KAUNAS, 2019



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INformatikos fakultetas

Matas kulkovas

Realaus laiko stalo futbolo žaidimo sekimas

Baigiamasis bakalauro projektas

Vadovas

doc. Karolis Ryselis

KAUNAS, 2019

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INformatikos fakultetas

DARBO PAVADINIMAS

Baigiamasis bakalauro projektas

Programų sistemos (kodas 612I30002)

Vadovas

doc. dr. V. Pavardaitis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(data, parašas)

Recenzentas

doc. dr. A. Kirvaitis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

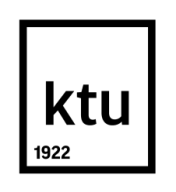
(data, parašas)

Projektą atliko

V. Pavardenis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(data, parašas)



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

|  |
| --- |
| Informatikos fakultetas |
|  |
|  |
| (Studento vardas, pavardė) |
| Programų sistemos (kodas 612I30002) |
|  |

„Baigiamojo projekto pavadinimas“

**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 20 |  | m. |  |  |  | d. |
|  |  |  | Kaunas |  |  |  |

Patvirtinu, kad mano, **Vardenio Pavardenio**,baigiamasis projektas tema „..................................................................................................“ 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)* |

Techninė užduotis

Užsakovo suformuluota techninė užduotis su abstrakčiais funkciniais ir nefunkciniais reikalavimais (1 lapas).

Pavardenis, Vardenis. Baigiamojo projekto pavadinimas. *Bakalauro / Magistro* *(pasirinkite)* baigiamasis projektas / vadovas doc. dr. Vardenis Pavardenis; Kauno technologijos universitetas, ................................ fakultetas.

Mokslo kryptis ir sritis:

Reikšminiai žodžiai: *vienas, du, trys, keturi, penki, šeši.*

Kaunas, 20XX. XX p.

**SANTRAUKA**

*(santraukos tekstas)*

Surname, Name. *Title of the Project (Each Word of the Title is Written in Capital Letters)*: *Bachelor's / Master‘s* *(choose one)* thesis in .......................................... / supervisor assoc. prof. Vardenis Pavardenis. The Faculty of ............................................., Kaunas University of Technology.

Research area and field:

Keywords:

Kaunas, 20XX. XX p.

**SUMMARY**

(*summary text*)

Turinys

[Lentelių sąrašas 9](#_Toc474841534)

[Paveikslų sąrašas 10](#_Toc474841535)

[Terminų ir santrumpų žodynas 11](#_Toc474841536)

[Įvadas 12](#_Toc474841537)

[1. Analizė 13](#_Toc474841538)

[1.1. Techninis pasiūlymas 13](#_Toc474841539)

[1.1.1. Sistemos apibrėžimas 13](#_Toc474841540)

[1.1.2. Bendras veiklos tikslas 13](#_Toc474841541)

[1.1.3. Sistemos pagrįstumas 13](#_Toc474841542)

[1.1.4. Konkurencija rinkoje 13](#_Toc474841543)

[1.1.5. Prototipai ir pagalbinė informacija 14](#_Toc474841544)

[1.1.6. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti 14](#_Toc474841545)

[1.2. Galimybių analizė 14](#_Toc474841546)

[1.2.1. Techninės galimybės 14](#_Toc474841547)

[1.2.2. Vartotojų pasiruošimo analizė 14](#_Toc474841548)

[2. Projektas 15](#_Toc474841549)

[2.1. Reikalavimų specifikacija 15](#_Toc474841550)

[2.1.1. Komercinė specifikacija 15](#_Toc474841551)

[2.1.2. Sistemos funkcijos 15](#_Toc474841552)

[2.1.3. Apribojimai 15](#_Toc474841553)

[2.1.4. Duomenų ir rezultatų specifikacija 15](#_Toc474841554)

[2.1.5. Vartotojo sąsajos specifikacija 15](#_Toc474841555)

[2.1.6. Realizacijai keliami reikalavimai. 15](#_Toc474841556)

[2.1.7. Techninė specifikacija. 16](#_Toc474841557)

[2.2. Projektavimo metodai 16](#_Toc474841558)

[2.2.1. Projektavimo valdymas ir eiga 16](#_Toc474841559)

[2.2.2. Projektavimo technologija 16](#_Toc474841560)

[2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos 16](#_Toc474841561)

[2.3. Sistemos projektas 16](#_Toc474841562)

[2.3.1. Statinis sistemos vaizdas 16](#_Toc474841563)

