****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Programavimo mokymosi sistema, paremta naršyklės technologijomis**

Baigiamasis bakalauro studijų projektas

|  |
| --- |
|  |
| **Rokas Sičiovas**  Projekto autorius |
|  |
| **prof. Tomas Blažauskas**  Vadovas |
|  |

**Kaunas, 2023**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Programavimo mokymosi sistema, paremta naršyklės technologijomis**

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

|  |  |
| --- | --- |
|  |  |
| **Rokas Sičiovas**  Projekto autorius | (parašas) (data) |
|  |  |
| **prof. Tomas Blažauskas**  Vadovas | (parašas) (data) |
|  |  |
| **doc. Sigitas Drąsutis**  Recenzentas | (parašas) (data) |
|  |  |

**Kaunas, 2023**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

Rokas Sičiovas

**Programavimo mokymosi sistema, paremta naršyklės technologijomis**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Rokas Sičiovas

*Patvirtinta elektroniniu būdu*

Rokas Sičiovas. Programavimo mokymosi sistema, paremta naršyklės technologijomis. Bakalauro studijų baigiamasis projektas vadovas prof. Tomas Blažauskas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: ................................(įrašykite).

Kaunas, 2023. XX p.

Santrauka

Lorem ipsum dolor sit amet, eam ex decore persequeris, sit at illud lobortis atomorum. Sed dolorem quaerendum ne, prompta instructior ne pri. Et mel partiendo suscipiantur, docendi abhorreant ea sit. Recteque imperdiet eum te.

Eu eum decore inimicus consetetur, cu usu habeo corpora intellegam. Ut antiopam efficiendi deterruisset sit. Mel sint eirmod id, qui quot virtute id, dolor nemore forensibus usu id. Fugit dolore voluptatum cu vim. An vix veniam graecis insolens, sit posse iusto id. Ut vim ceteros percipit, id quo ubique recusabo, eum sint lucilius ea. In sumo inani numquam has.

Rokas Sičiovas. Programming Learning System, Based on Browser Technologies. Bachelor's  Final Degree Project prof. Tomas Blažauskas; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: ................................ (type here).

Kaunas, 2023. Number of pages.

Summary

Lorem ipsum dolor sit amet, eam ex decore persequeris, sit at illud lobortis atomorum. Sed dolorem quaerendum ne, prompta instructior ne pri. Et mel partiendo suscipiantur, docendi abhorreant ea sit. Recteque imperdiet eum te.

Eu eum decore inimicus consetetur, cu usu habeo corpora intellegam. Ut antiopam efficiendi deterruisset sit. Mel sint eirmod id, qui quot virtute id, dolor nemore forensibus usu id. Fugit dolore voluptatum cu vim. An vix veniam graecis insolens, sit posse iusto id. Ut vim ceteros percipit, id quo ubique recusabo, eum sint lucilius ea. In sumo inani numquam has.

Turinys

[Lentelių sąrašas 7](#_Toc99973258)

[Paveikslų sąrašas 8](#_Toc99973259)

[Santrumpų ir terminų sąrašas 9](#_Toc99973260)

[Įvadas 10](#_Toc99973261)

[1. Analizė 11](#_Toc99973262)

[1.1. Techninis pasiūlymas 11](#_Toc99973263)

[1.1.1. Sistemos apibrėžimas 11](#_Toc99973264)

[1.1.2. Bendras veiklos tikslas 11](#_Toc99973265)

[1.1.3. Sistemos pagrįstumas 11](#_Toc99973266)

[1.1.4. Egzistuojančių sprendimų analizė 12](#_Toc99973267)

[1.1.5. Prototipai ir pagalbinė informacija 13](#_Toc99973268)

[1.2. Techninių galimybių analizė 14](#_Toc99973269)

[2. Projektas 15](#_Toc99973270)

[2.1. Reikalavimų specifikacija 15](#_Toc99973271)

[2.1.1. Komercinė specifikacija 15](#_Toc99973272)

[2.1.2. Sistemos funkcijos 15](#_Toc99973273)

[2.1.3. Naudotojo sąsajos specifikacija 16](#_Toc99973274)

[2.1.4. Realizacijai keliami reikalavimai 16](#_Toc99973275)

[2.1.5. Techninė specifikacija 17](#_Toc99973276)

[2.2. Projektavimo metodai 17](#_Toc99973277)

[2.2.1. Projektavimo valdymas ir eiga 17](#_Toc99973278)

[2.2.2. Projektavimo technologija 17](#_Toc99973279)

[2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos 17](#_Toc99973280)

[2.3. Sistemos projektas 17](#_Toc99973281)

[2.3.1. Statinis sistemos vaizdas 17](#_Toc99973282)

[2.3.2. Dinaminis sistemos vaizdas 18](#_Toc99973283)

[3. Testavimas 19](#_Toc99973284)

[3.1. Testavimo planas 19](#_Toc99973285)

[3.2. Testavimo kriterijai 19](#_Toc99973286)

[3.3. Vienetų testavimas 19](#_Toc99973287)

[3.4. Integracinis testavimas 19](#_Toc99973288)

[3.5. Naudotojo sąsajos testavimas 19](#_Toc99973289)

[4. Dokumentacija naudotojui 20](#_Toc99973290)

[4.1. Apibendrintas sistemos galimybių aprašymas 20](#_Toc99973291)

[4.2. Naudotojo vadovas 20](#_Toc99973292)

[4.3. Diegimo vadovas 20](#_Toc99973293)

[4.4. Administravimo vadovas 20](#_Toc99973294)

[Rezultatai ir išvados 21](#_Toc99973295)

[Literatūros sąrašas 22](#_Toc99973296)

[Priedai 23](#_Toc99973297)

[1 priedas. Priedo pavadinimas 23](#_Toc99973298)

Lentelių sąrašas

[**1 lentelė.** Sprendimų palyginimas 13](#_Toc132826760)

[**2 lentelė.** Pagrindiniai baigiamojo projekto stiliai ir jų aprašymai 23](#_Toc132826761)

Paveikslų sąrašas

**1.1 pav**. „Informiko akademija“ sistemos pavyzdys [1] 12

**1.2 pav**. „KTU Open Courses“ sistemos pavyzdys [2] 12

**1.3 pav.** „GDB Online Debugger“ sistemos pavyzdys [3] 13

**2.1 pav.** „Learn Programming“ panaudojimo atvejų diagrama 16

**2.2 pav.** „Administruoti temas“ veiklos diagrama 17

**2.3 pav.** „Administruoti potemes“ veiklos diagrama 18

**2.4 pav.** „Administruoti uždavinius“ veiklos diagrama 19

**2.5 pav.** „Administruoti mokinius“ veiklos diagrama 20

**2.6 pav**. „Peržiūrėti temas“ veiklos diagrama 21

**2.7 pav**. „Peržiūrėti temos potemes“ veiklos diagrama 21

**2.8 pav.** „Peržiūrėti potemės uždavinius“ veiklos diagrama 22

**2.9 pav.** „Spręsti uždavinį“ veiklos diagrama 23

**2.10 pav.** „Peržiūrėti savo įvertinimus“ veiklos diagrama 24

**2.11 pav.** „Registruotis“ veiklos diagrama 24

**2.12 pav.** „Prisijungti“ veiklos diagrama 25

**2.13 pav.** „Atnaujinti profilio informaciją“ veiklos diagrama 26

**2.14 pav.** „Peržiūrėti visus sistemos naudotojus“ veiklos diagrama 26

**2.15 pav.** „Administruoti el. parduotuvę“ veiklos diagrama 27

**2.16 pav.** „Peržiūrėti užsakymų istoriją“ veiklos diagrama 27

**2.17 pav.** „Peržiūrėti krepšelį“ veiklos diagrama 28

**2.18 pav.** „Įdėti prekę į krepšelį“ veiklos diagrama 28

**2.19 pav.** „Peržiūrėti prekes“ veiklos diagrama 29

**2.20 pav.** „Peržiūrėti prekę“ veiklos diagrama 30

**2.21 pav.** „Atlikti užsakymą“ veiklos diagrama 31

**2.22 pav.** Pagrindinio lango eskizas 32

**2.23 pav.** Kursų lango eskizas 33

**2.24 pav.** Kursų potemių lango eskizas 33

**2.25 pav.** Programavimo užduočių sąrašo lango eskizas 34

**2.26 pav.** Programavimo užduoties lango eskizas 34

**2.27 pav.** Elektroninės parduotuvės prekių sąrašo lango eskizas 35

**2.28 pav.** Elektroninės parduotuvės prekės lango eskizas 35

**2.29 pav.** Sistemos kūrimui naudotas *Agile* metodologija 36

Santrumpų ir terminų sąrašas

**Santrumpos:**

3 kokie

**Terminai:**

7 kokie

Įvadas

Supažindinama su darbo specifika, aktualumu, išdėstomi tikslai bei uždaviniai, aptariama dokumento struktūra. Šiame skyriuje apie darbą kalbama abstrakčiai, nederėtų pateikti nuorodų į kitus šaltinius (1 – 2 lapai).

Darbo problematika ir aktualumas

Apibrėžiama darbo problematika ir aptariamas aktualumas. Šiame poskyryje taip pat nurodoma su darbu susijusi sritis, praktinė darbo reikšmė.

