Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

(ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

Отчет по практической работе №1 по дисциплине

«Технология программирования»

Студент гр. з-39-2

\_\_\_\_\_\_\_\_\_\_\_\_ / В.О. Михайленко

« » 20 г

Проверил

Ассистент каф. ЭМИС

\_\_\_\_\_\_\_\_\_\_\_\_ / Я.В. Костелей

« \_\_ » \_\_\_\_\_\_\_ 20\_\_ г

Томск 2023

# Цель работы

Цель работы: Познакомится с основами системного тестирования и написать свой UI-автотест.

# Задачи

Для заданного в предыдущей лабораторной работе приложения провести системное тестирование и решить следующие задачи:

− описать задачу приложения и провести его реализацию (включая конструирование формы);

− написать тест-кейсы;

− на основании тест-кейсов написать UI-тесты.

Сформировать отчет, содержащий титульный лист, цели и задачи лабораторной работы, ход работы, описание тестируемой программы, описание ее интерфейса, тест-кейсы, код и описание UI-тестов, выводы.

# Подготовка проекта

Дополним проект, созданный для выполнения лабораторной работы №2, добавив в него класс подстановки заглушек ManageClass. В конструкторе форм разместим необходимые элементы управления и изменим их Name.

Итоговый внешний вид формы RegisterForm представлен на рисунке 1.

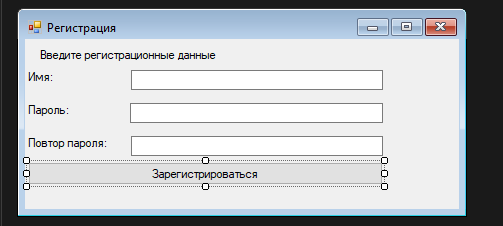


Рисунок 1 – Форма регистрации

1) ErrorLabel (тип Label) – надпись, служащая для вывода информации об успешном выполнении кода или исключениях на форму; при инициализации содержит справочный текст: «Введите регистрационные данные».

2) LoginLabel (тип Label) – надпись, указывающая на поле ввода имени пользователя.

3) PasswordLabel (тип Label) – надпись, указывающая на поле ввода пароля.

4) RepPasswordLabel (тип Label) – надпись, указывающая на поле ввода для повторения пароля.

5) LoginTextBox (тип TextBox) – поле ввода имени пользователя.

6) PasswordTextBox (тип TextBox) – поле ввода пароля.

7) RepPasswordTextBox (тип TextBox) – поле ввода для повтора пароля.

8) RegisterButton (тип Button) – кнопка для запуска основного потока событий.

Фрагмент кода для обработки события нажатия кнопки RegisterButton представлен на рисунке 2.

Рисунок 2 – Обработка нажатия кнопки RegisterButton

# Тест-кейсы

В первом кейсе проверяем все варианты ввода логина и пароля, а также проверяем внешний вид.

Тест-кейс 001. Внешний вид и ввод логина и пароля в формы RegisterForm.

|  |  |  |
| --- | --- | --- |
| № | Описание Step | Описание ожидаемого результата |
| 1 | Запустить приложение с флагом «3» | Менее чем за M мс отобразилось:  Окно с заголовком «Регистрация»  RegisterButton с текстом «Зарегистрироваться»  LoginLabel с текстом «Имя:»  LoginLabel с текстом «Пароль:»  RepPasswordLabel с текстом «Повтор пароля:»  ErrorLabel с текстом «Введите регистрационные данные»  LoginTextBox, PasswordTextBox, RepPasswordTextBox пустые. |
| 2 | Не вводить данных  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом EmptyLogin = "Логин не может быть пустым." |
| 3 | Ввести следующие данные:  login = "myname\_doctor"  password = ""  repPassword = "DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом EmptyPassword1 = "Пропущено поле первого ввода пароля." |
| 4 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123!"  repPassword = ""  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом EmptyPassword2 = "Пропущено поле второго ввода пароля." |
| 5 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123!"  repPassword = " DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом DifferentPasswords = "Пароли не совпадают!" |
| 6 | Ввести следующие данные:  login = "myname\_doctor"  password = "myname\_doctor"  repPassword = "myname\_doctor"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом SameLoginPassword = "Логин и пароль не могут совпадать." |
| 7 | Ввести следующие данные:  login = "myname\_doctor"  password = "Doc123!"  repPassword = "Doc123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом PasswordLess10Chars = "Пароль не может быть менее 10 символов." |
| 8 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest!"  repPassword = "DoctorSuperBest!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом PasswordNoNumber = "Пароль должен содержать хотя бы один символ цифры." |
| 9 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123"  repPassword = "DoctorSuperBest123"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом PasswordNoExtraChar = "Пароль должен содержать хотя бы один символ из @#$%^&\*! ." |
| 10 | Ввести следующие данные:  login = "myname\_doctor"  password = "doctorsuperbest123!"  repPassword = "doctorsuperbest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом PasswordNoUpperChar = "Пароль должен содержать хотя бы один символ в верхнем регистре." |
| 11 | Ввести следующие данные:  login = "myname\_doctor!"  password = "DoctorSuperBest123!"  repPassword = "DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом LoginForbidden = "Логин должен состоять только из цифр, букв и символа \_." |
| 12 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123!"  repPassword = "DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс:  Отобразился диалог с текстом "Пользователь " + doctor.Login + " зарегистрирован!" и кнопкой с текстом «ОК» |
| 13 | Нажать ОК в диалоге | Приложение завершилось |