[2.3.2. Dinaminis sistemos vaizdas 17](#_Toc474841564)

[2.3.3. Duomenų kontrolė 17](#_Toc474841565)

[3. Testavimas 18](#_Toc474841566)

[3.1. Testavimo planas 18](#_Toc474841567)

[3.2. Komponentų testavimas 18](#_Toc474841568)

[3.3. Testavimo kriterijai 18](#_Toc474841569)

[3.4. Vartotojo sąsajos testavimas 18](#_Toc474841570)

[4. Dokumentacija naudotojui 19](#_Toc474841571)

[4.1. Apibendrintas sistemos galimybių aprašymas 19](#_Toc474841572)

[4.2. Vartotojo vadovas 19](#_Toc474841573)

[4.3. Diegimo vadovas 19](#_Toc474841574)

[4.4. Administravimo vadovas 19](#_Toc474841575)

[5. Rezultatų apibendrinimas ir išvados 20](#_Toc474841576)

[6. Literatūra 21](#_Toc474841577)

[7. Priedai 22](#_Toc474841578)

[7.1. priedas. Testavimo rezultatai 22](#_Toc474841579)

Automatiškai generuojamas darbo turinys (į turinį įtraukiami numeruojami skyriai, lentelių ir paveikslų sąrašai, terminų ir santrumpų žodynas bei įvadas) (1 – 2 lapai). Skyrių, neįtraukiamų į turinį, pavadinimai turi būti rašomi *TOC Heading* stiliumi.

# Lentelių sąrašas

[1.1 lentelė Sistemų palyginimo lentelės pavyzdys 8](#_Toc354142358)

Automatiškai generuojamas darbe paminėtų lentelių sąrašas (lentelės numeris, pavadinimas ir puslapis) (1 – 2 lapai).

# Paveikslų sąrašas

[2.1 pav. Detalios projekto klasių diagramos pavyzdys (fragmentas) 8](#_Toc354142359)

Automatiškai generuojamas darbe paminėtų paveikslų sąrašas (paveikslo numeris, pavadinimas ir puslapis) (1 – 2 lapai).

# Terminų ir santrumpų žodynas

Darbe naudojami specifiniai terminai ir santrumpos bei jų paaiškinimas (1 – 2 lapai).

# 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). Prieš įvado poskyrius nurodoma, kokiai studijų programai ir specializacijai priklauso darbas.

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 – 6 uždaviniai). Išvados dokumento pabaigoje formuluojamos uždavinių pagrindu.

Darbo struktūra

Aptariama dokumento struktūra. Nurodoma kiek ir kokių skyrių dokumente yra ir kokia informacija juose pateikiama.

# Analizė

Su darbo problematika susijusios informacijos analizė (8 – 12 lapai). Skyriaus pavadinimas ir struktūra priklauso nuo baigiamojo darbo specializacijos ir pačios temos specifikos.

## Techninis pasiūlymas

### Sistemos apibrėžimas

Stalo futbolo realaus laiko sekimo sistemą sudaro trys dalys – vaizdo sekimo posistemė, serverio posistemė bei serverio klientas. Vaizdo sekimo posistemė, su kameros pagalba, seka stalo futbolo stalą, įrašo bei praneša serveriui apie kamuoliuko poziciją bei įvarčius. Serveris saugo žaidimų bei žaidėjų duomenis.

### Bendras veiklos tikslas

Sukurti sistemą, kuri pateiktų įdomią statistiką apie žaidimų eigas. Sistema taip pat automatizuotų rezultatų skaičiavimą, turnyrų progresiją. Sukurta sistema bus integruojama su įmonės futbolo stalu.

### Sistemos pagrįstumas

Iki šiol įmonėje stalo futbolo žaidimų rezultatai būdavo skaičiuojami mechaniškai, skaitliuko pagalba. Jei žaidimas turnyrinis – jo rezultatas įrašomas į skaičiuoklę, jei ne – nėra išsaugomas niekur.

### Konkurencija rinkoje

Rasti produktų, kurie vykdytų vaizdo analizę ir atliktų žaidimų organizavimo ir rezaltatų saugojimo funkciją – sudėtinga. Egzistuoja nemažai aplikacijų, kurios skirtos rezultatų saugojimui ir pateikimui, turnyrų organizavimui, tačiau jos nerenka duomenų apie žaidimą, rezultatus reikia suvedinėti pačiam. Yra parašyta nemažai straipsnių apie prototipines aplikacijas skirtas stalo futbolo vaizdo sekimui, tačiau jose retai yra dalinamasi programiniu kodu. Taip pat egzsituoja bendro naudojimo pobūdžio video analizės įrankis Amazon „Rekognition“. Tačiau dėl 0.10$ įkainio už analizuojamo vaizdo minutę įkainio šis pasirinkimas buvo greitai atmestas.

