****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Baigiamojo projekto pavadinimas**

Baigiamasis bakalauro studijų projektas

|  |
| --- |
|  |
| **Vardenis Pavardenis**  Projekto autorius / autorė |
|  |
| **Pareigų sutrumpinimas Vardenis Pavardenis**  Vadovas / Vadovė |
|  |

**Kaunas, 2019**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Baigiamojo projekto pavadinimas**

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

|  |  |
| --- | --- |
|  |  |
| **Vardenis Pavardenis**  Projekto autorius / autorė | (parašas) (data) |
|  |  |
| **Pareigų sutrumpinimas Vardenis Pavardenis**  Vadovas / Vadovė | (parašas) (data) |
|  |  |
| **Pareigų sutrumpinimas Vardenis Pavardenis**  Recenzentas / Recenzentė | (parašas) (data) |
|  |  |

**Kaunas, 2019**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

Vardenis Pavardenis

**Baigiamojo projekto pavadinimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Vardenio Pavardenio, baigiamasis projektas tema „Baigiamojo projekto pavadinimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (vardą ir pavardę įrašyti ranka) |  | (parašas) |

Autoriaus pavardė, vardas. Baigiamojo projekto pavadinimas. Bakalauro studijų baigiamasis projektas / vadovas / vadovė pareigų sutrumpinimas Vardenis Pavardenis; Kauno technologijos universitetas, Informatikos fakultetas.

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

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

Kaunas, 2019. 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.

Author’s surname, name. Title of the Final Degree Project. Bachelor's  Final Degree Project / supervisor abbreviation of the position, name and surname of the supervisor; Informatics Faculty, Kaunas University of Technology.

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

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

Kaunas, 2019. 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 8](#_Toc37514046)

[Paveikslų sąrašas 9](#_Toc37514047)

[Santrumpų ir terminų sąrašas 10](#_Toc37514048)

[Įvadas 11](#_Toc37514049)

[1. Analizė 12](#_Toc37514050)

[1.1. Techninis pasiūlymas 12](#_Toc37514051)

[1.1.1. Sistemos apibrėžimas 12](#_Toc37514052)

[1.1.2. Bendras veiklos tikslas 12](#_Toc37514053)

[1.1.3. Sistemos pagrįstumas 12](#_Toc37514054)

[1.1.4. Konkurencija rinkoje 13](#_Toc37514055)

[1.1.5. Prototipai ir pagalbinė informacija 13](#_Toc37514056)

[1.1.6. Ištekliai, reikalingi sistemai sukurti 14](#_Toc37514057)

[1.2. Galimybių analizė 14](#_Toc37514058)

[1.2.1. Techninės galimybės 14](#_Toc37514059)

[1.2.2. Vartotojų pasiruošimo analizė 14](#_Toc37514060)

[2. Projektas 15](#_Toc37514061)

[2.1. Reikalavimų specifikacija 15](#_Toc37514062)

[2.1.1. Komercinė specifikacija 15](#_Toc37514063)

[2.1.2. Sistemos funkcijos 15](#_Toc37514064)

[2.1.3. Vartotojo sąsajos specifikacija 21](#_Toc37514065)

[2.1.4. Realizacijai keliami reikalavimai 21](#_Toc37514066)

[2.1.5. Techninė specifikacija 22](#_Toc37514067)

[2.2. Projektavimo metodai 22](#_Toc37514068)

[2.2.1. Projektavimo valdymas ir eiga 22](#_Toc37514069)

[2.2.2. Projektavimo technologija 22](#_Toc37514070)

[2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos 22](#_Toc37514071)

[2.3. Sistemos projektas 22](#_Toc37514072)

[2.3.1. Statinis sistemos vaizdas 22](#_Toc37514073)

[2.3.2. Dinaminis sistemos vaizdas 23](#_Toc37514074)

[3. Testavimas 24](#_Toc37514075)

[3.1. Testavimo planas 24](#_Toc37514076)

[3.2. Testavimo kriterijai 24](#_Toc37514077)

[3.3. Komponentų testavimas 24](#_Toc37514078)

[3.3.1. Automatinis testavimas 24](#_Toc37514079)

[3.3.2. Rankinis API testavimas 26](#_Toc37514080)

[3.4. Statinė kodo analizė 28](#_Toc37514081)

[3.5. Vartotojo sąsajos testavimas 29](#_Toc37514082)

[4. Dokumentacija naudotojui 30](#_Toc37514083)

[4.1. Apibendrintas sistemos galimybių aprašymas 30](#_Toc37514084)

[4.2. Vartotojo vadovas 30](#_Toc37514085)

[4.3. Diegimo vadovas 30](#_Toc37514086)

[4.4. Administravimo vadovas 30](#_Toc37514087)

[Rezultatai ir išvados 31](#_Toc37514088)

[Literatūros sąrašas 32](#_Toc37514089)

[Priedai 33](#_Toc37514090)

[1 priedas. Priedo pavadinimas 33](#_Toc37514091)

Lentelių sąrašas

[**1 lentelė.** Konkurentų apžvalga 13](#_Toc3716736)

[**2 lentelė.** Pagrindiniai baigiamojo projekto stiliai ir jų aprašymai 22](#_Toc3716737)

Paveikslų sąrašas

[**1 pav.** Sistemos panaudojimo atvejų diagrama 15](#_Toc3716738)

Santrumpų ir terminų sąrašas

**Santrumpos:**

Doc. – docentas;

Lekt. – lektorius;

Prof. – profesorius.

