

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Internetinės aplikacijos Cronometer testavimas

Web appliaction Cronometer testing

Programų sistemų testavimo laboratorinis darbas

| | | |
|----------------|--------------------------------|-----------|
| Atliko: | 3 kurso studentas | |
| | Matas Savickis | (parašas) |
| Darbo vadovas: | Vytautas Valaitis, Asist., Dr. | (parašas) |

Vilnius – 2018

Anotacija

Šiuo darbu yra siekiama sukurti testavimo planą, testavimo atvejus ir atlikti testavimą www.cronometer.com internetinei aplikacijai skirtai sekti maisto informaciją, sporto aktyvumą ir biologinius rodiklius.

- Matas Savickis - savickis.matas@gmail.com

TURINYS

| | |
|--|---|
| ANOTACIJA | 1 |
| TESTAVIMO PLANAS | 2 |
| 1. TESTAVIMO PLANO IDENTIFIKATORIAI | 2 |
| 2. NUORODOS | 2 |
| 3. ĮVADAS | 2 |
| 4. TESTAVIMO OBJEKTAI | 2 |
| 5. RIZIKOS | 2 |
| 6. FUNKCIONALUMAS KURĮ TESTUOSIME | 2 |
| 7. FUNKCIONALUMAS KURIO NETESTUOSIME | 3 |
| 8. STRATEGIJA | 3 |
| 9. SĖKMĖS IR NESEKMĖS KRITERIJAI | 3 |
| 10. TESTO NUTRAUKIMO KRITERIJAI | 3 |
| 11. TESTŲ PLANAI | 4 |
| 12. LIKUSIOS UŽDUOTYS | 4 |
| 13. APLINKOS REIKALAVIMAI | 4 |
| 14. PERSONALAS IR REIKALINGI APMOKYMAI | 4 |
| 15. ATSAKOMYBĖS | 4 |
| 16. TVARKARAŠTIS | 5 |
| 17. RIZIKOS | 5 |
| TESTAVIMO ATVEJAI | 5 |
| 18. ĮVADAS | 5 |
| 19. TESTAVIMO ATVEJAI | 6 |
| 20. DEFEKTŲ APRAŠAS | 9 |

Testavimo planas

1. Testavimo plano indentifikatoriai

CMT-1.1.1 - pirmas skaičius rodo ar testavimas buvo atliktas telefonu ar kompiuterius(1 - kompiuteris 2 - telefonas) antras skaičius rodo kelinta funkcionalumą testuosime, trečias skaičius rodo kelintą scenarijų testuojame

2. Nuorodos

- <https://jmpovedar.files.wordpress.com/2014/03/ieee-829.pdf>
- <http://pst.valaitis.net/PST>

3. Įvadas

Šis testavimo planas yra skirtas internetinės aplikacijos Cronometer testavimui. Pagrindinis testavimo tikslas bus įsitikinti ar su skirtingais įvesties duomenimis programa veikia korektiškai ir gražina korektiškus rezultatus. Bus testuojama maisto pridėjimas, sporto pridėjimas, biometrinių duomenų pridėjimas ir prisijungimas prie tinklalapio. Visas testavimas vyks rankiniu būdu.

4. Testavimo objektai

1. www.cronometer.com

5. Rizikos

- Nėra programos dokumentacijos
- Jeigu kažką užlaušiu ant manęs pyks Cronometer kūrėjai

6. Funkcionalumas kurį testuosime

1. Registravimasis į tinklalapį
2. Prisijungimas prie tinklalapio
3. Maisto vienetų pridėjimas sekimui, įvesties ir išvesties korektiškumas
4. Sporto sekimo pridėjimas sekimui, įvesties ir išvesties korektiškumas
5. Biometrinių duomenų pridėjimas į sistemą, įvesties ir išvesties korektiškumas

7. Funkcionalumas kurio netestuosi

1. Premium narystės funkcionalumas - kadangi testavimą atliks biednas studentas kuris neturi pinigų nusipirkti Premium narystės todėl šio funkcionalumo ir netestuosi
2. Sistemos suderinamumą su išnamiaisiais įrenginiais sekimui(Fitbit) - priežastis ta pati kaip ir Premium narystės atveju
3. Profilio funkcija - funkcionalumas nekritinis

8. Strategija

- Specialūs įrankiai - nėra
- Apmokymas naudotis įrankiais - nėra
- Metrikos - bus sekama kiek laiko užtruko atlikti testavimą, surastų klaidų skaičius, teisingai praėtų testų skaičius
- Metrikos bus sekamos atliekant rankinį testavimą
- Konfigūracijos valdymas - nėra
- Bus testuojama tik pradinė konfigūracija, kurią pateikia kūrėjai
- Bus naudojama stacionarus kompiuteris ir išmanusis telefonas
- Bus naudojama Google Chrome naršyklė abiem atvejais(PC ir telefonas)
- Regresiniai testai - nėra

9. Sėkmės ir nesėkmės kriterijai

Testą laikysime pasisekusiu jeigu įvedus duomenis ar padarius veiksmą programa pateiks korektišką ir tikėtasi rezultatą ar neparodys klaidos pranešimų. Nepasisekusiu testu laikysime jeigu atlikus veiksmą arba įvedus duomenis programos išvestis bus nenumatyta, netiksli, ne tai ko tikėjome arba programa gražins klaidos pranešimą.