Rezultatų sekimui yra nemažai sistemų, tiek mokamų tiek nemokamų. Papraščiausią funkcionalumą išpildanti aplikacija buvo pasirinkta „Kicktrack“. Tai web sąsajoje veikiantis produktas, leidžiantis susikurti savo įmonės grupę, suvedinėti rezultatus ir juos sekti. Tačiau ne visos siūlo turnyrų kūrimą.

Paieškojus GitHub atvirojo kodo sistemoje, buvo aptiktas naudotojo „mtszkw“ projektas „foosball“ (*<https://www.github.com/mtszkw/foosball>***).** Šio analizės įrankio teikiami duomenys yra pakankamai išsamūs planuojamam projektui – aptinkama kamuoliuko pozicija, figurėlių pozicija ant žaidėjų juostos. Tačiau įrankis analizuoja tik įrašytą video medžiagą. Jei modifikavus jo kodą, kad analizuotų vaizdo srautą gyvai, jis veikia mažiau nei 10 kadrų per sekundę ant dedikuoto kompiuterio. Taip pat programinis kodas parašytas su C++ 17. Nors tai yra vaizdo analizės kalbų standartas, užsakovas plėtojimo tikslais pageidavo programinio kodo Python 3 programavimo kalba.

1.1 lentelė. Sistemų palyginimo lentelės pavyzdys

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lyginimo kriterijai** | **Kicktrack** | **Kickertool** | **github.com/mtszkw/foosball** | **Kuriama sistema** |
| Žaidėjų išsaugojimas | Realizuota | Realizuota | Nerealizuota | Realizuota |
| Pavienės rungtynės | Realizuota | Nerealizuota | Nerealizuota | Realizuota |
| Turnyrų kūrimas | Nerealizuota | Realizuota | Nerealizuota | Realizuota |
| Figurėlių aptikimas vaizde | Nerealizuota | Nerealizuota | Realizuota, bet tik vaizdo įrašuose | Realizuota su vaizdo įrašais ir vaizdo transliacijoje apie 30 FPS |
| Kamuoliuko pozicijos aptikimas vaizde | Nerealizuota | Nerealizuota | Realizuota, bet tik vaizdo įrašuose | Realizuota su vaizdo įrašais ir vaizdo transliacijoje apie 30 FPS |
| Licencijos kaina | Free – nemokama  Stadium – 7$ / mėn  Arena – 29$ / mėn | Nemokama | Nemokama | Nemokama |
| Vaizdo analizės programavimo kalba | Nėra | Nėra | C++ 17 su OpenCV 3.4.1 biblioteka | Python 3.7 su OpenCV 3.4.4 biblioteka |
| Vaizdo iškreipimo panaikinimas | Nėra | Nėra | Yra | Gali būti įtraukta sekančioje versijoje |

### Prototipai ir pagalbinė informacija

Serveris bei naudotojo sąsaja projektuoti nenaudojant prototipų, tačiau buvo naudojamos kelios bibliotekos. Serverio dalyje buvo naudojama Entity Framework Core 2.2.0 biblioteka operacijoms su duomenų baze. EF Core 2.2.0 padeda išvengti tekstinio užklausų rašymo, užklausas paverčia į operacijas su objektais, taip leidžiant neprisirišti prie konkrečios duomenų bazės bei vykdo visas duomenų bazės užklausas naudodama repozitorijos projektavimo šabloną (angl. Reposity Design Pattern).

Vaizdo analizės posistemėje buvo remtasi kuriant Indianos Universiteto studentų darbo ataskaita [1]. Programos kodas nebuvo pateiktas, tačiau buvo apytiksliai aprašyti vaizdo analizės metodai, kurie pasirodė tinkami ir kurių dauguma buvo remtasi programuojant žaidėjų sekimą. Kamuoliuko sekimui nuo prototipo buvo pakeistas paskutinis algoritmo žingsnis – vietoj šablonų lyginimo (angl. Template matching) buvo pasirinktas kontūro sekimo metodas (angl. Contour following), kuris eksperimentiškai pateikė didesnės greitaveikos ir panašaus tikslumo rezultatus. Taip pat vaizdo analizei buvo naudoja OpenCV 3.4.4. Ji suteikia metodus palengvinančius darbą su vaizdo srauto skaitymu, keletą algoritmų, kurie naudojami analizuoti paveikslėliams.