**Terminai:**

**Saityno analitika** – 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.

**Tinklaraštis** – 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.

Beje, darbe rekomenduojame pateikti tik svarbesnes ir mažiau žinomas santrumpas bei terminus (tarkime tokių santrumpų kaip HTML, PC, IT paaiškinti nereikia)

Įvadas

Modernioje visuomenėje dėl vis didėjančio gyvenimo tempo tampa vis sunkiau planuoti ir paskirstyti darbus. Atlikti paskirtas užduotis, neatsilikti nuo darbo grafiko padeda išankstinis planavimas. Be tokio planavimo neretai atidedami svarbūs, net būtini, darbai, o vėlavimas juos atlikti žmogui sukelia stresą, kenkiantį jo fizinei ir psichologinei būklei, o įmonėms - grėsmę patirti nuostolius.

Mano sistemos tikslas išspręsti problemas, susijusias su laiko planavimu, paversti jį patogesniu, taip taupant vartotojo laiką. Norėdamas padėti išspręsti šią problemą sugalvojau sukurti programinę įrangą, kuri būtų patogi vartotojui, bei neperkrauta nereikalingo funkcionalumo. Tai bandysiu pasiekti atlikęs šios uždavinius:

1. Atlikti analizę
2. Palyginti kuriamą sistemą su konkurentais.
3. Sudaryti kuriamos sistemos projektą
4. Realizuoti sistemą
5. Ištestuoti sistemą
6. Parengti dokumentaciją

Darbas struktūrą sudaro šios pagrindinės dalys:

1. Analizė

Apžvelgiama į problemą, kuri bus išspręsta sukūrus sistemą. Apžvelgiami rinkoje jau esantys konkurentai, bandoma ieškoti kuriamos sistemos pranašumo.

1. Projektavimas

Šioje dalyje pristatomas projektas, jam iškelti reikalavimai, projektavimo metodika bei kūrimui naudotos technologijos.

1. Dokumentacija

Šioje dalyje pateikiama API specifikacija, vartotojo, diegimo bei administravimo vadovai.

1. Testavimas

Šioje dalyje matomas parengtas testavimo planas, pristatomas jo atlikimas ir gauti rezultatai.

Sistemos apimtis:

Bendras kodo eilučių kiekis : ~46000

* Testavimui-555
* Back-end-2144
* Front-end-1757517515

Puslapių skaičius : 9

# Analizė

Su darbo problematika susijusios informacijos analizė (4 – 8 lapai). Skyriaus pavadinimas ir struktūra priklauso nuo baigiamojo darbo specializacijos ir pačios temos specifikos.

## Techninis pasiūlymas

### Sistemos apibrėžimas

Laiko planavimo sistema tai interneto svetainė skirta visiems kas turi naršyklę, bei moka ja naudotis. Sistema turėtų sudaryti šios posistemės.

1. Pakvietimų posistemė
2. Autentifikavimo posistemė
3. Tvarkaraščio posistemė
4. Pranešimų posistemė
5. Nustatymų posistemė

Laiko planavimo sistema padeda :

* Žmonėms lengviau planuoti veiklas
* Įmonėms lengviau organizuoti darbo procesą, paskirstyti darbus darbuotojams
* Prioretizuoti darbus.

### Bendras veiklos tikslas

Padėti lengviau susiplanuoti, paskirstyti darbus tiek žmonėms, tiek ir įmonėms. Taip pat vienas iš svarbiausių sistemai iškeltų tikslų būtų paprastumas, kad kiekvienas sugebėtų išmokti naudotis per nedidelį laiko tarpą.

### Sistemos pagrįstumas

Žmonės visada skuba, ir laiko taupymas jiems padeda nuveikti daugiau. Tai pagrindinė priežastis kodėl kuriama sistema yra reikalinga. Nesvarbu ar esi paprastas vartotojas, kuriam patogiau jau ryte turėti svarbių veiklų tvarkaraštį, nes tai padaro jo dieną produktyvesnę, ar žmogus atsakingas už kitų žmonių atliktus darbus ši sistema sumažins tau problemų dėliojant grafiką. Sudėtingas skaičiavimo algoritmas atsižvelgs į tavo nustatytus prioritetus ir taip iškels svarbiausius darbus į priekį.

Dar vienas svarbus faktas panašių sistemų trūkumas( daugiau apie tai kitame skyriuje).

Taigi iš tiesų tokia sistema skirta tiems kurie nori viską atlikti laiku.

Aprašoma pagrindinė problema, kodėl reikia kurti sistemą (kitaip tariant, iš kur kilo toks poreikis) bei sistemos aktualumas (kodėl svarbu sukurti sistemą).

Pvz.: „Egzistuoja daug vartotojų, kurie negali arba tiesiog nenori eiti į parduotuvę ar kavinę norėdami pavalgyti. Esamos sistemos (detali analizė pateikiama sekančiame skyrelyje) nesuteikia galimybės greitai ir pigiai patenkinti maistinius poreikius, todėl šios sistemos sukūrimas iš esmės išsprendžia greito ir saugaus maisto perdavimo problemas“.

### Konkurencija rinkoje

Dauguma esamų produktų yra skirti laiko sekimui.

Panašių egzistuojančių ar šiuo metu dar tik kuriamų produktų trumpa apžvalga.

