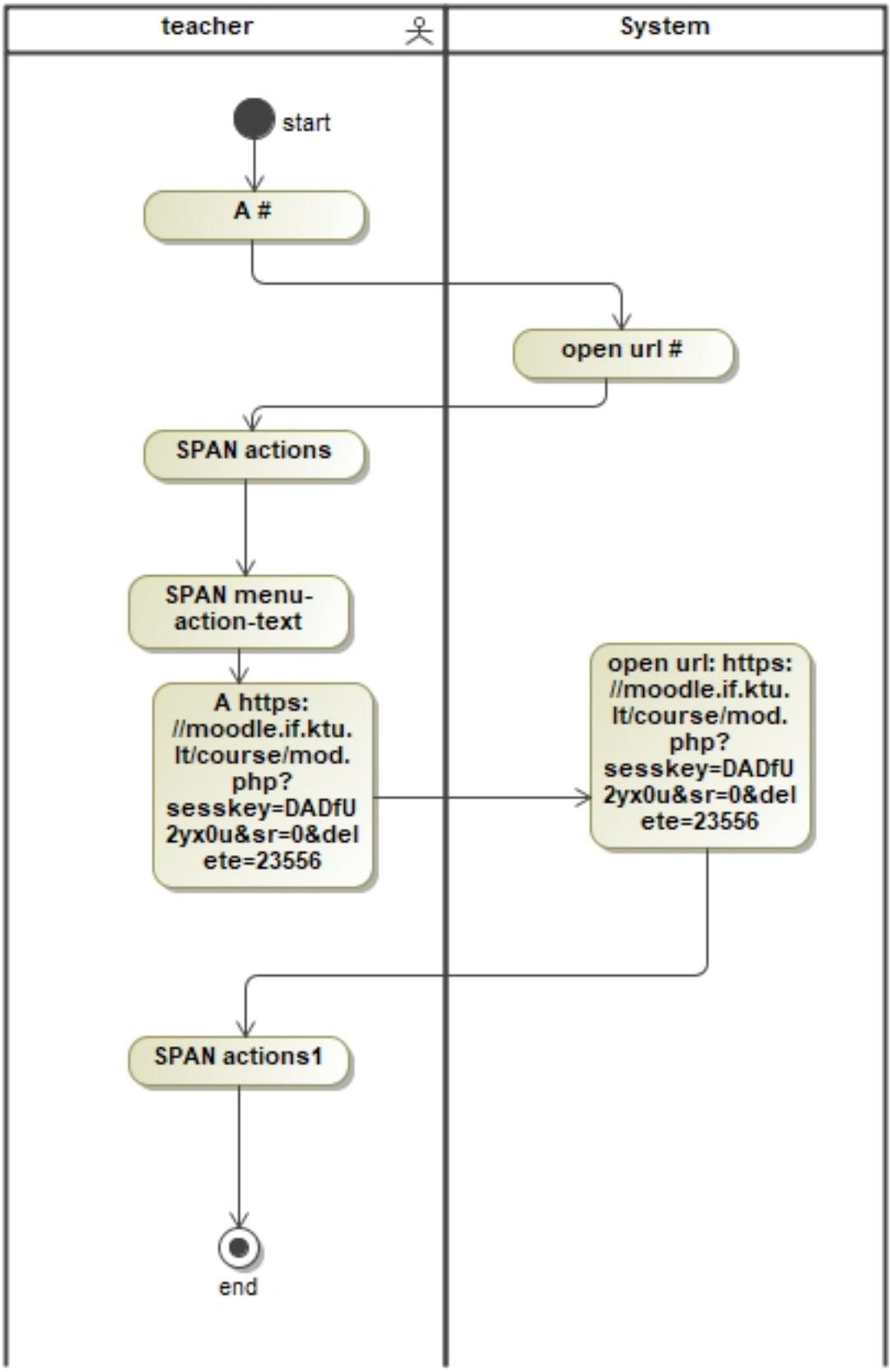

System
teacher
start
: open administrator menu
A https: //moodle.if.ktu. lt/report/outline /index.php? id=428
open url: https: //moodle.if.ktu. lt/report/outline /index.php? id=428
A https: //moodle.if.ktu. lt/course/view. php?id=428
open url: https: //moodle.if.ktu. lt/course/view. php?id=428
end


teacher
System
start
A #
open url: #
A https: //moodle.if.ktu. lt/course/admin .php? courseid=428
open url: https: //moodle.if.ktu. lt/course/admin .php? courseid=428
end


System
teacher
start
: open administrator menu
A #linkusers
open url: #linkusers
A https: //moodle.if.ktu.lt/user/index.php?id=428
open url: https: //moodle.if.ktu.lt/user/index.php?id=428
A https: //moodle.if.ktu.lt/enrol/editenrolment.php?page=0&perpage=20&contextid=0&id=428&newcourse=0&ue=16243
open url: https: //moodle.if.ktu.lt/enrol/editenrolment.php?page=0&perpage=20&contextid=0&id=428&newcourse=0&ue=16243
SPAN
SELECT timestart[day]
SPAN
SPAN
FORM Submit
process form data
end

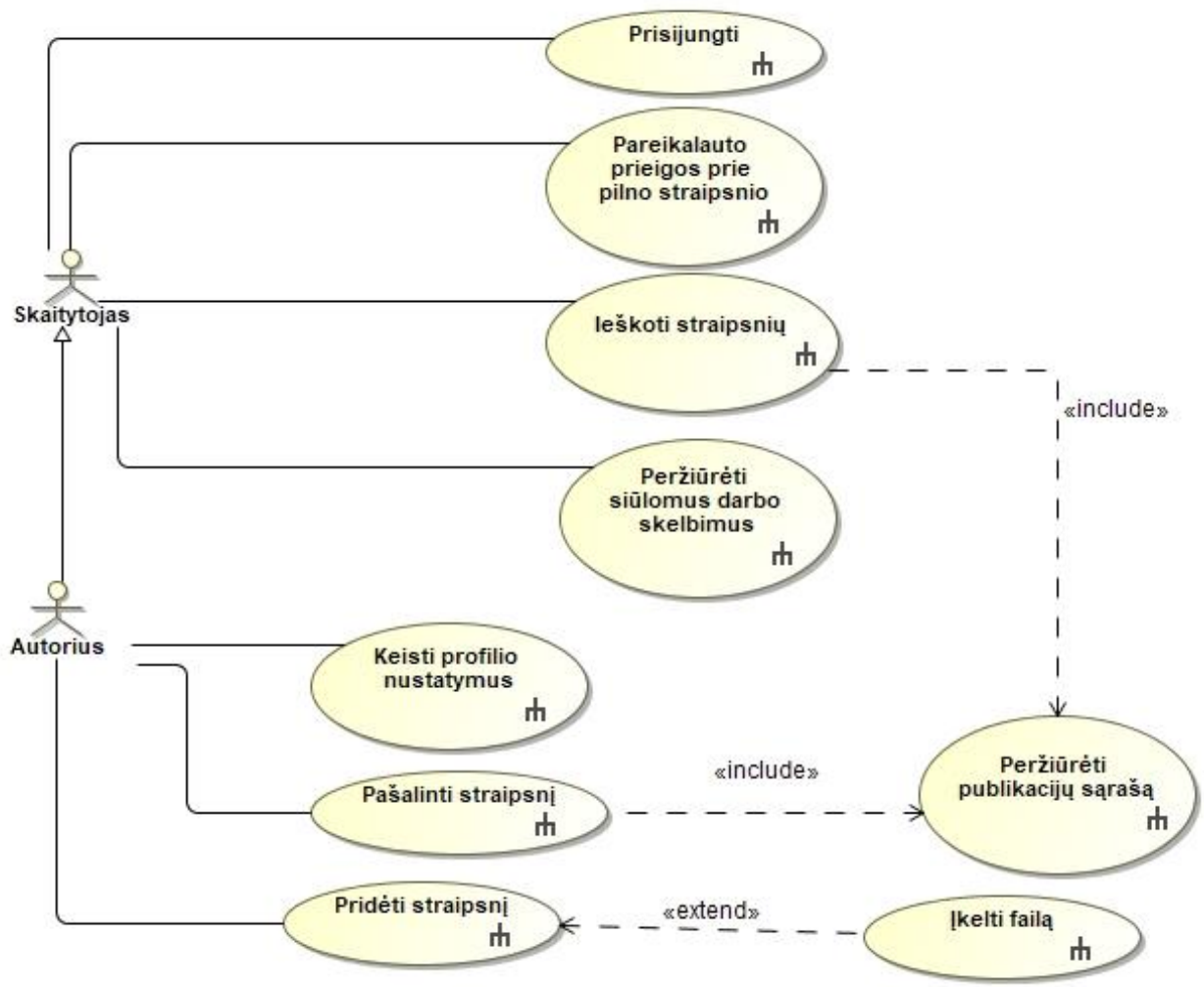

System
teacher
start
A #
open url: #
A https: //moodle.if.ktu. lt/course/view. php? id=428&sesske y=DADfU2yx0u& edit=on
open url: https: //moodle.if.ktu. lt/course/view. php? id=428&sesske y=DADfU2yx0u& edit=on
[create new resource]
[else]
: create new resource
[ remove resource]
: remove resource
[else]
FORM Submit
process form data
A #
open url: #
A https: //moodle.if.ktu. lt/course/view. php? id=428&sesske y=DADfU2yx0u& edit=off
open url: https: //moodle.if.ktu. lt/course/view. php? id=428&sesske y=DADfU2yx0u& edit=off
end