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 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“.

### Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti

Skyrelyje įvertinkite sistemos apimtį (matai gali būti įvairūs: darbo valandos, kodo eilučių skaičius, modulių kiekis, teikiamų paslaugų skaičius ir pan.), nurodykite techninių ir žmoniškųjų išteklių poreikį, kad sistema būtų sukurta.

Pvz.: „Maisto perdavimo protokolui sukurti reikalingi 3 metai. Tiek laiko reikės aprašyti sistemos standartui ir realizuoti prototipams. Papildomai reikėtų 10 metų, norint išpopuliarinti ir išvystyti pasaulinę protokolo infrastruktūrą. Pradinėse projekto fazėse reikės bent 10 žmonių (IT specialistų) komandos, tačiau projektui plečiantis turės didėti ir personalas“.

## Galimybių analizė

### Techninės galimybės

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

Jei tai produktas, kuris bus naudojamas įmonėje, tai įmonės vartotojų analizė. Jei ne – bent jau abstrakti tikslinės auditorijos (programos ar produkto vartotojų segmento) analizė. Šiame punkte reiktų aprašyti ar žmonės sugebės naudotis ta įranga, kokio išsilavinimo reikia norint ja naudotis.

Pvz.: „Maisto perdavimo protokolas bus toks paprastas, kad juo galės naudotis tiek penkerių metų vaikas, tiek jo močiutė“.

# Projektas

Aprašoma sistemos projektavimo stadija, pateikiama detali specifikacija (8 – 12 lapai). 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 – atvirojo kodo. Jo užsakovas – įmonė MB „Tvirtas baitas“. Jos naudotojai – įmonės ofiso stalo futbolo žaidėjai.

Projekto biudžetas nėra numatytas. Numatyta pristatymo data – gegužės 8d. 2018m. Iki šios datos turi būti atlikti visi pirminiai sistemos programavimo, testavimo ir diegimo darbai.

Sistema kuriama Windows aplinkoje, naudojant Visual Studio 2017 aplinką.

### Sistemos funkcijos

Čia būtų išvardinamos visos sistemos funkcijos (aprašomi **funkciniai reikalavimai**), pavaizduotos UML diagramomis. Po diagramomis turi būti papildomi kiekvienos funkcijos detalūs žodiniai aprašymai. Jei norite, galite aprašyti kiekvieną panaudos atvejį išsamiai (pagal Volere šabloną): atvejo pavadinimas, aprašymas, išankstinės sąlygos, galutinės sąlygos (pasibaigus panaudos atvejui), galimas klaidingas veikimas.

### Apribojimai

Šiame punkte išvardinami (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.

### Vartotojo sąsajos specifikacija

Vartotojo sąsajos specifikacijoje turi būti nurodomi reikalavimai vartotojo sąsajos vaizdai. Čia nereikia ir negalima dėti jau egzistuojančios programos screenshot‘ų! Šiame etape tik nusakoma, kokia turi būti vartotojo sąsaja (rekomenduojame Balsamiq Mockups, Axure RP ir panašius įrankius), tačiau galutinis sąsajos vaizdas nurodomas tik vėlesniuose skyriuose. Jei pradinėje kūrimo fazėje buvo naudojami vartotojo sąsajos eskizai ar prototipai, juos reikia dėti būtent į šį skyrelį.

### 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.

Pvz.: „Maisto perdavimo protokolas privalo būti saugus (neautentifikuoti kaimynai negali sužinoti kokį maistą siunčiasi vartotojas). Maistas, suskaidytas paketais, turi pasiekti vartotoją nepagedęs. Maisto perdavimo protokole naudojamas linijas turi būti nesunku valyti“.

### 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 projektavimo modelį (ar modelius) naudojote kurdami savo sistemą. Tai gali būti krioklio, iteracinis ar kitas modelis. Galite nurodyti kaip suskirstėte projektavimo darbus ir kokiu eiliškumu juos atlikote.

### Projektavimo technologija

Šiame punkte nurodykite, kokią naudojote projektavimo technologiją, standartus ir programinius įrankius projekto kūrimui. Aprašykite kokia ar kokiomis notacijomis (formaliais tekstiniais ir grafiniais žymėjimo / aprašymo standartais) naudojotės kurdami sistemos projektą.