Pvz.: „Šiuo metu egzistuoja portalai, kurie leidžia užsisakyti maistą į namus (portalo aprašymas ir funkcionalumas 1, portalo aprašymas ir jo funkcionalumas 2, <...>). Tačiau kol kas mano kuriamai sistemai nėra analogų“.

Konkurentų apžvalgai iliustruoti siūloma pateikti lyginamosios analizės santrauką lentelės pavidalu (pavyzdys – **1 lentelė**). Palyginimui svarbu pasirinkti kriterijus, pagal kuriuos įmanoma objektyviai palyginti jūsų kuriamą sistemą su konkurentais. Taip rekomenduojama, kad (pagal poreikį) kriterijai būtų įvairūs – būtų palyginamos ne tik funkcionalumas, bet ir vartotojų kiekis, kaina, operacinė sistema ar kitos ypatybės.

**1.1** **lentelė.** Konkurentų apžvalga

| **Lyginimo kriterijai** | Sistema A | Sistema B | Sistema C |
| --- | --- | --- | --- |
| Savybė 1 | Realizuota | Nerealizuota | Realizuota iš dalies |
| Savybė 2 | 1000 naudotojų[[1]](#footnote-2) | 5000 naudotojų | 20000 naudotojų |
| Savybė 3 | Android | iOS | Android |
| Savybė 4 | + | + | - |
| Savybė 5 | 3.99€ | 19.99€ | Nemokama |
| ... | ... | ... | ... |

Po lentele taip pat rekomenduojama aprašyti palyginimo kriterijus – ką jie reiškia, kodėl jie svarbūs, kodėl buvo pasirinkti.

### Prototipai ir pagalbinė informacija

Programą ar sistemą galima kurti ir turint tam tikrą pagrindą, t.y. modifikuojant ar ištobulinant jau esamą produktą. Jei kuriate produktą nesinaudodami kitu produktu kaip pagrindu, taip ir parašykite. Jei naudojatės prototipais, detaliai aprašykite kuo jūsų sprendimas yra geresnis už prototipą. Čia taip pat galite abstrakčiai parašyti kokie informaciniai šaltiniai[[2]](#footnote-3) jums labiausiai padėjo kuriant produktą (pvz., naudojotės prototipų dokumentacija ar tiesiog pasikliovėte eksperto žiniomis, klausinėjote forumuose).

Šiuo metu labiausiai paplitęs yra failų perdavimo protokolas (FTP). Vienas labiausiai išvystytų protokolų yra Bittorent, tačiau egzistuoja ir daug kur kas mažiau paplitusių protokolų. Tai patikimi, rinkoje įsitvirtinę protokolai. Jų didžiausias trūkumas yra tas, kad jie gali perduoti tinklu tik duomenis, tačiau perduoti objektus (ypač maistą) yra neįmanoma. Savo darbe daugiausiai remsiuosi FTP protokolu, nes kuriamas produktas skiriamas parduotuvei, kuri persiunčia nupirktą maistą klientams, t.y. veikia kaip serveris kliento-serverio architektūroje. Bittorent labiau tiktų maisto dalybai socialiniuose tinkluose“.

### Ištekliai, reikalingi sistemai sukurti

Sistemą kūrimas užtuko apie 2 mėnesius. Tai pat dar 1 mėnesis buvo skirtas dokumentacijos rengimui, į kurią įėjo dokumentacijos rengimas, bei vartotojų gido parengimas. Norint tiksliau plėsti sistemą reikalinga vartotojų poreikių analizė dėl papildomo ar nereikalingo funkcionalumo. Jei funkcionalumas didėtų, gali tekti ir praplėsti personalą. Augant sistemai gali prireikti ir administratoriaus.

## Galimybių analizė

### Techninės galimybės

Sistemai realizuoti techninių kliūčių nėra. .Net Core technologija yra labai plačiai naudojama visame pasaulyje. Tai pat Angular karkasas yra vienas iš populiariausių ir dažniausiai naudojamų Javascript karkasų pasaulyje.

Vienintelė didesnė problema itin sudėtingas algoritmas skirtas darbų išdėstymui pagal laiką. Norint jį patobulinti yra tikimybė jo kodo eilučių skaičius išaugtų iki 1000 eilučių. Taip pat jo sudėtingumas padidėtų daug kartų. Todėl pasirinkta supaprastina, tačiau dalinai funkcionuojanti jo versija.

Kartais sukurti galutinį produktą nepakanka finansų, žmogiškųjų resursų ar kitų išteklių. Kliūtys gali kilti tiek įmonės viduje (pvz., per smulki įmonė tokiam projektui realizuoti), tiek globalios (pvz., rinkoje per mažai paplitusi technika, kurią būtų galima naudoti su kuriama programine įranga ar vartotojai dar nėra pasiruošę priimti produkto). Tokiu atveju reiktų detaliai aprašyti visas kliūtis, dėl kurių neįmanoma iki galo realizuoti pradinės idėjos.

Pvz.: „Šiuo metu maisto perdavimo protokolui trūksta infrastruktūros (vamzdžių, kuriais būtų galima perduoti maistą), be to rinkoje nedaug įrenginių, kurie gali būti suderinami ar panaudojami bendroje protokolo veikloje. Prognozuojamas infrastruktūros ir įrenginių išplitimas – apytiksliai po 10 metų“.

### Vartotojų pasiruošimo analizė

Vartotojas gali būti tiek individualus asmuo, tiek ir įmonė.