10. Testo nutraukimo kriterijai

Testavimą nutraukime tuo atveju jeigu su panašia įvestimi programa gražina tą patį rezultatą. Jeigu to pačio funkcionalumo varijacijoje(pvz. Maisto produktų dėjimas į sąrašą) klaida liks, laikysime, kad testas daugiau informacijos mums nebesuteiks ir testavimą nutrauksime įvertinti savo testavimo atvejams ir tolimesniam testavimui.

11. Testų planai

Planas skirtas rankiniam testavimui juodosios dėžės metodu

12. Likusios užduotys

| Užduotis | Vykdytojas | Būsena |
|---|----------------|----------|
| Sukurti testavimo planą | Matas Savickis | Pabaigta |
| Maisto pridėjimo testavimo atvejai | Matas Savickis | Pabaigta |
| Sporto pridėjimo testavimo atvejai | Matas Savickis | Pabaigta |
| Biometrikos pridėjimo testavimo atvejai | Matas Savickis | Pabaigta |

13. Aplinkos reikalavimai

- Turi veikti www.cronometer.com tinklalapis
- Reikalinga elektra
- Reikalingas interneto ryšys
- Atlikti testavimui reikia išmanioji telefono
- Atlikti testavimui reikia stacionaraus kompiuterio
- Norint pasiekti tinklalapį reikia modernios interneto naršyklės

14. Personalas ir reikalingi apmokymai

‘ Testavimas bus atliekamas programų sistemų studentų. Komandoje iš viso puse vienas testuotojas. Svarbu kad testuotojas mokėtų naudotis kompiuterio pele bei klaviatūra. Taip pat svarbu, kad testuotojas neturėtų motorinių kurie keltų sunkumų naudotis telefono, pele ar klaviatūra. Taip pat yra svarbu, kad testuotojas mokėtų naudoti išmaniaisiais telefonais.

15. Atsakomybės

| Atsakomybė | Matas Savickis |
|---------------------------|----------------|
| Rankinių testų kūrimas | X |
| Rankinių testų vykdymas | X |
| Testavimo plano sudarymas | X |

Už visą testavimo planavimo, sudarymo ir vykdymo procesą bus atskingas neapmokamas programų sistemų studentas Matas Savickis.

16. Tvarkaraštis

Testavimo plano ir testavimo atvejų sudarymas ir testų atlikimas bus įvykdytas pagal gerai žinomą studentišką laiko valdymo metodą pavadinimu „viską padarysiu per vieną naktį prieš atsiskaitymą”. Tai yra yra pats populiariausias darbo pasiskirstymo metodas, o kaip mes visi žinome tai kas populiariu yra gerai.

Darbas bus atliekamas tokiais žingsniais:

1. Testavimo plano sudarymas
2. Testavimo atvejų dokumento sudarymas
3. Testavimo atvejų internetinei versijai sudarymas
4. Testavimo atvejų mobiliam versijai sudarymas
5. Internetinės versijos testų vykdymas
6. Mobilios versijos testų vykdymas
7. Rezultatų ir išvadų sudarymas

17. Rizikos

Testavimas gali būti nesėkmingas dėl testuotojų motyvacijos stokos. Kadangi darbas yra neapmokamas, darbai dažniausiai būna atidedami iki paskutinės minutės, kas gali sukelti problemų laiku spėjant pristatyti atliktus darbus. Rizika gali kirti ir dėl to, nes testuotojas neturės sistemos dokumentacijos ir testo atvejai nebus pilni arba korektiški. Dėl dokumentacijos trūkumo gali nepavykti sugalvoti visų testavimo atvejų.