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

Šiame punkte, tiesiog, aprašykite, kokią programinę įrangą naudojote kurdami savo baigiamąjį darbą. Tiesa, rašyti kokią programinę įrangą naudojote šiai ataskaitai sukurti – nebūtina.

## 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 diagramomis:

* 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 atvaizduoti esmines logines sistemos dalis.
* Paketų (*UML package diagram*) – labai naudinga tuomet, jei sistemos apimtis yra labai didelė.
* 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 UML klasių diagrama.

### Dinaminis sistemos vaizdas

Šiame punkte pagal poreikį reiktų pavaizduoti sistemos veiksmus sekų, būsenų ar (ir) veiklos diagramomis. Galite pasirinkti vieną iš jų, galite naudoti ir kelias (priklausomai nuo sistemos specifikos).

### Duomenų kontrolė

Čia reikėtų detaliai aprašyti įvedamų duomenų kontroliavimo principus. Pavyzdžiui, tai gali būti serverio pusės duomenų kontrolė (kai duomenys tikrinami išsiuntus juos į serverį), gali būti ir vartotojo pusės kontrolė (kai duomenys tikrinami lokaliai, pvz. JavaScript pagalba). Tai aprašydami nepamirškite nurodyti ir kokių laukų duomenys tikrinami, ir kokia tiksliai kontrolė atliekama (pvz. pavardės laukelyje tikrinama ar nėra skaičių).

# Testavimas

Aprašoma su sukurtos įrangos testavimu susijusi informacija (8 – 12 lapai). Skyriaus struktūra ir pavadinimas priklauso nuo baigiamojo darbo specializacijos ir pačios temos specifikos.

Nurodomas įrangos testavimo planas, testavimo duomenų rinkiniai ir gauti rezultatai. Nurodoma sistemos specifikacija ir sąlygos, prie kurių buvo atliekamas testavimas.

## Testavimo planas

Testavimo planas – tai jūsų pasirinkta testų atlikimo tvarka. Galimas testavimo planas: komponentų testavimas, po kurio seka integracinis testavimas, o vėliau būna sąsajos testavimas. Testavimo planą patogu pateikti paprasčiausia schema.

## Komponentų testavimas

Šiame punkte reiktų aprašyti kokiais metodais testavote komponentus. Dažniausiai pasitaikantis metodas – „rankinis“, t.y. kai sąsaja testuojama vartotojui (testuotojui) bandant atsitiktinai ar pagal scenarijų spaudyti mygtukus, įvedinėti tekstą į laukus ir kt. Kur kas geresnis variantas tuomet, kai testuojama automatiškai – pavyzdžiui, sukuriama programa ar testavimo tvarkyklė, kuris spaudymo ar įvedimo veiksmus atlieka be vartotojo įsikišimo. Panaudotas automatinis testavimas, dažniausiai, papildomai (teigiamai) įvertinamas baigiamojo darbo gynimo metu.

## Testavimo kriterijai

Šiame punkte aprašykite kriterijus, kurie jums buvo svarbūs testavimo metu. Tai gali būti ne tik informacijos ar skaičiavimų korektiškumas, bet ir kodo pertekliškumo analizė, informacijos perdavimo laikas, sistemos atitikimas funkciniams ir nefunkciniams reikalavimams.

## Vartotojo sąsajos testavimas

Šiame punkte reiktų aprašyti kokiais metodais testavote vartotojo sąsają. Dažniausiai pasitaikantis metodas – „rankinis“, t.y. kai sąsaja testuojama vartotojui (testuotojui) bandant atsitiktinai ar pagal scenarijų spaudyti mygtukus, įvedinėti tekstą į laukus ir kt. Kur kas geresnis variantas tuomet, kai testuojama automatiškai – pavyzdžiui, sukuriama programa ar testavimo tvarkyklė, kuris spaudymo ar įvedimo veiksmus atlieka be vartotojo įsikišimo. Panaudotas automatinis testavimas, dažniausiai, papildomai (teigiamai) įvertinamas baigiamojo darbo gynimo metu.

# Dokumentacija naudotojui

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

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.).

# Rezultatų apibendrinimas ir išvados

Darbo metu gautų ir dokumentacijoje užfiksuotų rezultatų apibendrinimas bei išvadų formulavimas (1 – 2 lapai).

