A black background with a black square

Description automatically generated with medium confidence

**Kauno technologijos universitetas**

Informatikos Fakultetas

**Programų Sistemų Testavimas (T120B162)**

1-ojo laboratorinio darbo ataskaita

|  |  |
| --- | --- |
| Atliko IFF-2/1 grupės studentai: |  |
| Rokas Gudžiūnas  Gedmantas Šilinskas  Rugilė Jovaišaitė  Simona Gerikaitė | (parašas) (data) |
| Priėmė: |  |
| doc. prakt. Guogis Evaldas | (parašas) (data) |
|  |  |

**KAUNAS, 2024**

TURINYS

[Įžanga 3](#_Toc161705314)

[Testavimo apimtis 3](#_Toc161705315)

[Testavimo strategija 4](#_Toc161705316)

[Pradiniai reikalavimai 4](#_Toc161705317)

[Testavimo prioritetai 4](#_Toc161705318)

[Testavimo tikslai 5](#_Toc161705319)

[Testavimo technikos 5](#_Toc161705320)

[Rolės ir atsakomybės 6](#_Toc161705321)

[Rezultatai 6](#_Toc161705322)

[Testavimo aplinka 6](#_Toc161705323)

[Testavimo scenarijai 6](#_Toc161705324)

[Testų valdymo planas 9](#_Toc161705325)

[Testavimo tvarkaraštis 9](#_Toc161705326)

[Testavimo rizikos 9](#_Toc161705327)

[Appendix B. Test Cases in Gherkin 11](#_Toc161705328)

[Darbo užduotys: 12](#_Toc161705329)

# Įžanga

Šio dokumento tikslas – pateikti, kokias testavimo metodikas naudosime, jog užtikrintumėme, kad mūsų internetinė treniruočių ir asmeninio progreso sekimo platforma „GYMZII“ funkcionuotų, kaip reikia.

Paveikslėlis, kuriame yra tekstas, ekrano kopija, dizainas

Automatiškai sugeneruotas aprašymas

### Testavimo apimtis

Testavimą sudaro šie testavimo būdai:

1. Programinės įrangos funkcinis testavimas – tai būdas nustatyti, ar programinė įranga veikia taip, kaip tikimasi. Tai susiję ne su tuo, kaip vyksta apdorojimas, o su tuo, ar apdorojant gaunami teisingi rezultatai.
   1. Siekiama ištestuoti 80% parašyto kodo vieneto (unit) testais.
   2. Testavimo tikslas – įsitikinti, jog kuriamos funkcijos programai veikia teisingai ir grąžina tuos atsakymus, kurių yra tikimasi.
2. Programinės įrangos nefunkcinis testavimas – tikrinami nefunkciniai programinės įrangos kūrimo aspektai, t. y. tikrinama ne programos rezultatai, o kaip veikia programa. Naudojami testavimo būdai : streso testavimas (skirtas patvirtinti programos stabilumą ir patikimumą ekstremaliomis sąlygomis) ir našumo testavimas (iškeltų sistemos našumo reikalavimų patvirtinimui).
   1. Tikslai :
      1. Patikrinti, ar programa veikia pagal testavimo scenarijų ir yra stabili;
      2. Patikrinti, kiek daugiausiai vartotojų gali būti prisijungę vienu metu, kol programa nebeveikia;
      3. Patikrinti programos reakciją – laiką, per kurį programa atsako vartotojui.
3. Vartotojo sąsajos programinės įrangos testavimas – vertinamas programos vizualiųjų elementų našumas ir bendras funkcionalumas.
   1. Tikslas – įsitikinti, jog grafinė vartotojo sąsaja veikia be trikdžių ir grąžina vartotojui tokius atsakymus, kurių yra tikimasi.
4. Sistemos sąveika su kitomis sistemomis – tikrinamas programos sąveikavimas su kitomis sistemomis.
   1. Tikslas – įsitikinti, jog sistema tinkamai sąveikauja su kitomis sistemomis, pavyzdžiui, „back-end“ su duomenų baze.

### Testavimo strategija

Mūsų testavimo strategija yra grindžiama bendru aukštu testavimo lygiu, kuris apima įvairius testavimo tipus, siekiant užtikrinti, kad mūsų internetinis puslapis, sukurtas naudojant ABP karkasą, veiktų teisingai, stabiliai ir patikimai.

