Printshop Workflow Automation System
P.W.A.S.

PWAS Test Cases

Presented by:

Dulcardo Arteaga | Naveen Gowda | Larissa Guerrero
Erik Kessler | Lenny Markus | Javier Mesa | Rolando Vicaria

Name	**Register_Test_1**
Test type	Sunny day
Version	1.0
Use case ID	Register
Purpose	The purpose of this test case is to test the Use Case Register. The system shall create an account for the customer Roger Smith by adding his information into the data repository after Roger enters the valid data into the required fields.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The customer account should not exist on the Database
Input	1. HTTP request = "register.aspx"' 2. Customer information: email address = 'roger@yahoo.com' password = '12345678' confirm password = '12345678' first name = 'Roger' last name = 'Smith' company name = 'Qualcomm' phone number = '7863021019' address line 1= '110 SW 140 Ave' city = 'Miami' State = 'Florida' Zip Code = '33174'
Expected output	The system should display a message acknowledging that the customer was registered to the system.
Actual output	Welcome to the XYZ Printshop company website !!

Name	**Register_Test_2**
Test type	Rainy day
Version	1.0
Use case ID	Register
Purpose	The purpose of this test case is to test the Use Case Register. The system shall not create a new customer account if such account already exists on the Database.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The customer account should already exist on the Database
Input	1. HTTP request = "register.aspx"' Customer information: 2. email address = 'roger@yahoo.com' 3. password = '12345678' 4. confirm password = '12345678' 5. first name = 'Roger' 6. last name = 'Smith' 7. company name = 'Qualcomm' 8. phone number = '7863021019' 9. address line 1= '110 SW 140 Ave' 10. city = 'Miami' 11. State = 'Florida' 12. Zip Code = '33174'
Expected output	The system should display a message acknowledging that the customer already exists on the database and cannot be added. For example "The **username** is already in the database".
Actual Output	Username (email) already in use.

Name	**Login_Test_1**
Test type	Rainy day
Version	1.0
Use case ID	Login
Purpose	The purpose of this test case is to test the Use Case Login. The system shall let the customer Roger smith log in to the system after Roger enters the valid data into the required fields.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. There should be a data record on the Database with the same information as the input data.
Input	1. Customer information: username = 'roger@yahoo.com' password = 'abc' 2. HTTP request = 'customerView_Home.aspx'
Expected output	System should display a message acknowledging that the customer was successfully logged in. For example, "Welcome back Roger" where Roger is the First Name of the user who is registered.
Actual output	Welcome back Roger

Name	**Login_Test_2**
Test type	Rainy day
Version	1.0
Use case ID	Login
Purpose	The purpose of this test case is to test the Use Case Login. The system shall prompt the customer Roger Smith an error message if the required field "password" or "user" is invalid.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The data record with the same information as the input data should not exist on the Database.
Input	1. Customer information: username = 'roger@yahoo.com' password = '12345' 2. HTTP request = 'customerView_Home.aspx'
Expected output	System should display a message acknowledging that the customer was unable to log in. For example, "Login Information is not Valid".
Actual output	Login Information is not Valid

Name	**Logout_Test_1**
Test type	Sunny day
Version	1.0
Use case ID	Logout
Purpose	The purpose of this test case is to test the Use Case Logout. The system shall let the customer Roger Smith to logout from the system after he finishes interacting with the system.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. There should be a login session for the customer.
Input	1.HTTP request = ‘’
Expected output	The system should successfully logout and display the login page.
Actual output	The system logs out and displays the Login Page.

Name	**Forgot Password_Test_1**
Test type	Sunny day
Version	1.0
Use case ID	Forgot Password
Purpose	The purpose of this test case is to test the Use Case Forgot Password. The system shall let the customer Roger Smith to reset the Password and the new password should be mailed to the registered email of the customer.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up.
Input	1.HTTP request = 'ForgotPassword.aspx' 2. Customer Information Email Address = "roger@yahoo.com"
Expected output	The system should display a message that the password was reset. For example , "Success! A new password has been sent if the email provided was registered to an account"
Actual output	Success! A new password has been sent if the email provided was registered to an account.

