

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Socialinis Vilniaus universiteto tinklalapis

SocialVU

Programų sistemų inžinerijos I laboratorinis darbas

Atliko:	2 kurso 4 grupės studentai	
	Andrejus Voitovas	(parašas)
	Eglė Puodžiūnaitė	(parašas)
	Kasparas Kralikas	(parašas)
	Ieva Vizgirdaitė	(parašas)
Darbo vadovas:	asist. dr. Vytautas Valaitis	(parašas)

ANOTACIJA

Šiame dokumente kuriamos sistemos architektūra aprašoma naudojant UML 4+1 požiūrių rinkinį. Žvelgiama į sistemą 5 skirtingais požiūriais. Loginis pjūvis skirtas parodyti sistemos funkcionalumą, aprašyti santykius tarp esybių. Užduočių pjūvyje pateiksime, kokias užduotis gali įgyvendinti naudotojas, kokie jų įgyvendimo scenarijai. Kūrimo pjūvyje pateikiama, kaip susiję atskiri komponentai. Fiziniame pjūvyje parodoma, kaip sistema išdėstoma tinkle, kaip ji diegiama, kokia įranga naudojama. Procesų pjūvis pateikia dinaminį sistemos modelį: paaiškina sistemoje vykstančius procesus, parodo, kaip procesai komunikuoja, galimus sistemos darbo atvejus.

Rašant šį dokumentą buvo naudojamosi:

1. dr. Vytauto Valaičio internetinis puslapis (<https://klevas.mif.vu.lt/valaitis/>)
2. doc., dr. Karolio Petrausko internetiniu puslapiu (<http://klevas.mif.vu.lt/karolis/>)
3. <https://www.magicdraw.com/files/manuals/MagicDrawTutorials.pdf>
4. Latex programa ir jau sukurtais šablonais

TURINYS

ANOTACIJA	2
ĮVADAS	4
1. LOGINIS PJŪVIS	5
1.1. Esybių klasių diagrama (nulinis lygis)	5
1.2. Klasių diagrama (pirmas lygis)	6
2. UŽDUOČIŲ PJŪVIS	7
2.1. Sistemoje vykdomos užduotys	7
2.2. Užduočių vykdymo scenarijai	8
2.3. Užduoties „Pridėti naujieną“ scenarijus.....	8
2.4. Užduoties „Ištrinti naujieną“ scenarijus.....	9
2.5. Užduoties „Redaguoti naujieną“ scenarijus.....	10
2.6. Užduoties „Peržiūrėti naujienas“ scenarijus	11
2.7. Užduoties „Pridėti renginį“ scenarijus	11
2.8. Užduoties „Ištrinti renginį“ scenarijus	12
2.9. Užduoties „Redaguoti renginį“ scenarijus	13
2.10. Užduoties „Peržiūrėti renginius“ scenarijus	14
2.11. Užduoties „Pridėti D.U.K.“ scenarijus	14
2.12. Užduoties „Peržiūrėti D.U.K.“ scenarijus.....	15
2.13. Užduoties „Redaguoti pasirinktą klausimą“ scenarijus	16
2.14. Užduoties „Peržiūrėti pasirinkto dėstytojo teikiamą informaciją“ scenarijus	17
2.15. Užduoties „Peržiūrėti pasirinktą žinutę“ scenarijus	18
2.16. Užduoties „Siųsti žinutę“ scenarijus	19
2.17. Užduoties „Redaguoti asmeninį dėstytojo puslapį“ scenarijus.....	20
3. KŪRIMO PJŪVIS	21
3.1. Komponentų diagramos nulinis lygmuo	21
3.2. Komponentų diagramos pirmasis lygmuo	22
3.3. Komponentų diagramos antrasis lygmuo	23
4. FIZINIS PJŪVIS	24
4.1. Dislokavimo diagrama nr. 1 (komponentų ir artefaktų ryšių diagrama).....	24
4.2. Dislokavimo diagrama nr. 2 (mazgų ir artefaktų ryšių diagrama)	25
4.3. Dislokavimo diagrama nr. 3 (mazgų ir artefaktų egzempliorių diagrama).....	26
5. PROCESO PJŪVIS	27
5.1. Proceso sekų diagramos	27
5.2. Veiklos diagramos	28
REZULTATAI	31

ĮVADAS

Šio dokumento tikslas - sukurti socialinio tinklalapio prototipą, kurį įgyvendinus būtų palengvinta universiteto bendruomenės komunikacija.

Temos aktualumas

Šiuo metu studentams dėstytojų skelbiama informacija yra išbarstyta internete, kurią surasti užima galybės laiko. Yra atskiras universiteto naujienų puslapis, kiekvienas dėstytojas turi savo asmeninį tinklalapį, atskiras elektroninis paštas. Tiek dėstytojui pasiekti studentus, tiek studentui dėstytoją yra komplikauta ir nepatogu.

Dalykinė sritis

Socialinis Vilniaus Universiteto tinklapis.

Probleminė sritis

Socialinis Vilniaus Universiteto tinklapis suteiktų galimybę greitai ir paprastai pasiekti šio universiteto dėstytojų puslapius, informaciją juose, susisiekti sus pačiais dėstytojais. Pagrindinis tinklalapio išskirtinumas - greitai ir patogiai pasiekama informacija, viskas vienoje vietoje. Itin patogus valdymas dėstytojams.

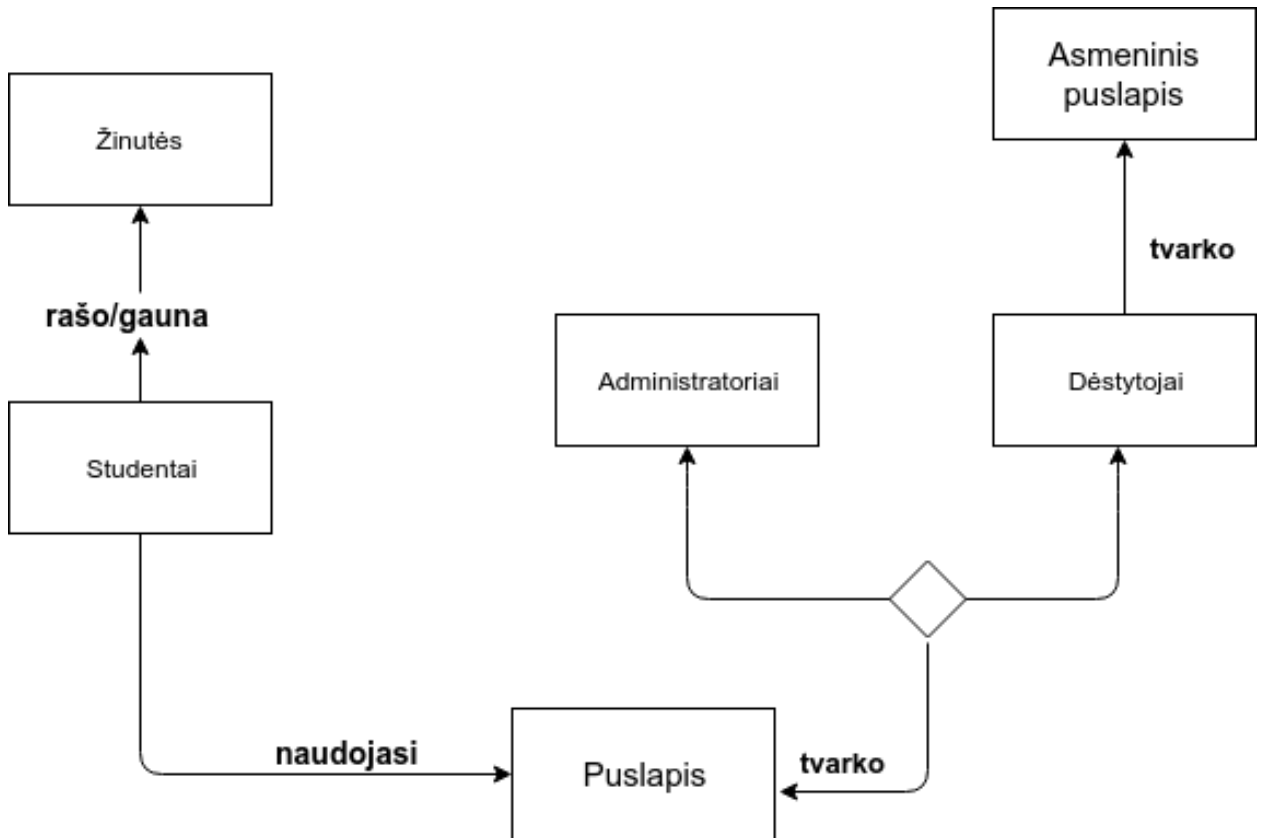
Darbo pagrindas

Dokumentas parengtas kaip Programų sistemų inžinerijos I laboratorinis darbas.

1. LOGINIS PJŪVIS

Loginį pjūvį sudaro klasių diagramos, kurios naudojamos pavaizduoti sistemos architektūros projektavimo etapus.

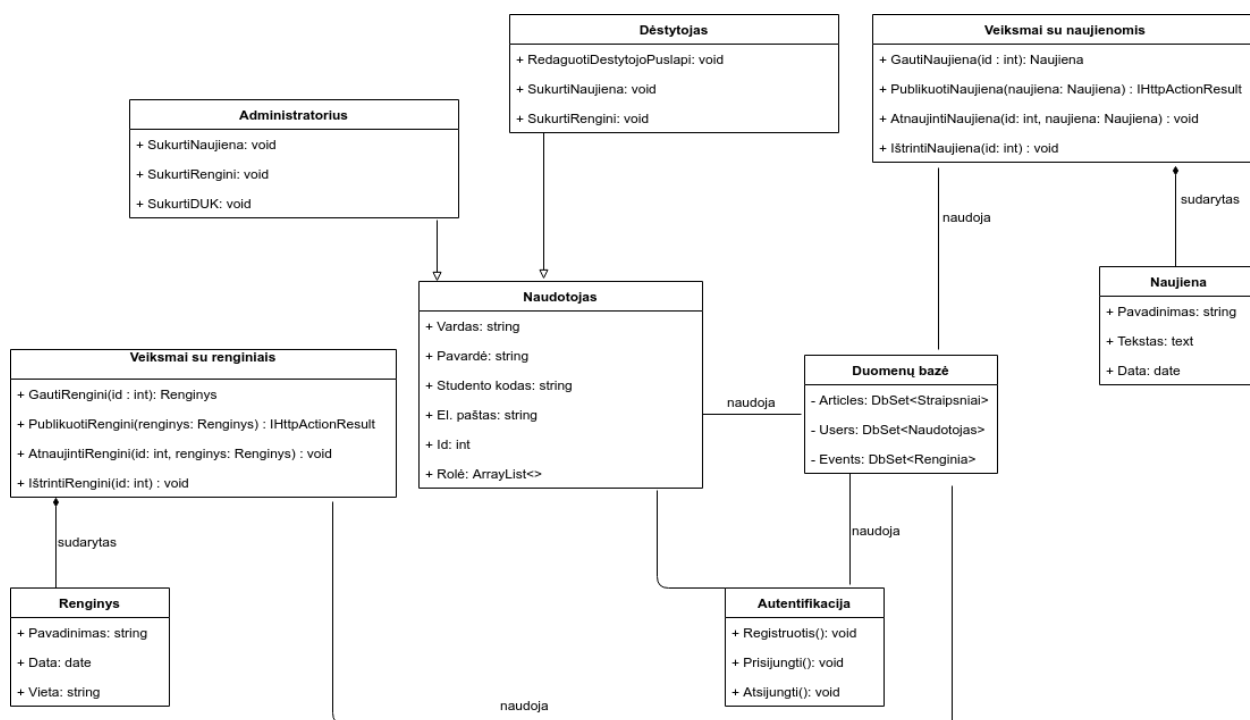
1.1. Esybių klasių diagrama (nulinis lygis)



1 pav. Dalykinė srities UML diagrama

1 pav. esybių diagramoje vaizduojamos esybių sąsajos. Pagrindinė esybė Naudotojas, kuris gali būti Studentas, Dėstytojas arba Administratorius. Studentas turi galimybę naudotis pagrindinėmis puslapio funkcijomis, o dėstytojai ir administratoriai pateikti naudingą studentams medžiagą. Taip pat studentai bei dėstytojai gali komunikuoti tarpusavyje nesinaudojant trečiųjų šalių komunikacinėmis priemonėmis. Administratoriai, savo ruožtu, pateikia informaciją apie renginius, naujienas ir D.U.K.