Darbo tikslas ir uždaviniai

Suformuluojamas pagrindinis darbo tikslas, kuris išskaidomas į kelis uždavinius (3 – 7 uždaviniai). Darbo tikslas turėtų nusakyti kodėl yra kuriama sistema (kokią problemą norima išspręsti), o uždaviniai – kaip to tikslo bus pasiekta (dažniausiai tai sutampa su baigiamojo darbo struktūra – reikės išanalizuoti konkurentus, suformuluoti reikalavimus, ištestuoti sistemą ir t.t.). Rašant išvadas rekomenduojama atsižvelgti į darbo pradžioje suformuluotus uždavinius.

1. Pirmasis uždavinys
2. Antrasis uždavinys
3. Trečiasis uždavinys

Darbo struktūra

Aptariama dokumento struktūra. Nurodoma kiek ir kokių skyrių dokumente yra ir kokia informacija juose pateikiama. Labai svarbu, kad šiame skyriuje nerašytumėte bendrinių dalykų, kurie tinkami kone bet kuriam darbui – pvz. „analizės dalyje apžvelgiami egzistuojantys sprendimai, testavimo skyriuje aprašomas testavimas“. Šis apibendrinimas turi atspindėti būtent jūsų darbo struktūrą, vadinasi – „apžvelgiami egzistuojantys buhalterinės apskaitos paketai bei karkasai, skirti darbui su dideliais duomenimis“ ar pan.

Sistemos apimtis

Nors tai nėra tiesiogiai įvadinis sistemos aprašas, būtent čia reikėtų nurodyti jau realizuotos sistemos apimtį. Matai gali būti įvairūs (pasirinkti tinkamus):

* Kodo eilučių skaičius (galima išskaidyti pagal į vidinę (angl. *backend*) ir išorinę (angl. *frontend*) puses, pagal programavimo kalbas ir kt.)
* Testų eilučių skaičius
* Darbo valandos
* Komponentų, klasių, modulių kiekis
* Teikiamų paslaugų skaičius
* Sukurtų 3D modelių kiekis ar kita su projektu siejama metrika

# Analizė

## Techninis pasiūlymas

### Sistemos apibrėžimas

„Learn Programming“ – tai sistema, kuri skirta mokiniams išmokti bei pagerinti programavimo žinias *Python* programavimo kalba, mokytojams stebėti mokinių progresą ir jų įverčius bei elektroninė parduotuvė skirta visiems užsiregistravusiems nariams, kurioje galima nusipirkti knygas, kurios yra skirtos pagilinti programavimo žinias.

### Bendras veiklos tikslas

Šio projekto tikslas yra populiarinti programavimą jaunų mokinių rate, taip suteikiant galimybę mokytojams kurti uždavinius, o mokiniams spręsti uždavinius vienoje sistemoje ir gauti automatinį grįžtamąjį ryšį.

„Learn Programming“ sistemos tikslas yra programavimo populiarinimas Lietuvoje, tai yra stipri nauda visuomenei, kadangi vis daugiau jaunimo nori pradėti programuoti ar tobulėti programavimo srityje ir taip kurti asmeninę ateitį. Populiarinant programavimą vis daugiau jaunimo susidomės šia sritimi ir taip bus plečiama vis stipresnė programuotojų bendruomenė Lietuvoje.

### Sistemos pagrįstumas

Projekto idėja kilo, kuomet savanoriavau Kauno Technologijų Universiteto organizuojamuose programavimo kursuose – „Informiko Akademija“ (konkursai.if.ktu.lt [1]) ir „KTU Open Courses“ (open.ktu.edu [2]).

„Informiko akademija“ svetainėje universiteto studentai kūrė uždavinius, o mokiniai sprendė uždavinius naudodami asmeniniuose kompiuteriuose įdiegta integruota kūrimo aplinka („Code::Block“, „Visual Studio Code“ ir kitas įvairias integruotas kūrimo aplinkas).

Užsiimant savanorystę „KTU Open Courses“ programavimo kursuose teko ne tik kurti uždavinius ir prižiūrėti puslapį, bet ir taisyti mokinių darbus. Mokiniai vėl sprendė uždavinius asmeniniuose kompiuteriuose įdiegtose integruotose kūrimo aplinkose bei išsprendę uždavinį galėjo įkelti kodo failą, kurį savanoriai atsisiuntę peržiūrėdavo ir ištestuodavo, o ištestavę įrašydavo pažymį ir pateikdavo grįžtamąjį ryšį.

Kilus idėjai norėjosi, jog sistema būtų greitesnė, patogesnė ir, žinoma, unikalesnė. Todėl prieš užtikrinant, jog bus kuriama ši sistema, turėjau sugalvoti, jog kuriama sistema stipriai išsiskirtų nuo sistemų, kurias prižiūrėjau ir jau nuo kitų egzistuojančių internete. Pagrindinė mintis buvo, jog svetainėje bus įdiegtas kodo kompiliatorius, kur parašius kodą ir sukompiliavus būtų gaunamas atsakymas ir kodas automatiškai ištestuojamas ir mokiniui pateikiamas įvertis.

### Egzistuojančių sprendimų analizė

Šiuo metu rinkoje jau egzistuoja panašių edukacinių lietuviškų ir užsienietiškų sistemų. Geriausiai šias sistemas kuria ir plečia Kauno Technologijos Universitetas, kuris šiuo metu turi sukūręs keletą tokių sistemų – „Informiko Akademija“ ir „KTU Open Courses“. Viena iš užsienietiškų būtų, kurią taip pat teko naudoti – „GDB Online Debugger“.

Graphical user interface, text, application, email

Description automatically generated

**1.1** **pav**. „Informiko akademija“ sistemos pavyzdys [1]

„Informiko Akademija“ yra lietuviškas puslapis, sukurtas Kauno Technologijos Universiteto, kuris šiuo metu nebelabai yra atnaujinamas bei nebėra dedamos naujos užduotys, tačiau mokiniai gali pasiekti senesnių metų užduotis ir taip praktikuotis atliekant sukurtomis universiteto studentų užduotimis, kurios ir šiuo metu yra naudojamos gimnazijose, kaip kontrolinių darbų ar namų darbų užduotys.

Mokiniai atsidarę užduotis (žr. 1.1 pav.) galėdavo matyti išsamų uždavinio aprašymą bei užduoties testų variantus, norint pasitikrinti ar uždavinys teisingai išspręstas. Taip pat mokiniai turėjo galimybę įvertinti savanorio sukurtą užduotį penkiabalėje sistemoje, paspausdami įvertinimą ant žvaigždučių simbolio (1 – labai blogai, 5 – labai gerai).

Graphical user interface, text, application

Description automatically generated

**1.2** **pav**. „KTU Open Courses“ sistemos pavyzdys [2]

„KTU Open Courses“ taip pat yra lietuviškas puslapis, sukurtas Kauno Technologijos Universiteto, kuris šiuo metu yra labai plačiai naudojamas ir vystomas pačio universiteto. Atliekant savanorystę „KTU Open Courses“ mokinių skaičius *C++* programavimo kalbos kursuose šiuo metu siekia 333 dalyvių. *C++* programavimo kalbos pamokėlės studentui leidžia studentui gauti aukštos kokybės uždavinius ir įkelti pagrindini kodo failą.

Mokinys atsidaręs failą (žr. 1.2 pav.) „Teorinė medžiaga, pavyzdinė programa“ galėdavo susipažinti su naujos temos teorine medžiaga, perskaityti bei pamatyti kelias pavyzdines parašytas programas, taip norint suteikti platesnę informaciją mokiniui. Atsidaręs puslapį „Uždaviniai (Paprastų programų sudarymas)“ studentui atidaromas dokumentas *.pdf* formatu, kuriame matomos visos tos temos užduotys bei taip pat visų užduočių testų variantai. Kiek pavyko mokiniui išspręsti užduočių, tiek jis galėdavo įkelti į puslapį „Atliktų užduočių įkėlimas“, o universiteto studentai savanoriai šiuos darbus atsisiuntę ištaiso ir pateikia grįžtamąjį ryšį.

Graphical user interface, text

Description automatically generated

**1.3** **pav.** „GDB Online Debugger“ sistemos pavyzdys [3]

Trečioji sistema – „GDB Online Debugger“, ši sistema nėra lietuviška, todėl viskas yra anglų kalba. Ši sistema išsiskiria nuo kitų sistemų, kadangi joje yra mokytojo rolė, kurioje mokytojas gali pridėti studentą į savo sąrašą ir kurti uždavinius būtent studentui. Didžiausias pliusas šios sistemos – įdiegtas kodo kompiliatorius sistemoje bei galima pasirinkti skirtingą programavimo kalbą, o tai leidžia mokiniui spręsti uždavinius mėgstama programavimo kalba. Taip pat ši sistema yra pranašesnė, kadangi joje galima pridėti testų variantus uždaviniams ir įdiegtas automatiškas testavimas, o dėstytojas gali įrašyti pažymį peržiūrėjęs kodą arba nustatyti, jog mokiniai įvertį gauna iš automatinio testavimo.

Žemiau 1 lentelėje yra pateikiami sprendimų palyginimai, kuriame yra jau anksčiau paminėtos trys sistemos bei mano kuriama sistema.

