



**UŽSIĖMIMŲ TVARKARAŠČIŲ PROJEKTAVIMO  
SISTEMA  
MAGISTRINIS DARBAS**

**Atliko:** .....  
IFN-3 gr. stud.Svetlana Neurova

**Vadovas:** .....  
D. Rubliauskas

# TURINYS

1. ĮVADAS .....	3
2. ANALIZĖ .....	4
3. SISTEMOS REIKALAVIMŲ SPECIFIKACIJA .....	14
4. PROJEKTAS .....	31
5. EKSPERIMENTINIS SISTEMOS TYRIMAS .....	46
6. IŠVADOS.....	70
7. TERMINAI .....	70
8. LITERATŪROS SĄRAŠAS.....	71

## 1. Įvadas

Užsiėmimų tvarkaraščių projektavimo sistema – tai programinė įranga skirta pamokų tvarkaraščių mokykloje sudarymui, kuri sudaro tvarkaraštį pagrindinių mokyklų mokiniams bei mokytojams. Svarbiausias reikalavimas sistemai - sistema turi sumažinti vartotojų laiko sąnaudas pateikiant ir gaunant svarbiausią informaciją apie mokyklos tvarkaraštį bei mokytojų veiklą.

Pagrindiniai IS vartotojai – mokyklos administracija, mokytojai, mokiniai, mokinių tėvai.

Pagrindinės IS veiklos sritys – pateikti informacija apie mokyklos tvarkaraštį, bei mokytojų ir kabinetų užimtumą.

Šiuolaikinių technologijų progresas įpareigoja ieškoti efektyviausių sprendimų optimizuojant įmonių/organizacijų veiklos planavimo procesą. Automatizuoti įmonės ar organizacijos procesų, svarbių įvykių planavimas ir kontrolė, išteklių apskaitos planai gerina įmonės darbo kokybę, mažina valdymo kaštus. Jei organizacijos tikslai ir veikla yra mokomojo pobūdžio (švietimo org.) tiesiog gyvybiškai svarbus tinkamas resursų paskirstymas sudarant mokymo patalpų ir įrankių panaudojimo, mokytojų darbo planus. Tačiau daugeliu atveju planas sudaromas rankiniu metodu ir rezultatas daugeliu atitinka išanksto nusibrėžtus reikalavimus. Pakeitimams ir analizei sugaištama daug brangaus laiko. Žvelgiant abstrakčiau kiekvieno visuomenės individo gyvenime tinkamai ir greitai suplanuojant kiekvieną etapą išvengiama daug problemų ir pasiekiami daug svaresnių rezultatų per trumpesnę laiką.

### *Sistemos tikslai.*

Užsiėmimų tvarkaraščių sudarymo sistema (UTSS) – tai programinė įranga skirta pamokų tvarkaraščių mokykloje sudarymui, kuri sudaro tvarkaraštį pagrindinių mokyklų mokiniams bei mokytojams.

*Svarbiausias reikalavimas sistemai* - sistema turi sumažinti vartotojų laiko sąnaudas pateikiant ir gaunant svarbiausią informaciją apie mokyklos tvarkaraštį bei mokytojų veiklą.

Tvarkaraščio Sudarymo Sistema (toliau TSS) yra įrankis, automatizuojantis įstaigos ar organizacijos išteklių paskirstymo plano sudarymą. Jo paskirtis – sudaryti, redaguoti ir optimizuoti įvairių išteklių panaudojimo planus nustatytame laiko intervale, parodyti visus galimus išteklių išdėstymo, persidengimo, neišnaudojimo variantus, perspėti apie tai TSS vartotoją.

Kompiuterizuojamų funkcijų sąrašas:

- ✓ sudaryti, redaguoti ir atvaizduoti įvairių resursų paskirstymo tvarkaraščius;
- ✓ optimizuoti resursų paskirstymo planą, ir atlikti reikalingas plano korekcijas;
- ✓ pateikti resursų paskirstymo planą vartotojui įvairia forma (elektroniniu formatu, popieriuje, ir t.t.).

Tokio pobūdžio projektai tikrai nėra naujiena. Pasaulyje yra nemažai garsių firmų siūlančių savo sistemas, kurios skirtos projektų, resursų, įvykių planams sudaryti.

## **2. Analizė**

Šiuolaikinių technologijų progresas įpareigoja ieškoti efektyviausių sprendimų optimizuojant įmonių/organizacijų veiklos planavimo procesą. Jei organizacijos tikslai ir veikla yra mokomojo pobūdžio (švietimo org.) tiesiog gyvybiškai svarbus tinkamas resursų paskirstymas sudarant mokymo patalpų ir įrankių panaudojimo, mokytojų darbo planus.

### **2.1 Analizės tikslas:**

*Analizės tikslas yra tas, kad populiariausios sistemų grafiniai redaktoriai bei komponentės bus apžvelgtos tikslu įsisavinti savybes, kurios galėtų būti (t.y. užsakovo nepaminėtos) realizuotos kuriamoje sistemoje.*

### **2.2 Tyrimo sritis, objektas ir problema**

Darbo tyrimo sritis yra išanalizuoti jau esančias ir sukurti programą, kuri sudaro tvarkaraštį pagrindinių mokyklų mokiniams bei mokytojams.

Eilę metų asmenys atsakingi už tvarkaraščių sudarymą mokyklose sprendė pakankamai sunkų ir opų klausimą kaip sudaryti tvarkaraštį tokį, jog tenkintų kuo didesnį skaičių mokinių ir būtų patogus mokytojams. Paprastai tai buvo daroma rankiniu būdu nes programinės įrangos tam tikslui išspręsti nebuvo. Žinoma stengiamasi sudaryti tvarkaraštį tokį, jog jis turėtų kuo mažiau "langų" ir būtų priimtinas tiek mokiniams, tiek mokytojams. Vienas iš svarbiausių ir būtinas kriterijus, kuris turi būti išpildytas - fiziniai apribojimai.

### **2.3 Aplinkos (organizacijos veiklos, proceso ir pan.) analizė**

Užsakovą atstovaujanti įstaiga, t. y. pagrindinė mokykla šiuo metu nenaudoja jokios automatizuoto išteklių (pamokų, mokytojų, kabinetų ir pan.) planų sudarymo įrankio – planus sudaro, analizuoja ir redaguoja žmogus.

### **2.4 Vartotojų analizė**

Užsakovą atstovaujanti įstaiga, t. y. pagrindinė mokykla šiuo metu nenaudoja jokios automatizuoto išteklių (pamokų, mokytojų, kabinetų ir pan.) planų sudarymo įrankio – planus sudaro, analizuoja ir redaguoja žmogus. Tai yra pakankamai neefektyvu, nes planų sudarymo procese didelę reikšmę turi žmogiškasis faktorius. Taip pat nėra

(arba neefektyviai) atliekamas plano optimizavimas, todėl daugeliu atveju sudarytu planu lieka nepatenkinti vartotojai (mokiniai, mokytojai).

TSS užsakovas yra Klaipėdos „A.Rubliovo“ pagrindinė mokykla. Šios sistemos vartotojai – įstaigos/organizacijos, kuriose įdiegtas šia sistema, darbuotojai.

## 2.5 Panašių sistemų analizė

### “Mimoza”

Tai tvarkaraščio sudarymo kompiuterinė programa skirta darbo automatizavimui.

Tvarkaraščio formavimui galima panaudoti susikurtą savo mokyklos duomenų bazę (Excel, Access ir kitose programose), galima atspausdinti įvairias ataskaitas.

- Testuota programa su esamais serveryje duomenimis per atstumą naudojant naršyklę Internet Explorer5. Tvarkaraščio programa veikia naudojant optimizavimui Mig1 ir Bayes metodus. Gautus optimizavimo rezultatus galima matyti ekrane, nusikopijuoti bei atsispausdinti.
- Sudarytas eksperimentinį mokyklos tvarkaraščio failas su 20 langų (duomenų failas buvo nusiųstas į serverį optimizuoti).
- Pasinaudojus "SheduleLoader" tvarkataštis ištaisytas ir nusiųstas pakartotinai.
- Atlikus nusiųsto tvarkaraščio optimizavimą, gauti tokie rezultatai:

Metodas	Nustatytas iteracijų skaičius	Iteracijų skaičius optimizuojant	F(x)	Tikimybė	Langų skaičius
Mig	1	0	25,0	0,893	25
Mig	1	0	25,0	0,828	25
Mig	10	7	3,0	0,031	3
Mig	10	7	3,0	0,528	3
Mig	20	15	1,0	0,889	1
Mig	20	15	0,0	0,402	0
Bayes	1	4	3,0	0,125	3
Bayes	1	4	11,0	0,125	11
Bayes	10	17	0,0	0,407	0
Bayes	10	6	2,0	0,02	2
Bayes	20	15	0,0	0,363	0
Bayes	20	12	3,0	0,332	3

*Rezultatai*

### **Trūkumai:**

- ✓ Kadangi programa iki galo nėra realizuota, kompiuteris optimizuodamas pamokų tvarkaraštį perkilnoja ir pamokas, kurios turėtų būti nejudinamos. Pradiniam duomenims reikėtų uždėti svorį ar kaip nors kitaip juos išskirti. Pagal Sidorov aprašymą, tai nėra pilnai sutvarkyta. Šio ribojimo reikėtų iš viso atsisakyti, nes toks pamokų išdėstymas galimas mokant ne tik kalbas, bet ir kitus dalykus.
- ✓ Nusiunčiamo failo pavadinimas neturėtų būti konkretus. Programa turėtų priimti bet kokio pavadinimo failą.
- ✓ Optimizuojant labai keistai pasirenkamas iteracijų skaičius. Nesutampa užduotas su vykdomu skaičiumi.
- ✓ Optimizuojant tuo pačiu metodu, pasirenkant tą patį iteracijų skaičių gaunami skirtingi rezultatai. (žiūr. lentelę)
- ✓ Naudojant mažesnę iteracijų skaičių geriau veikia Bayes metodas, prie didesnio iteracijų skaičiaus - panašiai abu.
- ✓ Programą reikia tobulinti ir plėsti atsižvelgiant į realią situaciją ir poreikius, kurie yra nurodyti programuotojų faile "geri\_norai", kuria taip pat nurodyti mano pasiūlymai dėl programos algoritmo tobulinimo.
- ✓ Dirbant nutolusiame kompiuteryje piko valandomis programos darbas lėtas, o tai sukelia nepatogumų. Kai linijos laisvesnės, greitis pakankamai geras.

### **Kompiuterinė programa "Pamokos"**

#### **Privalumai:**

- ✓ Programos pagalba galima sudaryti pamokų tvarkaraščius bet kurio tipo mokykloms.
- ✓ Dvi pamainos viename tvarkaraštyje;
- ✓ klasę galima skirstyti į 2-5 grupes;
- ✓ galima sujungti kiek norima klasių, ir tada skirstyti į 1-5 grupes;
- ✓ laisvų dienų ar valandų norimą dieną pasirinkimas mokytojams;
- ✓ programinis "pamokų paskirstymo" lentelės suformavimas;
- ✓ programinis pamokos formavimas iš "pamokų paskirstymo" lentelės;
- ✓ programinis "likučių" įrašymas;
- ✓ tvarkaraščio koregavimas, leidžiantis sukeisti pamokas vietomis kartu keičiant ir pamokas, kurios trukdo sukeitimui;
- ✓ kabinetų skirstymas;
- ✓ tvarkaraščio spausdinimas - mokytojams, klasėms, mokyklai, kabinetams;
- ✓ tvarkaraščio archyvas - galima sukaupti iki 15 tvarkaraščio variantų.
- ✓ užprogramuoti reikalavimai moksleivių tvarkaraščiams, bet palikta galimybė laisvai rinktis:
- ✓ tą pačią dieną gali būti ne daugiau kaip 2 tikslieji dalykai (matematika, chemija, fizika, informatika);
- ✓ tą pačią dieną gali būti ne daugiau kaip 2 lengvieji dalykai (muzika, darbai, dailė, teatras, choreografija, kūno kultūra);
- ✓ dvigubos lietuvių kalbos pamokos vyresnėse klasėse tik antradienį, trečiadienį, ketvirtadienį ne vėliau, kaip ketvirta pamoka;

- ✓ dalykai turintys 2 savaitines pamokas dedami kas antra diena;
- ✓ po kūno kultūros nepageidaujama daryti matematikos;
- ✓ pamokos, kuriose dalyvauja dalis klasės, dedamos nuo 5 pamokos ir vėliau.

**Trūkumai:**

- ✓ Nesuformuojamos pamokos pagal moksleivių pasirinkimą fiksuojant kiekvienoje grupėje moksleivių, kurie lankys šį dalyką, kodus.
- ✓ Nesuformuojama kasdieninis tvarkaraščio koregavimas ir vestų pamokų apskaita.
- ✓ Programa neišbandyta aukštosioms mokykloms.
- ✓ Realizuota pasenusioje OS (MSDOS) terpėje, todėl nepalaiko MS Windows 2000, NT OS.
- ✓ Prasta grafinė sąsaja su vartotoju.

**“CELCAT” tvarkaraščio sudarymo programa**

**Privalumai:**

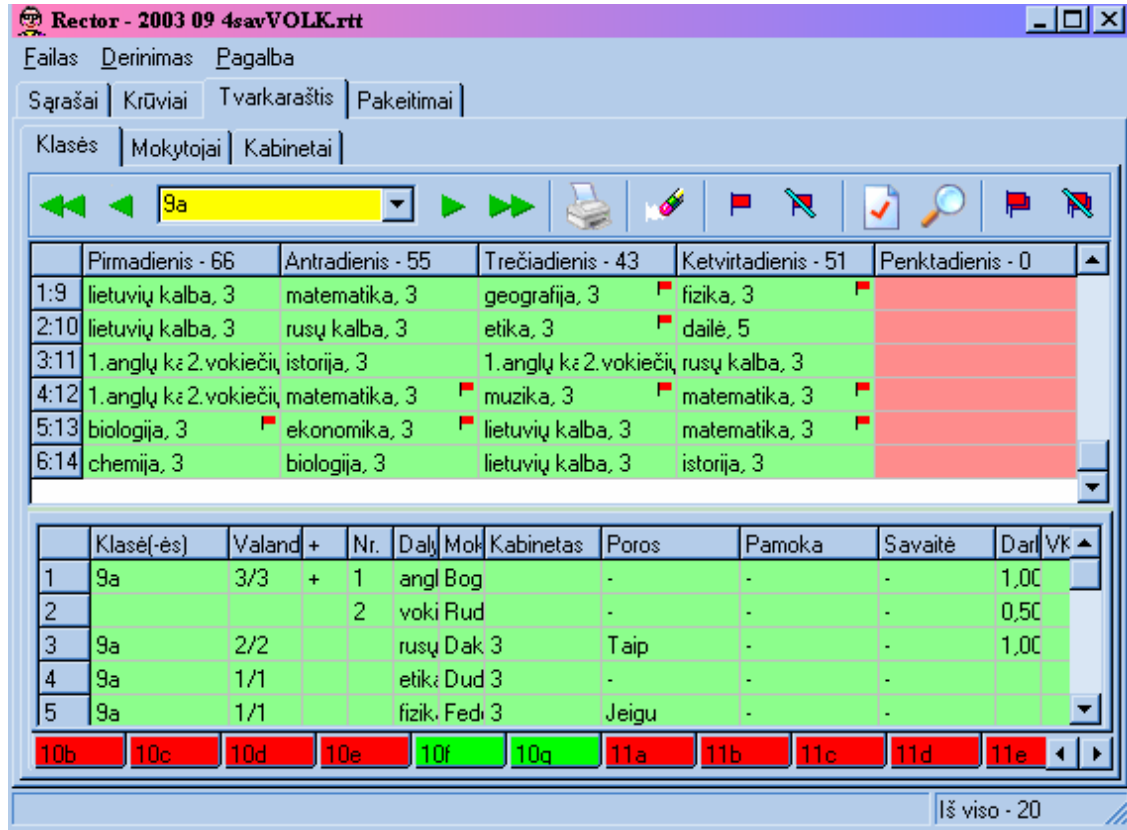
- ✓ Veikia net ant 486 DX4/100 PC su 16MB RAM atminties, OS – Windows.
- ✓ Leidžiama nustatyti 36 laiko periodus per dieną, 7 per savaitę, 56 savaites metuose.
- ✓ Periodas nustatomas vartotojo
- ✓ Įvykiai tvarkaraštyje gali būti nustatomi 1 minutės tikslumu.
- ✓ Įvykiai tvarkaraštyje gali būti suplanuoti vienai savaitei arba bet kokiai daugelio savaitių kombinacijai.
- ✓ Nėra apribojimų personalo, modulių, kambarių, kitiems resursams.
- ✓ Apie kiekvieną “langą” tvarkaraštyje pranešama vartotojui.
- ✓ “Langų” tikrinimas gali būti konfigūruojamas. Pranešimai yra detalūs ir informatyvūs.
- ✓ Tvarkaraštis gali būti nustatytas pagal modulius, grupę, studentą, kambarį ar aptarnaujantį personalą – dėstytojus,...
- ✓ Kiekvieno tvarkaraščio kūrimo etapo metu galima prieiti prie visos saugomos informacijos.
- ✓ Tvarkaraščiai gali būti atspausdinti forma, kuri gali būti vartotojo keičiama.
- ✓ Lengvai generuojamos statistinės ataskaitos.
- ✓ Galima naudotis kontekstinė pagalba, pagalba internetu(on-line).

**Trūkumai:**

- ✓ Grafinė sąsaja, pagalba anglų kalba.
- ✓ Nepakankamai atliekamas “langų” optimizavimas.

## “RECTOR” tvarkaraščio sudarymo programa

*Rector* nereikalauja specialių žinių IT srityje. Interfeisas labai paprastas ir naudojama standartinės interfeiso priemonės – mygtukai, iškrentantis meniu, žymės ir kt.



The screenshot shows the 'Rector - 2003 09 4savVOLK.rtt' window. It features a menu bar with 'Failas', 'Derinimas', and 'Pagalba'. Below the menu are tabs for 'Sąrašai', 'Krūviai', 'Tvarkaraštis', and 'Pakeitimai'. The 'Tvarkaraštis' tab is active, showing a dropdown for 'Klasės' with 'Mokytojai' and 'Kabinetai' options. A toolbar contains navigation and action icons. The main area displays a weekly schedule for class '9a' with columns for days of the week and rows for lessons. Below this is a detailed lesson table with columns for 'Klasė(-ės)', 'Valand +', 'Nr.', 'Dalj', 'Mok', 'Kabinetas', 'Poros', 'Pamoka', 'Savaitė', and 'Darl/VK'. At the bottom, there are buttons for other classes (10b-10e, 10f, 11a-11e) and a status bar showing 'Iš viso - 20'.

	Pirmadienis - 66	Antradienis - 55	Trečiadienis - 43	Ketvirtadienis - 51	Penktadienis - 0
1:9	lietuvių kalba, 3	matematika, 3	geografija, 3	fizika, 3	
2:10	lietuvių kalba, 3	rusų kalba, 3	etika, 3	dailė, 5	
3:11	1. anglų k. 2. vokiečių, istorija, 3		1. anglų k. 2. vokiečių, rusų kalba, 3		
4:12	1. anglų k. 2. vokiečių, matematika, 3		muzika, 3	matematika, 3	
5:13	biologija, 3	ekonomika, 3	lietuvių kalba, 3	matematika, 3	
6:14	chemija, 3	biologija, 3	lietuvių kalba, 3	istorija, 3	

	Klasė(-ės)	Valand +	Nr.	Dalj	Mok	Kabinetas	Poros	Pamoka	Savaitė	Darl/VK
1	9a	3/3	+	1	angl Bog		-	-	-	1,00
2				2	voki Rud		-	-	-	0,50
3	9a	2/2			rusų Dak 3		Taip	-	-	1,00
4	9a	1/1			etik: Dud 3		-	-	-	
5	9a	1/1			fizik. Fed: 3		Jeigu	-	-	

### Privalumai:

- ✓ Sudarant pamokų tvarkaraštį, programa suteikia galimybę: sudaryti tvarkaraštį taip, kad pamokų krūvis moksleiviams būtų išdėstytas tolygiai per savaitę;
- ✓ sudaryti grupes, klases jungiant į srautus iki šešių klasių;
- ✓ mažinti langus mokytojams;
- ✓ pakeisti pamokas, mokytojams nesant;
- ✓ sudaryti skirtingus tvarkaraščius porinėms ir neporinėms savaitėms;
- ✓ numatyti metodines dienas mokytojams;
- ✓ sudaryti tvarkaraštį, kai užsiėmimai vyksta keliomis pamainomis.
- ✓ programos pagalba galima sudaryti pamokų tvarkaraščius bet kurio tipo mokykloms.
- ✓ klasę galima skirstyti į 2 grupes;
- ✓ galima sujungti kiek norima klasių, ir tada skirstyti į grupes;
- ✓ laisvų dienų ar valandų norimą dieną pasirinkimas mokytojams;
- ✓ programinis "pamokų paskirstymo" lentelės suformavimas;
- ✓ Priskiriant krūvius, automatiškai sudaroma ir tarifikacija



- ✓ Programa suskaičiuoja valandas mokytojams pagal klasių grupes
- ✓ Suskaičiuoja valandas, skirtas darbų taisymui
- ✓ Automatiškai paskaičiuojamas mokytojų darbo stažas, vieną kartą jį įvedus
- ✓ Mokytojams galima priskirti auklėjamąsias klases
- ✓ Galima paderinti tarifkacinę lentelę, pasirenkant, kokias skiltis norima matyti ir spausdinti
- ✓ programinis pamokos formavimas iš "pamokų paskirstymo" lentelės;
- ✓ tvarkaraščio koregavimas, leidžiantis sukeisti pamokas vietomis kartu keičiant ir pamokas, kurios trukdo sukeitimui;
- ✓ kabinetų skirstymas;
- ✓ tvarkaraščio spausdinimas - mokytojams, klasėms, mokyklai, kabinetams;
- ✓ tvarkaraščio archyvas - galima sukaupti iki 15 tvarkaraščio variantų.
- ✓ tą pačią dieną gali būti ne daugiau kaip 2 tikslieji dalykai (matematika, chemija, fizika, informatika);
- ✓ tą pačią dieną gali būti ne daugiau kaip 2 lengvieji dalykai (muzika, darbai, dailė, teatras, choreografija, kūno kultūra);
- ✓ dalykai turintys 2 savaitines pamokas dedami kas antra diena;
- ✓ po kūno kultūros nepageidaujama daryti matematikos;

#### **Trūkumai:**

- ✓ Nesuformuojamos pamokos pagal moksleivių pasirinkimą fiksuojant kiekvienoje grupėje moksleivių, kurie lankys šį dalyką, kodus.
- ✓ Nesuformuojama kasdieninis tvarkaraščio koregavimas ir vestų pamokų apskaita.

