



**Kauno technologijos universitetas**

Informatikos fakultetas

## **Modulis „Programų kokybės užtikrinimo metodai“**

Testavimo planas

**IFM 4/2 gr. Danielė Stasiūnaitė**  
Studentė

**Dr. Šarūnas Packevičius**  
Dėstytojas

**Kaunas, 2025**

# Turinys

1	Įvadas	2
2	Testavimo apimtis	2
3	Testavimo strategijos	3
4	Pradinės sąlygos	4
5	Testavimo prioritetai	5
6	Rezultatai	6
7	Testavimo aplinka	7
8	Testavimo scenarijai	8
9	Testavimo kalendorius	11
10	Testavimo rizikos	12

# 1 Įvadas

Šiame dokumente yra aprašytas testavimo planas, kuris yra skirtas patikrinti, ar savarankiškos suverenios asmens tapatybės valdymo sistema, skirta darbui su asmenų biologiniais duomenimis, veikia taip, kaip yra numatyta parengtoje sistemos specifikacijoje.

## 2 Testavimo apimtis

Numatyta, kad sistemos testavimas bus vykdomas, atliekant visų sistemos funkcinių dalių testavimą, kur funkcinės dalys yra:

- Registracijos modulio veikimas (paskyros kūrimas skirtingoms sistemos naudotojų grupėms: pacientams, gydytojams - genetikams, tyrėjams);
- Sistemos naudotojų - pacientų - biologinių duomenų prieigos valdymas;
- Duomenų šifravimo ir pseudonimizavimo funkcijos;
- Biologinių duomenų įkėlimo funkcionalumas;
- Biologinių duomenų analizės funkcionalumas;
- Užklausų biologinės analizės atlikimui generavimas;
- Analizės rezultatų pateiktis.

### 3 Testavimo strategijos

Antrame skyriuje (žr. *antrą skyrių*) aprašytos sistemos funkcinės dalys bus testuojamos, taikant šias testavimo strategijas:

- **Vienetų testavimą** (angl. *Unit testing*), kai bus testuojami atskiri sistemos komponentai;
- **Integracijos testavimą** (angl. *Integration testing*), kai bus testuojama, kaip atskiri sistemos komponentai veikia tarpusavyje;
- **Sistemos testavimą** (angl. *System testing*), kai bus testuojama, kaip sistema veikia kaip visuma;
- **Priėmimo testavimą** (angl. *Acceptance testing*), kai bus tikrinama, ar sistema atitinka visus užsakovo reikalavimus;
- **Alfa ir beta testavimą**, kai sistemos funkcionalumą testuos sistemos kūrėjai (*alfa* testavimas) ir grupelė galutinių sistemos naudotojų (*beta* testavimas);
- **Našumo testavimą**, kai bus tikrinama, ar sistema tenkina sistemos našumo reikalavimus (veikia pakankamai greitai ir efektyviai);
- **Saugumo testavimą**, kai bus tikrinama, ar sistema yra pasiekiamo tik autorizuotiems sistemos naudotojams.

## 4 Pradinės sąlygos

Tam, jog galėtų būti pradėtas testavimo etapas, turi būti:

- Pilnai suformuota sistemos testavimo komanda, kur kiekvienas komandos narys tiksliai žino savo roles ir atsakomybes, atliekant testavimą.
- Pilnai paruošta ir patvirtinta sistemos reikalavimų specifikacija, pagal kurią būtų galima testuoti sistemos funkcionalumą.
- Paruošti ir patvirtinti testavimo scenarijai, pagal kuriuos bus atliekamas testavimas.
- Paruošti testavimo duomenys, kurie bus naudojami testavimo metu (skirtingų kategorijų sistemos naudotojų paskyros, skirtingi biologinių duomenų failai, skirtingi analizės atlikimo užklausų pavyzdžiai).
- Pilnai paruošta ir sukonfigūruota testavimo aplinka, kurioje bus atliekamas sistemos testavimas (duomenų bazės paruošimas, reikalingų programinės įrangos komponentų ir testavimo įrankių įdiegimas ir konfigūravimas).
- Paruoštas testavimo planas, kuriame pateikta visa su testavimu susijusi informacija bei pateikti testavimo scenarijai, kuriuose turės būti fiksuojamas testavimo rezultatas.
- Paruoštas testavimo klaidų ataskaitos šablonas, pagal kurį bus fiksuojamos testavimo metu identifikuotos klaidos.

