A logo with a person in the middle

AI-generated content may be incorrect.

VILNIAUS KOLEGIJA   
ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS

**PASLAUGŲ VALDYMO SISTEMOS DOKUMENTACIJA**

PARENGĖ: Darius Jurgis Guščia

2025

### Turinys

1. PAVEIKSLŲ SĄRAŠAS 3

2. ĮVADAS 4

2.1. Užduotis 4

2.2. Tikslai 4

2.3. Projekto apimtis ir architektūra 4

3. TECHNINĖ DOKUMENTACIJA 5

3.1. Duomenų modelis 5

3.2. Sekų diagramos 7

3.3. API dokumentacija 8

4. NAUDOTOJO DOKUMENTACIJA 13

4.1. Paskirų kūrimas, duomenų generavimas ir prisijungimas 13

4.2. Rolės, teisės ir funkcijos 13

4.3. Prisijungimas prie sistemos 13

5. IŠVADOS 16

# PAVEIKSLŲ SĄRAŠAS

[1 pav. Projekto struktūros schema. 5](#_Toc209924788)

[2 pav. Sistemos duomenų modelių diagrama (ER) 5](#_Toc209924789)

[3 pav. UML klasių diagrama 6](#_Toc209924790)

[4 pav. Vartotojo prisijungimo sekos diagrama 8](#_Toc209924791)

[5 pav. Administratoriaus, vartotojo kūrimo sekos diagrama 8](#_Toc209924792)

[6pav. Vartotojų rolių funkcijos. 14](#_Toc209924793)

[7 pav. Pradinis sistemos prisijungimo langas. 15](#_Toc209924794)

[8 pav. Administratoriaus pagrindinis langas. 15](#_Toc209924795)

[9 pav. Vadybininko pagrindinis langas. 16](#_Toc209924796)

[10 pav. Vartotojo pagrindinis langas. 16](#_Toc209924797)

# ĮVADAS

## Užduotis

Sistemos tikslas – sukurti paslaugų valdymo platformą, kuri leistų administruoti sistemos vartotojus, jų bendrijas ir jiems teikiamas paslaugas. Sistema skirta užtikrinti greitą paslaugų administravimą bei patogų informacijos pateikimą. Taip pat atskirti vartotojų teises bei funkcionalumus priklausomai nuo jų vartotojų grupės, leidžiant ta pačia sistema naudotis tiek paprastiems vartotojams, tiek vadybininkams ar administratoriams.

## Tikslai

* Užtikrinti patogų vartotojų registracijos ir autentifikacijos procesą.
* Administratoriams suteikti galimybę kurti/redaguoti/trinti vartotojus, bendrijas bei teikiamas paslaugas.
* Vadybininkams suteikti galimybę priskirti vartotojus bei paslaugas bendrijoms, taip pat koreguoti teikiamų paslaugų kainas.
* Vartotojams suteikti galimybę peržiūrėti informaciją apie savo bendriją, priskirtas paslaugas ir jų kainas.

## Projekto apimtis ir architektūra

Projektas realizuotas trijų dalių sistema:

* „React“ kliento dalis. Atsakinga už vartotojo sąsają. Vartotojai čia mato duomenis ir vykdo veiksmus (pvz., prisijungimas, paslaugų peržiūra, paslaugų administravimas ir t.t.).
* „ASP.NET“ serverio dalis (Web API). Apdoroja iš kliento dalies gaunamas užklausas, vykdo autentifikaciją, taiko verslo logiką ir grąžina atsakymus klientui.
* „MySQL“ duomenų bazė. Saugo sistemos duomenis.

GitHub nuoroda: <https://github.com/JurgisGuscia/antroji_praktika>

Sistemos schema pavaizduota 1 paveiksle.

A diagram of a computer

AI-generated content may be incorrect.

1 pav. Projekto struktūros schema.

# TECHNINĖ DOKUMENTACIJA

## Duomenų modelis

2 paveiksle pavaizduota duomenų bazės ER diagrama.

A screenshot of a computer

AI-generated content may be incorrect.