Išvados formuluojamos darbo uždavinių pagrindu (kiekvienam uždaviniui po išvadą). Išvados nenurodo, kas buvo padaryta darbe, bet pabrėžia atrastus dėsningumus, pastebėtas technologijų ar rinkos spragas, esminius įrangos privalumus ir pan. Išvados gali būti formuluojamos tik darbo metu sukurtos įrangos, technologijos, metodo ar susistemintos informacijos pagrindu (pvz. negalima cituoti šaltinių, vadovautis kitų autorių atrastais dėsningumais).

Bene svarbiausia viso darbo dalis – išvados. Stenkitės, kad išvadų būtų ne viena – gerai, jei darbas turi 5-7 išvadas. Išvadose reikia mažiau koncentruotis į tai, kas padaryta, ir daugiau dėmesio skirti tam, kas sužinota (pvz. įsisavintos naujos technologijos).

Pvz. „Darbo metu buvo įsisavintos kompiuterinių tinklų technologijos (protokolų pagrindai, architektūros principai), kurios padėjo sukurti maisto perdavimo protokolą (Food Transfer Protocol). Analizės metu buvo išsiaiškinta, jog pasaulyje šiuo metu neegzistuoja konkurentų šiai sistemai, todėl galima daryti išvadą, jog projektas turi daug galimybių susilaukti finansinės sėkmės ir pripažinimo“.

Dar vienas galimų išvadų pavyzdys (šablonas):

• Buvo sukurta tokia ir tokia sistema, gebanti atlikti tokius ir tokius veiksmus...

• Buvo išanalizuotos analogiškos (konkrečiai nurodant kokios) sistemos, kurios pasižymėjo tokiais ir tokiais privalumais (apibendrintai), tačiau dėl tokių ar anokių trūkumų buvo nuspręsta kurti atskirą (savo) sistemą...

• Kūrimo metu buvo susipažinta su tokia ir tokia kūrimo technologija, panaudoti va tokie ir anokie projektavimo metodai...

• (nebūtina, bet galima pridėti) Kūrimo metu buvo susidurta su tokiomis ar anokiomis problemomis, kurios buvo sprendžiamos taip arba anaip...

• Sukurta sistema buvo įdiegta įmonėje ir šiuo metu yra naudojama ARBA sistema yra viešai prieinama vartotojams ir dėl tokių ar anokių technologijų bei savybių turi išliekamąją vertę. ..

• (nebūtina, bet galima pridėti) Ateityje planuojama sistemą tobulinti suteikiant jai tokias ir tokias savybes ar funkcijas...

# Literatūra

Darbe naudotos literatūros sąrašas (1 – 3 lapai). Sąrašas sudaromas vadovaujantis IEEE 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. Literatūros šaltinių sąrašo pavyzdys pateiktas žemiau:

|  |  |
| --- | --- |
| [1] | D. U. Library, „IEEE Citation style guide,“ 2009. [Tinkle]. Available: http://libraries.dal.ca/content/dam/dalhousie/pdf/library/Style\_Guides/IEEE\_Citation\_Style\_Guide.pdf. [Kreiptasi 11 04 2013]. |
| [2] | K. Masiulis ir A. Krupavičius, Valstybės tarnyba Lietuvoje: praeitis ir dabartis: kolektyvinė monografija, Vilnius: Praction, 2007, p. 430. |
| [3] | V. Biržiška, „Spaudos draudimo klausimai,“ *Kultūra,* nr. 5, pp. 249-235, 1929. |
| [4] | E. Vyšniauskas ir L. Nemuraitė, „Transforming Ontology Representation from OWL to Relational Database,“ *Information Technology and Control,* t. 35A, nr. 3, p. 333–343, 2006. |
| [5] | „Apie LITNET,“ 05 birželio 2012. [Tinkle]. Available: http://www.litnet.lt/index.php/apie-litnet. [Kreiptasi 04 balandžio 2013]. |
| [6] | I. Valiulytė, „Išlaidos krašto apsaugai, jų pagrįstumas ir tikslingumas,“ vasaris 2000. [Tinkle]. Available: http://www.sociumas.lt. [Kreiptasi 12 gruodžio 2001]. |
| [7] | R. Gradauskas, „Hibridinis velomobilis,“ įtraukta *Transporto priemonės - 99*, Kaunas, 2000. |

Literatūros aprašo pavyzdžiai. Knyga aprašoma: [2]; Žurnalo straipsniai: [3], [4]; Interneto svetainė: [5]; Straipsnis iš interneto: [6]; Konferencijos medžiaga: [7].

# Priedai

Papildoma informacija ir dokumentai, neįeinanti į pagrindinį dokumentą. Yra, jeigu reikalingi.

## priedas. Testavimo rezultatai