****

**Kauno technologijos universitetas**

Informatikos fakultetas

**T120M010 Programinės įrangos įdiegimo valdymas**

Testavimo planas

|  |  |
| --- | --- |
|  |  |
| **IFM8-2 gr.stud.**  **Marius Ambrazevičius**  Studentas/studentė | (parašas) (data) |
|  |  |
| **Lekt. Virginija Limanauskienė**  Dėstytojas/ dėstytoja | (parašas) (data) |
|  |  |

**Kaunas, 2019**

Turinys

[Lentelių sąrašas 3](#_Toc20868279)

[Paveikslų sąrašas 4](#_Toc20868280)

[Santrumpų ir terminų sąrašas 5](#_Toc20868281)

[1. Įvadas 6](#_Toc20868282)

[2. Testavimo planas 8](#_Toc20868283)

[3. Testavimo procedūra 12](#_Toc20868284)

[Išvados 14](#_Toc20868285)

[Literatūros sąrašas 15](#_Toc20868286)

Lentelių sąrašas

[**1 lentelė.** Nuorodos 6](#_Toc20868274)

[**2 lentelė.** Testavimo tvarkaraštis 11](#_Toc20868275)

Paveikslų sąrašas

[1 pav. Pagrindinis vartotojo sąsajos meniu 8](#_Toc20868207)

[2 pav. Analizių sąrašas 9](#_Toc20868208)

[3 pav. Analizės kūrimo langas 9](#_Toc20868209)

[4 pav. Segmentacijos apmokymo duomenų kūrimo langas 9](#_Toc20868210)

Santrumpų ir terminų sąrašas

**Santrumpos:**

SEG – neuroninis tinklas atsakingas už segmentacijos veiksmą;

KLAS – neuroninis tinklas atsakingas už klasifikacijos veiksmą;

VPGSA – vartotojo patirties grafinės sąsajos analizės sistema;

NEUR – neuroninio tinklo posistemė;

**Terminai:**

**Komitas - (angl. commit)** terminas neturintisgero lietuviško atitikmens. Tai „Git“ sistemos veiksmas reiškiantis kodo įkėlimą į internete patalpintą repozitoriją.

**Repozitorija –** talpinimo vieta. Šiuo atveju – kodo talpinimo vieta.

**Saitynas** – pasaulinis tinklas, internetas, žiniatinklis.

# Įvadas

Šio testavimo plano dokumento paskirtis yra aprašyti kaip bus testuojama kuriama sistema „Vartotojo patirties ir grafinės sąsajos analizės sistema“. Dokumentas skirtas naudotis projektą realizuojantiems asmenims – užsakovams, projekto vadovui ir projekto programuotojui. Šis dokumentas rašomas nebaigus realizuoti sistemos todėl galutinis testavimo rezultatas gali skirtis nuo plano aprašyto dokumente.

## Testavimo tikslai ir objektai

Sistemoje yra 2 pagrindiniai testavimo objektai: saityno posistemė – sistemos dalis atsakinga už vartotojo sąsaja, bei komunikavimu su neuroninio tinklo posisteme ir neuroninių tinklų posistemė.

Testavimo tikslai:

* Užtikrinti saityno posistemės sklandų veikimą.
* Užtikrinti saityno posistemės stabilumą įvedant naujas ypatybes.
* Užtikrinti neuroninio tinklo posistemės tikslumą.
* Užtikrinti sistemos atitikimą reikalavimams.

## Testavimo apimtis ir tipai

* Vienetų testavimo metu bus testuojamas saityno posistemės komponentų funkcionalumas.
* Integracinio testavimo metu bus testuojamas saityno posistemės susijusių komponentų integracijos veikimas.
* NEUR testavimas – testavimas, kuris bus atliekamas tikrinti SEG ir KLAS tikslumą.
* Priėmimo testavimo metu bus testuojama kaip sistema atitinka aprašytus funkcinius reikalavimus.
* Aukšto lygio testavimo metu bus testuojama kaip tiksliai veikia sistema.

## Pagrindiniai apribojimai

Pagrindinis apribojimas yra techninis - neuroninių tinklų posistemės testavimui, kadangi neuroninių tinklų operacijos yra brangios ir reikalauja daug kompiuterinių išteklių, tai apribojimas pasirodo kaip testavimo įrangos išteklių trūkumas (efektyviam ištestavimui).