Проверяем реакцию на отсутствие доступа к БД.

Тест-кейс 002. Сообщения формы RegisterForm при отсутствии подключения к БД.

|  |  |  |
| --- | --- | --- |
| № | Описание Step | Описание ожидаемого результата |
| 1 | Запустить приложение с флагом «1» | Менее чем за M мс отобразилось:  Окно с заголовком «Регистрация»  RegisterButton с текстом «Зарегистрироваться» |
| 2 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123!"  repPassword = "DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом NoConnectionDB = "Нет доступа к базе данных, проверьте подключение." |
| 3 | Закрыть приложение | Приложение завершилось |

Проверяем реакцию на неуникальный логин.

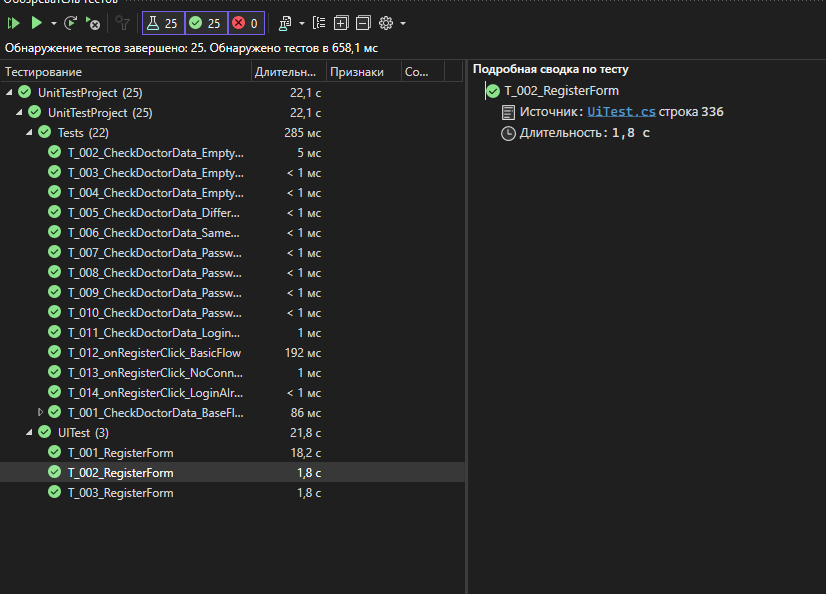
Тест-кейс 003. Сообщения RegisterForm при наличии пользователя с введенным логином.

|  |  |  |
| --- | --- | --- |
| № | Описание Step | Описание ожидаемого результата |
| 1 | Запустить приложение с флагом «2» | Менее чем за M мс отобразилось:  Окно с заголовком «Регистрация»  RegisterButton с текстом «Зарегистрироваться» |
| 2 | Ввести следующие данные:  login = "myname\_doctor"  password = "DoctorSuperBest123!"  repPassword = "DoctorSuperBest123!"  Нажать на кнопку RegisterButton | Менее чем за M мс отобразился ErrorLabel c тестом LoginAlreadyExists = "Уже существует пользователь с данным логином." |
| 3 | Закрыть приложение | Приложение завершилось |

