



IEGULDĪJUMS TAVĀ NĀKOTNĒ!

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

Projekts: "Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem"

Projekta sākuma datums: 2014.gada 1.janvāris.

Projekta beigu datums: 2015.gada 30.jūnijs.

Līguma Nr. 2013/0031/2DP/2.1.1.1.0/13/APIA/VIAA/010

ESF finansējuma saņēmējs: SIA, SWH SETS

Sadarbības partneris: Elektronikas un datorzinātņu institūts (EDI)

Projekta aktivitātes Nr. 5.6 "Realizētās programmatūras funkcionāla pārbaude" gala pārskats

Pārskats Nr. 39 par periodu no 2015.gada 1.janvāra līdz 2015.gada 30.jūnijam.

SATURS

1.	Kopsavilkums	3
2.	Ievads	4
3.	Funkcionāla pārbaude	5
3.1.	Paraugprojekta funkcionālās prasības	5
3.2.	Testu izstrāde	5
3.2.1.	Automatizētie testi	6
3.2.2.	Manuālie testi	6
4.	Secinājumi	7
5.	Literatūras saraksts	8
6.	Pielikumi	9
6.1.	Testu ģenerācijas transformācijas piemērs	9
6.2.	Noģenerēto testu piemērs	10
6.3.	Manuālā testa piemērs	44

1. Kopsavilkums

Pārskata periodā (2015-01-01 – 2015-06-30) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes "Realizētās programmatūras funkcionāla pārbaude" ietvaros veikti šādi darbi:

1. Automatizēto testu izstrāde;
2. Veikta funkcionālā pārbaude, izmantojot gan automātiskos, gan manuālos testus.

2. Ievads

Šis pārskats ir veltīts projekta apakšaktivitātes Nr. 5.6 "Realizētās programmatūras funkcionāla pārbaude" progresā ietvaros veiktajiem darbiem.

Funkcionālā testēšana ir viens no programmatūras izstrādes standarta posmiem. Tās mērķis ir pārbaudīt izstrādātas sistēmas atbilstību funkcionālām prasībām, aprakstītām specifikācijā.

3. Funkcionāla pārbaude

3.1. Paraugprojekta funkcionālās prasības

Praugprojekta specifikācijā ir sekojošās funkcionālās prasības [1]:

1. Lietotāju vadības funkcija
2. Lietotāju lomas
3. Datu objektu caurskatīšana, pievienošana, rediģēšana un dzēšana
4. Datu importa funkciju definēšana
5. Datu apstrādes funkciju definēšana
6. Datu eksporta funkciju definēšana
7. Projektu vadība (izveidošana, papildināšana, dzēšana)
8. Datu imports uz projektu
9. Datu kopēšana no cita projekta
10. Kategoriju definēšana
11. Lauka datu imports un vizualizācija
12. Kategoriju apgabalu atzīmēšana attēlā
13. Datu apstrāde projekta ietvaros
14. Datu eksports
15. Drošība - parole jāglabā šifrētā veidā.
16. Kļūdu žurnāls
17. Auditēšana pieraksti
18. Lietotāja saskarnes valoda

Katra prasība tika detalizēti aprakstīta specifikācijā un tika realizēta kā viena vai vairākas funkcijas.

3.2. Testu izstrāde

Funkcionālā pārbaude ir melnās kastes testēšana. Tiek testētas funkcijas ar ieejas datiem, un tiek pārbaudīts izejas rezultāts, neņemot vērā programmas iekšējo struktūru. Funkcionālo pārbaudi veic, sastādot testa piemēru (vai vairākus testa piemērus) atbilstoši katrai prasībai. Testa piemēram apraksta ieejas datus un gaidāmo rezultātu, uztaisa testa plānu, kas organizē testa piemērus.

