|  |  |  |
| --- | --- | --- |
| VILNIAUS KOLEGIJA  ELEKTRONIS IR INFORMATIKOS FAKULETETAS  PROGRAMINĖS ĮRANGOS KATEDRA | | |
| 1 | | |
| PROGRAMŲ SISTEMŲ INŽINERIJA | | |
| DOKUMENTACIJA  SD - 6531BX028 PI17B | | |
| STUDENTAS | (PARAŠAS)  2019-12-13 | Lukas KAŠĖTA  Domantas MAKŪNAS  Vaidotas GRUŽAS |
| DĖSTYTOJA | (PARAŠAS)  2019-12-13 | Lekt. J.VAIČIULYTĖ |
|  |  |  |
| 2019 | | |

TURINYS

[ĮVADAS 4](#_Toc27126449)

[TRUMPAS PROJEKTO PLANO APRAŠYMAS 5](#_Toc27126450)

[FUNKCINIAI, NEFUNKCINIAI REIKALAVIMAI 5](#_Toc27126451)

[ARCHITEKTŪRA / DIZAINAS 6](#_Toc27126452)

[PS REALIZAVIMAS 13](#_Toc27126453)

[TESTAVIMO PLANAS 14](#_Toc27126454)

[VARTOTOJO INSTRUKCIJA 16](#_Toc27126455)

[IŠVADOS 25](#_Toc27126456)

OBJEKTŲ TURINYS

Paveikslėliai

[1pav. Sprinto pavyzdys 5](#_Toc27126375)

[2pav. UML diagrama 7](#_Toc27126376)

[3pav. Klasių diagrama 8](#_Toc27126377)

[4pav. Vartotojo būsenos diagrama 9](#_Toc27126378)

[5pav. Administratoriaus būsenos diagrama 9](#_Toc27126379)

[6pav. Vartotojo ištrinimo veiksmų diagrama 10](#_Toc27126380)

[7pav. Rolės pakeitimo veiksmų diagrama 10](#_Toc27126381)

[8pav. Užrašinės atvaizdavimo veiksmų diagrama 11](#_Toc27126382)

[9pav. Vartotojo sąrašo išvedimo veiksmų diagrama 11](#_Toc27126383)

[10pav. Užrašinės sukūrimo veiksmų diagrama 11](#_Toc27126384)

[11pav.Užrašo ištrynimo veiksmų diagrama 12](#_Toc27126385)

[12pav. Užrašinės veiksmų diagrama 13](#_Toc27126386)

[13 pav. Testavimo sceanrijai 14](#_Toc27126387)

[14 pav. Testavimo scenarijai 2 15](#_Toc27126388)

[15 pav. Testavimo scenarijų aprašymai 15](#_Toc27126389)

[16pav. Sistemos pradinio lango pavyzdys 16](#_Toc27126390)

[17pav. Registracijos forma. 16](#_Toc27126391)

[18pav. Sėkmingos registracijos langas 16](#_Toc27126392)

[19pav. Prisijungimo ekranas 17](#_Toc27126393)

[20pav. Sistemos funkcijos 17](#_Toc27126394)

[21pav. Naujos užrašinės sukūrimas 18](#_Toc27126395)

[22pav. Užrašinės teksto išsaugojimas 18](#_Toc27126396)

[23pav. Antros užrašinės pavyzdys 19](#_Toc27126397)

[24pav. Trečios užrašinės pavyzdys 19](#_Toc27126398)

[25 pav. Trinimo pavyzdys 20](#_Toc27126399)

[26pav*.* Pašalinta užrašinė 21](#_Toc27126400)

[27pav*.* Administratoriaus prisijungimas 21](#_Toc27126401)

[28pav. Atvaizdavimo panelė 22](#_Toc27126402)

[29pav. Administratoriaus teisių atimimas 22](#_Toc27126403)

[30 pav. Administratoriaus teisių suteikimas 23](#_Toc27126404)

[31pav. Vartotojo trynimas 23](#_Toc27126405)

[32pav. Ištrintas vartotojas 24](#_Toc27126406)

[33pav. Vartotojo sugražinimas 24](#_Toc27126407)

**Lentelės**

[1lentelė. Architektūros sluoksnių pavyzdys 6](#_Toc27123897)

ĮVADAS

**Kam skirta kuriama sistema:** kuriama programų sistema , bus skirta asmenims, kuriems reikia ne tik paprastos užrašinės, kurioje galima užsirašyti informaciją, bet ir šią informaciją apsaugoti nuo kitų vartotojų naudojant registracijos ir prisijungimo funkcijas. Būtent šios dvi funkcijos padaro šią užrašinę unikalią, kadangi užrašinė veiks tarsi privatus dienoraštis, kuriame vartotojų informacija bus prieinama tik pačiam vartotojui ir niekam kitam.

**Būsimi vartotojai:** Vartotojas šios programų sistemos gali būti bet kas, kam reikia šiek tie labiau privačios užrašinės.

**Paskirtis:** Šios programų sistemos vienintelis tikslas yra saugoti vartotojų informaciją privačiai, jog tik pačiam vartotojui būtų prieinama užrašinės informacija ir be prisijungimo duomenų, niekas jos negalėtų peržiūrėti.

**Patalpinto projekto nuorodą:** <https://github.com/vgruzas/PSI_pietu_klubas>

TRUMPAS PROJEKTO PLANO APRAŠYMAS

Projektas buvo pradėtas spalio 1d. Išsaiškinus sistemos funkcinius ir nefunkcinius reikalavimus spalio 4 d. buvo pradėtas programavimo procesas, baigtas programuoti gruodžio 11 d.