**1 lentelė.** Sprendimų palyginimas

| **Lyginimo kriterijai** | Informiko Akademija | KTU Open Courses | GDB Online Debugger | Learn Programming |
| --- | --- | --- | --- | --- |
| Automatinis uždavinio kodo testavimas | - | - | + | + |
| Programavimo kalbos kompiliatoriaus integracija | - | - | + | + |
| Bankinių pavedimų integracija | - | - | + | + |
| Galimybė pasirinkti puslapio kalbą | - | + | - | - |
| Įverčių sistema | - | + | + | + |
| Mokytojo uždavinių kūrimas savo mokiniams | - | - | + | + |
| Susisiekimas su administratoriumi sistemoje | - | + | + | + |
| Elektroninės parduotuvės sistema | - | - | - | + |
| Skirtingos programavimo kalbos pasirinkimas | - | - | + | - |

Galime teigti, jog artimiausias konkurentas kuriamai sistemai yra „GDB Online Debugger“ edukacinė sistema. Tuo tarpu „Informiko Akademija“ ir „KTU Open Courses“ teikiami sprendimai nėra tokie unikalūs ar išskirtiniai. Tačiau, vis dėlto kuriama sistema yra pranašesnė už abejas lietuviškas sistemas, kadangi yra padarytas *Python* programavimo kalbos kompiliatoriaus integracija, bankinių pavedimų integracija bei automatinis uždavinio kodo testavimas. Tačiau atliekant sprendimų palyginimą, pastebėjau, jog tiek „KTU Open Courses“, tiek „GDB Online Debugger“ sistemos turi įverčių sistemą, kurioje kiekvienas mokinys gali matyti gautą įvertinimą už tam tikrą užduotį. Buvo nuspręsta, jog kuriama sistema taip pat privalo turėti tokį funkcionalumą.

### Prototipai ir pagalbinė informacija

Kuriant sistemą buvo naudojama pagalbinė informacija. Norint, jog sistemoje būtų kodo kompiliatorius, buvo būtina padaryti kodo kompiliatoriaus integraciją. Jog tai įvykdyti buvo pasirinktas *Python* programavimo kalbos kompiliatorius *Skulpt Python* [4], bankiniams pavedimams *Paysera* [5], grafinės vartotojo sąsajos realizuotos, remiantis *Chakra UI* [6]biblioteka.

Projekto starto pradžia 2023 m. vasario 1d., apytiksliai kiekvieną dieną po 4-6 valandas. Bendrai sumoje tai yra apie 495 valandas. Į šias valandas įskaičiuojamas tiek sistemos kodo rašymas, dizaino kūrimas, testų rašymas, klaidų taisymas.

## Techninių galimybių analizė

„Learn Programming“ buvo realizuojama naudojant viena populiariausių pasaulyje programavimo kalbų – *C#* [7], ši kalba buvo pasirinkta atsižvelgiant į ateities perspektyvas, į norą ją išmokti bei susipažinti su kalbos įmantrumais didesniame projekte. Grafinė vartotojo sąsaja buvo realizuojama taip pat pasaulyje sparčiai populiarėjančia *TypeScript* [8] kalba, jos pasirinkimas taip pat buvo vien dėl ateities perspektyvų ir masišku populiarėjimu. Na, žinoma, svetainės kūrimui buvo naudojama taip pat pasaulyje populiaria atvirojo kodo *JavaScript* biblioteka – *React*, kuri yra skirta kurti grafinę vartotojo sąsają, pagrįstas komponentais. O komponentams realizuoti buvo naudojama *Chakra UI* [6] biblioteka.

Tiek *Skulpt Python* [4]*,* tiek *Paysera* [5]yra naudojami įvairiose svetainėse, o šių integracijų dokumentacijos yra viešai prieinamos.

Įvertinus taikomas technologijas, galima teigti, jog projektas yra įgyvendinamas. Kadangi visos reikalingos technologijos yra pasiekiamos ir naudojamos kituose sprendimuose, o ypač tai, jog programavimo mokymasis vyktų svetainėje ir kodas kompiliuotųsi pačioje naršyklėje. Todėl jokių papildomų kliūčių nėra.

# Projektas

## Reikalavimų specifikacija

### Komercinė specifikacija

Šis vykdomas projektas nėra komercinis užsakymas. Tai universitetinis projektas, kuris skirtas bakalauro baigiamajam darbui ir taip įrodyti sukauptas žinias per ketverius metus universitete.

Šios sistemos – „Learn Programming“ – sumanytojas ir vykdytojas – Kauno Technologijos Universiteto 4 – ojo kurso Programų sistemų studijų programos studentas Rokas Sičiovas.

Tai yra asmeninis projektas, todėl kainos apribojimai neegzistuoja, tačiau projektas numatytas baigti 2023m. gegužės 16d. Šią sutartą dieną yra vykdomas darbų atidavimas universitetui. Todėl prieš atiduodant projektą universitetui, jis privalo būti užbaigtas, ištestuotas ir paruoštas pristatymui.

Nuo birželio 5d. iki birželio 12d. bus vykdomi vieši gynimai, kurios metu sistema ir bus pristatyta.

### Sistemos funkcijos

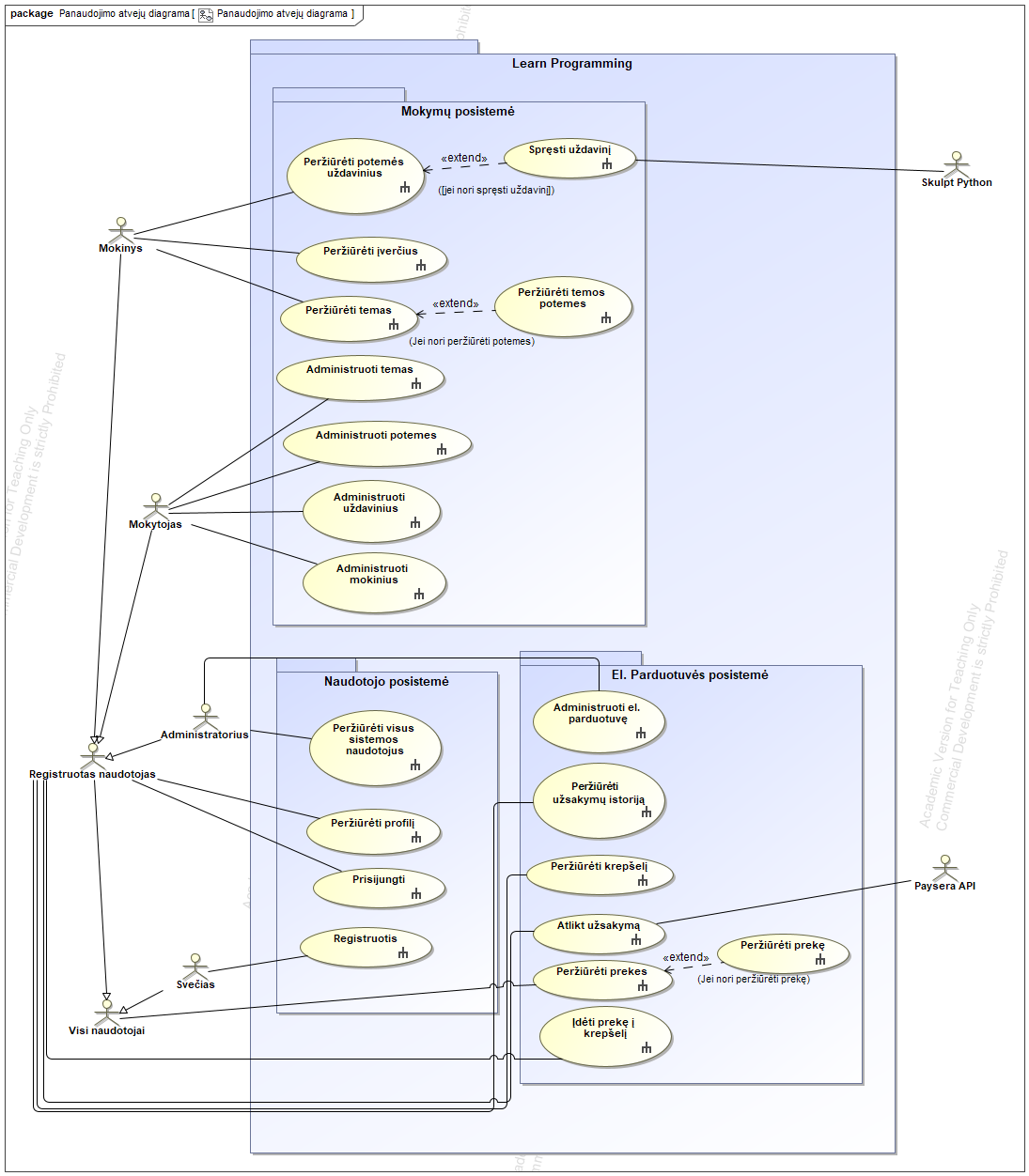
„Learn Programming“ sistemos funkciniai reikalavimai pateikti UML panaudojimo atvejų diagrama (angl. *Use Case Diagram*) ir veiklos diagramose (angl. *Activity Diagram*). 2.1 pav. yra pateikta visos sistemos panaudojimo atvejų diagrama.

Sistema yra atskirta į tris skirtingas posistemes – mokymų, naudotojų ir elektroninės parduotuvės posistemes. Mokymų posistemė yra atsakinga už temų, potemių, uždavinių, mokinių administravimą ir užduočių sprendimą, naudotojo posistemė yra atsakinga už paprastas funkcijas, kurias gali atlikti naudotojas su savo paskyra, tai registracija, prisijungimas, peržiūrėti profilį ar administratoriui peržiūrėti visos sistemos naudotojus, elektroninės parduotuvės posistemė atsakinga už elektroninės parduotuvės administravimą, peržiūrėti prekes, peržiūrėti krepšelį ir užsakymų istoriją, įdėti prekę į krepšelį ir, žinoma, atlikti užsakymą.