Ši strategija apima:

* **Unit testing**: Šis testavimas yra skirtas patikrinti atskirų saityno programos komponentų funkcionalumą. Planuojama ištestuoti 80% programos kodo.
* **Integration testing**: Šis testavimas yra skirtas patikrinti, kaip skirtingi komponentai veikia kartu. Planuojama ištestuoti 60% programos kodo.
* **Static testing**: Šis testavimas yra atliekamas analizuojant kodą be jo vykdymo, siekiant atrasti klaidas. Planuojama ištestuoti 100% programos kodo.
* **Performance testing**: Šis testavimas yra atliekamas, siekiant įvertinti saityno programos veikimą esant įvairioms apkrovoms.
* **Acceptance testing**, kuris yra ypač svarbus, nes patikrina, ar saityno programa atitinka vartotojų lūkesčius ir ar ji yra aukštos kokybės. Šis testavimas yra orientuotas į saityno programos testavimą iš galutinio vartotojo perspektyvos ir yra vienas iš paskutinių vykdomų testavimo veiksmų.

Ši strategija yra taikoma mūsų internetiniame puslapyje ir yra būtina, siekiant užtikrinti, kad internetinis puslapis veiktų taip, kaip numatyta. Be šių testavimų, internetinis puslapis negalėtų veikti taip, kaip numatyta.

### Pradiniai reikalavimai

Šie reikalavimai bus įvykdyti prieš pradedant testavimą:

1. Yra veikianti internetinė treniruočių ir asmeninio progreso sekimo platformos implementacija.
2. Naudojamas GIT, kad būtų galima atkurti senesnę projekto versiją katastrofinės klaidos atveju.
3. Apibrėžtos svetainės panaudojimo galimybės.
4. Išsirinkta testavimo aplinka.
5. Paskirti testavimo resursai.
6. Sudarytas testavimo planas.

### Testavimo prioritetai

Mūsų svetainės testavimo prioritetai yra tokie:

1. Duomenų bazė – tikrinama, ar vartotojų duomenys bei kitos esybės yra duomenų bazėje.
2. Veikianti svetainė– tikrinama, ar svetainė funkcionuoja;
3. Esybės – tikrinama implementacija ir su esybėmis susiję metodai.
4. Vizualizacija – testuojama ar svetainės dizainas nepasikeičia nuo kitokių naudojimo aplinkybių (kita interneto naršyklė ir pan.).

### Testavimo tikslai

1. „Unit testing“. Bus rašomi vienetų testai tikrinti funkcionalumą iš „backend“ pusės.
2. „Integration testing“. Užtikrinti, kad skirtingi komponentai (pavyzdžiui, front-end ir back-end) tinkamai sąveikautų tarpusavyje, teisingai mainytųsi duomenimis ir užtikrintų sklandų bendrą aplikacijos veikimą. Integracijos testams skiriamas padengimas turėtų būti apie 60%.
3. „Acceptance / Qualification Testing“. Programinės įrangos priėmimo testavimas, siekiant patikrinti, ar jos funkcionalumas atitinka vartotojo lūkesčius. Šio tipo testavimui bus skiriamas 70% padengimas. Tai apimims galimų vartotojų scenarijų testavimą. Šio tipo testavimas leidžia nustatyti, ar sistema atitinka priėmimo kriterijus, palyginant norimas sistemos elgsenas su kliento reikalavimais.
4. „Performance Testing“. Tikrina, ar programinė įranga atitinka nustatytus našumo reikalavimus: puslapio įkėlimo laikas neviršyja 2-3 sekundžių, kiekvieno naudotojo veiksmo atsako laikas, pavyzdžiui, treniruočių planų peržiūros ar asmeninės informacijos atnaujinimas, neviršyja 1 sekundės, TTFB (Time to First Byte) turėtų būti ne ilgesnis kaip 200 ms (milisekundžių). Šių reikalavimų išpildymas leis užtikrinti, kad internetinėms svetainė veikia greitai ir efektyviai.
5. „Security Testing“ . Saugumo testavimas, kuris patikrina sistemos ir jos duomenų konfidencialumą, vientisumą ir prieinamumą. Tai yra svarbu norint užtikrinti, kad internetinė svetainė būtų saugi naudotojams.