## Nuorodos

**1 lentelė.** Nuorodos

| **Dokumentas** | Data | Dokumento autorius | Nuoroda |
| --- | --- | --- | --- |
| Projekto planas | 2019-09-14 | Studentas | <https://github.com/Duderis/masters-thesis-project> |
| Reikalavimų specifikacija | 2019-05-25 | Studentas | <https://github.com/Duderis/masters-thesis-project> |
| Architektūros specifikacija | 2019-05-25 | Studentas | <https://github.com/Duderis/masters-thesis-project> |

## Dokumento struktūra

Šį dokumentą sudaro trys pagrindinės dalys: Įvadas, Testavimo planas ir Testavimo procedūros. Kiekviena pagrindinė dalis yra toliau detalizuojama savo skyriuose. Įvado dalyje aprašyta bendra dokumento paskirtis, testavimo tikslai, objektai, testavimo apimtis, testavimo apribojimai ir nuorodos į susijusius dokumentus. Testavimo plano dalyje aprašoma testuojama programinė įranga, testavimo strategija ir detalizuojami testavimo metodai. Testavimo procedūrų dalyje pateikiamas panašus detalizavimas kaip ir testavimo plano dalyje, tačiau iš testavimo procedūrų perspektyvos.

# Testavimo planas

Testavimo planas naudojamas aprašyti testuojama programinę įrangą, testavimo strategiją, bei planuojamus naudoti testavimo metodus. Testavimo metodai yra pateikiami abstrakčiai, neliečiant susijusių procedūrų – tai atliekama testavimo procedūrų dalyje (3 skyriuje).

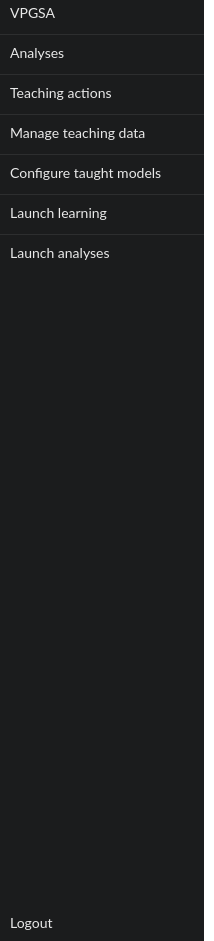
## Testuojama programų sistema

Testuojama programinė įranga yra skirta analizuoti grafinės sąsajos nuotraukas – programinė įrangą analizuoja nuotrauką pasitelkiant neuroninių tinklų pagalba. Nuotrauka pirmiausiai analizuojama neuroninio tinklo atsakingo už segmentaciją (SEG), po kurios, segmentacijos metu gauta nuotrauka analizuojama neuroniniu tinklu atsakingu už klasifikacija (KLAS).

### Sąsajos

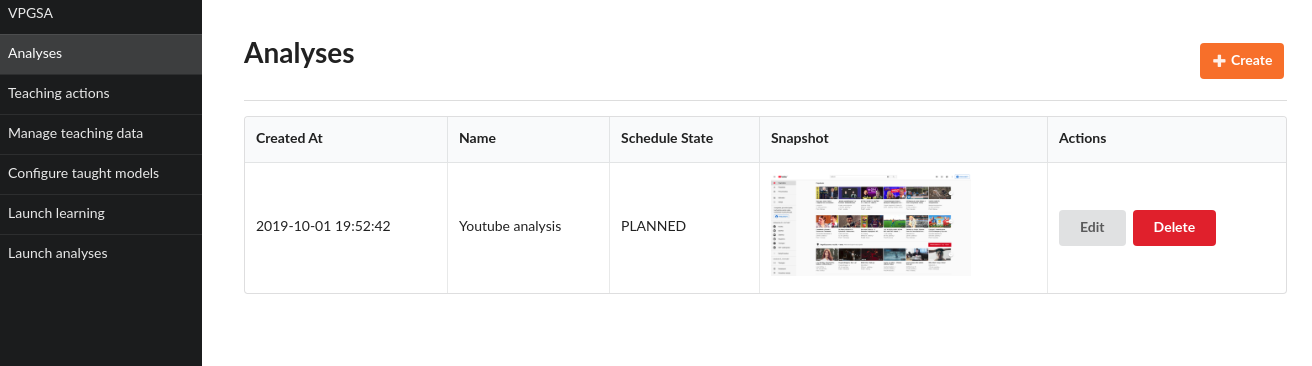
Pateikiama dalis vartotojo sąsajos langų, kurie dar nėra galutiniai ir tikriausiai keisis. Šiais langais bus naudojamasi atliekant priėmimo bei aukšto lygio testavimus.