Visas funkcionālās prasības tika sadalītas 3 grupās:

1. standarta funkcijas darbam ar biznesa objektiem (objekta izveidošana, nolasīšana, atjaunošana, dzēšana);
2. vispārīgas programmatūras funkcijas, tādas kā paroles šifrēšana, lietotāju pārvaldība, lietotāju lomas pielietošana, kļūdu žurnāls, auditēšana;
3. funkcijas, kuras nosaka pielietošanas apgabals, biznesa atkarīgas funkcijas (piem., Datu importa funkciju definēšana vai Lauka datu imports un vizualizācija)

Tādā veidā var nodrošināt paralēlo un neatkarīgo katras grupas testēšanu.

Visefektīvākais un stabilākais veids, kā pārbaudīt prasības, ir automātisko testu izmantošana. Automātiskais tests ir testa piemērs, kas ir vai nu atsevišķā programma, vai scenārija apraksts, kuru spēj interpretēt kaut kāds rīks.

Protams, pastāv prasības, kuras var tikt pārbaudītas tikai manuāli, imitējot lietotāja darbības ar sistēmu. Tādām prasībām var sastādīt tikai manuālus testus, kurus izpildīs ar rokām.

3.2.1. Automatizētie testi

Testu automatizēšanai tika izmantoti NUnit testi, kas ir Microsoft infrastruktūra testu izpildei un apkopošanai. Katrs tests te ir C# procedūra, kuru var izpildīt gan atsevišķi, gan kopā ar citiem testiem (procedūrām).

Tāda testu automatizēšana tika pielietota pirmās grupas prasību (standartmanipulācijas ar modeļa biznesa objektiem) testiem. Tā kā šīs prasības realizē MEDUS ietvars, testi, kas pārbauda šo daļu, ir vienveidīgi un atkarīgi no modeļa satura. Līdz ar to, tādus testus var arī automātiski ģenerēt no modeļa. Daļa no otras grupas prasībām arī tika pārbaudītas, izmantojot automātiskus testus. Pielikumā 1. ir attiecīgas transformācijas piemērs un pielikumā 2. ir noģenerētās testu procedūras piemērs.

3.2.2. Manuālie testi

Pārējās funkcionālās prasības tika pārbaudītas manuāli. Manuālus testus var sadalīt grupās:

- Lietotāja saskarnes testi – funkcionālās pārbaudes ietvaros lietotāja saskarne mūs interesē tikai no funkciju izsaukšanas iespēju viedokļa, citiem vārdiem vizuāla izskata konsekvence attiecībā pret katra biznesa objekta tipu.
- Integrēto funkciju palaišana – sanāk pārāk sarežģītas programmas, kurām pašām par sevi varētu būt nepieciešama testēšana, kā arī iekļauj darbības ar saskarni (piemērs pielikumā 3.)
- Pārējie testi – citas biznesa prasības, piemēram kļūdu apstrādes testi.

Protams, pastāv automatizēšanas rīki, kas ļauj vairāk vai mazāk automatizēt arī šos testus, bet ņemot vērā paraugprojekta specifiku un testu automatizēšanai nepieciešamo darba apjomu, šī iespēja netika attīstīta šī projekta ietvaros.

4. Secinājumi

Aktivitātes ietvaros tika izstrādāti testi paraugprojekta funkcionālai pārbaudei. Kā arī tika veikta testu izpilde un konstatēta izstrādātās sistēmas atbilstība prasību specifikācijā izvirzītām prasībām.

5. Literatūras saraksts

[1] Nr.5.1 "Prasību specifikācijas izstrāde";