1.2. Klasių diagrama (pirmas lygis)



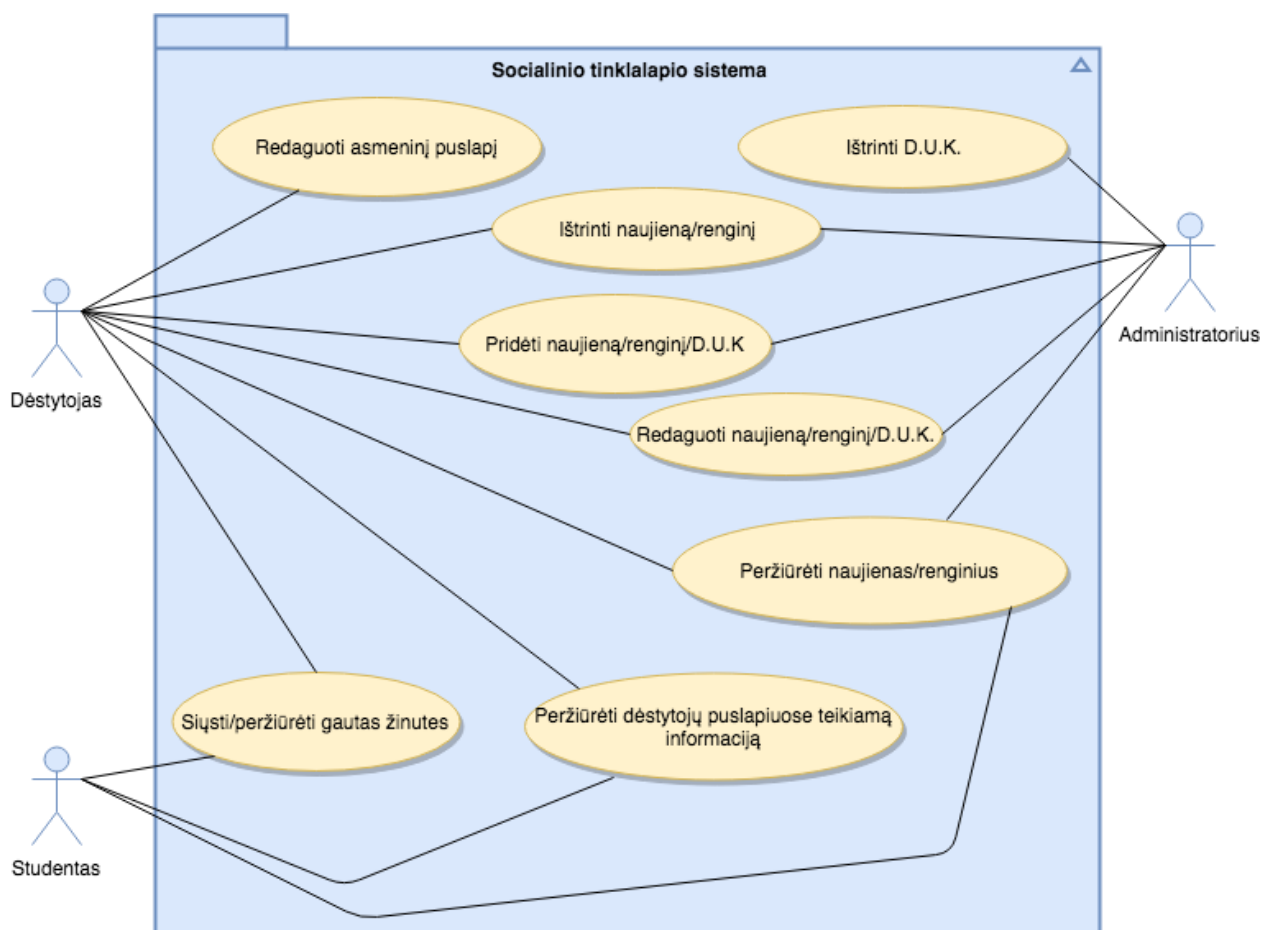
2 pav. Dalykinė srities UML diagrama

Pagrindinį programos funkcionalumą užtikrina šios klasės: Studentas, Dėstytojas, Administratorius, Reitingas, Dėsytojo puslapis, Naujienos, Autentifikacija, Duomenų bazė, D.U.K., Renginiai. Veikimą įgyvendinačių klasių tarpusavio bendradarbiavimas vaizduojamas asociacija, generalizacija, kompozicija bei kardinalumus (2 pav.).

2. UŽDUOČIŲ PJŪVIS

Šiame skyriuje aprašomas kuriamo socialinio tinklalapio galimi panaudojimo atvejai. Pasinaudojant užduočių diagrama pateikiami studento ir dėstytojo (SocialVU naudotojų) tikslai socialiniam tinklalapiui. Kiekvienai užduočiai pateikiamas scenarijus, kuris parodo, kaip užduotis įgyvendinama.

2.1. Sistemoje vykdomos užduotys



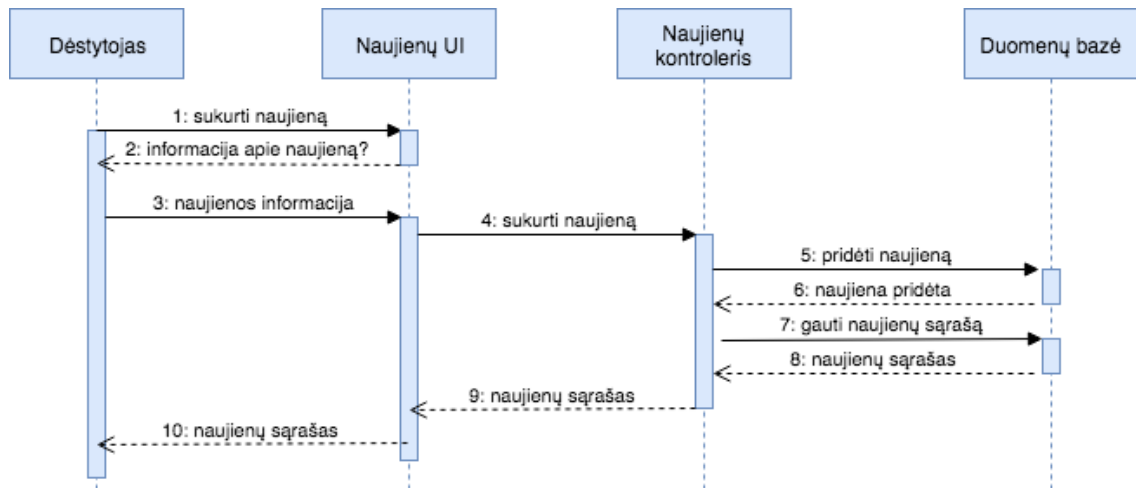
3 pav. Socialiniame tinklalapyje vykdomos užduotys

Sistemoje vykdomos pagrindinės užduotys: studentas gali peržiūrėti pasirinkto dėstytojo puslapius. Dėstytojams sudaroma galimybė pateikti aktualias naujienas, informaciją apie renginius, redaguoti savo asmeninius tinklalapius, kuriuose gali talpinti informaciją apie savo dėstomus dalykus bei kitą naudingą informaciją, kurią matys jų studentai. Tiek dėstytojai, tiek studentai gali gauti informaciją apie juos dominančius renginius, matyti aktualias naujienas, gauti bei siųsti žinutes (3 pav.).

2.2. Užduočių vykdymo scenarijai

Užduočių vykdymo scenarijai, atvaizduoja agentų, šiuo atveju studento ir dėstytojo, įmanomų įvykdyti užduočių veiksmus paeiliui, nuo pradžios iki užduoties vykdymo pabaigos.

2.3. Užduoties „Pridėti naujieną“ scenarijus

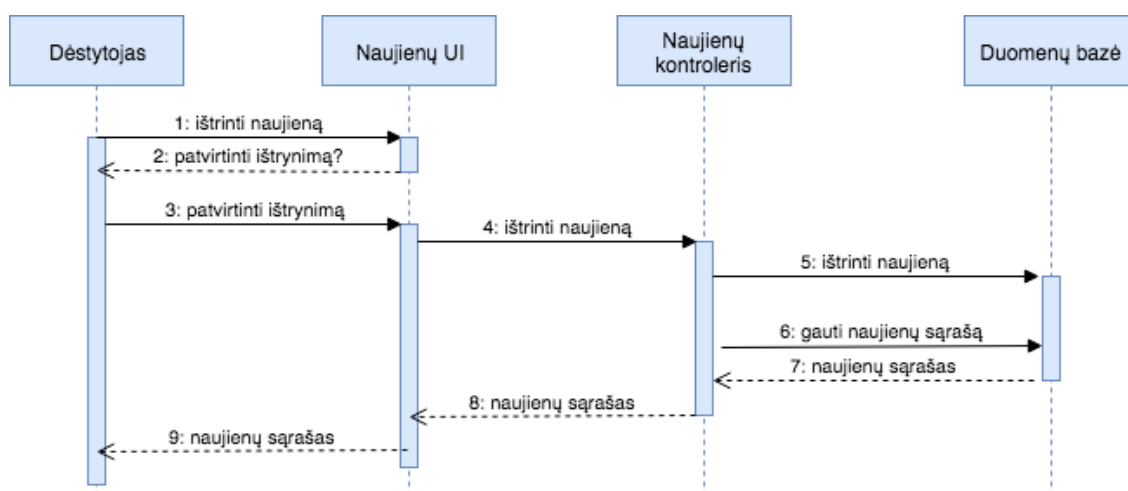


4 pav. Užduoties „Pridėti naujieną“ scenarijus

Žingsnių seka (4 pav.):

1. Sukurti naujieną - dėstytojas paspaudžia ant nuorodos leidžiančios sukurti naujieną.
2. Informacija apie naujieną? - naujienų UI išmeta dėstytojui naujienos formą užpildymui.
3. Naujienos informacija - naudotojas išsiunčia užpildytą formą naujienų UI.
4. Sukurti naujieną - įvesta naujienos informacija yra siunčiama kontrolieriui.
5. Pridėti naujieną - darbų UI įdeda naujieną į duomenų bazę.
6. Naujiena pridėta - duomenų bazė parsiunčia naujienos patalpinimo patvirtinimą.
7. Gauti naujienų sąrašą - naujienų kontrolieris prašo duomenų bazės pateikti naują naujienų sąrašą.
8. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.
9. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.
10. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.4. Užduoties „Ištrinti naujieną“ scenarijus

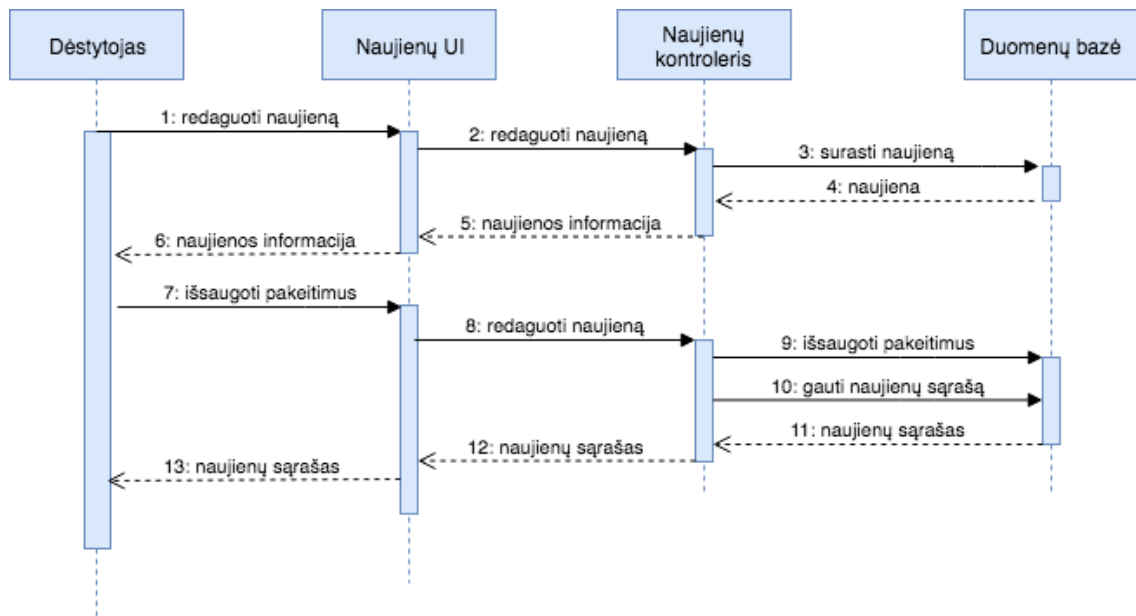


5 pav. Užduoties „Ištrinti naujieną“ scenarijus

Žingsnių seka (5 pav.):

1. Ištrinti naujieną - dėstytojas paspaudžia nuorodą ištrinančią naujieną.
2. Patvirtinti ištrynimą? - naudotojas prašomas patvirtinti ištrynimą.
3. Patvirtinti ištrynimą - dėstytojas patvirtina ištrynimą.
4. Ištrinti naujieną - pateikiama naujiena ištrynimui įvykdyti.
5. Ištrinti naujieną - prašoma duomenų bazės surasti ir ištrinti naujieną.
6. Gauti naujienų sąrašą - siunčiamas prašymas naujienų sąrašui gauti iš duomenų bazės.
7. Naujienų sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.
8. Naujienų sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.
9. Naujienų sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.

