Экзамен по дисциплине: Тестирование ПО

Группа: СПУ 14в-1

Дата: 21.04.2016

Студент: Новичихин Павел

1. Тестирование черного ящика

Ответ: Тестирование черного ящика - это тестирование, при котором не известно устройство программного кода программы, тестирование производится с точки зрения пользователя исследуя функционал ПО по техническому заданию.

1. Тестирование серого ящика

Ответ: Тестирование серого ящика – это тестирование, при котором используются подход, как белого, так и черного ящика, известно устройство кода, но тестирование производится функционала ПО по техническому заданию.

1. Тестирование доступности

Ответ: Тестирование доступности – это тестирование направленное на определение насколько ПО доступно пользователю, например людям с ограниченными возможностями.

Практика

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using OpenQA.Selenium;

using OpenQA.Selenium.Support.UI;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.IE;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

private string baseURL = "http://mystat.itstep.org";

private IWebDriver driver;

[TestCleanup]

public void CleanUp()

{

driver.Quit();

}

[TestMethod]

public void TestAuthenticationErrorInFF()

{

driver = new FirefoxDriver();

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("adsdasd");

driver.FindElement(By.Id("loginform-password")).SendKeys("asddasdasd");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(60));

bool containsMessage = driver.FindElement(By.Id("w0")).Text.Contains("Неверный логин и/или пароль");

Assert.AreEqual(true, containsMessage);

}

[TestMethod]

public void TestAuthenticationSuccessInFF()

{

driver = new FirefoxDriver();

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("Novi\_of07");

driver.FindElement(By.Id("loginform-password")).SendKeys("\*\*\*\*\*");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(60));

Assert.AreEqual("Mystat Rank", driver.Title);

}

[TestMethod]

public void TestAuthenticationErrorInGC()

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("adsdasd");

driver.FindElement(By.Id("loginform-password")).SendKeys("asddasdasd");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(1));

wait.Until(e => driver.FindElement(By.CssSelector("div#w0")));

bool containsMessage = driver.FindElement(By.Id("w0")).Text.Contains("Неверный логин и/или пароль");

Assert.AreEqual(true, containsMessage);

}

[TestMethod]

public void TestAuthenticationSuccessInGC()

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("Novi\_of07");

driver.FindElement(By.Id("loginform-password")).SendKeys("\*\*\*\*\*");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(1));

wait.Until(e => driver.Title == "Mystat Rank");

Assert.AreEqual("Mystat Rank", driver.Title);

}

[TestMethod]

public void TestAuthenticationErrorInIE()

{

InternetExplorerOptions options = new InternetExplorerOptions();

options.EnableNativeEvents = false;

driver = new InternetExplorerDriver(options);

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("adsdasd");

driver.FindElement(By.Id("loginform-password")).SendKeys("asddasdasd");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(1));

wait.Until(e => driver.FindElement(By.CssSelector("div#w0")));

bool containsMessage = driver.FindElement(By.Id("w0")).Text.Contains("Неверный логин и/или пароль");

Assert.AreEqual(true, containsMessage);

}

[TestMethod]

public void TestAuthenticationSuccessInIE()

{

InternetExplorerOptions options = new InternetExplorerOptions();

options.EnableNativeEvents = false;

driver = new InternetExplorerDriver(options);

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).SendKeys("Novi\_of07");

driver.FindElement(By.Id("loginform-password")).SendKeys("\*\*\*\*\*");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(1));

wait.Until(e => driver.Title == "Mystat Rank");

Assert.AreEqual("Mystat Rank", driver.Title);

}

}

}