„Learn Programming“ sistemoje galima išskirti 5 pagrindinius aktorius:

* Administratorius – pagrinde atsakingas už elektroninės parduotuvės posistemę;
* Mokytojas – atsakingas už mokymų posistemę, gali kurti temas, potemes ir uždavinius bei prisiskirti mokinį ar atsisakyti;
* Mokinys – gali spręsti uždavinį, užsisakyti programavimo knygas, peržiūrėti savo įverčius;
* *Skulpt Python* [4] – programavimo kalbos *Python* kompiliatoriaus biblioteka;
* *Paysera API* [5] – trečios šalies sistema, teikianti atsiskaitymo paslaugas visame pasaulyje.

Visi posistemių panaudojimo atvejai yra pateikti UML diagramomis, kurios parodo supaprastintus vykdymo scenarijus. **2.1 pav.** pateikta „Learn Programming“ sistemos panaudojimo atvejų diagrama. **2.2 pav. – 2.21 pav.** paveikslėliuose yra pateikiami mokymų posistemės, naudotojo posistemės ir elektroninės parduotuvės posistemės veiklos diagramos.



**2.1** **pav.** „Learn Programming“ panaudojimo atvejų diagrama

**2.2 pav. – 2.11 pav.** pateikta mokymų posistemės veiklos diagramos.

Diagram, schematic

Description automatically generated

**2.2** **pav.** „Administruoti temas“ veiklos diagrama

Iš visų egzistuojančių rolių administruoti temas yra suteikta tik mokytojui. Paspaudęs mygtuką „Mano temos“, mokytojui bus atvaizduotas visas temų sąrašas. Norėdamas manipuliuoti duomenimis, jis turės galimybę temą ištrinti, redaguoti ar pridėti naują. Ištrinant temą, tereikės paspausti ant trynimo simbolio, pasirodžius perspėjimo langui ir paspaudus patvirtinimo mygtuką, tema bus ištrinta. Pridedant ar redaguojant temas, reikės suvesti duomenis: temos pavadinimą ir nustatyti kokio sunkumo tema bus, taip jog mokiniui bus lengviau pasirinkti ar lengvo lygio temą nori pasirinkti, ar vidutinio, ar sunkaus. Redaguojant temą, duomenys bus parnešti iš duomenų bazės ir pateikti formoje, norint atnaujinti ir parašyti naujus duomenis, reikės ištrinti jau egzistuojančius duomenis ir parašyti naujus. Įvykdžius pridėjimą, bus atnaujintas temų sąrašo langas ir iš karto bus matoma naujai pridėta tema.

Diagram, schematic

Description automatically generated

**2.3** **pav.** „Administruoti potemes“ veiklos diagrama

Kaip ir temas, taip ir potemes galės administruoti mokytojas. Veikimas yra identiškas. Tiek trynimas, tiek pridėjimas, tiek redagavimas. Todėl norint ištrinti, reikės paspausti ant trynimo simbolio ir pasirodžius perspėjimo langui, trynimą patvirtinti. Redagavimo metu duomenys yra parnešami iš duomenų bazės ir pateikiami formoje. Pridėjimo metu reikės parašyti tik potemės pavadinimą ir patvirtinti pridėjimą, patvirtinus langas bus automatiškai atnaujintas ir bus galima pamatyti naujai pridėtą potemę.

Patekimas į potemę vyksta iš temų sąrašo lango. Paspaudžius ant pasirinktos temos, atsidarys temos potemės.

Diagram, schematic

Description automatically generated

**2.4** **pav.** „Administruoti uždavinius“ veiklos diagrama

Uždavinius taip pat galės administruoti mokytojas. Pasirinkus potemę, pasimatys uždavinių sąrašo langas. Kaip matome uždavinį bus galima tik pridėti arba ištrinti. Nėra redagavimo mygtuko, kadangi kuriant uždavinį yra pridedami testų variantai ir norima išvengti, jog uždavinio redagavimas nevyktų jau mokiniams bent kartą išsprendus uždavinį. Norint pridėti temą, mokytojui bus parodoma pridėjimo forma, kurioje reikės įvesti uždavinio pavadinimą, užduoties aprašymą, pradinį kodo skeletą ir testų variantus – įvestį ir išvestį.

Diagram, schematic

Description automatically generated

**2.5** **pav.** „Administruoti mokinius“ veiklos diagrama

Mokytojas taip pat galės ir administruoti mokinius. Jis turės galimybę priskirti mokinį į savo sąrašą, atsisakyti mokinio, jog jis nebegalėtų matyti mokytojo sukurtų temų, potemių ir uždavinių ir, žinoma, peržiūrėti kiekvieno savo mokinio įverčius. Mokinio priskyrimas yra vykdomas, kuomet mokytojas paspaudžia mygtuką „Priskirti mokinį“, tuomet pasimatys visų mokinių sąrašas sistemoje ir mokytojas galės pasinaudoti filtravimu arba susirasti mokinį ir paspaudus mygtuką „Priskirti mokinį“ ir pasirodžius perspėjimo langui jį patvirtinus. Mokinio atsisakymas vyksta labai panašiai. Paspaudus mygtuką „Mano mokiniai“, mokytojas turės galimybę peržiūrėti jam priklausančių mokinių sąrašą. Prie kiekvieno mokinio bus mygtukas „Atsisakyti“, mokytojui jį paspaudus, pasirodys perspėjimo langas, patvirtinus tai – mokinys nebepriklausys mokytojui ir mokinys nebematys mokytojo sukurtų temų, potemių ir uždavinių. Mokytojas taip pat galės peržiūrėti savo mokinių įverčius, tereikės paspausti mygtuką „Mokinių įverčiai“ ir pasirodys visas jam priklausančių mokinių sąrašas su egzistuojančiais įverčiais.

Diagram

Description automatically generated

**2.6** **pav**. „Peržiūrėti temas“ veiklos diagrama

Kuomet mokinys bus priskirtas mokytojui, tuomet mokinys turės galimybę peržiūrėti mokytojo sukurtas temas. Paspaudęs mygtuką „Mano temos“, bus pateiktas temų sąrašas.

Diagram

Description automatically generated

**2.7** **pav**. „Peržiūrėti temos potemes“ veiklos diagrama

Mokiniui atsivertus temų sąrašą ir paspaudus ant pasirinktos temos, bus atvaizduotas temos potemių sąrašas.

Diagram

Description automatically generated

**2.8** **pav.** „Peržiūrėti potemės uždavinius“ veiklos diagrama

Mokiniui atsivertus temos potemių sąrašą ir paspaudus ant pasirinktos potemės, bus atvaizduotas potemės uždavinių sąrašas.

Diagram

Description automatically generated

**2.9** **pav.** „Spręsti uždavinį“ veiklos diagrama

Paspaudęs ant pasirinkto uždavinio, mokinys bus nukreiptas į uždavinio langą. Jame mokinys galės matyti temos, potemės ir uždavinio pavadinimus, perskaityti uždavinio aprašymą ir bus pateiktas *Python* programavimo kalbos kompiliatorius. Kompiliatoriuje mokinys galės rašyti kodą, kompiliuoti kodą ar gauti įvertį. Mokiniui parašius kodą ir paspaudus kompiliuoti, sistema kvies kompiliavimo funkciją, kuri kreipsis į *Skulpt Python* [4]*,* kompiliatoriui sukompiliavus kodą, bus pateiktas arba išvesties rezultatas, arba klaidos pranešimas, jei bus sintaksės klaidų. Mokiniui paspaudus mygtuką „Gauti įvertinimą“, bus vykdoma panaši eiga, kadangi bus kreipiamasi į *Skulpt Python* [4], jog būtų sukompiliuotas kodas, sukompiliavus kodą, bus pateiktas išvesties rezultatas ir taip pat vykdoma tiek iteracijų, kiek yra testo variantų, kuomet testų variantų nebeliks, bus apskaičiuojamas įvertis ir pateikiamas įvertis mokiniui. Jei mokiniui įvertinimas netiks, jis uždavinį galės perspręsti ir vėl paspaudus mygtuką „Gauti įvertinimą“, įvertis bus atnaujintas.

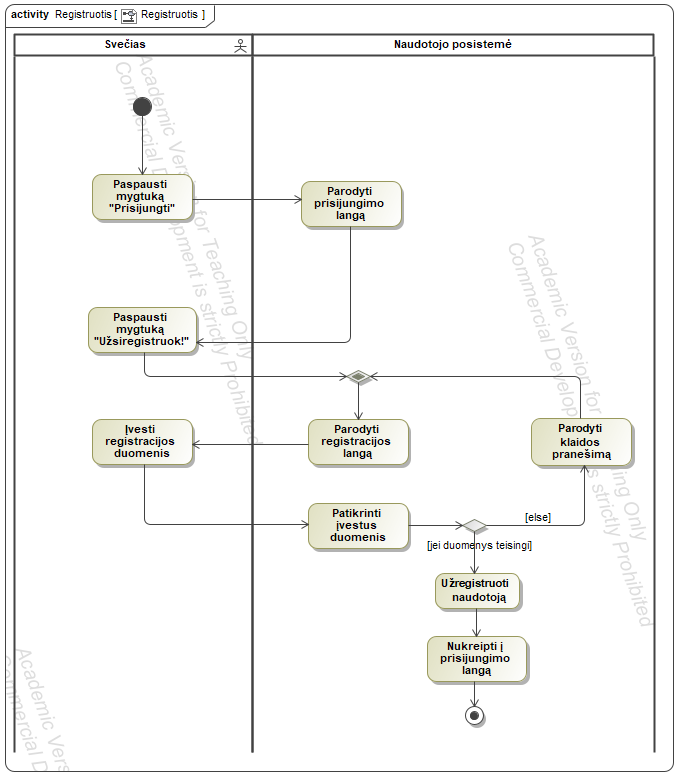
Diagram

Description automatically generated

**2.10** **pav.** „Peržiūrėti savo įvertinimus“ veiklos diagrama

Mokinys norėdamas peržiūrėti savo įverčius, reikia paspausti mygtuką „Įverčiai“ ir bus pateiktas įverčių sąrašo langas.