## 5 Testavimo prioritetai

Testavimas turi būti atliekamas pagal šiuos nustatytus prioritetus:

- **Iš pradžių turi būti testuojami sistemos naudotojui matomi sistemos funkciniai reikalavimai:**
  - Ar naujas sistemos naudotojas gali užsiregistruoti sistemoje kaip pacientas, gydytojas - genetikas arba tyrėjas.
  - Ar prisijungęs naudotojas pagal savo kategoriją gali pasiekti jam skirtą funkcionalumą. Pavyzdžiui, ar pacientas gali įkelti savo biologinius duomenis ir valdyti, kokie asmenys gali šiuos duomenis pasiekti.
  - Ar gydytojas - genetikas gali sukurti pacientų medicininės kortelės įrašus, pasiekti pacientų biologinius duomenis ir sugeneruoti analizės atlikimo užklausą tyrėjui.
  - Ar tyrėjas gali pasiekti atitinkamų pacientų biologinius duomenis (jei iš gydytojo - genetiko yra gavęs analizės atlikimo užklausą), ar gali atlikti biologinių duomenų analizę ir perduoti analizės atlikimo rezultatus gydytojui - genetikui.
- **Toliau turi būti testuojami sistemos naudotojui nematomi sistemos funkciniai reikalavimai:**
  - Ar korektiškai veikia naudotojo autentifikavimo funkcijos.
  - Ar sistemos naudotojų - pacientų - biologiniai duomenys yra korektiškai šifruojami ir pseudonimizuojami.
- **Po funkcinų reikalavimų testavimo vykdomas aukšto prioriteto sistemos nefunkcinių reikalavimų testavimas:**
  - Ar autentifikacijos operacijos atlikimo greitis neviršija 1 sekundės.
  - Ar paciento biologinių duomenų prieigos patikrinimas netrunka ilgiau nei 500 milisekundžių.
- **Galiausiai vykdomas žemo prioriteto sistemos nefunkcinių reikalavimų testavimas:**
  - Ar naudotojo sąsajos elementai yra tinkamai išdėstyti.
  - Ar sistemoje realizuoti iššokantys langeliai netrukdo ir neerzina.
  - Ar sistema yra intuityvi ir pritaikyta vyresnio amžiaus žmonėms.
  - Ar sistemoje yra pateikti skirtingų techninių terminų bei privalomų įvesties laukų paaiškinimai.

## 6 Rezultatai

Atlikus testavimą bus pilnai užpildyti arba sukurti dokumentai - testavimo rezultatai:

- Bus pilnai užpildytas testavimo planas (prie testavimo scenarijų nurodant testavimo rezultatą ir testo būseną);
- Bus sukurta testavimo klaidų ataskaita;
- Su užsakovu bus pasirašytas dokumentas, patvirtinantis, kad sistema atitinka visus užsakovo sistemai iškeltus reikalavimus.

## 7 Testavimo aplinka

Žemiau aprašyta testavimo aplinkos - aparatinės ir programinės įrangos bei testavimo įrankių - konfigūracija:

- **Aparatinė įranga:**

- Serveris. 8-16 branduolių CPU, 32-64 GB RAM, SSD diskas (min. 512 GB).
- Darbo kompiuteriai. Testavimui turi būti naudojami kompiuteriai, turintys Windows 10/11 ir Linux (Ubuntu 20.04+) operacines sistemas.

- **Programinė įranga:**

- MySQL (8.0+ versija). Duomenų bazės valdymas.
- Python (3.10+ versija). *Back-end* API kūrimas ir testavimas.

- **Testavimo įrankiai:**

- PyTest. Vienetų (angl. *unit testing*) ir integracijos testavimas.
- Selenium. UI testavimas.
- Postman. API testavimas.
- SoapUI. Saugumo testavimas.
- Apache JMeter. Našumo testavimas.



## 8 Testavimo scenarijai

Žemiau aprašyti keli sistemos funkcionalumo testavimo scenarijai, pagal kuriuos bus atliekamas testavimas.

1 lentelė. Testavimo scenarijus Nr. 1.