Testavimo atvejai

18. Įvadas

Šioje dalyje bus pateikti Cronometer internetinės aplikacijos testavimo scenarijai. Šiuo darbu siekiama įsitikinti, kad pagrindiniai aplikacijos funkcionalumai veikia korektiškai tiek naudojantis internetine naršykle tiek mobiliuoju telefonu. Bus testuojamos šios dalys

1. Maisto pridėjimo funkcija internetinėje aplikacijoje
2. Sporto pridėjimo funkcija internetinėje aplikacijoje
3. Biometrinių duomenų pridėjimas internetinėje aplikacijoje
4. Maisto pridėjimo funkcija mobiliojoje aplikacijoje
5. Sporto pridėjimo funkcija mobiliojoje aplikacijoje
6. Biometrinių duomenų pridėjimas mobiliojoje aplikacijoje

19. Testavimo atvejai

| | |
|----------------------------|---|
| Testavimo atvejo numeris | CMT-1.1.1 |
| Testavimo atvejis | Maisto produkto pridėjimas |
| Testavimo atvejo aprašymas | Ištestuoti ar pridedant naują maisto produktą jo kiekis ir maistinės naudos specifikacija teisingai pridedama į bendrą maisto sąrašą |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Atsidarius pagrindiniam langui jame patikriname ar maisto informacija atitinka tą kurią mums nurodė prieš pridedant maistą. |
| Tikėtasis rezultatas | Pridėjus naują maisto produktą pagrindiniame lange maisto informacija atitiks tą kuri buvo parodyta pridėjimo lange |
| Gautas Rezultatas | Maisto produktas pridėtas korektiškai |
| Statusas | Atlikta |

| | |
|----------------------------|---|
| Testavimo atvejo numeris | CMT-1.1.2 |
| Testavimo atvejis | Maisto informacijos atvaizdavimas dideliais kiekiais |
| Testavimo atvejo aprašymas | Ištestuoti ar pridedant naują produktą ir nurodant jam didelius kiekius programa duomenis patetiks ir atvaizduos korektiškai. |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 2147483648g(Int32 MAX + 1) maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Atsidarius pagrindiniam langui jame patikriname ar maisto informacija atitinka tą kurią mums nurodė prieš pridedant maistą. |
| Tikėtasis rezultatas | Programa leis pridėti pasirinktą kiekį maisto, informacija pagrindiniame lange bus pateikta ir nurodyta korektiškai |
| Gautas Rezultatas | Bandant pridėti pasirinkto produkto 2147483648 gramus programa to padaryti neleidžia, kiekis automatiškai nusistato į 999999 gramus. Pagrindiniame lange pridėto produkto informacija iškraipovartotojo sąsajos elementus. |
| Statusas | Nepaėjo |

| | |
|----------------------------|--|
| Testavimo atvejo numeris | CMT-1.1.3 |
| Testavimo atvejis | Maisto produkto pridėjimas pateikiant neskaitinę informaciją |
| Testavimo atvejo aprašymas | Ištestuoti ar bandant pridėti pridėti maisto produktą pateikiant neskaitinį jo kiekio įvedimą(prie kiekio pateikti ne sakaičiu o raides) programa neleis pridėti produkto |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange į kiekio lauką įvedame „test” įvestį 4. Pasižymime kokią maisto informacija mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. |
| Tikėtasis rezultatas | Bandant pridėti maistą nurodant jo kiekį tekstu programa turi parodyti mums informacinį pranešimą ir neleisti mums pridėti maisto kol įvestis nebus pakeista į skaitinę įvestį |
| Gautas Rezultatas | Bandant pridėti maistą nurodant tekstinį kiekį programa atpažysta, kad įvestis yra nekorektiška ir paryškina įvesties lauką raudonai, tačiau bandant patvirtinti pridėjimą paspaudus „Add Serving” programa vistiek prideda maitą prie sąrašo nurodydama, kad jo pridėjome 1 gramo kiekio. |
| Statusas | Nepraėjo |

| | |
|---|---|
| Testavimo atvejo numeris | CMT-1.1.4 |
| Testavimo atvejis | Maisto produktas nedingsta iš sąrašo perkrovus svetainę |
| Testavimo atvejo aprašymas | Ištestuoti ar perkrovus svetainę maisto sąrašo informacija lieka ta pati |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informacija mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Atsidarius pagrindiniam langui jame patikriname ar maisto informacija atitinka tą kurią mums nurodė prieš pridedant maistą. |
| 7. Perkrauname puslapį Tikėtasis rezultatas | Perkrovus svetainę maisto informacija pagrindiniame saraše maisto informacija nepasikeis |
| Gautas Rezultatas | Maisto informacija po perkrovimo pateikiama korektiškai |
| Statusas | Atlikta |

| | |
|---|---|
| Testavimo atvejo numeris | CMT-1.1.5 |
| Testavimo atvejis | Maisto produktų galime pridėti tiek kiek norime |
| Testavimo atvejo aprašymas | Ištestuoti ar galima pridėti daugiau negu 100 skirtingų maisto vienetų į sąrašą ir pridėjus šiuos produktus informacija bus pateikta korektiškai |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Atsidarius pagrindiniam langui jame patikriname ar maisto informacija atitinka tą kurią mums nurodė prieš pridedant maistą. |
| 7. Kartojame pirmjus 6 žingsnius 101 kartą Tikėtasis rezultatas | Pridėjus daugiau negu 100 skirtingų maisto tipų informacija bus pateikta korektiškai tiek atvaizdavimo atžvilgiu tiek ir informacijos korektiškumu. |
| Gautas Rezultatas | Pridėjus 101 maisto vienetą informacija sąrašą pateikta ir nurodoma korektiškai |
| Statusas | Atlikta |