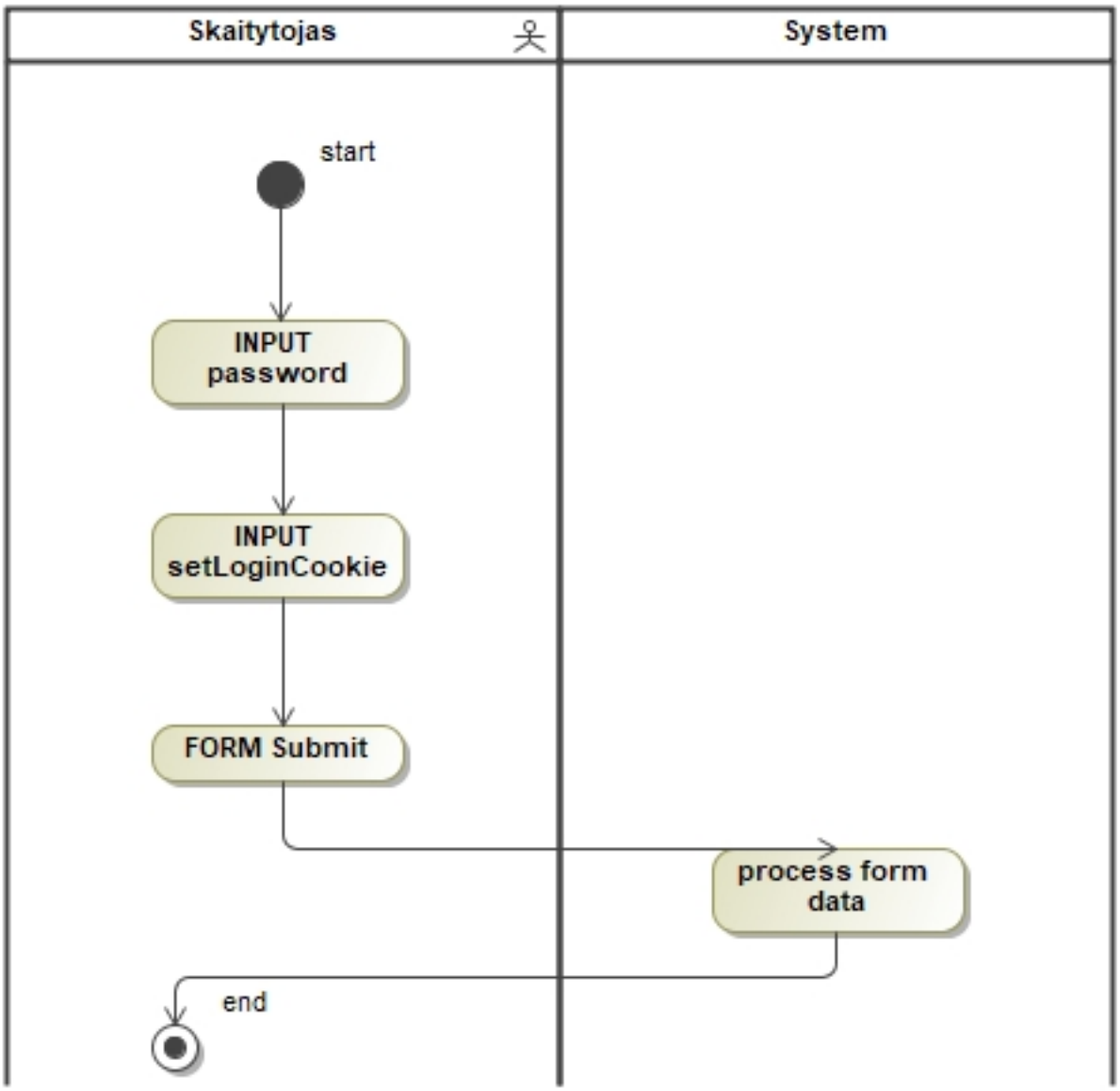

teacher
System
start
SPAN section-modchooser-text
SPAN section-modchooser-link
LI
INPUT name
TEXTAREA introeditor[text]
INPUT showdescription
A #
open url: #
SPAN
end

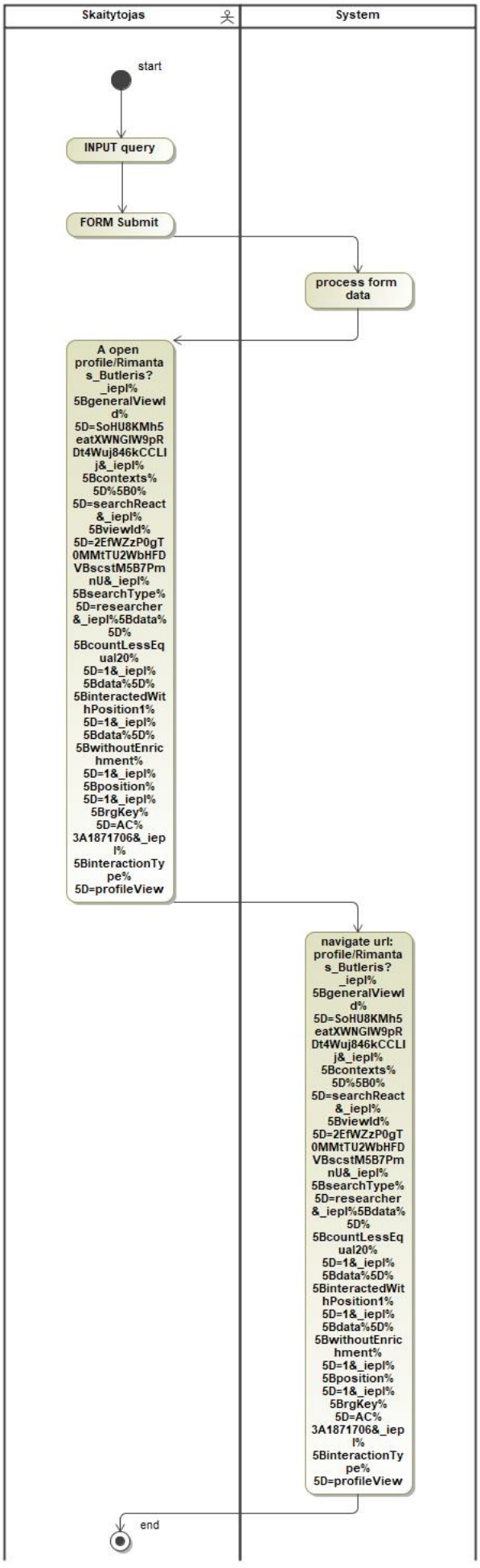

teacher
System
start
A #
open url #
SPAN actions
SPAN menu-action-text
A https://moodle.if.ktu.lt/course/mod.php?sesskey=DADfU2yx0u&sr=0&delete=23556
open url: https://moodle.if.ktu.lt/course/mod.php?sesskey=DADfU2yx0u&sr=0&delete=23556
SPAN actions1
end

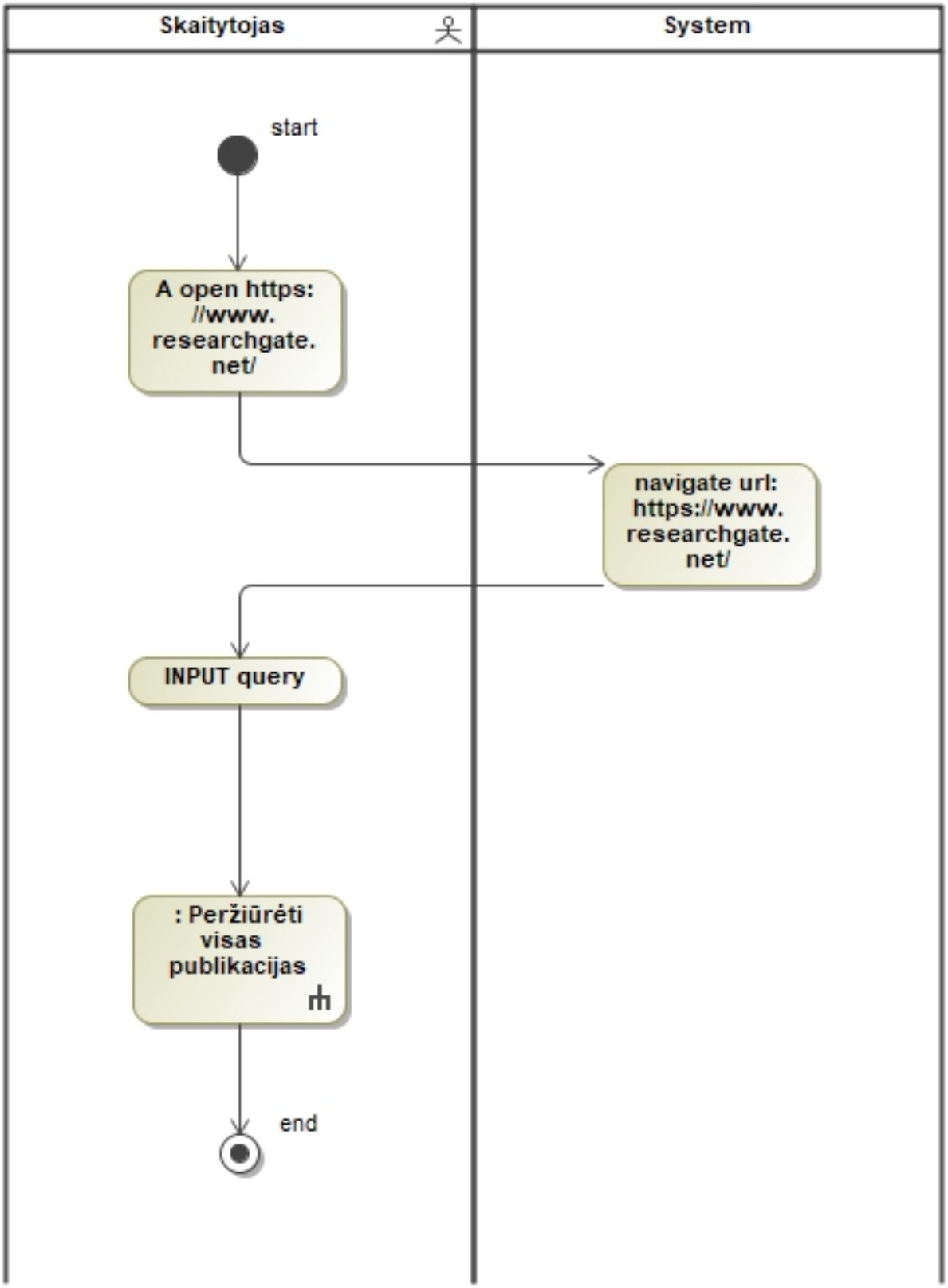
### 9.1.4. priedas. *ResearchGate.com* scenarijui sugeneruotas panaudojimo atvejų modelis.