### Testavimo technikos

Testavimo technikos, kurias naudosime mūsų internetinės treniruočių ir asmeninio progreso sekimo platforma projekte, sukurtame naudojant ABP karkasą ir Angular, yra šios:

1. Testavimo Scenarijai su Duomenimis – šie scenarijai apibrėžia aiškiai suformuluotas naudojimo atvejų sekas, turinčias iš anksto apibrėžtus įvesties duomenis ir laukiamus rezultatus. Tai leidžia tiksliai patikrinti konkrečių funkcijų elgseną ir užtikrinti, kad jos atitiktų verslo reikalavimus.
2. Testavimo Scenarijai be Duomenų – testuotojai šiuose scenarijuose patys renkasi įvesties duomenis testavimo metu. Tai suteikia lankstumo testuojant ir leidžia atlikti išsamesnius, neplanuotus testus, kurie gali atskleisti netikėtas klaidas ar elgsenos modelius.
3. Neautorizuoto Naudojimo Testai – šie testai simuliuoja neautorizuotų vartotojų bandymus prieiti prie ribojamos informacijos ar funkcionalumo. Tai padeda ištestuoti programos saugumo mechanizmus ir užtikrinti, kad duomenys bei funkcijos būtų saugiai apsaugoti nuo nepageidaujamų prieigos bandymų.
4. Naudojamumo Kontrolinis Sąrašas – tai iš anksto sudarytas kontrolinis sąrašas, kuriame įvardijami kriterijai ir veiklos, padedančios įvertinti sistemos naudojamumą. Šis metodas leidžia objektyviai įvertinti, ar vartotojo sąsaja yra patogi ir intuityvi galutiniams naudotojams.
5. Našumo Statistika – Rinkdami ir analizuodami našumo charakteristikas, palyginame jas su iš anksto nustatytais parametrais. Tai apima tokias metrikas kaip atsako laikas, sistemos apkrova ir gebėjimas tvarkyti vartotojų užklausas. Ši technika leidžia mums nustatyti, ar internetinis puslapis atitinka našumo lūkesčius ir gebės veikti efektyviai net ir didelio apkrovimo sąlygomis.

### Rolės ir atsakomybės

1. Rokas Gudžiūnas – kokybės užtikrinimo vadovas. Atsakingas už testavimo proceso planavimą ir jo įgyvendinimą, tačiau taip pat dalyvauja testavimo procese.
2. Gedmantas Šilinskas – testuotojas. Vykdo išsamius funkcionalumo, našumo, saugumo ir kitus svarbius testus, remiantis testavimo planu.
3. Rugilė Jovaišaitė – testuotoja. Užtikrina, kad techninė įranga ir infrastruktūra būtų paruošta testavimui, asmuo taip pat dalyvauja paties testavimo procese.
4. Simona Gerikaitė – testuotoja. Dalyvauja kuriant testavimo scenarijus ir patikrina, ar produktas atitinka lūkesčius.

### Rezultatai

Tikimasi tokių testavimo rezultatų – internetinė treniruočių ir asmeninio progreso sekimo platforma funkcionuoja taip, kaip tikėtasi, atitinka visus kokybės reikalavimus, svetainė veikia sklandžiai ir ja lengva naudotis.

Po testavimo turėtų būti:

1. Įvykdyta bent 80% šio testavimo plano punktų.
2. Dalis testavimo rezultatų užfiksuoti per Zephyr – testavimo valdymo įskiepį, kuris leidžia mums efektyviai sekti ir valdyti testavimo ciklus, užduotis ir rezultatus. Zephyr įskiepis JIRA platformoje pateikia testų vykdymo ataskaitas, suteikia greitą prieigą prie testų vykdymo rezultatų ir rodo testų vykdymo rezultatus pagal testavimo ciklą.

### Testavimo aplinka

Integravus ABP karkasą į mūsų internetinę treniruočių ir asmeninio progreso sekimo platformą, vienas iš svarbiausių privalumų yra tai, kad ABP automatiškai generuoja testų klases, ženkliai supaprastindamas ir pagreitindamas testavimo procesą. Ši funkcija leidžia mūsų komandai greitai pradėti rašyti ir vykdyti testus, nes nereikia praleisti papildomo laiko testų klasėms kurti rankiniu būdu. Tai suteikia galimybę efektyviau identifikuoti ir ištaisyti klaidas ankstyvose vystymo stadijose, taip užtikrinant aukštesnę kodo kokybę ir stabilumą. Dėl šios priežasties mūsų testavimo aplinka yra dar labiau praturtinta ir optimizuota, nes galime naudoti ABP generuotas testų klases kartu su pažangiais testavimo įrankiais ir priemonėmis, kaip Cucumber scenarijų testavimui ir JMeter našumo testavimui. Tai leidžia mums sukurti griežtą ir kokybišką testavimo procesą, kuris užtikrina, kad mūsų internetinis puslapis atitiktų aukščiausius kokybės standartus ir vartotojų lūkesčius.

