**VT Foundation Portal**

**UI Testing Documentation/Use Case Overview**

**by**

**Brandon Shinault**

**Summer 2014 Internship**

Solution and Project Name: FundPortalUITests

**Documentation**

|  |  |
| --- | --- |
| **Class Name**: UITestSetup | |
| **Class field name** | **Description** |
| private string url | Contains the url value that the driver will initially navigate to |
| public string targetUser |  |
| **Class property** | **Description** |
| public IWebDriver driver { get; set; } | Declares a driver of type IWebDriver |
| **Class constructor** | **Description** |
| public UITestSetup() | Sets the field variable url statically |
| **Class method name** | **Description** |
| public void FirefoxSetup() | Instantiates a new FirefoxDriver(). Then navigates the Driver to the statically instantiated url of the class field url. Then maximizes the browser window. The FirefoxDriver is built in and does not need to be passed a FirefoxDriver location. |
| public void ChromeSetup() | Same as FirefoxSetup() except instantiates a new ChromeDriver. The ChomeDriver must be passed the location of the ChromeDriver. |
| public void IeSetup() | Same as FirefoxSetup() except instantiates a new InternetExplorerDriver. The InternetExplorerDriver must be passed the location of the InternetExplorerDriver. |
| Internal void LoginCas() | Instantiates a new Credentials object of type ICredentials. The Credentials object is a class that contains the PID (username) and password in order to login to the Foundation Portal system. The username and password are submitted to the page and then the submit button is clicked. |
| public void takeScreenShot() | Gets the screen shot. Gets the current date of month, day, year, hour, minute, second. The file name is created with the value that's passed into the takeScreenShot method concatenated with the current date. Then the file is saved in the C:\FundPortalScreenShots directory as a .png file type. |

**Use Cases**

FundPortalUITests: AdministratorRolesManageFundsTests

|  |  |
| --- | --- |
| **Use Case Name**: AdministratorRolesManageFundsTests | **Use Case Type**: Testing |
| **Test Cases**:   * ManageFunds\_Granted\_Then\_Funds\_Browse\_Manage\_Visible\_Accessible() * ManageFunds\_NotGranted\_Then\_Funds\_Browse\_Manage\_NotVisible\_NotAccessible() | |
| **Primary Actor**: Tester | **Interested Stakeholders**: Foundation Portal Users, Administrators, Developers, and Testers |
| **Brief Description**: This use case describes the functionality of the AdministratorRolesManageFundsTests TestFixture/Test Class. The test checks if the user can access the funds dropdown menu given that the user has been granted or denied access to manage funds. The funds dropdown menu contains a link for browse and manage of funds. | |
| **Goal**:   * If user has been granted access to manage funds then the funds dropdown menu will be visible and accessible. * If user hasn't been granted access to manage funds then the funds dropdown menu will not be visible or accessible. | |
| **Success Measurement**: Test Case Assertions pass. | |
| **Precondition**:   * Successful UITestSetup initialization | |
| **Trigger**: Run the test class | |
| **Relationships**:  **Depends on**: NUnit, Selenium WebDriver, Selenium Support Classes, UITestSetup | |
| **Typical flow of events**:   1. This test is initialized and begins to run 2. Upon initialization this test instance will contain a new instance of UITestSetup and a targetUser string which is the user to be targeted for privileges to manage funds 3. NUnit [Setup] function is ran at start of each test, which calls uiTestSetup.FirefoxDriver() and then uiTestSetup.LoginCas() 4. The first test case (ManageFunds\_Granted\_Then\_Funds\_Browse\_Manage\_Visible\_Accessible()) begins 5. This first test finds the "Users" link and then clicks it 6. The current page is now the manage users page 7. It finds the targeted user (string targetUser), and clicks its link 8. The current page is now the targeted users privileges page 9. It then finds the Manage Funds checkbox in the Administrator Roles section of the current page 10. If the Manage Funds checkbox isn't selected then it's selected by clicking it, then the save button is clicked 11. Ctrl + Shift + r is sent as a string to the body of the page in order to refresh it 12. Call IsFundsDropdownVisible() in order to confirm that we can access this dropdown which includes the links for browse and manage 13. Assert that fundsDropdownMenuVisible -- if so then test passes 14. If test fails -- call takeScreenShot() and Assert that test failed and pass to this assertion the exceptions stack trace 15. The second test case (ManageFunds\_NotGranted\_Then\_Funds\_Browse\_Manage\_NotVisible\_NotAccessible()) is mostly the same as the first 16. Except for in the second test it unchecks the Manage Funds text box if it's selected then we click the save button 17. And we later Assert that fundDropdownMenuVisible is false 18. After each test is ran the test classes [Teardown] function is called 19. This Teardown function disposes of the current instance of the IWebDriver by closing it | |
| **Assumptions**: Manage Funds logic and user interaction hasn't been modified as of 6.25.2014 | |