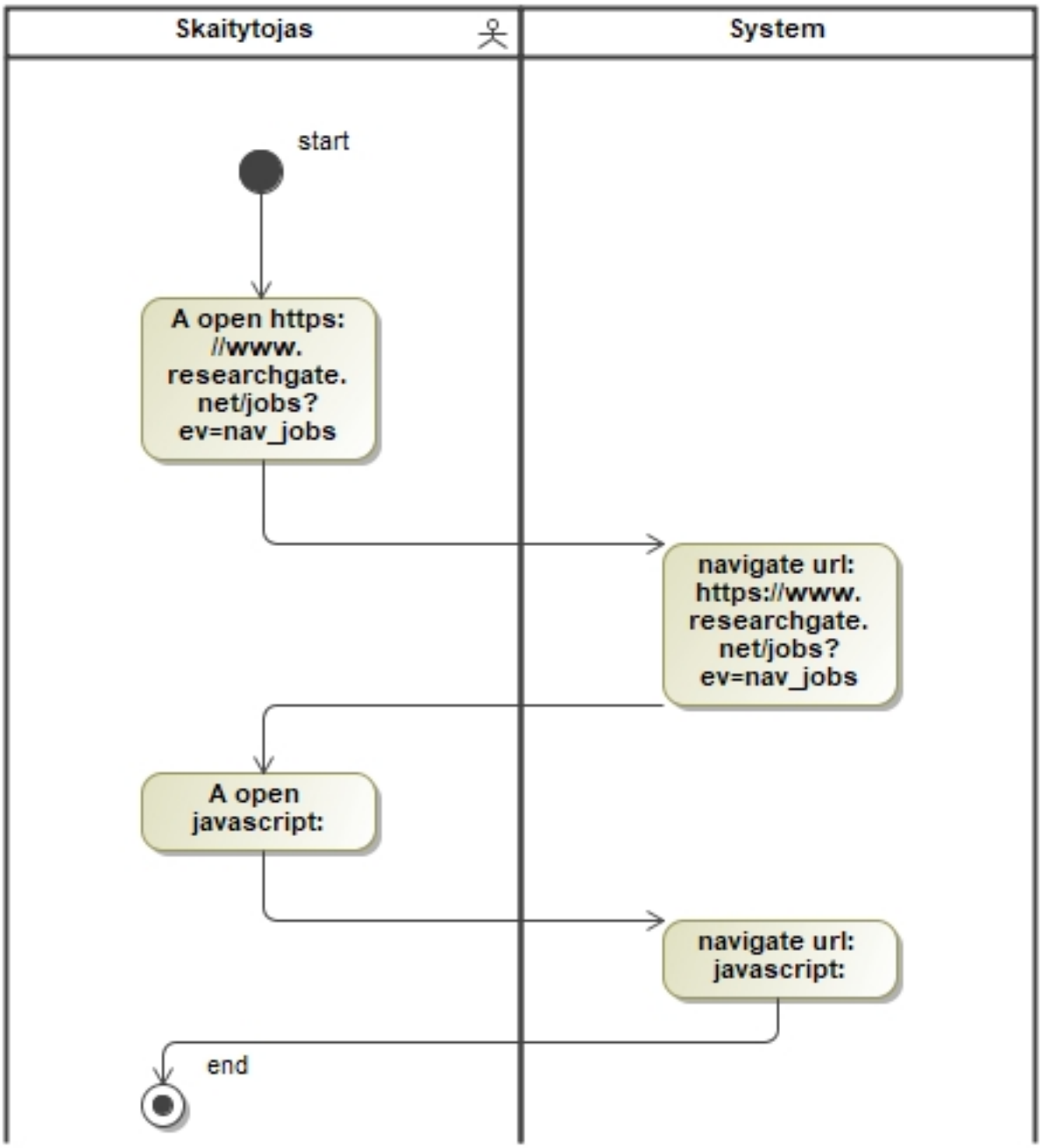
Šiame skyriuje pateikiamos panaudojimo atvejų ir veiklos diagramos, kurios buvo vertintos ekspertų.