2.5. Užduoties „Redaguoti naujieną“ scenarijus

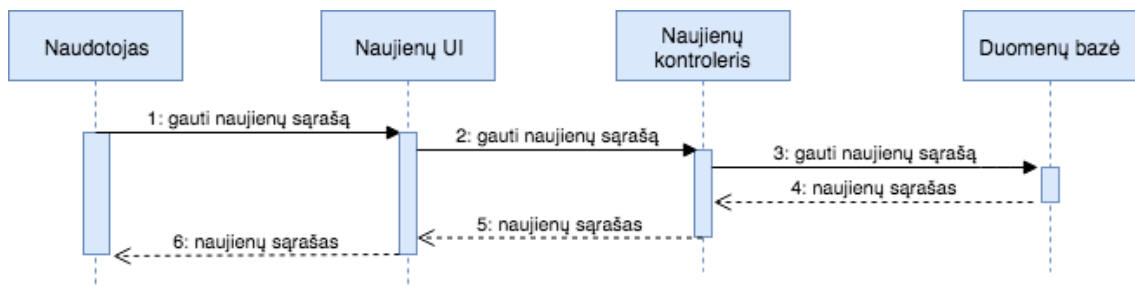


6 pav. Užduoties „Redaguoti naujieną“ scenarijus

Žingsnių seka (6 pav.):

1. Redaguoti naujieną - dėstytojas paspaudžia nuorodą leidžiančią redaguoti naujieną.
2. Redaguoti naujieną - pateikiama naujiena naujienų kontrolieriui.
3. Surasti naujieną - naujiena ieškoma duomenų bazėje.
4. Naujiena - duomenų bazė pateikia naujieną kontrolieriui.
5. Naujienos informacija - informacija apie naujieną keliauja iki naudotojo.
6. Naujienos informacija - informacija apie naujieną keliauja iki naudotojo.
7. Išsaugoti pakeitimus - naudotojas prašo išsaugoti įvykdytus pakeitimus.
8. Redaguoti naujieną - naujiena siunčiama redagavimui.
9. Išsaugoti pakeitimus - pakeitimai išsaugomi duomenų bazėje.
10. Gauti naujienų sąrašą - kontrolieris prašo duomenų bazės gauti atnaujintą naujienų sąrašą.
11. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.
12. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.
13. Naujienų sąrašas - naujienų sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.6. Užduoties „Peržiūrėti naujienas“ scenarijus

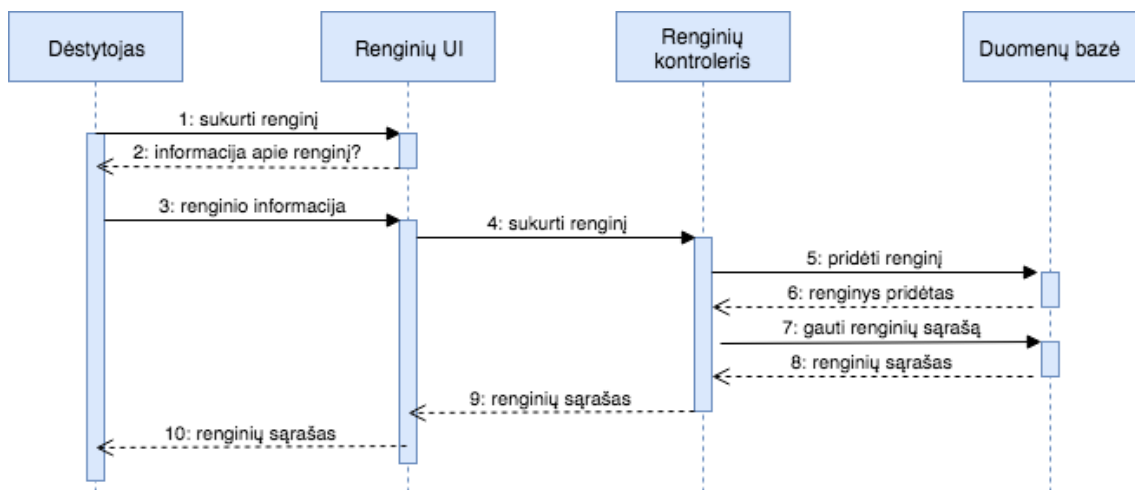


7 pav. Užduoties „Peržiūrėti naujienas“ scenarijus

Žingsnių seka (7 pav.):

1. Gauti naujienų sąrašą - dėstytojas paspaudžia nuorodą į naujienų sąrašą.
2. Gauti naujienų sąrašą - prašymas gauti naujienų sąrašą keliauja iki duomenų bazės.
3. Gauti naujienų sąrašą - prašymas gauti naujienų sąrašą keliauja iki duomenų bazės.
4. Naujienų sąrašas - naujienų sąrašas iš duomenų bazės keliauja iki naudotojo.
5. Naujienų sąrašas - naujienų sąrašas iš duomenų bazės keliauja iki naudotojo.
6. Naujienų sąrašas - naujienų sąrašas iš duomenų bazės keliauja iki naudotojo.

2.7. Užduoties „Pridėti renginį“ scenarijus



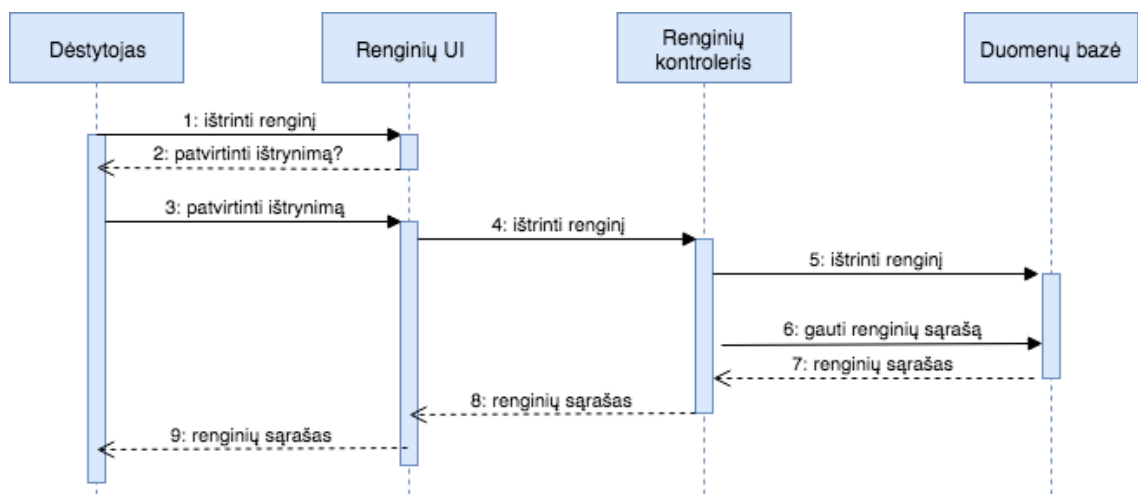
8 pav. Užduoties „Pridėti renginį“ scenarijus

Žingsnių seka (8 pav.):

1. Sukurti renginį - dėstytojas paspaudžia ant nuorodos leidžiančios sukurti renginį.
2. Informacija apie renginį? - renginių UI išmeta dėstytojui renginio formą užpildymui.
3. Renginio informacija - naudotojas išsiunčia užpildytą formą renginių UI.
4. Sukurti renginį - įvesta renginio informacija yra siunčiama kontrolieriui.

5. Pridėti renginį - renginių UI įdeda renginį į duomenų bazę.
6. Renginys pridėtas - duomenų bazė parsiunčia renginio patalpinimo patvirtinimą.
7. Gauti renginių sąrašą - renginių kontrolieris prašo duomenų bazės pateikti naują renginių sąrašą.
8. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.
9. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.
10. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.8. Užduoties „Ištrinti renginį“ scenarijus

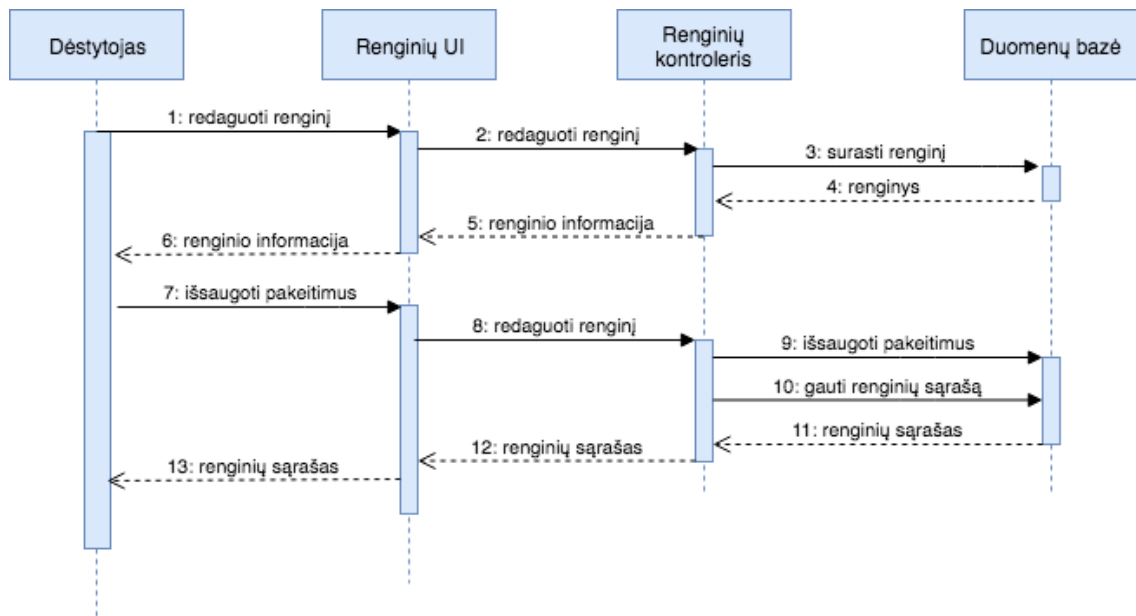


9 pav. Užduoties „Ištrinti renginį“ scenarijus

Žingsnių seka (9 pav.):

1. Ištrinti renginį - dėstytojas paspaudžia nuorodą ištrinančią renginį.
2. Patvirtinti ištrynimą? - naudotojas prašomas patvirtinti ištrynimą.
3. Patvirtinti ištrynimą - dėstytojas patvirtina ištrynimą.
4. Ištrinti renginį - pateikiamas renginys ištrynimui įvykdyti.
5. Ištrinti renginį - prašoma duomenų bazės surasti ir ištrinti renginį.
6. Gauti renginių sąrašą - siunčiamas prašymas renginių sąrašui gauti iš duomenų bazės.
7. Renginių sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.
8. Renginių sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.
9. Renginių sąrašas - atnaujintas sąrašas keliauja iki dėstytojo.

2.9. Užduoties „Redaguoti renginį“ scenarijus

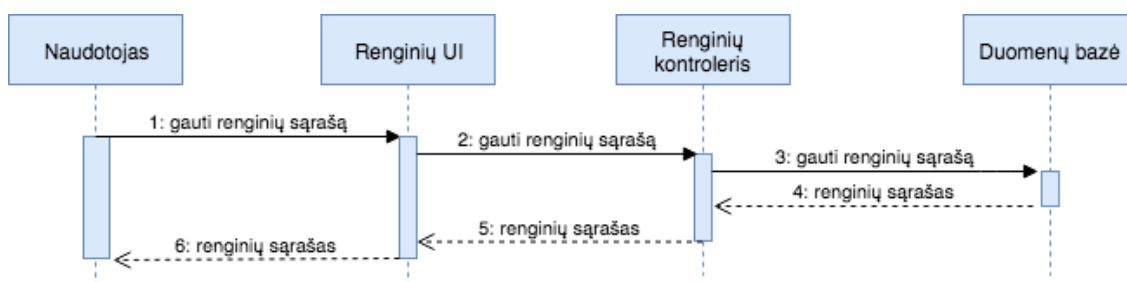


10 pav. Užduoties „Redaguoti renginį“ scenarijus

Žingsnių seka (10 pav.):

1. Redaguoti renginį - dėstytojas paspaudžia nuorodą leidžiančią redaguoti renginį.
2. Redaguoti renginį - pateikiamas renginys renginių kontrolieriui.
3. Surasti renginį - renginys ieškomas duomenų bazėje.
4. Renginys - duomenų bazė pateikia renginį kontrolieriui.
5. Renginio informacija - informacija apie renginį keliauja iki naudotojo.
6. Renginio informacija - informacija apie renginį keliauja iki naudotojo.
7. Išsaugoti pakeitimus - naudotojas prašo išsaugoti įvykdytus pakeitimus.
8. Redaguoti renginį - renginys siunčiamas redagavimui.
9. Išsaugoti pakeitimus - pakeitimai išsaugomi duomenų bazėje.
10. Gauti renginių sąrašą - kontrolieris prašo duomenų bazės gauti atnaujintą renginių sąrašą.
11. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.
12. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.
13. Renginių sąrašas - renginių sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.10. Užduoties „Peržiūrėti renginius“ scenarijus