| | |
|----------------------------|---|
| Testavimo atvejo numeris | CMT-1.1.6 |
| Testavimo atvejis | Galime pridėti tą patį maisto produktą |
| Testavimo atvejo aprašymas | Ištestuoti ar pridedant tą patį maisto produktą prie sąrašo naują įrašą rodo kaip atskirą elementą sąrašą |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 7. Iš pateikto sąrašo pasirenkame tą patį produktą kurį rinkomės 2 veiksmuose. 8. Atsidariusiame lange nurodome 100g maisto kiekio. 9. Pasižymime kokią maisto informaciją mums pateikta. 10. Spaudžiame mygtuką „Add Serving”. 11. Pagrindiniame lange patikriname ar tas pats maistas sąrašą buvo pridėtas kaip atskiras elementas |
| Tikėtasis rezultatas | Du kartus pridėjus tą patį maistą pagrindiniame sąrašą bus atvaizduoti du atskiri maisto elementai |
| Gautas Rezultatas | Pridėjus du identiškus maisto elementus į sąrašą jie sąrašą buvo atvaizduoti kaip du atskiri elementai |
| Statusas | Atlikta |

| | |
|----------------------------|--|
| Testavimo atvejo numeris | CMT-1.1.7 |
| Testavimo atvejis | Maisto kiekio keitimas pagrindiniame lange |
| Testavimo atvejo aprašymas | Ištestuoti ar pakeitus maisto kiekį pagrindiniame lange jis korektiškai pakeičiamas ir bendra informacija pasikeičia korektiškai |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Pagrindiniame lange paspaudžiam ant sąrašo elemento maisto kiekio skaičiaus 7. Pakeičiame elemento kiekį į 200 gramų 8. Patikriname ar maisto elemento informacija pasikeitė korektiškai 9. Patikriname ar bendra maisto informacija pasikeitė korektiškai |
| Tikėtasis rezultatas | Pakeitus vieno iš sąrašo elementų kiekį jis bus korektiškai pakeistas tiek sąrašo elemento informacijoje tiek ir bendroje maisto informacijoje |
| Gautas Rezultatas | Pakeitus vieno iš sąrašo elementų kiekį visa informacija pasikeitė korektiškai |
| Statusas | Atlikta |

| | |
|----------------------------|--|
| Testavimo atvejo numeris | CMT-1.1.8 |
| Testavimo atvejis | Maisto kalorijų keitimas pagrindiniame lange |
| Testavimo atvejo aprašymas | Ištestuoti ar pakeitus maisto kalorijas pagrindiniame lange jis korektiškai pakeičiamas ir bendra informacija pasikeičia korektiškai |
| Testavimo eiga | 1. Pagrindiniame programos lange spaudžiame mygtuką „ADD FOOD”. 2. Iš pateikto sąrašo pasirenkame vieną iš produktų. 3. Atsidariusiame lange nurodome 100g maisto kiekio. 4. Pasižymime kokią maisto informaciją mums pateikta. 5. Spaudžiame mygtuką „Add Serving”. 6. Pagrindiniame lange paspaudžiam ant sąrašo elemento maisto kalorijų skaičiaus 7. Pakeičiame elemento kalorijas pakeičiame į 100 kalorijų 8. Patikriname ar maisto elemento informacija pasikeitė korektiškai 9. Patikriname ar bendra maisto informacija pasikeitė korektiškai |
| Tikėtasis rezultatas | Pakeitus vieno iš sąrašo elementų kalorijas jis bus korektiškai pakeistas tiek sąrašo elemento informacijoje tiek ir bendroje maisto informacijoje |
| Gautas Rezultatas | Pakeitus vieno iš sąrašo elementų kalorijas visa informacija pasikeitė korektiškai |
| Statusas | Atlikta |

20. Defektų aprašas

| Nr | Funkcionalumas | Aprašymas | Sunkumas |
|-----------|--|---|----------|
| CMT-1.1.2 | Maisto informacijos atvaizdavimas dideliais kiekiais | Bandant pridėti maisto kiekius didesnius negu 999999 gramų programa pasirinktą kiekį automatiškai padaro lygiu 999999g. Dėl per didelių skaičių būna iškraipomas programos pagrindinis langas ir informacija parodoma nekorektiškai | Lengvas |
| CMT-1.1.3 | Maisto pridėjimas nurodant nekorektišką įvestį | Bandant pridėti maistą su tekstine kiekio įvestimi programa leidžia pridėti maistą automatiškai nurodydama 1 gramą kiekio | Lengvas |