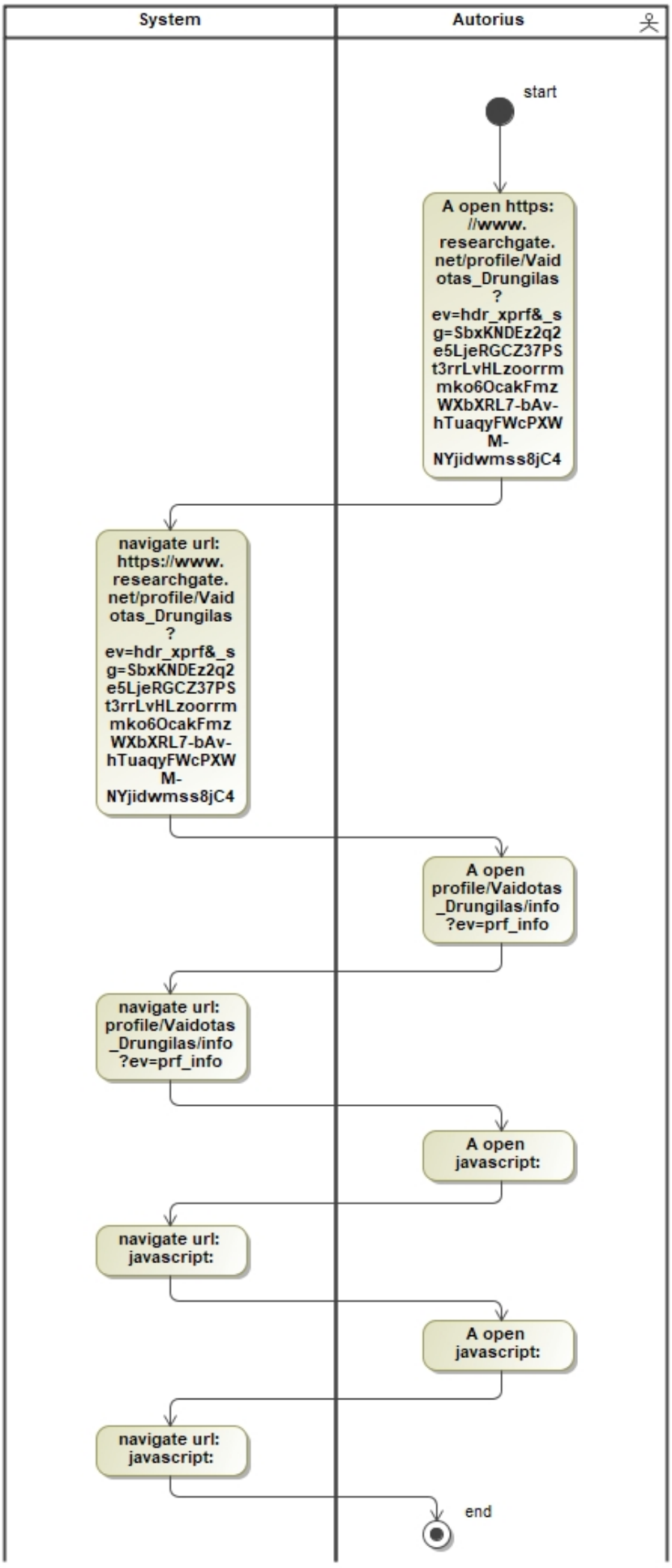


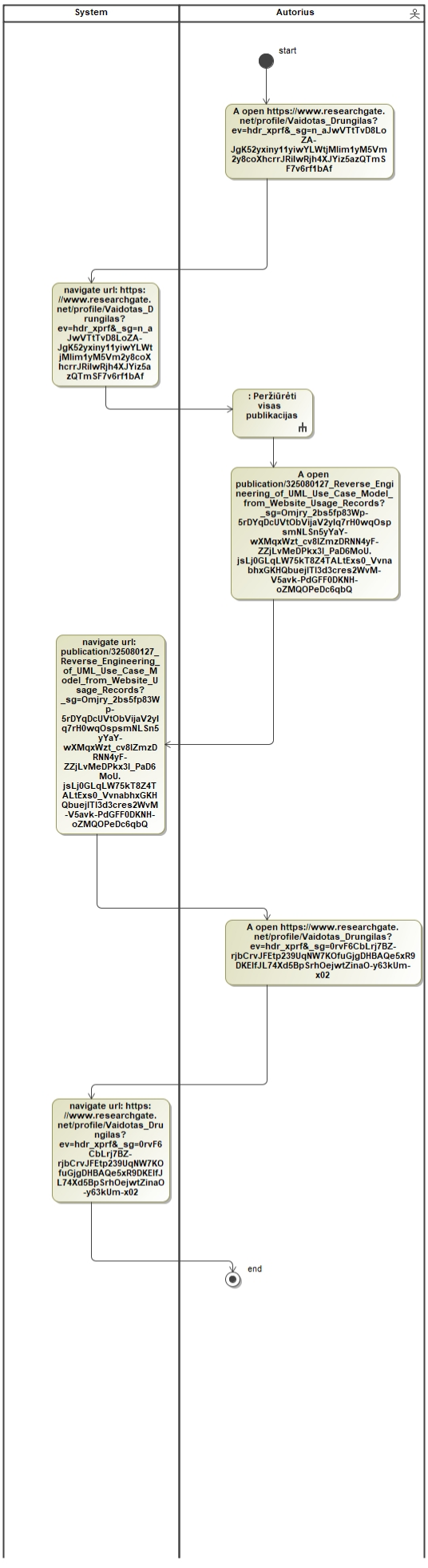

Skaitytojas
System
start
INPUT
password
INPUT
setLoginCookie
FORM Submit
process form
data
end

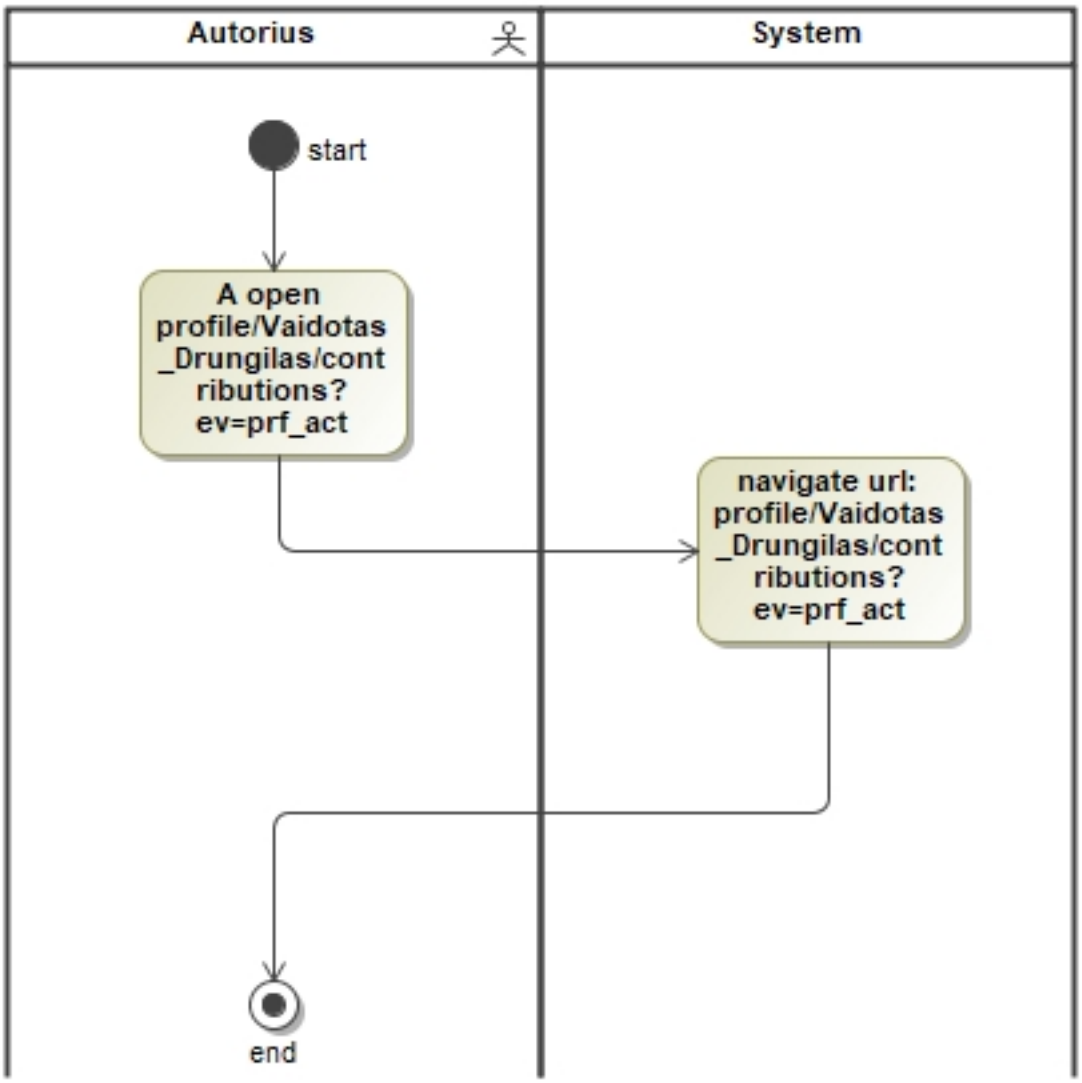

Skaitytojas
System
start
INPUT query
FORM Submit
process form data
A open profile/Rimantas_Butleris?_iepl%5BgeneralViewId%5D=SoHU8KMh5eatXWNGlW9pRDt4Wuj846kCCLIj&_iepl%5Bcontexts%5D%5B0%5D=searchReact&_iepl%5BviewId%5D=2EfWZzP0gT0MMtTU2WbHFDVBscstM5B7PmnU&_iepl%5BsearchType%5D=researcher&_iepl%5Bdata%5D%5BcountLessEqual20%5D=1&_iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&_iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&_iepl%5Bposition%5D=1&_iepl%5BrgKey%5D=AC%3A1871706&_iepl%5BinteractionType%5D=profileView
navigate url: profile/Rimantas_Butleris?_iepl%5BgeneralViewId%5D=SoHU8KMh5eatXWNGlW9pRDt4Wuj846kCCLIj&_iepl%5Bcontexts%5D%5B0%5D=searchReact&_iepl%5BviewId%5D=2EfWZzP0gT0MMtTU2WbHFDVBscstM5B7PmnU&_iepl%5BsearchType%5D=researcher&_iepl%5Bdata%5D%5BcountLessEqual20%5D=1&_iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&_iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&_iepl%5Bposition%5D=1&_iepl%5BrgKey%5D=AC%3A1871706&_iepl%5BinteractionType%5D=profileView
end

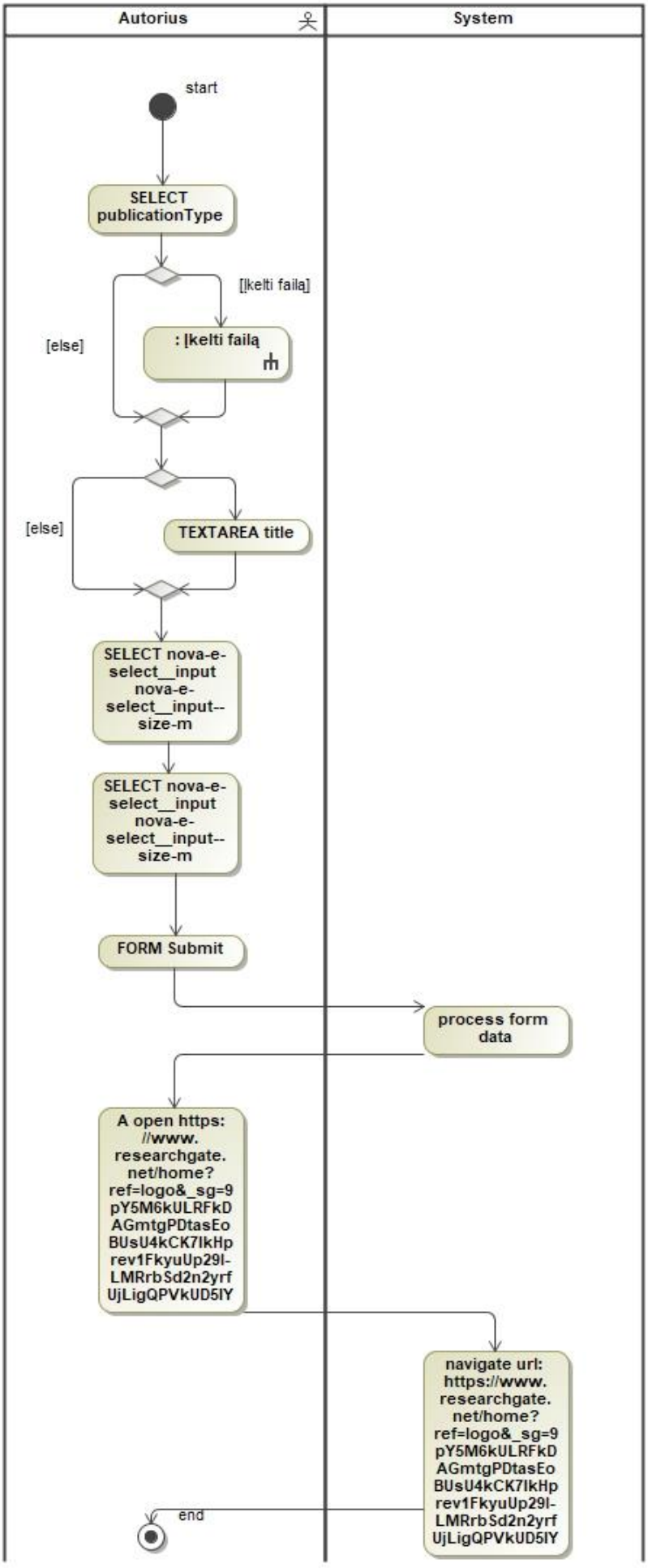

Skaitytojas
System
start
A open https: //www. researchgate. net/
navigate url: https://www. researchgate. net/
INPUT query
: Peržiūrėti visas publikacijas
end