### **“EventPro” tvarkaraščio sudarymo programa**

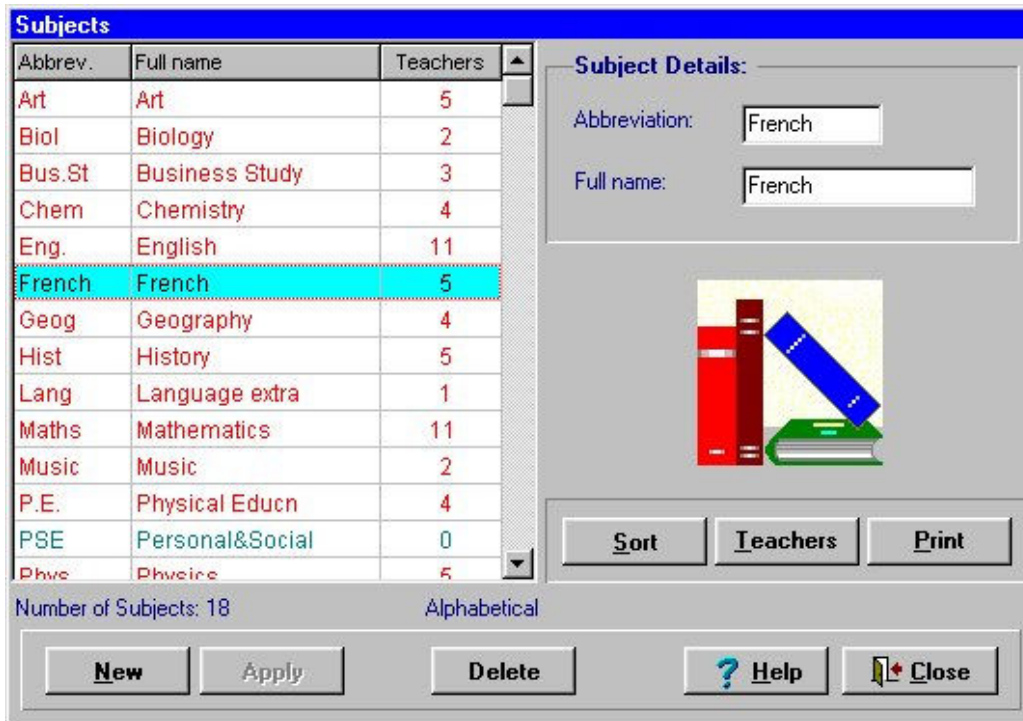
#### **Privalumai:**

- ✓ Pagerinta sąsaja su vartotoju. EventPro 2002 naudoja ACCESS 2000 DB leidžia vartotojams naviguoti po programą greičiau ir patogiau. Sutrumpinimai leidžia greitai peršokti iš vieno tabo į įvykių tvarkaraščio sudarymo detales, taip pat į iš anksto nustatytas vietas bei įvykio šaltinius.
- ✓ ACCESS DB priėjimas – pagerintas duomenų prieinamumas.
- ✓ 32 bitų sistema – leidžia išnaudoti maksimaliai OS 32-bit charakteristikas.
- ✓ Pagerintas saugumas – administratoriaus teisės
- ✓ Suteikia galimybę grupuoti vartotojus.
- ✓ Automatinis pranešimų generavimas suderinamas su M.A.P.I elektroniniu paštu.
- ✓ Vaizdo optimizavimas – planas mėnesiui, savaitei, nurodytam periodui.
- ✓ Nustatymo universalumas.
- ✓ Nuoseklus ir pilnas įvykių tvarkaraščio sudarymo langas. Patogi sąsaja su vartotoju – pelės paspaudimai, standartiniai controls'ai.
- ✓ Globalūs ir lokalūs apribojimai.
- ✓ Pilna finansinė ataskaita.
- ✓ Ataskaitų generavimas.

## “TimeTabler” programa

*TimeTabler* yra ir greitai ir vartotojui patogi programa, vystyta daugely metų. Jos pagrindas – didelė tvarkaraščio sudarymo patirtis.

*TimeTabler* nereikalauja specialių žinių IT srityje. Interfeisas labai paprastas ir naudojama standartinės interfeiso priemonės – mygtukai, iškrentantis meniu, žymės ir kt.



- ✓ *TimeTabler 4 for Windows* pasileis ant PC su Windows 95/98/2K.
- ✓ *TimeTabler* buvo suprojektuotas užtikrinti patogų tvarkaraščio sukūrimą išnaudojant mažai PC resursų.
- ✓ Galimybė rankiniu ir automatiniu būdu optimizuoti tvarkaraščius.
- ✓ Dėstytojų normalaus(kasdieninio) plano sudarymui
- ✓ Leidžiama susikurti pasirinkimo šablonus
- ✓ Interaktyvus pagalbos meniu.
- ✓ Interaktyvi pagalba sudarinėjant tvarkaraštį
- ✓ Automatinė ir interaktyvus visų reikalavimų sutikrinimo komponentas.

### 2.6 Darbo tikslas ir siejami privalumai.

Projektuojant sistema buvo siekiama sukurti kiek galima lankstesnę sistemą, kurią būtų galima nesunkiai išplėsti papildant naujomis funkcionalumo galimybėmis. Siekiant anksčiau paminėtų tikslų didelis dėmesys bus skiriamas tam, kad kuriama sistema būtų suprantama, lanksti ir teisingai optimizuotų tvarkaraščius (nes bus stebimi ir valdomi technologiniai procesai), paprasta naudotis vartotojams.

## **2.7 Kompiuterizuojamos sistemos funkcijos**

Kompiuterizuojamų funkcijų sąrašas:

- ✓ sudaryti, redaguoti ir atvaizduoti įvairių resursų paskirstymo tvarkaraščius;
- ✓ optimizuoti resursų paskirstymo planą, ir atlikti reikalingas plano korekcijas;
- ✓ pateikti resursų paskirstymo planą vartotojui įvairia forma (elektroniniu formatu, popieriuje, ir t.t.).

## **2.8 Nefunkciniai reikalavimai ir apribojimai**

### **2.8.1 Reikalavimai sistemos išvaizdai**

1. Sąsaja turėtų būti patraukli vartotojui: spalvos neturėtų "skaudinti" akių, turėtų būti suderintos tarpusavyje.

2. Vartotojo sąsaja privalo būti grafinė, naudojami tokie sąsajos elementai kaip:

- Langai;
- Dialogai;
- Meniu
- Įrankių juosta.

3. Lengvai skaitoma sąsaja. Kompiuterio ekrane visa informacija turi būti vaizduojama tvarkingai. Tekstinės informacijos išvedimui turi būti parinktas lengvai skaitomas šriftas.

### **2.8.2 Reikalavimai panaudojamumui**

4. Vartotojo sąsaja turi atspindėti taikomąją sritį: naudoti meniu ir mygtukus, pavadintus dalykinės srities sąvokomis.

5. Programa orientuota į Lietuvos rinką, t.y. sukurta programinė įranga turės lietuviškus tekstus ir pagalbos sistemas, tačiau turi būti numatyta galimybė kalbos keitimui.

6. Kiekvienai operacijai, kuri pasirenkama iš meniu, priskirti klavišų kombinaciją, kad patyrę sistemos vartotojai galėtų greičiau iškviešti norimas operacijas.

### **2.8.3 Reikalavimai vykdymo charakteristikoms**

7. TS darbo metu neturi būti neatlaisvinamų resursų (memory, resource leak'ų).

8. Programa turi veikti stabiliai, nepadarant žalos aplinkai, vartotojo programinei bei techninei įrangai.

9. TS turi būti suprojektuota taip, kad būtų galima ją lengvai papildyti naujomis funkcijomis, elementais.

#### **2.8.4 Reikalavimai veikimo sąlygoms**

Nėra.

#### **2.8.5 Reikalavimai sistemos priežiūrai**

15. Sistema turi veikti Windows 98/ Windows XP platformose.

16. Komunikavimas su DB turi vykti naudojant ODBC (o ne OCI) sąsają, kad ateityje būtų galima pereiti prie kito tipo DB serverio (pvz.: Sybase).

#### **2.8.6 Reikalavimai saugumui**

17. Sistema turi būti apsaugota nuo nesankcionuotų priėjimo prie jos resursų, t.y.: prisijungimas prie duomenų bazės turi būti apsaugotas slaptažodžiu, norint atlikti svarbesnį redagavimą ar valdymą reikia autentifikuoti vartotoją.

#### **2.8.7 Kultūriniai – politiniai reikalavimai**

18. Programos kontekste neturėtų būti žargonų, barbarizmų ar necenzūrinių žodžių.

#### **2.8.8 Teisiniai reikalavimai**

19. TS kurti naudoti tik licenzijuotą programinę įrangą.

20. Programa turi nepažeisti Lietuvos Respublikoje galiojančių įstatymų bei teisės normų.

21. Visos teisės į produktą priklauso užsakovui.

## 2.9 Rezultato kokybės kriterijai

Projektavimo kriterijų santrauka pateikta 1 lentelėje.

Prioritetas / Kriterijus	*****	****	***	**	*
Naudojamumas			+		
Saugumas		+			
Efektyvumas		+			
Teisingumas	+				
Patikimumas		+			
Palaikomumas			+		
Testuojamumas				+	
Lankstumas	+				
Suprantamumas	+				
Pakartotinis panaudojamumas			+		
Pernešamumas					+
Įsiliejamumas	+				

### *Projektavimo kriterijai*

Projektuojant sistemos architektūra turi būti naudojama Objektiškai-Orientuota projektavimo metodologija. Projektuojant turi būti naudojami stebėtojo, kompozitoriaus, dekoratoriaus, lankytojo, komandos bei kiti projektavimo šablonai

### 2.9.1 Įvertinimo būdai ir priemonės

Produkto kokybės įvertinimui bus naudojamos tokios priemonės:

- Funkcinis testavimas: testuojama, kaip sistema atlieka įvairias operacijas ir tikrinamas rezultatų teisingumas. Tačiau atliekant funkcinį testavimą dalis klaidų gali būti ir nepastebėta - jos turi būti surastos kitais testavimo metodais.
- Vartotojo sąsajos testavimas. Tikrinam meniu, duomenų pateikimo teisingumą. Šis testavimo metodas skirtas aptikti sąsajos netikslumus. Taip pat testuojant sąsają galima aptikti ir funkcionalumo klaidų, kurias detalizuojame atlikdami papildomą funkcinį testavimą ir vėliau - atitinkamų elementų testavimą.

- Elementų (modulių) testavimas. Šis testavimo metodas pagrindinai bus naudojamas realizavimo metu, jis atliekamas naudojant derinimo priemones (debuger). Taip pat šis metodas leis lokalizuoti funkciniam ar sąsajos testavime aptiktas klaidas.
- Palyginimas su jau sukurtais produktais: kompiuterine programa “Pamokos”, RECTOR,...

## **2.9.2 Kokybės vertinimo kriterijai**

- Korektiškumas – sistema turi dirbti korektiškai, turi atitikti vartotojo pageidavimus.
- Patikimumas – sistema turi būti patikima, negali jos darbas sutrikti dažniau nei numatyta.
- Patvarumas – turėtų būti sumažinta rizika duomenų sugadinimui po sistemos sutrikimo.
- Naudojimo lengvumas – ar reikia daug laiko norint apmokyti vartotoją dirbti su naująja sistema.
- Vykdyto greitis – per kiek laiko sistema sugebės atlikti jai numatytą užduotį.

## **2.10 Analizės išvados**

Kuriama programa turėtų pagerinti pamokų tvarkaraščio sudarymą.

Ji ne tik turės sudaryti tvarkaraštį, atsižvelgiant į nurodytus apribojimus, bet ir atliks daugelį optimizavimo veiksmų.

Sistema neturėtų nereikalauti didelių vartotojo pastangų, mokėjimo dirbti kompiuteriu bei su programa, lyginant darbą su programa „Mimoza“. Tvarkaraštį galės sudaryti netgi mokyklos sekretorė.

Išnagrinėjus kitas sistemas nustatyta jog:

- ✓ Visose tvarkaraščio sudarymo sistemose iš dalies arba visiškai trūksta optimizavimo metodų pasirinkimo.
- ✓ Nėra nutolusio sistemos administravimo galimybės.
- ✓ Nėra nutolusios DB nuskaitymo galimybės.

# **3. Sistemos reikalavimų specifikacija**

## **3.1 Apribojimai sistemai**

*Sistemos paskirtis:*

Šiuolaikinių technologijų progresas įpareigoja ieškoti efektyviausių sprendimų optimizuojant įstaigų/organizacijų veiklos planavimo procesą. Automatizuoti įmonės ar organizacijos procesų, svarbių įvykių planavimas ir kontrolė, išteklių apskaitos planai

gerina įmonės darbo kokybę, mažina valdymo kaštus. Jei organizacijos tikslai ir veikla yra mokomojo pobūdžio (švietimo org.) tiesiog gyvybiškai svarbus tinkamas resursų paskirstymas sudarant mokymo patalpų ir įrankių panaudojimo, mokytojų darbo planus. Tačiau daugeliu atveju planas sudaromas rankiniu metodu ir rezultatas daugeliu atitinka iš anksto nusibrėžtus reikalavimus. Pakeitimams ir analizei sugaištama daug brangaus laiko. Žvelgiant abstrakčiau kiekvieno visuomenės individo gyvenime tinkamai ir greitai suplanuojant kiekvieną etapą išvengiama daug problemų ir pasiekiami daug svaresnių rezultatų per trumpesnę laiką.

Užsiėmimų tvarkaraščių sudarymo sistema (UTSS) – tai programinė įranga skirta pamokų tvarkaraščių mokykloje sudarymui, kuri sudaro tvarkaraštį pagrindinių mokyklų mokiniams bei mokytojams.

Svarbiausias reikalavimas sistemai - sistema turi sumažinti vartotojų laiko sąnaudas pateikiant ir gaunant svarbiausią informaciją apie mokyklos tvarkaraštį bei mokytojų veiklą.

Pagrindiniai IS vartotojai – mokyklos administracija, mokytojai, mokiniai, mokinių tėvai.

Pagrindinės IS veiklos sritys – pateikti informacija apie mokyklos tvarkaraštį, bei mokytojų užimtumą.

Tvarkaraščio Sudarymo Sistema (toliau TSS) yra įrankis, automatizuojantis įstaigos ar organizacijos išteklių paskirstymo plano sudarymą. Jo paskirtis – sudaryti, redaguoti ir optimizuoti įvairių išteklių panaudojimo planus nustatytame laiko intervale, parodyti visus galimus išteklių išdėstymo, persidengimo, neišnaudojimo variantus, perspėti apie tai TSS vartotoją.

TSS tikslas - sudaryti, redaguoti ir optimizuoti įvairių išteklių panaudojimo planus.

Kompiuterizuojamų funkcijų sąrašas:

- ✓ sudaryti, redaguoti ir atvaizduoti įvairių resursų paskirstymo tvarkaraščius;
- ✓ optimizuoti resursų paskirstymo planą, ir atlikti reikalingas plano korekcijas;
- ✓ pateikti resursų paskirstymo planą vartotojui įvairia forma (elektroniniu formatu, popieriuje, ir t.t.).

### **3.2 Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys**

#### **Užsakovas**

Užsakovas: Klaipėdos miesto „A.Rubliovo“ pagrindinė mokykla

#### **Pirkėjas**

Tvarkaraščio Sudarymo Sistema (TSS) yra įrankis, automatizuojantis įmonės ar organizacijos išteklių paskirstymo plano sudarymą, todėl TSS galima naudotis visur, kur

reikalingas tam tikrų išteklių plano sudarymas. Sistemos pirkėjas – tai asmuo siekiantis automatizuoti išteklių paskirstymą įmonėje. Šiuo metu realus sistemos pirkėjas – Klaipėdos miesto „A. Rubliovo“ pagrindinė mokykla.

Produktas kuriamas užsakovo darbuotojo. Todėl jo finansavimas kur kas mažesnis nei esamų produktų. Be to, patogiau ir pigiau apmokyti vartotojus dirbti su sistema, kai ją kuria užsakovo darbuotojas. Sistema bus pritaikyta būtent „A. Rubliovo“ pagrindinei mokyklai, atsižvelgiant į vartotojų reikalavimus kūrimo etape.

### **Kiti sprendimus priimančios asmenys**

- ✓ Projekto vadovas: doc. Dalius Rubliauskas.
- ✓ Sistemos projektuotojas, programuotojas, testuotojas: Svetlana Neurova, (s\_next@mail.ru).



### **3.3 Vartotojai**

Programine įranga gali naudotis kiekvienas, mokantis dirbti kompiuteriu (mokantis naudotis standartine Microsoft Windows operacinės sistemos grafine vartotojo sąsaja). Tačiau darbas bus efektyvus, jei vartotojas turės bent minimalias žinias apie taikomąją sritį.

Vartotojams būtinos lietuvių kalbos žinios. Programa yra skirta lietuviškai kalbančių vartotojų grupei, todėl visi paaiškinimai pateikiami būtent šia kalba. Išskiriamos keturios vartotojų grupės: mokiniai/tėvai, mokytojai, administracija, TSS administratorius.

#### **Mokiniai**

Vartotojo sprendžiami uždaviniai (galimos funkcijos):

- ✓ Užsiregistruoti.
- ✓ Peržiūrėti ir atspausdinti jau sukurtus tvarkaraščius.
- ✓ Atspausdinti pasirinktą informaciją apie dalykus.

Šios vartotojų grupės poreikiai (nuomonė ir pageidavimai) turi žemiausią prioritetą.

#### **Mokytojai**

Vartotojo sprendžiami uždaviniai (galimos funkcijos):

- ✓ Užsiregistruoti.
- ✓ Peržiūrėti ir atspausdinti jau sukurtus tvarkaraščius.
- ✓ Peržiūrėti ir atspausdinti pasirinktą informaciją apie dalykus.
- ✓ Peržiūrėti ir atspausdinti pateiktą viešą informaciją apie mokytojus.

Šios vartotojų grupės poreikiai (nuomonė ir pageidavimai) turi vidutinį prioritetą.

#### **Administracija**

- ✓ Užsiregistruoti.
- ✓ Peržiūrėti, redaguoti, atspausdinti jau sukurtus tvarkaraščius.
- ✓ Sukurti naujus tvarkaraščius. Panaudojama visa surinkta ir prieinama informacija apie dalykus, mokinius, mokytojus.
- ✓ Peržiūrėti, redaguoti ir atspausdinti pasirinktą informaciją apie dalykus.
- ✓ Peržiūrėti pateiktą viešą informaciją apie mokytojus.
- ✓ Peržiūrėti apribojimus dėstomiems dalykams.
- ✓ Pareikšti atsiliepimus dėl programinės įrangos.

## TSS administratorius

Vartotojo sprendžiami uždaviniai (galimos funkcijos):

- ✓ Instaliuoti ir nustatyti tvarkaraščio sudarymo programinę įrangą.
- ✓ Konfigūruoti sistemą pagal kitų vartotojų atsiliepimus.

Šios vartotojų grupės poreikiai (nuomonė ir pageidavimai) turi aukščiausią prioritetą.

## 3.4 Apribojimai reikalavimams

### Apribojimai sprendimui

Sistema turėtų veikti Microsoft Windows XP operacinėje sistemoje.

Naudojamos mokyklos mokinių duomenų bazė (mokyklos pedagoginio darbo planas) serveris turi būti Msq1 .

Sukuriamos vidinės duomenų bazės (mokytojų, kabinetų, apribojimų) DB turi būti Msq1.

### Diegimo aplinka

Procesorius	Pentium 188 MHz;
RAM	16;
HDD	500 MB;
Monitorius	17 colių;
Vaizdo plokštė	1 MB;
Tinklo korta	10 Mbps.

*Techninės įrangos parametrai*

### Sistemos kūrimo biudžetas

Produktas kuriamas užsakovo darbuotojo. Sistema kuriama mokymosi tikslu ir neturi finansinės paramos. Sistemos projektuotojas naudojasi savo turima kompiuterine technika ir programine įranga. Be to, patogiau ir pigiau apmokyti vartotojus dirbti su sistema, kai ją kuria užsakovo darbuotojas. Sistema bus pritaikyta būtent „A. Rubliovo“ pagrindinei mokyklai, atsižvelgiant į vartotojų reikalavimus kūrimo etape. Sukurtas projektas galės būti toliau vystomas ir įdiegtas realiam darbui.

## Terminų žodynas

Terminas	Paaiškinimas
TSS	Tvarkaraščio Sudarymo Sistema. Tai programinė įranga skirta automatizuoti tvarkaraščio sudarymui
PI	Programinė Įranga
KTU	Kauno Technologijos Universitetas
OS	Operacinė Sistema.
Registruoti vartotojai	Vartotojai, prisiregistravę sistemoje, turintys vartotojo vardą bei slaptažodį
Administracija	Mokyklos direktorius, pavaduotoja, sekretorė, buhalterė, dalykų dėstytojai (ne klasių vadovai).
TSS administratorius	Mokyklos kompiuterinės įrangos inžinerius
Techninė įranga	Kompiuterinių įrenginių rinkinys (pvz., procesorius, monitorius), kurie priima duomenis, saugo, apdoroja ir išveda juos.
Programinio produkto vartotojai	Mokyklos bendruomenės nariai
IT	Informacinės Technologijos.

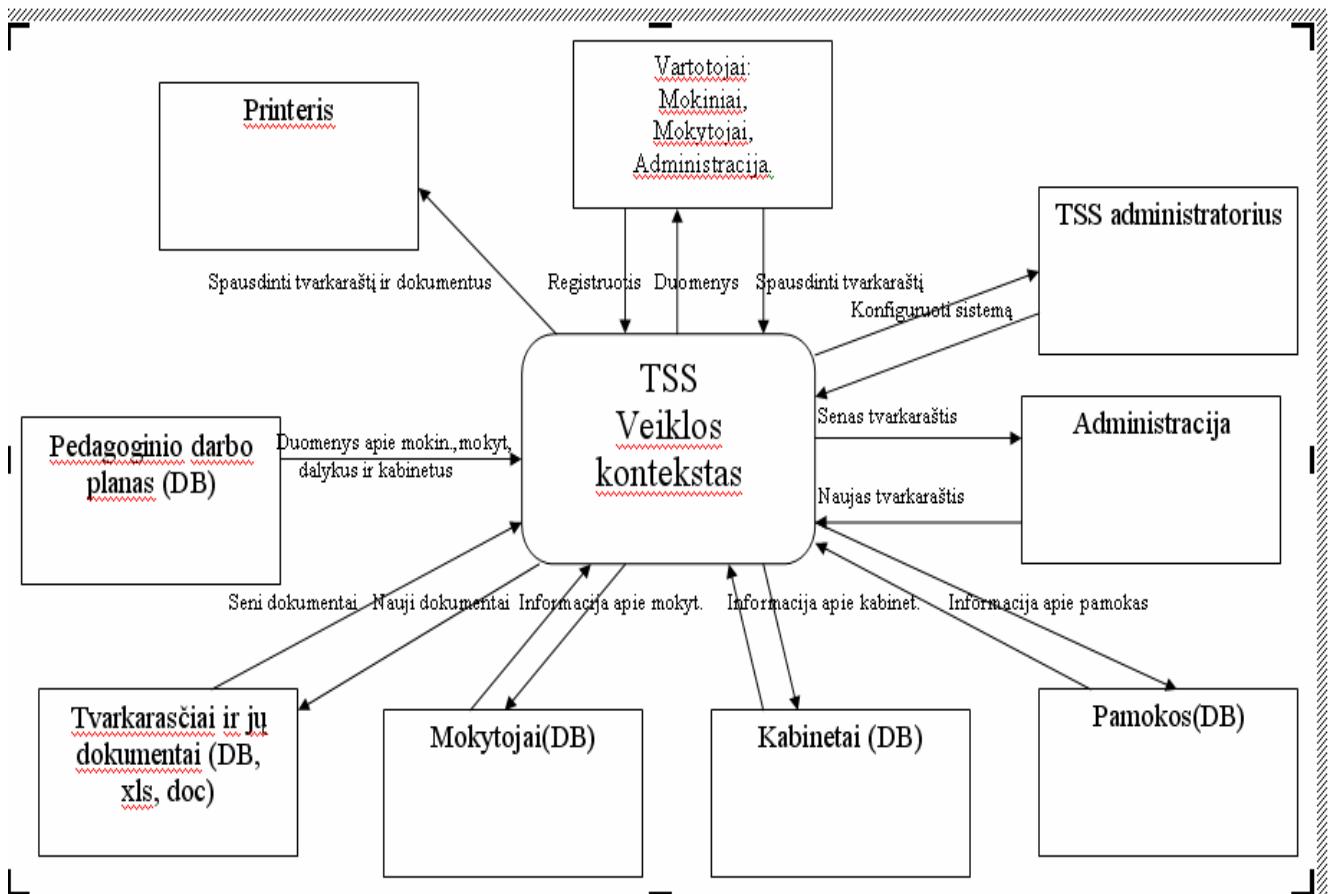
### **Svarbūs faktai**

Jau sukurti tvarkaraščių sudarymo moduliai yra specializuoti pagal įstaigos ar organizacijos veiklą. Taip pat dabartiniam vartotojui reikalingas adaptuotas specialiai „Ä. Rubliovo“ pagrindinėj mokyklai automatizuotas tvarkaraščio sudarymo įrankis.