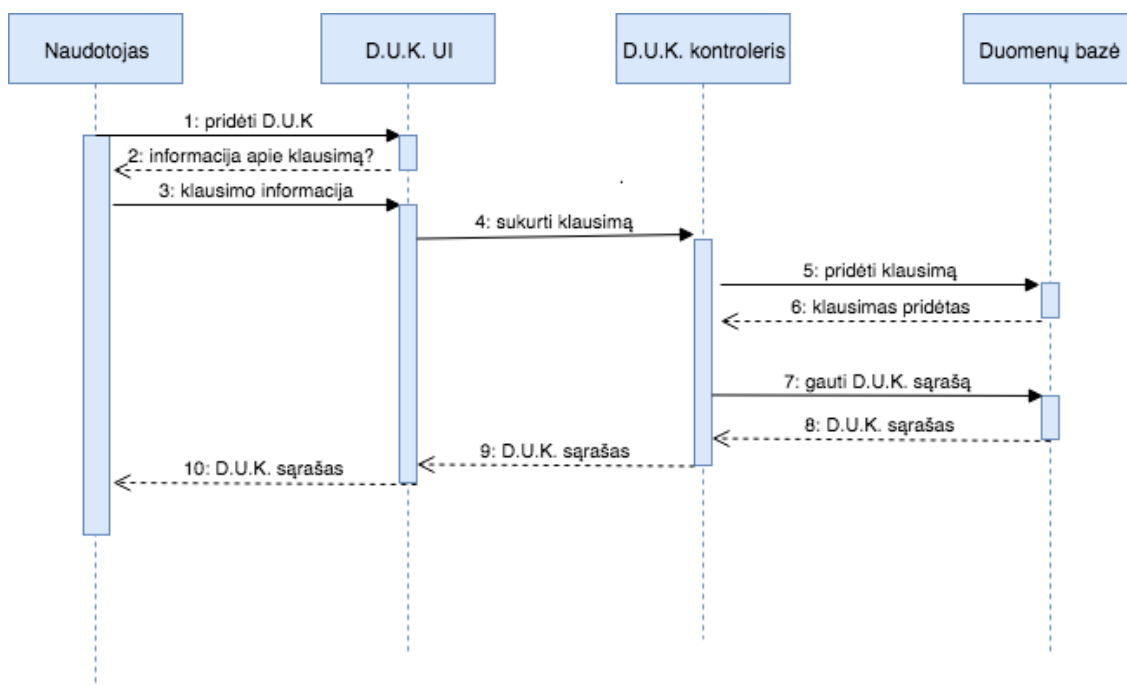


11 pav. Užduoties „Peržiūrėti renginius“ scenarijus

Žingsnių seka (11 pav.):

1. Gauti renginių sąrašą - dėstytojas paspaudžia nuorodą į renginių sąrašą.
2. Gauti renginių sąrašą - prašymas gauti renginių sąrašą keliauja iki duomenų bazės.
3. Renginių sąrašas - renginių sąrašas iš duomenų bazės keliauja iki naudotojo.
4. Renginių sąrašas - renginių sąrašas iš duomenų bazės keliauja iki naudotojo.
5. Renginių sąrašas - renginių sąrašas iš duomenų bazės keliauja iki naudotojo.

2.11. Užduoties „Pridėti D.U.K.“ scenarijus



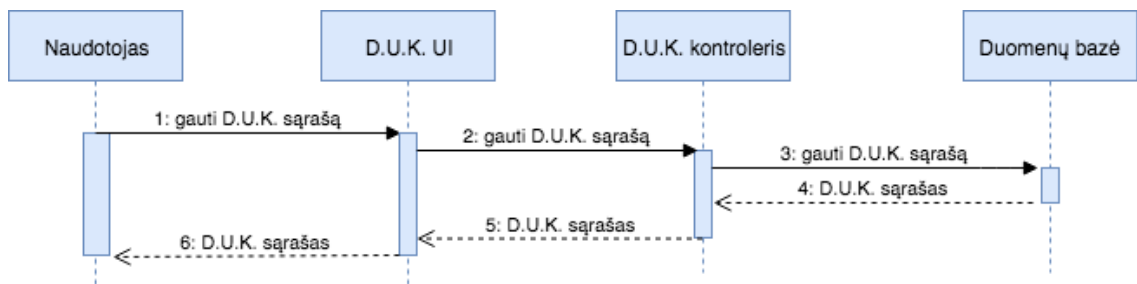
12 pav. Užduoties „Pridėti D.U.K.“ scenarijus

Žingsnių seka (12 pav.):

1. Pridėti D.U.K. - dėstytojas paspaudžia ant nuorodos leidžiančios sukurti D.U.K.

2. Informacija apie klausimą? - D.U.K. UI išmeta dėstytojui D.U.K. formą užpildymui.
3. Klausimo informacija - naudotojas išsiunčia užpildytą formą D.U.K. UI.
4. Sukurti klausimą - įvesta klausimo informacija yra siunčiama kontrolieriui.
5. Pridėti klausimą - D.U.K. UI įdeda klausimą į duomenų bazę.
6. Klausimas pridėtas - duomenų bazė parsiunčia klausimo patalpinimo patvirtinimą.
7. Gauti D.U.K. sąrašą - D.U.K. kontrolieris prašo duomenų bazės pateikti naują D.U.K. sąrašą.
8. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.
9. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.
10. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.12. Užduoties „Peržiūrėti D.U.K.“ scenarijus

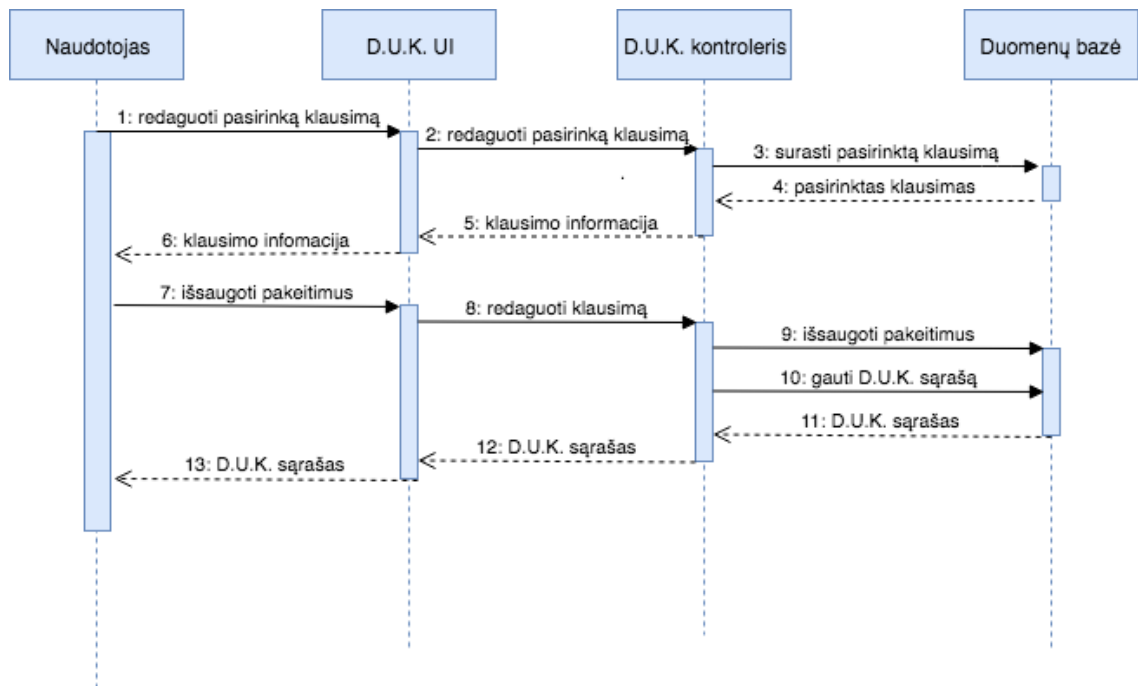


13 pav. Užduoties „Peržiūrėti D.U.K.“ scenarijus

Žingsnių seka (13 pav.):

1. Gauti D.U.K. sąrašą - dėstytojas paspaudžia nuorodą į D.U.K. sąrašą.
2. Gauti D.U.K. sąrašą - prašymas gauti D.U.K. sąrašą keliauja iki duomenų bazės.
3. D.U.K. sąrašas - D.U.K. sąrašas iš duomenų bazės keliauja iki naudotojo.

2.13. Užduoties „Redaguoti pasirinktą klausimą“ scenarijus

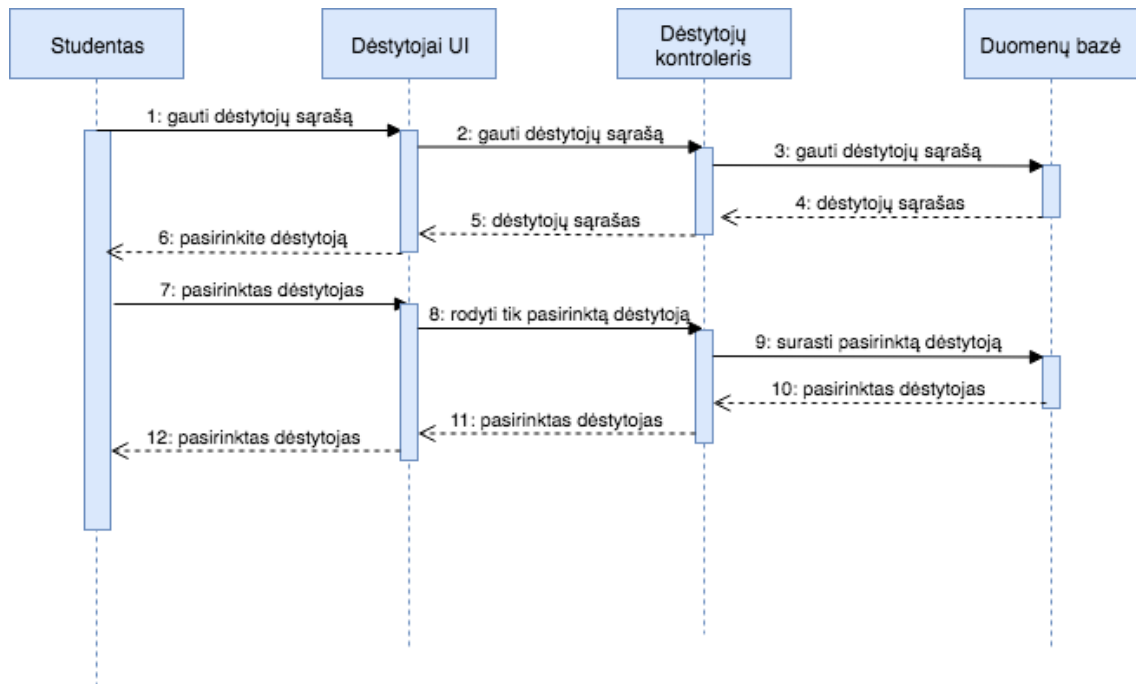


14 pav. Užduoties „Redaguoti pasirinktą klausimą“ scenarijus

Žingsnių seka (14 pav.):

1. Redaguoti pasirinktą klausimą - dėstytojas paspaudžia nuorodą leidžiančią redaguoti klausimą.
2. Redaguoti pasirinktą klausimą - pateikiamas klausimas D.U.K. kontrolieriui.
3. Surasti pasirinktą klausimą - klausimas ieškomas duomenų bazėje.
4. Pasirinktas klausimas - duomenų bazė pateikia klausimą kontrolieriui.
5. Klausimo informacija - informacija apie klausimą keliauja iki naudotojo.
6. Klausimo informacija - informacija apie klausimą keliauja iki naudotojo.
7. Išsaugoti pakeitimus - naudotojas prašo išsaugoti įvykdytus pakeitimus.
8. Redaguoti klausimą - klausimas siunčiamas redagavimui.
9. Išsaugoti pakeitimus - pakeitimai išsaugomi duomenų bazėje.
10. Gauti D.U.K. sąrašą - kontrolieris prašo duomenų bazės gauti atnaujintą D.U.K. sąrašą.
11. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.
12. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.
13. D.U.K. sąrašas - D.U.K. sąrašas keliauja iš duomenų bazės iki dėstytojo.