2 pav. Sistemos duomenų modelių diagrama (ER)

Subjektų (angl. entities) aprašymas:

* Vartotojai – saugo vartotojų duomenis (vardas, pavardė, slapyvardis, slaptažodis (hash), lytis, rolė, grupė).
* Roles – saugo sistemoje sukurtas vartotojų grupes: administratorius, vadybininkas, vartotojas.
* Teisės – detalizuoja atskiras prieigos teises (pvz., kurti vartotojus, priskirti bendrijas ir .t.t).
* Bendrijos – atvaizduoja bendrijas prie kurių gali būti priskiriami vartotojai ir paslaugos.
* Paslaugos – paslaugų sąrašas su kainomis. Paslaugos gali būti priskirtos grupėms.

Ryšiai:

* Viena rolė gali turėti daug Vartotojų (1:N).
* Viena rolė gali turėti daug teisių, o vienos teisės gali priklausyti kelioms rolėms (M:N).
* Viena bendrija gali turėti daug vartotojų (1:N).
* Bendrija gali turėti daug paslaugų, o paslauga gali būti priskirta kelioms bendrijoms (M:N).

3 paveiksle atvaizduota UML klasių diagrama.

A diagram of a server

AI-generated content may be incorrect.

3 pav. UML klasių diagrama

Klasių aprašymas:

* „Users“ – reprezentuoja sistemos vartotoją. Laiko jo asmens duomenis (vardas, pavardė, vartotojo vardas, slaptažodį (hash), lytis) bei nuorodas į priskirtą rolę ir grupę.
* „Roles“ – aprašo rolę (pvz., administratorius, vadybininkas, vartotojas). Kiekviena rolė turi pavadinimą ir sąrašą jai priskirtų teisių.
* „Rights“ – aprašo konkrečias prieigos teises. Naudojamos detalizuoti, ką gali atlikti rolė. Pvz., teisė „peržiūrėti paslaugas“, „kurti vartotoją“.
* „Groups“ – apibrėžia bendriją, kuriai gali priklausyti vartotojai. Taip pat turi ryšį su paslaugomis.
* „Services“ – reprezentuoja paslaugas, kurios gali būti priskirtos grupėms (pvz., valymo paslauga, šildymas). Kiekviena paslauga turi pavadinimą ir kainą.

Ryšiai tarp klasių:

* „User“ priklauso vienai rolei „Role“, tačiau rolė gali turėti daug vartotojų (1:N).
* „Role“ gali turėti daug teisių „Rights“ ir teisė gali būti priskirta kelioms rolėms (M:N).
* „User“ gali priklausyti vienai grupei „Group“, bet grupė gali turėti daug vartotojų (1:N).
* „Group“ gali turėti daug paslaugų „Services“, o paslauga gali būti priskirta kelioms grupėms (M:N).

## Sekų diagramos

Paveiksluose 4 ir 5 pavaizduotos 2 scenarijų sekų diagramos.

A diagram of a computer program

AI-generated content may be incorrect.

4 pav. Vartotojo prisijungimo sekos diagrama

Sekos diagrama rodo, kaip vartotojas prisijungia prie sistemos:

* Vartotojas užpildo prisijungimo formą vartotojo sąsajoje „React“.
* Užklausa HTTP protokolu POST metodu siunčiama į „AuthController“ „/api/auth/login“ galinį tašką.
* Sistema patikrina duomenis duomenų bazėje (MySQL per EF Core ORM).
* Jei slaptažodis teisingas – sukuriama sesija ir grąžinamas sesijos slapukas. Jei ne – grąžinama klaida (401).

A diagram of a computer system

AI-generated content may be incorrect.

5 pav. Administratoriaus, vartotojo kūrimo sekos diagrama

Administratorius sukuria naują vartotoją:

* Administratorius per React pateikia vartotojo pridėjimo formą.
* Užklausa HTTP protokolu POST metodu siunčiama į „AuthController“ „/api/users“ galinį tašką.
* „AuthController“ patikrina duomenų korektiškumą.
* Jei duomenys tinkami, jie išsaugomi duomenų bazėje ir grąžinamas atsakymas su sukurtu vartotoju. Klaidos atveju – grąžinama 400 klaidos kodas.