# Projektas

Aprašoma detali sistemos specifikacija (20 – 50 lapų). Apibrėžiama kuriamo produkto vizija (koncepcija). Skyriaus struktūra ir pavadinimas priklauso nuo baigiamojo darbo specializacijos ir pačios temos specifikos, bet turi turėti funkcinių ir nefunkcinių reikalavimų skyrius.

## Reikalavimų specifikacija

### Komercinė specifikacija

Projektas sukurtas specialiai baigiamajam bakalauriniam darbui. Projekto užsakovas ir vadovas doc. Sigitas Drąsutis. Visus darbus, tiek programavimą, projektavimą, dokumentavimą. tiek ir rinkos analizę atliko studentas Mantvydas Razulevičius.

Biudžetas nėra planuojamas, kadangi tai ne komercinis projektas. Tik projekto sėkmės atveju planuojamos apmokestintos paslaugos. Sistemos naudotojai būtų tiek paprasti žmonės visame pasaulyje (reikia mokėti anglų kalbą, nes pradinėje versijoje jų daugiau nėra), tiek ir įmonės.

Trukmė pilnam paruošimui (programavimui ir dokumentacijai) apie 3 mėnesius. Pradžia – 2020 sausio 26d. Tiksli pabaigos data nėra numatyta, tačiau gegužės pradžioje projektas jau turi būti paruoštas pristatymui.

### Sistemos funkcijos

Sistemos reikalavimai išskirstomi į šias pagrindines posistemes :

1. Pakvietimų posistemė
2. Autentifikavimo posistemė
3. Tvarkaraščio posistemė
4. Pranešimų posistemė
5. Nustatymų posistemė

Toliau bus pateikta visi funkciniai reikalavimai, bet kiekvienos posistemės analizė bus pateikta atskirai.



Pav 2.1 Sistemos panaudojimo atvejų diagrama

Pradžioje pateikiam diagrama (Pav 1), kuri parodo visą bendrą sistemos atliekamas funkcijas. Diagramoje aiškiai matyti visus sistemos veiklai būtinos dalys, bendras jų funkcionalumas. Visos posistemės turi susijusio funkcionalumo, pavyzdžiui pranešimų positemė yra naudojama daugumoje kitų sistemos dalių.



Pav 2.2 Autentifikavimo posistemė

Autentifikavimo posistemė – viena svarbiausių visoje sistemoje. Ji užtikrina, kad vartotojas matytų tik savo informaciją, pavyzdžiui darbus, tvarkaraštį. Taip vartotojas gali matyti įvykdęs prisijungimą prie sistemos . Kita svarbi funkcija registracija. Jos forma panaši į daugelį šiuo metu internete egzistuojančių formų. Taip pat svarbu paminėti, jog po atsijungimo vartotojas nebegali pasiekti savo informacijos iki tol kol vėl prisijungs.



Pav 2.3 Darbų posistemė

Darbų posistemė – tai dėl ko buvo kuriamas šis projektas. Darbų sąrašas, naujų užduočių pridėjimas ir šalinimas. Kurdamas naują darbą vartotojas turi galimybę jį prioretizuoti. Redagavimas atliekamas iš sąrašo pasirenki viena darbą ir pakeiti jo informaciją, bei turi galimybę keisti ir prioritetą. Informacijos peržiūra atliekama prieš jį redaguojant, arba gavus pakvietimą atlikti darbą.



Pav 2.4 Nustatymų posistemė

Nustatymai yra svarbi sistemos dalis kadangi tai leidžia vartotojui patogiau jaustis sistemoje. Pavyzdžiui kai nustatomas puslapio dydis 10 elementų, tokiu atveju visoje sistemoje kur yra pasiskirstymas puslapiai bus atsižvelgta į prieš tai nustatytą reikšmę. Darbo pražios ir pabaigos nustatymai leis atsižvelgi į vartotojo norus ir apskaičiuoti darbams tinkamas vieta pagal šiuos laiko rėžius. Jei vartotojas mano, jog paskyra jam nereikalinga, jam suteikiama teisė ją pašalinti.



Pav 2.5 Pakvietimų posistemė

Pakvietimų posistemė skirta paprasčiau dalintis darbais. Vartotojui galimą išsiūti kvietimą kitam sistemoje registruotam asmeniui. Kvietimą gavęs asmuo gali jį priimti arba atmesti. Kaip ir visus čia turėtu būti filtras skirtas palengvinti reikalingų duomenų suradimą.



Pav 2.6 Pranešimų posistemė

Pranešimų posistemė – skirta vartotojo informavimui apie tam tikrus įvykius. Pagrindinis funkcionalumas yra atliekamas sistemos, kuri kuria pranešimus ir perduoda juos vartotojui. Vartotojas čia neturi daug funkcijų, tačiau gali peržiūrėti ir pašalinti pranešimus. Taip pat kaip ir daugelyje posistemių yra pranešimų filtravimo galimybė.



Pav 2.7 Tvarkaraščio posistemė

### Vartotojo sąsajos specifikacija

Vartotojo sąsajai reikalavimai sudaryti nebuvo.