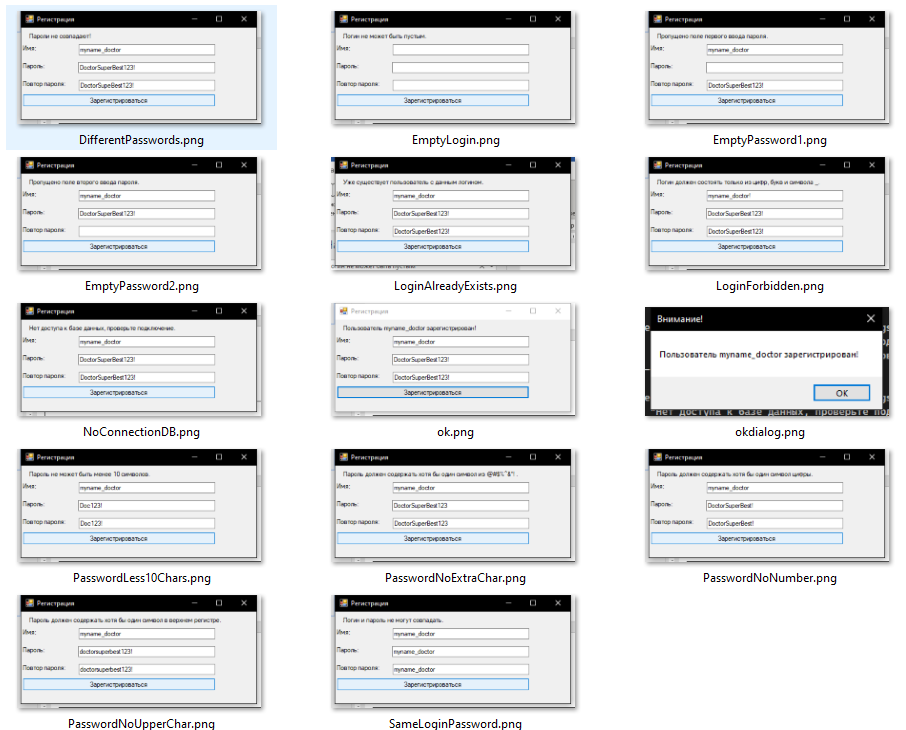
# Написание UI-автотестов

С помощью библиотеки FlaUI на основании тестового проекта из лабораторной работы №2 создадим UI-автотесты по составленным тест-кейсам.

Пройденные UI-автотесты продемонстрированы на рисунке 3.

Рисунок 3 – Пройденные автотесты

Скриншоты фиксации прохождения автотестов представлены на рисунке 4.

Рисунок 4 – Скриншоты для фиксации прохождения автотестов

# Вывод

В результате ознакомления с основами системного тестирования был создан UI-автотест. Кроме этого, для заданного модуля проведено тестирование и решены следующие задачи:

– в модуле реализован интерфейс пользователя с помощью WindowsForms;

– написаны тест-кейсы для UI-тестирования;

– на основании тест-кейсов написаны UI-автотесты.

# Приложение A UnitTestProject.UITest

using FlaUI.Core.AutomationElements;

using FlaUI.Core.Tools;

using FlaUI.UIA3;

using NUnit.Framework;

using System;

using System.Linq;

using WinFormsAppLabTest;

namespace UnitTestProject

{

internal class UITest

{

/// <summary>Путь до исполняемого файла приложения для тестирования</summary>

string PathTestingApp = @"D:\repos\WinFormsAppLabTest\WinFormsAppLabTest\bin\Debug\net6.0-windows\WinFormsAppLabTest.exe";

/// <summary>время мс допустимой задержки</summary>

int M = 5000;

//текст внутри элементов управления

const string registrationTitleString = "Регистрация";

const string registrationButtonString = "Зарегистрироваться";

const string loginString = "Имя:";

const string passwordString = "Пароль:";

const string repPasswordString = "Повтор пароля:";

const string errorLabelString = "Введите регистрационные данные";

