**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**PROGRAMŲ SISTEMŲ TESTAVIMAS (T120B162)**

**Testavimo planas**

Atliko: IFF – 4/3 gr. studentas

Dovydas Petrutis

**KAUNAS 2017**

Turinys

[Turinys 2](#_Toc499306426)

[1. Įvadas 3](#_Toc499306427)

[2. Tikslas 3](#_Toc499306428)

[3. Testų apimtis 3](#_Toc499306429)

[4. Reikalavimai prieš testų atlikimą 3](#_Toc499306430)

[5. TEstavimo prioritetai 3](#_Toc499306431)

[6. Testavimo technika 3](#_Toc499306432)

[7. Rezultatai 4](#_Toc499306433)

[8. Testavimo aplinka 4](#_Toc499306434)

[9. Testavimo scenarijai 4](#_Toc499306435)

[10. Testai 4](#_Toc499306436)

[10.1 CustomerController testai 4](#_Toc499306437)

[10.1.1 Testas 1 4](#_Toc499306438)

[10.1.2 Testas 2 4](#_Toc499306439)

[10.1.3 Testas 3 5](#_Toc499306440)

[10.1.4 Testas 4 5](#_Toc499306441)

[10.2 Customers API testai 5](#_Toc499306442)

[10.2.1 Testas 1 5](#_Toc499306443)

[10.2.2 Testas 2 5](#_Toc499306444)

[10.2.3 Testas 3 6](#_Toc499306445)

[10.2.4 Testas 4 6](#_Toc499306446)

[10.2.5 Testas 5 6](#_Toc499306447)

[11. Išvados 7](#_Toc499306448)

# Įvadas

Testavimui pasirinkau paties parašyta programėlę naudojantis C#, kuri naudojama filmų nuomos punkto svetainėje. Pasirinkau ištestuoti klientų posistemę, Ji atsakinga už veiksmus su klientais. API repozitorija: <https://github.com/DovPet/Testavimas>

Testavimo strategija:

* Integraciniai ir unit testai bus naudojami testuoti programos veikimo tikslumą.
* Statinė analizė – kodo stiliaus neatitikimams ar loginių klaidų paieškai.
* Issue tracking sistema github bus naudojama pranešti apie programoje rastas klaidas.

# Tikslas

Norima įsitikinti, kad programa veikia, taip kaip numatyta. Taip pat neturi klaidų ir kad keičiantis kodui būtų išsaugomas numatytas funkcionalumas.

# Testų apimtis

Integraciniai testai – projekto repozitorijos sluoksnio ir atvejų testai:

* Užtikrinti, kad CustomerController veikia gerai
* Užtikrinti, kad Api.CustomerController veikia teisingai, grąžina reikiamus pranešimus

Unit testai – pavienės funkcijos, kurios atsakingos už tikslias patikrinimo operacijas:

# Reikalavimai prieš testų atlikimą

Kriterijai, kurie turi būti įvykdyti prieš pradedant testavimą:

1. Kodas turi kompiliuotis ir veikti.
2. Programos elgesys ir funkcionalumas turi būti aiškus.
3. Turi būti paruošta testavimo aplinka.
4. Surastas statinės analizės įrankis.
5. Surastas issue tracking įrankis.
6. Surastas unit ir integracinių testų testavimo karkasas.

# TEstavimo prioritetai

1. Unit testai svarbioms funkcijoms testuoti.
2. Integraciniai testai – sistemos testavimui.
3. Statiniai testai kodo stiliui ir vientisumui išlaikyti.
4. Issue iškėlimas radus klaidų kode.

# Testavimo technika

Naudojama testavimo technika:

1. Testavimo scenarijai su įvedamais duomenimis ir norimu gauti rezultatu.
2. Statinė analizė klaidų aptikimui prieš programos paleidimą.
3. Issue kėlimas į sistemą atsiradus klaidoms.

# Rezultatai

1. Unit ir integracinių testų dokumentas su testų ir panaudotų įrankių aprašymais.
2. Statinės analizės dokumentas su aptiktomis klaidomis, taisyklių ir įrankio aprašymu.
3. Aptiktų klaidų dokumentacija pasinaudojus issue tracking sistema.

# Testavimo aplinka

Testavimo aplinka turi turėti tokius minimalius reikalavimus:

1. Turi būti įdiegtas .net 4.5 framework paketas.
2. Bent 500MB operatyviosios aplinkos.
3. Bent 500MB laisvos vietos diske.