6. Pielikumi

6.1. Testu ģenerācijas transformācijas piemērs

Transformācija ir bāzēta uz biznesa objektu tehnoloģiskā modeļa (*DataViews*)

```
<#@ include file="../../../../_TT/DataViews_dll.ttinclude" once="true" #>
<#@ output extension=".cs" #>
<#@ Assembly Name="System.Core" #>
<#@ Assembly Name="System.Windows.Forms" #>
<#@ Assembly Name="$(ProjectDir)/../../../../_bin/MEDUS.dll" #>
<#@ Assembly Name="$(ProjectDir)/../../../../_bin/DataViews.dll" #>
<#@ import namespace="System" #>
<#@ import namespace="System.IO" #>
<#@ import namespace="System.Diagnostics" #>
<#@ import namespace="System.Linq" #>
<#@ import namespace="System.Collections" #>
<#@ import namespace="System.Collections.Generic" #>
<#@ import namespace="MEDUS" #>
<#@ import namespace="DataViews" #>
<#
```

```
MDSmo_DataViews model =
new MDSmo_DataViews(loadXmodel("DataViews.xml", getLUpath()),
    MEDUS.Model.ExplorePolicy.ignore);
```

```
#>
using System;
using System.Linq;
using Microsoft.VisualStudio.TestTools.UnitTesting;
//using MDSDB_DUS;
using MEDUSP;
using MEDUSR;
using MEDUSBL;

namespace Protons.Tests
{
    [TestClass]
    public class Selects
    {
        <# foreach (DataView v in model.MDSfind_DataView()){ #>
            [TestMethod]
            public void select<#=# v.name #>()
            {
                IAuth am = new LocalAuthManager();
                am.Login("user", "user100", false);
                var cc = (new <#=# v.name
#>()).startSelect().Take(10).ToList();
                MEDUSContext tc = new MEDUSContext();
                var cc1 = (new <#=# v.name
#>()).startSelect(tc).Take(10).ToList();
                am.Logout();
            }
        }
    }
}
```

```
    }  
    <# } #>  
    }  
}
```

6.2. Noģenerēto testu piemērs

Elementārās atlasīšanas testi katram biznesa objektu tipam

```
using System;  
using System.Linq;  
using Microsoft.VisualStudio.TestTools.UnitTesting;  
//using MDSDB_DUS;  
using MEDUSP;  
using MEDUSR;  
using MEDUSBL;  
  
namespace Protons.Tests  
{  
    [TestClass]  
    public class Selects  
    {  
        [TestMethod]  
        public void selectProject_B_F()  
        {  
            IAuth am = new LocalAuthManager();  
            am.Login("user", "user100", false);  
            var cc = (new  
Project_B_F()).startSelect().Take(10).ToList();  
            MEDUSContext tc = new MEDUSContext();  
            var cc1 = (new  
Project_B_F()).startSelect(tc).Take(10).ToList();  
            am.Logout();  
        }  
        [TestMethod]  
        public void selectProject_B_R()  
        {  
            IAuth am = new LocalAuthManager();  
            am.Login("user", "user100", false);  
            var cc = (new  
Project_B_R()).startSelect().Take(10).ToList();  
            MEDUSContext tc = new MEDUSContext();  
            var cc1 = (new  
Project_B_R()).startSelect(tc).Take(10).ToList();  
            am.Logout();  
        }  
        [TestMethod]  
        public void selectProject_N_F()  
        {  
            IAuth am = new LocalAuthManager();
```

```

        am.Login("user", "user100", false);
        var cc = (new
Project_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
Project_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProject_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
Project_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
Project_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProject_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
Project_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
Project_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprojectData_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprojectData_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprojectData_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprojectData_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprojectData_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprojectData_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprojectData_N_F()

```

```

    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprojectData_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprojectData_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
[TestMethod]
public void selectRSprojectData_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
RSprojectData_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
RSprojectData_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectRSprojectData_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
RSprojectData_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
RSprojectData_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategory_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
Category_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
Category_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategory_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
Category_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
Category_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
}

```

```
[TestMethod]
public void selectCategory_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
Category_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
Category_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategory_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
Category_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
Category_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategory_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
Category_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
Category_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectFieldData_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
FieldData_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
FieldData_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectFieldData_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
FieldData_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
FieldData_B_R()).startSelect(tc).Take(10).ToList();
}
```

```
        am.Logout();
    }
    [TestMethod]
    public void selectFieldData_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
FieldData_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
FieldData_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectFieldData_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
FieldData_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
FieldData_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectFieldData_N_P()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
FieldData_N_P()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
FieldData_N_P()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectFieldData_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
FieldData_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
FieldData_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectFieldData_X_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
FieldData_X_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
```

```

        var cc1 = (new
FieldData_X_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLSensorType_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLSensorType_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLSensorType_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLSensorType_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLSensorType_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLSensorType_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLSensorType_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLSensorType_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLSensorType_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLSensorType_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLSensorType_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLSensorType_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLSensorType_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
    }