### 3.5 Funkciniai reikalavimai

#### Veiklos sudėtis

#### Veiklos kontekstas

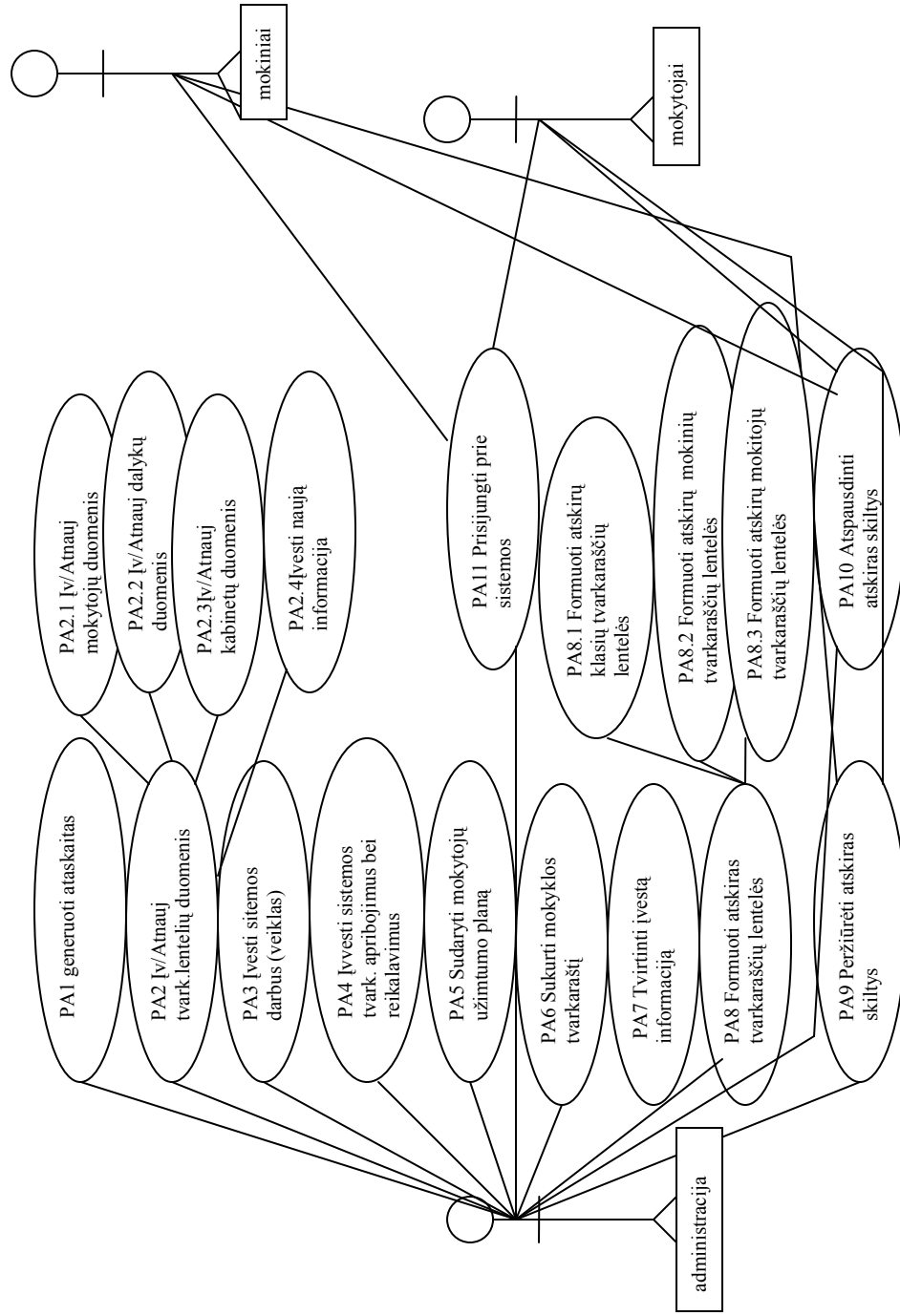


## Veiklos padalinimas

Eil.nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1.	Spausdinti tvarkaraštį, dokumentus	Tvarkaraštis, dokumentai .
2.	Duomenys iš pedagoginio darbo plano	Info. apie mokinius, mokytojus, dalykus ir kabinetus (in).
3.	Pakeisti tvarkaraštį.	Dokumentai ir tvarkaraštis xls, doc formatuose.(in) Pakeisti dokumentai ir tvarkaraštis xls, doc formatuose.(out)
4.	Sukurti tvarkaraštį.	Info. apribojimais(in). optimizuoti tvarkaraščiai.(out) tvarkaraščio dokumentai xls, doc formatuose.(out)
5.	Atnaujinti DB informaciją	Info apie mokinius, mokytojus, dalykus ir kabinetus (in). Info apie mokinius, mokytojus, dalykus ir kabinetus (out).
6.	Sukurti naują konfigūraciją	Nauja konfigūracija(in) Naujas vartotojas(in) Sena konfigūracija(out)
7.	Registruotis	Naujo vartotojo duomenys(in)

## Sistemos sudėtis

## Sistemos ribos



Sistemos panaudojimo atvejų diagrama

## **Panaudojimo atvejų sąrašas**

### **PA1 Generuoti ataskaitas**

Šis panaudojimo atvejis leidžia sistemos naudotojams peržiūrėti sistemoje kaupiamus duomenis ataskaitų pavidale. Ataskaitas galima gauti skirtingais pjūviais. Kiekvienas sistemos naudotojas gali peržiūrėti tik jam prieinamas ataskaitas. Panaudojimo atvejo veikėjas yra administracija.

### **PA2 Įvesti/atnaujinti tvarkaraščio lentelių duomenis**

Šis panaudojimo atvejis leidžia sistemos naudotojams įvesti ir koreguoti tvarkaraščio lentelių duomenis. Administratorius tai gali atnaujinti savo noru arba gavus specialų nurodymą iš jo vadovo (direktoriaus). Panaudojimo atvejo veikėjas yra administracija.

#### **PA2.1 Įvesti/atnaujinti mokytojų (darbuotojų) duomenis**

Šis panaudojimo atvejis leidžia sistemos naudotojams įvesti ir koreguoti sistemoje kaupiamus duomenis apie darbuotoją. Panaudojimo atvejo veikėjas yra administracija.

#### **PA2.2 Įvesti/atnaujinti dalykų duomenis**

Šis panaudojimo atvejis leidžia sistemos naudotojams įvesti ir koreguoti sistemoje kaupiamus duomenis apie dėstomus dalykus. Panaudojimo atvejo veikėjas yra administracija.

#### **PA2.3 Įvesti/atnaujinti kabinetų duomenis**

Šis panaudojimo atvejis leidžia sistemos naudotojams įvesti ir koreguoti sistemoje kaupiamus duomenis apie kabinetus. Panaudojimo atvejo veikėjas yra administracija.

#### **PA2.4 Įvesti įgautas žinias**

Šis panaudojimo atvejis leidžia sistemos naudotojams įvesti ir įgautas žinias. Turi būti atskirtos projekto vykdymo metu įgautos žinios/įgūdžiai bei ne projekto metu įgautos žinios. Taip pat turi būti fiksuojamas ir žinių šaltinis, rėmėjas ar kita informacija. Panaudojimo veikėjas yra administracija.

### **PA3 Įvesti sistemos darbus (veiklas)**

Šis panaudojimo atvejis leidžia sistemos naudotojams suformuoti sistemos vykdomus darbus/veiklas. Panaudojimo atvejo veikėjas yra administracija.

### **PA4 Įvesti sistemos tvarkaraščio apribojimus bei reikalavimus**

Šis panaudojimo atvejis apibrėžia sistemos naudojimo apribojimus bei reikalavimus. Panaudojimo atvejo veikėjas yra administracija.

### **PA5 Sudaryti mokytojų užimtumo planą**

Šis panaudojimo atvejis leidžia sistemos naudotojams pagal sistemoje įvestas veiklas ir reikalavimus suformuoti tinkamą mokytojų užimtumo planą. Panaudojimo atvejo veikėjas yra administracija.

### **PA6 Sukurti mokyklos tvarkaraštį**

Šis panaudojimo atvejis leidžia sistemos naudotojams sukurti pamokų tvarkaraštį sistemoje. Panaudojimo atvejo veikėjas yra administracija.

**PA7 Tvirtinti įvestą informaciją**

Šis panaudojimo atvejis leidžia sistemos naudotojams patvirtinti įvestos informacijos teisingumą sistemoje.

Panaudojimo atvejo veikėjas yra administracija

**PA8 Formuoti atskiros tvarkaraščių lentelės**

Šis panaudojimo atvejis leidžia sistemos administracijai sukurti atskirų klasių tvarkaraščių lentelės struktūrą.

Panaudojimo atvejo veikėjas yra administracija.

**PA8.1 Formuoti atskirų klasių tvarkaraščių lentelės**

Šis panaudojimo atvejis leidžia sistemos administracijai sukurti atskirų mokinių tvarkaraščių lentelės struktūrą. Panaudojimo atvejo veikėjas yra administracija.

**PA8.2 Formuoti atskirų mokinių tvarkaraščių lentelės**

Šis panaudojimo atvejis leidžia sistemos administracijai sukurti atskirų mokinių tvarkaraščių lentelės struktūrą.

Panaudojimo atvejo veikėjas yra administracija.

**PA8.3 Formuoti atskirų mokytojų tvarkaraščių lentelės**

Šis panaudojimo atvejis leidžia sistemos administracijai sukurti atskirų mokytojų tvarkaraščių lentelės struktūrą.

Panaudojimo atvejo veikėjas yra administracija.

**PA9 Peržiūrėti atskiras skiltys.**

Šis panaudojimo atvejis leidžia sistemos naudotojams peržiūrėti sistemos atskiras skiltys.

Panaudojimo atvejo veikėjai yra Mokytojai, Mokiniai (tėvai), Administracija.

**PA10 Atspausdinti atskiras skiltys.**

Šis panaudojimo atvejis leidžia sistemos naudotojams atspausdinti sistemos atskiras skiltys.

Panaudojimo atvejo veikėjai yra Mokytojai, Mokiniai (tėvai), Administracija.

**PA11 Prisijungti prie sistemos**

Šis panaudojimo atvejis leidžia sistemos naudotojams prisijungti prie sistemos. Sistemos naudotojas gali atlikti veiksmus su sistemoje kaupiamais duomenimis tik prisijungęs prie sistemos.

Panaudojimo atvejo veikėjai yra Mokytojai, Mokiniai (tėvai), Administracija.

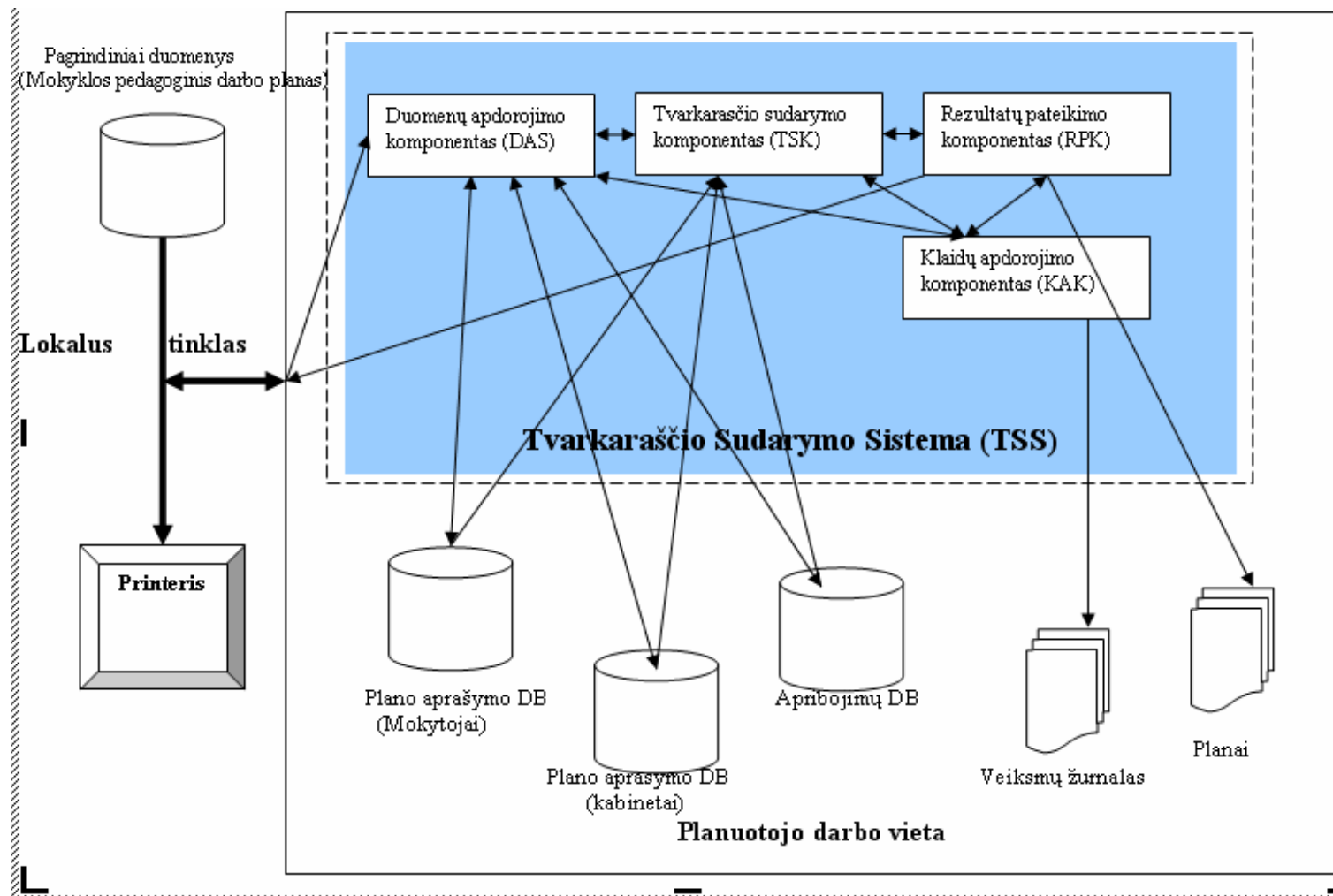
**PA12**

Šis panaudojimo atvejis leidžia sistemos administratoriui nustatyti sistemos konfigūraciją. Nustatoma naujai instaliuota sistema darbui. Taip pat atliekami nustatymai kiekvienam vartotojui.

Panaudojimo atvejo veikėjas yra TSS administratorius.



## Produkto vieta užsakovo sistemoje



## Funkciniai reikalavimai

Reikalavimas: 1	Reikalavimo tipas: 2	Panaudojimo atvejis: 1
Aprašymas:	TSS turi vartotojui suteikti galimybę atlikti ribotą (su nustatytomis teisėmis) informacijos paiešką pateiktuose duomenyse apie kabinetus, dalykus ir mokytojus bei mokinius.	
Pagrindimas:	Vartotojas gali peržiūrėti duomenis apie kabinetus, dalykus ir mokytojus bei mokinius, tačiau su apribojimais į tam tikrą informaciją.	
Tinkamumo kriterijus:	Duomenų formatą ir tipą pasirenka vartotojas	
Užsakovo tenkinimas: 5	Užsakovo netenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10 d.	

Reikalavimas: 2	Reikalavimo tipas: 2	Panaudojimo atvejis: 2
Aprašymas:	TSS turi vartotojui suteikti galimybę užsiregistruoti ir užregistruoti naujus sistemos vartotojus	

Pagrindimas:	Vartotojas nori kad jo suteikta ir naudojama informacija būtų konfidenciali. Taip pat siekiama išvengti nesankcionuoto naudojimosi programa.	
Tinkamumo kriterijus:	Saugumas pasiekiamas suteikiant vartotojams prisijungimo vardus bei slaptažodžius. Įvedus netinkamą slaptažodį apie tai turi pranešti sistema. Prisijungimo vardus suteikia sistemos administratorius.	
Užsakovo tenkinimas:	2	Užsakovo netenkinimas: 2
Priklausomybės:	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10 d.	

Reikalavimas: 3	Reikalavimo tipas: 2	Panaudojimo atvejis: 4
Aprašymas:	TSS turi vartotojui suteikti galimybę atspausdinti jau sukurtą tvarkaraštį bei su juo susijusius dokumentus	
Pagrindimas:	Vartotojas nori, kad būtų galima spausdinti ir pateikti tvarkaraštį popierine forma. Tai vienas iš pagrindinių reikalavimų	
Tinkamumo kriterijus:	Spausdinimas atliekamas lokaliame spausdintuve. Kokius dokumentus ir kaip spausdinti pasirenka vartotojas	
Užsakovo tenkinimas:	2	Užsakovo netenkinimas: 2
Priklausomybės:	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10d.	

Reikalavimas: 4	Reikalavimo tipas: 2	Panaudojimo atvejis: 5
Aprašymas:	TSS turi vartotojui suteikti galimybę vartotojui pateikti tvarkaraščio sudarymo apribojimus.	
Pagrindimas:	Vartotojas nori, kad būtų galima įvesti sudaromo tvarkaraščio apribojimus pagal nustatytą tvarką. Kiti apribojimai turi būti suteikiami pagal nutylėjimą. Jei apribojimų nesilaikoma, sistema automatiškai praneša apie jų pažeidimus.	
Tinkamumo kriterijus:	Apribojimai suteikiami kabinetams, laikui, mokytojų apkrautumui ir pan. Norimi apribojimai pateikiami tolesniuose reikalavimuose.	
Užsakovo tenkinimas:	1	Užsakovo netenkinimas: 1
Priklausomybės:	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10 d.	

Reikalavimas: 5	Reikalavimo tipas: 2	Panaudojimo atvejis: 5
Aprašymas:	TSS turi vartotojui suteikti galimybę vartotojui pateikti tvarkaraščio sudarymo pagal kabinetus apribojimus.	
Pagrindimas:	Vartotojas nori, kad būtų galima įvesti sudaromo tvarkaraščio apribojimus kabinetams pagal nustatytą tvarką. Kiti apribojimai turi būti suteikiami pagal nutylėjimą. Jei apribojimų nesilaikoma, sistema automatiškai praneša apie jų pažeidimus.	
Tinkamumo kriterijus:	Suteikiami apribojimai auditorijai ✓ mokinių grupes ✓ specializuotos, kompiuterizuotos ✓ Grupių paskirstymas į pogrupius	

Užsakovo tenkinimas: 1	Užsakovo netenkinimas: 1
Priklausomybės: Nėra	Konfliktai: Nėra
Papildoma medžiaga: Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10 d.

Reikalavimas: 6	Reikalavimo tipas: 2	Panaudojimo atvejis: 5
Aprašymas:	TSS turi vartotojui suteikti galimybę vartotojui pateikti tvarkaraščio sudarymo pagal pamokas apribojimus.	
Pagrindimas:	Vartotojas nori, kad būtų galima įvesti sudaromo tvarkaraščio apribojimus pamokoms pagal nustatytą tvarką. Kiti apribojimai turi būti suteikiami pagal nutylėjimą. Jei apribojimų nesilaikoma, sistema automatiškai praneša apie jų pažeidimus.	
Tinkamumo kriterijus:	<p>Suteikiami apribojimai paskaitoms</p> <ul style="list-style-type: none"> <li>✓ Pagal svarbumą</li> <li>✓ mokinių grupes</li> <li>✓ Pagal mokinių klases</li> <li>✓ Pagal statistinius išankstinius apribojimus.</li> <li>✓ Pagal pamainą (pamaininių pamokų paskirstymas).</li> <li>✓ Pagal kabinetą (kiekviena pamoka privalo būti dėstoma būtent toje klasėje).</li> </ul>	
Užsakovo tenkinimas: 1	Užsakovo netenkinimas: 1	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija:	Užregistruotas 2004 m. kovo 10 d.	

Reikalavimas: 7	Reikalavimo tipas: 2	Panaudojimo atvejis: 5
Aprašymas:	TSS turi vartotojui suteikti galimybę vartotojui pateikti tvarkaraščio sudarymo pagal mokytojus apribojimus.	
Pagrindimas:	Vartotojas nori, kad būtų galima įvesti sudaromo tvarkaraščio apribojimus mokytojams pagal nustatytą tvarką. Kiti apribojimai turi būti suteikiami pagal nutylėjimą. Jei apribojimų nesilaikoma, sistema automatiškai praneša apie jų pažeidimus.	
Tinkamumo kriterijus:	<p>Suteikiami apribojimai mokytojams</p> <ul style="list-style-type: none"> <li>✓ Pamokų nuoseklumas(keletas pamoku iš eilės)</li> <li>✓ Pamokų pasikartojamumas (tos pačios pamokos tą pačią savaitės dieną).</li> <li>✓ Mokytojų apkrautumas(yra užimti būtent tuo metu).</li> <li>✓ Specialūs pageidavimai(1, 2, 4 pamoka).</li> </ul>	
Užsakovo tenkinimas: 1	Užsakovo netenkinimas: 1	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija:	Užregistruotas 2004m. kovo 10 d.	

Reikalavimas: 8	Reikalavimo tipas: 2	Panaudojimo atvejis: 6
Aprašymas:	TSS turi vartotojui suteikti galimybę patogiai konfigūruoti sistema iš kompiuterio.	
Pagrindimas:	Administratorius, instaliavęs sistemą, turi sukonfigūruoti sistemą, kad nauji vartotojai galėtų registruotis ir naudotis sistema.	
Tinkamumo kriterijus:	<p>Sistemos konfigūravime atliekami šie veiksmai:</p> <ul style="list-style-type: none"> <li>✓ Nustatomos lokalsios vartotojų duomenų bazės parametrai</li> </ul>	

	<ul style="list-style-type: none"> <li>✓ Sukuriamas TSS konfiguracionis failas.</li> <li>✓ Sukuriami klaidų pranešimų templaite failai, dialogo templaite failai – vartotojų.</li> <li>✓ Startuojamas duomenų surinkimo servisas.</li> </ul>
Užsakovo tenkinimas: 2	Užsakovo netenkinimas: 2
Priklausomybės: Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra
Istorija:	Užregistruotas 2004 m. kovo 10 d.

Reikalavimas: 9	Reikalavimo tipas: 2	Panaudojimo atvejis: 7
Aprašymas:	TSS turi vartotojui suteikti galimybę peržiūrėti jau sukurtus tvarkaraščius.	
Pagrindimas:	Vartotojas gali peržiūrėti duomenis apie tvarkaraštį bet kuriuo laiko momentu.	
Tinkamumo kriterijus:	Peržiūrint tvarkaraščius turi būti leista pasirinkti formatą: <ul style="list-style-type: none"> <li>✓ Lentelinis (formuojamas lentelėmis).</li> <li>✓ Pagal mokytojus.</li> <li>✓ Pagal kabinetus.</li> </ul>	
Užsakovo tenkinimas: 5	Užsakovo netenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004m. kovo 10d.	

Reikalavimas: 10	Reikalavimo tipas: 2	Panaudojimo atvejis: 11, 12
Aprašymas:	TSS turi vartotojui suteikti galimybę redaguoti ir sukurti naujus tvarkaraščius	
Pagrindimas:	Maketuotojas nori, kad būtų galima sukurti naują tvarkaraštį.	
Tinkamumo kriterijus:	Pagrindinis reikalavimas tas, kad tvarkaraščio sukūrimas ir redagavimas jau esamų atliekamas per griežtai nustatytą sąsają. Atliekami veiksmai interaktyviai sekami sistemos. Kiekvienas neleistinas veiksmas (netenkinantis duomenų ir sistemos apribojimų) informuojamas vartotojui. Tvarkaraštis turi būti optimizuojamas pagal nustatytus apribojimus, siekiant sumažinti "langų" skaičių. Optimizavimas atliekamas: <ul style="list-style-type: none"> <li>✓ Automatinio būdu (pasinaudojant vienu iš galimų optimizavimo algoritmų: iteraciniai, ...)</li> <li>✓ Pusiau automatinio būdu – optimizuojant dalį tvarkaraščio ranką, dalį autmotizuojant pagal nustatytus apribojimus</li> <li>✓ Rankiniu būdu – viską atlieka maketuotojas.</li> </ul> Rezultatai pateikiami elektronine forma (failais) ir fiksuojami veiksmų žurnalo faile.	
Užsakovo tenkinimas: 1	Užsakovo netenkinimas: 1	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2004 m. kovo 10 d.	

## **3.6 Projekto išėiga**

### **Atviri klausimai**

- ✓ Kuriamos vidinės duomenų bazės nėra tinkamai suprojektuotos: kai kurios DB lentelės netenkina 3 normalinių formų, egzistuoja perteklinių laukų.
- ✓ Išnagrinėti tik iteraciniai skaičiavimo algoritmai. Egzistuojantys sprendimai

### **Sistemos kurios gali būti nupirktos**

Pasaulinėje rinkoje egzistuoja daug tvarkaraščio sudarymo sistemų. Užsakovui

Populiariausios tvarkaraščio sudarymo sistemos:

- ✓ Kompiuterinė programa "Pamokos"(Lietuvoje)
  - ✓ CELCAT tvarkaraščio sudarymo programa. (pasaulyje)
  - ✓ MIMOSA tvarkaraščio sudarymo programa.(Lietuvoje ir pasaulyje)
  - ✓ RECTOR tvarkaraščio sudarymo programa.(Lietuvoje ir pasaulyje)
  - ✓ AVTOR +2 tvarkaraščio sudarymo programa. (pasaulyje)
  - ✓ "FastTrack" tvarkaraščio sudarymo programa. (pasaulyje)
- ir kt.

### **Komponentai kuriuos galima panaudoti**

Tokių komponentų nėra.

### **Ką galima nukopijuoti**

Nukopijuoti galima esamą TS sistemą ir ateityje papildyti naujomis funkcijomis.