2.14. Užduoties „Peržiūrėti pasirinkto dėstytojo teikiamą informaciją“ scenarijus

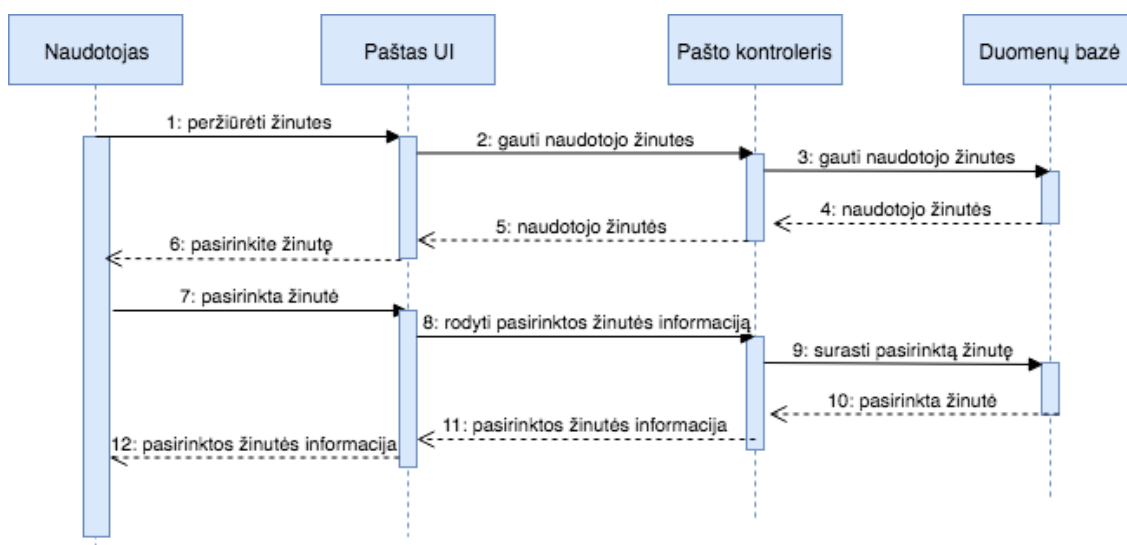


15 pav. Užduoties „Peržiūrėti pasirinkto dėstytojo teikiamą informaciją“ scenarijus

Žingsnių seka (15 pav.):

1. Gauti dėstytojų sąrašą - studentas paspaudžia nuorodą į dėstytojų sąrašą.
2. Gauti dėstytojų sąrašą - prašymas gauti dėstytojų sąrašą keliauja iki duomenų bazės.
3. Gauti dėstytojų sąrašą - prašymas gauti dėstytojų sąrašą keliauja iki duomenų bazės.
4. Dėstytojų sąrašas - duomenų sąrašas keliauja iki naudotojo.
5. Dėstytojų sąrašas - duomenų sąrašas keliauja iki naudotojo.
6. Pasirinkite dėstytoją - prašoma pasirinkti, kurio dėstytojo puslapį norima matyti.
7. Pasirinktas dėstytojas - pasirenkamas dėstytojas, kurio informacija domina.
8. Rodyti tik pasirinktą dėstytoją - dėstytojas siunčiamas kontrolieriui.
9. Surasti pasirinktą dėstytoją - D.U.K. kontrolieris prašo duomenų bazės pateikti dėstytojo puslapį.
10. Pasirinktas dėstytojas - dėstytojo puslapio informacija keliauja iš duomenų bazės iki studento.
11. Pasirinktas dėstytojas - dėstytojo puslapio informacija keliauja iš duomenų bazės iki studento.
12. Pasirinktas dėstytojas - dėstytojo puslapio informacija keliauja iš duomenų bazės iki studento.

2.15. Užduoties „Peržiūrėti pasirinktą žinutę“ scenarijus

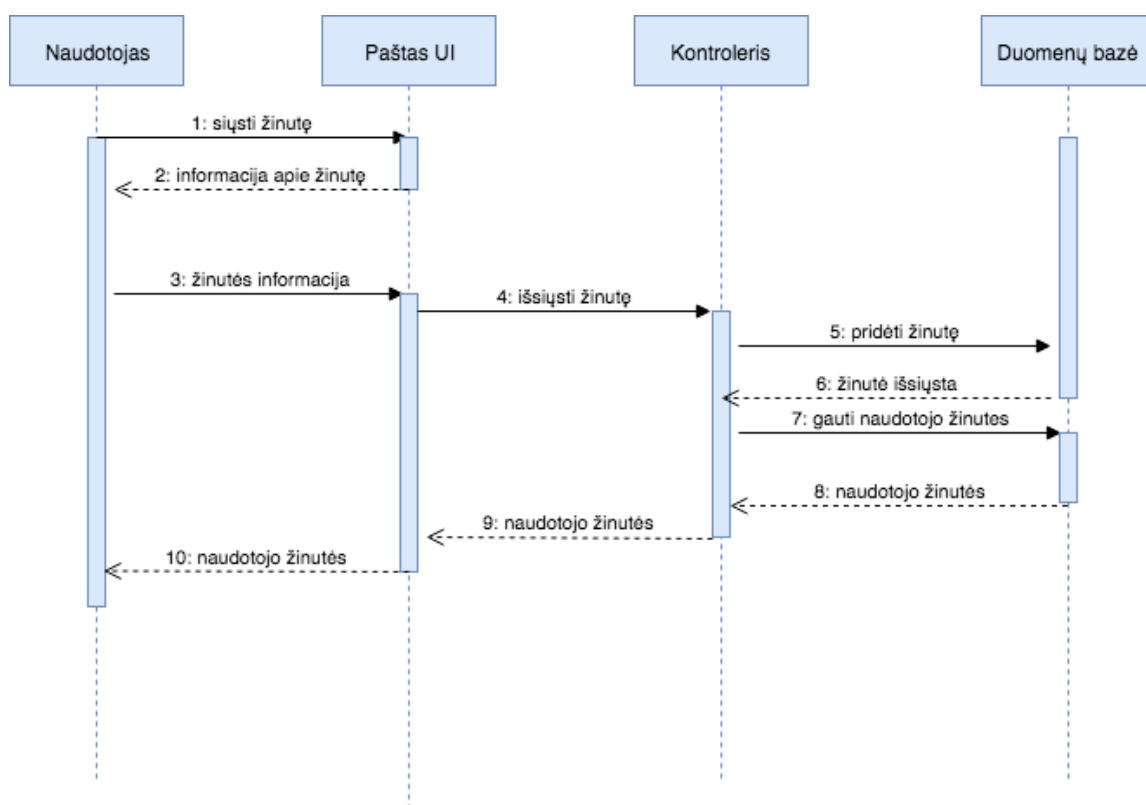


16 pav. Užduoties „Peržiūrėti pasirinktą žinutę“ scenarijus

Žingsnių seka (16 pav.):

1. Peržiūrėti žinutes - naudotojas paspaudžia nuorodą peržiūrėti žinutes.
2. Gauti naudotojo žinutes - prašymas gauti žinutes keliauja iki duomenų bazės.
3. Gauti naudotojo žinutes - prašymas gauti žinutes keliauja iki duomenų bazės.
4. Naudotojo žinutės - žinučių sąrašas keliauja iki naudotojo.
5. Naudotojo žinutės - žinučių sąrašas keliauja iki naudotojo.
6. Pasirinkite žinutę - leidžiama pasirinkti, kurią žinutę norima matyti.
7. Pasirinkta žinutė - pasirenkama žinutė, kurios informacija domina.
8. Rodyti pasirinktos žinutės informaciją - žinutė siunčiama kontrolieriui.
9. Surasti pasirinktą žinutę - Pašto kontrolieris prašo duomenų bazės pateikti žinutę.
10. Pasirinkta žinutė - žinutė keliauja iš duomenų bazės kontrolieriui.
11. Pasirinktos žinutės informacija - žinutė keliauja iš duomenų bazės naudotojui.
12. Pasirinktos žinutės informacija - žinutė keliauja iš duomenų bazės naudotojui.

2.16. Užduoties „Siųsti žinutę“ scenarijus

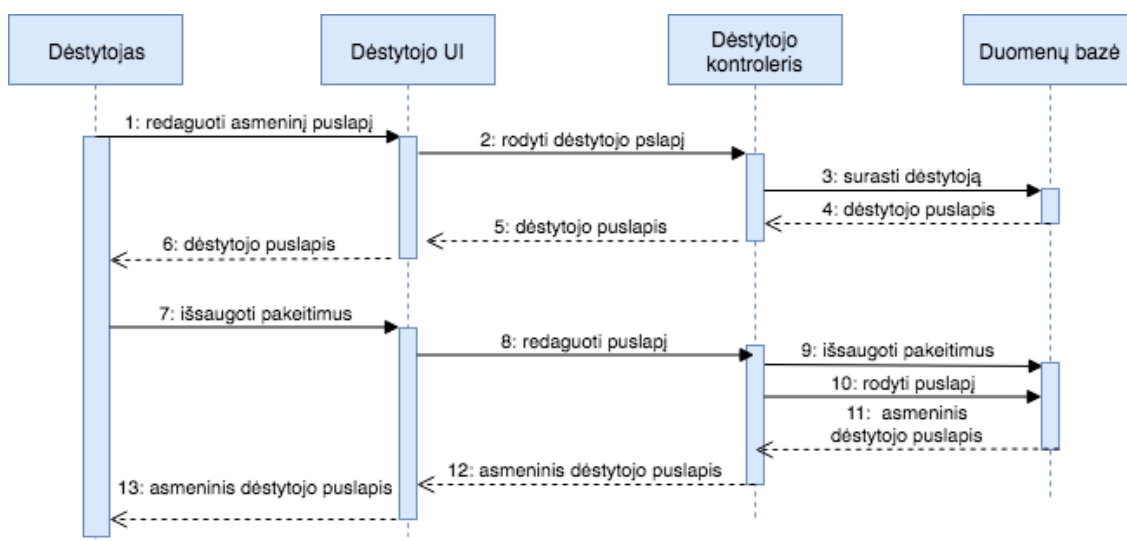


17 pav. Užduoties „Siųsti žinutę“ scenarijus

Žingsnių seka (17 pav.):

1. Siųsti žinutę - naudotojas paspaudžia nuorodą siųsti žinutę.
2. Informacija apie žinutę - pašto UI išmeta naudotojui žinutės formą užpildymui.
3. Žinutės informacija - naudotojas suveda žinutės informaciją ir patvirtina siuntimą.
4. Išsiųsti žinutę - įvesta informacija yra siunčiama kontrolieriui.
5. Pridėti žinutę - pašto UI įdeda žinutę į duomenų bazę.
6. Žinutė išsiųsta - kontrolieris gauna patvirtinimą apie žinutės išsiuntimą.
7. Gauti naudotojo žinutes - kontrolieris prašo duomenų bazės gauti naudotojo žinutes.
8. Naudotojo žinutės - naudotojo žinučių sąrašas keliauja iš duomenų bazės naudotojui.
9. Naudotojo žinutės - naudotojo žinučių sąrašas keliauja iš duomenų bazės naudotojui.
10. Naudotojo žinutės - naudotojo žinučių sąrašas keliauja iš duomenų bazės naudotojui.

2.17. Užduoties „Redaguoti asmeninį dėstytojo puslapį“ scenarijus



18 pav. Užduoties „Redaguoti asmeninį dėstytojo puslapį“ scenarijus

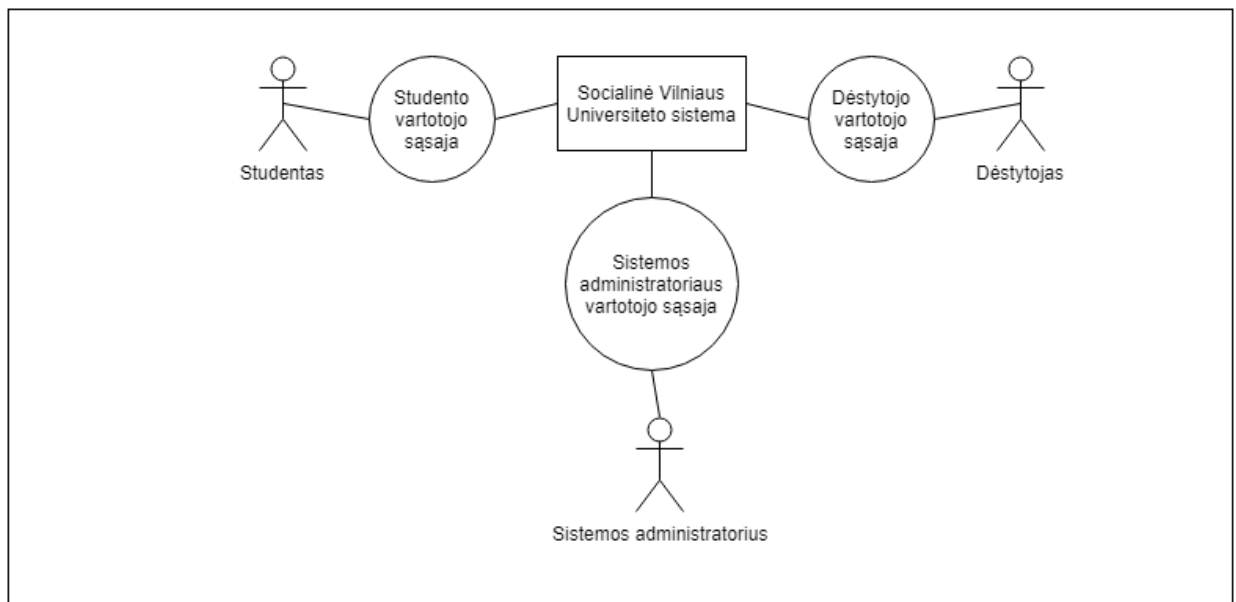
Žingsnių seka (18 pav.):