A screenshot of a cell phone

Description generated with very high confidence

1pav. Sprinto pavyzdys

FUNKCINIAI, NEFUNKCINIAI REIKALAVIMAI

**Funkciniai reikalavimai:**

1. Sukurti prisijungimą prie sistemos
2. Sukurti ir suteikti galimybę prisiregistruoti prie sistemos.
3. Sukurti ir suteikti vartotojui privilegijas kurti užrašinių puslapius.
4. Sukurti ir suteikti vartotojui privilegijas trinti užrašinių puslapius.
5. Sukurti ir suteikti vartotojui privilegijas redaguoti puslapių turinį.
6. Sukurti ir suteikti administratoriui privilegijas kurti naujus vartotojus.
7. Sukurti ir suteikti administratoriui privilegijas redaguoti vartotojus.

**Nefunkciniai reikalavimai:**

1. Sistema turi veikti korektiškai.
2. Sistema turi atitikti kliento lūkesčius.
3. Sistemos turinys neturi įžeisti naudotojo.
4. Sistema turi atitikti iš politinių apribojimų išplaukiančius reikalavimus.
5. Sistema turi atitikti iš ekonominių apribojimų išplaukiančius reikalavimus.
6. Sistema turi būti pritaikyta Windows 7 arba naujesniai operaciniai sistemai.

ARCHITEKTŪRA / DIZAINAS

**Architektūros aprašymas:**

1. **Našumas (Performance)**

Sistemoje galima vidinius galima lengvai modifikuoti daug kartų, nes naudojamas interfeisas. Viskas parašyta pagal „SOLID“ principus, kas padaro sistemą lengvai suprantamą ir plečiamą.

1. **Saugumas (Security)**

Naudojama sluoksninė architektūra, kur svarbiausi dalykai yra prieinami tik žemiausiame lygyje ir ne visiems vartotojams.

1. **Saugumas (Safety)**

Sistemos saugumas yra įgyvendinta inkapsuliacija, dėl to kiekviena programos dalis nėra laisvai prieinama, nesuteikus teisių.

1. **Prieinamumas(Availability)**

Sistema suprojektuota taip, kad tam tikras jos dalis keičiant reikėtų sustabdyti programą.

1. **Palaikomumas (Maintainability)**

Naudojamus komponentus galima nesunkiai pakeisti kitais, kaip ir programos grafinę sąsają.

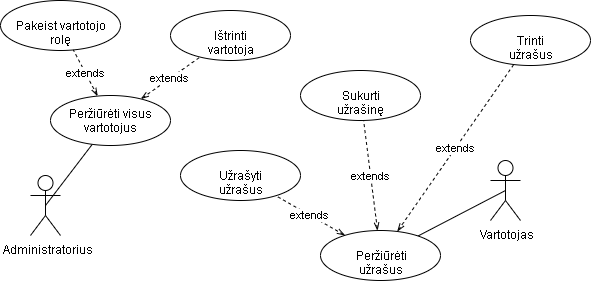
1. **Architekturos sluoksniai**

|  |
| --- |
| Duomenų bazė |
| Modeliai |
| Funkcionalumass |

1lentelė. Architektūros sluoksnių pavyzdys

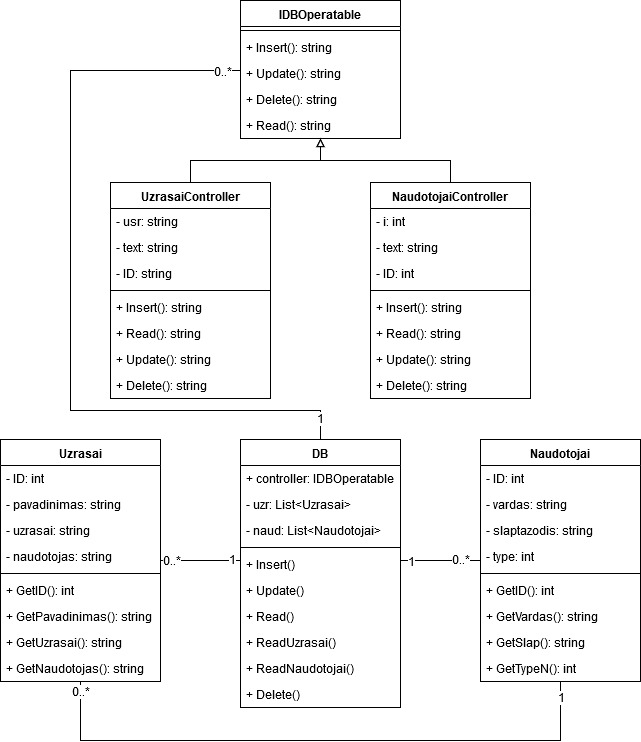
Paveikslėlyje (žr.2pav) galima matyti sistemos UML diagramą. Šioje diagramoje, vaizduojama, kokie veikėjai bus kuriamoje sistemoje bei ką kiekvienas veikėjas galės atlikti naudojant šią sistemą.

