**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**PROGRAMŲ SISTEMŲ TESTAVIMAS (T120B162)**

**Testavimo planas**

Atliko: IFF – 4/3 gr. studentas

Rokas Vaitkevičius

**KAUNAS 2017**

Turinys

[Turinys 2](#_Toc493607719)

[1. Įvadas 3](#_Toc493607720)

[2. Tikslas 3](#_Toc493607721)

[3. Testų apimtis 3](#_Toc493607722)

[4. Reikalavimai prieš testų atlikimą 3](#_Toc493607723)

[5. Testavimo prioritetai 3](#_Toc493607724)

[6. Testavimo technika 4](#_Toc493607725)

[7. Rezultatai 4](#_Toc493607726)

[8. Testavimo aplinka 4](#_Toc493607727)

[9. Testavimo scenarijai 4](#_Toc493607728)

[10. Testai 4](#_Toc493607729)

[10.1 EventCases testai 4](#_Toc493607730)

[10.1.1 Testas 1: ItFindsCreatedEvent\_WhenEventIsCreated 4](#_Toc493607731)

[10.1.2 Testas 2: ItHasTheSameData\_WhenEventIsCreated 5](#_Toc493607732)

[10.1.3 Testas 3: ItThrowsArgumentNullException\_WhenEventIsCreated 5](#_Toc493607733)

[10.1.4 Testas 4: ItHasUpdatedData\_WhenEventIsUpdated 5](#_Toc493607734)

[10.1.5 Testas 5: ItsFieldCancelHasChanged\_WhenMethodIsCalled 5](#_Toc493607735)

[10.1.6 Testas 6: ItResolve\_IEventsService 6](#_Toc493607736)

[10.2 SportType testai 6](#_Toc493607737)

[10.2.1 Testas 1: ItFinds\_CreatedSportTypes 6](#_Toc493607738)

[10.2.2 Testas 2: ItResolve\_ISportTypesService 6](#_Toc493607739)

[10.3 Mappers testai 6](#_Toc493607740)

[10.3.1 Testas 1: ItChecksIf\_EventPocoIsMappedToDtoCorrectly 6](#_Toc493607741)

[10.3.2 Testas 2: ItChecksIf\_SportTypePocoIsMappedToDtoCorrectly 7](#_Toc493607742)

[10.3.3 Testas 3: ItChecksIf\_UserPocoIsMappedToDtoCorrectly 7](#_Toc493607743)

[10.3.4 Testas 4: ItChecksIf\_VoterIsMappedToDtoCorrectly 7](#_Toc493607744)

[11. Išvados 7](#_Toc493607745)

# Įvadas

Testavimui pasirinkau paties parašyta API implementaciją naudojantis C#, kuri naudojama sporto renginių kūrimo, redagavimo, trynimo sistemoje. Ji saugo duomenis apie renginius ir juos pateikia atvaizduoti UI. API repozitorija: (<https://github.com/RokasVaitkevicius/Sport-Events>) .

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 EventCases veikia teisingai
* Užtikrinti, kad SportType yra sukuriami vos paleidus programą

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

* Įsitikinti, kad Mapperiai veikia teisingai

# 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 core framework 2.0 SDK paketas.
2. Turi būti įdiegta Windows 10 operacinė sistema
3. Bent 200MB operatyviosios aplinkos.
4. Bent 50MB laisvos vietos diske.

# Testavimo scenarijai

Kiekvienas testavimo scenarijus susideda iš:

1. Aprašymo, ką testas testuoja.
2. Testavimo žingsnių.
3. Įvedimo duomenų ir rezultatų.

# Testai

## EventCases testai

### Testas 1: ItFindsCreatedEvent\_WhenEventIsCreated

1. Testas patikrina ar buvo sukurtas renginys.
2. Duomenų kolekcijoje turi būti bent vieno renginio duomenys. Testo metu sukuriama duomenų bazė atmintyje. Sukurtas renginys, pagal jo id yra paimamas iš atmintyje laikomos duomenų bazės ir patikrinama ar jis egzistuoja.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventsValidationCollection – renginių kolekcija. | Egzistuojantis duomenų bazės objektas. |

### Testas 2: ItHasTheSameData\_WhenEventIsCreated

1. Testas patikrina ar teisingai užpildomi visi duomenų laukai.
2. Duomenų kolekcijoje turi būti bent vieno renginio duomenys. Testo metu sukuriama duomenų bazė atmintyje. Sukurtas renginys, pagal jo id yra paimamas iš atmintyje laikomos duomenų bazės ir sulyginami visi jo laukai su gautais duomenimis ir turimamais duomenimis kolekcijoje.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventsValidationCollection – renginių kolekcija. | Renginio objektas su teisingais duomenimis. |

### Testas 3: ItThrowsArgumentNullException\_WhenEventIsCreated

1. Testas patikrina ar pasirodo sistema iškelia klaidos pranešimą, kai bent vienas iš laukų yra neteisingas.
2. Testo duomenų kolekcijoje turėtų būti užtektinai duomenų ištestuoti ar yra visų laukų validacija. Pvz.: viename duomenų rinkinyje gali būti neteisingas pavadinimas, kitame neteisingas renginio laikas ir pan. Testas turi bandyti sukurti renginį atmintyje laikomoje duomenų bazėje ir tikėtis kad bus iškeltas klaidos pranešimas.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventsInvalidDataCollection – renginių kolekcija kurios bent vienas laukas yra neteisingas. | Sistema iškels klaidos pranešimą dėl neteisingo lauko. |