## API dokumentacija

API dokumentacija pateikiama lentelėmis. Lentelės stulpeliai:

* URL – API galinio taško adresas.
* Metodas – HTTP metodas (GET, POST, PUT, DELETE).
* Paskirtis – kam skirtas galinis taškas.
* Užklausa – kokių duomenų tikimasi užklausoje.
* Atsakymas – kokį atsakymą (duomenis, statuso kodus) grąžina serveris.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API adresas | Metodas | Paskirtis | Užklausa | Atsakymas |
| /api/Auth/login | Post | Prisijungti |  | Sesijos slapukas |
| /api/Auth/logout | Post | Atsijungti | - | Sesijos sunaikinimas |
| /api/Auth/register | Post | Registracija |  |  |
| /api/auth/me | Get | Gauti vartotojo informaciją | - |  |
| /api/users | Get | Gauti visų vartotojų sąrašą | - |  |
| /api/users/{id} | Get | Gauti nurodyto vartotojo duom. | - |  |
| /api/users | Post | Pridėti vartotoją |  |  |
| /api/users/{id} | Put | Atnaujinti vartotojo informaciją. |  | 200 Ok |
| /api/users/{id} | Delete | Panaikinti vartotoją | - | - |
| /api/users/assignUsers/{id} | Put | Priskirti vartotojui grupei {id} |  | 200 Ok |
| /api/rights | Get | Gauti teisių sąrašą | - |  |
| /api/rights/{id} | Get | Gauti {id} teisės informaciją | - |  |
| /api/services | Get | Gauti paslaugų informaciją | - |  |
| /api/services/{id} | Get | Gauti paslaugos informaciją | - |  |
| /api/services | Post | Pridėti paslaugą |  |  |
| /api/services/{id} | Put | Atnaujinti paslaugos pavadinimą |  | 200 Ok |
| /api/services/prices | Put | Atnaujinti paslaugų kainas |  | 200 Ok |
| /api/services/{id} | Delete | Panaikinti paslaugą | - | - |
| /api/roles | Get | Gauti rolių informaciją | - |  |
| /api/roles/{id} | Get | Gauti rolės informaciją | - |  |
| /api/roles | Post | Sukurti rolę |  |  |
| /api/roles/{id} | Put | Atnaujinti rolės informaciją |  | 200 Ok |
| /api/roles/{id} | Delete | Ištrinti rolę | - | - |
| /api/roles/{id}/rights | Put | Priskirti teises rolei (junction) |  |  |
| /api/roles/{id}/rights | Get | Gauti rolės teises (junction) | - |  |
| /api/groups | Get | Gauti bendrijų informaciją | - |  |
| /api/groups/{id} | Get | Gauti bendrijos informaciją | - |  |
| /api/groups | Post | Pridėti bendriją |  |  |
| /api/groups/{id} | Put | Atnaujinti bendrijos informaciją |  |  |
| /api/groups/{id} | Delete | Ištrinti bendriją | - | - |
| /api/groups/{id}/services | Get | Gauti bendrijos paslaugas | - |  |
| /api/groups/{id}/users | Get | Gauti bendrijos vartotojus | - |  |
| /api/groups/{id}/services | Put | Atnaujinti bendrijos paslaugas (junction) |  |  |

# NAUDOTOJO DOKUMENTACIJA

## Paskirų kūrimas, duomenų generavimas ir prisijungimas

Vartotojų paskyras kuria administratoriai pačioje sistemoje. Administratorius, kurdamas naują vartotoją turi nurodyti jo vardą, pavardę, bendriją ir rolę (administratorius, vadybininkas, vartotojas). Vartotojo vardui ir slaptažodžiui nustatyti yra 2 pasirinkimai – įvesti rankiniu būdu arba pažymėti varnelę „auto generavimas“. Pasirinkus automatinį generavimą, prisijungimo duomenys bus nustatyti automatiškai, nepriklausomai ar prieš tai vartotojo vardo ar slaptažodžio laukuose buvo įvesta informacija ar ne. Automatinio generavimo atveju, vartotojo vardu tampa vardas, o slaptažodžiu – pavardė.