**Sistemos modeliai:**



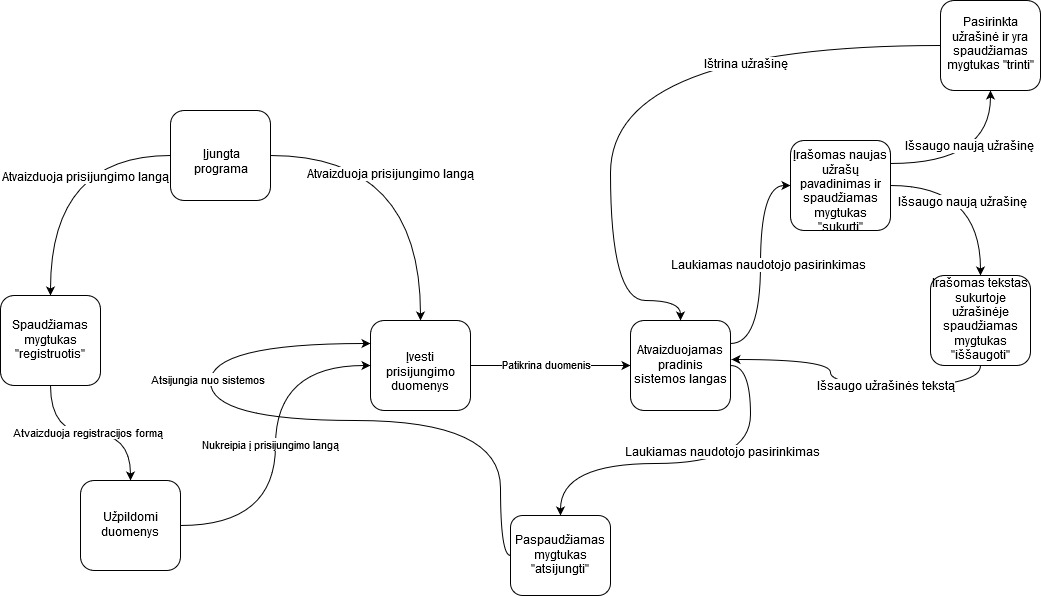
2pav. UML diagrama

Toliau yra pavaizduotą sistemos klasių diagrama (žr.3pav), kuri šiek tiek detaliau parodo sistemos funkcionalumą bei parodo kokios bus klasės bei kuri klasė su kokia klasę bendraus bei dalinsis duomenimis.

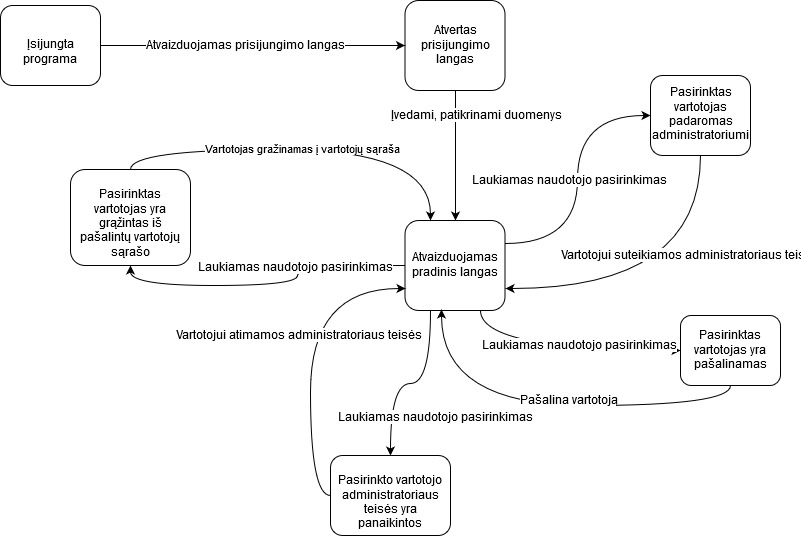


3pav. Klasių diagrama

Paveikslėliuose (žr..4,5pav) galima matyti būsenos diagramas. Pirmoji diagrama yra skirta parodyti visą funkcionalumą iš vartotojo pusės, ką jis gali padaryti ir kaip jis tai padaro, o antrojoje diagramoje yra parodytas panašus procesas, tačiau iš administratoriaus puses su visomis jo funkcijomis.

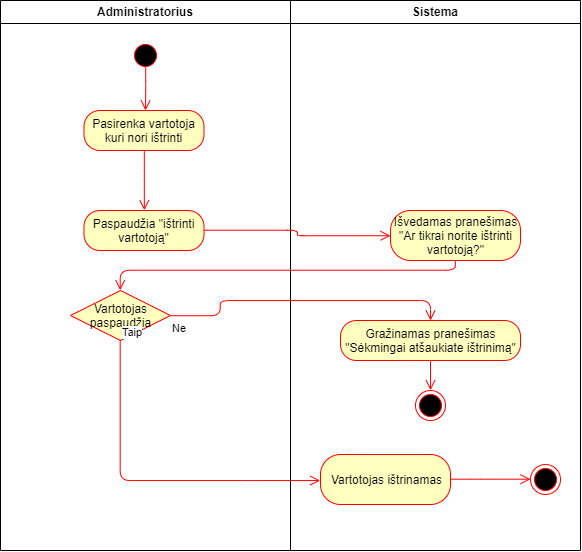


4pav. Vartotojo būsenos diagrama

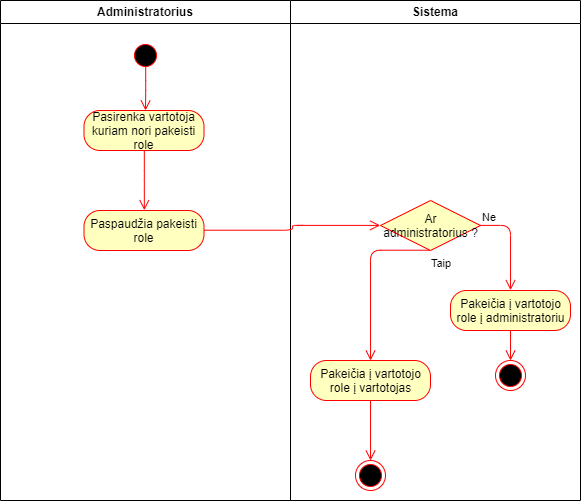


5pav. Administratoriaus būsenos diagrama

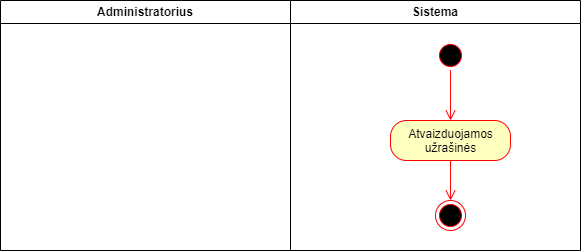
Paveikslėliuose (žr.6-12pav) yra pavaizduotos sistemos veiksmų diagramos. Šios diagramos detaliau aprašo vidinius procesus, kurie įvyksta kuriamoje sistemoje. Šiose diagramose yra aprašytas kiekvienas funkcijos metodas, kuris buvo atvaizduotas UML diagramoje (žr.2pav).