```

```

        var cc = (new
CLSensoryType_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLSensoryType_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLCategory_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLCategory_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLCategory_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLCategory_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLCategory_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLCategory_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLCategory_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLCategory_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLCategory_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLCategory_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLCategory_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLCategory_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLCategory_N_Filter()
    {

```



```

        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLCategory_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLCategory_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLColor_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLColor_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLColor_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLColor_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLColor_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLColor_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLColor_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLColor_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLColor_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCLColor_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLColor_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLColor_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```

```
public void selectCLColor_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CLColor_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CLColor_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCLFileType_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CLFileType_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CLFileType_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCLFileType_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CLFileType_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CLFileType_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCLFileType_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CLFileType_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CLFileType_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCLFileType_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CLFileType_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CLFileType_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
```

```
    }
    [TestMethod]
    public void selectCLFileType_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CLFileType_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CLFileType_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimage_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimage_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimage_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimage_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimage_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimage_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimage_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimage_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimage_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimage_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimage_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
```

```

        var cc1 = (new
RSImage_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSImage_N_P()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSImage_N_P()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSImage_N_P()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSImage_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSImage_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSImage_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSImage_X_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSImage_X_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSImage_X_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSband_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new RSband_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSband_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSband_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new RSband_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();

```

```

        var cc1 = (new
RSband_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSband_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new RSband_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSband_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSband_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new RSband_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSband_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSband_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSband_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSband_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();

```

```

        var cc1 = (new
LIDARdata_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_N_P()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_N_P()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_N_P()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
LIDARdata_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectLIDARdata_X_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
    }

```

```

        var cc = (new
LIDARdata_X_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
LIDARdata_X_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_N_P()
    {

```

```

        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_N_P()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_N_P()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectImageMask_X_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ImageMask_X_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ImageMask_X_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCategoryLabels_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CategoryLabels_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CategoryLabels_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectCategoryLabels_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
CategoryLabels_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
CategoryLabels_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```



```
public void selectCategoryLabels_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CategoryLabels_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CategoryLabels_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategoryLabels_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CategoryLabels_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CategoryLabels_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategoryLabels_N_P()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CategoryLabels_N_P()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CategoryLabels_N_P()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategoryLabels_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CategoryLabels_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CategoryLabels_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectCategoryLabels_X_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
CategoryLabels_X_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
CategoryLabels_X_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
```

```
}
[TestMethod]
public void selectUser_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new User_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new User_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectUser_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new User_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new User_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectUser_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new User_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new User_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectUser_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new User_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new User_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectUser_N_P()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new User_N_P()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new User_N_P()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectUser_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
User_N_Filter()).startSelect().Take(10).ToList();
```

```

        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
User_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectUser_X_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new User_X_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new User_X_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProjectUser_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ProjectUser_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ProjectUser_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProjectUser_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ProjectUser_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ProjectUser_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProjectUser_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ProjectUser_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ProjectUser_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProjectUser_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ProjectUser_N_R()).startSelect().Take(10).ToList();

```

```

        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ProjectUser_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectProjectUser_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ProjectUser_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ProjectUser_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimportMethod_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimportMethod_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimportMethod_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimportMethod_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimportMethod_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimportMethod_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimportMethod_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimportMethod_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimportMethod_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimportMethod_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
    }

```

```

        var cc = (new
RSimportMethod_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimportMethod_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSimportMethod_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSimportMethod_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSimportMethod_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprocMethod_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprocMethod_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprocMethod_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprocMethod_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprocMethod_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprocMethod_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprocMethod_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprocMethod_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprocMethod_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprocMethod_N_R()
    {

```

```

        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprocMethod_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprocMethod_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSprocMethod_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSprocMethod_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSprocMethod_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSexportMethod_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSexportMethod_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSexportMethod_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSexportMethod_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSexportMethod_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSexportMethod_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectRSexportMethod_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
RSexportMethod_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
RSexportMethod_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```