## **3.7 Naujos problemos**

### **Problemos sukeltos įdiegimo aplinkai**

Sistemai veikti vartotojo kompiuteryje turi būti instaliuota DB Client ir MySQL ODBC (versija priklauso nuo DB serverio versijos), kad TS galėtų vykdyti duomenų mainus su pedagoginio mokymo plano DB.

### **Ar nauja sistema paveiks instaliuotas sistemas**

Programa turi veikti stabiliai, nepadarant žalos aplinkai, vartotojo programinei bei techninei įrangai.

## **Kokie apribojimai, galintys kliudyti naujai sistemai egzistuoja diegimo aplinkoje**

Jeigu vartotojo kompiuteris netenkina techninių reikalavimų, apibrėžtų 1.4.2. skyriuje, yra didelė tikimybė, kad sistemos darbas bus nekorektiškas.

## **Ar kas nors iš esamų vartotojų gali neigiamai nusiteikti prieš naują sistemą**

Sistemos vartotojai dažnai sprendžia apie sistemą pagal jos sąsają su vartotoju, o ne pagal funkcionalumą. Prasta vartotojo sąsaja įtakoja vartotojo daromas klaidas. Todėl vartotojo sąsajos projektavimas turi būti orientuotas į vartotoją.

## **3.8 Uždaviniai**

### **Kokie žingsniai reikalingi sistemai pateikti**

Sistemos gyvavimo ciklas: specifikavimas, projektavimas, komponentų programavimas ir jų testavimas, komponentų apjungimas ir visumos testavimas, pateikimas ir palaikymas. Kiekvienam etape vyksta dokumentavimas.

Užsakovui bus pateikta tik pilnai funkcionuojanti sistema, atitinkanti reikalavimų specifikaciją, turinti vartotojo dokumentaciją.

### **Vystymo etapai**

Pirmi vystymo etapai yra pilnai aprašyti šiame dokumente. Sekantys numatomi vystymo etapai apims naujų TS funkcijų palaikymo realizaciją.

Sistemos kūrimui ir projektavimui bus naudojami šie programiniai įrankiai:

- ✓ Microsoft Word 2000 – dokumentacijos ruošimui.
- ✓ Microsoft PowerPoint 2000 – produkto pristatymo ruošimui.
- ✓ Borland C++ Builder – sistemos realizavimui skirtas įrankis.
- ✓ Projektavimui ir programavimui naudojami asmeniniai personaliniai kompiuteriai.
- ✓ Papildoma informacija programos projektavimo ir programavimo klausimais gaunama iš projekto vadovo, užsakovo, Kauno Technologijos Universiteto bendramokslų, vyresnių kolegų, Interneto puslapių, papildomos literatūros.

## **3.9 Pritaikymas**

### **Kokius specialius reikalavimus turime esamiems duomenims „paimti“ bei procedūroms pritaikyti darbui su nauja sistema**

Nėra duomenų, kuriuos reikėtų perkelti iš esamos į naująją sistemą. Esamoms procedūroms pritaikyti specialūs reikalavimai nekeliami.

### **Kokie duomenys turės būti transformuoti, perkeliant į naują sistemą**

Tokių duomenų nėra.

### **3.10 Rizikos įvertinimas**

Produkto kūrimo nesėkmės gali atsiliiepti magistrinio darbo įvertinimui. Visiško produkto neįvykdymo atveju kyla pavojus tolesnėms projekto vykdytojo studijoms Kauno Technologijos Universitete.

Galimos kliūtys projektui įvykdyti: laiko stoka, žinių stoka, modulių nesuderinamumas bei kiti nenumatyti išoriniai veiksniai.

Projekte nagrinėjama sritis yra gana plati, darbo apimtis yra nemaža, laiko yra skirta nedaug. Taip pat programavimo stadijoje išaiškėjus esant klaidų specifikacijoje, galimas visiškas projekto žlugimas, dėl labai riboto projekto atlikimo termino.

### **3.11 Kaina**

Projektas yra nefinansuojamas.

### **3.12 Vartotojo dokumentacija ir apmokymas**

Vartotojui bus pateikta sekanti dokumentacija:

- a. Sistemos funkcinis vadovas.
- b. Įžanginis sistemos vadovas.
- c. Detalus sistemos vadovas.
- d. Sistemos instaliavimo vadovas
- e. Sistemos administravimo vadovas
- f. Sistemos vartotojo vadovas

### **3.13 Perspektyviniai reikalavimai**

TS turės palaikyti kelis duomenų skaitymo iš DB interfeisus, pateikti tvarkaraštį įvairiais dokumentais.

## **4. Projektas**

### **4.1 Projekto tikslas**

TSS projekto tikslas - sudaryti, redaguoti ir optimizuoti įvairių išteklių panaudojimo planus.

Sistemos kompiuterizuojamų funkcijų sąrašas:

- ✓ sudaryti, redaguoti ir atvaizduoti įvairių resursų paskirstymo tvarkaraščius;
- ✓ optimizuoti resursų paskirstymo planą, ir atlikti reikalingas plano korekcijas;
- ✓ pateikti resursų paskirstymo planą vartotojui įvairia forma (elektroniniu formatu, popieriuje, ir t.t.)

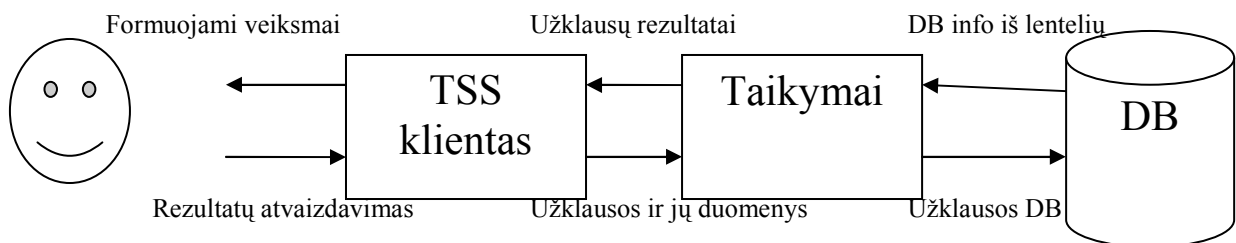
Adresatas (kam skirtas produktas):

TSS yra įrankis, automatizuojantis įmonės ar organizacijos išteklių paskirstymo plano sudarymą, todėl TSS galima naudotis visur, kur reikalingas tam tikrų išteklių plano sudarymas.

## 4.2 Panaudojimo atvejų ir realizacijos schema



Sistemos panaudojimo atvejų schema



Realizacijos schema



### **4.3 Projektavimo metodika**

Reikalavimų inžinerijos procesui yra naudojama sekanti metodika.

Galimybių analizė – ar įmanoma įvykdyti šią užduotį, ar realizavimas apsimokės finansiškai.

Reikalavimų išgavimas ir analizė – bendraujant su užsakovu interviu metodu yra išgaunami reikalavimai, jie analizuojami, po to tikslinami, stengiantis išgauti visus reikalavimus kuriamai sistemai.

Reikalavimų specifikacija – sudaromas dokumentas, kuriame aiškiai nurodoma kokia sistema turėtų būti, ką ji turėtų atlikti, kaip veikti.

Reikalavimų atestacija – reikalavimų patvirtinimas. Užsakovas turi patvirtinti reikalavimų specifikaciją. Patvirtinęs ją sistemos kūrėjai žinos kokią sistemą jie turi projektuoti ir kurti. Užsakovas žinos kokios jis sistemos gali tikėtis, patvirtinęs reikalavimų specifikaciją.

Programinės įrangos projektavimui ir kūrimui naudosime krioklio metodą. Įvykdę vieną etapą, jį atestuosime, t.y. patvirtinsime, ir pradėsime kitą etapą. Jei bus surasta klaidų grįšime atgal. Sistemos kūrimas bus paremtas prototipų kūrimu. Mes sukursime prototipą, kurį duosime užsakovui, jei jis tenkins užsakovą, jis bus vystomas toliau, taip iki galutinio produkto, kuris bus atiduodamas vartotojams.

Sistemos kūrimui ir projektavimui bus naudojami šie įrankiai:

- Microsoft Word 2000 – dokumentacijos ruošimui.
- Microsoft Powerpoint 2000 – produkto pristatymo ruošimui.
- Borland C++ Builder – sistemos realizavimui skirtas įrankis.

### **4.4 Turimi resursai**

Projektavimui ir programavimui naudojami asmeniniai personaliniai kompiuteriai. Projektas nėra finansuojamas.

Papildoma informacija programos projektavimo ir programavimo klausimais gaunama iš projekto vadovo, užsakovo, Kauno Technologijos Universiteto bendramokslų, vyresnių kolegų, Interneto puslapių, papildomos literatūros ir t.t.

### **4.5 Projekto personalas ir fazės**

Projektą įgyvendina šis personalas:

“A. Rubliovo” pagrindinė mokykla – užsakovas, kuris pateikia reikalavimus sistemai, taip pat projekto vadovas D. Rubliauskas, prižiūrintis kaip vykdomi darbai.

Svetlana Neurova – sistemos projektuotoja, programuotoja, testuotoja (sistemos kūrėja).

Projektas bus atliekamas keliomis fazėmis (skliausteliuose fazės atlikimo terminai):

1. Ruošiama projekto paraiška (2003 10 17 – 2004 01 22)
2. Sudaroma reikalavimų specifikacija (2003 01 22 – 2004 05 21)
3. Atliekamas sistemos projektavimas (2004 02 21 – 2005 05 21)
4. Atliekamas sistemos realizavimas (2005 05 21 – 2005 08 01)
5. Atliekamas sistemos testavimas (2005 08 01 – 2006 01 01)
6. Atliekamas eksperimentinis sistemos diegimas (2004 09 01 – 2005 12 23)
7. Galutinis produktas pateikiamas vartotojui. (nuo 2006 01 01)

## **4.6 Projekto procesų architektūra**

Šiuo metu daroma prielaida, kad TSS veiks 1 proceso ribose:

Kliento procesas: Tvardaraštis.exe – 1 pagrindinė gija, kurioje apdorojami sistemos įvykiai ir vartotojo elgsena. Duomenų nuskaitymas iš DB vyksta sinchroniškai.

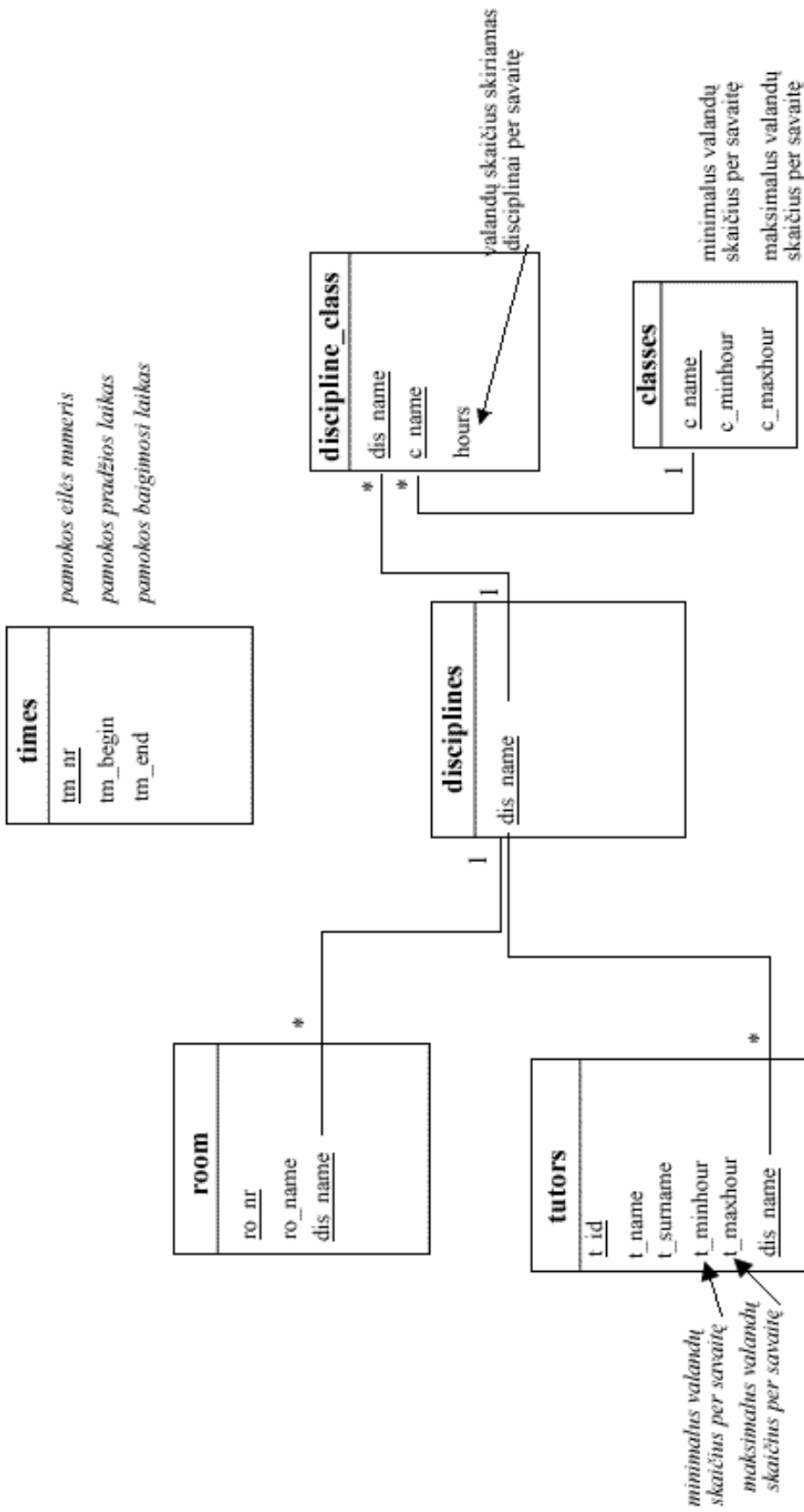
## **4.7 Projekto standartai**

Projektuojant sistemos architektūra naudoti Objektiškai - Orientuota projektavimo metodologija. Projektuojant naudojami stebėtojo, kompozitoriaus, strategijos, dekoratoriaus, lankytojo bei kiti projektavimo šablonai.

## **4.8 Duomenų bazės schema**



RELIACINIS MODELIS





## Tarpinės duomenų bazės galimas sukūrimo (SQL) skriptas.

```
-- phpMyAdmin SQL Dump
-- version 2.6.1-pl3
-- http://www.phpmyadmin.net
--
-- Darbinė stotis: localhost
-- Atlikimo laikas: 2005 m. Gruodžio 29 d. 15:40
-- Serverio versija: 4.1.7
-- PHP versija: 5.0.2
--
-- Duombazė: `tvarkarastis`
--
-- -----
--
-- Sukurta duomenų struktūra lentelei `classes`
--
CREATE TABLE `classes` (
  `c_name` varchar(4) NOT NULL default '',
  `c_minhour` int(10) unsigned NOT NULL default '0',
  `c_maxhour` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`c_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `classes`
--

INSERT INTO `classes` VALUES ('5d', 24, 27);
INSERT INTO `classes` VALUES ('5c', 24, 27);
INSERT INTO `classes` VALUES ('5a', 24, 27);
INSERT INTO `classes` VALUES ('5b', 24, 27);
INSERT INTO `classes` VALUES ('6a', 27, 29);
INSERT INTO `classes` VALUES ('6b', 27, 29);
INSERT INTO `classes` VALUES ('6d', 27, 29);
INSERT INTO `classes` VALUES ('7a', 28, 30);
INSERT INTO `classes` VALUES ('7b', 28, 30);
INSERT INTO `classes` VALUES ('7c', 28, 30);
INSERT INTO `classes` VALUES ('7d', 28, 30);
INSERT INTO `classes` VALUES ('8a', 14, 14);
INSERT INTO `classes` VALUES ('8b', 14, 14);
INSERT INTO `classes` VALUES ('8c', 14, 14);
INSERT INTO `classes` VALUES ('8d', 14, 14);
INSERT INTO `classes` VALUES ('9a', 15, 15);
INSERT INTO `classes` VALUES ('9b', 15, 15);
INSERT INTO `classes` VALUES ('9c', 15, 15);
INSERT INTO `classes` VALUES ('9d', 15, 15);
INSERT INTO `classes` VALUES ('10a', 16, 17);
INSERT INTO `classes` VALUES ('10b', 16, 17);
INSERT INTO `classes` VALUES ('10c', 16, 17);
INSERT INTO `classes` VALUES ('10d', 16, 17);

-- -----
--
-- Sukurta duomenų struktūra lentelei `discipline_class`
--

CREATE TABLE `discipline_class` (
  `dis_name` varchar(45) NOT NULL default '',
```

```

    `c_name` varchar(4) NOT NULL default '',
    `hours` int(10) unsigned NOT NULL default '0',
    PRIMARY KEY (`dis_name`,`c_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `discipline_class`
--

INSERT INTO `discipline_class` VALUES ('Biologija', '5d', 6);
INSERT INTO `discipline_class` VALUES ('Andglø kalba', '5d', 6);
INSERT INTO `discipline_class` VALUES ('Chemija', '5d', 6);
INSERT INTO `discipline_class` VALUES ('Dailė', '5d', 8);

-----

--
-- Sukurta duomenų struktūra lentelei `disciplines`
--

CREATE TABLE `disciplines` (
  `dis_name` varchar(45) NOT NULL default '',
  PRIMARY KEY (`dis_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `disciplines`
--

INSERT INTO `disciplines` VALUES ('Andglø kalba');
INSERT INTO `disciplines` VALUES ('Biologija');
INSERT INTO `disciplines` VALUES ('Chemija');
INSERT INTO `disciplines` VALUES ('Dailė');
INSERT INTO `disciplines` VALUES ('Ekonomika');
INSERT INTO `disciplines` VALUES ('Etika');
INSERT INTO `disciplines` VALUES ('Fizika');
INSERT INTO `disciplines` VALUES ('Geografija');
INSERT INTO `disciplines` VALUES ('Informatika');
INSERT INTO `disciplines` VALUES ('Istorija');
INSERT INTO `disciplines` VALUES ('Kūno kultūra');
INSERT INTO `disciplines` VALUES ('Lietuviø Kalba');
INSERT INTO `disciplines` VALUES ('Muzika');
INSERT INTO `disciplines` VALUES ('Rusø kalba');
INSERT INTO `disciplines` VALUES ('Vokieèiø kalba');

-----

--
-- Sukurta duomenų struktūra lentelei `rights`
--

CREATE TABLE `rights` (
  `r_id` int(10) unsigned NOT NULL auto_increment,
  `r_name` varchar(45) NOT NULL default '',
  PRIMARY KEY (`r_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Sukurta duomenų kopija lentelei `rights`
--

-----

```

```

--
-- Sukurta duomenų struktūra lentelei `room`
--

CREATE TABLE `room` (
  `ro_nr` int(10) unsigned NOT NULL default '0',
  `ro_name` varchar(45) NOT NULL default '',
  `dis_name` varchar(45) NOT NULL default '',
  PRIMARY KEY (`ro_nr`,`dis_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `room`
--

INSERT INTO `room` VALUES (110, 'Matematikos kab.', 'Muzika');

-----

--
-- Sukurta duomenų struktūra lentelei `times`
--

CREATE TABLE `times` (
  `tm_nr` int(10) unsigned NOT NULL default '0',
  `tm_begin` time NOT NULL default '00:00:00',
  `tm_end` time NOT NULL default '00:00:00',
  PRIMARY KEY (`tm_nr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `times`
--

INSERT INTO `times` VALUES (1, '13:30:00', '13:30:00');
INSERT INTO `times` VALUES (2, '13:30:00', '16:30:00');

-----

--
-- Sukurta duomenų struktūra lentelei `tutors`
--

CREATE TABLE `tutors` (
  `t_id` int(10) unsigned NOT NULL auto_increment,
  `t_name` varchar(45) NOT NULL default '',
  `t_surname` varchar(45) NOT NULL default '',
  `t_minhour` int(5) unsigned NOT NULL default '0',
  `t_maxhour` int(5) unsigned NOT NULL default '0',
  `dis_name` varchar(45) NOT NULL default '0',
  PRIMARY KEY (`t_id`,`dis_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;

--
-- Sukurta duomenų kopija lentelei `tutors`
--

INSERT INTO `tutors` VALUES (19, 'Valanetina', 'blabla', 20, 22, 'Chemija');
INSERT INTO `tutors` VALUES (20, 'Janina', 'Jonaitiene', 20, 20, 'Muzika');

-----

--
-- Sukurta duomenų struktūra lentelei `users`

```



```

--
CREATE TABLE `users` (
  `u_id` int(10) unsigned NOT NULL auto_increment,
  `u_name` varchar(45) NOT NULL default '',
  `u_surname` varchar(45) NOT NULL default '',
  `u_user` varchar(45) NOT NULL default '',
  `u_pass` varchar(45) NOT NULL default '',
  PRIMARY KEY (`u_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Sukurta duomenų kopija lentelei `users`
--

INSERT INTO `users` VALUES (1, 'Svetlana', 'Neurova', 'admin', '123');

-----

--
-- Sukurta duomenų struktūra lentelei `users_rights`
--

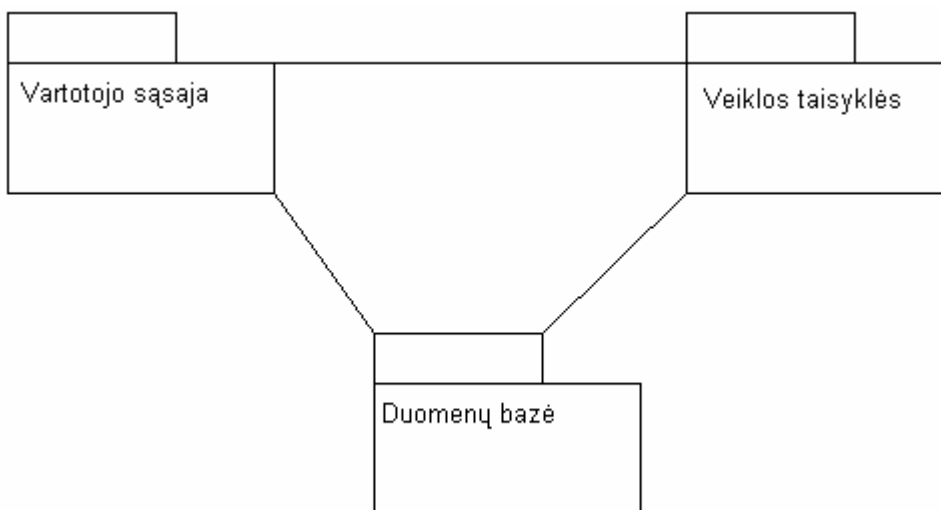
CREATE TABLE `users_rights` (
  `r_id` int(10) unsigned NOT NULL default '0',
  `u_id` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`r_id`,`u_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Sukurta duomenų kopija lentelei `users_rights`
--

```

#### 4.9 Projekto komponentų architektūra

TSS projektavimą sudaro 3 pagrindinių paketų architektūra: vartotojo sąsaja, veiklos taisyklės, standartinė biblioteka, DB. Ši architektūra suteikia galimybę lengvai vykdyti vieno lygmens pakeitimus, pridėti/pašalinti naują/seną funkcionalumą neličiant kito lygio.



*Bendra schema*

## **Lygmuo Vartotojo sąsaja**

*Vartotojo sąsajos* paketą sudaro formos (ribinės klasės), per kurias sistemos naudotojai bendrauja su sistema. Šiame sluoksnyje yra prisijungimo forma, mokytojų duomenų tvarkymo formos, sistemos informacijos tvarkymo formos, sistemos administravimo formos. Vartotojo sąsajos sluoksnis skirtas rezultatų, paruoštų taikomosios srities veiklos taisyklių sluoksnyje, apipavidalinimui ir pateikimui vartotojui.

## **Lygmuo Veiklos taisyklės**

*Veiklos taisyklių* paketą sudaro valdančios klasės, kuriomis įgyvendinamos veiklos taisyklės.

Visi reikalingi duomenys yra paaimami iš duomenų bazės naudojantis procedūromis, apdorojami remiantis taikomosios srities taisyklėmis, gauti rezultatai yra perduodami vartotojo sąsajos sluoksniui apipavidalinimui ir pavaizdavimui kliento pusėje. Vartotojo pateikti duomenys yra perduodami atgal taikomosios srities taisyklių sluoksniui, kur yra apdorojami ir procedūrų pagalba patalpinami į duomenų bazę.

Taikomosios srities *veiklos taisyklių* sluoksnyje yra objektai, užtikrinantys veiklos taisyklių vykdymą.