<b>Kodas:</b>	TS_001.
<b>Pavadinimas:</b>	Naudotojo (paciento) paskyros sukūrimo testas.
<b>Tikslas:</b>	Patikrinti, ar sistemos svečias gali susikurti sistemos paskyrą kaip sistemos naudotojas - pacientas.
<b>Pradinės testavimo sąlygos:</b>	Sistemos svečias turi būti atsidaręs sistemos paskyros kūrimo langą.
<b>Įvestis:</b>	Svečio vardas, pavardė, el. paštas, telefono numeris, lytis, adresas, slaptažodis.
<b>Testo etapai<sup>a</sup>:</b>	<ol style="list-style-type: none"><li>1. Naudotojas užpildo pateiktos asmeninės paskyros kūrimo formos laukus.</li><li>2. Naudotojas išsaugo įvestą informaciją, paspausdamas išsaugojimo mygtuką.</li><li>3. Parodomas informacinis pranešimas, informuojantis apie sėkmingai sukurtą asmeninę paskyrą.</li><li>4. Sistema prijungia naudotoją prie jo asmeninės paskyros.</li><li>5. Sistema atidaro naudotojo asmeninės paskyros langą.</li></ol>
<b>Tikėtinas rezultatas:</b>	Sukurtas naujas sistemos naudotojas, kuris priklauso kategorijai „Pacientas“.
<b>Tikras rezultatas:</b>	<i>Užpildoma testavimo metu...</i>
<b>Būsena:</b>	<i>Užpildoma testavimo metu: „Testas sėkmingai įvykdytas“ arba „Gauta klaida“.</i>

<sup>a</sup>Čia ir toliau žalia spalva pažymėti naudotojo veiksmai.

**2 lentelė. Testavimo scenarijus Nr. 2.**

<b>Kodas:</b>	TS_002.
<b>Pavadinimas:</b>	Biologinių duomenų įkėlimo testas.
<b>Tikslas:</b>	Patikrinti, ar sistemos naudotojas - pacientas - gali įkelti savo biologinius duomenis į sistemą.
<b>Pradinės testavimo sąlygos:</b>	Sistemos naudotojas - pacientas - turi būti prisijungęs prie sistemos ir atsidaręs biologinių duomenų įkėlimo langą.
<b>Įvestis:</b>	Genominių sekų failas su kokybiniais įverčiais ( <i>.fastq</i> ).
<b>Testo etapai:</b>	<ol style="list-style-type: none"> <li>1. Naudotojas užpildo pateiktos duomenų įkėlimo formos laukus ir prideda biologinius duomenis saugantį failą.</li> <li>2. Naudotojas išsaugo įvestą metainformaciją bei pridėtą failą, paspausdamas išsaugojimo mygtuką.</li> <li>3. Sistema validuoja failo formatą ir turinį.</li> <li>4. Sistema užšifruoja duomenis ir išsaugo juos duomenų bazėje.</li> <li>5. Sistema priskiria įrašui identifikatorių ir susieja jį su naudotojo paskyra.</li> <li>6. Parodomas informacinis pranešimas, informuojantis apie sėkmingai įkeltus duomenis.</li> <li>7. Naudotojas peržiūri įkeltų duomenų įrašą savo paskyros skiltyje.</li> </ol>
<b>Tikėtinas rezultatas:</b>	Failas sėkmingai įkeltas, užšifruotas ir išsaugotas sistemoje.
<b>Tikras rezultatas:</b>	<i>Užpildoma testavimo metu...</i>
<b>Būsena:</b>	<i>Užpildoma testavimo metu: „Testas sėkmingai įvykdytas“ arba „Gauta klaida“.</i>

3 lentelė. Testavimo scenarijus Nr. 3.