Svarbiausia sistemos dalis yra navigacijos meniu, kuriame laikoma sąsaja į pagrindinį sistemos funkcionalumą (pavaizduota 1 pav.).



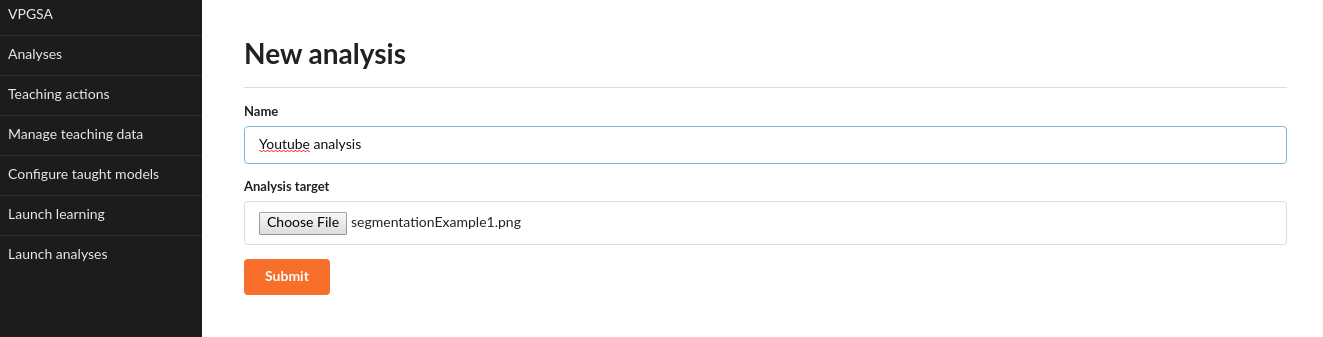
1 pav. Pagrindinis vartotojo sąsajos meniu

Pagrindinis sistemos funkcionalumas bus prieinamas per „Analyses“ meniu punktą. Pasirinkus jį, vartotojas supažindinamas su sąrašu visų analizių (pavaizduota 2 pav.):



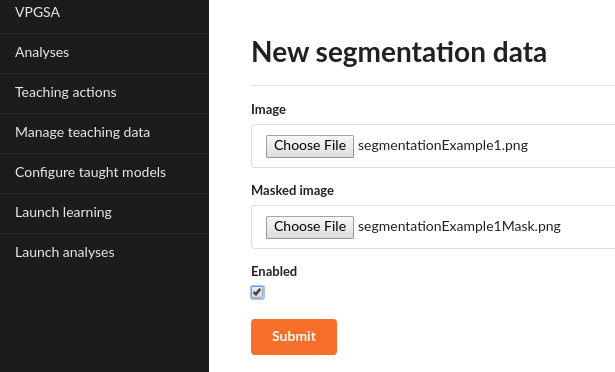
2 pav. Analizių sąrašas

Analizių kūrimo langas kaip ir kiti langai, šuo metu yra gana primityvus turi tik analizuojamos nuotraukos įkėlimo lauką, pavaizduota 3 pav.:



3 pav. Analizės kūrimo langas

Egzistuoja ir mokymosi duomenų valdymo langai, sąrašo langas analogiškas analizių sąrašui, todėl 4 pav. atvaizduojamas tik SEG kūrimas:



4 pav. Segmentacijos apmokymo duomenų kūrimo langas

## Testavimo strategija

Tolesniuose skyreliuose detalizuojama strategija kiekvienam planuojamam testavimo metodui.

### Vienetų testavimas

Vienetų testavimo metu bus skiriamas didelis dėmesys saityno posistemei. Šis testavimo metodas bus taikomas tolygiai su projekto realizavimu. Testavimo metodo pagalba bus aiškinamasi kaip pagrindiniai testavimo komponentai ir klasės išpildo savo individualų funkcionalumą. Testai bus pagrinde leidžiami automatizuotai. Testavimo rezultatai nebus saugomi, rezultatai bus paruošiami atliekant testus tik tam reikalingu laiku – nėra prasmės saugoti rezultatų sistemos, kuri buvo prieš kažkokį laiko tarpą.