## **Lygmuo DB**

*Duomenų bazės paketas*. Šiame pakete realizuojamos DB lentelės, minimalūs apribojimai ir ryšiai tarp jų, generatoriai. DB sistemos duomenys, saugomi reliaciniame pavidale.

Šiame sluoksnyje yra saugojami sistemos duomenys reliaciniame pavidale. Sistemoje bus naudojama MySQL reliacinė duomenų bazių valdymo sistema. Duomenų paėmimui iš duomenų bazės, modifikavimui bei naujų duomenų įterpimui į duomenų bazę bus sukurtos. Bendravimas su duomenų baze bus vykdomas ODBC sujungimu.

## **4.10 Projekto vartotojo sąsajos komponentai**

### **Apibrėžimas**

Vartotojo sąsaja - sistemos modulis, atsakingas už sąsają su sistemos naudotoju. Jį sudaro sąsajos klasės.

Vartotojo sąsajos komponentas – tai įvedimo formų, meniu ir ataskaitų klasės, kurių pagalba sistemos naudotojas pasirenka norimus veiksmus, įveda į duomenis į sistema ar atlieka kitą veiksmą su sistema.

### **Atsakomybės**

Vartotojo sąsajos komponentas skirtas užtikrinti, kad sistemos naudotojas galėtų lengvai pasirinkti norimus veiksmus su sistema, įvesti reikiamus duomenis bei gauti reikiamus duomenis.

### **Apribojimai**

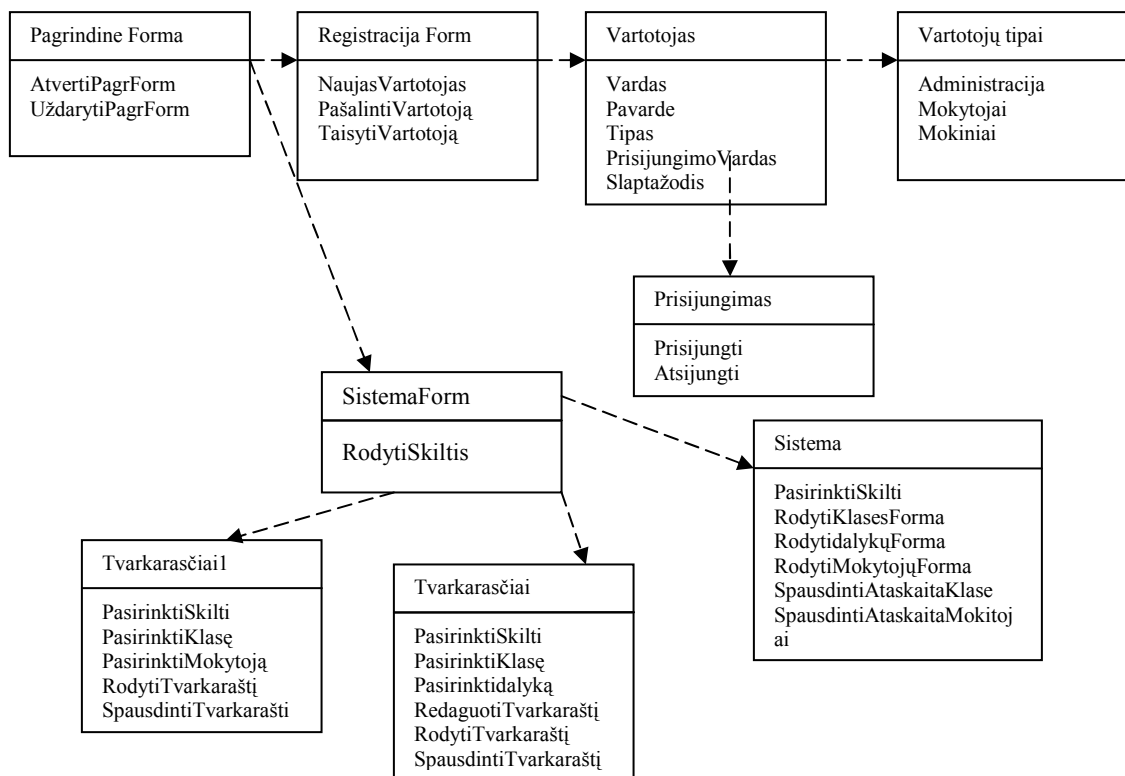
Turi turėti ryšį su veiklos taisyklių komponentu.

Turi užtikrinti apdorojamų duomenų korektiškumą.

Turi apsaugoti sistemos naudotoją nuo atsitiktinių klaidų atliekant svarbius veiksmus.

## 4.11 Projekto vartotojo sąsajos struktūra

Vartotojo sąsajos struktūra pateikiama vartotojo sąsajos diagrama.

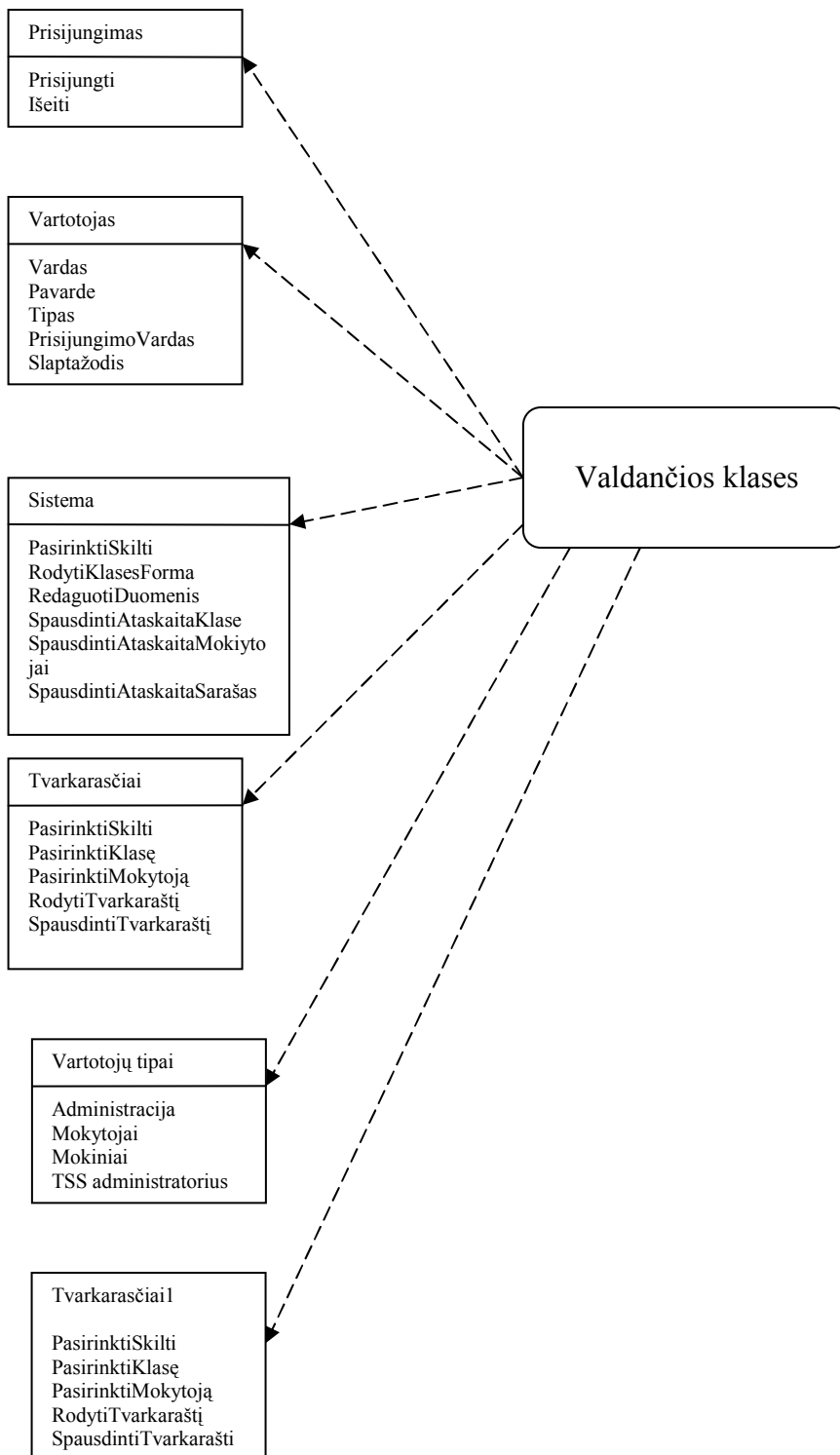


<b>Pavadinimas</b>	Prisijungimas
<b>Klasifikacija</b>	Forma
<b>Aprašymas</b>	Prisijungimo prie sistemos pagal vartotojo prisijungimo vardą ir slaptažodį.
<b>Sąsaja</b>	Forma sistemos vartotojui parodoma monitoriaus ekrane

<b>Pavadinimas</b>	Tvarkaraščiai
<b>Klasifikacija</b>	Forma
<b>Aprašymas</b>	Veiksmų su tvarkaraščiais forma (tik administracijos darbuotojams)
<b>Sąsaja</b>	Forma sistemos vartotojui parodoma monitoriaus ekrane
<b>Resursai</b>	DB forma
<b>Sąveikavimas</b>	Kai paspaudžiamas mygtukas Išrinkimas, duomenys perduodami veiklos taisyklių objektui .

<b>Pavadinimas</b>	Prisijungimas
<b>Klasifikacija</b>	Forma
<b>Aprašymas</b>	Forma skirta naujo vartotojo įvedimui, prisijungimui sistemoje.
<b>Sąsaja</b>	Forma sistemos vartotojui parodoma monitoriaus ekrane
<b>Resursai</b>	DB forma
<b>Sąveikavimas</b>	Kai paspaudžiamas mygtukas Prisijungti (Išeiti), veiksmai perduodami veiklos taisyklių objektui .
<b>Pavadinimas</b>	Forma Sąrašai
<b>Klasifikacija</b>	Forma
<b>Aprašymas</b>	Forma skirta įėjimui į sistemos skiltis .
<b>Sąsaja</b>	Forma sistemos vartotojui parodoma monitoriaus ekrane
<b>Resursai</b>	DB forma .
<b>Sąveikavimas</b>	Kai paspaudžiamas bet kuris esantis formoje mygtukas pereinama prie kitos formos
<b>Pavadinimas</b>	Vartotojas
<b>Klasifikacija</b>	Forma
<b>Aprašymas</b>	Pagrindinė sistemos forma.
<b>Sąsaja</b>	Forma sistemos vartotojui parodoma monitoriaus ekrane
<b>Resursai</b>	DB forma .
<b>Sąveikavimas</b>	Kai paspaudžiamas mygtukas Prisijungti, įvesti duomenys perduodami veiklos taisyklių objektui .

## 4.12 Valdančiosios klasės



## 5. Eksperementinis sistemos tyrimas

### 5.1 Eksperementinio diegimo aprašymas

#### Reikalavimai kompiuteriui

##### Minimali konfigūracija:

Procesorius	Pentium II 333MHz;
RAM	256 MB;
HDD	50 MB;
Displėjus	19 colių;
Video	8 MB;
Tinklo korta	10 Mb/s.

##### Rekomenduojama konfigūracija:

Procesorius	Pentium III 500MHz;
RAM	512 MB;
HDD	50 MB;
Displėjus	19 colių
Video	8 MB;
Tinklo korta	100 Mb/s.

#### Reikalavimai kompiuterio programinei įrangai

TSS sistemos funkcionavimui reikalinga sekanti programinė įranga.

##### GRAFINIAM TSS KLIENTUI:

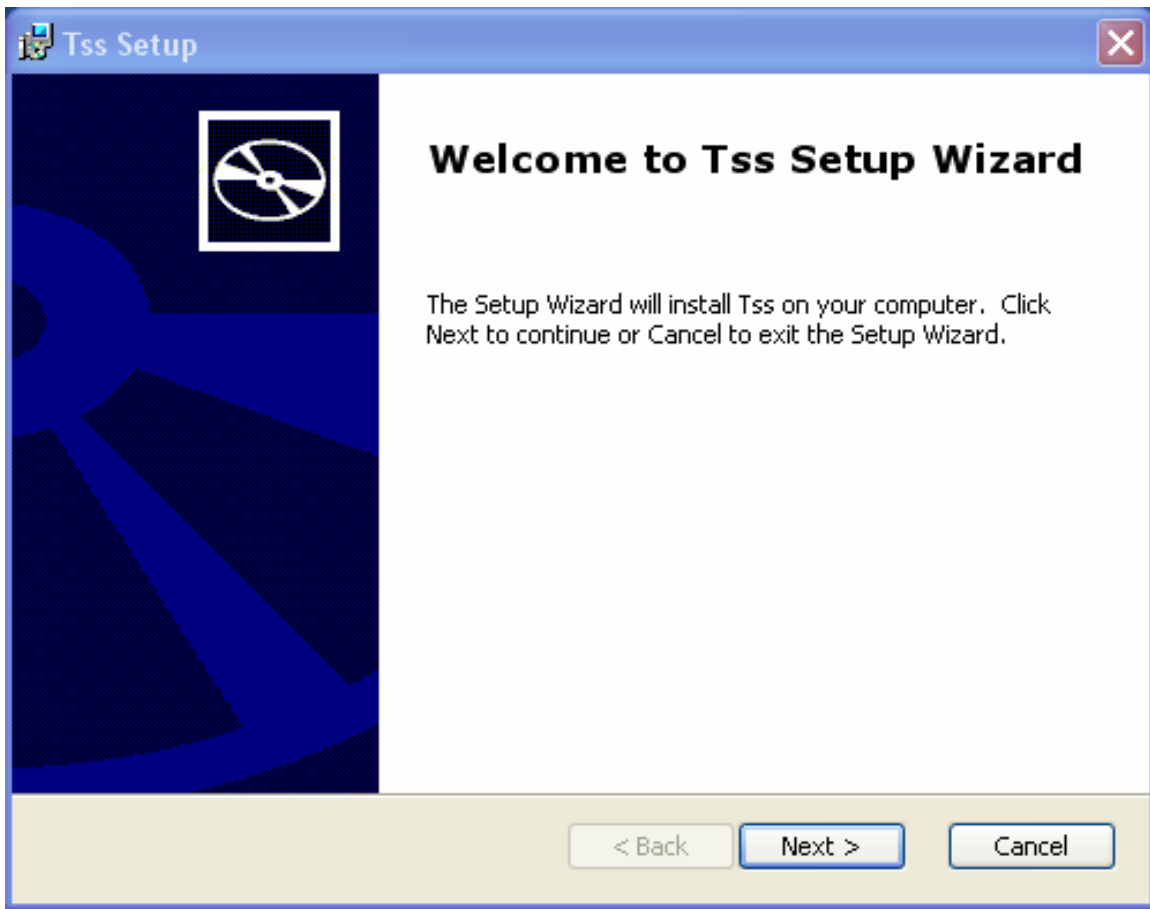
- Ryšys su MySQL serveriu;
- MySQL ODBC tvarkyklė
- sukurta schema duomenų lentelėms.
- ADO komponentai
- Borland C++ Builder vykdomosios bibliotekos.
- Windows 98 arba vėlesnės versijos

#### Instaliacija

##### TSS nustatymas

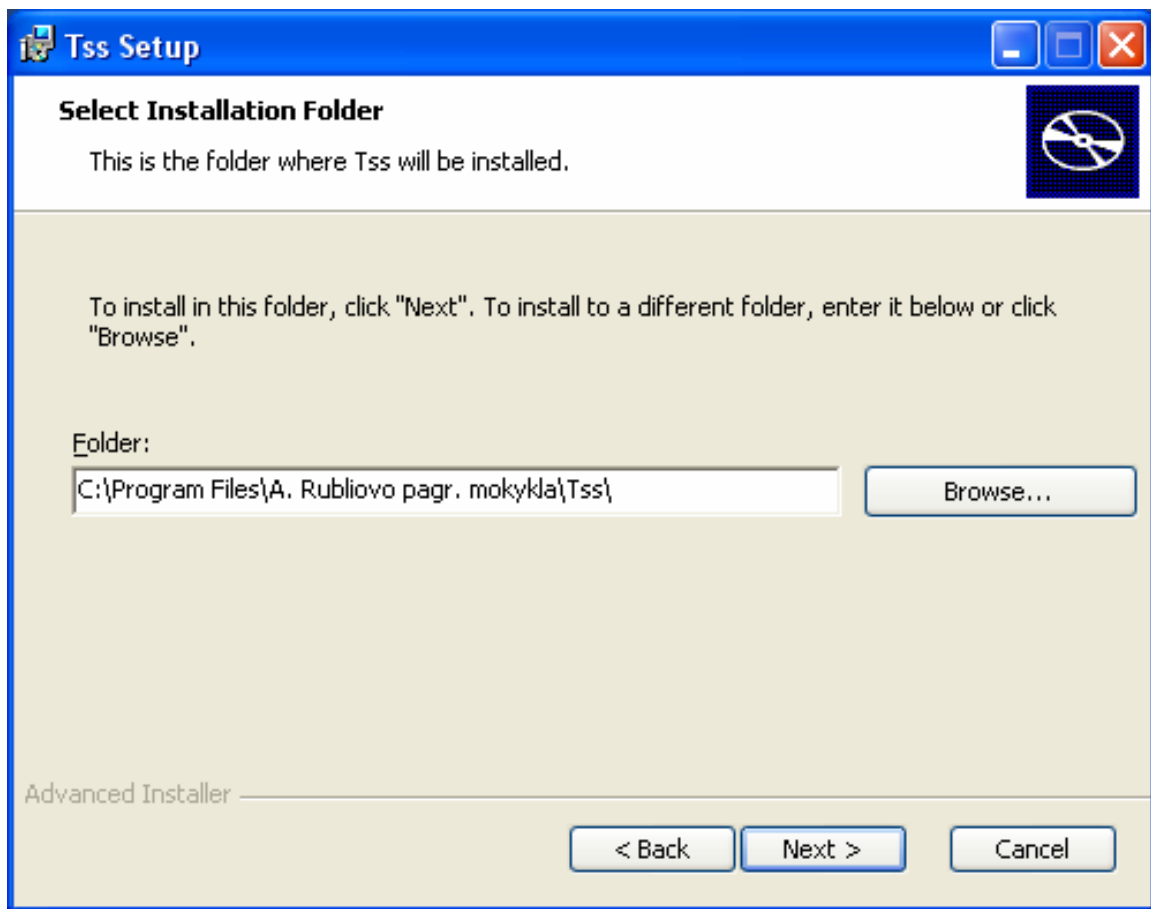
TSS klientą nustato sistemos administratorius.

TSS kliento nustatymas atliekamas programinės įrangos nustatymo (Install wizard) sugeneruoto paketo pagalba. Paketui startavus, matomas langas:



*Instaliavimo pradžia*

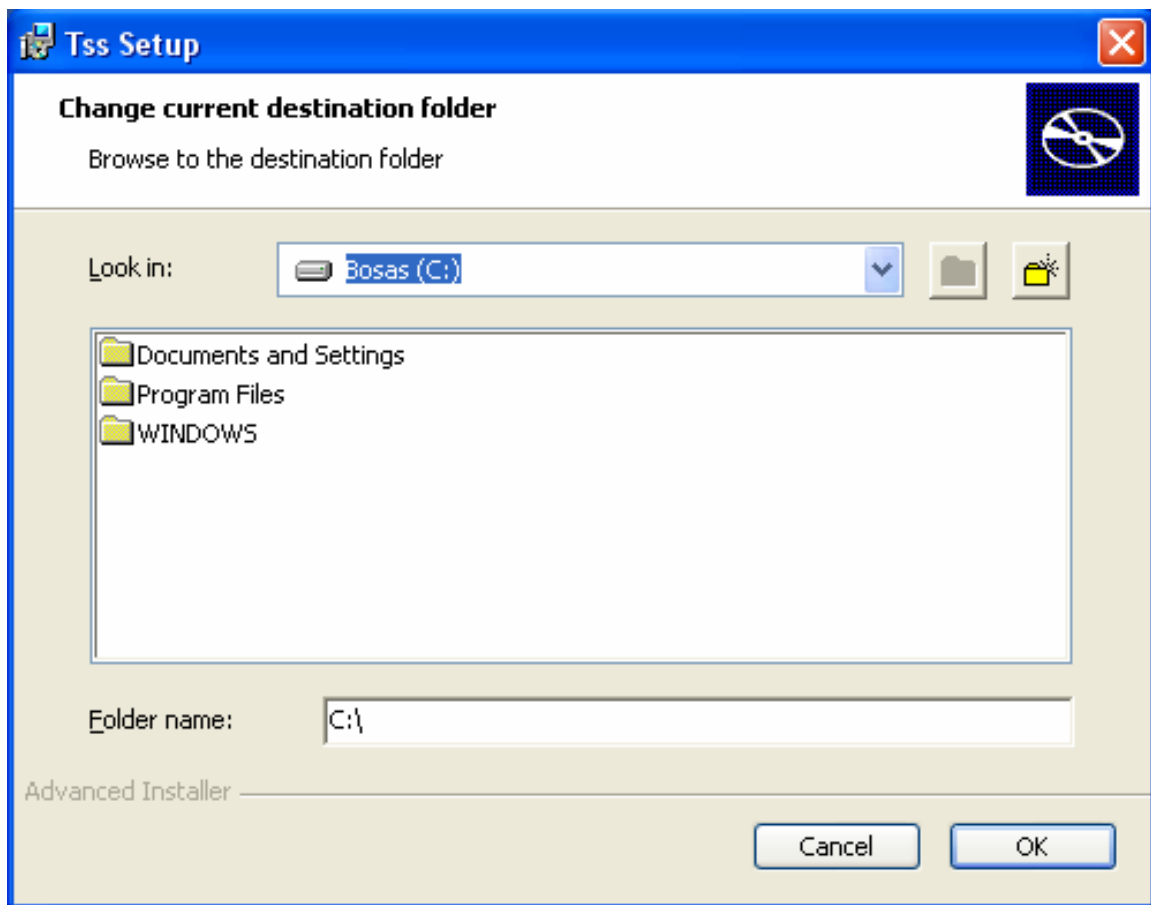
Pasirenkamas mygtukas [Next], perskaitoma programos licenzija. Jei vartotojas nesutinka su licenzijos dokumente išdėstyta informacija, jis bet kuriuo metu gali atšaukti instaliavimą . Perskaičius licenziją spaudžiamas [Yes] mygtukas. Suvedama informacija apie kompaniją, vartotoją. Pasirenkamas katalogas, kuriame bus saugomi programos failai:



*Katalogo pasirinkimas*

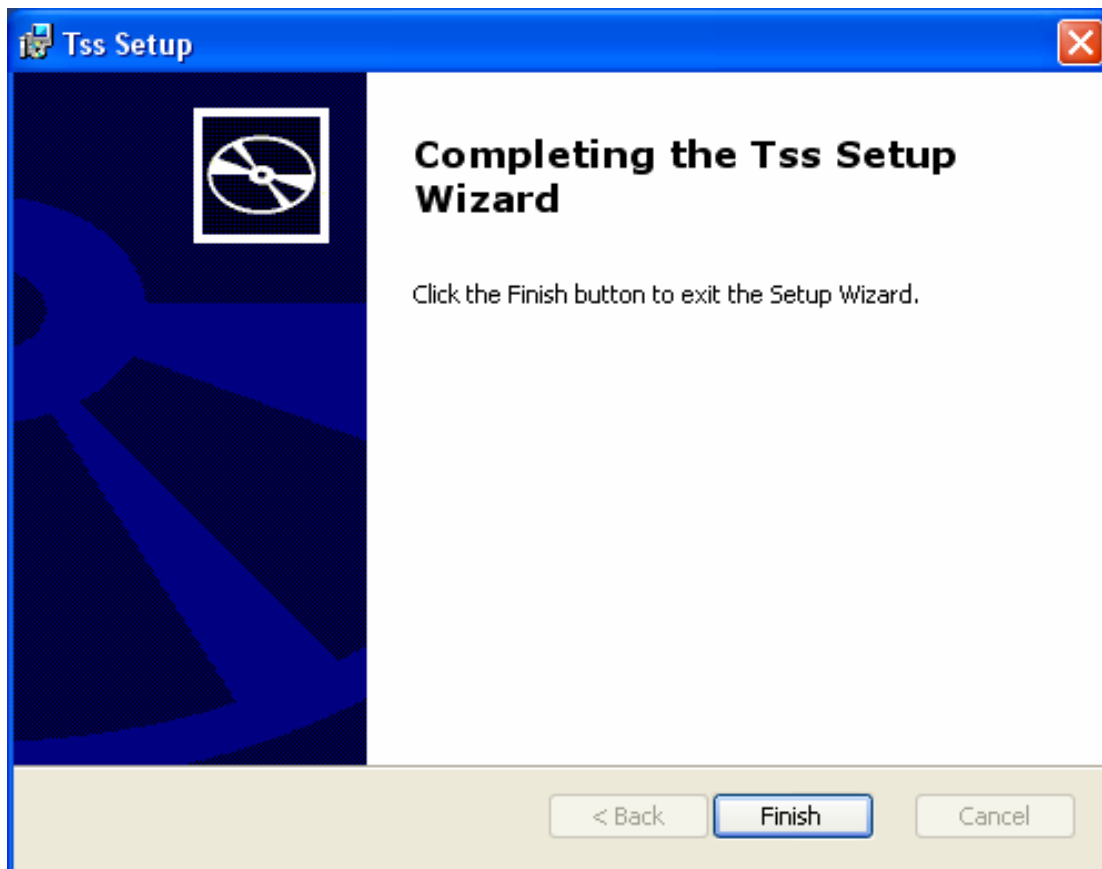
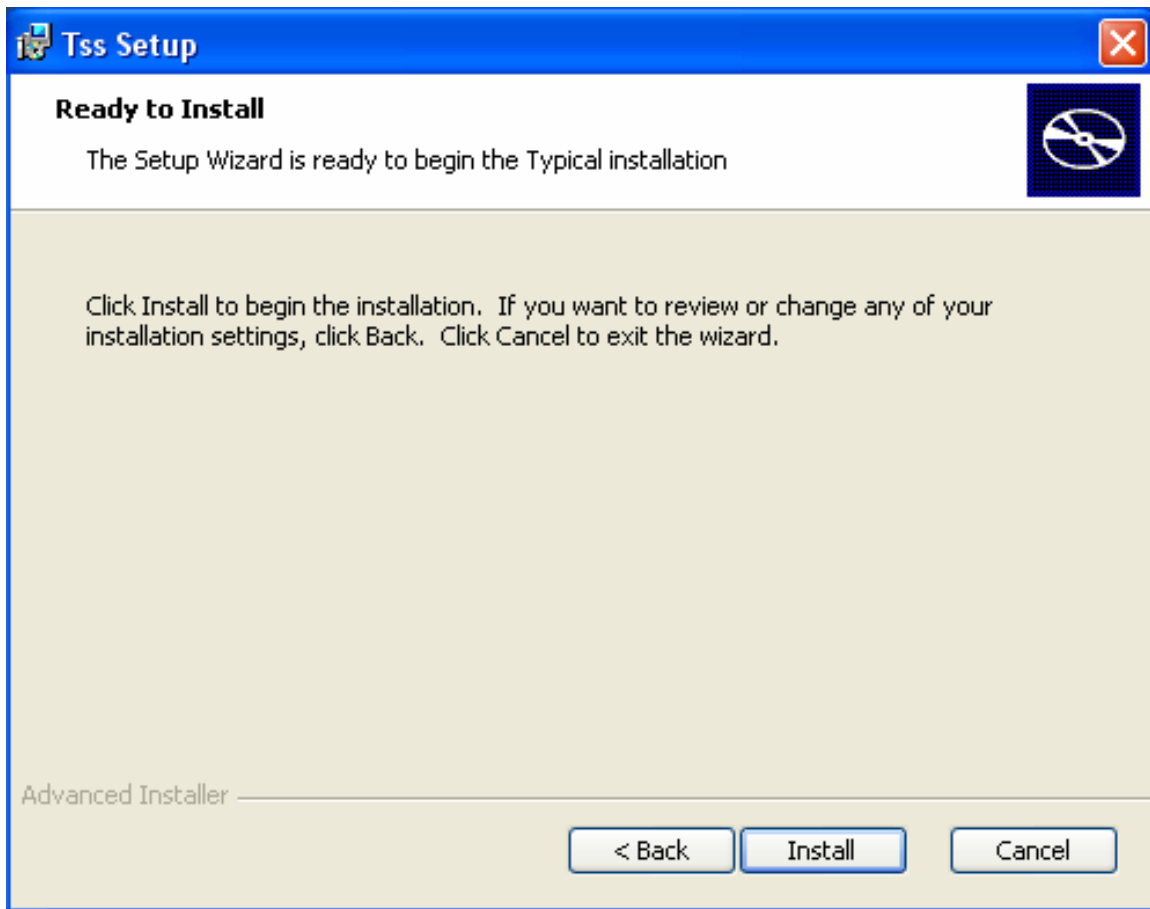
Paspaudus mygtuką [Browse] matomas failų sistemos langas, palengvinantis instaliavimo vietos pasirinkimą:





*Rodyklė po failų sistemą*

Išrinkus katalogą, spaudžiamas mygtukas [OK] ir [Next] ir pereinama prie nustatymo. Paspaudus mygtuką [Install], sistemos konfigūraciją baigiama nustatinėti ir parodomas sėkmingai instaliuotos sistemos instaliavimo pabaigos langas:



*Kliento instaliavimo pabaigos  
langas*

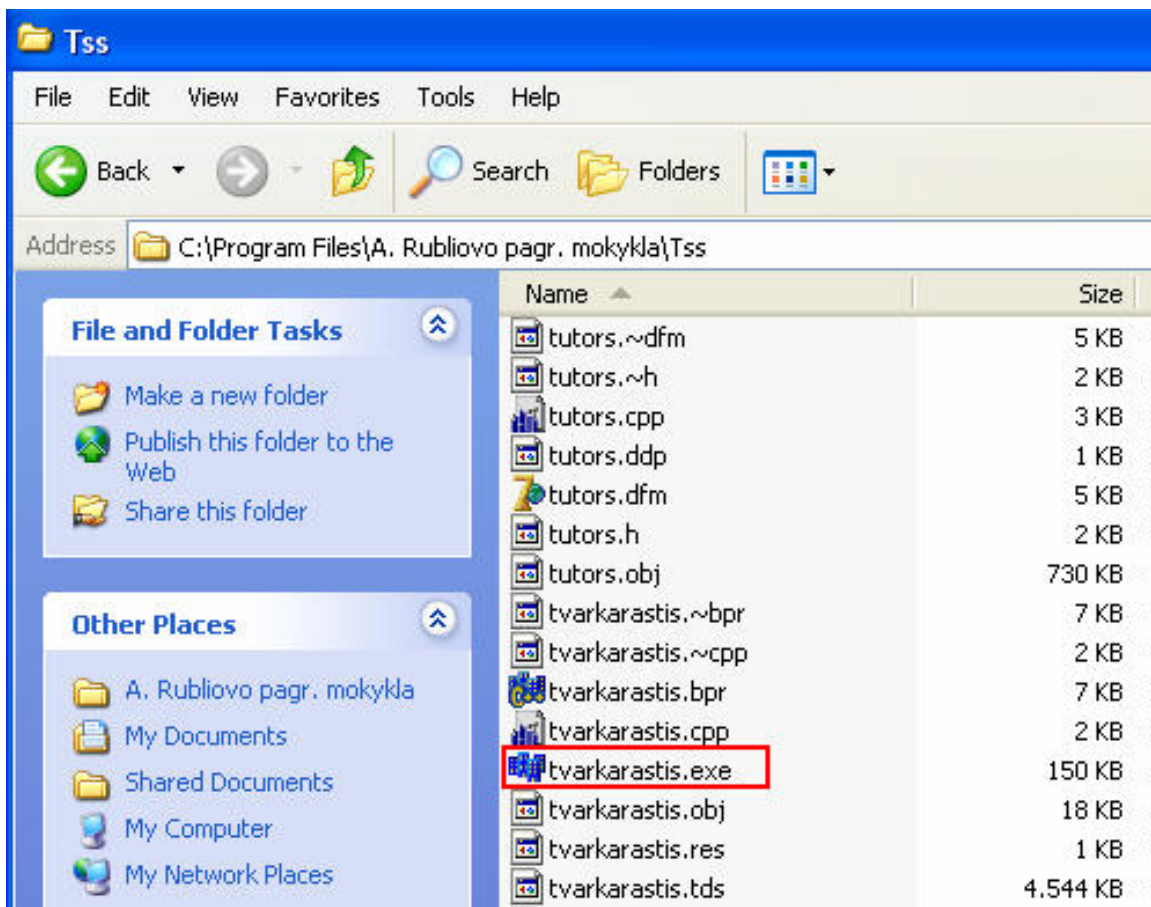
Sėkmingai nustačius TSS , sukuriami tokie resursai:

g. Programos pakė:



*TSS pakė*

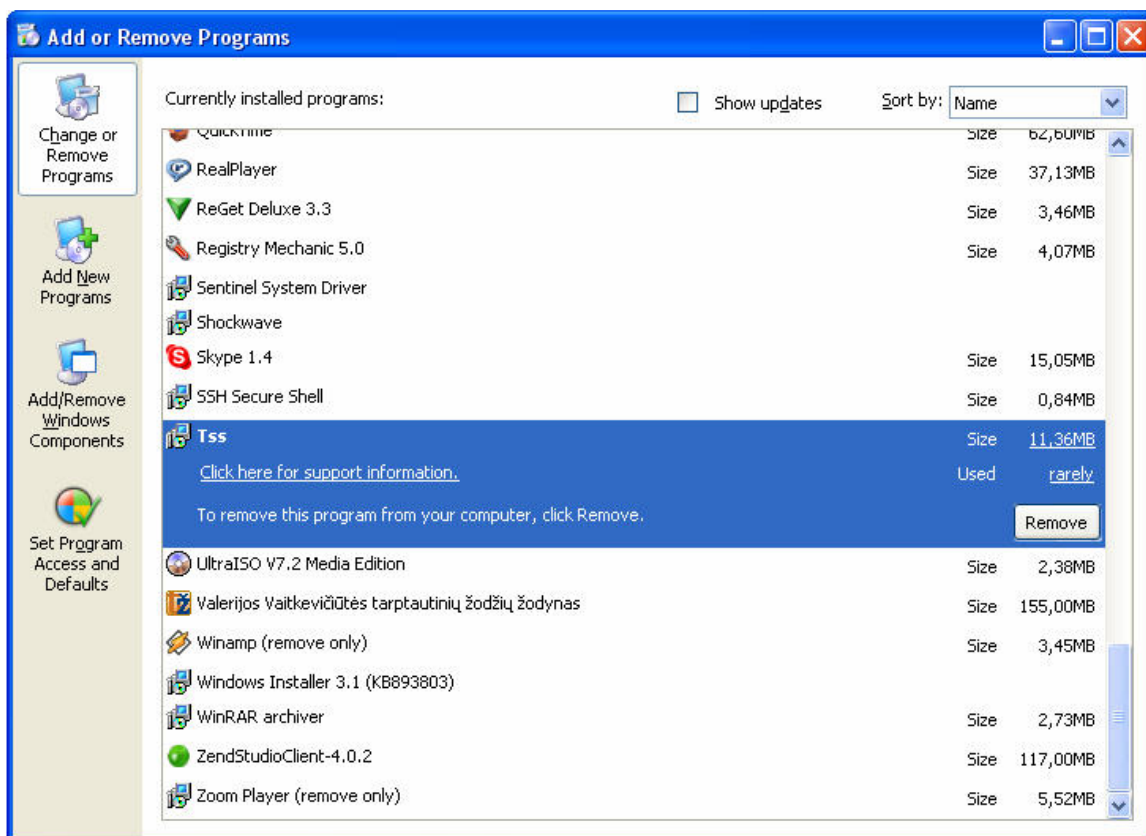
h. Kliento paleidžiamasis failas



*TSS paleidžiamasis failas*

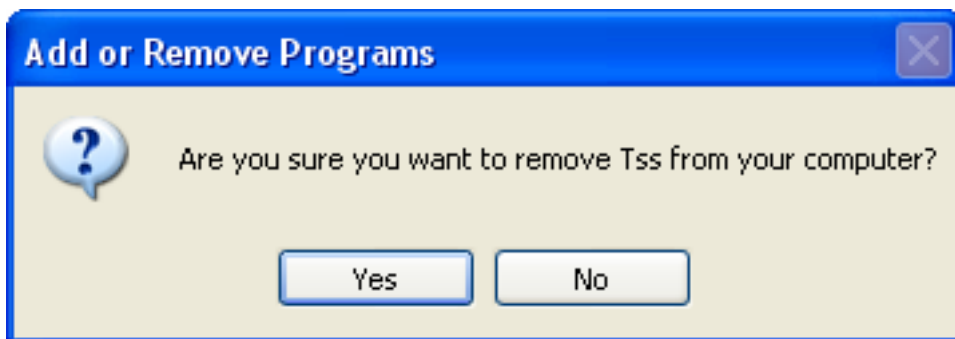
## Sistemos išinstaliavimas

Sistemos komponentų išinstaliavimas atliekamas sistemos administratoriaus. Sistemos išinstaliavimą atlieka standartinis IsUnist paketas. Kadangi instaliavimo metu buvo įrašinėjami instaliavimo veiksmai, todėl sistemos išinstaliavimas atliekamas gana aukštame lygyje. Pasirenkamas punktas “Remove”:



*Išinstaliavimo  
pradžia*

Sistemos išinstaliavimas laikomas sėkmingu, kai rezultate pamatomas pranešimų langas:

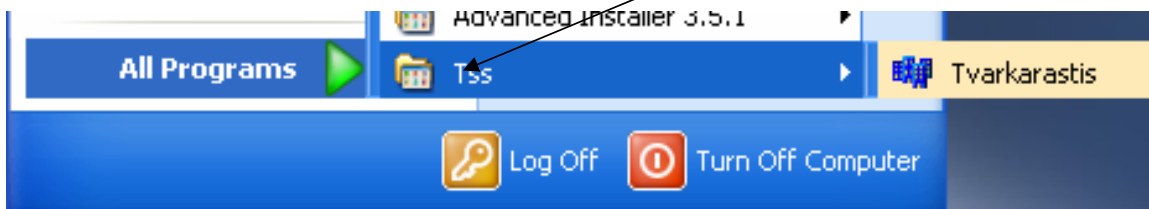


*Sistema sėkmingai  
išinstaliuota*

Po sistemos išvalymo reikia perkrauti PC, kad būtų išvalyti tempiniai failai.

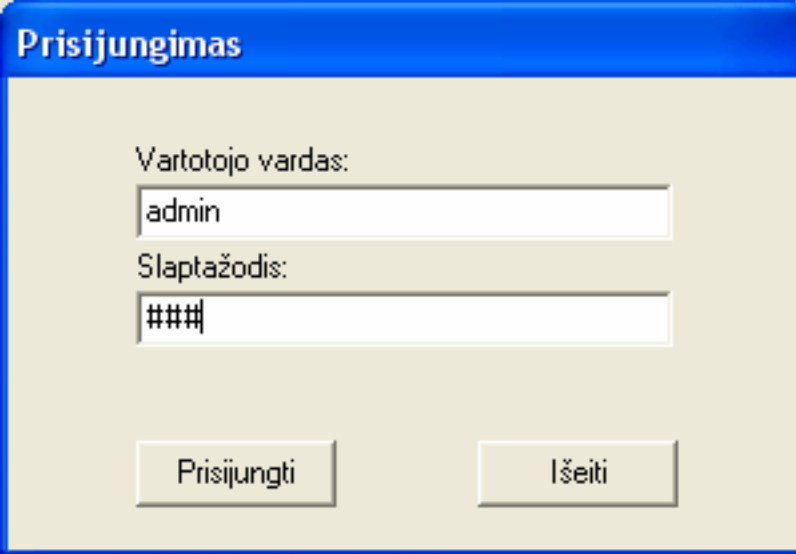
## **5.2 Sistemos naudojimo instrukcija**

Pagrindinis sistemos eksploatavimas yra veiksmai su tvarkaraščiu todėl jiems bus skiriama daugiausiai dėmesio. Pasirinkus iš TSS meniu punktą "Tss".



*TSS meniu*

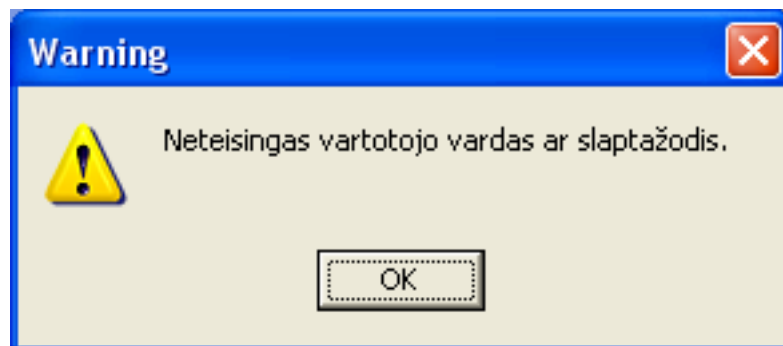
Sistema startuoja. Pirmiausiai atvaizduojamas prisijungimo prie sistemos langas:



A login window titled "Prisijungimas" (Login) with a blue header. It contains two text input fields: "Vartotojo vardas:" (Username) with the text "admin" and "Slaptažodis:" (Password) with masked characters "####". Below the fields are two buttons: "Prisijungti" (Login) and "Išeiti" (Exit).

*Prisijungimo prie sistemos langas*

Suvedus slaptažodį ir vartotojo vardą atliekamas prisijungimas. Jei prisijungimas nepavyko matomas langas:



*Prisijungimas nepavyko*

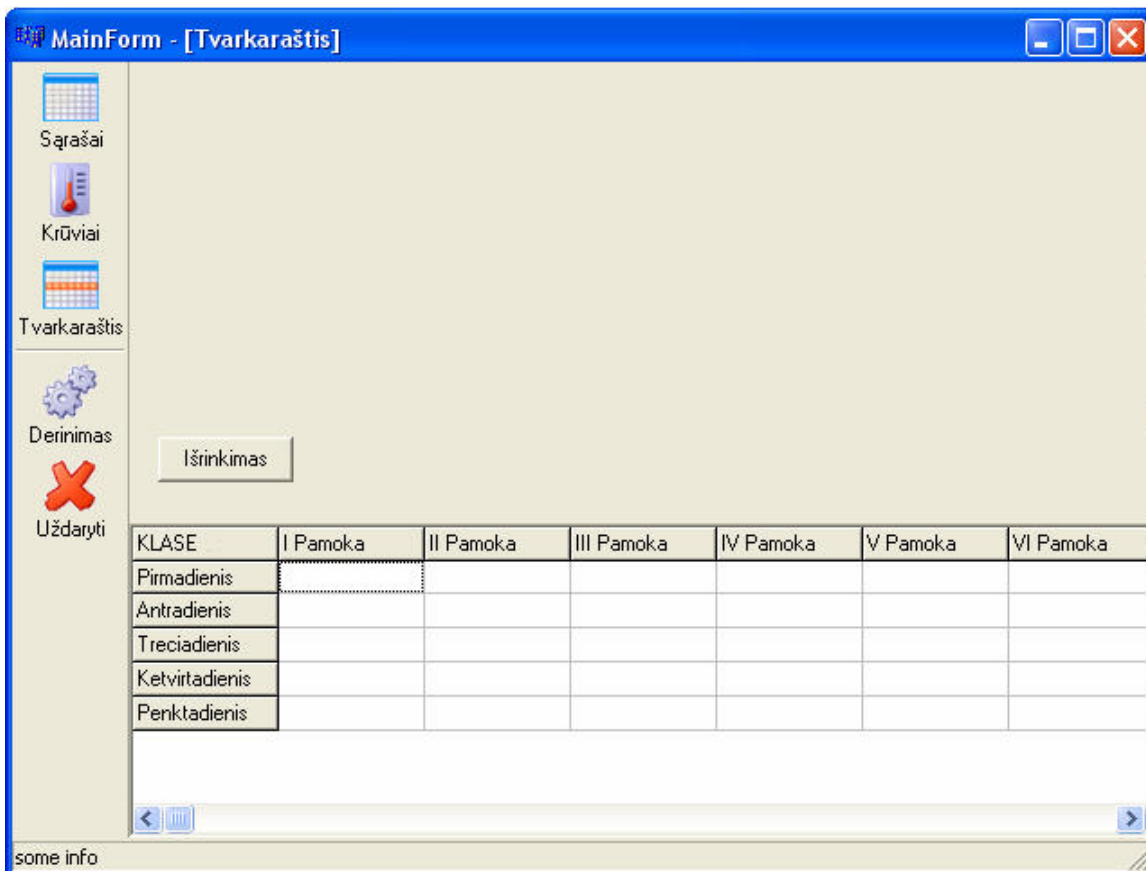
Jei prisijungimas pavyko, startuoja pagrindinė TSS, iš kurios vartotojas nukreipiamas į atitinkamus veiksmus su tvarkaraščiu ar su sistema.



*TSS pagrindinė nuorodų forma*

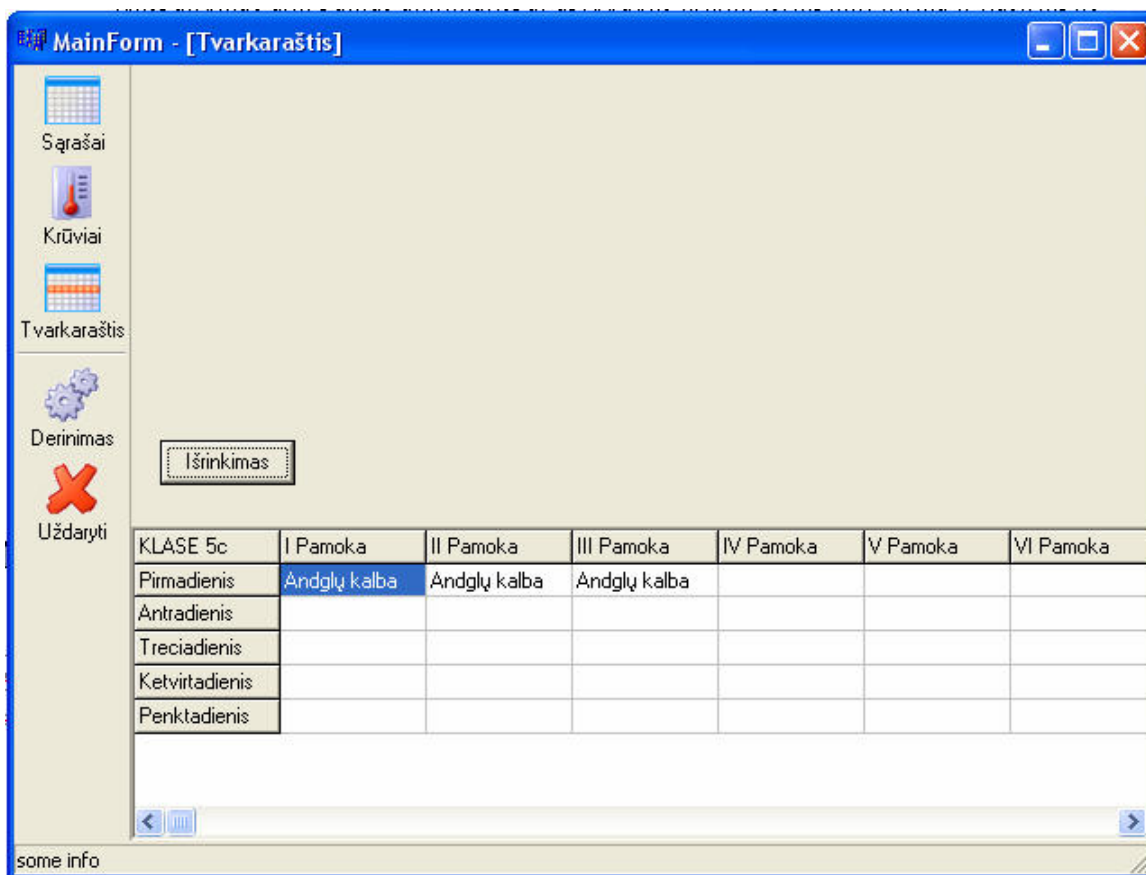
## Tvarkaraštis

Tai tvarkaraščio sudarymo forma: Pirmiausiai, kad patekti į ją reikia pasirinkti Tvarkaraštis .



*Pagrindinė klasių pasirinkimo forma*

Kai klasių sąrašas jau sudarytas, tada spaudžiamas mygtukas [Išrinkimas] ir vartotojui atvaizduojama forma su išrinktų tvarkaraščių:



Pamokų lentelė užima visą ekraną.