**2.11 pav. – 2.14 pav.** pateikta naudotojo posistemės veiklos diagramos.



**2.11** **pav.** „Registruotis“ veiklos diagrama

Norint užsiregistruoti, svečiui reikia paspausti mygtuką „Prisijungti“, pasirodžius prisijungimo langui, matysis mygtukas „Užsiregistruok!“, paspaudus šį mygtuką, pasirodys registracijos langas, užpildžius duomenis ir jeigu pastarieji yra teisingi, paskyra bus sukurta ir naudotojas bus nukreiptas į prisijungimo langą.

Diagram

Description automatically generated

**2.12** **pav.** „Prisijungti“ veiklos diagrama

Registruotam naudotojui norint prisijungti reikia paspausti mygtuką „Prisijungti“, pasirodžius prisijungimo formai reikės užpildyti duomenis, jei duomenys teisingi, naudotojas bus prijungtas prie paskyros.

Diagram

Description automatically generated

**2.13** **pav.** „Atnaujinti profilio informaciją“ veiklos diagrama

Kiekvienas registruotas naudotojas galės atnaujinti savo profilio informaciją, o gal ir net ištrinti paskyrą. Paspaudus mygtuką „Paskyra“, naudotojas bus nukreiptas į profilio langą, kuriuose jau bus registracijos metu užpildyta informacija, juos bus galima atnaujinti, taip pat ir slaptažodį bus galima pasikeisti. Įvykdžius tai, profilio langas bus atnaujintas ir nauja informacija pateikta. Naudotojas norėdamas atsisveikinti su puslapiu, galės ištrinti savo paskyrą ir taip panaikinti visus paliktus pėdsakus puslapyje. Paspaudęs mygtuką „Ištrinti paskyrą“, naudotojas privalės patvirtinti profilio trynimą perspėjimo lange, tai padaręs, paskyra bus ištrinta ir naudotojas bus perkeltas į pradinį svetainės puslapį.

Diagram

Description automatically generated

**2.14** **pav.** „Peržiūrėti visus sistemos naudotojus“ veiklos diagrama

Administratoriui norint peržiūrėti visus sistemos naudotojus, reikės paspausti mygtuką „Visi nariai“, po mygtuko paspaudimo bus atvaizduotas visų sistemos naudotojų sąrašas, kuriuose prie kiekvieno naudotojo bus pateiktas statusas – ar tai yra administratorius, ar tai yra mokytojas, ar mokinys yra priskirtas mokytojui, ar nepriskirtas.

**2.15 pav. – 2.21 pav.** pateikta elektroninės parduotuvės posistemės veiklos diagramos.

Diagram

Description automatically generated

**2.15** **pav.** „Administruoti el. parduotuvę“ veiklos diagrama

Administratorius turės visą prieigą administruoti elektroninę parduotuvę. Pridėti, ištrinti ar redaguoti prekę. Norint ištrinti prekę, administratoriui reikės paspausti trynimo simbolį, atsidarius perspėjimo langui, trynimą patvirtinti. Patvirtinus trynimą, prekė bus ištrinta ir bus pateiktas atnaujintas prekių sąrašas. Redaguojant prekę, duomenys bus paimti iš duomenų bazės ir pateikti prekės pridėjimo formoje. Pridedant prekę, bus galima pridėti nuotrauką, suteikti prekei pavadinimą, kainą, aprašymą, knygos kalbą, puslapių skaičių, leidėją, leidinio datą ir pasirinkti viršelį – minkštas viršelis arba storas viršelis. Pridėjus prekę, bus pateiktas atnaujintas prekių sąrašas, kuriame jau matytis naujai pridėta prekė.

Diagram

Description automatically generated

**2.16** **pav.** „Peržiūrėti užsakymų istoriją“ veiklos diagrama

Registruotam naudotojui, norint peržvelgti savo užsakymų istoriją, tereikės spustelėti mygtuką „Užsakymų istorija“ ir bus pateiktas užsakymų istorijos langas.

Diagram

Description automatically generated

**2.17** **pav.** „Peržiūrėti krepšelį“ veiklos diagrama

Registruotam naudotojui, norint peržvelgti krepšelį, tereikės spustelėti mygtuką krepšelio simbolį ir bus pateiktos krepšelyje esančios prekės.

Diagram

Description automatically generated

**2.18** **pav.** „Įdėti prekę į krepšelį“ veiklos diagrama

Registruotam naudotojui norint pridėti prekę į krepšelį reikės pasirinkti prekę, atsidarius prekės langui, bus mygtukas „Į krepšelį“, paspaudus šį mygtuką, prekė bus pridėta į krepšelį.

Diagram

Description automatically generated

**2.19** **pav.** „Peržiūrėti prekes“ veiklos diagrama

Visi naudotojai, nesvarbu ar tai registruotas, ar svečias, turės galimybę peržvelgti visas prekes, esančias sistemoje. Paspaudus mygtuką „El. Parduotuvė“ bus pateiktas prekių sąrašas.

Diagram

Description automatically generated

**2.20** **pav.** „Peržiūrėti prekę“ veiklos diagrama

Norint peržiūrėti pasirinktą prekę, tereikės paspausti ant jos ir bus atvertas prekės langas, kuriame bus galima perskaityti detalesnę informaciją apie prekę. Šį veiksmą galės atlikti visi naudotojai – registruoti ir neregistruoti naudotojai.

Diagram, schematic

Description automatically generated

**2.21** **pav.** „Atlikti užsakymą“ veiklos diagrama

Registruotas naudotojas turės galimybę atlikti užsakymą. Užsakymas yra atliekamas pasinaudojant *Paysera* [5]integracija. Ši bendrovė, teikianti elektroninio atsiskaitymo paslaugas, leis susimokėti norimu būdu už prekes. Tai visų pirma, naudotojui reikės turėti bent vieną prekę krepšelyje. Jei naudotojas turi bent vieną prekę krepšelyje, jis galės paspausti mygtuką „Toliau“. Spustelėjus šį mygtuką, pasirodys pristatymo forma, kurioje bus galima suvesti asmeninius duomenis ir pristatymo duomenis, kurie reikalingi būtent pristatymui. Spustelėjus mygtuką „Apmokėti“, bus patikrinti ar suvesti duomenys yra užpildyti ir ar užpildyti teisingai, nesėkmės atveju bus pateiktas klaidos pranešimas. Sėkmės atveju, naudotojas bus nukreiptas į *Paysera* [5] sistemą susimokėti. Pasirinkus mokėjimo būdą ir susimokėjus už prekes, duomenys bus išsaugoti ir pateiktas apmokėto užsakymo langas, kuriame naudotojas matys unikalų užsakymo numerį, datą ir sumokėta sumą ir taip prekes, kurias nusipirko. Taip pat ši informacija atsidurs ir užsakymų istorijoje.

### Naudotojo sąsajos specifikacija

„Learn Programming“ sistemos prototipo kūrimui buvo naudota *Figma*. Sistemos kūrimo metu buvo sukurta apytiksliai 15 eskizų internetinei svetainei. Toliau šiame skyriuje buvo pateikti tik svarbiausi „Learn Programming“ sistemos eskizai.

Žinoma, programuojant ir stebint kaip vystomas projektas, kai kurių funkcionalumų buvo atsisakyta. Projekte nebėra mygtuko „Prašymai“ ir viskas, kas buvo vadinama žodžiu „kursai“ buvo pakeista į žodį „temos“.

**2.21 pav. – 2.27 pav.** pateikiami eskizai iš *Figma* įrankio.

Matome, jog pagrindinio lango eskizo viršuje yra meniu, kuriame galime būti nukreipti į pagrindini langą paspaudę mygtuką „Pagrindinis“*,* taip pat iš mygtuko „Mano kursai“ būsime nukreipti į temų sąrašo langą, „El. Parduotuvė“ mygtukas nukreips į elektroninės parduotuvės prekių sąrašą. Toliau seka tekstas „Naudojatės mokytojo prieiga“, kuris identifikuoja naudotojo rolę. Krepšelio mygtukas nukreips į krepšelio sąrašą, kuriame užpildžius informaciją bus galima susimokėti už knygas per *Paysera* [5]. Paspaudus paskutinįjį dešinėje esantį mygtuką pasirodys meniu, kuriame pasirodys pasirinkimai, kurie nukreips į kitus galimus puslapius. Meniu pasirinkimai priklauso nuo vartotojo rolės.