//automatisation-id для элементов управления

const string idRegisterButton = "RegisterButton";

const string idLoginLabel = "LoginLabel";

const string idPasswordLabel = "PasswordLabel";

const string idRepPasswordLabel = "RepPasswordLabel";

const string idErrorLabel = "ErrorLabel";

const string idLoginTextBox = "LoginTextBox";

const string idPasswordTextBox = "PasswordTextBox";

const string idRepPasswordTextB = "RepPasswordTextBox";

public T WaitForElement<T>(Func<T> getter)

{

var retry = Retry.WhileNull<T>(

getter,

TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.Fail($"Невозможно найти элемент {M} ms");

}

return retry.Result;

}

/// <summary>проверяем все варианты ввода логина и пароля, а также проверяем внешний вид и событие изменения размера формы</summary>

[Test]

public void T\_001\_RegisterForm()

{

//Step #001

FlaUI.Core.Application app = FlaUI.Core.Application.Launch(PathTestingApp, "3");

using (var automation = new UIA3Automation())

{

var window = app.GetMainWindow(automation);

//Окно с заголовком «Регистрация»

string title = window.Title;

Assert.AreEqual(registrationTitleString, title);

//получаем ссылки на все элементы управления

var registerButton = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRegisterButton)).AsButton());

var loginTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginTextBox)).AsTextBox());

var passwordTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordTextBox)).AsTextBox());

var repPasswordTextB = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordTextB)).AsTextBox());

var loginLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginLabel)).AsLabel());

var passwordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordLabel)).AsLabel());

var repPasswordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordLabel)).AsLabel());

var errorLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idErrorLabel)).AsLabel());

//Проверяем строки

Assert.AreEqual(registrationButtonString, registerButton.AsLabel().Text);

Assert.AreEqual(loginString, loginLabel.Text);

Assert.AreEqual(passwordString, passwordLabel.Text);

Assert.AreEqual(repPasswordString, repPasswordLabel.Text);

Assert.AreEqual(errorLabelString, errorLabel.Text);

Assert.AreEqual("", loginTextBox.Text);

Assert.AreEqual("", passwordTextBox.Text);

Assert.AreEqual("", repPasswordTextB.Text);

//Step #002

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("EmptyLogin.png");

var retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyLogin, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyLogin, errorLabel.Text);

}

//Step #003

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("");

repPasswordTextB.Enter("DoctorSuperBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("EmptyPassword1.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyPassword1, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyLogin, errorLabel.Text);

}

//Step #004

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("EmptyPassword2.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyPassword2, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.EmptyPassword2, errorLabel.Text);

}

//Step #005

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("DoctorSupeBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("DifferentPasswords.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.DifferentPasswords, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.DifferentPasswords, errorLabel.Text);

}

//Step #006

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("myname\_doctor");

repPasswordTextB.Enter("myname\_doctor");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("SameLoginPassword.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.SameLoginPassword, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.SameLoginPassword, errorLabel.Text);

}

//Step #007

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("Doc123!");

repPasswordTextB.Enter("Doc123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("PasswordLess10Chars.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordLess10Chars, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordLess10Chars, errorLabel.Text);

}

//Step #008

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest!");

repPasswordTextB.Enter("DoctorSuperBest!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("PasswordNoNumber.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoNumber, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoNumber, errorLabel.Text);

}

//Step #009

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123");

repPasswordTextB.Enter("DoctorSuperBest123");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("PasswordNoExtraChar.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoExtraChar.Replace("&&", "&"), errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoExtraChar.Replace("&&", "&"), errorLabel.Text);

}

//Step #010

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("doctorsuperbest123!");

repPasswordTextB.Enter("doctorsuperbest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("PasswordNoUpperChar.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoUpperChar, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.PasswordNoUpperChar, errorLabel.Text);

}

//Step #011

loginTextBox.Enter("myname\_doctor!");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("DoctorSuperBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("LoginForbidden.png");

retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.LoginForbidden, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.LoginForbidden, errorLabel.Text);

}

//Step #012

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("DoctorSuperBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("ok.png");

retry = Retry.WhileException(() =>

{

var msg = window.ModalWindows.FirstOrDefault().AsWindow();

Assert.NotNull(msg);

var message=msg.FindFirstChild(cf => cf.ByText("Пользователь " + "myname\_doctor" + " зарегистрирован!")).AsLabel();

Assert.NotNull(message);