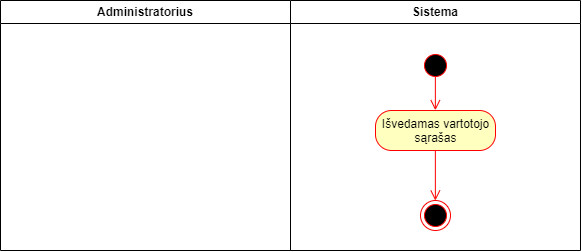
6pav. Vartotojo ištrinimo veiksmų diagrama



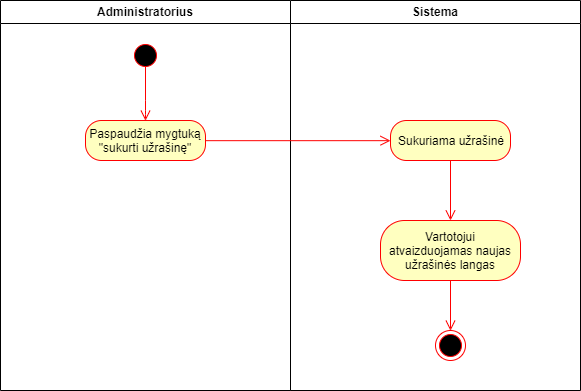
7pav. Rolės pakeitimo veiksmų diagrama



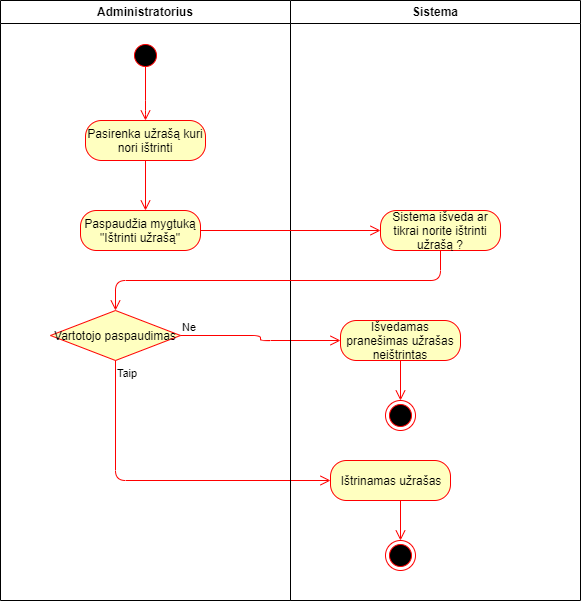
8pav. Užrašinės atvaizdavimo veiksmų diagrama



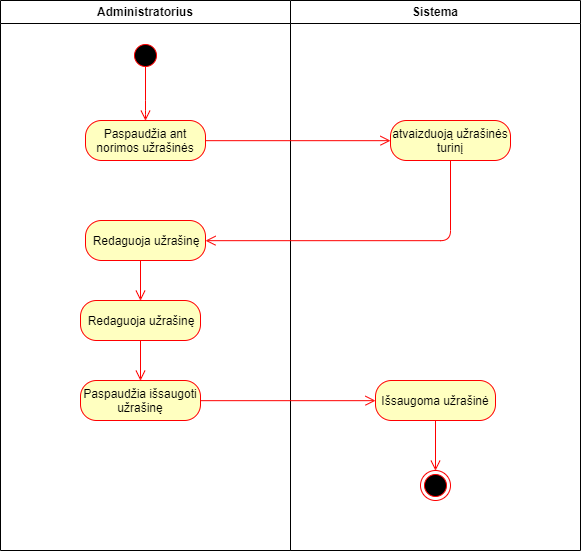
9pav. Vartotojo sąrašo išvedimo veiksmų diagrama