A picture containing text, indoor

Description automatically generated

**2.22** **pav.** Pagrindinio lango eskizas

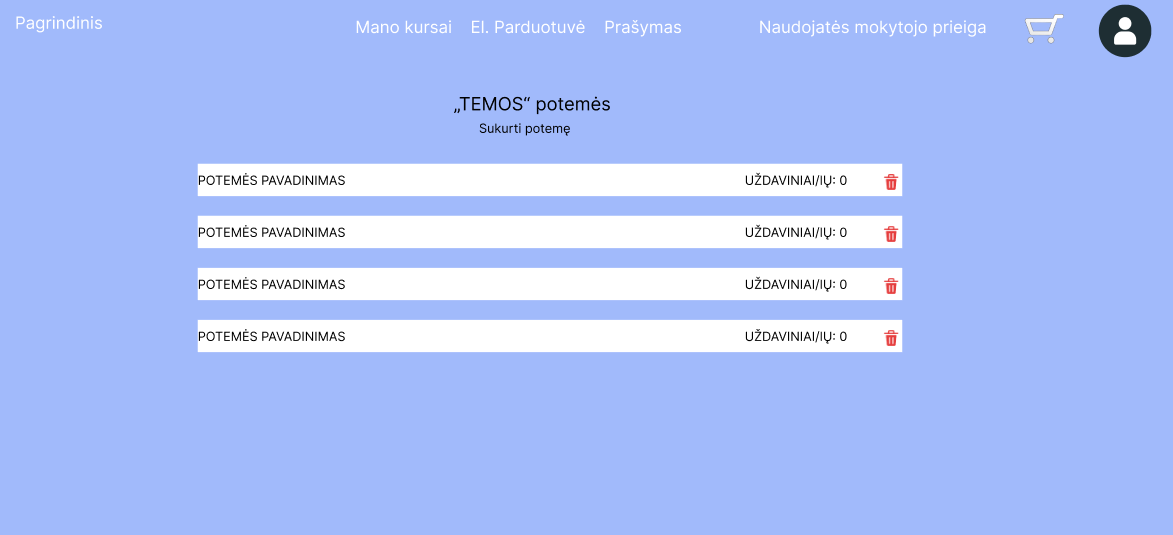
Temų lango eskize matome viršuje esantį ta patį meniu, kuris atlieka anksčiau išvardintas funkcijas. Pagrindinis dėmesys kreipiamas į kursų sąrašą, kuriame tik mokytojas gali kurti kursų temas, todėl yra mygtukas su pavadinimu „Pridėti naują temą“*.* Paspaudus jį, bus atidaromas modalinis langas, kuriame suvedus teisingą informaciją, yra sukuriama naujo kurso tema. Kurso langas susideda iš automatiškai priskiriamos nuotraukos, kurso temos pavadinimo, lygio sunkumo, kurse esančių potemių ir uždavinių skaičiaus ir, žinoma, vartotojas turintis mokytojo rolę, gali ištrinti sukurtą kurso temą.

Graphical user interface, application, PowerPoint

Description automatically generated

**2.23** **pav.** Kursų lango eskizas

Paspaudus bet kurią temą iš kursų esame nukreipiami būtent į unikalaus kurso potemes. Kuriame mokytojas gali pridėti naują potemę paspaudęs mygtuką „Sukurti potemę“, paspaudus šį mygtuką taip pat iššoks modalinis langas, kuriame reikės užpildyti naujos potemės pavadinimą. Šalia trynimo mygtuko matome, jog privalo būti pateikta, kiek potemėje yra jau sukurta uždavinių.



**2.24** **pav.** Kursų potemių lango eskizas

Paspaudus bet kurią potemę iš potemių sąrašo būsime nukreipti į unikalios potemės programavimo uždavinių sąrašą. Naujo uždavinio kūrimas vyksta paspaudus mygtuką „Sukurti uždavinį“, paspaudus šį mygtuką būsime nukreipti į naują langą, kuriame bus privaloma užpildyti laukus, norint patvirtinti uždavinio kūrimą.

Graphical user interface, application, table

Description automatically generated

**2.25** **pav.** Programavimo užduočių sąrašo lango eskizas

Programavimo užduoties langas susideda iš temos, potemės, uždavinio pavadinimo, integruoto *Python* programavimo kalbos kompiliatoriaus, užduoties aprašymo ir, žinoma, išvestis bei įvertis. Išvestyje galima pamatyti ir atsakymą, jei kodas yra teisingai parašytas ir nėra padaryta jokių sintaksės klaidų ir taip pat, jei yra padaryta sintaksės klaida, išvestyje paspaudus mygtuką „Kompiliuoti“, pamatysime klaidos pranešimą anglų kalba. Paspaudus oranžinį mygtuką „Gauti įvertinimą“ sistema automatiškai ištestuos programos kodą pagal pateiktus testų variantus ir bus pateiktas pažymys procentais (0 – 100 %).

Graphical user interface, application

Description automatically generated

**2.26** **pav.** Programavimo užduoties lango eskizas

Taip pat „Learn Programming“ edukacinėje sistemoje bus ir elektroninė parduotuvė, kurioje registruotas vartotojas turės galimybę nusipirkti programavimo knygų. Tik vartotojas, turintis administratoriaus rolę, turės galimybę pridėti naują prekę į elektroninės parduotuvės prekių sąrašą. Prekių sąraše bus matoma knygos nuotrauka, kaina, puslapių skaičius, kalba, jos pavadinimas.

Graphical user interface, website

Description automatically generated

**2.27** **pav.** Elektroninės parduotuvės prekių sąrašo lango eskizas

Vartotojui atsidarius unikalią prekę, bus galimybė peržiūrėti detalesnę prekės informaciją, kurioje taip pat ir bus mygtukas „Į krepšelį“*,* paspaudus šį mygtuką, prekė bus pridėta į krepšelį ir taip bus galimybė nusipirkti prekę ar prekes ir susimokėti per *Paysera* [5]sistemą*.*

Graphical user interface, application

Description automatically generated

**2.28** **pav.** Elektroninės parduotuvės prekės lango eskizas

*Figma* įrankis leidžia labai tiksliai atvaizduoti eskizus, kuriuos prieš atvaizduojant būname sukūrę jau mintyse. Pateikti tiksliais matmenimis, spalvomis, įvairūs komponentai. Tai puikus įrankis, kurį galima naudoti tiek rimtiems projektams, tiek asmeniniams projektams, kadangi yra įmanoma išreikšti visas norimas mintis ir paversti jas realybe.

### Realizacijai keliami reikalavimai

Sistemai buvo iškelti šie nefunkciniai reikalavimai:

* Naudotojo sąsaja turi būti realizuota lietuvių kalba;
* Sistema turi būti ištestuota bent vienetų testais;
* Kodas turi būti talpinamas kodo talpykloje *GitHub*;
* Sistema neturi atskleisti jokios asmeninės informacijos apie sistemos vartotojus;
* Grafinė naudotojo sąsaja turi būti palaikoma *Chrome, Opera, Microsoft Edge, Firefox* naršyklėse;
* Grafinė naudotojo sąsaja turi būti pritaikyta ir mobiliesiems įrenginiams, ir planšetėms.

### Techninė specifikacija

Norint naudotis „Learn Programming“ sistema yra reikalingas kompiuteris (namų ar viešas) su *Windows*, *Linux* ar *macOS* operacine sistema, turintis interneto prieigą bei naršyklę (*Microsoft Edge*, *Google Chrome, Opera* ar kita).

## Projektavimo metodai

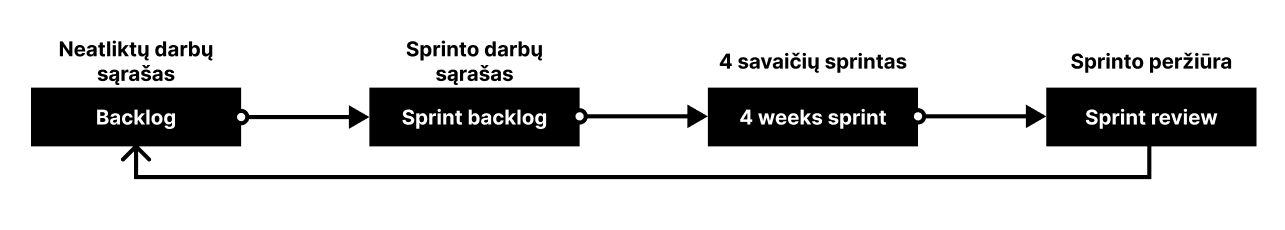
### Projektavimo valdymas ir eiga

Kadangi šio projekto autorius ir kūrėjas yra vienas asmuo ir projektas nėra daromas įmonei ar kartu komandoje, buvo galima pasirinkti kaip dirbti ir kaip dėliotis projekto darbus.

Daugiausia patirties turima dirbant *Agile* metodologija. Prieš pradedant programuoti yra suplanuojamos užduotys, kurios iš pat pradžių yra įdedamos į neatliktų darbų sąrašą (angl. *backlog*), tada vykdomas sprinto planavimas ir sprintui paskirtos užduotys yra įdedamos į sprinto darbų sąrašą (angl. *sprint backlog*). *Agile* metodologija dirbantys žmonės būna susitarę, kiek laiko trunka vienas sprintas (nuo 1 savaitės iki 4 savaičių).

Kuriant šį projektą nutarta, jog vienas sprintas truks 4 savaites, kadangi buvo aiškus darbų kiekis ir 4 savaitės buvo tinkamas laikas išspręsti sudėtas užduotis į sprinto darbų sąrašą.

Taikant *Agile* metodologiją visos sukurtos užduotys buvo sukeltos į neatliktų darbų sąrašą, prieš prasidedant naujam sprintui, naujos užduotys sudedamos į sprinto darbų sąrašą, prieš baigiantis sprintui yra peržvelgiama ar sprinto tikslas yra įvykdytas ir visos užduotys yra išspręstos. Užduotys, kurios nebuvo spėtos išspręsti, perkeliamos į naująjį sprintą. Šis visas planavimas yra pateiktas **2.29 pav.** diagramoje.



**2.29** **pav.** Sistemos kūrimui naudotas *Agile* metodologija

### Projektavimo technologija

Projektavimui pasirinktas *MagicDraw* įrankis. „Learn Programming“ sistema buvo suprojektuota naudojant *UML* grafinius elementus. Buvo suprojektuota klasių diagrama, panaudojimo atvejų bei veiklos diagramos, sekų diagrama ir komponentų diagrama.