```

public void selectRSEXportMethod_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
RSEXportMethod_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
RSEXportMethod_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectRSEXportMethod_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
RSEXportMethod_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
RSEXportMethod_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserLogins_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserLogins_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserLogins_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserLogins_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserLogins_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserLogins_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserLogins_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserLogins_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserLogins_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}

```

```
}
[TestMethod]
public void selectAspNetUserLogins_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserLogins_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserLogins_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserLogins_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserLogins_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserLogins_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserClaims_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserClaims_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserClaims_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserClaims_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserClaims_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetUserClaims_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetUserClaims_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetUserClaims_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
```



```

        var cc1 = (new
AspNetUserClaims_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserClaims_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserClaims_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserClaims_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserClaims_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserClaims_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserClaims_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
    }

```

```

        var cc = (new
AspNetUserRoles_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetUsers_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetUsers_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetUsers_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetUsers_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetUsers_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetUsers_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetUsers_N_Filter()
    {

```

```

        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetUsers_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetUsers_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetRoles_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetRoles_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetRoles_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetRoles_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetRoles_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetRoles_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUserRoles_AspNetRoles_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUserRoles_AspNetRoles_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUserRoles_AspNetRoles_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationUserAppOperations_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationUserAppOperations_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationUserAppOperations_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```

```

public void selectApplicationUserAppOperations_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_AspNetUsers_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_AspNetUsers_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_AspNetUsers_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}

```

```
}
[TestMethod]
public void selectApplicationUserAppOperations_AspNetUsers_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_AspNetUsers_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_AspNetUsers_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_AspNetUsers_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_AspNetUsers_N_Filter()).startSelect().Take(10).ToList
());
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_AspNetUsers_N_Filter()).startSelect(tc).Take(10).ToLi
st());
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_AppOperations_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_AppOperations_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_AppOperations_N_F()).startSelect(tc).Take(10).ToList
());
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_AppOperations_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationUserAppOperations_AppOperations_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationUserAppOperations_AppOperations_N_R()).startSelect(tc).Take(10).ToList
());
    am.Logout();
}
[TestMethod]
public void selectApplicationUserAppOperations_AppOperations_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
```

```

        var cc = (new
ApplicationUserAppOperations_AppOperations_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationUserAppOperations_AppOperations_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```

```

public void selectApplicationRoleAppOperations_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationRoleAppOperations_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationRoleAppOperations_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectApplicationRoleAppOperations_AppOperations_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationRoleAppOperations_AppOperations_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationRoleAppOperations_AppOperations_N_F()).startSelect(tc).Take(10).ToList(
);
    am.Logout();
}
[TestMethod]
public void selectApplicationRoleAppOperations_AppOperations_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationRoleAppOperations_AppOperations_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationRoleAppOperations_AppOperations_N_R()).startSelect(tc).Take(10).ToList(
);
    am.Logout();
}
[TestMethod]
public void selectApplicationRoleAppOperations_AppOperations_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationRoleAppOperations_AppOperations_N_Filter()).startSelect().Take(10).ToLi
st();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
ApplicationRoleAppOperations_AppOperations_N_Filter()).startSelect(tc).Take(10).To
List();
    am.Logout();
}
[TestMethod]
public void selectApplicationRoleAppOperations_AspNetRoles_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
ApplicationRoleAppOperations_AspNetRoles_N_F()).startSelect().Take(10).ToList();

```

```

        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_AspNetRoles_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_AspNetRoles_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_AspNetRoles_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_AspNetRoles_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectApplicationRoleAppOperations_AspNetRoles_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
ApplicationRoleAppOperations_AspNetRoles_N_Filter()).startSelect().Take(10).ToList
());
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
ApplicationRoleAppOperations_AspNetRoles_N_Filter()).startSelect(tc).Take(10).ToLi
st();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUsers_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUsers_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUsers_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUsers_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUsers_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUsers_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUsers_N_F()
    {

```