//Step #013

var yesButton = msg.FindFirstChild(cf => cf.ByName("ОК")).AsButton();

Assert.NotNull(yesButton);

msg.CaptureToFile("okdialog.png");

yesButton.Click();

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.Fail("Нет диалогового окна регистрации");

}

System.Threading.Thread.Sleep(1000);

app.Close();

}

}

/// <summary>нет подключение к бд</summary>

[Test]

public void T\_002\_RegisterForm()

{

//Step #1

FlaUI.Core.Application app = FlaUI.Core.Application.Launch(PathTestingApp, "1");

using (var automation = new UIA3Automation())

{

var window = app.GetMainWindow(automation);

//Окно с заголовком «Регистрация»

string title = window.Title;

Assert.AreEqual(registrationTitleString, title);

//получаем ссылки на все элементы управления

var registerButton = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRegisterButton)).AsButton());

var loginTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginTextBox)).AsTextBox());

var passwordTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordTextBox)).AsTextBox());

var repPasswordTextB = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordTextB)).AsTextBox());

var loginLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginLabel)).AsLabel());

var passwordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordLabel)).AsLabel());

var repPasswordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordLabel)).AsLabel());

var errorLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idErrorLabel)).AsLabel());

//Проверяем строки

Assert.AreEqual(registrationButtonString, registerButton.AsLabel().Text);

Assert.AreEqual(loginString, loginLabel.Text);

Assert.AreEqual(passwordString, passwordLabel.Text);

Assert.AreEqual(repPasswordString, repPasswordLabel.Text);

Assert.AreEqual(errorLabelString, errorLabel.Text);

Assert.AreEqual("", loginTextBox.Text);

Assert.AreEqual("", passwordTextBox.Text);

Assert.AreEqual("", repPasswordTextB.Text);

//Step #2

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("DoctorSuperBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("NoConnectionDB.png");

var retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.NoConnectionDB, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.NoConnectionDB, errorLabel.Text);

}

//Step #3

app.Close();

}

}

/// <summary>неуникальный логин</summary>

[Test]

public void T\_003\_RegisterForm()

{

//Step #1

FlaUI.Core.Application app = FlaUI.Core.Application.Launch(PathTestingApp, "2");

using (var automation = new UIA3Automation())

{

var window = app.GetMainWindow(automation);

//Окно с заголовком «Регистрация»

string title = window.Title;

Assert.AreEqual(registrationTitleString, title);

//получаем ссылки на все элементы управления

var registerButton = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRegisterButton)).AsButton());

var loginTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginTextBox)).AsTextBox());

var passwordTextBox = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordTextBox)).AsTextBox());

var repPasswordTextB = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordTextB)).AsTextBox());

var loginLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idLoginLabel)).AsLabel());

var passwordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idPasswordLabel)).AsLabel());

var repPasswordLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idRepPasswordLabel)).AsLabel());

var errorLabel = WaitForElement(() => window.FindFirstDescendant(cf => cf.ByAutomationId(idErrorLabel)).AsLabel());

//Проверяем строки

Assert.AreEqual(registrationButtonString, registerButton.AsLabel().Text);

Assert.AreEqual(loginString, loginLabel.Text);

Assert.AreEqual(passwordString, passwordLabel.Text);

Assert.AreEqual(repPasswordString, repPasswordLabel.Text);

Assert.AreEqual(errorLabelString, errorLabel.Text);

Assert.AreEqual("", loginTextBox.Text);

Assert.AreEqual("", passwordTextBox.Text);

Assert.AreEqual("", repPasswordTextB.Text);

//Step #2

loginTextBox.Enter("myname\_doctor");

passwordTextBox.Enter("DoctorSuperBest123!");

repPasswordTextB.Enter("DoctorSuperBest123!");

registerButton.Click();

System.Threading.Thread.Sleep(1000);

window.CaptureToFile("LoginAlreadyExists.png");

var retry = Retry.WhileException(() =>

{

Assert.AreEqual(RegisterForm.ExceptionStrings.LoginAlreadyExists, errorLabel.Text);

}, TimeSpan.FromMilliseconds(M));

if (!retry.Success)

{

Assert.AreEqual(RegisterForm.ExceptionStrings.LoginAlreadyExists, errorLabel.Text);