Name	**Forgot Password_Test_2**
Test type	Rainy day
Version	1.0
Use case ID	Forgot Password
Purpose	The purpose of this test case is to test the Use Case Forgot Password. The system shall let the customer Roger Smith to reset the Password and the new password should be mailed to the registered email of the customer.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The data record with the same information as the input data should not exist on the Database.
Input	1.HTTP request = 'ForgotPassword.aspx' 2. Customer Information Email Address = "harry@yahoo.com"
Expected output	The system should display a message that the password was reset. For example , "Please enter a valid email address"
Actual output	Please enter a valid email address.

Name	**Edit Profile_Test_1**
Test type	Sunny day
Version	1.0
Use case ID	Edit Profile
Purpose	The purpose of this test case is to test the Use Case Edit Profile. The system shall let the customer Roger Smith to edit the profile and the new profile should be updated in the database.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The customer 'Roger' has to be logged in.
Input	1.HTTP request = 'customerView_EditProfile.aspx' 2. Login Information email address = 'roger@yahoo.com' new password = 'abc' confirm new password = 'abc' 4. Contact Information first name = 'Roger' last name = 'Smith' company name = 'Qualcomm' phone number = '7863021019' 5. Billing Address address line 1= '110 SW 140 Ave' city = 'Boca Raton' State = 'Florida' Zip Code = '33174' 6. Shipping Address

	address line 1= ‘110 SW 140 Ave’ city = ‘Boca Raton’ State = ‘Florida’ Zip Code = ‘33174’
Expected output	The system should display a message that the profile was updated. For example , “Profile Updated”
Actual output	“Request Completed”

Name	**Edit Profile_Test_2**
Test type	Rainy day
Version	1.0
Use case ID	Edit Profile
Purpose	The purpose of this test case is to test the Use Case Edit Profile. The system shall let the customer Roger Smith to edit the profile and the new profile should be updated in the database.
Test setup (preconditions)	1.All the classes and subsystems that interact with this test case should work properly. 2.The hardware required to run this test case should be properly set up. 3. The customer 'Roger' has to be logged in.
Input	1.HTTP request = 'customerView_EditProfile.aspx' 2. Login Information email address = 'roger@yahoo.com' new password = 'abc' confirm new password = 'abc' 3. Contact Information first name = 'Roger' last name = 'Smith' company name = 'Qualcomm' phone number = '7863021019' 4. Billing Address address line 1= '110 SW 140 Ave' city = 'Boca Raton' State = 'Florida' Zip Code = '33174' 5. Shipping Address address line 1= '110 SW 140 Ave'

	city = ‘Boca Raton’ State = ‘Florida’ Zip Code = ‘33174’ 6. Credit Card Information Credit card # : ABCD1234EFGH5678 Card type : Apex Exp Date : ABCD1234 Security Code : ABCD Name on Card : XYZ
Expected output	The system should display a message to input the correct values.
Actual output	Server Error in '/' Application. *String or binary data would be truncated.* *The statement has been terminated.*

Name	**CreateOrder_Test_1**
Test type	Sunny day
Version	1.0
Use case ID	Create Order
Purpose	The purpose of this test case is to test the Use Case Create Order by not upload the artwork. The system shall
Test setup (preconditions)	7. All the classes and subsystems that interact with this test case should work properly. 8. The hardware required to run this test case should be properly set up.
Input	1.HTTP request = 'customer_create_order.aspx' 2. Create Order Job name = " Roger_Order_1" Final size = "10" in x "15" in Quantity to Print = "200" Stock finish = "Glossy" Stock Weight = "Heavy" Folded = "Checked" Ship = "Checked" Upload File = "Select file to upload" Click Upload Save Order
Expected output	The system should display a message that the Order was created.
Actual output	"Order Created Successful! with ID :1"

Name	**CreateOrder_Test_2**
Test type	Rainy day
Version	1.0
Use case ID	Create Order
Purpose	The purpose of this test case is to test the Use Case Create Order without uploading the artwork. The system shall prompt the customer Roger Smith with an error message to upload the art work
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up.
Input	1.HTTP request = 'customer_create_order.aspx' 2. Create Order Job name = " Roger_Order_2" Final size = "10" in x "15" in Quantity to Print = "200" Stock finish = "Glossy" Stock Weight = "Heavy" Folded = "Checked" Ship = "Checked" Save Order
Expected output	The system should display a message which asks the customer to upload an image.
Actual output	The system does not generate an error Message. It creates an Order , and displays a message that the error was created.