### Realizacijai keliami reikalavimai

Realizacijai gali būti keliami tokie **nefunkciniai reikalavimai**: reikalavimai sistemos išvaizdai, reikalavimai panaudojamumui, reikalavimai vykdymo charakteristikoms, reikalavimai veikimo sąlygoms, reikalavimai sistemos priežiūrai, reikalavimai saugumui, kultūriniai-politiniai reikalavimai, teisiniai reikalavimai. Jie išvardinami ir aprašomi šiame skyrelyje. Pavyzdžiui:

1. Maisto perdavimo protokolas privalo būti saugus (neautentifikuoti kaimynai negali sužinoti kokį maistą siunčiasi vartotojas)
2. Maistas, suskaidytas paketais, turi pasiekti vartotoją nepagedęs
3. Maisto perdavimo protokolas turėtų palaikyti lietuviškos virtuvės produktus

Šiame punkte gali būti išvardinti (jeigu nustatyti) tokio tipo apribojimai: apribojimai sprendimui, diegimo aplinka, bendradarbiaujančios sistemos, komerciniai specializuoti programų paketai, numatoma darbo vietos aplinka, sistemos kūrimo terminai, sistemos kūrimo biudžetas. Jei reikalinga specifinė duomenų kontrolė (kokia informacija turi būti tikrinama įvedimo ar sistemos veikimo metu), ji taip pat aprašoma šiame skyrelyje.

### Techninė specifikacija

Skyrelyje aprašykite techninę ir papildomą programinę įrangą, reikalingą sistemai. Nurodykite minimalius įrangos parametrus. Šis skyrelis, priklausomai nuo situacijos, gali būti formuluojamas kaip sąrašas, ką užsakovams reikės turėti, jeigu norės naudotis sistema, arba kokios aplinkos reikalauja užsakovas.

Pvz.: „Maisto perdavimo protokolui realizuoti būtinas interneto ryšys, 1 kilolitro maisto ir gėrimų perdavimo linija, specializuotas šaldytuvas, namų kompiuteris“.

## Projektavimo metodai

### Projektavimo valdymas ir eiga

Šiame punkte nurodykite, kokį programinės įrangos kūrimo modelį (ar modelius) naudojote kurdami savo sistemą. Tai gali būti krioklio, iteracinis ar kitas modelis. Galite nurodyti kaip suskirstėte darbus ir kokiu eiliškumu juos atlikote.

### Projektavimo technologija

Modeliavimui naudotas įrankis Magic Draw, kuris palaiko UML modeliavimo kalbą.

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

Sistema sukurta C# programavimo kalba. Programa sukompiliuojama ir jos paketai vykdomi .Net Core 3.1 aplinkoje. Sistema kurta Windows aplinkoje, tačiau naudojant .Net Core veiktų ir Linux ar MacOs aplinkose.

Programavimui naudota Windows10 aplinka, bei Visual Studio 2019 programavimo aplinką.

Duomenų bazė Microsft Sql Server Managment Studio 2019. Tai reliacinė duomenų bazė.

Vartotojo sąsajai programuoti naudota Angular 8 karkasas, kuris yra paremtas TypeScript programavimo kalba, kartu su HTML5 žymėjimo ir CCS3 stilių aprašymo kalbomis.

Testavimu naudota Xuint testavimo įrankių paketas.

## 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 (rekomenduojame pasikonsultuoti su vadovu) turėtumėte aprašyti savo sistemą panaudodami UML diagramas:

* Išdėstymo (*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 esybių ryšių diagrama arba (geriausia) UML klasių diagrama. Jei naudojama ne reliacinė duomenų bazė, tuomet naudoti tokį duomenų bazės specifikavimo būdą, kurį siūlo kūrėjai arba bendruomenė.

### Dinaminis sistemos vaizdas

Dinaminiame sistemos vaizde parodoma kaip sistema veikia naudojama. 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).

# Testavimas

Skyrius skirtas testavimo plano sudarymui, bei jo įgyvenimui. Planuoju pateiki ir naudojamas technologijas bei aprašyti gautus rezultatus.

## Testavimo planas

Sistemai būtinas testavimas įsitikinti, kad jo veikimas atitinka projektavimo metu nustatytus tikslus. Kad tikslus įgyvendinti teisingai buvo sudarytas toks testavimo planas:

1. Atlikti kiekvieno serviso komponentų testus.
2. Atlikti API metodų testus naudojant Swagger įrankį.
3. Atlikti statine kodo analizė, nepaliekant kodo dalių kurios ateityje gali turėti įtakos klaidos atsiradimui.
4. Atlikti svetainės grafinės vartotojo sąsajos testavimą rankiniu būdu

## Testavimo kriterijai

Kad testavimas būtų sėkmingai įvykdytas, turi būti įgyvendinti šiek kriterijai:

1. Servisų kodo padengimas virš 90%.
2. API testavimas visada gražina rezultatus kurių tikėtasi.
3. Statinė kodo analizė negali turėti nei vieno klaidos pranešimo, taip pat kuo mažiau perspėjimų.
4. Visi sistemos funkcijos turi būti atliekamos taip kaip nurodyta specifikacijoje.

## Komponentų testavimas

### Automatinis testavimas

Komponentų testavimas bus naudojamas tik servisams, nes juose naudodami visi kiti metodai. Taip jie bus ištestuojami kartu. Kontrolerių testuoti nėra prasmės, jie tik kviečia servisus.

Servisų kodo padengimo kriterijus įgyvendintas sėkmingai, kadangi pateiktas 90 % kriterijus yra pasiektas.



**3.1 pav. Kodo padengimas**

Plačiau apie komponentų testavimą galima pasakyti, jog atlikti 56 testai. Kaip matome iš paveikslėlio (3.1 pav.) visi testai buvo sėkmingi. Kodo padengimas svyruoja nuo 81% iki 100%. Tas parodo, kad testavimas atliktas gan nuosekliai, atsižvelgiant daugumą galimų scenarijų.

