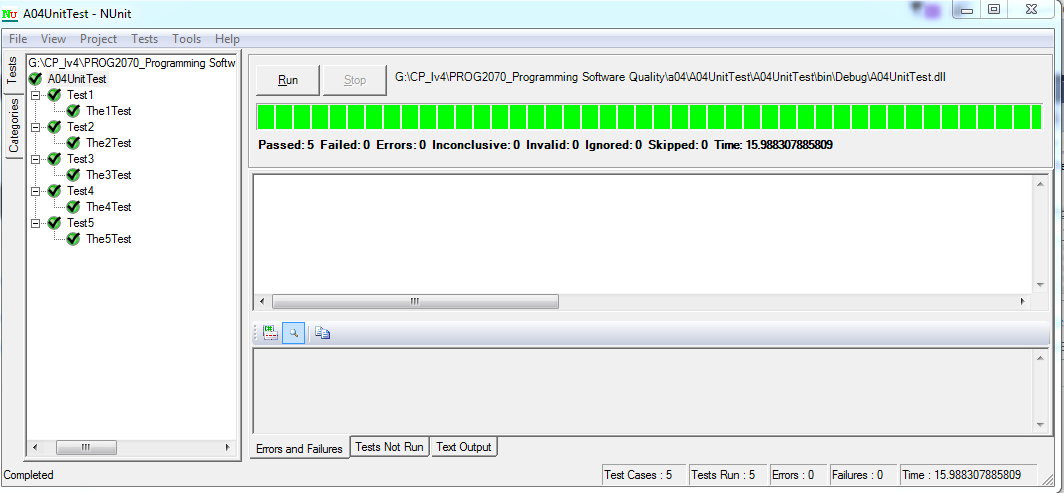
**Nunit screen shot**



**C# src code**

**Test1 – Normal case**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment4 - Test Program (Normal Case)

///

/// Yoonsuk Cho #7135551

/// Apr 13, 2017

/// </summary>

namespace A04UnitTest

{

[TestFixture]

public class Test1

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

//private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

FirefoxDriverService service = FirefoxDriverService.CreateDefaultService("c:\\Users\\ycho5551\\Downloads\\selenium-dotnet-3.3.0\\net40");

driver = new FirefoxDriver(service);

//driver = new FirefoxDriver();

baseURL = "http://localhost";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

//try

//{

// driver.Quit();

//}

//catch (Exception)

//{

// Ignore errors if unable to close the browser

//}

//Assert.AreEqual("", verificationErrors.ToString());

if (driver != null)

{

driver.Quit();

}

}

[Test]

public void The1Test()

{

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

//driver.FindElement(By.Id("inputPage")).Click();

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

driver.FindElement(By.Id("name")).Clear();

driver.FindElement(By.Id("name")).SendKeys("Seller Zero");

driver.FindElement(By.Id("address")).Clear();

driver.FindElement(By.Id("address")).SendKeys("100 King st");

driver.FindElement(By.Id("city")).Clear();

driver.FindElement(By.Id("city")).SendKeys("Kitchener");

driver.FindElement(By.Id("phoneNumber")).Clear();

driver.FindElement(By.Id("phoneNumber")).SendKeys("519-555-0001");

driver.FindElement(By.Id("email")).Clear();

driver.FindElement(By.Id("email")).SendKeys("abcd@efgh.ca");

driver.FindElement(By.Id("make")).Clear();

driver.FindElement(By.Id("make")).SendKeys("Ford");

driver.FindElement(By.Id("model")).Clear();

driver.FindElement(By.Id("model")).SendKeys("mustang");

driver.FindElement(By.Id("year")).Clear();

driver.FindElement(By.Id("year")).SendKeys("2016");

driver.FindElement(By.Id("testBtn")).Click();

//WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

// wait.Until(d => d.Title.Contains("JD"));

// ERROR: Caught exception [ERROR: Unsupported command [waitForPopUp | | 30000]]

// driver.FindElement(By.Id("testBtn2")).Click();

// wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

// wait.Until(d => d.Title.Contains("Assignment"));

// ERROR: Caught exception [unknown command [assertTest1]]

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

//private string CloseAlertAndGetItsText() {

// try {

// IAlert alert = driver.SwitchTo().Alert();

// string alertText = alert.Text;

// if (acceptNextAlert) {

// alert.Accept();

// } else {

// alert.Dismiss();

// }

// return alertText;

// } finally {

// acceptNextAlert = true;

// }

//}

}

}

**Test2 – Error case**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment4 - Test Program (Error Case)

///

/// Yoonsuk Cho #7135551

/// Apr 13, 2017

/// </summary>

namespace A04UnitTest

{

[TestFixture]

public class Test2

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

FirefoxDriverService service = FirefoxDriverService.CreateDefaultService("c:\\Users\\ycho5551\\Downloads\\selenium-dotnet-3.3.0\\net40");

driver = new FirefoxDriver(service);

baseURL = "http://127.0.0.1:53948/";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void The2Test()

{

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

//driver.FindElement(By.Id("inputPage")).Click();

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

driver.FindElement(By.Id("name")).Clear();

driver.FindElement(By.Id("name")).SendKeys("Seller Err");

driver.FindElement(By.Id("address")).Clear();

driver.FindElement(By.Id("address")).SendKeys("234 Wide rd");

driver.FindElement(By.Id("city")).Clear();

driver.FindElement(By.Id("city")).SendKeys("Waterloo");

driver.FindElement(By.Id("phoneNumber")).Clear();

driver.FindElement(By.Id("phoneNumber")).SendKeys("112-1123");

driver.FindElement(By.Id("email")).Clear();

driver.FindElement(By.Id("email")).SendKeys("abcdf@");

driver.FindElement(By.Id("make")).Clear();

driver.FindElement(By.Id("make")).SendKeys("Toyota");

driver.FindElement(By.Id("model")).Clear();

driver.FindElement(By.Id("model")).SendKeys("yaris");

driver.FindElement(By.Id("year")).Clear();

driver.FindElement(By.Id("year")).SendKeys("2014");

driver.FindElement(By.Id("testBtn")).Click();