Name	**CreateOrder_Test_3**
Test type	Rainy day
Version	1.0
Use case ID	Create Order
Purpose	The purpose of this test case is to test the Use Case Create Order without uploading the artwork. The system shall prompt the customer Roger Smith with an error message to upload the art work
Test setup (preconditions)	3. All the classes and subsystems that interact with this test case should work properly. 4. The hardware required to run this test case should be properly set up.
Input	1.HTTP request = 'customer_create_order.aspx' 2. Create Order Job name = " Roger_Order_2" Final size = "10" in x "15" in Quantity to Print = "200" Stock finish = "Glossy" Stock Weight = "Heavy" Folded = "Checked" Ship = "Checked" Save Order
Expected output	The system should display a message which asks the customer to upload an image.
Actual output	The system generates the error message "Please upload your artwork before submitting your order"

Name	**EditOrder_Test_1**
Test type	Sunny Day
Version	1.0
Use case ID	Edit Order
Purpose	The purpose of this test case is to test the Use Case Edit Order. The system shall allow the customer to edit an unpaid order.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. The Customer should have placed an order in order to edit it.
Input	1.HTTP request = 'customer_view_order.aspx' 2. Select to Edit an Order which is unpaid. Job name = " Roger_Order_2" Final size = "20" in x "30" in Quantity to Print = "500" Stock finish = "Glossy" Stock Weight = "Heavy" Folded = "Checked" Ship = "Checked" Save Order
Expected output	The system should return to the Orders List with the saved changes.
Actual output	The system returns to the Orders list with the saved changes.

Name	**ManageUserAccount (Edit)_Test_1**
Test type	Sunny Day
Version	1.0
Use case ID	ManageUserAccount (Edit)
Purpose	The purpose of this test case is to test the ManageUserAccount (Edit). The system shall let the Administrator to edit the profile and the new profile should be updated in the database.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. User must be logged in as the Administrator
Input	1. HTTP request = 'adminView_ManageAccounts.aspx' 2. Select to Edit "Roger" email address = 'roger@yahoo.com' new password = 'abc12345' confirm new password = 'abc12345' 3.Contact Information first name = 'Roger' last name = 'Smith' company name = 'Qualcomm' phone number = '1234567890' 4.Billing Address address line 1= '110 SW 140 Ave' city = 'Miami' State = 'Florida' Zip Code = '33199' 5.Shipping Address

	address line 1= ‘110 SW 140 Ave’ city = ‘Miami’ State = ‘Florida’ Zip Code = ‘33199’
Expected output	The system should display a message that the profile was updated. For example , “Profile Updated”
Actual output	“Request Completed”

Name	**ManageUserAccount (Edit)_Test_2**
Test type	Rainy Day
Version	1.0
Use case ID	ManageUserAccount (Edit)
Purpose	The purpose of this test case is to test the ManageUserAccount (Edit). The system shall let the Administrator to edit the profile and the new profile should be updated in the database.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. User must be logged in as the Administrator
Input	1. HTTP request = 'adminView_ManageAccounts.aspx' 2. Select to Edit "Roger" email address = 'roger@yahoo.com' new password = 'abc123456789' confirm new password = 'abc123456789' 3.Contact Information first name = 'Roger' last name = 'Smith' company name = 'Qualcomm' phone number = '1234567890' 4.Billing Address address line 1= '110 SW 140 Ave' city = 'Miami' State = 'FL' Zip Code = '33199' 5.Shipping Address

	address line 1= ‘110 SW 140 Ave’ city = ‘Miami’ State = ‘Florida’ Zip Code = ‘33199’ 6.Credit Card Information Credit card # : ABCD1234EFGH5678 Card type : Apex Exp Date : ABCD1234 Security Code : ABCD Name on Card : XYZ
Expected output	The system should display a message to input the correct values.
Actual output	Server Error in '/' Application. *String or binary data would be truncated.* *The statement has been terminated.*