Iš to galime teigti, kad komponentų testavimas buvo įvykdytas. Svarbu paminėti jo testuoti ne visi metodai bei klasės. Kontrolerių testavimas aprašomas kitame skyriuje.

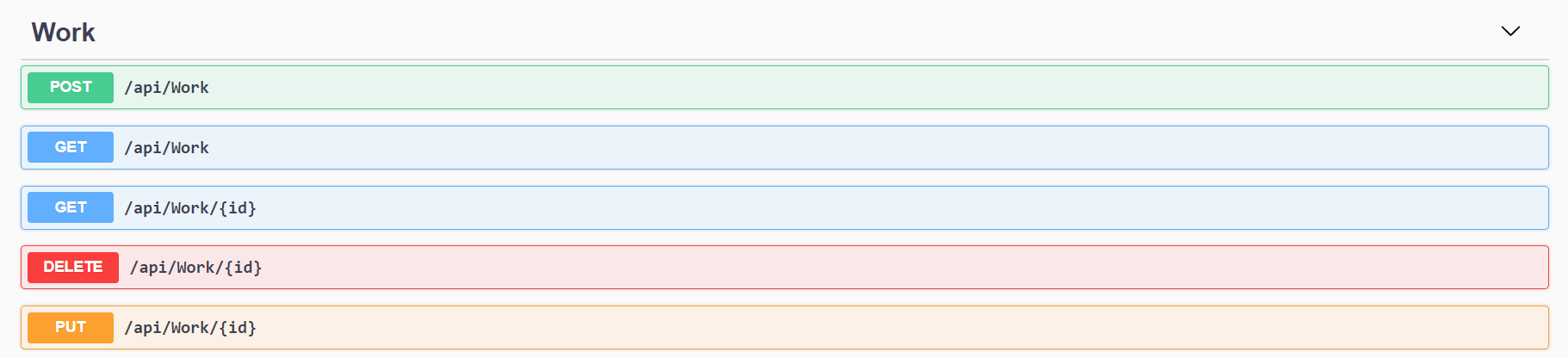


**3.2 pav. Visi atlikti testai**

Kiekvienam servisui parašyta nuo 1 iki 13 skirtingų testavimo scenarijų. Toks skaičius priklauso nuo serviso dydžio bei atliekamu funkcijų kiekio.

### Rankinis API testavimas

Rankinio API testavimo metu bus kviečiami visi metodai. Tam bus sudaryti scenarijai su tikėtinais rezultatais, gautais rezultatais juos įvykdžius ir tai pateikiama lentelėmis. Tie patys scenarijai tai pat bus panaudojami grafinės vartotojo sąsajos testavimui.



**3.3pav. Swagger grafinė vartotojo sąsaja**

|  |  |
| --- | --- |
| Metodo paskirtis | Sukurti naują darbą |
| Užlausos duomenys | {  "title": "work title",  "description": "long work description ",  "userId": 1,  "durationInMinutes": 20,  "activityPriority": 0,  "willBeParticipant": false  } |
| Sėkmingo veikimo scenarijus | Visi duomenys nurodyti |
| Sėkmingo veikimo rezultatas | Gražinamas 200 kodas, ir naujo darbo id |
| Nesėkmingo veikimo scenarijus | Trūksta dalies ar visų duomenų |
| Nesėkmingo veikimo rezultatas | Gražinamas 404 kodas |

**3.1 lentelė naujo darbo sukūrimas**

|  |  |
| --- | --- |
| Metodo paskirtis | Gauti visus darbus |
| Užlausos duomenys | {} |
| Sėkmingo veikimo scenarijus | Bent vienas darbas sukurtas |
| Sėkmingo veikimo rezultatas | Gražinamas 200 kodas ir visų darbų sąrašas |
| Nesėkmingo veikimo scenarijus | Nėra nei vieno darbo |
| Nesėkmingo veikimo rezultatas | Gražinamas 404 kodas |

**lentelė 3.2**

|  |  |
| --- | --- |
| Metodo paskirtis | Gauti darbą pagal jo id |
| Užlausos duomenys | {} |
| Sėkmingo veikimo scenarijus | Visi duomenys nurodyti |
| Sėkmingo veikimo rezultatas | Gražinamas 200 kodas, ir naujo darbo id |
| Nesėkmingo veikimo scenarijus | Trūksta dalies ar visų duomenų |
| Nesėkmingo veikimo rezultatas | Gražinamas 404 kodas |

**lentelė 3.3**

|  |  |
| --- | --- |
| Metodo paskirtis | Redaguoti darbą |
| Užlausos duomenys | {  "title": "updated title",  "description": "long updated description",  "userId": 1,  "durationInMinutes": 15,  "activityPriority": 1,  "willBeParticipant": true  } |
| Sėkmingo veikimo scenarijus | Visi duomenys nurodyti |
| Sėkmingo veikimo rezultatas | Gražinamas 204 kodas |
| Nesėkmingo veikimo scenarijus | Trūksta dalies ar visų duomenų |
| Nesėkmingo veikimo rezultatas | Gražinamas 404 kodas |

**lentelė 3.4**

|  |  |
| --- | --- |
| Metodo paskirtis | Šalinti darbą |
| Užlausos duomenys | {} |
| Sėkmingo veikimo scenarijus | Darbas buvo pašalintas |
| Sėkmingo veikimo rezultatas | Gražinamas 204 kodas |
| Nesėkmingo veikimo scenarijus | Darbo nebuvo rastas |
| Nesėkmingo veikimo rezultatas | Gražinamas 404 kodas |

**lentelė 3.5**

Visi 5 anksčiau pavaizduoti testai (3.1-3.5 lentelės) buvo atlikti naudojant Swagger grafinę sąsają (3.3 pav.). Iš viso atlikta 27 API metodų testavimo atvejai ir jie visi buvo sėkmingi.