Kiekviena sukurta diagrama yra apibūdinama tiksliu aprašymu su tikslu, jog būtų geriau suprasti, kas yra atvaizduojama ir kokia logika naudojama.

### Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos

Sistema buvo sukurta naudojant *C# (.NET Core 6)* programavimo kalbą, o grafinė vartotojo sąsaja buvo programuojama naudojant *TypeScript* programavimo kalbą, *JavaScript React* biblioteka. Duomenų bazė – *MySQL*, o duomenų bazės valdymo sistema – *MySQL Workbench 8.0 CE*. Programavimas buvo atliktas naudojant *Microsoft Visual Studio 2022* integruotoje programavimo aplinkoje ir *Microsoft Visual Studio* kodo redaktoriuje.

## Sistemos projektas

Sistemos projektas – tai jūsų sistemos veikimo aprašymas. Tai dažniausiai nagrinėjama dokumento vieta (be išvadų) darbo peržiūros ir gynimo metu.

### Statinis sistemos vaizdas

Šiame punkte reikėtų detalizuoti sistemos struktūrą. Priklausomai nuo projekto tipo turėtumėte aprašyti savo sistemą panaudodami UML diagramas:

* Diegimo (*UML deployment diagram*) – nepakeičiama tuo atveju, jei sistema naudoja išorinius servisus ar yra paskirstyta per keletą įrenginių. Geriausia pradėti nuo šios diagramos, nes ji greičiausiai supažindina su bendra sistemos sudėtimi.
* Komponentų (*UML component diagram*) – geriausiai tinka tuomet, kai naudojamas komponentinis sistemos kūrimo būdas ir sistema susideda iš komponentų teikiančių programavimo sąsają (API).
* Paketų (*UML package diagram*) – labai naudinga tuomet, jei sistema sugrupuota paketais.
* Klasių (*UML class diagram*) – geriausiai tinka atvaizduoti sistemos struktūros detales. Jei projekte klasių naudojama daug, rekomenduojama detalizuoti tik esmines klases, o likusią struktūrą pateikti paketų diagrama.
* Aprašant statinį sistemos vaizdą taip pat turėtų būti pateikta ir duomenų bazės schema. Šiam tikslui gali būti naudojama UML klasių diagrama ar jūsų naudojamų įrankių sugeneruotas duomenų bazės vaizdas. Jei naudojama ne reliacinė duomenų bazė, tuomet naudoti tokį duomenų bazės specifikavimo būdą, kurį siūlo kūrėjai arba bendruomenė.

Ne visas sistemas galima lengvai aprašyti UML notacija (pvz. esant funkcinėms kalboms vietoj klasių diagramų rekomenduojame naudoti komponentų ar paketų diagramas). Esant dar sudėtingesniems atvejams galima pasirinkti ir kitą notaciją ar diagramas, tačiau dėl to pasitarkite su vadovu.

### Dinaminis sistemos vaizdas

Dinaminiame sistemos vaizde parodoma kaip sistema veikia naudojama. Rekomenduojame aprašyti ne visus, tačiau tik esminius sistemos veikimo scenarijus. Šiame punkte pagal poreikį galima pavaizduoti sistemos veiksmus UML veiklos, sekų ir/arba būsenų diagramomis. Galite pasirinkti vieną iš jų, galite naudoti ir kelias (priklausomai nuo sistemos specifikos). Visgi jei naudojate veiklos diagramas, jos turėtų būti žemesnio abstrakcijos lygmens nei tas, kurias galimai naudojote aprašydami sistemos reikalavimus.

# Testavimas

## Testavimo planas

Testavimas buvo vykdomas dvejais etapais:

1. Statinė kodo analizė – vykdomas viso projekto metu.
2. Vienetų testais – taip pat vykdomas viso projekto metu. Parašius funkcionalumą, buvo parašomas ir testas, norint išsiaiškinti ar teisingai veikia parašytas metodas.

## Testavimo kriterijai

Norint, jog testavimas vyktų nuosekliai buvo nustatyti testavimo kriterijai, kurie buvo padalinti į dvi aiškias dalis – statinės kodo analizės kriterijus ir sistemos funkcionalumo kriterijus.

Statinės kodo analizės kriterijai:

* Projekto kode nepalikti nenaudojamų metodų, kintamųjų, konstantų ar bibliotekų;

Sistemos funkcionalumo kriterijai:

* Teisių apsauga. Sistema privalo užtikrinti, jog apsaugoti API galiniai taškai (angl. *endpoints*) negali būti pasiekti registruotų vartotojų, jei jie neturi reikalingų teisių;
* Puslapių apsauga. Sistema privalo užtikrinti, jog apsaugoti API galiniai taškai (angl. *endpoints*) negali būti pasiekti neregistruotų ar registruotų vartotojų, jei jie neturi privalomos vartotojo rolės;
* Validacija. Sistema privalo užtikrinti duomenų validaciją.

## Vienetų testavimas

Buvo atliktas automatinis vienetų testavimas, kurio metu buvo testuojami serverio pusės metodai. Vienetų testų rašymui buvo naudotas *Xunit* bei testų patikrai buvo naudotas *Stryker.NET,* kuris pateikdavo išsamią informaciją apie testuojamą funkcionalumą.

Table

Description automatically generated with low confidence

**3.1 pav.** „Stryker.NET Report“ analizė

Parašyta lygiai 76 vienetų testų serverio pusės metodams, o grafinė vartotojo sąsaja buvo testuojama rankiniu būdu.

Išsiaiškinti testų padengimo procentą buvo naudotas *Cobertura* generavimo įrankis.

Graphical user interface

Description automatically generated with medium confidence

**3.2 pav.** „Cobertura“ generavimo įrankis

## Naudotojo sąsajos testavimas

Naudotojo sąsajos testavimas buvo vykdomas rankiniu būdu pagal pateiktus UML panaudojimo atvejų diagramą. Kiekvienam panaudojimo atvejui buvo sukurtas testavimo scenarijus, o visi veiksmai buvo atliekami grafinėje naudotojo sąsajoje ir buvo stebima ar gaunamas teigiamas, ar neigiamas rezultatas. Nefunkciniuose reikalavimuose yra nurodyta, jog naudotojo sąsaja turi būti realizuota lietuvių kalba, naudotojo sąsaja turi būti palaikoma *Chrome*, *Opera*, *Microsoft Edge* bei *Firefox* naršyklėse ir, žinoma, naudotojo sąsaja turi būti pritaikyta mobiliesiems įrenginiams ir planšetėms.

Pirmasis nefunkcinis reikalavimas – naudotojo sąsaja turi būti realizuota lietuvių kalba. Tai yra įgyvendinta, nes kiekvieno mygtuko pavadinimas, kiekvienas žodis sistemoje yra pateikti lietuvių kalba.

Antrasis nefunkcinis reikalavimas – naudotojo sąsaja turi būti palaikoma *Chrome*, *Opera*, *Microsoft Edge* bei *Firefox* naršyklėse.

Trečiasis nefunkcinis reikalavimas - naudotojo sąsaja turi būti pritaikyta mobiliesiems įrenginiams ir planšetėms.

# Dokumentacija naudotojui

## Apibendrintas sistemos galimybių aprašymas

„Learn Programming“ edukacinė sistema leidžia mokytis *Python* programavimo kalbos ir pačio programavimo ypatumų sistemos svetainėje. Mokytojams priskyrus mokinį į savo sąrašą, mokinys turės galimybė matyti mokytojo sukurtus uždavinius. Mokinys gali matyti mokytojo sukurtas kursų temas, potemes bei pačius uždavinius. Tiek mokinys, tiek mokytojas gali pateikti administratoriui prašymą, taip išdėstant problemas apie puslapį ar užklausti norimo klausimo. Registruotas narys gali nusipirkti sistemoje esančias knygas, įsidėti knygas į krepšelį ir apmokėti per *Paysera* [5] sistemą.

## Naudotojo vadovas

Naudotojo vadovas yra neformalus įvadas į sistemą, aprašantis jos „normalų“ naudojimą. Kitaip tariant, naudotojui draugiška instrukcija su daug iliustracijų ir paaiškinimų. Neišvengiamai pradedantieji, nepriklausomai nuo patirties, daro klaidas. Lengvai randama informacija, kaip nuo šių klaidų grįžti prie naudingo darbo ir atstatyti galimus klaidų padarinius, turi būti sudėtinė šio dokumento dalis.

## Diegimo vadovas

Sistemos diegimo dokumentas yra skiriamas sistemos administratoriams (dažniausiai tai kompiuterius prižiūrintis personalas, tačiau šie žmonės nebūtinai būna ir sistemos naudotojai). Jame turi būti nurodytos diegimo konkrečioje aplinkoje detalės, turi būti supažindinama su sistemą sudarančiais failais, minimalia reikalingos techninės įrangos konfigūracija.

## Administravimo vadovas

Sistemos administratoriaus vadove turi būti aprašyti pranešimai, kaip sistema bendrauja su kitomis sistemomis ir kaip reaguoti į šiuos pranešimus. Būtų gerai nurodyti, kaip reaguoti į sistemos klaidas (sisteminių pranešimų paaiškinimai). Jei sistema apima ir techninę įrangą, jame turi būti aprašyti operatoriaus veiksmai palaikant šią techninę įrangą (pvz., kaip prijungti naujus periferinius įrenginius ir t.t.).