1. Redaguoti asmeninį puslapį - dėstytojas paspaudžia nuorodą redaguoti asmeninį puslapį.
2. Rodyti dėstytojo puslapį - prašymas rodyti puslapį keliauja į kontrolierį.
3. Surasti dėstytoją - kontrolieris pateikia prašymą duomenų bazei surasti prisijungusį dėstytoją.
4. Dėstytojo puslapis - rasto dėstytojo puslapio informacija keliauja dėstytojui.
5. Dėstytojo puslapis - rasto dėstytojo puslapio informacija keliauja dėstytojui.
6. Dėstytojo puslapis - rasto dėstytojo puslapio informacija keliauja dėstytojui.
7. Išsaugoti pakeitimus - atlikęs pakeitimus savo puslapyje, dėstytojas paspaudžia nuorodą išsaugoti.
8. Redaguoti puslapį - įvesta informacija yra siunčiama kontrolieriui.
9. Išsaugoti pakeitimus - dėstytojo kontrolieris įdeda pakeistą informaciją į duomenų bazę.
10. Rodyti puslapį - kontrolieris prašo duomenų bazės gauti redaguotą dėstytojo puslapio informaciją.
11. Asmeninis dėstytojo puslapis - dėstytojo puslapio informacija keliauja iš duomenų bazės dėstytojui.
12. Asmeninis dėstytojo puslapis - dėstytojo puslapio informacija keliauja iš duomenų bazės dėstytojui.
13. Asmeninis dėstytojo puslapis - dėstytojo puslapio informacija keliauja iš duomenų bazės dėstytojui.

3. KŪRIMO PJŪVIS

Programų sistemos komponentai yra vaizduojami trimis lygmenimis: nuliniu, pirmuoju ir antruoju. Toks komponentų pateikimas leidžia išsamiau apibrėžti sistemos fizinius komponentus, jų konfigūraciją bei tarpusavio ryšius. Komponentų diagramos, atvaizduodamos struktūrą, priklausomybes bei sąsajas, leidžia susidaryti fizinį sistemos vaizdą. Taip pat suteikia galimybę apžvelgti išoriškai matomą komponentų elgseną. Komponentai atvaizduojami naudojant UML komponentų diagramas.

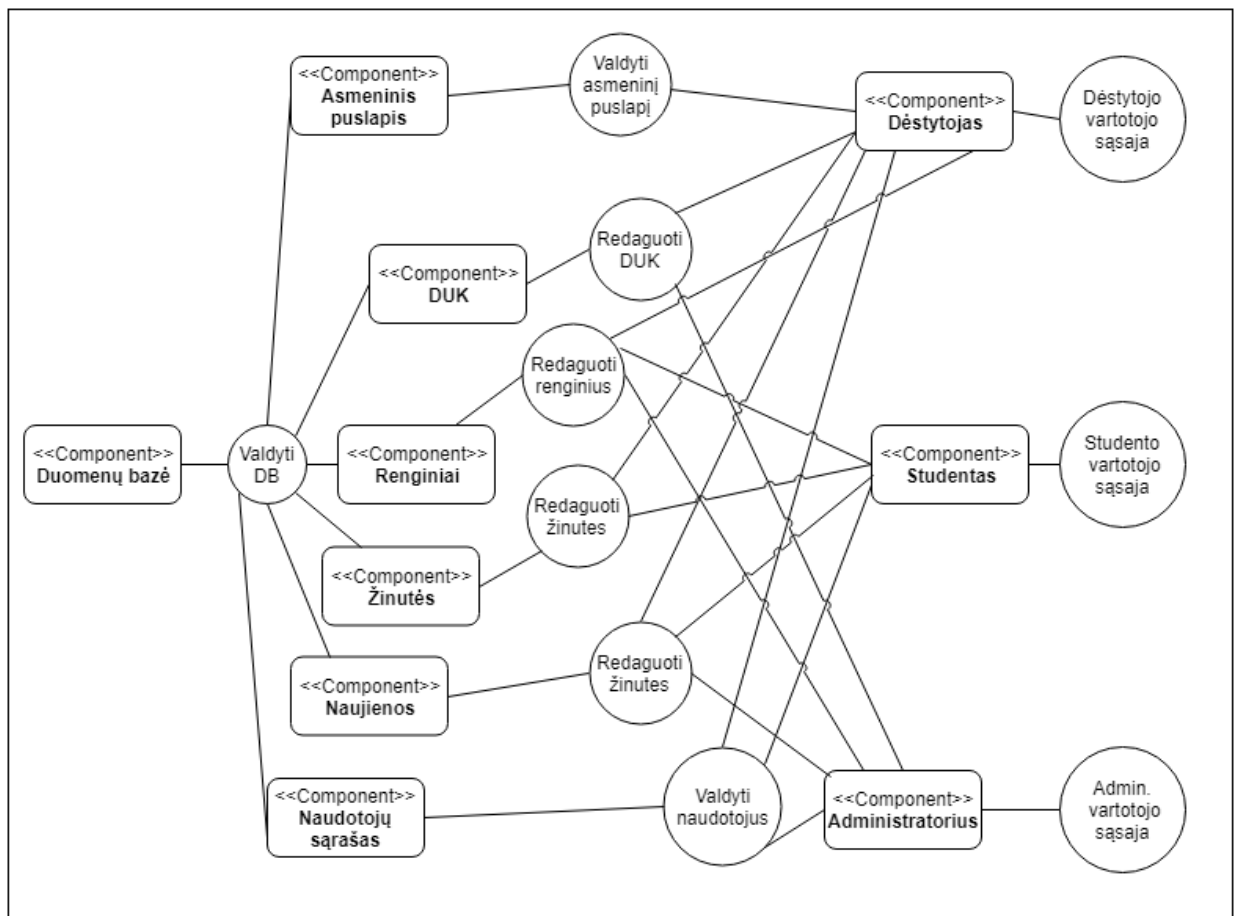
3.1. Komponentų diagramos nulinis lygmuo



19 pav. Komponentų diagramos nulinis lygmuo

Komponentų diagramos nuliniame lygmenyje (19 pav.) vaizduojamas bendras komponentų vaizdas. Pagrindinis ir vienintelis šio lygio komponentas yra „Socialinė Vilniaus Universiteto sistema“. Šis komponentas sąveikauja su keliomis vartotojo sąsajomis. Studento ir dėstytojo grafinė vartotojo sąsaja įgalina šiuos vartotojus naudotis sistema.

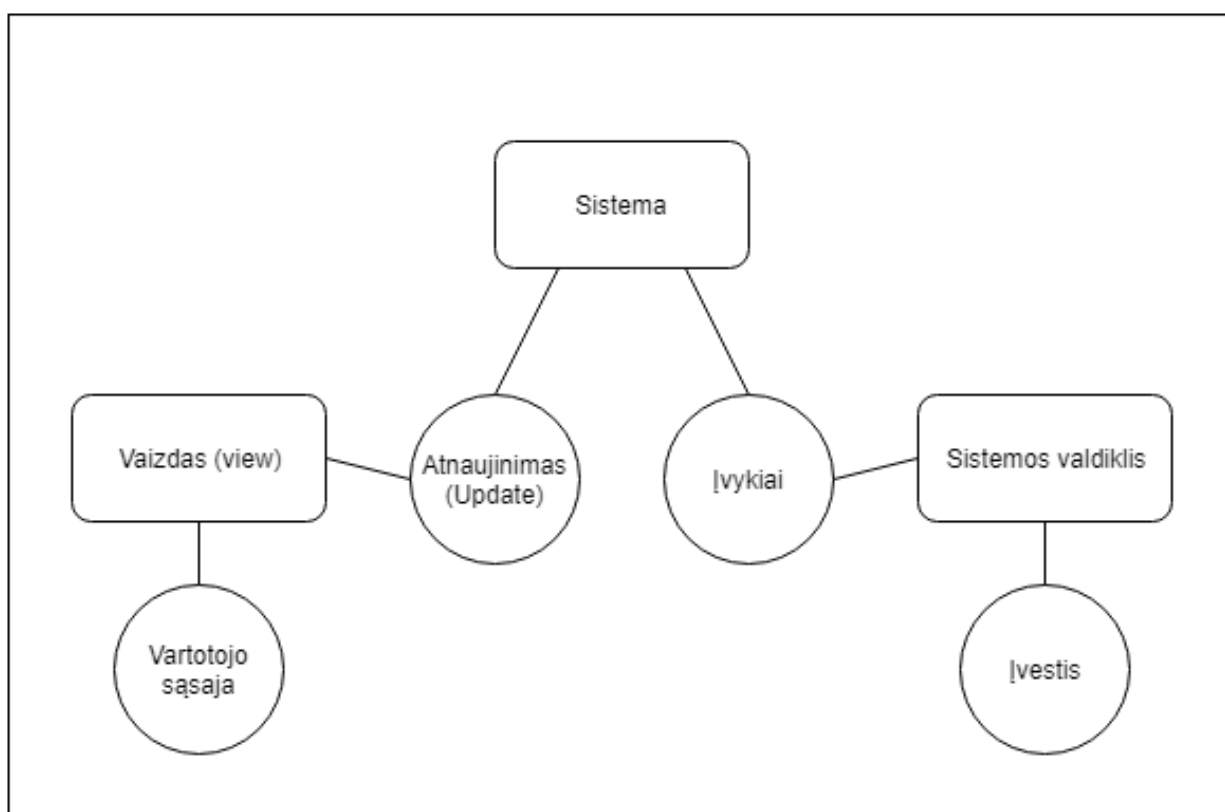
3.2. Komponentų diagramos pirmasis lygmuo



20 pav. Komponentų diagramos pirmasis lygmuo

Komponentų diagramos pirmame lygmenyje komponentų diagrama (20 pav.) yra suskaidoma. Socialinė Vilniaus universiteto sistema yra suskaidoma į šiuos komponentus: Duomenų bazė, Asmeninis puslapis, DUK, Renginiai, Žinutės, Naujienos, Naudotojų sąrašas. Šiame lygyje kiekviena vartotojo sąsaja turi už ją atsakingus komponentus. Taip pat kiekvienas komponentas turi sąsajas su kitais komponentais tam, jog galėtų vykdyti sąveiką ir keitimasis paslaugomis. Tokiu būdu yra užtikrinama visapusiška komponentų realizacija bei tarpusavio darna.

3.3. Komponentų diagramos antrasis lygmuo



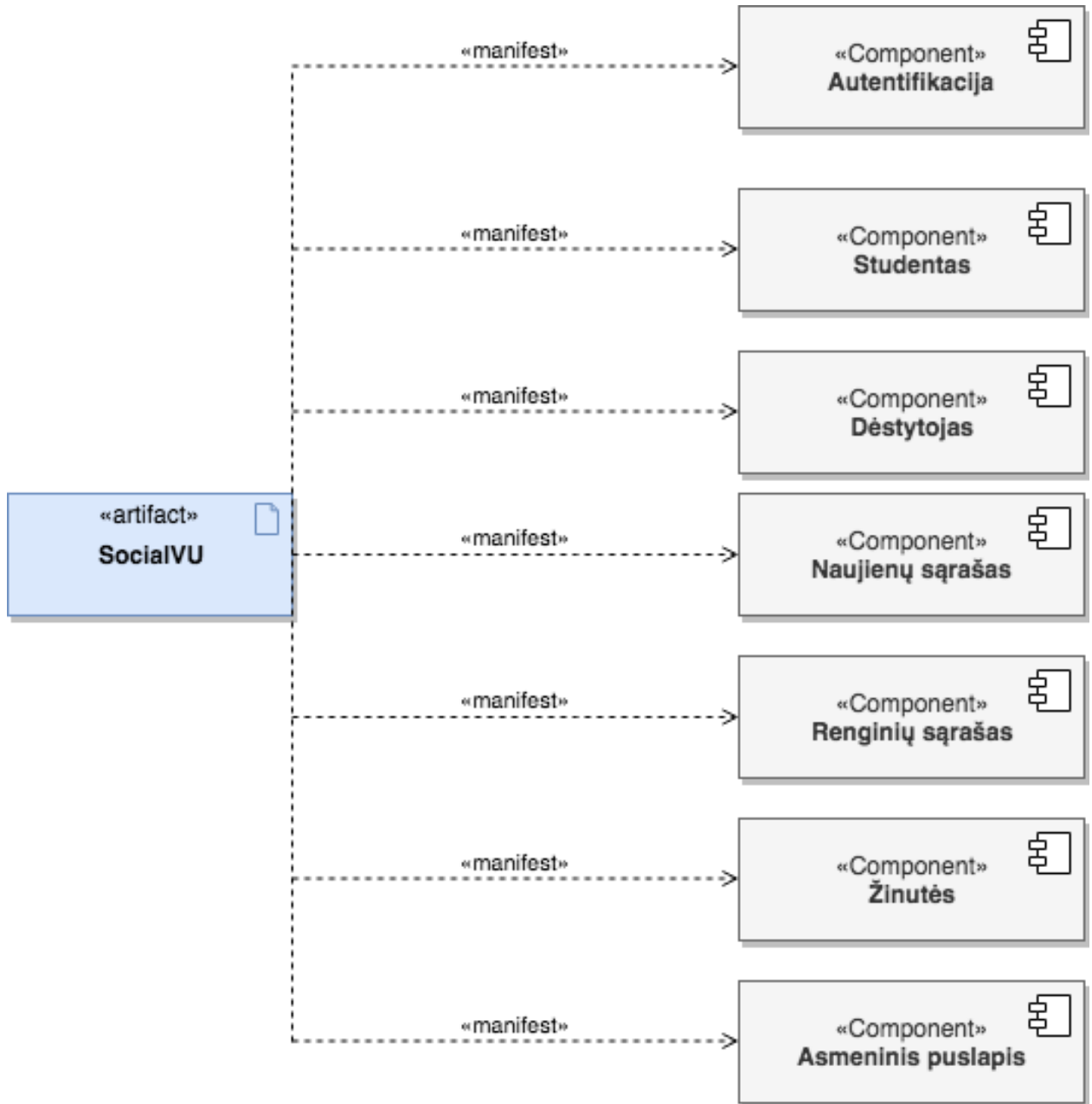
21 pav. Komponentų diagramos antrasis lygmuo