Name	**CreateRun_Test_1**
Test type	Sunny Day
Version	1.0
Use case ID	CreateRun
Purpose	The purpose of this test case is to test the CreateRun. The system shall let worker to create a new run and acknowledge when the run is created.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. User must be logged in as Worker
Input	1. HTTP request = 'workerView_CreatePrintRun.aspx' 2. Create Print Run Print run name = "Print Run 1" Run size = "123" x "123" Run Quantity = "20" Stock Finish = "Glossy" Stock Weight ="Heavy" Click "Create Print Run"
Expected output	The system should display a message that a new Print Run was created. For example " New Print Run Created"
Actual output	"Print Run Created Sucessfully"

Name	**EditRun_Test_1**
Test type	Sunny Day
Version	1.0
Use case ID	EditRun
Purpose	The purpose of this test case is to test the EditRun. The system shall let the worker to add orders to a PrintRun.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. User must be logged in as Worker
Input	1. HTTP request = 'workerView_AddToPrintRun.aspx' 2. Select "Print Run 1" from the drop down box. Select Order 1 Select Order 2 Select Order 3 Click "Move Orders"
Expected output	The system should display a message that orders were moved to the selected Print Run. For example, " Orders moved to Print Run 1 Successfully"
Actual output	"Your orders have been added to the print run successfully"

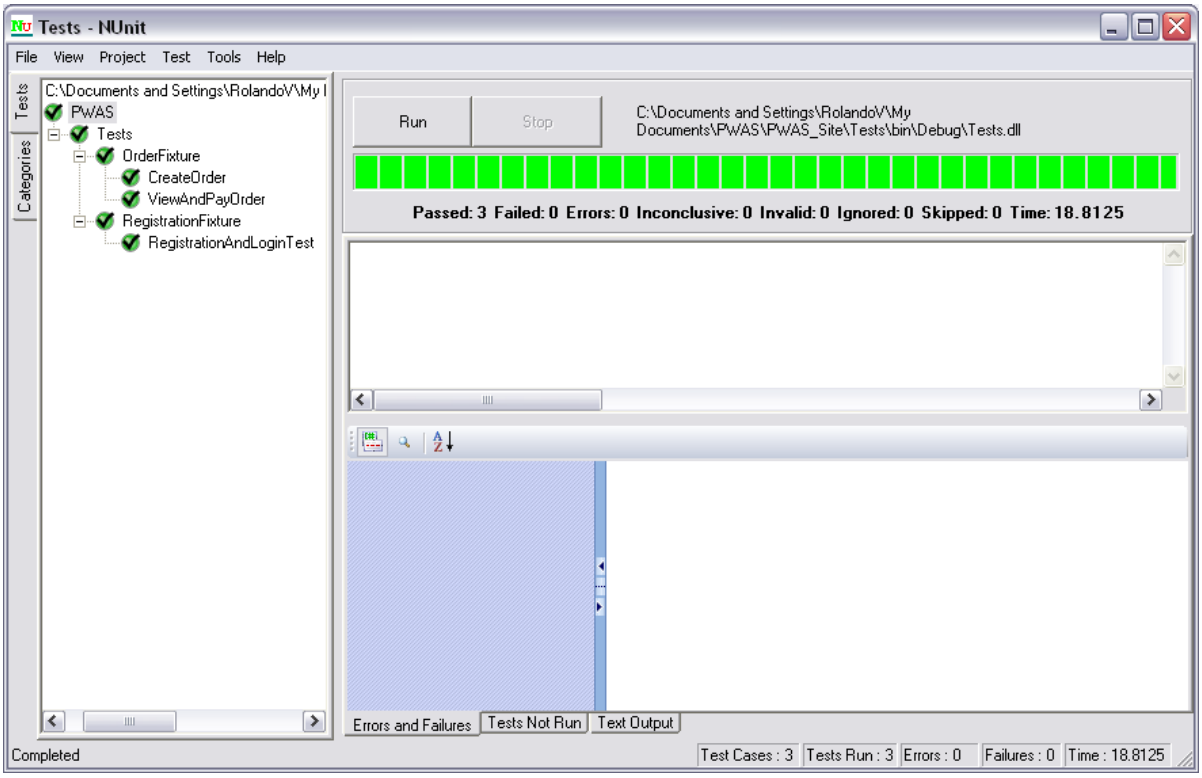
Name	**UpdateRunStatus_Test_1**
Test type	Sunny Day
Version	1.0
Use case ID	UpdateRunStatus
Purpose	The purpose of this test case is to test the UpdateRunStatus. The system shall let the worker to update the status of the Order.
Test setup (preconditions)	1. All the classes and subsystems that interact with this test case should work properly. 2. The hardware required to run this test case should be properly set up. 3. User must be logged in as Worker
Input	1. HTTP request = 'view_print_run.aspx' 2. Select "Print Run 1" from the list. Click on the dropdown box and select the status to "Pre-Printing" if the Print Run is ready for Printing. Click Update Click on the dropdown box and update the status to "Printing" if the Print Run is in the Printing phase. Click Update Click on the dropdown box and update the status to "Finishing" if the Print Run is in the Finishing phase. Click Update Click on the dropdown bo and update the status to "Shipping" if the Order is in Shipping phase. Click Update Click on the dropdown box and update the status to "Closed" if the Order is closed. Click Update
Expected output	The system should update the status to the new status selected.
Actual output	System updates the status to the selected status.