## Statinė kodo analizė

Statinė kodo analizė buvo atlikta naudojant Resharper įrankį. Beje svarbu nepamiršti jog šis įrankis sugeba aptikti TypeScript, CSS, HTML bei C# kalbų, naudotų šiame projekte klaidas.

Pagrindinės šio įrankio aptinkamos klaidos yra šios:

* kintamųjų pavadinimo rašymo (išskyrus HTML ir CSS)
* formatavimo
* kodo stiliaus

Ir daug kitų, kurias ne mažiau svarbių aptikti, klaidų.

Jis įvykdė viso projekto kodo analizę bei joje aptiko 2 klaidas (3.3 pav). Tiesa jų nebuvo galima taisyti, kadangi tai yra Angular projektui paleisti reikalingas failas. Todėl ši statinės kodo analizės dalis laikoma sėkmingai įvykdyta.



**3.4 pav. Resharper aptiktos klaidos**

Peržiūrėjus informaciją apie klaidas buvo nuspręsta peržiūrėti ir perspėjimus. Iš viso aptikti 169 perspėjimai (3.3 pav.). Jie buvo net 56 skirtinguose failuose. Pamačius, jog jų skaičius yra gana didelis buvo nuspręsta patikrinti tuos failus ir įsitikinti ar galima jų kiekį sumažinti.



**3.5 pav. Perspėjimai dėl kodo kokybės, prieš peržiūrą**

Tada buvo atlikta kodo peržiūra tuose failuose kur aptikti perspėjimai, bei buvo ištaisyti visi, kurie neturėjo įtakos projekto veikimui ir liko tik tie, kurių nebuvo galima pakeisti pavyzdžiui migracijose, kurios yra generuojamos įvedus komandą ir jų redagavimas negalimas. Rezultatai (3.5 pav.) rodo gerokai sumažėjusį perspėjimų skaičių.



**3.6 pav. Perspėjimai dėl kodo kokybės, po peržiūros**

Galime padaryti išvadą, kad statinis kodo testavimas padėjo padaryti projekto kodą kokybiškesni, paprasčiau suprantamą ir padidino jo skaitomumą. Taip pat mažina klaidos tikimybę rašant didesnės apimties projektą.

## Vartotojo sąsajos testavimas

Vartotojo sąsajai testuoti pasirinkau rankinį testavimo būdą. Bus pateikta (3.6 lentelė) testavimo scenarijai bei numatomas rezultatas.

Kiekviena užduočiai atlikti yra būtina registracija ir prisijungimas, todėl atskiras šių panaudojimo atvejų testavimas nėra įtrauktas.

|  |  |  |
| --- | --- | --- |
| **Veiksmas** | **Aprašymas** | **Rezultatas** |
| Sukurti naują darbą: | * spaudžiamas „Works“ mygtukas pagrindiniame meniu * spaudžiama „Add new work“ mygtukas * užpildomi laukai * spaudžiamas „Create“ mygtukas | Sąraše atsirado naujas darbas |
| Redaguoti darbą: | * spaudžiamas „Works“ mygtukas pagrindiniame meniu * atsidariusiame sąraše paspausti mygtuką redaguoti * pakeisti kelis laukus * spausti „Edit“ | Darbas buvo atnaujintas |
| Pašalinti darbą: | * spaudžiamas „Works“ mygtukas pagrindiniame meniu * atsidariusiame sąraše paspausti raudoną „X“ mygtuką esantį sąraše | Darbas buvo ištrintas |
| Pakvieti kitus vartotojos atlikti darbą: | * spaudžiamas „Works“ mygtukas pagrindiniame meniu * spausti „Invite“ mygtuką esanti sąraše * įvesti norimo vartotojo vardą * spausti „Send invitation“ | Jei vartotojas yra atsiranda pranešimas apie išsiūtą pakvietimą, jei ne pranešimas apie klaidą |
| Perskaičiuoti darbų laiką | * Paimti viena darbą ir pertemti į kitą vietą | Darbų laikas perskaičiuotas teisingai |
| Keisti diena | * Pereiti diena pirmyn ir atgal, pažiūrėti kaip keičiasi dienos naudojant „Next“ ir „Previous“ mygtukus | Diena keičiasi teisingai, po „Previous“ paspaudimo mažėja, po „Next“ didėja |
| Atnaujinti tvarkaraštį | * Paspausti ant „Schedule“ pagrindiniame meniu * Paspausti mygtuką „Refresh“ | Tvarkaraštis atsinaujins |
| Peržiūrėti žinutes | * Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu | Žinučių sąrašas matomas |
| Filtruoti žinutes | * Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu * Įvesti įvarius filtrus, pažiūrėti kaip kinta žinučių sąrašas | Žinučių sąrašas filtruojamas |
| Pašalinti vieną žinutę | * Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu * paspausti raudoną „X“ mygtuką esantį žinučių sąraše | Pasirinkta žinutė ištrinta |
| Pakeisti darbo pradžios ir pabaigos nustatymus | * Paspausti ant vartotojo vardo pagrindiniame meniu * Pasirinkti laikus * Spausti „Update“ mygtuką | Laikai atnaujinti |
| Šalinti paskyra | * Paspausti ant vartotojo vardo pagrindiniame meniu * Paspausti mygtuką „Delete“ * Patvirtinti paskyros šalinimą * Pabandyti prisijungti su buvusios paskyros vardu ir slaptažodžiu | Paskyra ištrinta, prisijungti negalima |
| Atsijungti | * Spausti „Log out“ mygtuką pagrindiniame meniu | Bus atsijungta nuo paskyros, informacija taps nepasiekiama |