Rezultatai ir išvados

Bene svarbiausia viso darbo dalis – išvados. Išvados nenurodo, kas buvo padaryta darbe, bet pabrėžia atrastus dėsningumus, pastebėtas technologijų ar rinkos spragas, esminius įrangos privalumus. Išvados gali būti formuluojamos tik darbo metu sukurtos įrangos, technologijos, metodo ar susistemintos informacijos pagrindu (pvz., negalima cituoti šaltinių, vadovautis kitų autorių atrastais dėsningumais). Išvados numeruojamos, jų turėtų būti maždaug 4-9 (pvz., kūrimo etapams – reikalavimų analizei, projektavimui, realizacijai, testavimui, diegimui). Įprastai kiekviena išvada turėtų būti sudaryta iš atlikto veiksmo aprašymo ir gautų rezultatų. Išvadas galima gauti:

* Atlikus konkurentų analizę, kuomet būna išsiaiškinama esminiai konkurentų sistemų pranašumai ir trūkumai (pvz., „Buvo išanalizuotos analogiškos (konkrečiai nurodant kokios) sistemos, kurios pasižymėjo tokiais ir tokiais privalumais (apibendrintai), tačiau dėl tokių ar anokių trūkumų buvo nuspręsta kurti naują sistemą...“).
* Atlikus technologijų analizę, kuomet būna pagrindžiamas konkrečių programavimo kalbų, karkasų ar kitų technologijų pasirinkimas (pvz., „Išanalizavus x, y ir z technologijas buvo pasirinkta technologija z. Tai padėjo lengviau suprojektuoti, o vėliau ir realizuoti įrankio serverio pusės dalį, palaikyti vientisą programos kodo struktūrą...“).
* Atlikus testavimą, kuomet būna nurodoma kokį kodo padengimą pavyko pasiekti, kokias klaidas pavyko aptikti panaudojus pasirinktus testavimo metodus.
* Susidūrus su tam tikromis specifinėmis problemomis, kurioms išspręsti buvo panaudotas jūsų sugalvotas metodas („Kūrimo metu buvo susidurta su tokiomis ar anokiomis problemomis, kurios buvo sprendžiamos taip arba anaip...“). Galima įdėti ir išvadą apie nepasiteisinusius, tačiau jūsų išbandytus sprendimus (siekiant, kad kiti „neliptų ant to paties grėblio“). Jūsų parinkti problemų sprendimo būdai yra svarbios išvados, parodančios jūsų kompetenciją ir įsigilinimą į darbą.
* Realizavus pačią programą ar sistemą, kuri (greičiausiai) pakeitė ar pagerino iki tol vykusius verslo procesus (tai susiję su skyreliais „Bendras veiklos tikslas“ ir „Sistemos pagrindimas“) ar (jei tai buvo mokslinio pobūdžio darbas) tiesiog iki tol buvusius algoritmo / sprendimo rezultatus.

Šiame skyrelyje taip pat būtina pridėti ir papildomas išvadas-rezultatus apie tai:

* Kokia yra sistemos esamą būklė. Verta paminėti, jei sistema yra praktiškai naudojama įmonėje ar (programėlės kūrimo atveju) programėlė yra įkelta į Google Play ar AppStore parduotuvę.
* Kas planuojama atlikti tobulinant sistemą ateityje. Kadangi baigiamajam darbui sukurti yra skiriamas ribotas laikas, galbūt verta paminėti tas savybes, kurių dėl laiko apribojimų tiesiog nespėjote, bet planuojate įgyvendinti.

Literatūros sąrašas

|  |  |
| --- | --- |
| 1. [1] | 1. „Informiko Akademija,“ [Tinkle]. Available: https://konkursai.if.ktu.lt. |
| 1. [2] | 1. „KTU Open Courses,“ [Tinkle]. Available: https://open.ktu.edu. |
| 1. [3] | 1. „GDB Online Debugger,“ [Tinkle]. Available: https://www.onlinegdb.com. |
| 1. [4] | 1. „Skulpt Python,“ [Tinkle]. Available: https://skulpt.org/docs/index.html. |
| 1. [5] | 1. „Paysera,“ [Tinkle]. Available: https://developers.paysera.com/lt. |
| 1. [6] | 1. „Chakra UI,“ [Tinkle]. Available: https://chakra-ui.com. |
| 1. [7] | 1. „.NET Core,“ [Tinkle]. Available: https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0. |
| 1. [8] | 1. „React TypeScript,“ [Tinkle]. Available: https://www.typescriptlang.org/docs/handbook/react.html. |

Priedai

Papildoma informacija ir dokumentai, neįeinanti į pagrindinį dokumentą. Dažniausiai į priedus keliamos specifikacijos, sąsajų (API) aprašai, diagramos ar kita informacija, kuri yra svarbi projektui, tačiau užima dokumente ganėtinai daug vietos. Jei priedai nėra naudojami, šis puslapis yra ištrinamas.

1. priedas. Priedo pavadinimas

**2 lentelė.** Pagrindiniai baigiamojo projekto stiliai ir jų aprašymai

| **Stiliaus pavadinimas** | Stiliaus pavadinimas galerijoje | Stiliaus formalieji reikalavimai | Stiliaus naudojimo aprašymas |
| --- | --- | --- | --- |
| Antraštė non-TOC | Antraštė non-TOC | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, centruota lygiuotė. | Antraštėms, kurios nėra įtraukiamos į turinį: „Santrauka“, „Summary“, „Turinys“. |
| Antraštė be nr. | Antraštė be nr. | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, centruota lygiuotė, antraštė rašoma naujame puslapyje – po puslapio skirtuko. | Antraštėms, kurios įtraukiamos į turinį, bet nėra numeruojamos: „Lentelių sąrašas“, „Paveikslų sąrašas“, „Santrumpų ir terminų sąrašas“, „Įvadas“, „Išvados“, „Literatūros sąrašas“, „Informacijos šaltinių sąrašas“, „Priedai“. |
| 1. Heading 1,Skyrius | Skyrius | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas po antraštės – 10 pt, abipusė lygiuotė, antraštė rašoma naujame puslapyje – po puslapio skirtuko. | Skyrių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| 1.1. Heading 2,Poskyris | Poskyris | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, abipusė lygiuotė, numeracija siejama su aukštesnio lygio antrašte. | Poskyrių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| 1.1.1. Heading 4,Skyrelis | Skyrelis | Skyrelių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| Tekstas | Tekstas | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas prieš pastraipą – 10 pt, abipusė lygiuotė. | Tekstui visose projekto dalyse (santraukose, įvade, skyriuose, poskyriuose ir t.t.). |
| List Bullet;Sąrašas (suženklintas) | Sąrašas (suženklintas) | Pirmos pastraipos eilutės įtrauka – 0,63 cm, šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 0 pt, atstumas po sąrašo – 10 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas suženklintu sąrašu. |
| List Number;Sąrašas (numeruotas) | Sąrašas (numeruotas) | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 0 pt, atstumas po sąrašo – 10 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas sunumeruotu sąrašu. |
| Footnote Text;Išnašos tekstas | Išnašos tekstas | Šrifto dydis 10 pt, intervalas tarp eilučių – 1,15, atstumas prieš ir po sąrašo – 0 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas išnašose. |
| Lentelės pavad. | Lentelės pavad. | Šrifto dydis 11 pt, intervalas tarp eilučių – 1,15, atstumas prieš pavadinimą – 10 pt, po pavadinimo – 3 pt, lygiuotė prie kairiojo krašto. | Lentelių pavadinimams (numeris ir žodis lentelė rašomas paryškintu šriftu). |
| Lentelės I eil. | Lentelės I eil. | Šrifto dydis 10 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po pastraipos – 3 pt, lygiuotė prie kairiojo krašto. | Tekstui lentelės antraštinei (pirmai) eilutei. |
| Lentelė | Lentelė | Šrifto dydis 10 pt, intervalas tarp eilučių – 1, atstumas prieš ir po pastraipos – 3 pt, lygiuotė prie kairiojo krašto. | Tekstui lentelėje. |
| Caption,Paveikslo pavad. | Paveikslo pavad. | Šrifto dydis 11 pt, intervalas tarp eilučių – 1,15, atstumas prieš ir po pavadinimo – 10 pt, centruota lygiuotė. | Paveikslų pavadinimams (numeris ir santrumpa pav. rašoma paryškintu šriftu). |
| Figure;Paveikslas | Paveikslas | Atstumas prieš ir po paveikslo – 10 pt, centruota lygiuotė. | Paveikslui, iliustracijai . |
| Bibliography,Bibliografija | Bibliografija | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 2 pt, abipusė lygiuotė. | Literatūros ir Informacijos šaltinių sąrašuose nurodytiems šaltiniams. |
| Priedas | Priedas | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, lygiuotė prie kairiojo krašto. | Priedo numeriui, žodžiui *priedas*, priedo pavadinimui. |
| TOC 1,Turinys 1 | Turinys 1 | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, įtrauka – 0,64. | Turinyje esančioms antraštėms, kurios nėra numeruojamos („Lentelių sąrašas“, „Paveikslų sąrašas“, „Santrumpų ir terminų sąrašas“, „Įvadas“, „Išvados“, „Literatūros sąrašas“, „Informacijos šaltinių sąrašas“, „Priedai“) ir numeruojamai antraštei „Skyriaus pavadinimas“. |
| TOC 2,Turinys 2 | Turinys 2 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 0,96. | Turinyje esančiai antraštei „Poskyrio pavadinimas“. |
| TOC 3,Turinys 3 | Turinys 3 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 1,28. | Turinyje esančiai antraštei „Skyrelio pavadinimas“. |
| TOC 4,Turinys 4 | Turinys 4 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 0,64. | Numeruotiems priedams. |