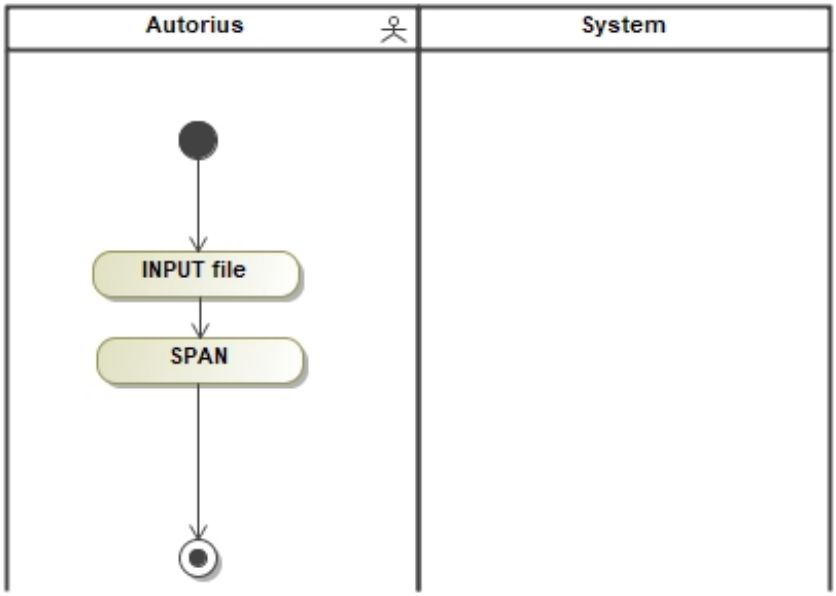

Skaitytojas
System
start
A open https: //www. researchgate. net/jobs? ev=nav_jobs
navigate url: https://www. researchgate. net/jobs? ev=nav_jobs
A open javascript:
navigate url: javascript:
end


System
Autorius
start
A open https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=SbxKNDEz2q2e5LjeRGCZ37PSt3rrLvHLzoorrmmko6OcakFmzWXbXRL7-bAv-hTuaqyFWcPXWM-NYjidwmss8jC4
navigate url: https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=SbxKNDEz2q2e5LjeRGCZ37PSt3rrLvHLzoorrmmko6OcakFmzWXbXRL7-bAv-hTuaqyFWcPXWM-NYjidwmss8jC4
A open profile/Vaidotas_Drungilas/info?ev=prf_info
navigate url: profile/Vaidotas_Drungilas/info?ev=prf_info
A open javascript:
navigate url: javascript:
A open javascript:
navigate url: javascript:
end


System
Autorius
start
A open https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=n_aJwVTtTvD8LoZA-JgK52yxiny11yiwYLWtjMlim1yM5Vm2y8coXhcrrJRilwRjh4XJYiz5azQTmSF7v6rf1bAf
navigate url: https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=n_aJwVTtTvD8LoZA-JgK52yxiny11yiwYLWtjMlim1yM5Vm2y8coXhcrrJRilwRjh4XJYiz5azQTmSF7v6rf1bAf
: Peržiūrėti visas publikacijas
A open publication/325080127_Reverse_Engineering_of_UML_Use_Case_Model_from_Website_Usage_Records?_sg=Omjry_2bs5fp83Wp-5rDYqDcUVtObVijaV2yIq7rH0wqOspsmNLSn5yYaY-wXMqxWzt_cv8lZmzDRNN4yF-ZZjLvMeDPkx3l_PaD6MoU.jsLj0GLqLW75kT8Z4TALtExs0_VvnabhxGKHQbuejITl3d3cres2WvM-V5avk-PdGFF0DKNH-oZMQOPeDc6qbQ
navigate url: publication/325080127_Reverse_Engineering_of_UML_Use_Case_Model_from_Website_Usage_Records?_sg=Omjry_2bs5fp83Wp-5rDYqDcUVtObVijaV2yIq7rH0wqOspsmNLSn5yYaY-wXMqxWzt_cv8lZmzDRNN4yF-ZZjLvMeDPkx3l_PaD6MoU.jsLj0GLqLW75kT8Z4TALtExs0_VvnabhxGKHQbuejITl3d3cres2WvM-V5avk-PdGFF0DKNH-oZMQOPeDc6qbQ
A open https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=0rvF6CbLrj7BZ-rjbCrvJFEtp239UqNW7KOfuGjgDHBAQe5xR9DKElfJL74Xd5BpSrhOejwtZinaO-y63kUm-x02
navigate url: https://www.researchgate.net/profile/Vaidotas_Drungilas?ev=hdr_xprf&_sg=0rvF6CbLrj7BZ-rjbCrvJFEtp239UqNW7KOfuGjgDHBAQe5xR9DKElfJL74Xd5BpSrhOejwtZinaO-y63kUm-x02
end


Autorius
System
start
A open profile/Vaidotas_Drungilas/contributions?ev=prf_act
navigate url: profile/Vaidotas_Drungilas/contributions?ev=prf_act
end


Autorius
System
start
SELECT publicationType
[Įkelti failą]
[else]
: Įkelti failą
[else]
TEXTAREA title
SELECT nova-e-select__input nova-e-select__input--size-m
SELECT nova-e-select__input nova-e-select__input--size-m
FORM Submit
process form data
A open https://www.researchgate.net/home?ref=logo&_sg=9pY5M6kULRFkDAGmtgPDtasEoBUsU4kCK7IkHprev1FkyuUp29l-LMRrbSd2n2yrfUjLigQPVkUD5lY
navigate url: https://www.researchgate.net/home?ref=logo&_sg=9pY5M6kULRFkDAGmtgPDtasEoBUsU4kCK7IkHprev1FkyuUp29l-LMRrbSd2n2yrfUjLigQPVkUD5lY
end


Autorius
System
INPUT file
SPAN

# Reverse Engineering of UML Use Case Model from Website Usage Records

Vaidotas Drungilas
Department of Information Systems,
Kaunas University of Technology
Informatics faculty
Kaunas, Lithuania
vaidotas.drungilas@ktu.lt

Lina Čeponienė
Department of Information Systems,
Kaunas University of Technology
Informatics faculty
Kaunas, Lithuania
lina.ceponiene@ktu.lt

Mantas Jurgelaitis
Department of Information Systems,
Kaunas University of Technology
Informatics faculty
Kaunas, Lithuania
mantas.jurgelaitis@ktu.lt

***Abstract*—Though UML is rather commonly used for modelling various software systems, if not properly maintained, UML models could lose their practical value. Fixing the mismatch between documentation and the current state of software, requires significant effort from development team. This also applies to systems that have no documentation at all or legacy systems, which documentation is not available. Reverse engineering can be used for generating UML diagrams for existing systems. In this paper we present a method for reverse engineering UML Use Case model from websites. This method enables generating UML Use Case and Activity diagrams from the recorded user activity in the website.**

***Keywords—UML; reverse engineering; website; Use Case diagram; Activity diagram.***

## I. INTRODUCTION

Unified Modelling Language (UML) is rather commonly used for modelling various software systems [1]. UML is applied not only during development of complex software systems but also during maintenance of the systems in use. As for deployed systems that require support and updates, models help to analyze and understand inner structure and functionality of a system [2]. Most of the models and documentation are usually created during initial software development stages. If not properly maintained these models lose their practical value. An example of improper maintenance could be a situation when the final product receives updates and new features, without updating documentation and leaving it obsolete. While using this kind of obsolete documentation, maintenance of software and introducing new features becomes more difficult. Fixing the mismatch between documentation and the current state of software, requires significant effort from development team. This also applies to systems that have no documentation at all or legacy systems, which documentation is not available.

Websites more than any other type of software demand constant updates and fixes to meet changing user demand and to beat harsh competition [3]. This demand and competition puts pressure on web developers to implement changes as fast as possible, without wasting valuable resources and time. As demand grows, website developers tend to concentrate on maintaining created and implementing new software features rather than spending time for updating the documentation. To match the tendency of directing most of the effort into implementation stage, Agile project management methodologies have a tendency to be rather popular among web developers. Software products that are built using Agile methodology, usually do not concentrate on having detailed documentation. Consequently, in this paper we tackle a problem of increasing efficiency of modelling process for websites and suggest reverse engineering as a possible solution.