### Integravimo testavimas

Integravimo testavimo metu bus taip pat fokusuojama į saityno posistemę. Šio testavimo metodo metu bus bandomas vienetų tarpusavio sąveika. Integracinis testavimas padės išsiaiškinti ar visi moduliai sąveikaudami išpildo bendrą funkcionalumą. Šie testai bus realizuojami, kaip ir vienetų testai, vykdant projekto kūrimą. Jie bus leidžiami kartu su vienetų testais. Rezultatai nebus saugomi dėl tų pačių priežasčių kaip ir vienetų testai.

### NEUR testavimas

Svarbu paminėti neuroninių tinklų testavimą. Šis testavimas bus surištas su kuriama sistema – sistemos paskirtis atlikti vartotojo sąsajos nuotraukų analizes, todėl, norint pagerinti jų efektyvumą, reikia nuolatos tobulinti analizes atliekančius tinklus. Šio testavimo rezultatai bus saugomi pačioje sistemoje – apmokinto tinklo versijos bus surištos su šio testavimo metu įgyvendinto tikslumo procentu.

### Priėmimo testavimas

Priėmimo testavimas bus atliekamas baigus realizuoti sistemą. Metodo metu bus bandoma išsiaiškinti kaip programa atitinka, jai paruoštas specifikacijas. Priėmimo testavimo metu bus naudojami įeinantys duomenys ir tiriamas išeinančių rezultatų atitikimas specifikacijose nurodytiems rezultatams. Metodas išskirstytas į dvi dalis. Pirmos dalies metu bus žiūrimą į sistemos funkcionalumo išbaigtumą, minėto testavimo pagalba. Antros dalies metu bus atkreipiamas dėmesys į vartotojo sąsają. Rezultatai bus saugojami kaip sąrašas sunumeruotų specifikacijos punktų su gretimai laikomu jų įvertinimu.

### Aukšto lygio testavimas

Aukšto lygio testavimo metodo pagalba bus bandomi sistemos nefunkciniai reikalavimai. Šio testavimo metu bus skiriamas dėmesys SEG ir KLAS tinklams. Jo metu bus matuojamas veikimo su testavimo duomenimis tikslumas ir greitaveika.

## Testavimo ištekliai

Techniniai ištekliai:

* VPGSA sistema,
* Kompiuteris, kuriame bus atliekamas testavimas.

Žmogiškieji ištekliai:

* Programuotojas ruošiantis ir atliekantis testus.

## Testavimo rezultatai

Testavimo rezultatai tiesiogiai kaupiami bus tik priėmimo, NEUR testų ir aukšto lygio testavimo metu.

## Testavimo įrankiai ir aplinka

Didžioji dalis testavimų bus atliekami kompiuteryje, kuriame jie ir buvo paruošti. Automatiniams testams automatizuoti planuojama naudoti „Travis“ įrankį.

Kaip ir sistemos kūrimas testavimai bus atliekami konteinerių technologijos pagalba – naudojant „Docker“ įrankį.

Testavimo sistemos aplinkos reikalavimai atlikti vienetų, integracinius ir NEUR testus:

* Linux OS,
* Docker,
* Docker Compose,
* Docker Nvidia.

Testavimo sistemos aplinkos reikalavimai atlikti priėmimo ir aukšto lygio testus:

* Naršyklė

## Testavimo tvarkaraštis

**2 lentelė.** Testavimo tvarkaraštis

|  |  |  |
| --- | --- | --- |
| **Veikla** | **Terminas** | **Atliekami darbai** |
| Testavimo planas | 2019-09-25 – 2019-10-02 | Kuriamas testavimo plano dokumentas. |
| Sistemos testavimas | 2019-10-02 – 2019-11-15 | Pradedami kurti vienetų ir integracijos testavimai. |
| Ruošimasis galutiniam sistemos testavimui | 2019-11-15 – 2019-12-22 | Ruošimasis aukšto lygio ir priėmimo testavimams. |
| Galutinis sistemos testavimas | 2019-12-23 | Atliekamas aukšto lygio ir priėmimo testavimas. |

# Testavimo procedūra

## Testuojama programų sistema

Testuojama programinė įranga aprašyta 2.1 skyriuje.

## Testavimo procedūros

Šiame skyriuje toliau specifikuojami testavimo metodai aptarti 2.2 skyriuje.