Realizuojami pagrindiniai vartotojo veiksmai su tvarkaraščiu:

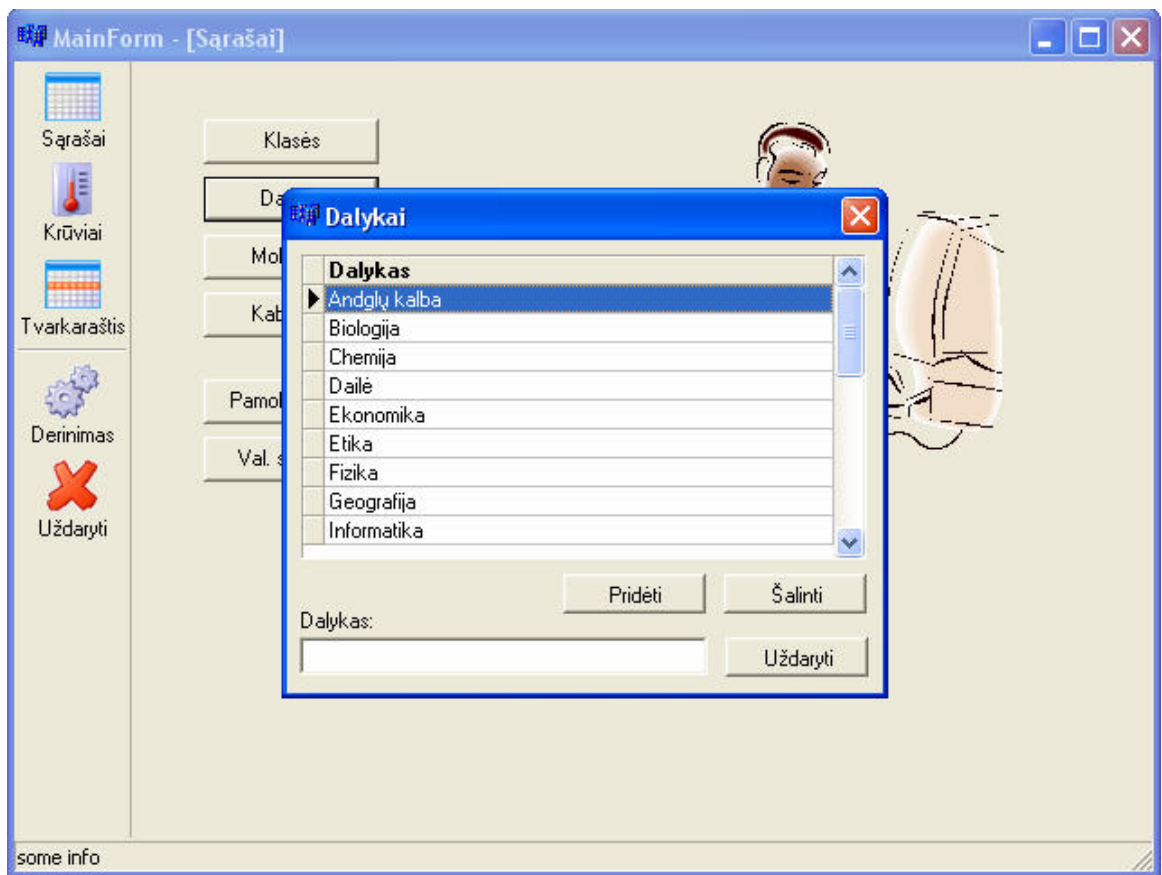
- ✓ Nuskaityti pasirinktos klasės tvarkaraštį

Nuskaitymas atliekamas automatiškai aktyvavus klasės išrinkimo formą. Vartotojui ekrane parodomas sąrašas, kuri jis gali modifikuoti.

- ✓ Pasirinkti dalyką:

Nuskaitymas atliekamas automatiškai aktyvavus modulio išrinkimo formą (pasirinkus meniu punktą "Sąrašai-Dalykai")



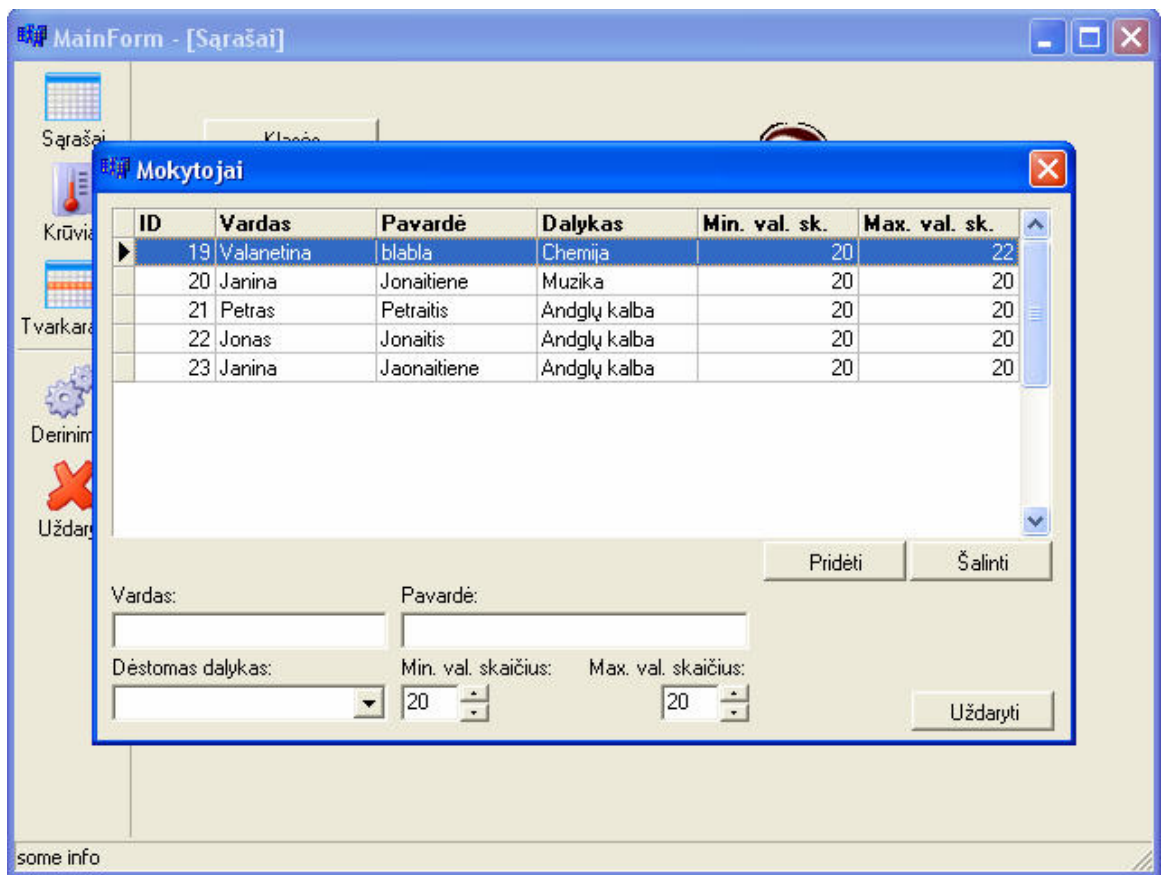


*Dalykų išrinkimo  
forma*

Pasirinkus kitą parametrų keitimo formą, šita uždaroma.

- ✓ Pasirinkti kabinėtą
- ✓ Pasirinkti mokytoją

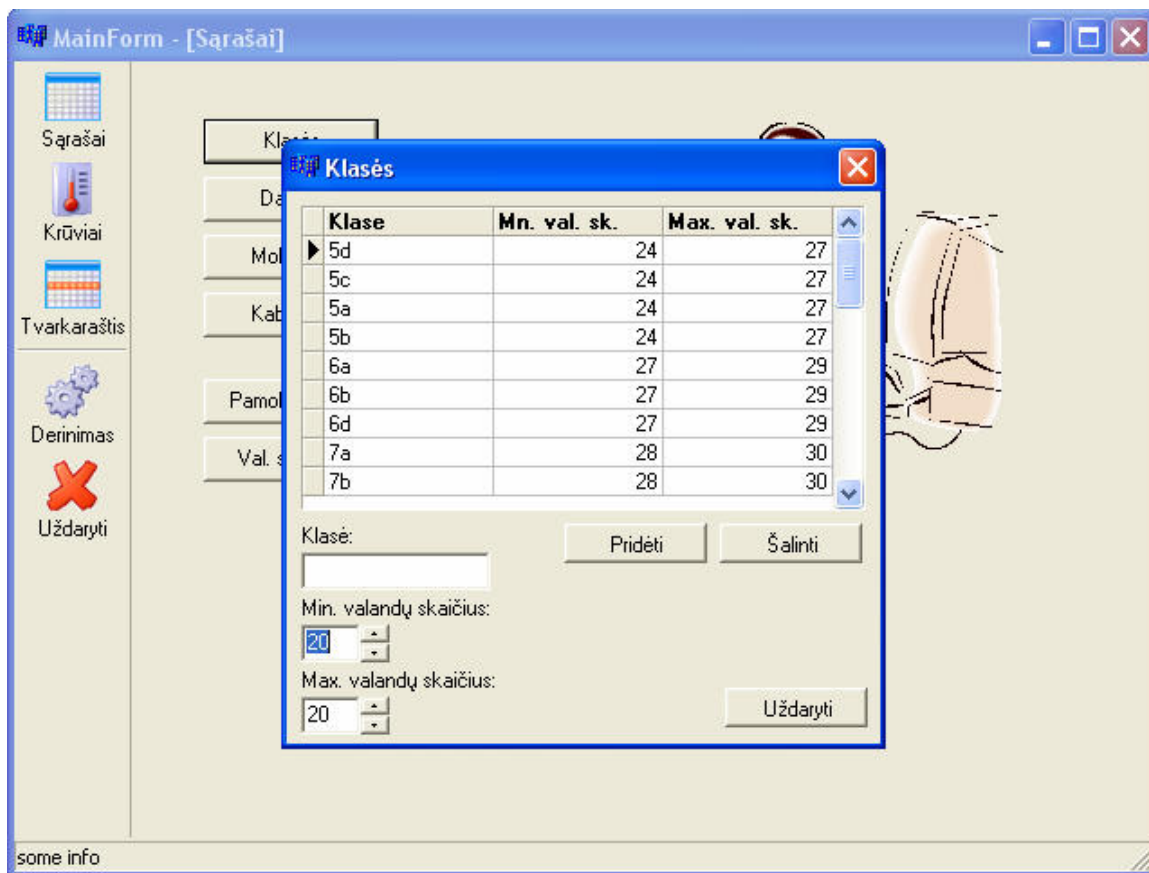
Nuskaitymas atliekamas automatiškai aktyvavus mokytojo išrinkimo formą (pasirinkus meniu punktą "Sąrašai - Mokytojai")



*punktas  
"Mokytojai"*

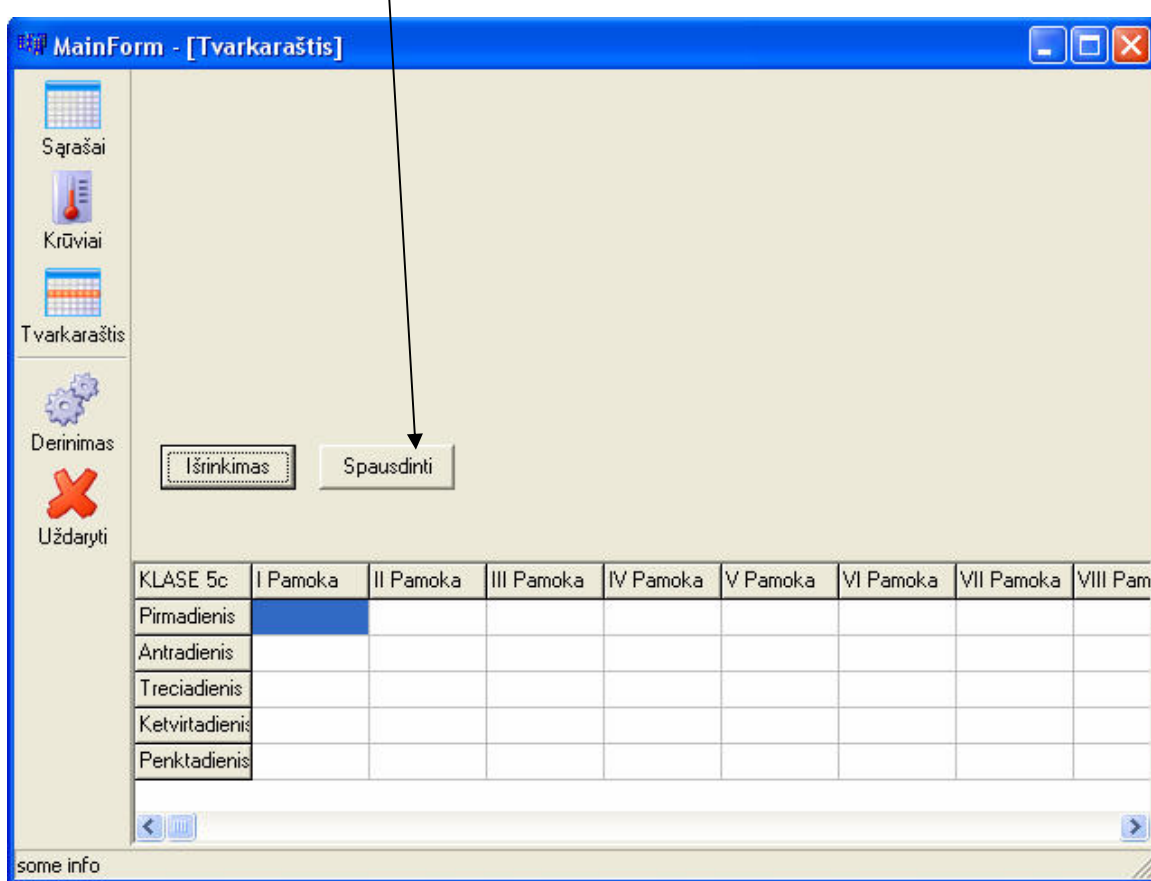
Duomenis galima ieškoti betarpiškai suvedinėjant mokytojo vardą. Tuomet mokytojų sąrašo lange rodomas arčiausiai tiesos esantis dalykas. Pasirinkus kitą paskaitos parametrų keitimo formą, šita uždaroma.

- ✓ Pasirinkti klasę  
Nuskaitymas atliekamas automatiškai aktyvavus klasės išrinkimo formą (pasirinkus meniu punktą "Sąrašai - Klasės")



*punktas "Klasės"*

- Pašalinti pamoką  
Pasirenkamas punktas "Šalinti":  
Lange "Dalykai" nurodomas su pele pašalinamos dalyko laukelis ir vieną kartą spustelimas pelės dešinys klavišas. Sėkmingai pašalinus paskaitą toje vietoje laukelis lieka tuščias.
- ✓ Atspausdinti naujai sudarytą tvarkaraštį  
Pasirenkamas punktas "Spausdinti":



*Meniu punktas  
“Spausdinti  
tvarkaraštį”*

Atspausdinimo punktas dabar rekonstruojamas.

## **Apribojimai**

Apribojimų punktas dabar rekonstruojamas.

## **Konfigūracija**

Konfigūracijos punktas dabar rekonstruojamas.

## **5.2 Sistemos testavimo aprašymas**

Programinės įrangos tvarkaraščiui sudaryti (UTSS) testavimo plano tikslai:

- Apibrėžti testavimo tikslus ir siekius.
- Aprašyti naudojamus testavimo technikas.
- Aprašyti naudojamus testavimo strategijas.
- Kiekvienai testavimo strategijai pateikti detalius testavimo scenarijus.
- Nustatyti testavimui reikiamus resursus.
- Apibrėžti testavimo aplinkybes.

UTSS programinės įrangos kokybės bei patikimumo užtikrinimui šiame dokumente yra aprašytos sistemos testavimo strategijos (Duomenų ir duomenų bazės integralumo, Funkcionalumo, Verslo ciklo, Vartotojo sąsajos, Našumo, Apkrovimo, Stresiniai, Saugumo ir priėjimo valdymo, Sutrikimo/atsistatymo, Konfigūracijos, Instaliavimo), testavimo scenarijai bei reikalingos aplinkybės ir technika.

UTSS testavimo planą sudaro 3 skyriai.

Pirmasis skyrius, apžvelgia skyrius bei pateikia nuorodas į dokumentus, kuriais remiantis buvo sudarytas UTSS testavimo planas bei aprašyti scenarijai.

Antrasis skyrius aprašo testavimo strategijas, kuriomis remiantis buvo sudaryti testavimo scenarijai UTSS moduliams testuoti.

Trečias skyrius aprašo testavimo scenarijus UTSS moduliui testuoti.

## Duomenų ir duomenų bazės integralumo testai

Duomenų bazės ir duomenų bazės procesai turi būti testuojami kaip atskiros sistemos. Šios sistemos turi būti testuojamos be taikomųjų programų (kaip sąsajos su duomenimis). Įrankių/metodų identifikavimui reikia atlikti papildomą TSS DB tyrimą. To gali prireikti toliau aprašytam testavimui.

Testavimo tikslas:	Užtikrinti, jog tinkamai ir be duomenų iškraipymo funkcionuoja duomenų bazės pasiekimo metodai bei procesai.
Metodai:	<ul style="list-style-type: none"><li>Iššaukti kiekvieną duomenų bazės pasiekimo metodą bei procesą su teisingais ir klaidingais duomenimis.</li><li>Ištirti duomenų bazę, jog įsitikinti, kad duomenys buvo patalpinti taip, kaip ir buvo ketinta, visi duomenų bazės įvykiai startavo teisingai. Arba galima peržiūrėti grąžintus duomenis įsitikinimui, jog išgauti buvo teisingi duomenys (su teisingų duomenų įėjimu).</li></ul>
Užbaigimo kriterijus:	Visi duomenų bazės pasiekimo metodai bei procesai funkcionuoja taip, kaip suprojektuota ir be jokių duomenų iškraipymo.
Papildoma informacija:	<ul style="list-style-type: none"><li>Jeiti ar modifikuoti duomenis tiesiogiai duomenų bazėje, testavimui gali prireikti TSS DB kūrimo aplinkos ar tvarkyklių.</li><li>Procesai turi būti iššaukti rankiniu būdu.</li><li>Kad lengviau būtų susekti nepriimtinius įvykius, reikia naudoti nedideles (su ribotu įrašų kiekiu) duomenų bazes.</li></ul>

*Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.*

## Funkcionalumo ir vartotojo sąsajos testai

TSS modulis yra testuojamas, naudojant vartotojo sąsajos testavimo bei funkcinio testavimo strategijas.

Vartotojo sąsajos testavimas verifikuoja vartotojo sąveiką su PĮ. Vartotojo sąsajos testavimas reikalingas tam, kad užtikrinti, jog vartotojo sąsaja suteikia vartotojui galimybę pasiekti testuojamo objekto funkcijas ir kad sistema gali valdyti klaidingus GUI įvykius. Taip pat šis testavimas užtikrina, jog objektai vartotojo sąsajos funkcijoje veikia kaip tikimasi ir atitinka bendrus ar sukurtus standartus.

Vartotojo funkcionalumo strategija identifikuoja klaidas, susijusias su teisingu programos funkcijų atlikimu, t.y. ar programa atlieka funkcijas teisingai, pagal reikalavimus. Šio testo tikslas yra verifikuoti tinkamą duomenų priėmimą, apdorojimą ir išvedimą bei tinkamą verslo taisyklių įdiegimą. Šio tipo testavimas yra paremtas juodos dėžės principu. T.y., programos (ir jos vidinių procesų) verifikavimas pasiekiant ją per GUI ir analizuojant išvedimo rezultatus.

### Žymėjimai:

<tekstas> - tekstas, parašytas tarp tokių skliaustų simbolizuoja programoje esančius atitinkamo pavadinimo mygtukus (pvz.: <Uždaryti>);

*Tekstas* - tekstas, parašytas kursyvu, simbolizuoja vieno ar kito objekto konkrečius pavadinimus (pvz.: Forma *Bandyamas* – reiškia, kad tolesnis aprašymas bus susietas konkrečiai su Forma *Bandyamas* objektais).

“*Tekstas*” - tekstas, parašytas kursyvu ir apskliaustas kabutėmis, atvaizduoja dėl vienos ar kitos priežasties ekrane atsiradusius programos pranešimus (pvz.: “*Klaida vaizduojant pranešimus.*”).

### Prisijungti prie sistemos

#	Veiksmas	Laukiamas rezultatas
	Atsiradus prisijungimo langui vartotojo vardą ir įveskite slaptažodį.	Ekrane atvaizduojamas vartotojo vardas ir šifruotas (* - simboliu) slaptažodis.
	Paspauskite mygtuką <Prisijungti>.	Parodomas sėkmingo prisijungimo langas
	Spauskite mygtuką <OK>	Sistema startuoja, nuskaitomi pradiniai duomenys iš DB ir parodoma pagrindinių sistemos veiksmų pasirinkimo forma.

### Pasirinkti dalyką

	Paspauskite įrankių juostoje mygtuką <Sąrašai> po to meniu punktą „Dalykai“	Startuoja dalyko pasirinkimo forma, duomenys atnaujinami iš DB ir parodomas išrinkimo langas
	Su pele išrinkite kokį nors pamokos langelį lentelėje Dalykai.	Nauja dalyko reikšmė (pavadinimas) bus parodomas pagalbinio pranešimo lange.

### Pasirinkti Mokytoją

	Paspauskite <Sąrašai> mygtuką, toliau meniu punktą „Mokytojai“.	Startuoja mokytojų pasirinkimo forma, duomenys atnaujinami iš DB ir parodomas išrinkimo langas
	Įveskite pradinę mokytojo vardą, pavardę, dėstomą dalyką, reikšmę filtro <i>Vardas, Pavardė</i> įvedimo lange.	Peršokama į įrašą, artimiausia pagal reikšmę
	Formos lange spauskite mygtuką <pridėti> .	Įsimenamas pasirinktas naujas mokytojas.
	Išrinkite lentelėje “Dėstomas dalykas” meniu punktą “Pamoka”	Įjungiamas pagalbinio pranešimo langas .
	Su pele išrinkite kokį nors pamokos langelį lentelėje “Dėstomas dalykas”.	Nauja mokytojo reikšmė ( <i>vardas + pavardė</i> ) parodoma pagalbinio pranešimo lange.

## Pasirinkti kabinetą

	Paspauskite mygtuką <Sąrašai> , toliau pasirinkite meniu punktą „Kabinetai“	Startuoja modulių pasirinkimo forma, duomenys atnaujinami iš DB ir parodomas išrinkimo langas
	Įveskite pradinę kabineto reikšmę filtro <i>Kabineto numeris</i> įvedimo lange	Peršokama į įrašą, artimiausia pagal reikšmę
	Formos lange spauskite Pridėti .	Įsimenamas pasirinktas naujas kabinetas.
	Išrinkite lentelėje “Kabinetai” meniu punktą “kabineto pavadinimas”	Įjungiamas pagalbinio pranešimo langas .
	Su pele išrinkite kokį nors kabineto langelį formoje “Kabinetai”	Nauja kabineto reikšmė parodoma pagalbinio pranešimo lange.

## Įterpti dalyką

	Atlikite 3.2.4 – 6 veiksmus, kad pasirinktumėte teisingus duomenis.	Paruošiami teisingi duomenys
	Išrinkite lentelėje “Dalykai” meniu punktą “Pridėti”	Įjungiamas pagalbinio pranešimo langas o taip pat atliekamas paruošimas įterpimui
	Lentelės “Dalykai” lange pasirinkite laukelyje į kurį norėtumėte įterpti paskaitą ir kliktelkite toje vietoje su pele kariu klavišu.	Įterpiama nauja pamoka, o jei toje vietoje jau buvo kita pakeičiami senos parametrai. Pamoka įterpiama pagal apribojimus ✓ Mokytojas tuo metu neužimtas ✓ Kabinetas tuo metu neužimtas.

## Pašalinti pamoką

	Atlikite 3.2.4 – 6 veiksmus, kad pasirinktumėte teisingus duomenis.	Paruošiami teisingi duomenys
	Išrinkite formoje “Dalykai” meniu punktą “Šalinti”	Išjungiamas pagalbinio pranešimo langas o taip pat atliekamas paruošimas šalinimui.
	Formos “Dalykai” lange pasirinkite pamokos laukelį, tą kurio norėtumėte pašalinti ir kliktelkite toje vietoje su pele kariu klavišu	Pašalinama paskaita, o jei toje vietoje buvo tuščias laukelis – nedaroma nieko.

## Atspausdinti tvarkaraštį

	Išrinkite mygtuką <Tvarkaraštis> meniu punktą “Spausdinti tvarkaraštį”	Duomenys paruošiami spausdinimui
	Failo vietos išrinkimo lange nurodykite failo vardą, į kurį spausdinsite ir spauskite OK .	Įvesti spausdinimo parametrai ir startuojamas spausdinimas. Sėkmingai atspausdinus gaunamas rtf failas.