Reverse engineering is the process of analyzing a system to identify its structure and behavior in order to create its visual representation [4]. Reverse engineering can be used to understand how software works and to transform some kind of static information, like program code, into models and documentation [5].

Reverse engineering can also be used for generating UML diagrams [6] [7] [8]. In this paper we present a method for reverse engineering UML Use Case model from websites. This method enables generating UML Use Case and Activity diagrams from the recorded user activity in the website and websites' HTML files.

The rest of the paper is organized as follows. The second section presents related work in the field of reverse engineering UML diagrams. The third section is dedicated to describing our proposed method for generating UML diagrams from registered user actions. Section four describes the prototype developed for our proposed method. The fifth section presents the results of evaluation of the prototype by applying it for a particular website. The last section summarizes the findings of our research and discusses the future work.

## II. RELATED WORK

Reverse engineering of UML models is rather common in the field of software engineering. Reverse engineering can greatly reduce the effort required to construct UML diagrams. UML diagrams can be divided into two categories, one describes structure of the software system, and the other defines its behavior [9].

Structural UML diagrams can be reverse engineered from code or other static structure. Class diagrams can be easily transformed from static code [8], [10]. Many tools support this option through plugins or default functionality, e.g. Eclipse [11] or Visual Paradigm [12].

On the other hand, reverse engineering of behavioral diagrams, like Use Case and Activity diagrams, is not so commonly implemented and used. Nevertheless, there have been significant effort to create a working method for reverse engineering of UML behavioral diagrams [6] [7] [13] [14].

El-Attar and Miller proposed a method to reverse engineer Use Case models from structured Use Case descriptions [7]. This method requires structured text as an input which is analyzed and transformed into the Use Case diagram. This method if used correctly could greatly improve consistency of software documentation and would allow creating precise and unambiguous Use Case models. The best result using this method could be in early development stages. On the other hand, while using Agile methodologies this method would not be the best choice because it requires additional effort for creating and formatting Use Case specifications.

Another method for reverse engineering UML diagrams has been proposed by Muhairat [6]. It uses event table as an input for generating Use Case diagram. Event table, as defined in [6], has four main elements: event, source of event, action and object. These main elements are later transformed, using the proposed process, which consists of actor identification, relation between actors' identification, use case identification, relation between use cases identification and integration of all found elements. Just like [7] method, this method requires a lot of effort for creating sufficiently detailed event table to generate informative Use Case model. This method could be the most useful during requirement analysis phase.

Much can be learned not only form reverse engineering Use Case diagrams, but also from reverse engineering other behavioral diagrams. An excellent example of reverse engineering behavioral diagrams is presented in research by Ziadi, da Silva, Hillah, and Ziane [13]. They proposed an approach how to fully dynamically reverse engineer UML Sequence diagrams. This dynamic method is intended for the systems where static code analysis is not applicable directly. Approach also defined how to extract the traces of a working system. This idea is used as a basis in one of the steps of our proposed method for extracting website usage information.

Di Lucca, Fasolino and Tramontana proposed a specialized tool specifically intended for website reverse engineering, called WARE [14]. This tool is capable to reverse engineer UML Use Case and Sequence diagrams as well as Class diagrams. But instead of working with dynamic content, this tool uses static code as an input for reverse engineering. This tool is only applicable in situations where the full access to source code is granted. In contrast, our research focuses on reverse engineering UML behavioral diagrams independently of the availability of the websites' source code.

Most of the research conducted in the area of reverse engineering Use Case model is not intended particularly for websites. Our approach focuses on reverse engineering UML behavioral diagrams from websites, specifically from the information recorded during website usage. Our proposal is based on the idea that websites use a common architecture which can be used to extract information about user activities.

## I. Proposed method for Reverse Engineering UML Use Case model

As UML Use Case model provides detailed overview of systems functionality, it is one of the main components of high quality system documentation [15]. Our approach in reverse engineering UML Use Case model should provide ability to flexibly analyze web applications, and transform analysis results into UML Use Case model. The created Use Case model should include Use Case diagram along with each Use Case specified by an Activity diagram defining the functionality of that Use Case.

The proposed method consists of two main steps for reverse engineering UML Use Case model from the selected website (Fig. 1):

1) registering usage of the analyzed system;
2) transforming registration results into XMI file.

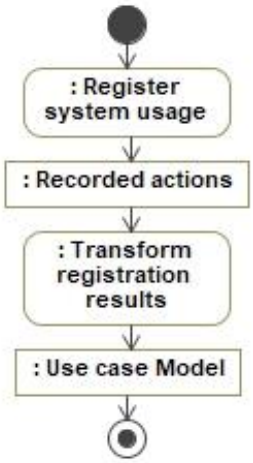



Fig. 1 Structure of the proposed method

During the first phase of reverse engineering UML Use Case model, user opens the system that will be analyzed. He inputs his role in the system and activity that he will be performing. User then continues to use system while his activity is being recorded by reverse engineering system in the background. User can input as many roles and activities as it is required to completely represent his usage of the system. Registering of user actions should be performed by a number of users that is required to cover all functionality of system. After all users register their activities, they should be able to export result files and send them to the system analyst. System analyst then should be able to merge all the result recordings together, into one full structure that represents actions performed in the analyzed system.

### A. *System usage registration process*

The component that will be used for registering system usage should not interfere with system functionality by any way. The recording component should be able to read HTML files that user is interacting with. It should work as a background process that captures user input events, such as clicks and form submits. To describe these events correctly, recording component should also store information about HTML elements that user interacts with. These elements should be uniquely described with an identifying element. Registration component should allow user to define what kind of role he is performing in the system and to define what kind of activity he will be performing. The process of registration consists of initialization and recording steps as can be seen in Fig. 2

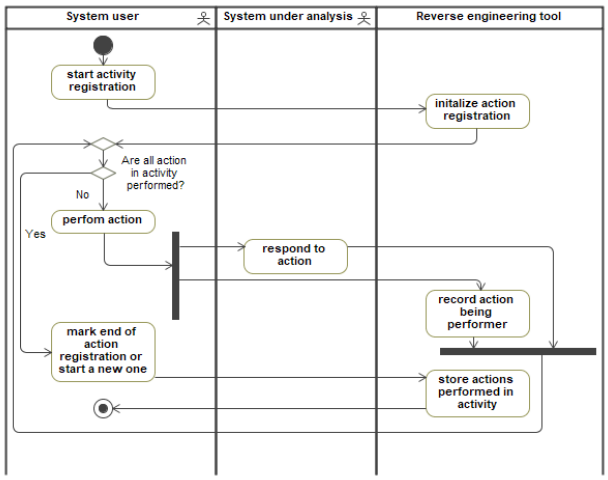



Fig. 1 Activity diagram representing the process of system usage registration

### B. *Registration result*

Registration results are then provided as an input for the component that transforms user actions into UML Use Case model. This input should be stored in a structure that has elements described in Fig. 2. This structure should store all Website URLs that user interacted with during time of recording. In addition, user should provide the name of the Role, which exists in given Website. Each Role will be performing some kind of Activity. Each Activity should be defined by Webpage that it was performed on and events that were performed in that same Webpage. Each event is described with detailed information about type of Action, and information about HTML element that was in use during that event.

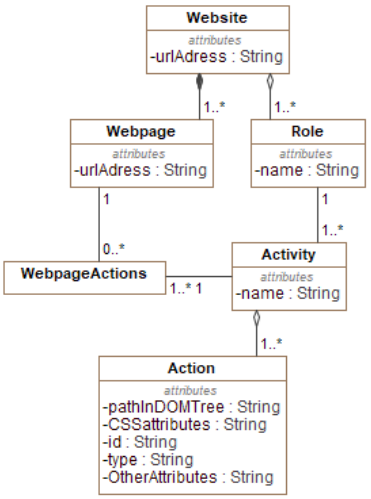



Fig. 2 The structure of the user activity registration result

### C. *Transformation process*

Transformation process is defined in Fig. 4. In order to create a detailed UML model, transformation step should be performed. During this step, registration input is being analyzed for detecting relations between actors and use cases, also between use cases themselves.

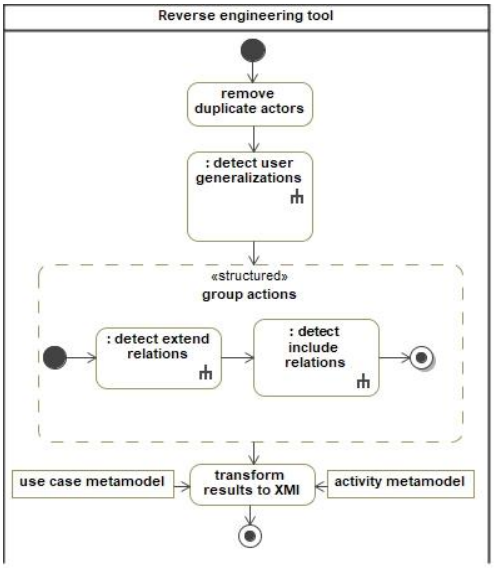



Fig. 3 Activity diagram representing the process of transformation to XMI file

Activity diagram defining the transformation (presented in Fig. 4) specifies what actions are required to transform registered results to XMI. The first action removes duplicate actors, to keep the model concise. The second action is required to detect and create generalization relations between actors. The third action performs grouping

include relations between use cases. The final step takes the results from all relation detection steps and creates XMI file basing on the structure of metamodels of UML Use Case and Activity diagrams.