Antrame lygmenyje dekomponavimui buvo pasitelktas MVC dizaino šablonas. Šis modelis buvo pasirinktas dėl jo paprastumo, universalumo ir populiarumo. MVC šabloną sudaro trys pagrindiniai komponentai: Model, View ir Controller. Model (šiuo atveju, mūsų Sistema) – pagrindinis šablono komponentas, jis atsakingas už visos sistemos elgesį probleminėje situacijoje, nepriklausomai nuo vartotojo sąsajos, taip pat jis atsakingas už duomenis, logiką ir taisykles. Vaizdas (view) – komponentas, kuris yra atsakingas už informacijos atvaizdavimą, jos atnaujinimą. Valdiklis – komponentas, kuris rūpinasi duomenų įvestimi ir įvesties apdorojimu (21 pav.).

4. FIZINIS PJŪVIS

Fizinis pjūvis sudarytas iš dislokavimo diagramų. Šiose diagramose vaizduojamas programos komponentų išdėstymas tinkle bei komunikacijos protokolai tarp jų.

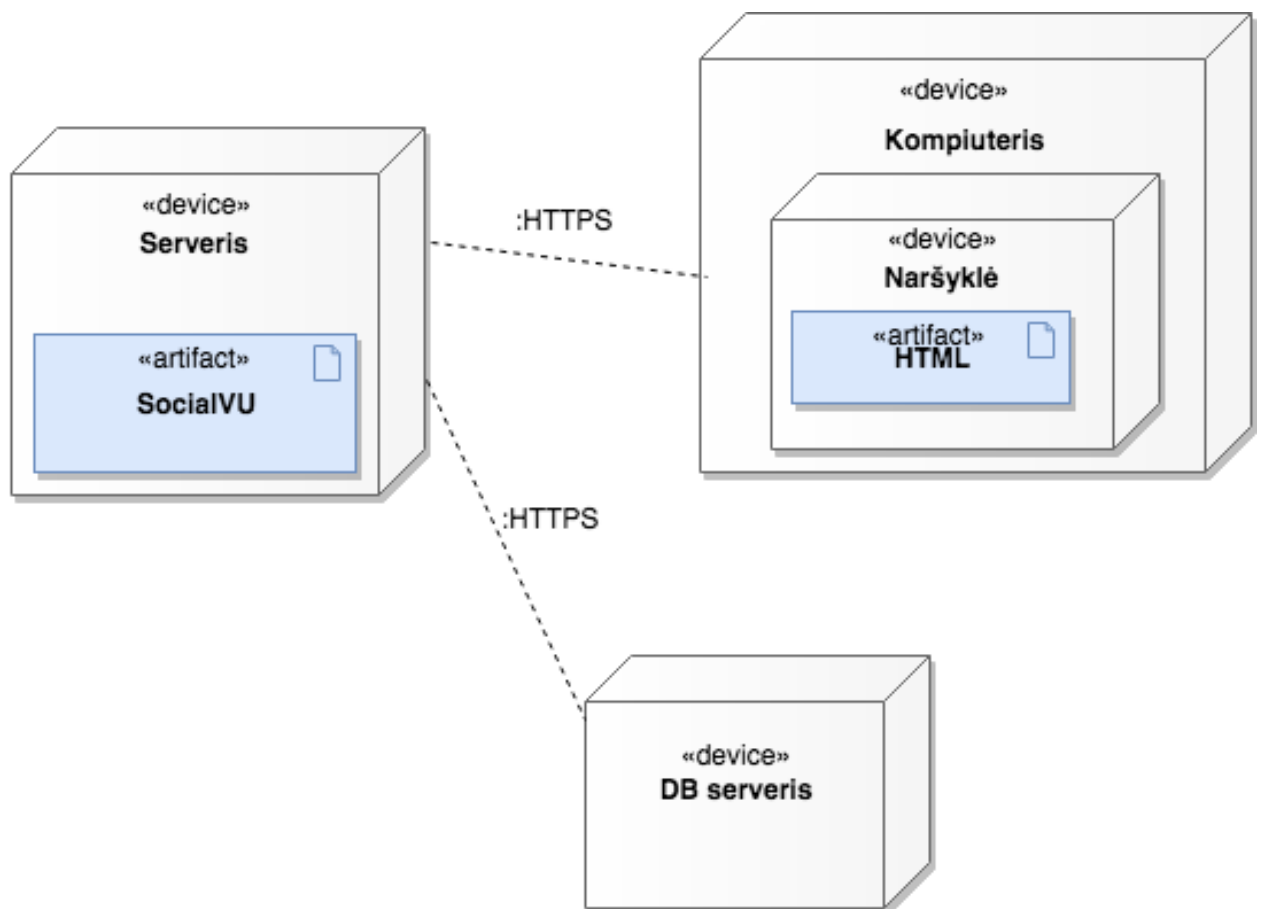
4.1. Dislokavimo diagrama nr. 1 (komponentų ir artefaktų ryšių diagrama)



22 pav. Komponentų ir artefaktų ryšių diagrama

Komponentų ir artefaktų ryšių diagramoje (22 pav.) vaizduojamas artefaktas SocialVU, kuris įgyvendina šiuos programos komponentus: Autentifikacija, Studentas, Dėstytojas, Naujienų sąrašas, Renginių sąrašas, Žinutės, Asmeninis puslapis.

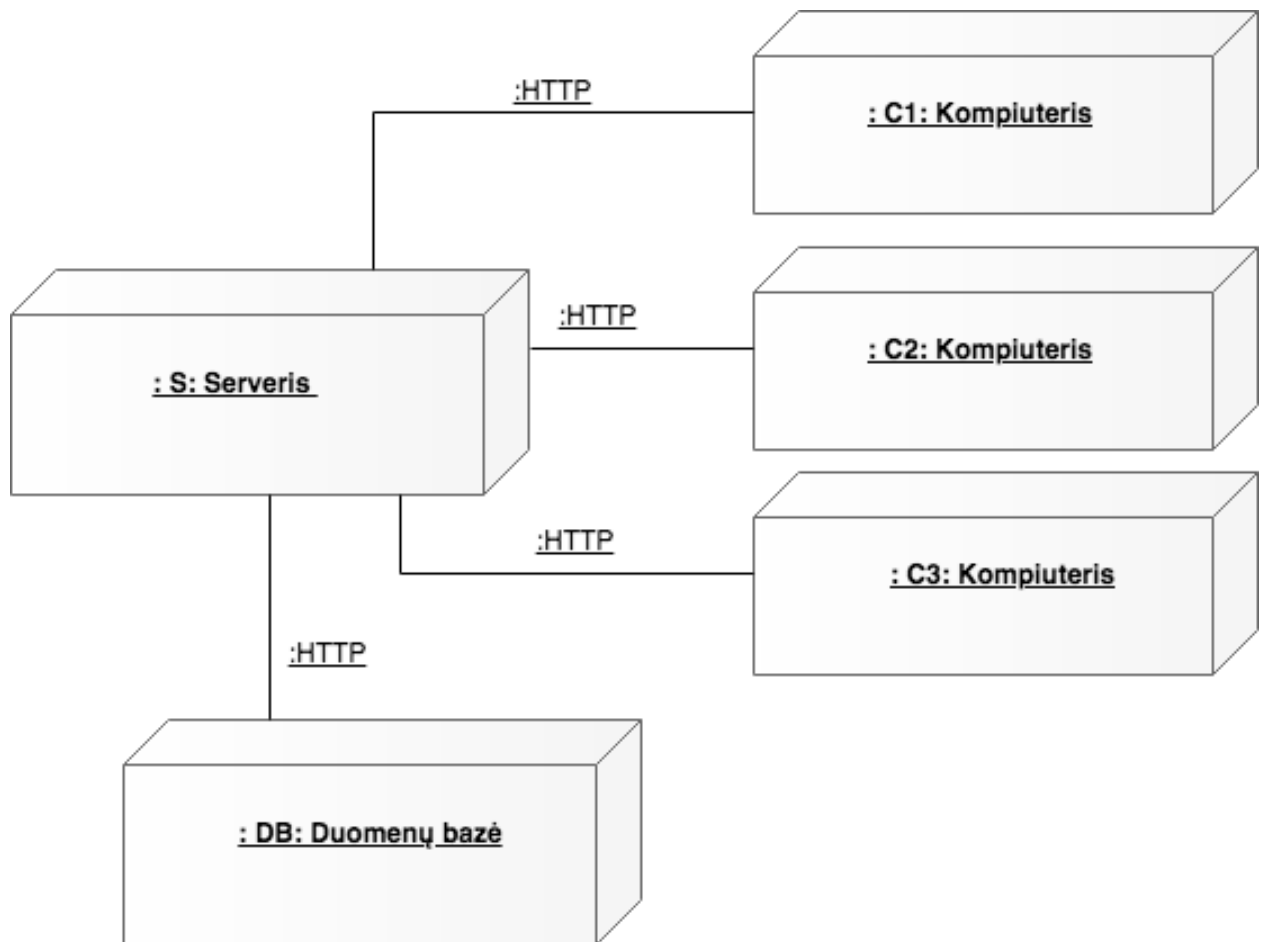
4.2. Dislokavimo diagrama nr. 2 (mazgų ir artefaktų ryšių diagrama)



23 pav. Mazgų ir artefaktų ryšių diagrama

Mazgų ir artefaktų diagramoje (23 pav.) parodo, kad tinklalapis yra serveryje, kuris bendrauja http protokolu su SQL serveriu, kuriame saugoma duomenų bazė. Naudotojai turi galimybę pasiekti sistemos teikiamas paslaugas savo pasirinkta naršykle, kuri palaiko http protokolą.

4.3. Dislokavimo diagrama nr. 3 (mazgų ir artefaktų egzempliorių diagrama)



24 pav. Mazgų ir artefaktų egzempliorių diagrama

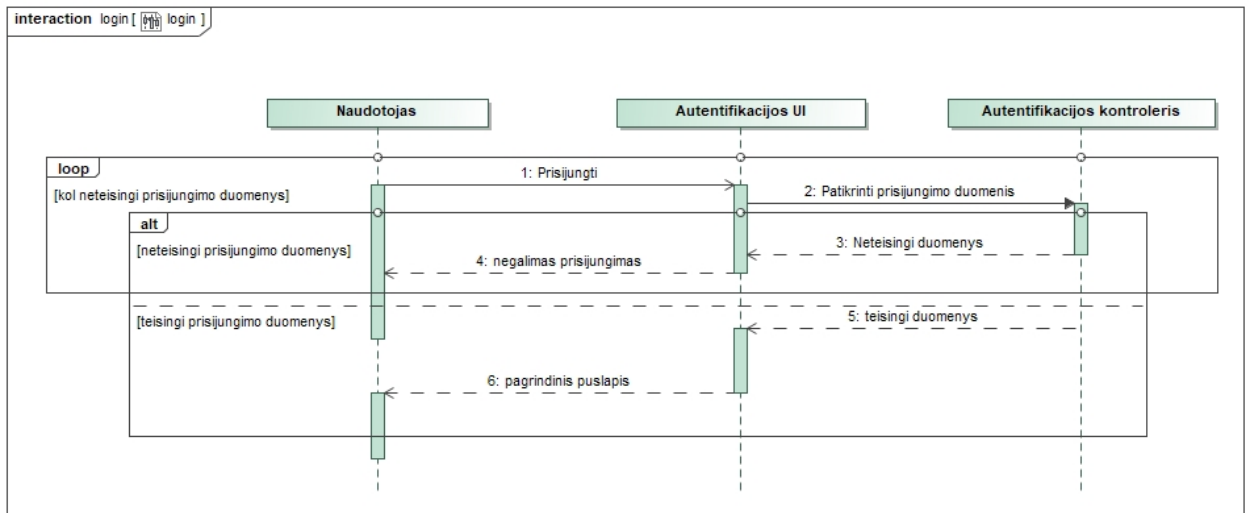
24 pav. vaizduojamas įrenginių (mazgų) išsidėstymas tinkle. Serveris turi tiesioginį ryšį su duomenų baze, o naudotojai gali prisijungti prie serverio. Tačiau naudotojai negali tiesiogiai pasiekti duomenų bazės ir joje saugomų duomenų.