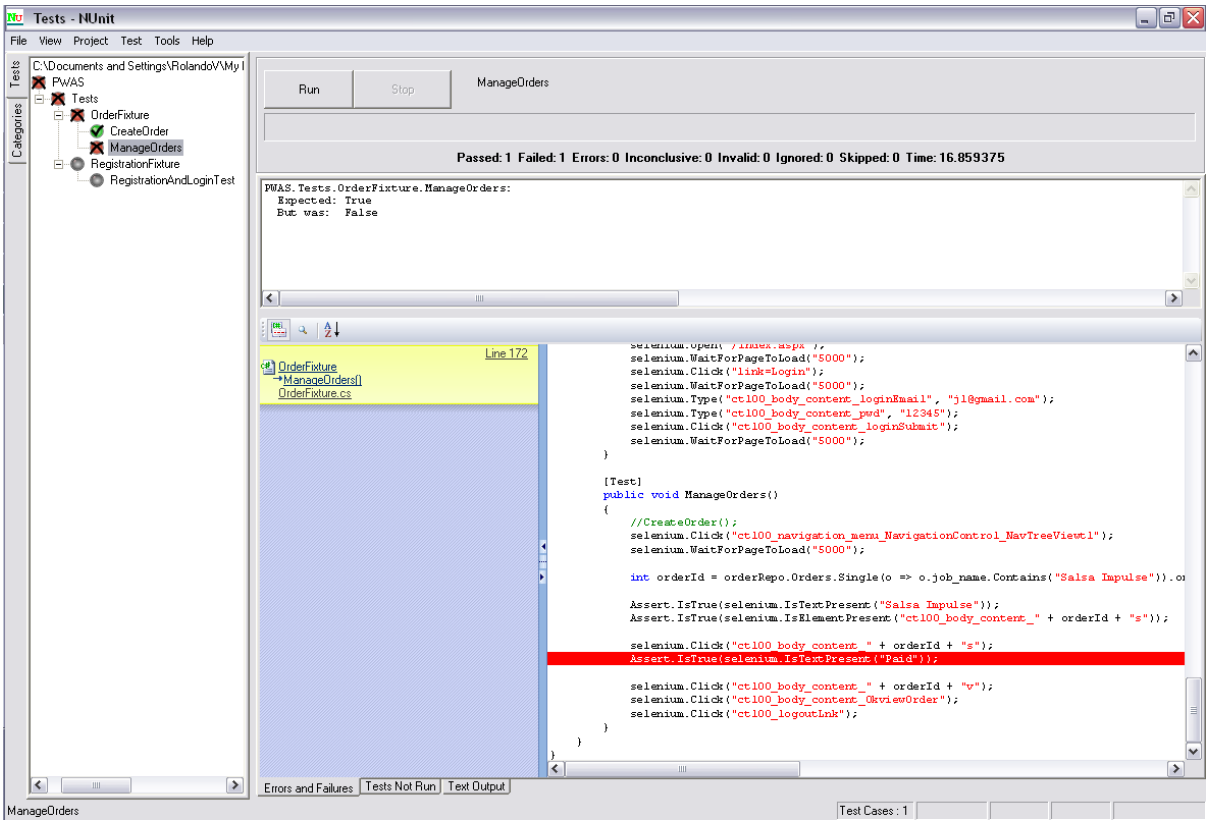
Automated Unit Tests using the NUnit tool