### Testavimo scenarijai

1. Naudotojo paskyros sukūrimas

* Aprašymas – šis testavimas skirtas patikrinti, ar naudotojas gali sukurti savo paskyrą tik tada, kai jo įvesti duomenys atitinka tam tikras sąlygas.
* Pradiniai duomenys – pradinė duomenų bazė, kai naudotojas dar nėra užsiregistravęs.
* Testavimo žingsniai:
  1. Atidaryti internetinį puslapį.
  2. Paspausti mygtuką „I AM READY“.
  3. Užpildyti registracijos formą teisingais naudotojo duomenimis (slapyvardis, el. pašto adresu, slaptažodžiu, pakartotinu slaptažodžiu, pirminė raumenų masė, svoris ir kt.).
  4. Paspausti mygtuką „Registruotis“.
* Testavimo atvejai:

|  |  |  |
| --- | --- | --- |
| Naudotojas | El. paštas | Laukiamas rezultatas |
| Jonas | Jonas123@gmail.com | Paskyra sukurta |
| Petras | Petras1.com | Paskyra nėra sukurta |

1. Progreso duomenų atnaujinimas

* Aprašymas – šis testavimas skirtas patikrinti, ar naudotojui pakeitus savo raumenų masės ir svorio parametrus pasikeis raumenų masės ir svorio kitimo grafikai. Atspirties tašku duomenų lyginimui laikomi registracijos metu įvesti pradinės raumenų masės ir svorio matmenys.
* Pradiniai duomenys - pradinė duomenų bazė, kai naudotojas jau yra užsiregistravęs.
* Testavimo žingsniai:
  1. Prisijungti prie paskyros.
  2. Atidaryti viršutinėje antraštėje esančią skiltį „MY PROGRESS“.
  3. Atnaujinti raumenų masės ir svorio duomenis.
  4. Paspausti mygtuką „Išsaugoti pakeitimus“.
* Testavimo atvejai:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Esama raumenų masė | Atnaujinta raumenų masė | Esamas svoris | Atnaujintas svoris | Laukiamas rezultatas |
| 35 | 36 | 90 | 95 | Raumenų masės ir svorio grafikai rodo didėjimo tendenciją. |
| 30 | 28 | 70 | 65 | Raumenų masės ir svorio grafikai rodo mažėjimo tendenciją. |

1. Komentaro palikimas

* Aprašymas – šis testavimas skirtas patikrinti, ar naudotojas gali palikti komentarą, kurį vėliau galėtų pasiekti administratorius.
* Pradiniai duomenys – pradinė duomenų bazė, kai naudotojas jau yra užsiregistravęs.
* Testavimo žingsniai:
  1. Prisijungti prie paskyros.
  2. Atidaryti viršutinėje antraštėje esančią skiltį „CONTACTS“.
  3. Įvesti komentarą į laukelį „LEAVE A COMMENT“.
  4. Paspausti mygtuką „Send“.
* Testavimo atvejai:

|  |  |  |
| --- | --- | --- |
| Komentaro egzistavimas | Komentaro būsena | Laukiamas rezultatas |
| Sukurtas ir išsiųstas | Užfiksuotas ir matomas duomenų bazėje | Administratorius mato ir gali pasiekti komentarą |
| Nesukurtas ir neišsiųstas | Neužfiksuotas ir nematomas duomenų bazėje | Administratorius mato anksčiau buvusius komentarus |

1. Pasiekimų rinkimas ir saugojimas
   * Aprašymas – šis testavimas skirtas patikrinti, ar naudotojas, pasiekęs tam tikrus tikslus, gauna atitinkamus pasiekimus ir gali prie jų prieiti.
   * Pradiniai duomenys - pradinė duomenų bazė, kai naudotojas jau yra užsiregistravęs.
   * Testavimo žingsniai:
     1. Prisijungti prie paskyros.
     2. Atidaryti viršutinėje antraštėje esančią skiltį „ACHIEVEMENTS“.
     3. Patikrinti surinktų pasiekimų sąrašą.
   * Testavimo atvejai:

|  |  |  |
| --- | --- | --- |
| Naudotojo progresas | Pasiekimo būsena | Laukiamas rezultatas |
| Pradinis svoris: 90 kg  Norimas svoris: 80 kg  Dabartinis svoris: 85 kg | Įgytas pasiekimas – „Tu jau pusiaukelėje“, rodantis, jog pasiekta 50% numatyto tikslo | Pasiekimas įrašytas ir matomas „ACHIEVEMENTS“ skiltyje |
| Pradinis svoris: 90 kg  Norimas svoris: 80 kg  Dabartinis svoris: 88 kg | Naujų pasiekimų neįgyta | Naujų įrašų „ACHIEVEMENTS“ skiltyje nėra |