## Rolės, teisės ir funkcijos

6 paveikslėlyje pavaizduotos visos vartotojų rolės ir joms priskirtos funkcijos.

A black and white list with white text

AI-generated content may be incorrect.

6 pav. Vartotojų rolių funkcijos.

Šita struktūra leidžia aiškiai atskirti administratoriaus atsakomybes (sistemos valdymas), vadybininko funkcijas (bendrijų, paslaugų ir gyventojų administravimas) bei paprastiems vartotojams pateikiamą informaciją.

## Prisijungimas prie sistemos

Prie sistemos vartotojai prisijungia naudodamiesi interneto naršykle, adresu: <https://www.praktika.jurgisguscia.lt>.

Prisijungimo žingsniai:

* Atidaryti sistemos svetainę naršyklėje.
* Prisijungimo lange įvesti vartotojo vardą ir slaptažodį.
* Paspausti mygtuką „Prisijungti“.
* Jei duomenys teisingi – sistema sukuria sesiją ir naudotojas nukreipiamas į savo paskyros puslapį.
* Jei duomenys neteisingi – parodomas klaidos pranešimas.

5 paveiksle pavaizduotas pradinis prisijungimo langas.

A screenshot of a computer

AI-generated content may be incorrect.

7 pav. Pradinis sistemos prisijungimo langas.

Administratorius mato administravimo funkcijas (8 pav.):

* Bendrijų valdymą (sukurti, redaguoti, šalinti).
* Paslaugų valdymą (sukurti, redaguoti, šalinti).
* Vartotojų valdymą (sukurti, redaguoti, šalinti).

A screenshot of a computer

AI-generated content may be incorrect.

8 pav. Administratoriaus pagrindinis langas.

Vadybininkas mato valdymo funkcijas (9 pav.):

* Bendrijų sąrašą
* Bendrijai priskirtų ir galimų priskirti paslaugų sąrašą
* Bendrijai priskirtų ir galimų priskirti vartotojų sąrašą.
* Kainų koregavimo langą.

A screenshot of a computer

AI-generated content may be incorrect.

9 pav. Vadybininko pagrindinis langas.

Vartotojas mato tik savo informaciją (10 pav.):

* Asmeninį profilį.
* Bendriją, kuriai priklauso.
* Tos grupės paslaugas.

A white background with black text

AI-generated content may be incorrect.

10 pav. Vartotojo pagrindinis langas.

Tokiu būdu kiekvienas prisijungęs naudotojas turi tik savo rolės leidžiamas funkcijas.

# IŠVADOS

Projekto metu buvo sukurta vartotojų, rolių, teisių, bendrijų ir paslaugų valdymo sistema. Ji realizuota naudojant React technologiją vartotojo sąsajai, ASP.NET Core Web API serverio daliai bei MySQL duomenų bazei. Sistema užtikrina vartotojų registracijos, autentifikacijos, sesijų valdymo funkcijas, o administratoriams bei vadybininkams suteikia galimybę centralizuotai valdyti vartotojus, bendrijas, paslaugas ir jų kainas.

Pagrindinės įgyvendintos funkcijos:

* Prisijungimas ir autentifikacija.
* Vartotojų valdymas.
* Bendrijų valdymas.
* Paslaugų valdymas.
* Rolių sistema su skirtingomis funkcijomis.

Galimi ateities patobulinimai:

* Vietoje sesijų autentifikacijai naudoti JWT.
* Pridėti serverio dalies autentifikaciją.
* Patobulinti vartotojo sąsają.
* Įdiegti serverio pranešimų saugojimo sistemą (angl. logger).
* Pridėti papildomas funkcijas, pvz., paslaugų paiešką, filtravimą arba ataskaitų generavimą.
* Optimizuoti duomenų bazės užklausas didesniems duomenų kiekiams.
* Įdiegti centralizuotą klaidų valdymo sistemą.
* Serveryje įdiegti užklausų ribotuvą (angl. debouncer).