Vienetų ir Integravimo testai paremti tuo, kad jie bus atliekami automatizuotai, naudojantis „Travis“ įrankį, kuris kiekvieno **Komito** metu paleis testus ir išves rezultatus apie kodo stabilumą.

### Vienetų testavimas

Kiekviena saityno posistemės klasė turės atitinkamą testų failą, kuriame bus laikomi visi testuojamos klasės metodų vienetų ir integraciniai testai. Vienetų testai bus paprasti jie turės du pagrindinius tikslus – teisinga veikimą ir kodo aprėptį. Siekiant pirmo tikslo metu - jeigu testuojamas metodas turi grąžinamą reikšmę, bus testuojama ši grąžinama reikšmė su visais prieinamais kintamaisiais – ar rezultatas atitinka numatytus rezultatus. Antro tikslo metu – bus stengiamasi padengti visas kodo eilutes ir sąlygų šakas siekiant užtikrinti, kad įvykdžius šitą testą su numatytomis įvestimis kodo vykdymas nesutiks nenumatytų klaidų.

### Integravimo testavimas

Integravimo testavimas bus atliekamas panašiai kaip ir vienetų testavimas, tačiau apims didesnes apimtis sąveikaujančių komponentų. Visi integravimo testai bus pradedami atlikti nuo vartotojo sąsajos pusės – bus simuliuojama vartotojų įvestis per naršyklę. Integravimo testai bandys išpildyti tuos pačius tikslus kaip ir vienetų testai. Pirmo tikslo metu bus tikrinama ar rezultatai atitiks numatytus rezultatus prie numatytų simuliuotų įvesčių. Antro tikslo metu bus tikrinamas kuo didesnis kiekis numatytų vartotojo įvesčių, kad padengti didžiausia kiekį kodo eilučių ir stebėti ar kodo vykdymas nesutinka nenumatytu klaidų.

### NEUR testavimas

Šis testavimas bus atliekamas naudojant kaupiamus apmokymo ir testavimo duomenis. Kuriant naujus SEG ir KLAS modelius jie bus apmokinami sukauptais duomenimis ir tada testuojami su atsitiktine tvarka parinktais testavimo duomenimis. Pasibaigus testui apmokintas modelis bus užsaugomas atskirame faile gretimai su savo versijos identifikatoriumi ir procentu rodančiu tikslumą teste panaudotiems duomenims.

### Priėmimo testavimas

Priėmimo testavimo metu bus kreipiamas dėmesys į vartotojo naudojimą ir sistemos atitikimą reikalavimams numatytiems specifikacijoje. Šio testavimo metu bus vykdomi veiksmai patikrinantys 2.2.4 skyriuje aprašytą sąrašą. Ir vertinama vartotojo reakcija į paruoštą sistemos grafinę sąsają.

### Aukšto lygio testavimas

Aukšto lygio testavimo metu bus kreipiamas dėmesys į pagrindinį SEG ir KLAS veikimą naudojantis saityno posisteme. Bus pasirenkamas vienas, atsitiktinai parinkto saityno puslapis ir jam atliekama analizė.

## Testavimo išteklių paskirstymas

Techniniai ištekliai:

* VPGSA sistema – bus pagrindinis testuojamas objektas.
* Kompiuteris – bus naudojamas atlikti testavimą.

Žmogiškieji ištekliai:

* Programuotojas atliks minėtus testus.

## Testavimo rezultatų kaupimas

Vienetų ir integravimo testai neturi saugomų rezultatų po automatiniais testais – jų veikimo efektyvumas bus nuskaitomas paleidžiant abejas rūšis testų ruošiant galutinę ataskaitą.

Priėmimo testų rezultatai bus saugomi „Microsoft Word“ dokumento sąraše.

Aukšto lygio testavimas bus atliekamas užsakovo aplinkoje, todėl rezultatai bus užsaugomi „Microsoft Word“ dokumente to testavimo metu.

Išvados

1. Buvo paruoštas pradinis Testavimo planas VPGSA sistemai.
2. Plano rengimo metu buvo pastebėtos vietos vertos tolimesnio apmastymo – rezultatų saugojimas ir vartotojo sąsajos automatiniai testai.

Literatūros sąrašas

1. Ambrazevičius, M. (2019). *Architektūros specifikacija.*
2. Ambrazevičius, M. (2019). *Reikalavimų specifikacija.*