### A. *Generalization relation detection*

Generalization relation detection starts by scanning all registered use cases and searching for two or more actors which have matching use cases. Algorithm also checks whether it needs to create a new user, in order to display a generalization correctly. If the user creation is required, the system analyst should be prompted to input actor name for new actor. After these steps, the system creates generalization relations between the actors. As a last step, the system maps all required use cases to required actors. Generalization relation detection step is the only step that changes configuration with actors in the model, so after this step we will have the final number of actors in the model. The activity diagram describing generalization detection process is defined in Fig 5.

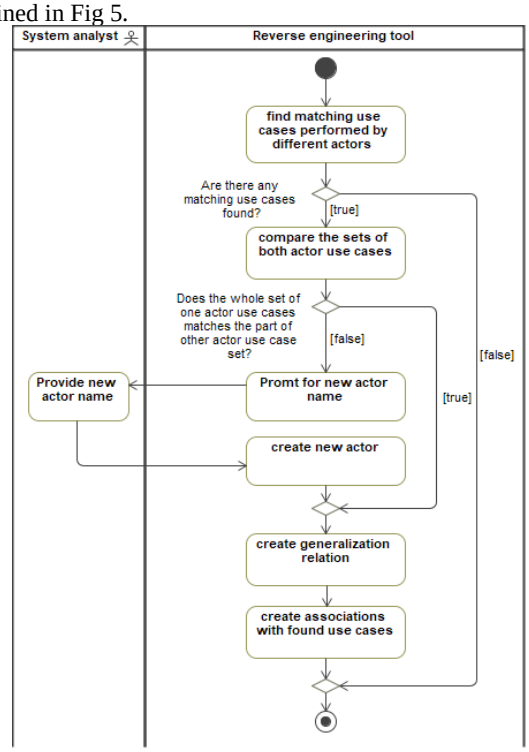



Fig. 1 The process of detecting generalization relations

### B. *Detecting relations between Use Cases*

Detection of use case relations like extend and include, depends on a data set provided by the user of proposed method. Our method does not detect generalization between Use Cases, only include and extend relationship. To detect extend relations, the user of a system should record the same activity on a website twice or more. If these data sets of the same activity will provide exactly the same information, detection would just discard it. Otherwise, if some differences would be found in these data sets, the proposed algorithm can create a more detailed use case model. Success of relation detection depends on an amount of data that user provides during the recording. The higher amount of recorded data should transform into a model that is more detailed and thus more informative.

#### 1) *Detection of extend type relations*

To enhance the Use Case model, our proposed approach detects extend relations. All the actions required to detect extend relations are represented in Fig. 6. These extend relations are important in describing alternative scenarios in the model. Extend relation detection starts with checking all recorded action sequences and finding partially repeating actions. From these sequences, the matching parts are extracted, by comparing them to each other. At the beginning and end of the extracted sequence, decision and merge points are created respectively. Afterwards the extracted, remaining and newly created elements are merged together to create the complete activity diagram. For each path that is now separated from the main path of the activity flow by a decision point, a new use case can be created. User provides the names for these use cases and the algorithm creates them in a model. The next step in extension relation detection is creating extension relation between newly created use cases and the use case they originated from. Finally the extension points are created for the use case, which has incoming extend relations.

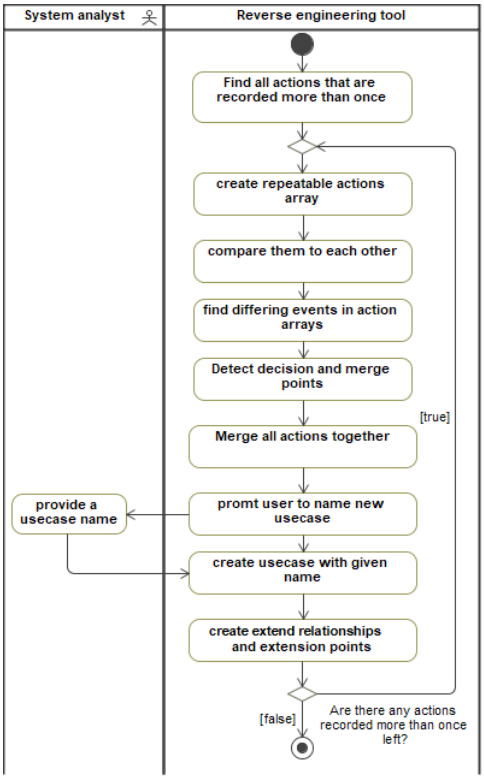



Fig. 2 The process of detecting extend relations

In order to decrease redundancy in use case model, include relations can be used between use cases. As in our method the amount of recorded data should be quite big, include relations help to reduce the number of repeated actions in the model. Include relation detection starts with finding all repeating action sequences where repeated sequence length is higher than user defined number N. For each sequence found, the algorithm starts use case creation by prompting the user to input a use case name and creating use case element with that name. These found sequences are then removed form use cases where they originated. To finalize, include type relations are created by joining newly created use cases and use cases that the sequences originated from. The process of include relation detection is presented in Fig. 7

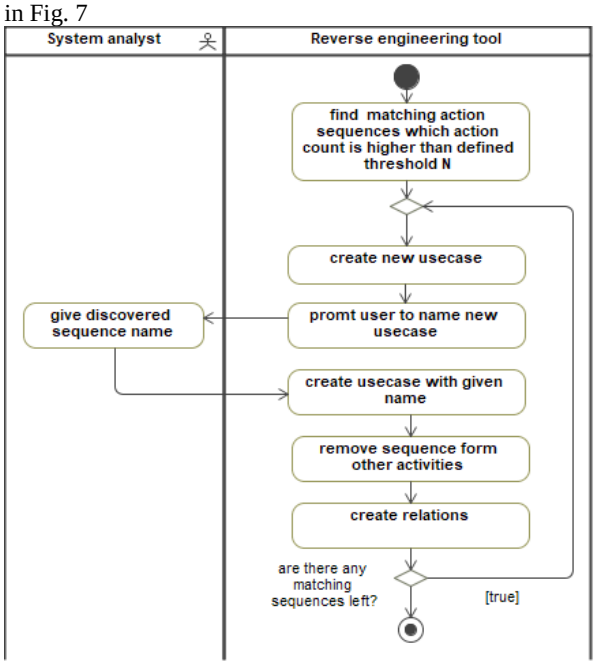



Fig. 1 The process of detecting include relations

### A. *Transformation to XMI format*

During the transformation to XMI step, all the information gathered in previous steps is transformed into Use Case and Activity diagrams. Transformation starts by creating a Use Case diagram. For each role that the user defined, an actor is created. For each activity, that the user defined, the use case is created. Moreover, the use cases that were found during detection of extension and include relationships are added to the model. All use cases and actors are joined using the detected relationship. For each use case, an Activity diagram is created. In this diagram, the algorithm creates two swimlanes, the first one for the actor that is interacting with the given use case, and the second for the system under analysis. For each recorded action our algorithm creates the action in activity diagram. Actions are named referring to the action naming rules, selecting the verb corresponding to the action performed on a HTML element as well as a noun extracted from HTML element attributes. These rules define how each action on HTML element should transform into semantically correct action name. In a systems swimlane, actions are created describing the opening of new webpages. Initial and final activity nodes are also created and all nodes and activities are joined together in a continuous flow.

### B. *Transformation results.*

Results after generation are stored in XMI file. This file consists of Use Case diagram with elements that are described by UML Use Case metamodel [9]. For each use case, an activity diagram is created, representing all actions that the user performs in the system under analysis. Activity diagram is also based on UML metamodel for Activities and Actions [9]. The main elements that this method detects are roles, use cases, and actions. Relations join each of these elements: association relationship joins use cases and actors, generalization is used between actors, include and extend relations can join two use cases, and control flow relations connect the actions in activity diagrams. User can download the generated XMI file and store it as needed. As most of UML modeling tools support XMI as their import format, users should just import this file and have the working version of Use Case model.

## II. The Implemented Prototype of the Proposed Method

To test whether the proposed method could be utilized in practice, a prototype has been implemented. Prototype was realized as a Chrome plugin using JavaScript. It enables users to submit information about their role in the website activity they will be performing. As user continues to use the system that is being analyzed, his actions are recorded. Recorded actions are then stored in JSON file. After user indicates that he has ended registration process, he can start transformation process. The system transforms registered JSON structure to Use Case and Activity diagrams. Example of this JSON structure is presented in *Fig. 2*.


Fig. 2 An example of JSON structure displaying use case "Download assignments data"

This model can later be viewed, analyzed and modified by the analyst.

## I. Example of UML diagrams generated using the implemented prototype

An experiment was conducted to verify whether the created prototype is capable of reverse engineering UML Use Case model. Website selected for this experiment was a virtual learning environment Moodle, customized for Kaunas University of Technology. The roles of student and teacher were analyzed.