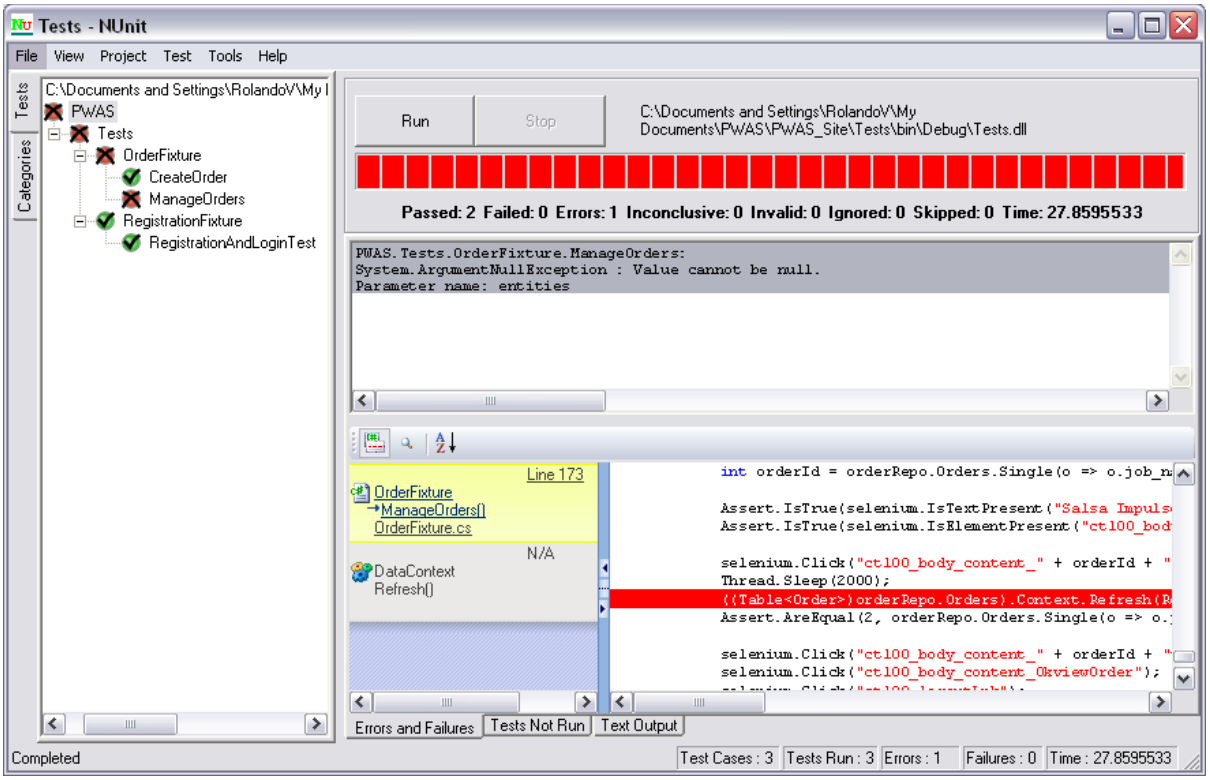
NUnit is a unit-testing framework for all .Net languages. Initially ported from JUnit, the current production release, version 2.5, is the sixth major release of this xUnit based unit testing tool for Microsoft .NET. It is written entirely in C# and has been completely redesigned to take advantage of many .NET language features, for example custom attributes and other reflection related capabilities. NUnit brings xUnit to all .NET languages. *1