**3.6 lentelė vartotojo sąsajos testavimas**

Testuojant vartojo sąsają jokių problemų neaptikta. Vartotojas gali atlikti visus anksčiau (3.6 lentelėje) nurodytus veiksmus.

# Dokumentacija naudotojui

Dokumento dalis, skirta naudotojui, kur aprašomas visas naudotojui aktualus programinės (aparatūrinės) įrangos funkcionalumas (4 – 10 lapų).

Dokumentacija naudotojui – tai instrukcija kaip naudotis sistema. Dokumentacijoje turi būti aiškiai aprašyti naudojimosi sistema ypatumai, pradedant diegimu ir baigiant įprastinėmis funkcijomis. Rašydami dokumentaciją atsižvelkite į naudojamą terminologiją. Pavyzdžiui, jei sistemą instaliuos administratorius, o naudos paprasti vartotojai, pastarųjų stenkitės neapkrauti sudėtingesnėmis sąvokomis.

## Apibendrintas sistemos galimybių aprašymas.

Sistemos galimybės nuo reikalavimuose aprašyto funkcionalumo skiriasi tuo, kad ne visiems vartotojams būtina žinoti technines projekto detales. Pavyzdžiui, internetinio portalo vartotojui svarbu žinoti kokios naudingos funkcijos yra portale (pvz., paieška, naujienlaiškio prenumerata ir kt.), tačiau ne visos funkcijos įprastam vartotojui yra aktualios (pvz., reklamos skydelių palaikymas, SSL protokolas vartotojų autentifikacijai ir t.t.).

## Vartotojo vadovas

Vartotojo vadovas yra neformalus įvadas į sistemą, aprašantis jos „normalų“ vartojimą. Kitaip tariant, vartotojui 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., kiekvienam kūrimo etapui – 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. Apie LITNET. *Litnet.* [Tinkle] 2012 m. birželio 05 d. [Cituota: 2013 m. balandžio 04 d.] http://www.litnet.lt/index.php/apie-litnet.

2. *Transforming Ontology Representation from OWL to Relational Database.* Vyšniauskas, E. ir Nemuraitė, L. 3, 2006 m., Information Technology and Control, T. 35A, p. 333–343.

3. Masiulis, K. ir Krupavičius, A. *Valstybės tarnyba Lietuvoje: praeitis ir dabartis: kolektyvinė monografija.* Vilnius : Praction, 2007. p. 430.

4. *Spaudos draudimo klausimai.* Biržiška, V. 5, 1929 m., Kultūra, p. 249-235.

5. Valiulytė, Ieva. Išlaidos krašto apsaugai, jų pagrįstumas ir tikslingumas. *Sociumas.* [Tinkle] 2000 m. vasaris. [Cituota: 2001 m. gruodžio 12 d.] http://www.sociumas.lt.

6. Library, Dalhousie University. IEEE Citation style guide. [Tinkle] 2009 m. [Cituota: 2013 m. 04 11 d.] http://libraries.dal.ca/content/dam/dalhousie/pdf/library/Style\_Guides/IEEE\_Citation\_Style\_Guide.pdf.

7. *Hibridinis velomobilis.* Gradauskas, R. Kaunas : s.n., 2000. Transporto priemonės - 99. p. 81-83.

Darbe naudotos literatūros sąrašas (1 – 3 lapai). Sąrašas sudaromas vadovaujantis ISO 690 priimtu literatūros sąrašo ir citavimo stiliumi (1). Kaip sudarinėti literatūros sąrašą Word priemonėmis galite paskaityti <http://office.microsoft.com/en-us/word-help/create-a-bibliography-HA102809686.aspx> arba <http://office.microsoft.com/lt-lt/word-help/create-a-bibliography-HA102809686.aspx>.

Literatūros sąrašas turėtų apimti visus naudotus šaltinius. Literatūros šaltiniai pateikiami sunumeruoti citavimo tvarka. Darbo apraše turi būti pacituoti visi naudoti šaltiniai, pateikiant tekste nuorodas. Daugiau informacijos apie bendras citavimo taisykles galite rasti <https://biblioteka.ktu.edu/mokymai/#mokymosi-medziaga> „Kaip cituoti šaltinius ir parengti literatūros sąrašą. ISO 690:2010 standartas (skirta technologijos mokslams)“.

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

**4.1** **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 po pastraipos – 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. |

1. Pateikiant statistiką ar skaičius reikia nurodyti šaltinį, kurį rekomenduojama įtraukti į literatūros sąrašą [↑](#footnote-ref-2)
2. Šiuos informacinius šaltinius taip pat rekomenduojama įtraukti į literatūros sąrašą [↑](#footnote-ref-3)