# Testavimo scenarijai

Kiekvienas testavimo scenarijus susideda iš:

1. Testavimo sritis ir aprašymas.
2. Testavimo žingsnių.
3. Įvedimo duomenų.
4. Tikimosi rezultato.
5. Gauto rezultato

# Testai

## CustomerController testai

### Testas 1

1. Testas patikrina ar metodas “Details” grąžina sukurto kliento duomenis

|  |  |
| --- | --- |
| 1. Testavimo žingsniai | 1. Duomenys |
| Sukurti klientą ir jį paimti iš atmintyje laikomos duomenų bazės ir patikrinti ar jo duomenis. | Customer objektas kurį sudaro teisingai duomenys |

1. Tikėtinas rezultatas:

Sukurtas klientas randamas duomenų bazėje ir jo duomenys atitinka duomenis kurių tikimasi.

1. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 2

1. Testas patikrina ar pagrindinis klientų puslapis grąžina klientų sarašo objektą

|  |  |
| --- | --- |
| 1. Testavimo žingsniai | 1. Duomenys |
| Sukurti vartotojus ir juos peržiūrėti | Sąrašas sudarytas ir customer ojbektų |

1. Tikėtinas rezultatas:

Klientai atvaizduojami o jų tipas klientų sarašas.

1. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 3

1. Testas patikrina ar buvo atnaujinti kliento duomenys.

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Sukurti vartotoją ir redaguoti jo duomenys | Customer objektas kurį sudaro su teisingais duomenimis. |

1. Tikėtinas rezultatas:

Sukurtas klientas, jį redaguojant, duomenų bazėje pasikeičia jo duomenys

1. Gautas rezultatas:

Toks koks ir tikėtasi.

## Customers API testai

### Testas 1

1. Testas patikrinti ar gaunamas teisingas rezultatas, kai bandomas gauti /api/customers

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Sukurti klientą  Bandyti gauti jo api duomenis | Customer objektas kurį sudaro su teisingais duomenimis. |

1. Tikėtinas rezultatas:

Sukurtas klientas, bandant gauti jo duomenis gaunamas OK http request pranšimas

1. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 2

1.Testas patikrinti ar gaunamas teisingas rezultatas, kai bandomas gauni /api/customer/{id}

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Sukurti klientą  Bandyti gauti jo api duomenis | Customer objektas kurį sudaro su teisingais duomenimis. |

4.Tikėtinas rezultatas:

Sukurtas klientas, bandant gauti jo duomenis gaunamas OK http request pranšimas

5. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 3

1.Testas patikrinti ar gaunamas klaidingas pranešimas, kai bandoma gauti /api/customer/{id} su vienu id bet tikimasi kitokių rezultatų. (Pvz po duomenų atnaujinimo, patikrinti ar jie atsinaujino)

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Sukurti klientą  Bandyti gauti jo api duomenis | Customer objektas kurį sudaro su teisingais duomenimis. Taip pat kitas objektas, kurio kitokie duomenys. |

4.Tikėtinas rezultatas:

Tikrinant kliento duomenis gaunamas http request pranešimas not found

5. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 4

1.Testas patikrinti ar gaunamas geras pranešimas, kai bandoma atnaujinti kažkokio kliento duomenis

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Atnaujinti kliento duomenis, suvesti juos teisingai. | Customer objektas kurį sudaro su teisingais duomenimis. |

4.Tikėtinas rezultatas:

Atnaujinami kliento duomenys ir gaunamas OK pranešimas

5. Gautas rezultatas:

Toks koks ir tikėtasi.

### Testas 5

1.Testas patikrinti ar gaunamas geras pranešimas, kai yra ištrinimas klientas.

|  |  |
| --- | --- |
| 2. Testavimo žingsniai | 3.Duomenys |
| Ištrinti egzistuojantį klientą | Reikalinga, kad būtų sukurtas nors vienas klientas. |

4.Tikėtinas rezultatas:

Ištrynus klientą gaunamas OK pranešimas

5. Gautas rezultatas:

Toks koks ir tikėtasi.

# Išvados

Testai labai reikalingi norint turėti patikimą ir stabilią sistemą. Jie padeda sugaudyti nepageidaujamas klaidas prieš joms pasiekiant galutinį vartotoją. Testai ypač padeda kai programos kūrimas yra labai spartus. Naudojant testų paleidimų automatizavimą kiekvieną kartą kažką pridėjus sistemoje galima labai padidinti sistemos patikimumą, nes vartotojai išvengs ištestuotų klaidų. Testavimo planas yra dokumentas, kuris užtikrina, kad testų tipai ir testuojamas funkcionalumas būtų įgyvendinti.