10pav. Užrašinės sukūrimo veiksmų diagrama



11pav.Užrašo ištrynimo veiksmų diagrama



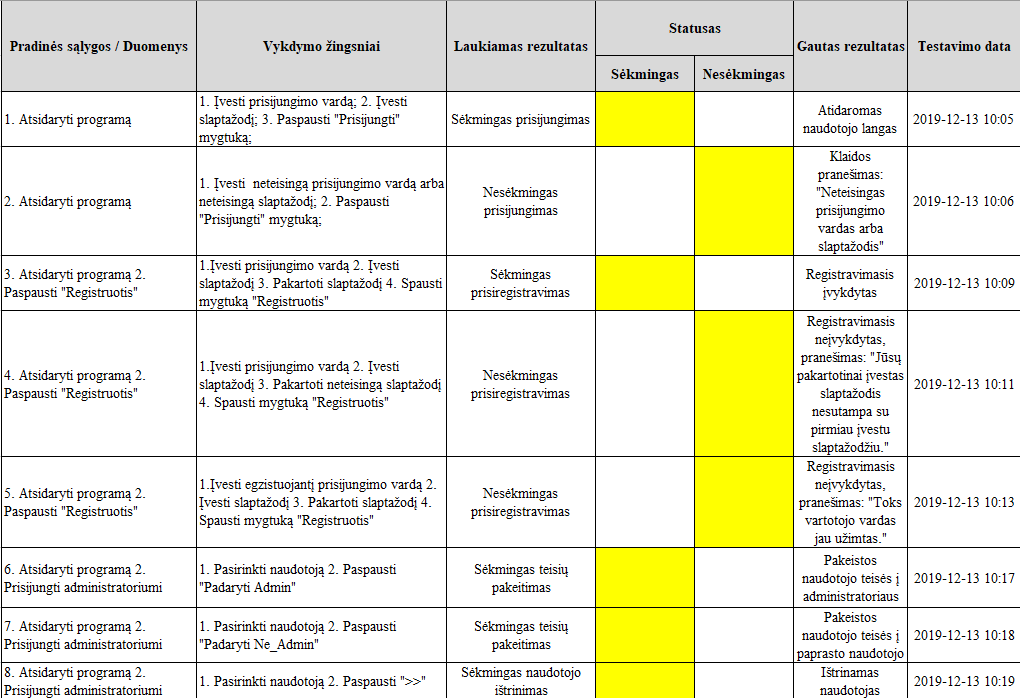
12pav. Užrašinės veiksmų diagrama

PS REALIZAVIMAS

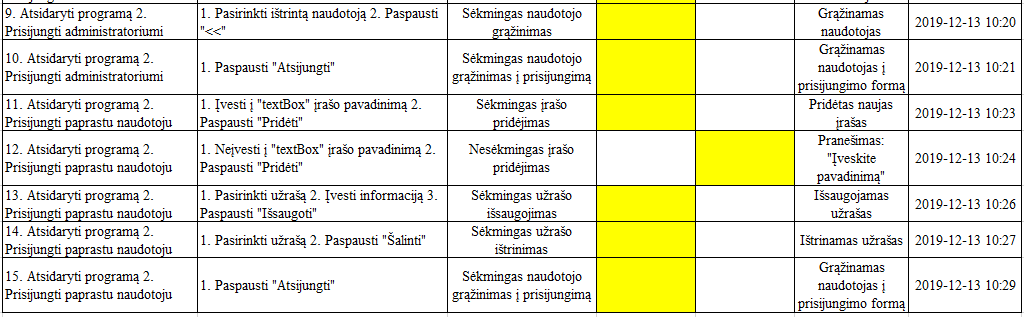
Šiai sistemai realizuoti buvo naudojami tik Microsoft produktai. Visas programavimas buvo vykdomas „Microsoft Visual Studio 2019“ programų kūrimo įrankiu, joje naudojama programavimo kalba buvo C# ir buvo kuriama „Windows Form“ aplikacija, naudojanti .NET karkasą. Programoje naudojama MSSQL duomenų bazė, kuri taip pat yra pilnai palaikoma „Visual Studio“ programoje, „mdf“ failo formatu.

TESTAVIMO PLANAS

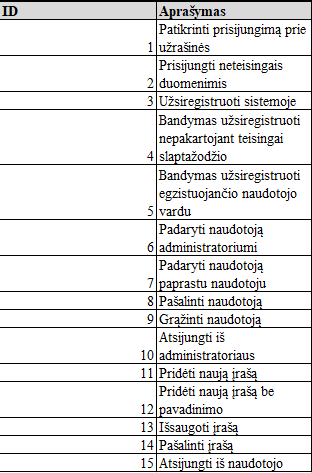
Kuriamai programų sistemai buvo sukurti testai. Šių testų paskirtis buvo patikrinti , ar korektiškai veikia programos informacijos talpinimas bei kitos funkcijos. Testai buvo kuriami tokia tvarka: visu pirmą buvo aprašomas rezultatas , kurio tikimės gauti testui įvykus , tada buvo aprašomas realus rezultatas , kurį gauną programa įvykdžius mums duotą funkciją ir galiausiai šiuos du rezultatus palyginame. Jeigu rezultatai sutampa , tada yra aišku , jog informaciją keliauja sėkmingai ir mes gauname tokius rezultatus , kokių tikėjimos ir programa veikia korektiškai. Testams atlikti buvo naudojami unit testai , kadangi šiuos testus palaiko bei siūlo pati Visual Studio aplinka.



13 pav. Testavimo sceanrijai



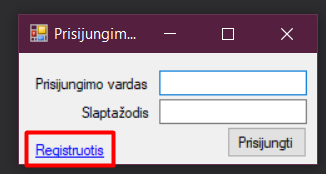
14 pav. Testavimo scenarijai 2



15 pav. Testavimo scenarijų aprašymai

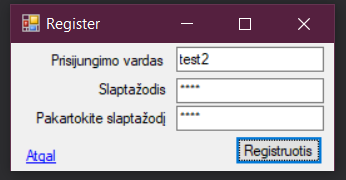
VARTOTOJO INSTRUKCIJA

Šioje skiltyje bus parodomas programos funkcionalumas bei bus pademonstruojama , kaip reikia naudotis programa. Pirmas žingnis yra užsiregistruoti programos sistemoje paspaudus mygtuką „registruotis“ (žr. 13 pav.).



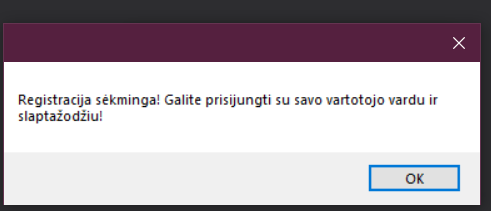
16pav. Sistemos pradinio lango pavyzdys

Paspaudus mygtuką „registruotis“ sistema pateiks registracijos formą (žr. 14 pav.) kurioje vartotojas turės pateikti savo sugalvotą prisijungimo vardą bei slaptažodį.