<b>Kodas:</b>	TS_003.
<b>Pavadinimas:</b>	Biologinių duomenų prieigos valdymo testas.
<b>Tikslas:</b>	Patikrinti, ar sistemos naudotojas - pacientas - gali valdyti, kas gali pasiekti jo įkeltus biologinius duomenis.
<b>Pradinės testavimo sąlygos:</b>	Sistemos naudotojas - pacientas - turi būti prisijungęs prie sistemos ir atsidares biologinių duomenų prieigos valdymo langą.
<b>Įvestis:</b>	Duomenų prieigos valdymo lange iš teisių sąrašo reikia pasirinkti konkrečioms asmenims priskiriamas teises.
<b>Testo etapai:</b>	<ol style="list-style-type: none"> <li>1. Sistema pateikia paciento įkeltų biologinių duomenų sąrašą.</li> <li>2. Naudotojas pasirenka konkretų biologinių duomenų sąrašą įrašą.</li> <li>3. Sistema pateikia naudotojų, turinčių prieigą prie konkrečių biologinių duomenų, sąrašą.</li> <li>4. Naudotojas redaguoja suteiktas prieigos teises sistemos naudotojams: pratęsia prieigos laikotarpį arba atšaukia prieigą.</li> <li>5. Naudotojas suteikia naujas prieigas naujiems sistemos naudotojams.</li> <li>6. Sistema atnaujina naudotojams suteiktų prieigų sąrašą.</li> <li>7. Parodomas informacinis pranešimas, informuojantis apie sėkmingai atliktą atnaujinimą.</li> <li>8. Sistema informuoja atitinkamus sistemos naudotojus apie prieigos teisių pasikeitimus.</li> </ol>
<b>Tikėtinas rezultatas:</b>	Sistema leidžia sistemos naudotojui - pacientui - valdyti prieigą prie duomenų. Asmuo, kuriam suteikiama prieiga prie paciento įkeltų biologinių duomenų, turi galėti vykdyti šių duomenų peržiūrą.
<b>Tikras rezultatas:</b>	<i>Užpildoma testavimo metu...</i>
<b>Būsena:</b>	<i>Užpildoma testavimo metu: „Testas sėkmingai įvykdytas“ arba „Gauta klaida“.</i>

## 9 Testavimo kalendorius

Žemiau pateiktoje lentelėje yra pateiktas testavimo kalendorius, kuriame nurodyta skirtingų testavimo etapų pradžia ir pabaiga.

4 lentelė. Testavimo kalendorius.

Testavimo metodas	Pradžia	Pabaiga
Vienetų testavimas	2025-09-01	2025-09-14
Integracijos testavimas	2025-09-15	2025-09-28
Našumo testavimas	2025-09-29	2025-10-12
Saugumo testavimas	2025-10-13	2025-10-26
Sistemos testavimas	2025-10-27	2025-11-16
<i>Alfa</i> ir <i>beta</i> testavimas	2025-11-17	2025-11-30
Priėmimo testavimas	2025-12-01	2025-12-14

## 10 Testavimo rizikos

Žemiau pateiktoje lentelėje yra aprašytos rizikos, kurios gali sutrikdyti testavimo proceso vykdymą.

5 lentelė. Testavimo rizikų aprašymas.

Rizika	Aprašymas	Rizikos tikimybės mažinimas
Nepakankamas testuotojų skaičius.	Testuotojai gali dirbti prie kelių skirtingų projektų. Dėl to išauga vėluojančių testavimo veiklų rizika.	Sprinto planavimo susitikimų metu suplanuoti visas testavimo veiklas ir, esant poreikiui, į projektą įtraukti daugiau žmonių.
Identifikuojamos kritinės klaidos.	Testavimo metu identifikuojamos klaidos, dėl kurių negalima testuoti kitų sistemos funkcionalumų.	Užtikrinti, kad parašytas programinis kodas yra peržiūrimas visų komandai priklausančių programuotojų.
Sistemos realizacijos metu pasikeičia reikalavimai.	Testavimo metu užsakovas išreiškia poreikį pakeisti tam tikrus sistemai keltus reikalavimus - testuojamas funkcionalumas gali neatitikti užsakovo poreikių.	Kuriamos sistemos reikalavimų analizės metu detalai išsiaiškinti sistemai keliamus funkcinius ir nefuncinius reikalavimus bei juos pasitvirtinti su užsakovu. Su užsakovu pasirašomoje sutartyje nurodyti, kad reikšmingai sistemos veikimą keičiančių reikalavimų realizacija galima tada ir tik tada, kai pirmoji sistemos versija yra sudiegta į užsakovo aplinkas.
Nestabili arba nepilna testavimo aplinka.	Testavimo aplinka gali būti nepilnai paruošta testavimo vykdymui (duomenų bazės struktūros neatitikimai, neteisingai sukonfigūruotas serveris, nepasiekiamas API).	Į sistemos kūrimo komandą turi būti įtraukti patyrę programuotojai, analitikai bei testuotojai.