Assert.AreEqual("Enter a valid email address. [eMail].\r\n", CloseAlertAndGetItsText());

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

**Test3 – Error case**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment4 - Test Program (Error Case)

///

/// Yoonsuk Cho #7135551

/// Apr 13, 2017

/// </summary>

namespace A04UnitTest

{

[TestFixture]

public class Test3

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

FirefoxDriverService service = FirefoxDriverService.CreateDefaultService("c:\\Users\\ycho5551\\Downloads\\selenium-dotnet-3.3.0\\net40");

driver = new FirefoxDriver(service);

baseURL = "http://localhost";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void The3Test()

{

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

//driver.FindElement(By.Id("inputPage")).Click();

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

driver.FindElement(By.Id("name")).Clear();

driver.FindElement(By.Id("name")).SendKeys("Seller April");

driver.FindElement(By.Id("address")).Clear();

driver.FindElement(By.Id("address")).SendKeys("99 Narrow rd");

driver.FindElement(By.Id("city")).Clear();

driver.FindElement(By.Id("city")).SendKeys("Kitchener");

driver.FindElement(By.Id("phoneNumber")).Clear();

driver.FindElement(By.Id("phoneNumber")).SendKeys("11111");

driver.FindElement(By.Id("email")).Clear();

driver.FindElement(By.Id("email")).SendKeys("abcde@");

driver.FindElement(By.Id("testBtn")).Click();

Assert.AreEqual("Enter a valid email address. [eMail].\r\nPlease check the mandatory field. [Vehicle make].\r\nPlease check the mandatory field. [Vehicle model].\r\nPlease check the mandatory field. [Vehicle year].\r\nWrong format error. [Phone Number - ex) 123-456-7890].\r\n", CloseAlertAndGetItsText());

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

**Test4 – Error case**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment4 - Test Program (Error Case)

///

/// Yoonsuk Cho #7135551

/// Apr 13, 2017

/// </summary>

namespace A04UnitTest

{

[TestFixture]

public class Test4

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

FirefoxDriverService service = FirefoxDriverService.CreateDefaultService("c:\\Users\\ycho5551\\Downloads\\selenium-dotnet-3.3.0\\net40");

driver = new FirefoxDriver(service);

baseURL = "http://localhost";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void The4Test()

{

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

//driver.FindElement(By.Id("inputPage")).Click();

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

driver.FindElement(By.Id("name")).Clear();

driver.FindElement(By.Id("name")).SendKeys("Seller Error");

driver.FindElement(By.Id("testBtn")).Click();

Assert.AreEqual("Please check the mandatory field. [Address].\r\nPlease check the mandatory field. [City].\r\nPlease check the mandatory field. [Phone Number].\r\nPlease check the mandatory field. [eMail].\r\nPlease check the mandatory field. [Vehicle make].\r\nPlease check the mandatory field. [Vehicle model].\r\nPlease check the mandatory field. [Vehicle year].\r\nWrong format error. [Phone Number - ex) 123-456-7890].\r\n", CloseAlertAndGetItsText());

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

**Test5 – Normal case**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment4 - Test Program (Normal Case)

///

/// Yoonsuk Cho #7135551

/// Apr 13, 2017

/// </summary>

namespace A04UnitTest

{

[TestFixture]

public class Test5

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

FirefoxDriverService service = FirefoxDriverService.CreateDefaultService("c:\\Users\\ycho5551\\Downloads\\selenium-dotnet-3.3.0\\net40");

driver = new FirefoxDriver(service);

baseURL = "http://localhost";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void The5Test()

{

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

//driver.FindElement(By.Id("listPage")).Click();

//driver.FindElement(By.Id("testBtn")).Click();

//driver.FindElement(By.Id("inputPage")).Click();

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

driver.FindElement(By.Id("name")).Clear();

driver.FindElement(By.Id("name")).SendKeys("Sellor abc");

driver.FindElement(By.Id("address")).Clear();

driver.FindElement(By.Id("address")).SendKeys("99 avenue street");

driver.FindElement(By.Id("city")).Clear();

driver.FindElement(By.Id("city")).SendKeys("guelph");

driver.FindElement(By.Id("phoneNumber")).Clear();

driver.FindElement(By.Id("phoneNumber")).SendKeys("987-654-4321");

driver.FindElement(By.Id("email")).Clear();

driver.FindElement(By.Id("email")).SendKeys("akakak@kkkd.net");

driver.FindElement(By.Id("make")).Clear();

driver.FindElement(By.Id("make")).SendKeys("Ford");

driver.FindElement(By.Id("model")).Clear();

driver.FindElement(By.Id("model")).SendKeys("mustang");

driver.FindElement(By.Id("year")).Clear();

driver.FindElement(By.Id("year")).SendKeys("2011");

driver.FindElement(By.Id("testBtn")).Click();

// ERROR: Caught exception [ERROR: Unsupported command [waitForPopUp | | 30000]]

//driver.FindElement(By.Id("testBtn2")).Click();

//driver.FindElement(By.Id("listPage")).Click();

//driver.FindElement(By.Id("testBtn")).Click();

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText() {

try {

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert) {

alert.Accept();

} else {

alert.Dismiss();

}

return alertText;

} finally {

acceptNextAlert = true;

}

}

}

}

**Screen Capture**