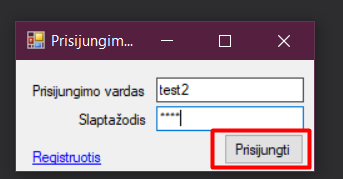
17pav. Registracijos forma.

Jeigu registracija yra sėkminga , sistema praneš apie tai. (žr. 15 pav.). Jeigu registracija yra nesėkminga , sistema paaiškins vartotojui , kodėl registracija įvyko nesėkmingai ir vartotojas galės per naujo bandyti užsiregistruoti.



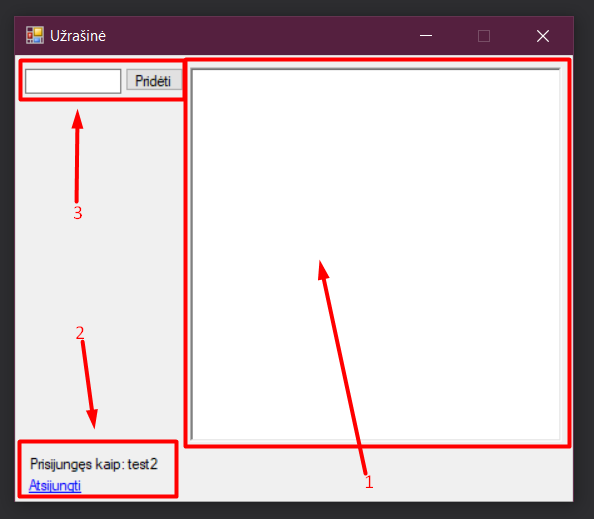
18pav. Sėkmingos registracijos langas

Toliau reikia prisijungti prie sistemos naudojant katik suvestus duomenis ir spaudžiant mygtuką „Prisijungti“. (žr. 16 pav.)



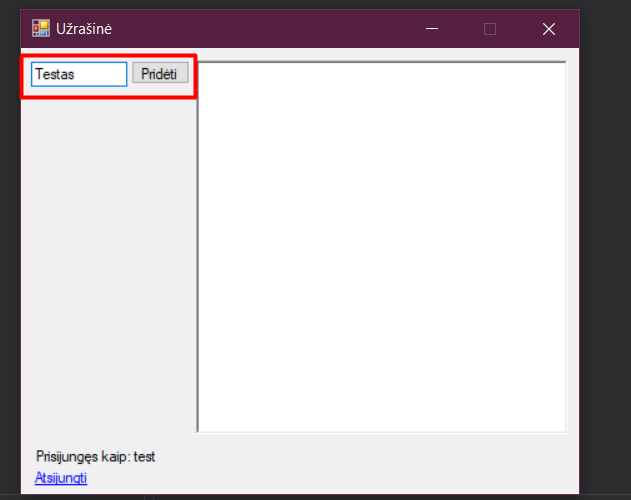
19pav. Prisijungimo ekranas

Toliau sistem atvaizduoja užrašinės ekraną (žr. 17 pav.) kuriame galite: 1.Įrašyti kokį norime teskta ir jį saugoti. 2. Atsijungti nuo sistemos 3.Sukurti užrašine irašydami juos pavadinima.



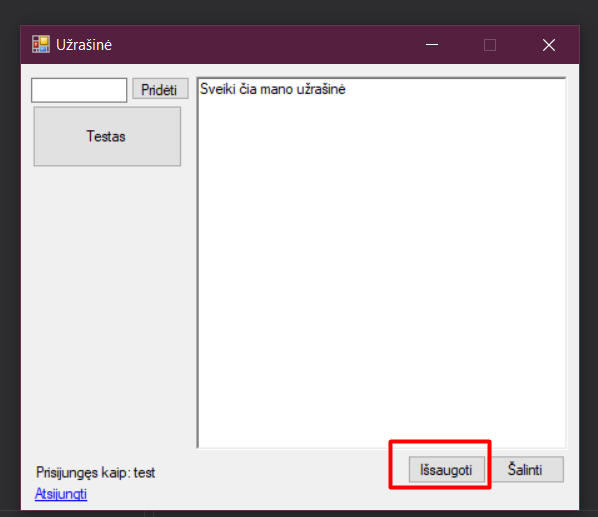
20pav. Sistemos funkcijos

Norint sukurti nauja užrašine vartotojas turės irašyti jos pavadinima ir paspausti mygtuką „Pridėti“ (žr. 18 pav.)



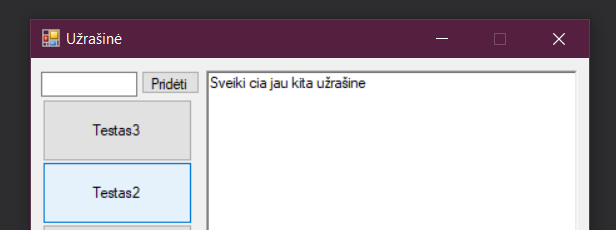
21pav. Naujos užrašinės sukūrimas

Pridėjus nauja užrašinę vartotojai galės joje irašyti informaciją , tačiau norint išsaugoti šia informaciją vartotojas turės paspausti mygtuką „Išsaugoti“ (žr. 19 pav.)

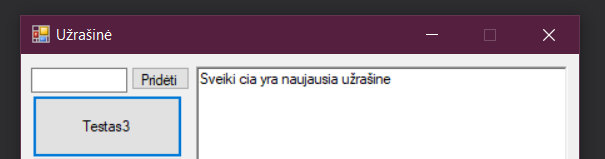


22pav. Užrašinės teksto išsaugojimas

Išsaugota informacija liks užrašineje , ir kartojant šiuos žingsnius galima susikurti daug užrašinių su skirtingais tekstais. (žr. 20, 21 pav.)