### Testas 4: ItHasUpdatedData\_WhenEventIsUpdated

1. Testas patikrina ar atnaujinus renginio duomenis, duomenys atsinaujina
2. Testo duomenų kolekciją turi sudaryti duomenys naujam renginiui sukurti ir duomenys tam renginiui atnaujinti. Testo metu turi būti sukurtas renginys atmintyje laikomoje duomenų bazėje, paskui turi būti tas įrašas atnaujinamas ir paimamas iš duomenų bazės. Tik tada sulyginami gauti duomenys iš atminties ir turim duomenys kolekcijoje.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventsUpdateDataCollection – renginių kolekcija kurioje yra du objektai, vienas pradinių duomenų, kitas atnaujinamų duomenų. | Renginio duomenys po atnaujinimo turi teisingus duomenis. |

### Testas 5: ItsFieldCancelHasChanged\_WhenMethodIsCalled

1. Testas patikrins ar renginio būsena pasikeitė iškvietus tam skirtą metodą.
2. Testui galima naudoti duomenų kolekciją naudotą renginio kūrimo testo metu. Šis testas patikrins ar kai kviečiama funkcija „ChangeEventState()” duomenų bazėje pasikeičia laukelio canceled reikšmė. Testo metu sukuriamas įrašas, kviečiama aprašyta funkcija ir tikrinama ar laukelio reikšmė pasikeitė nuo tos, kuri pateikta duomenų kolekcijoje.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventsValidationCollection – renginių kolekcija. | Renginio laukelis cancelled po funckijos iškvietimo pasikeitė. |

### Testas 6: ItResolve\_IEventsService

1. Testas patikrins ar teisingai injectintas interfeisas.
2. Testui duomenų nereikia. Testo metu užtenka iškviesti pagalbinę klasę, kuri iškvies DI, tada patikrinti ar EventCases egzistuoja.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
|  | Teisingai inject‘intas EventsCases interfeisas. |

## SportType testai

### Testas 1: ItFinds\_CreatedSportTypes

1. Testas patikrins ar paleidus sistemą sukuriamas bent vienas sporto tipas.
2. Testui duomenų nereikia, nes duomenis duomenų bazėje sistema turi sukurti automatiškai. Testo metu paleidus sistemą tikimasi kad duomenų bazėje atsiras bent vienas sporto tipų įrašas.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
|  | Egzistuoja bent vienas renginio tipas. |

### Testas 2: ItResolve\_ISportTypesService

1. Testas patikrins ar teisingai injectintas interfeisas.
2. Testui duomenų nereikia. Testo metu užtenka iškviesti pagalbinę klasę, kuri iškvies DI, tada patikrinti ar SportTypeCases egzistuoja.

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
|  | Teisingai inject‘intas SportTypesCases interfeisas. |

## Mappers testai

### Testas 1: ItChecksIf\_EventPocoIsMappedToDtoCorrectly

1. Testas patikrins ar teisingai veikia renginių mapper’is.
2. Duomenų rinkinyje turi būti renginio objekto POCO duomenys. Testo metu paimamas renginių mapperis ir sumapinami duomenys iš POCO į DTO. Tada patikriname ar duomenys dto atitinka duomenis POCO

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| EventPocoToDtoDataCollection – renginio poco duomenis, kurie bus lyginami su gautu po mappinimo dto. | Mapperis teisingai sumappin‘o duomenis iš poco į dto. |

### Testas 2: ItChecksIf\_SportTypePocoIsMappedToDtoCorrectly

1. Testas patikrins ar teisingai veikia renginių sporto tipų mapper’is.
2. Duomenų rinkinyje turi būti sporto tipo objekto POCO duomenys. Testo metu paimamas sporto tipų mapperis ir sumapinami duomenys iš POCO į DTO. Tada patikriname ar duomenys dto atitinka duomenis POCO

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| SportyTypePocoToDtoDataCollection – renginio poco duomenis, kurie bus lyginami su gautu po mappinimo dto. | Mapperis teisingai sumappin‘o duomenis iš poco į dto. |

### Testas 3: ItChecksIf\_UserPocoIsMappedToDtoCorrectly

1. Testas patikrins ar teisingai veikia vartotojų mapper’is.
2. Duomenų rinkinyje turi būti vartotojo objekto POCO duomenys. Testo metu paimamas vartotojų mapperis ir sumapinami duomenys iš POCO į DTO. Tada patikriname ar duomenys dto atitinka duomenis POCO

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| UserPocoToDtoDataCollection – renginio poco duomenis, kurie bus lyginami su gautu po mappinimo dto. | Mapperis teisingai sumappin‘o duomenis iš dto į poco. |

### Testas 4: ItChecksIf\_VoterIsMappedToDtoCorrectly

1. Testas patikrins ar teisingai veikia balsuotojų mapper’is.
2. Duomenų rinkinyje turi būti balsuotojo objekto POCO duomenys. Testo metu paimamas balsuotojų mapperis ir sumapinami duomenys iš POCO į DTO. Tada patikriname ar duomenys dto atitinka duomenis POCO

|  |  |
| --- | --- |
| Duomenys | Rezultatai |
| VoterPocoToDtoDataCollection – renginio poco duomenis, kurie bus lyginami su gautu po mappinimo dto. | Mapperis teisingai sumappin‘o duomenis iš dto į poco. |

# 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.