1. Naudotojo paskyros šalinimas
   * Aprašymas – šis testas skirtas patikrinti, ar administratorius gali sėkmingai ištrinti naudotojo profilį iš sistemos, jei naudotojo profilis nebuvo naudojamas ilgą laiko tarpą.
   * Pradiniai duomenys - administratorius yra prisijungęs prie sistemos ir turi tinkamas teises valdyti naudotojų profilius.
   * Testavimo žingsniai:
     1. Prisijungti prie sistemos kaip administratorius.
     2. Atsidaryti naudotojų paskyrų sąrašą.
     3. Nurodyti laiko periodą, kuriam praėjus naudotojo paskyra laikoma nebeaktyvia.
     4. Surasti atitinkamą paskyrą.
     5. Paspausti mygtuką „Delete profile“.
   * Testavimo atvejai:

|  |  |  |
| --- | --- | --- |
| Paskyros būsena | Atliekamas veiksmas | Laukiamas rezultatas |
| Paskyra neaktyvi 10 metų | Paskyra yra ištrinama | Naudotojo paskyra yra pašalinta iš sistemos duomenų bazės. Rodomas pranešimas apie sėkmingai įvykdytą veiksmą |
| Paskyra neaktyvi 10 dienų | Paskyra nėra ištrinama | Administratorius atšaukia šalinimą, naudotojo paskyra išlieka nepakitusi |

### Testų valdymo planas

Testų valdymo planas susideda iš šių dalykų:

1. Testų registravimas. Kiekvienas testas privalo būti aiškiai apibrėžtas – turėti tikslą, aiškius rezultatus. Šie duomenys registruojami testų valdymo sistemoje.
2. Testų defektų registravimas. Kiekvienas defektas turi būti užregistruotas testų valdymo sistemoje. Registruojamas defekto pobūdis, aptikimo data, jo svarba ir kiti susiję duomenys. Priskiriamas atsakingas žmogus, kuris defektą analizuoja ir tvarko.
3. Testavimo proceso stebėjimas. Testavimo procesas ir eiga turi būti prižiūrimi – analizuojami gauti duomenys, užtikrinamas struktūrizuotas ir efektyvus testavimas.
4. Testų vykdymo ataskaitos. Po kiekvieno testo vykdymo pateikiama ataskaita, kurioje aprašomi testavimo rezultatai, aptikti defektai ir kiti pastebėjimai.

Naudojama testavimo įranga – JIRA įskiepis Zephyr Scale. Šis įskiepis leidžia registruoti testus bei jų defektus, sekti testų vykdymą realiu laiku bei generuoja ir leidžia redaguoti testų vykdymo ataskaitas.

### Testavimo tvarkaraštis

|  |  |  |
| --- | --- | --- |
| Testavimo užduotis | Pradžia | Pabaiga |
| Unit testing | 2024-04-03 | 2024-04-15 |
| Performance testing | 2024-04-10 | 2024-04-21 |
| Acceptance testing | 2024-04-25 | 2024-05-10 |

### Testavimo rizikos

|  |  |  |
| --- | --- | --- |
| Rizika | Aprašymas | Sprendimo būdas |
| Laiko trūkumas testavimui | Dėl griežtų projektų terminų testavimui gali būti skiriama per mažai laiko. | Anksti projekto planavimo etape numatyti realistinius terminus testavimo fazėms. Taip pat, esant būtinybei, peržiūrėti prioritetus ir sutelkti dėmesį į svarbiausius funkcionalumus. |
| Techninės problemos | Techninės problemos, pvz., testavimo įrankių ar testavimo aplinkos gedimai, gali sutrikdyti testavimo procesą. | Periodiškai atnaujinti ir tikrinti testavimo įrankius bei aplinkas, kad užtikrintumėte jų stabilumą. Įdiegti atsarginių planų testavimo procesams, pvz., alternatyvių įrankių naudojimą. |
| Nepakankama komunikacija | Komunikacijos trūkumas tarp komandos narių gali lemti nesusipratimus ir potencialias klaidas. | Skatinti reguliarią ir atvirą komunikaciją tarp visų projekto dalyvių. Naudoti projektų valdymo įrankius, pvz., Jira, kad užtikrintumėte sklandų darbų sekimą ir atsakomybių priskyrimą. |