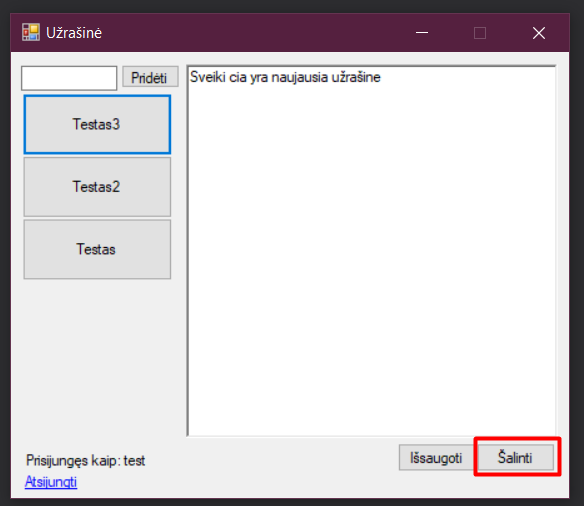
23pav. Antros užrašinės pavyzdys



24pav. Trečios užrašinės pavyzdys

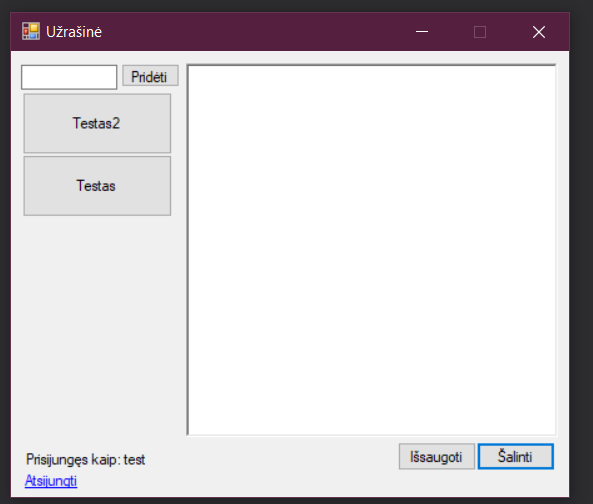
Taip pat vartotojai gali ištrinti užrašinę pasirinke norima užrašinę ir paspaude mygtuką „Šalinti“

(žr. 22 pav.)



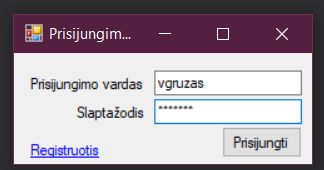
25 pav. Trinimo pavyzdys

Paspaudus mygtuką „Šalinti“ sistema paklaus ar tikrai vartotojas nori ištrinti užrašinę ir jeigu bus pasirinkta taip sistema pašalins užrašine (žr. 23 pav.)



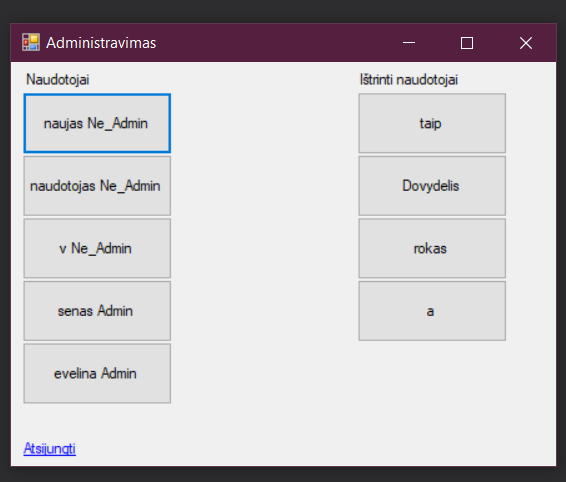
26pav*.* Pašalinta užrašinė

Toliau galima prisijungti prie sistemos kaip administratorius. Tam reikės suvesti administratoriaus prisijungimo duomenis. (žr. 24 pav.)



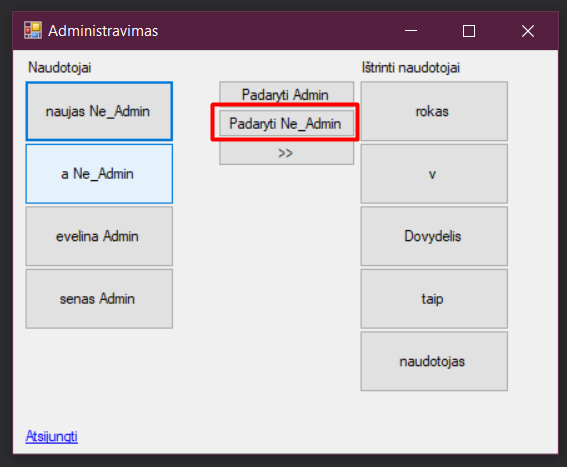
27pav*.* Administratoriaus prisijungimas

Administratorius prisijunges matys administratoriaus langą (žr. 25 pav.) kuriame bus atvaizduoti visi vartotojai.