*1 - http://www.nunit.org/index.php

Our unit testing was done using this tool, which includes features to test the front end (UI), as well as the backend (server side code). Below are a couple of sample screen shots to give an idea of how the tool works, and the output it produces after it's run.




Tests - NUnit
File View Project Test Tools Help
C:\Documents and Settings\RolandoV\My I
PWAS
Tests
OrderFixture
CreateOrder
ManageOrders
RegistrationFixture
RegistrationAndLoginTest
Run
Stop
ManageOrders
Passed: 1 Failed: 1 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Time: 16.859375
PWAS.Tests.OrderFixture.ManageOrders:
Expected: True
But was: False
Line 172
OrderFixture
ManageOrders()
OrderFixture.cs
selenium.WaitForPageToLoad("5000");
selenium.Click("link=Login");
selenium.WaitForPageToLoad("5000");
selenium.Type("ctl00_body_content_loginEmail", "jl@gmail.com");
selenium.Type("ctl00_body_content_pwd", "12345");
selenium.Click("ctl00_body_content_loginSubmit");
selenium.WaitForPageToLoad("5000");
[Test]
public void ManageOrders()
{
//CreateOrder();
selenium.Click("ctl00_navigation_menu_NavigationControl_NavTreeViewt1");
selenium.WaitForPageToLoad("5000");
int orderId = orderRepo.Orders.Single(o => o.job_name.Contains("Salsa Impulse")).or
Assert.IsTrue(selenium.IsTextPresent("Salsa Impulse"));
Assert.IsTrue(selenium.IsElementPresent("ctl00_body_content_" + orderId + "s"));
selenium.Click("ctl00_body_content_" + orderId + "s");
Assert.IsTrue(selenium.IsTextPresent("Paid"));
selenium.Click("ctl00_body_content_" + orderId + "v");
selenium.Click("ctl00_body_content_OkviewOrder");
selenium.Click("ctl00_logoutLnk");
Errors and Failures
Tests Not Run
Text Output
ManageOrders
Test Cases : 1


Tests - NUnit
File View Project Test Tools Help
C:\Documents and Settings\RolandoV\My I
PWAS
Tests
OrderFixture
CreateOrder
ManageOrders
RegistrationFixture
RegistrationAndLoginTest
Run
Stop
C:\Documents and Settings\RolandoV\My Documents\PWAS\PWAS_Site\Tests\bin\Debug\Tests.dll
Passed: 2 Failed: 0 Errors: 1 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Time: 27.8595533
PWAS.Tests.OrderFixture.ManageOrders:
System.ArgumentNullException : Value cannot be null.
Parameter name: entities
Line 173
OrderFixture
ManageOrders()
OrderFixture.cs
N/A
DataContext
Refresh()
int orderId = orderRepo.Orders.Single(o => o.job_n
Assert.IsTrue(selenium.IsTextPresent("Salsa Impuls
Assert.IsTrue(selenium.IsElementPresent("ctl00_bod
selenium.Click("ctl00_body_content_" + orderId +
Thread.Sleep(2000);
((Table<Order>)orderRepo.Orders).Context.Refresh(R
Assert.AreEqual(2, orderRepo.Orders.Single(o => o.
selenium.Click("ctl00_body_content_" + orderId +
selenium.Click("ctl00_body_content_OkviewOrder");
Errors and Failures
Tests Not Run
Text Output
Completed
Test Cases : 3
Tests Run : 3
Errors : 1
Failures : 0
Time : 27.8595533