## Verslo ciklo testai

Verslo ciklo testai turi atkartoti projekto veiklą ilgą laiką. Reikia identifikuoti periodą, pvz., vieni metai ir vykdyti tas transakcijas bei veiklas, kurios bus realiai vykdomos to periodo laikotarpyje. Tai apima kasdieninius, kassavaitinius, kasmėnesinius ciklus ir įvykius, kurie yra priklausomi nuo datos, pvz., užrašai, padedantys prisiminti.

Testavimo tikslas:	Užtikrinti, jog testuojamas objektas bei foniniai procesai tinkamai funkcionuoja pagal reikalaujamus verslo modelius ir tvarkaraštį.
Metodai:	Testavimas imituos keletą verslo ciklų ir bus atliekamas taip: <ul style="list-style-type: none"><li>• Testai, naudoti testuojamo objekto funkcionalumo testavimui bus modifikuoti/patobulinti (padidintas kiekvienos funkcijos vykdymo kartų skaičių) tam, kad imituotų keletą skirtingų vartotojų per apibrėžtą laikotarpį.</li><li>• Visos funkcijos, priklausančios nuo laiko ir datos naudojant teisingus ir klaidingus datos ar laiko periodus.</li><li>• Visos funkcijos, kurios įvyksta pagal tvarkaraštį bus atliktos/paleistos atitinkamu laiku.</li><li>• Testavimas apims teigiamus ir neigiamus duomenis tam, kad verifikuotų, jog: gaunami laukiami rezultatai, kai naudojami teisingi duomenys. išvedami atitinkami klaidų/įspėjimų pranešimai, kai naudojami klaidingi duomenys. tinkamai taikoma kiekviena verslo taisyklė.</li></ul>
Užbaigimo kriterijus:	<ul style="list-style-type: none"><li>• Įvykdyti visi planuoti testai.</li><li>• Visos rastos klaidos pateiktos, ištaisytos ir verifikuotos.</li></ul>
Papildoma informacija:	<ul style="list-style-type: none"><li>• Sistemos datos ir įvykiai gali reikalauti specialių palaikymo veiksmų.</li></ul>

Verslo ciklo testavimui reikia naudoti funkcionalumo testavimo atvejus. Verslo ciklo testavimas gali būti atliekamas tik tada, kai jau atliktas funkcionalumo testavimas ir klaidų jame likę ne daugiau 15%.

*Pastaba: TSM šiuo metu nenumatoma ši testavimo strategija.*

## Našumo testai

Našumo testais matuojami ir įvertinami atsakymo laikai, transakcijų intensyvumas ir kiti reikalavimai, priklausantys nuo laiko. Našumo testų tikslas verifikuoti pasiektus našumo reikalavimus sistemai. Šie testai kuriami ir vykdomi nustatyti testuojamo objekto našumo savybes kaip sąlygų (pvz., darbo apkrova ar aparatūrinė konfigūracija) funkcijas.

**Pastaba:** nerekomenduojama našumo testavimo atlikti rankiniu būdu. Tokiu būdu atliktas testavimas dažniausiai įvertinamas netiksliai.



Testavimo tikslas:	Verifikuoti našumą nustatytoms transakcijoms ar verslo funkcijoms tokiomis sąlygomis: <ul style="list-style-type: none"> <li>• Numatomas normalus darbo apkrovimas</li> <li>• Numatomas didžiausias darbo apkrovimas</li> </ul>
Metodai:	<ul style="list-style-type: none"> <li>• Naudoti testavimo procedūras, sukurtas funkcionalumo bei verslo ciklo testavimui.</li> <li>• Modifikuoti duomenų failus (padidinti transakcijų kiekį) ar skriptus, kad padidėtų iteracijų, iššaukiančių transakcijas, skaičius.</li> <li>• Skriptai turi būti leidžiami tame pačiame kompiuteryje (idealiausia testuoti vieną vartotoją, vieną transakciją) ir turi būti pakartojami su daug klientų (realių ar virtualių, žr. žemiau pateiktą papildomą informaciją).</li> </ul>
Užbaigimo kriterijus:	<ul style="list-style-type: none"> <li>• Viena transakcija / vienas vartotojas: sėkmingas be klaidų skriptų įvykdymas per laukimą/reikalaujamą laikotarpį (transakcijai).</li> <li>• Daug transakcijų / daug klientų: sėkmingas be klaidų skriptų įvykdymas per priimtina laikotarpį.</li> </ul>
Papildoma informacija:	<p>Norint išsamiai ištestuoti našumą, reikia turėti foninį serverio apkrovimą. Yra keli būdai, kaip tai atlikti:</p> <ul style="list-style-type: none"> <li>• Transakcijas pateikti tiesiai serveriui, dažniausiai per SQL užklausas.</li> <li>• Sukurti virtualius vartotojus, kad imituoti daug klientų (dažniausiai kelis šimtus). Tokio apkrovimo sukūrimui galima naudoti Remote Terminal Emulation įrankius. Toks pats metodas gali būti naudojamas tinklo trafiko apkrovimui.</li> <li>• Naudoti daug klientų, kurių kiekvienas paleidęs testavimo skriptus, kad apkrautų sistemą.</li> <li>• Našumo testavimas turi būti atliekamas ant testavimui skirto kompiuterio ar tam tikru laiku. Tik tada galima pilnai valdyti ir tiksliai atlikti matavimus.</li> <li>• Duomenų bazės, naudojamos našumo testavimui turi būti tikro dydžio.</li> </ul>

*Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.*

## Apkrovimo testai

Apkrovimo testavimas yra našumo testavimas, kuris pateikia testuojamam objektui įvairaus dydžio apkrovas, kad išmatuotų ir įvertintų sistemos našumą ir testuojamo objekto galimybes tinkamai funkcionuoti esant skirtingoms darbo apkrovoms. Apkrovimo testavimo tikslas nustatyti ir užtikrinti, jog sistema tinkamai funkcionuoja pateikus jai kiek daugiau nei tikimą maksimalų apkrovimą. Be to šie testai įvertina našumo charakteristikas (atsakymo laikus, transakcijų intensyvumą bei kitus nuo laiko priklausančius parametrus).

**Pastaba:** apkrovimo testavimas dažniausiai atliekamas serveryje, kuriam reikės palaikyti daug klientų užklausų vienu metu, pvz., tinklo serveris. Šio testo nerekomenduojama atlikti rankiniu būdu.

Testavimo tikslas:	Verifikuoti nurodytoms transakcijoms ar verslo atvejams našumo laiką skirtingomis apkrovimo sąlygomis.
Metodai:	<ul style="list-style-type: none"> <li>• Reikia naudoti funkcionalumo ar verslo ciklo testavimui sukurtus testus.</li> <li>• Modifikuoti duomenų failus (padidinti transakcijų kiekį) ar testus, kad padidėtų iteracijų, iššaukiančių transakcijas, skaičius.</li> </ul>
Užbaigimo kriterijus:	Daug transakcijų / daug klientų: sėkmingas be klaidų skriptų įvykdymas per priimtina laikotarpį.
Papildoma informacija:	<ul style="list-style-type: none"> <li>• Apkrovimo testavimas turi būti atliekamas ant testavimui skirto kompiuterio ar tam tikru laiku. Tik tada galima pilnai valdyti ir tiksliai atlikti matavimus.</li> <li>• Duomenų bazės, naudojamos našumo testavimui turi būti tikro dydžio.</li> </ul>

## Stresiniai testai

Stresinis testavimas yra našumo testavimas, sukurtas ir vykdomas, kad rasti klaidas dėl nepakankamų resursų ar resursų varžybų. Gali išaiškėti nematomos normaliomis sąlygomis testuojamo objekto klaidos (esant per mažai atminties ar disko vietos). Kitos klaidos gali kilti besivaržant dėl bendrai naudojamų resursų (pvz., duomenų bazės blokavimo ar tinklo dažnių juostos pločio). Taip pat stresinis testavimas gali būti naudojamas didžiausio apkrovimo, kokią testuojamas objektas dar gali valdyti, identifikavimui.

Testavimo tikslas:	Verifikuoti, jog testuojamas objektas funkcionuoja tinkamai ir be klaidų šiomis stresinėmis sąlygomis: <ul style="list-style-type: none"><li>• Serveryje labai mažai ar iš viso nėra atminties (RAM ir disko).</li><li>• Prisijungę (ar imituota) maksimalus (faktinis ar fiziškai įmanomas) skaičius klientų.</li><li>• Daug vartotojų atlieka tą pačią transakciją tiems patiems duomenims.</li><li>• Maksimaliai apkraunamos transakcijos (žr. aukščiau esantį našumo testavimą)</li></ul> <p><b>Pastaba:</b> Kaip pradinis stresinio testavimo tikslas gali būti ir sąlygų, kurioms esant sutrinka sistemos darbas, identifikacija bei dokumentavimas.</p>
Metodai:	<ul style="list-style-type: none"><li>• Naudoti testus, sukurtus našumo testavimui bei apkrovimo testavimui.</li><li>• Ribotų resursų testavimui, testus reikia leisti tame pačiame kompiuteryje. Serveryje turi būti sumažinta (apribota) RAM bei disko atmintis.</li><li>• Likusiems stresiniams testams reikia panaudoti daug klientų. Leisti tuos pačius ar papildomus testus, kurie maksimaliai apkrautų transakcijas.</li></ul>
Užbaigimo kriterijus:	Įvykdyti visi planuoti testai, bei be sistemos sutrikimų pasiektos nustatytos sistemos ribos (ar sąlygos, dėl kurių atsiranda sistemos sutrikimai yra jau virš užsibrėžtų sąlygų).
Papildoma informacija:	<ul style="list-style-type: none"><li>• Norint atlikti stresinį testavimą tinklui, apkrauti jį pranešimais/paketais, gali prireikti tinklo įrankių.</li><li>• Sistemos duomenų bazės augimo apribojimui reikia laikinai sumažinti turimą laisvą disko vietą.</li><li>• Reikia sinchronizuoti tuo pačiu metu egzistuojančius klientus, kurie bando pasiekti tuos pačius įrašus.</li></ul>

Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.

## Saugumo ir priėjimo valdymo testai

Saugumo ir priėjimo valdymo testai skirti dviems pagrindinėms saugumo sritims:

Taikomosios programos lygio saugumui, apimanti priėjimą prie duomenų bei verslo funkcijų ir Sistemos lygio saugumui, apimanti prisijungimą prie sistemos / nutolusios sistemos.

Taikomosios programos lygio saugumas užtikrina, jog veikėjų veiksmai su tam tikromis funkcijomis / panaudojimo atvejais yra apriboti arba yra apribotas jiems pasiekiamų duomenų kiekis pagal užduotą norimą saugumą. Pvz., kiekvienas veikėjas gali įvesti duomenis ir sukurti naujas sąskaitas, bet tik administratorius gali jas pašalinti. Jei naudojamas saugumas yra duomenų lygio, tai testavimas užtikrina, jog vienas vartotojo tipas gali matyti visą pirkėjo informaciją, apimant ir finansinius duomenis, o kitas vartotojo tipas gali matyti tik demografinius to pačio pirkėjo duomenis.

Sistemos lygio saugumas užtikrina, jog tik tie veikėjai gali pasiekti taikomąją programą, kuriems yra suteiktos teisės prisijungti prie sistemos ir tik per atitinkamą tinklų sąsają.

**Pastaba:** jei yra keli vartotojų tipai, būtinai reikia naudoti saugumo ir priėjimo valdymo testavimą.

Testavimo tikslas:	Taikomosios programos lygio saugumas: verifikuoti, jog veikėjas gali pasiekti tik tas funkcijas/duomenis, kuriems jo vartotojo tipui yra suteiktos teisės. Sistemos lygio saugumas: verifikuoti, jog tik tie veikėjai, kuriems yra suteiktos teisės prisijungti prie sistemos ir taikomųjų programų, gali prie jų prisijungti.
Metodai:	<ul style="list-style-type: none"> <li>• Taikomios programos lygis: turi identifikuoti ir išvardinti visus veikėjų tipus ir funkcijas/duomenis, kuriems šie tipai turi teises.</li> <li>• Sukurti testus kiekvieno tipo veikėjams bei verifikuoti kiekvieną teisę, kuriant transakcijas, specifines kiekvienam veikėjų tipui.</li> <li>• Pakeisti vartotojo tipą ir iš naujo leisti tuos pačius testus. Kiekvienu atveju reikia verifikuoti, jog pateikiamos atitinkamos funkcijos/duomenys vartotojui arba jis prie jų neprileidžiamas.</li> <li>• Sistemos lygio priėjimas (žr. žemiau pateiktą informaciją)</li> </ul>
Užbaigimo kriterijus:	Kiekvienas žinomas veikėjo tipas gali pasiekti atitinkamas funkcijas/duomenis, visos transakcijos funkcionuoja kaip tikėtasi ir buvo paleistos prieš funkcionalumo testavimą.
Papildoma informacija:	Priėjimas prie sistemos turi būti peržiūrėtas/apartas su atitinkamo tinklo ar sistemos administratoriumi. Šio testo galima ir neatlikti, jei jis yra kaip tinklo ar sistemos administravimo funkcija.

*Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.*

## Sutrikimo/atsistatymo testai

Sutrikimo/atsistatymo testai užtikrina, jog testuojamas objektas gali sėkmingai atsistatyti po įvairių aparatūrinės, programinės įrangos ar tinklo sutrikimų, neprarandant duomenų ar duomenų integralumo.

Sutrikimų testavimas užtikrina, jog kai tik atsiranda sutrikimas, toms sistemos, kurios turi pastoviai veikti, alternatyvi ar dubliavimo sistema perima sutrikusios sistemos darbą be duomenų bei transakcijų praradimo.

Atsistatymo testavimas yra atvirkščias procesas, kai taikomoji programa ar sistema yra paveikta ekstremalių sąlygų (ar imituotų sąlygų) ir sutriko jos veikimas (pvz., I/O įrenginio sutrikimas ar klaidingos duomenų bazės rodyklės/raktai). Turi būti iššaukiami atsistatymo procesai, stebima taikomoji programa/sistema. Verifikuojama, ar atitinkamai atstatyti duomenys, programa ar sistema.

Testavimo tikslas:	Verifikuoti, jog atstatymo procesas (rankinis ar automatizuotas) tinkamai atstato duomenų bazę, taikomąsias programas ir sistemą į norimą žinomą būseną. Testavime reikia įtraukti šias sąlygas: <ul style="list-style-type: none"> <li>• Įtampos nutraukimas pas klientą</li> <li>• Įtampos nutraukimas serveryje</li> <li>• Komunikacijos nutrūkimas per tinklo serverį (-ius).</li> <li>• Įtampos ar komunikacijos praradimas su disku.</li> <li>• Nepilni ciklai (duomenų filtravimo, sinchronizacijos ir kt. procesai apdoroja išgadintus duomenis)</li> <li>• Klaidingos duomenų bazės rodyklės/raktai</li> <li>• Klaidingi/išgadinti duomenų bazės elementai</li> </ul>
--------------------	--

Metodai:	<p>Transakcijoms kurti turi būti naudojami sukurti funkcionalumo ir verslo ciklo testai. Kai tik pasiektas norimas testavimo taškas, reikia atskirai atlikti (ar imituoti) šiuos veiksmus:</p> <ul style="list-style-type: none"> <li>• Įtampos nutraukimas pas klientą: Išjungti PC</li> <li>• Įtampos nutraukimas serveryje: išjungti ar imituoti serverio išjungimo procedūrą</li> <li>• Komunikacijos nutrūkimas per tinklo serverį: imituoti ar nutraukti komunikaciją tinklu (fiziškai atjungti tinklo laidus ar išjungti tinklo serverį(-ius)/maršrutizatorius).</li> <li>• Įtampos ar komunikacijos praradimas su disku: imituoti ar fiziškai eliminuoti komunikaciją su vienu ar daugiau diskų.</li> </ul> <p>Kai tik būna sukeltos šios sąlygos / imituotos sąlygos, reikia įvykdyti papildomas transakcijas ir turi būti iššauktos atstatymo procedūros (patenkama į antrą testo būseną).</p> <p>Nepilnų ciklą testavimas naudoja tuos pačius aukščiau aprašytus metodus bei duomenų bazių proceso nutraukimą.</p> <p>Testuojant su šiomis sąlygomis reikia pasiekti žinomą duomenų bazės būseną. Kai kuriuos duomenų bazės laukus, rodykles ir raktus reikia išgadinti rankiniu būdu bei naudojant duomenų bazės įrankius. Papildomai reikia įvykdyti transakcijas naudojant funkcionalumo bei verslo ciklo testus.</p>
Užbaigimo kriterijus:	<p>Visais aukščiau aprašytais atvejais taikomoji programa, duomenų bazė ir sistema turi įvykdyti atstatymo procedūras ir grįžti į žinomą, norimą būseną. Būsena gali būti žinomi išgadinti laukai, rodyklės/raktai ir pranešimai, rodantys transakcijas ar procesus, kurie nebuvo užbaigti dėl sutrikimų.</p>
Papildoma informacija:	<p>Atsistatymo procesas yra labai nepatikimas. Laidų atjungimas (įtampos nutraukimo imitavimas ar komunikavimo praradimas) gali būti negalimas ar nepageidaujamas. Gali prireikti alternatyvių metodų, tokių kaip diagnostiniai PĮ įrankiai.</p> <p>Reikalingi sistemos resursai (ar kompiuterinės operacijos), duomenų bazė bei tinklas.</p> <p>Šie testai turi būti atliekami po darbo valandų ar ant atskiro kompiuterio (-ių).</p>

*Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.*

## Konfigūracijos testas

Konfigūracijos testai verifikuoja testavimo objektų veikimą skirtingose PĮ ir aparatūrinės įrangos konfigūracijose. Klientų aparatūrinė įranga, tinklo konfigūracija ir duomenų bazių serveriai dažniausiai yra skirtingi. Jų darbo vietos gali turėti užkrautas skirtingas PĮ (pvz., taikomasias sistemas, tvarkykles ir t.t.) ir kiekvieną kartą gali būti aktyvios skirtingos šių parametrų kombinacijos bei naudojami skirtingi resursai.

Testavimo tikslas:	Verifikuoti, jog testuojamas objektas tinkamai funkcionuoja su reikalaujama aparatūrinės/programinės įrangos konfigūracija.
Metodai:	<ul style="list-style-type: none"> <li>• Naudoti funkcionalumo testų skriptus.</li> <li>• Atidaryti/uždaryti įvairią netestuojamą, bet susijusią PĮ (pvz., Microsoft taikomasias programas, Excel ir Word) kaip dalį testo ar prieš testo vykdymą.</li> <li>• Įvykdyti pasirinktas transakcijas, kurios imituotų veikėjo sąveiką sus testuojamu objektu bei netestuojama PĮ</li> <li>• Pakartoti visą procesą minimizuojant galimą operatyvinę atmintį.</li> </ul>
Užbaigimo kriterijus:	Kiekvienai testuojamo ir netestuojamo objekto PĮ kombinacijai visos transakcijos atliktos sėkmingai be klaidų.

Papildoma informacija:	<ul style="list-style-type: none"> <li>• Kokia reikalinga, galima, pasiekama netestuojamo objekto PĮ?</li> <li>• Kokios dažniausiai naudojamos taikomosios programos?</li> <li>• Kokiais duomenimis operuoja veikiančios taikomosios programos (t.y. atidarytos didelės Excel elektroninės lentelės, 100 puslapių Word dokumentas).</li> <li>• Šiame teste taip pat reikia dokumentuoti visą sistemą, tinklo serverius, duomenų bazes ir t.t.</li> </ul>
------------------------	--

Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.

## Instaliavimo testas

Instaliavimo testavimas turi tu tikslus. Pirmasis yra garantuoti, kad programinė įranga gali būti suinstaliuota skirtingomis sąlygomis (pvz., nauja instaliacija, atnaujinimas, pilna ar pagal vartotojo pasirinkimą instaliacija, normaliomis ir nenormaliomis sąlygomis). Nenormalios sąlygos tai –nepakankamai vietos diske, neturėjimas teisių sukurti direktorijoms ir kt. Antrasis tikslas yra verifikuoti, kad instaliavus sistemą, PĮ funkcionuoja teisingai. Tam dažniausiai reikia atlikti testus, sukurtus ištestuoti funkcionalumui.

Testavimo tikslas:	Verifikuoti, jog testavimo objektas tinkamai susiinstaliuoja ant kiekvienos reikalavimuose nurodytos aparatūrinės įrangos šiomis sąlygomis: <ul style="list-style-type: none"> <li>• Nauja instaliacija, naujas kompiuteris, ant kurio niekada anksčiau nebuvo instaliuota testuojama PĮ</li> <li>• Atnaujinti kompiuteryje instaliuotą versiją tokia pačia versija</li> <li>• Atnaujinti kompiuteryje instaliuotą versiją naujesne</li> </ul>
Metodai:	<ul style="list-style-type: none"> <li>• Rankiniu būdu arba sukurti automatizuotus skriptus.</li> <li>• Paleisti / atlikti instaliaciją.</li> <li>• Paleisti transakcijas naudojant iš anksto apibrėžtą aibę testavimo skriptų.</li> </ul>
Užbaigimo kriterijus:	Transakcijos atliktos sėkmingai be klaidų.
Papildoma informacija:	Kokios PĮ transakcijos turi įeiti į testą, kad sistema būtų sėkmingai suinstaliuota ir nebūtų prarasti jokie PĮ komponentai?

Pastaba: TSS šiuo metu nenumatoma ši testavimo strategija.

### 5.3 Kokybės kriterijų įvertinimas

Projektavimo kriterijų santrauka pateikta 1 lentelėje.

Prioritetas \ Kriterijus	*****	****	***	**	*
Naudojamumas			+		
Saugumas		+			
Efektyvumas		+			
Teisingumas	+				
Patikimumas		+			
Palaikomumas			+		

Testuojamumas				+	
Lankstumas	+				
Suprantamumas	+				
Pakartotinis panaudojamumas			+		
Pernešamumas					+
Įsiliejamumas	+				

### *Projektavimo kriterijai*

Projektuojant sistemos architektūra turi būti naudojama Objektiškai-Orientuota projektavimo metodologija. Projektuojant turi būti naudojami stebėtojo, kompozitoriaus, dekoratoriaus, lankytojo, komandos bei kiti projektavimo šablonai

## 6. Išvados

Kuriama programa turėtų pagerinti tvarkaraščio sudarymą.

Tvarkaraščio sudaryme vartotojas dalyvauja tik duomenų failo sudaryme, duomenų įvedime ir rezultatų gavime. Programa ne tik turės sudaryti tvarkaraštį, atsižvelgiant į nurodytus apribojimus, bet ir atliks daugelį optimizavimų veiksmų. Tai nereikalauja didelių vartotojo pastangų, mokėjimo dirbti kompiuteriu bei programa, lyginant darbą su programa „Mimoza“. Ši tvarkaraštį galės sudaryti netgi mokyklos sekretorė.

Darbo metu buvo:

- Atlikta vartotojo poreikių analizė.
- Atlikta projektavimo aplinkos analizė.
- Pasirinktas ir išanalizuotas sprendimas.

## 7. Terminai

TSS – Tvarkaraščio Sudarymo Sistema. Tai programinė įranga skirta automatizuoti tvarkaraščio sudarymui.

PĮ – Programinė Įranga.

KTU – Kauno Technologijos Universitetas.

IT – Informacinės Technologijos.

OS – Operacinė Sistema.

Techninė įranga – kompiuterinių įrenginių rinkinys (pvz., procesorius, monitorius), kurie priima duomenis, saugo, apdoroja ir išveda juos.

Kompiuterinis tinklas – sistema, jungianti informacinės sistemos dalis.

IT – Informacinės Technologijos.

DAK – Duomenų Apdorojimo Komponentas.

DSK – Duomenų apSaugos Komponentas

## 8. Literatūros sąrašas

- [1] **Germanavičius, V.** School schedule optimization program. <http://eta.ktl.mii.lt:8080/BMS/>, 2002.
- [2] **Hajdu, M.** Network Scheduling Techniques for Construction Project Management. *Kluwer Academic Publishers*, 1997.
- [3] **Mockus, J.** A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach. *Kluwer Academic Publishers*, 2000.
- [4] **Mockus, J.** Bayesian heuristic approach to global optimization and examples. *Kluwer Academic Publishers*, Journal of global optimization 22 (1-4): 191-203 JAN 2002.
- [5] **Mockus, J.** Set of Examples of Global and Discrete Optimization. <http://www.mockus.us/optimum/>, 2003.

## 9. Priedas

TSS programinis produktas.