In order to create Use Case model, student and teacher were asked to perform actions in the analyzed website. Student performed actions for uploading a file into the system. Teacher recorded a process in which he downloaded the students' submitted assignment. Fig. 9 displays the generated Use Case diagram. For this diagram, users provided two roles but the system identified one additional actor. In total three use cases were detected. The first actor was discovered by generalization relation detection step during registration result transformation. As both of the actors had to log into the system a new user named "Guest" was created. This user, as mentioned before, provides the ability to subtract the amount of excess use cases. For each of these use cases, the algorithm created an activity diagram as expected.

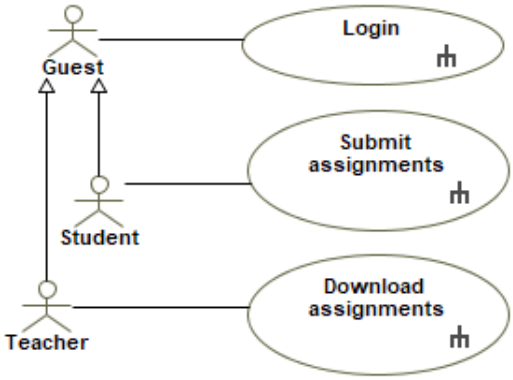



Fig. 1 The generated Use Case diagram

Activity diagram describing Use Case "Login" is presented in *Fig. 2*. In this diagram we can see that each non-authenticated user had to click login button styled by CSS class named "btn-login". After the button is clicked, the system transfers all non-authenticated users to unified authentication system where user fills out login fields and submits the form. To finish login use case, the user clicks Login button and is transferred to the virtual learning environment. The generated Activity diagram demonstrates the prototypes ability to specify common actions, like login and registration.

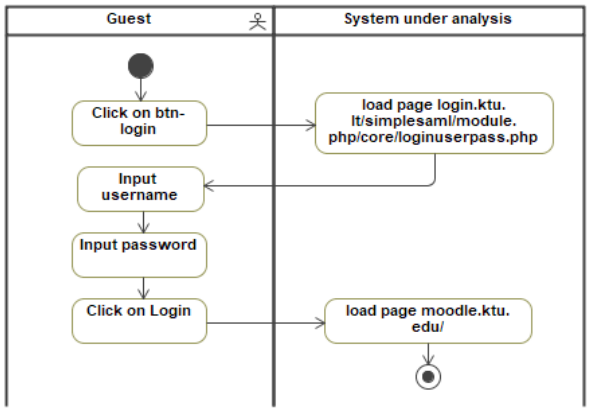



Fig. 2 Generated activity diagram describing *Login* use case

An example of generated activity diagram for use case "Submit assignments" is displayed in Fig. 3. This diagram displays interactions the user performed and webpages he opened. The diagram describes algorithms' ability to record actions, and to transform them correctly. The action naming conventions do not convey the performed action information clearly, it heavily depends on the systems configuration. Most of the students' actions in this use case were navigation through the website. One downside of using just URL changes to describe the opening of new windows in the system, is that it cannot record the opening of a modal window. The algorithms' ability to detect modal windows should be improved in the future.

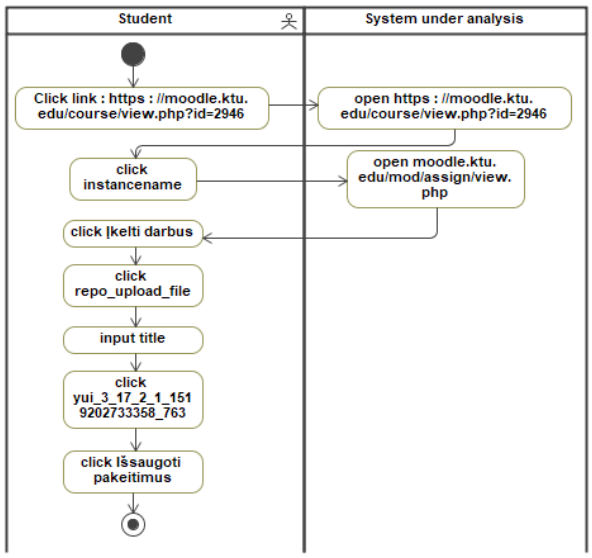



Fig. 3 Generated activity diagram describing *Submit assignment* use case

The example use case "Download assignment" is described in activity diagram presented in **Error! Reference source not found.**. This activity provides visual feedback, demonstrating that some of the URL naming

rules should be improved. On the other hand, this activity still provides enough detail to cover all the most important actions of the use case.

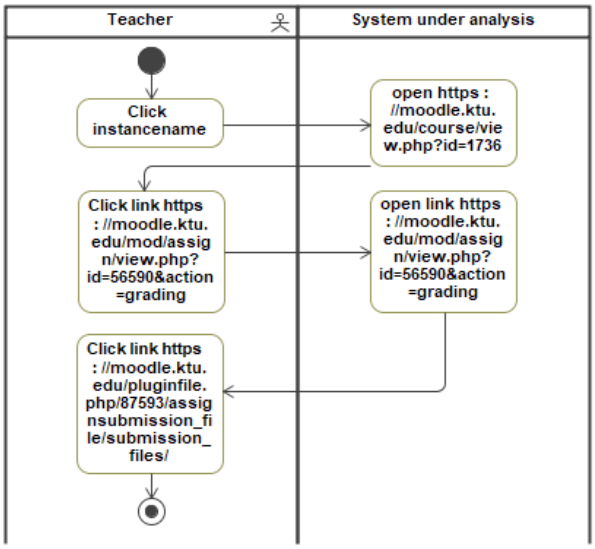



Fig. 1 The generated activity describing *Download assignment* use case.

The results of experiment indicate that created prototype is capable of creating UML Use Case model. Both Use Case and Activity diagrams were created describing system usage in great detail. As current prototype is only capable of detecting generalization relations, further iterations of this prototype will only increase the expressivity of generated models. Generation of activity diagrams describing each use case provide even more depth to generated UML Use Case model. As semantic value of these Activity diagrams can still be improved in future releases, generated Activity diagrams still provide enough information about system usage.

## I. Conclusions and Future Work

Reverse engineering of UML diagrams is utilized in various areas of software engineering. There are many applications of reverse engineering, but most of them are for structural UML diagrams. Behavioral UML diagrams can also be reversed and in our work we have proposed the methodology for reversing UML Use Case model from the data recorded during website usage. Our algorithm analyses recorded user activity and transforms it into UML Use Case and Activity diagrams. The prototype of the proposed algorithm was implemented as a Chrome extension. As this prototype is just the first of its kind, it generates use case and activity diagrams, but is not yet capable of detecting extend and include relations.

The results of experiment indicate that the implemented prototype is capable of generating UML Use Case model. Both Use Case and Activity diagrams were successfully generated using the prototype. The experiment results indicated that our method could be successfully utilized in practice. The experiment also provided valuable feedback about required improvements on action naming rules.

In the future, we are planning to implement the extended capability to support other UML modeling tools. The set of supported tools should include at least one open source tool that is free to use, so that the proposed method could be more accessible to wider audience.

## References


[1] M. R. Chaudron, W. Heijstek and A. Nugroho, "How effective is UML modeling?," in *Software and Systems Modeling (SoSyM)*, 2012.

[2] E. Arisholm, L. C. Briand, S. E. Hove and Y. Labiche, "The impact of UML documentation on software maintenance: an experimental evaluation," *IEEE Transactions on Software Engineering*, vol. 32, no. 6, pp. 365-381, 2006.

[3] G. Rossi, Ó. Pastor, D. Schwabe and L. Olsina, Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag London, 2008.

[4] E. J. Cross and J. H. Chikofsky, "Reverse engineering and design recovery: a taxonomy," in *IEEE Software, vol. 7, no. 1,*, 1990, pp. 13-17.

[5] J. Hibschman and H. Zhang, "Unravel: Rapid Web Application Reverse Engineering via Interaction Recording, Source Tracing, and Library Detection," in *UIST '15 Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, Daegu, Kyungpook, Republic of Korea, 2015.

[6] M. Mohammad I and E. A.-Q. Rafa, "An approach to derive the use case diagrams from an event table," in *8th WSEAS International Conference*, Cambridge, 2009.

[7] M. El-Attar and J. Miller, "Producing robust use case diagrams via reverse engineering," *Softw Syst Model*, pp. 7-67, 2008.

[8] M. I. Muhairat and A. Abdel, "A New Reverse Engineering Approach to Convert," *International Journal of Software Engineering & Applications (IJSEA)*, 2014.

[9] "UML 2.5 Specification," 1 03 2015. [Online]. Available: http://www.omg.org/spec/UML/2.5/PDF.

[10] E. Korshunova, M. P. M. v. d. Brand and M. Mousavi, "CPP2XMI: Reverse Engineering of UML Class, Sequence,," in *Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06)*, 2006.

[11] "eclipse," The Eclipse Foundation, 2108. [Online]. Available: http://www.eclipse.org.

[12] "visual-paradigm," Visual Paradigm, 2018. [Online]. Available: https://www.visual-paradigm.com/.

[13] T. Ziadi, M. A. A. d. Silva, L. M. Hillah and M. Ziane, "A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams," in *16th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*, Las Vegas, United States, 2011.

[14] G. A. D. Lucca, A. R. Fasolino and P. Tramontana, "WARE: a tool for the Reverse Engineering of Web Applications," *Journal of Software Maintenance and Evolution: Research and Practice - Special issue: Web site evolution*, pp. 71-101, 2004.

[15] Richard Soley and the OMG Staff Strategy Group, *Model Driven Architecture*, 2000.