5. PROCESO PJŪVIS

Procesų pjūvis sudarytas iš sekų ir veiklos diagramų. Diagramose parodoma, kokie procesai vyksta sistemoje bei išreiškiama komunikacija tarp jų.

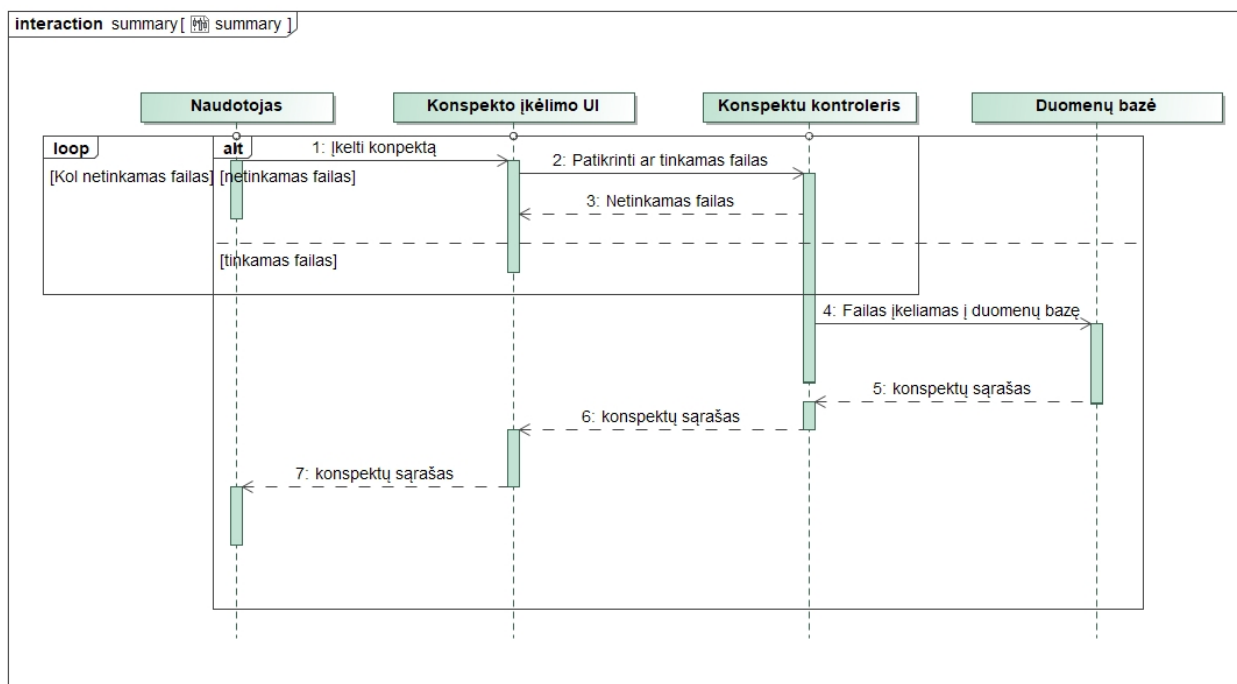
5.1. Proceso sekų diagramos

Procesų sekų diagramose, atspindi procesai, kurie yra vykdomi sistemoje. Iš proceso sekų diagramos galima matyti, kokie komponentai dalyvauja vykdyme, kaip procesas vykdomas.



25 pav. Proceso „Prisijungimas“ sekų diagrama

Pagal 25 pav. diagramą matoma, kad procesas prasideda naudotojo paspaudimu ant nuorodos įgalinančios prisijungimą. Autentifikacijos UI gautus duomenis siunčia patikrinimui į autentifikacijos kontrolierį. Iš kontrolierio gaunamas atsakymas, ar duomenys teisingi ar ne. Jei duomenys klaidingi, naudotojui išmetamas pranešimas, jog prisijungti negalima ir jis vėl gali kartoti prisijungimo procesą. Jei duomenys teisingi, naudotojas yra nukreipiamas į pagrindinį puslapį.

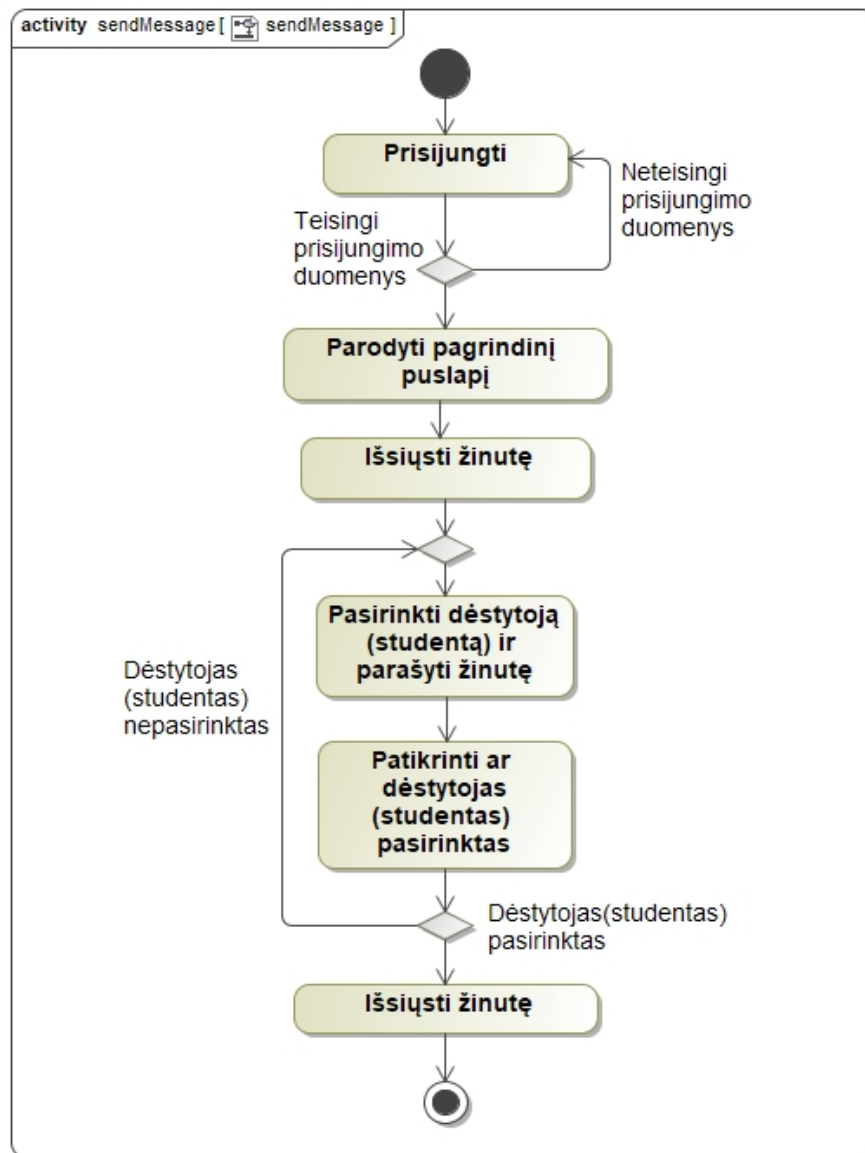


26 pav. Proceso „Konspekto įkėlimas“ sekų diagrama

Pagal 26 pav. diagramą matoma, kad procesas prasideda naudotojo paspaudimu ant nuorodos įgalinančios konspekto įkėlimą. Konspekto įkėlimo UI gautus duomenis siunčia patikrinimui į konspektų kontrolierį. Iš kontrolierio gaunamas atsakymas, ar failas tinkamas ar ne. Jei failas netinkamas, naudotojui išmetamas pranešimas, jog failas netinkamas ir jis vėl gali kartoti konspekto įkėlimo procesą. Jei failas tinkamas, jis yra įkeliamas į duomenų bazę, o naudotojas yra nukreipiamas į konspektų puslapį.

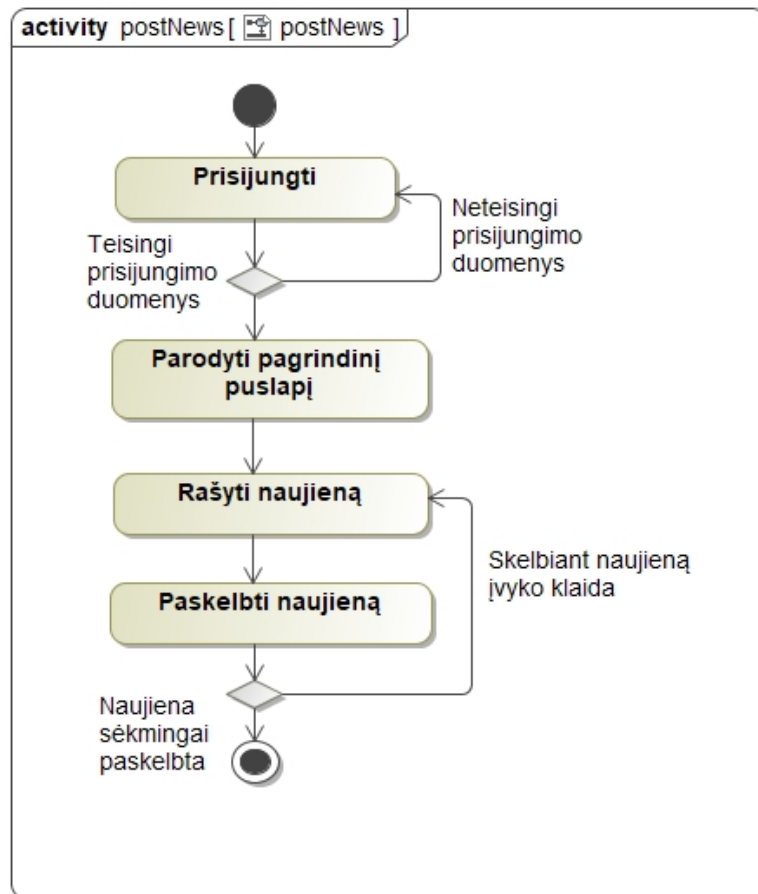
5.2. Veiklos diagramos

Veiklos diagramos padeda suprasti dinaminį sistemos veikimą, parodo, kokie veiksmai atliekami vykdant konkrečią veiklą, galimus vykdymo atvejus.



27 pav. Žinutės išsiuntimo veiklos diagrama

27 pav. diagramoje matomas, žinutės išsiuntimo dėstytojui procesas. Procesas prasideda naudotojo nuorodos paspaudimu, kreipiančios į žinutės išsiuntimo formą. Naudotojas pateikia parašo žinutę bei pasirenka dėstytoją, tuomet duomenys siunčiami patikrinimui. Jei dėstytojas nebuvo pasirinktas, naudotojas nukreipiamas atgal į žinutės siuntimo formą. Jei duomenys atitinka visus reikalavimus, tada žinutė išsiunčiama.



28 pav. Naujienos paskelbimo veiklos diagrama

28 pav. diagramoje matomas, naujienos paskelbimo procesas. Procesas prasideda naudotojo prisijungimu. Neteisingai suvedus prisijungimo duomenis naudotojas vėl nukreipiamas į prisijungimą, kitu atveju jis nukreipiamas į pagrindinį puslapį bei pasirenka naujienos paskelbimo nuorodą. Naudotojas parašo naujieną bei ją paskelbia. Jei skelbiant naujieną įvyksta klaida jis nukreipiamas į naujienos rašymo formą, kitu atveju naujiena paskelbiama.

REZULTATAI

Kuriamos sistemos architektūra išnagrinėta naudojant UML 4+1 požiūrių rinkinį. Iš loginio, užduočių, kūrimo, fizinio bei procesų pjūvių matome sistemos funkcionalumą, užduotis, kurias gali įgyvendinti naudotojas, sąsajas tarp atskirų sistemos komponentų, sistemos įrangą bei dinaminį sistemos modelį.