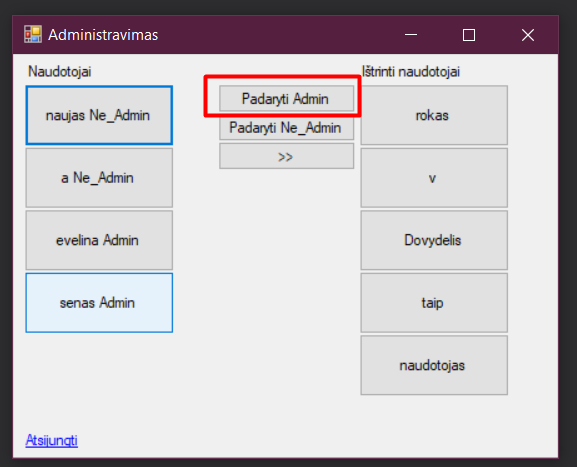
28pav. Atvaizdavimo panelė

Šioje panelėje administratorius gali norimą vartotoją padaryti administratorium arba atimti jo administratoriaus pareigas paspaudus ant norimo vartotojo ir pasirinkus mygtuką „Padaryti Ne\_Admin“ (žr. 26 pav.)



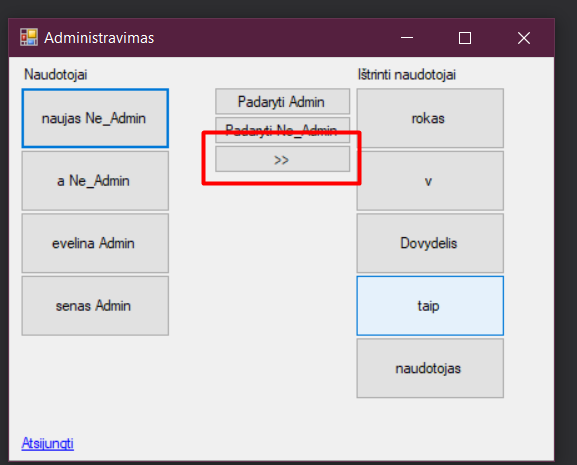
29pav. Administratoriaus teisių atimimas

Tai pat tokiu pačiu budų tačiau paspaudus mygtuką „Padaryti Admin „ administratorius gali suteikti administratoriaus teises vartotojui (žr. 27 pav.)



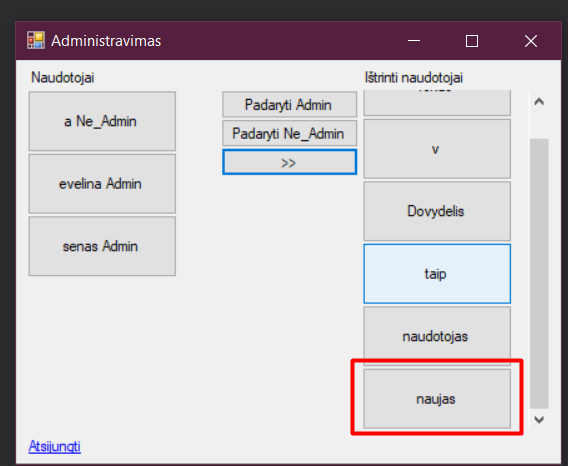
30 pav. Administratoriaus teisių suteikimas

Galiausiai administratorius gali pašalinti vartotoją paspaudęs mygtuką „>>“ (žr. 28 pav.)



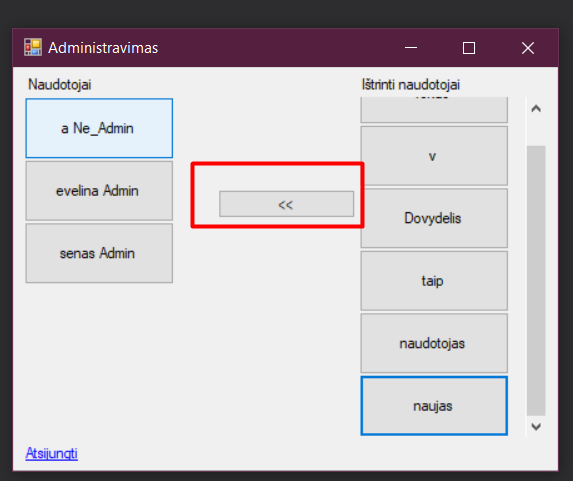
31pav. Vartotojo trynimas

Ištrinti vartotojai atsiras dešinėje pusėje. (žr. 29 pav.)



32pav. Ištrintas vartotojas

Ir paskutinis veiksmas ką gali padaryti administratorius tai sugražinti vartotoją iš ištrintų vartotojų sąrašo. Tai galima padaryti paspaudus ant norimo vartotojo ir paspaudus mygtuką „<<“ (žr. 30 pav.)



33pav. Vartotojo sugražinimas

IŠVADOS

Šio projekto tikslas buvo sukurti pilnai funkcionuojančia užrašinės programą, kurioje vartotojai galės saugoti norimus užrašus. Galima teigti, jog projektas buvo atliktas sėkmingai, kadangi įsivaizduotą sistemą buvo sukurtą be jokių trukdžių ar papildomų resursų. Kuriamos sistemos darbas buvo paskirstytas kiekvienam asmeniui po lygiai. Domantas Makūnas buvo atsakingas už testavimo atvejus, Vaidotas Gružas buvo atsakingas už pačia programą bei jos funkcionalumą, Lukas Kašėta buvo atsakingas už dokumento paruošimą bei komandos aplinkos valdymą ir galiausiai Jovaras Černiauskas buvo šios sistemos užsakovas, tačiau šis asmuo iki sistemos sukūrimo pabaigos paliko projektą ir prie jo galutinio rezultato neprisidėjo. Projekto kūrime nekilo jokių iššūkiu kadangi sistema buvo sukurta laiku be to iš sprintų lentelės (žr.1pav) galima matyti, jog sistema buvo pradėta kurti spalio 1 dieną ir baigta kurti gruodžio 11 dieną su vieno mėnesio pertrauką per kurią sistemos kūrėjai apleido sistemos kūrimo etapą ir galvojo idėjas kaip toliau bus dirbamą.