```

        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUsers_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUsers_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUsers_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUsers_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUsers_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetUsers_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetUsers_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetUsers_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetRoles_B_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetRoles_B_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetRoles_B_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAspNetRoles_B_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AspNetRoles_B_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AspNetRoles_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]

```

```

public void selectAspNetRoles_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetRoles_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetRoles_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetRoles_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetRoles_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetRoles_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAspNetRoles_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AspNetRoles_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AspNetRoles_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAppOperations_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AppOperations_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AppOperations_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAppOperations_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AppOperations_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AppOperations_B_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}

```

```
}
[TestMethod]
public void selectAppOperations_N_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AppOperations_N_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AppOperations_N_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAppOperations_N_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AppOperations_N_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AppOperations_N_R()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAppOperations_N_Filter()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AppOperations_N_Filter()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AppOperations_N_Filter()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAuditLog_B_F()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AuditLog_B_F()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
    var cc1 = (new
AuditLog_B_F()).startSelect(tc).Take(10).ToList();
    am.Logout();
}
[TestMethod]
public void selectAuditLog_B_R()
{
    IAuth am = new LocalAuthManager();
    am.Login("user", "user100", false);
    var cc = (new
AuditLog_B_R()).startSelect().Take(10).ToList();
    MEDUSContext tc = new MEDUSContext();
}
```

```

        var cc1 = (new
AuditLog_B_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAuditLog_N_F()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AuditLog_N_F()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AuditLog_N_F()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAuditLog_N_R()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AuditLog_N_R()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AuditLog_N_R()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
    [TestMethod]
    public void selectAuditLog_N_Filter()
    {
        IAuth am = new LocalAuthManager();
        am.Login("user", "user100", false);
        var cc = (new
AuditLog_N_Filter()).startSelect().Take(10).ToList();
        MEDUSContext tc = new MEDUSContext();
        var cc1 = (new
AuditLog_N_Filter()).startSelect(tc).Take(10).ToList();
        am.Logout();
    }
}
}
}

```

6.3. Manuālā testa piemērs

Ārējās datu importa funkcijas izsaukums

Solis	Lietotāja darbība	Programmas reakcija
	Piestartēt programmu. Pieslēgties kā testa lietotājs	Parādās galvenais programmas logs ar izvelni un noklusēto projektu
	Izvēlēties izvelnē "Metodes->RSimportMethod"	Logā parādās importa metožu pārvaldības forma.
	Nospiegt pogu [Jauns]	Parādās importa metodes reģistrēšanas

		forma
	Ievadīt datus: Metodes nosaukums: MEDUSimport Mfunction: importPIX TxtParameters: showImage DestRImage : <uzstrādīt>	
	Nospiest [Saglabāt]	Reģistrēšanas forma tiek aizvērta, atjaunināta metožu pārvaldības forma.
	Sarakstā izvēlēties jaunievadīto metodi	Metodes ieraksts tiek izdalīts
	Nospiest pogu [Palaist]	Parādās metodes izpildīšanas dialogs. Pārbaudīt, ka lauki "Metodes nosaukums", "Mfunction" un "TxtParameters" satur tikko ievadīto informāciju.
	Nospiest pogu [...] pie lauka "Importējamā datne" un izvēlēties .pix datni.	Parādās standarta datnes izvēles dialogs. Pēc datnes izvēlēšanās, dialogs aizveras, un metodes izpildīšanas dialogā lauks "Importējamā datne" ir aizpildīts ar izvēlētajās datnes ceļu.
	Nospiest pogu [Izpildīt]	
	Nogaidīt, kamēr ārpus programmas loga parādīsies logs ar importējamo attēlu (ārējās funkcijas izsaukuma efekts)	Metodes izpildīšanas dialogs pazūd.
	Izvēlnē izvēlēties "Projekta dati -> RImage"	Parādās RImage pārvaldības forma
	Parbaudīt, ka sarakstā ir parādījies jauns RImage.	
	Pārbaudīt, ka noklusētā projekta kokā ir parādījusies jauna datne.	