}

//Step #3

app.Close();

}

}

}

}

# Приложение Б WinFormsAppLabTest.RegisterForm

namespace WinFormsAppLabTest

{

public partial class RegisterForm : Form

{

public RegisterForm()

{

InitializeComponent();

controller= ManageClass.GetDatabaseController();

}

public static class ExceptionStrings

{

public const string EmptyLogin = "Логин не может быть пустым.";

public const string EmptyPassword1 = "Пропущено поле первого ввода пароля.";

public const string EmptyPassword2 = "Пропущено поле второго ввода пароля.";

public const string DifferentPasswords = "Пароли не совпадают!";

public const string SameLoginPassword = "Логин и пароль не могут совпадать.";

public const string PasswordLess10Chars = "Пароль не может быть менее 10 символов.";

public const string PasswordNoNumber = "Пароль должен содержать хотя бы один символ цифры.";

public const string PasswordNoExtraChar = "Пароль должен содержать хотя бы один символ из @#$%^&&\*! .";

public const string PasswordNoUpperChar = "Пароль должен содержать хотя бы один символ в верхнем регистре.";

public const string LoginForbidden = "Логин должен состоять только из цифр, букв и символа \_.";

public const string NoConnectionDB = "Нет доступа к базе данных, проверьте подключение.";

public const string LoginAlreadyExists = "Уже существует пользователь с данным логином.";

}

public IDatabaseController controller=null;

public IDoctorEntry onRegisterClick(string login, string password, string repPassword)

{

if (checkDoctorData(login, password, repPassword))

{

if (controller.tryConnectDB())

{

if (controller.tryCreateAccount(login, password))

{

IDoctorEntry doctor = controller.getNewDoctorEntry();

controller.login(doctor.Login, doctor.Password);

return doctor;

}

else

{

throw new Exception(ExceptionStrings.LoginAlreadyExists);

}

}

else

{

throw new Exception(ExceptionStrings.NoConnectionDB);

}

}

return null;

}

public static bool checkDoctorData(string login,

string password,

string repPassword)

{

if ( login==null || login.Length==0 )

{

throw new Exception(ExceptionStrings.EmptyLogin);

}

if (password == null || password.Length == 0)

{

throw new Exception(ExceptionStrings.EmptyPassword1);

}

if (repPassword == null || repPassword.Length == 0)

{

throw new Exception(ExceptionStrings.EmptyPassword2);

}

if (password != repPassword)

{

throw new Exception(ExceptionStrings.DifferentPasswords);

}

if (login == password)

{

throw new Exception(ExceptionStrings.SameLoginPassword);

}

if (password.Length < 10)

{

throw new Exception(ExceptionStrings.PasswordLess10Chars);

}

System.Text.RegularExpressions.Regex regex = new System.Text.RegularExpressions.Regex(@"[0-9]");

if (!regex.IsMatch(password))

{

throw new Exception(ExceptionStrings.PasswordNoNumber);

}

regex = new System.Text.RegularExpressions.Regex(@"[@#$%^&\*!]");

if (!regex.IsMatch(password))

{

throw new Exception(ExceptionStrings.PasswordNoExtraChar);

}

regex = new System.Text.RegularExpressions.Regex(@"[A-ZА-Я]");

if (!regex.IsMatch(password))

{

throw new Exception(ExceptionStrings.PasswordNoUpperChar);

}

regex = new System.Text.RegularExpressions.Regex(@"^[0-9A-ZА-Яa-zа-я\_]+$");

if (!regex.IsMatch(login))

{

throw new Exception(ExceptionStrings.LoginForbidden);

}

return true;

}

private void RegisterButton\_Click(object sender, EventArgs e)

{

ErrorLabel.Text = "";

try

{

string login = LoginTextBox.Text;

string password = PasswordTextBox.Text;

string repPassword = RepPasswordTextBox.Text;

IDoctorEntry doctor=onRegisterClick(login, password, repPassword);

ErrorLabel.Text = "Пользователь " + doctor.Login + " зарегистрирован!";

if (MessageBox.Show("Пользователь " + doctor.Login + " зарегистрирован!","Внимание!") == DialogResult.OK)

{

this.Close();

}

}

catch (Exception exception)

{

ErrorLabel.Text = exception.Message;

}

}

}

}