Додаток А

Графічний матеріал кваліфікаційної роботи

ГЮИК.501610.008

(позначення документу)

ЗАТВЕРДЖЕНО
ГЮИК.501610.008 – ЛЗ

Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину

Графічний матеріал

ГЮИК. 501610.008 – ЛЗ

ЛИСТІВ 24

2021 р.

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник кваліфікаційної роботи
проф. Мінухін С.В.

Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

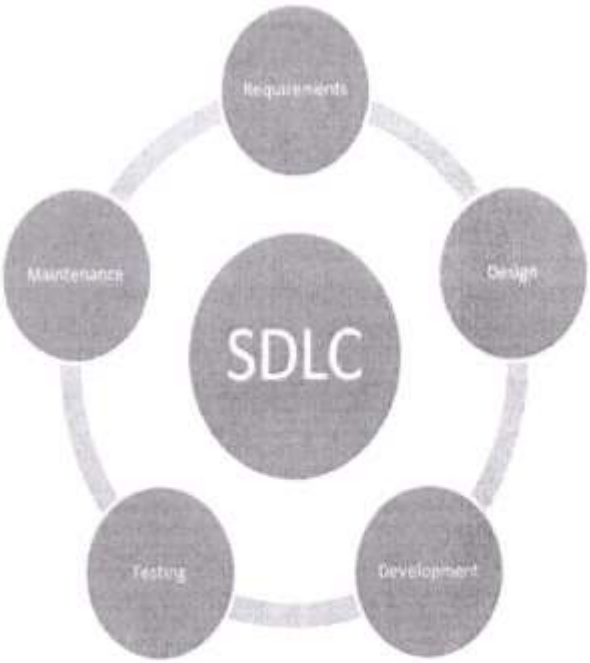
ГЮИК.501610.008 – ЛЗ

РОЗРОБИВ:
ст. гр. СПРм-20-1
Мороз М.Ю.

2021 р.

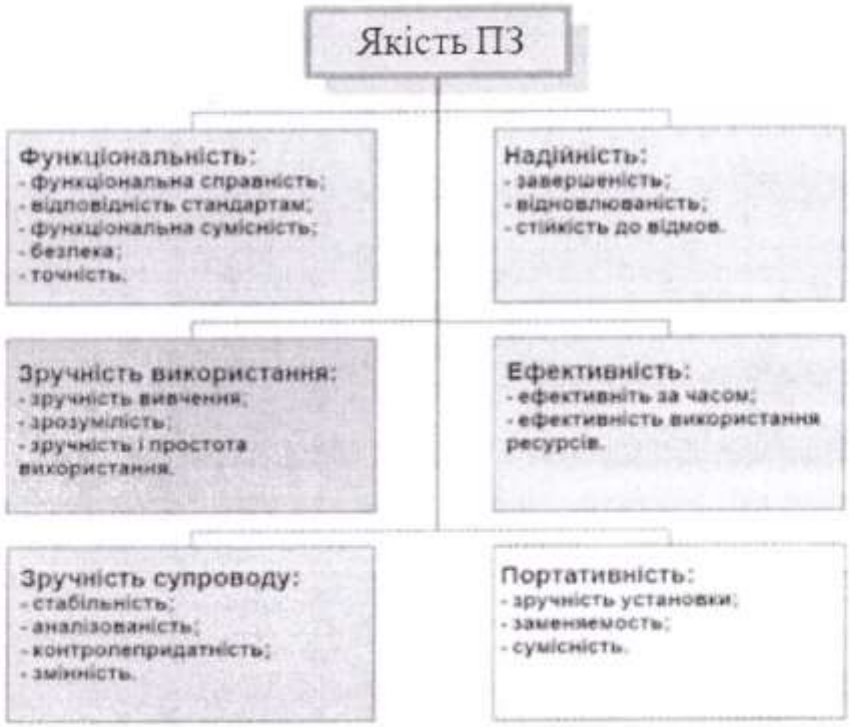
ЖИТТЄВИЙ ЦИКЛ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Збір та аналіз вимог
2. Дизайн
3. Розробка
4. Тестування
5. Підтримка



Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

МОДЕЛЬ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



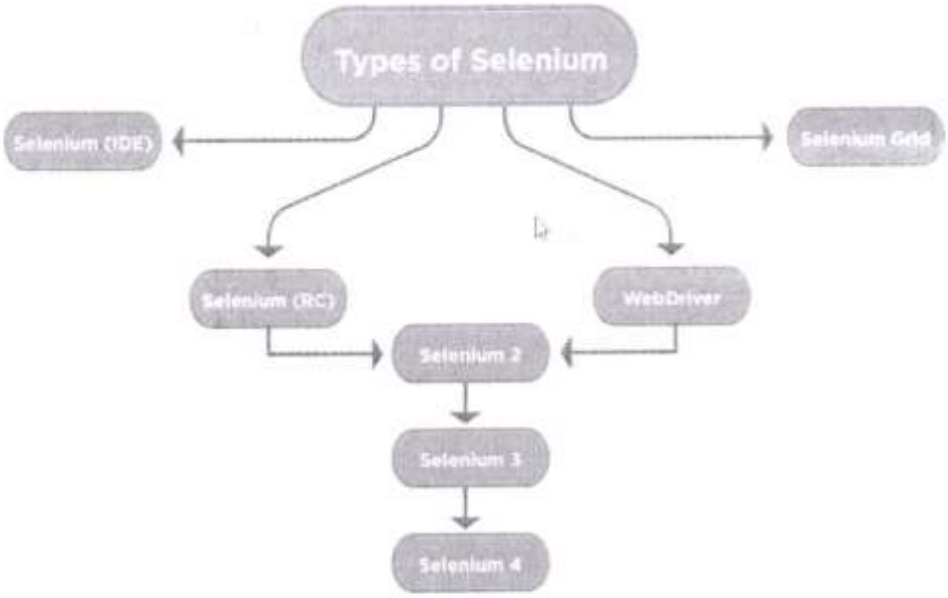
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 2*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ПРОЦЕС ЗАБЕЗПЕЧЕННЯ ЯКОСТІ

СТРУКТУРА ВЕБ-ДОДАТКУ

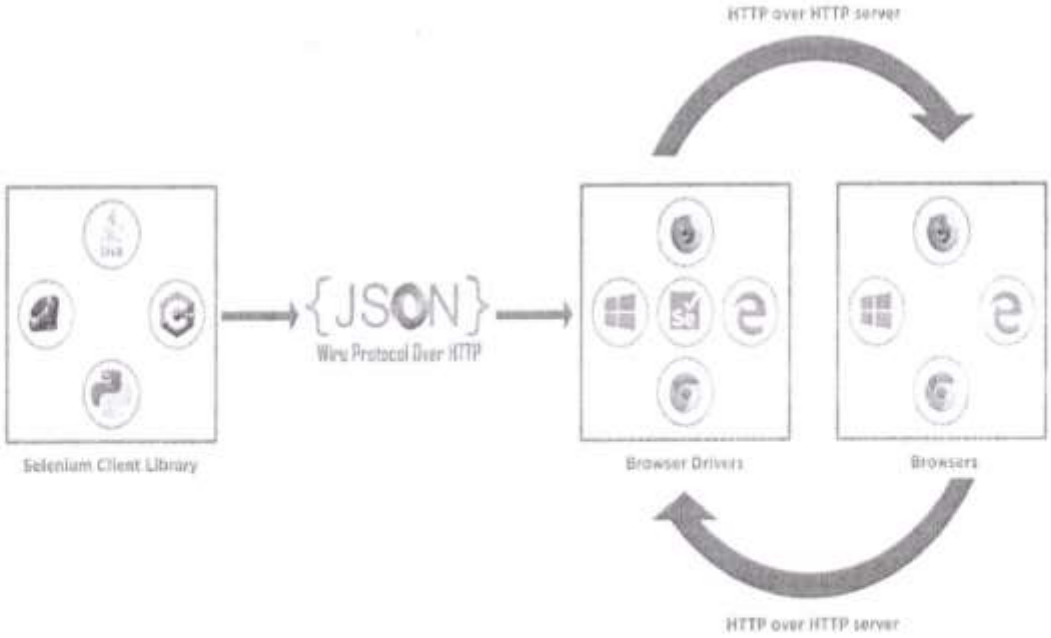
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 4*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

КОМПОНЕНТИ SELENIUM SUITE



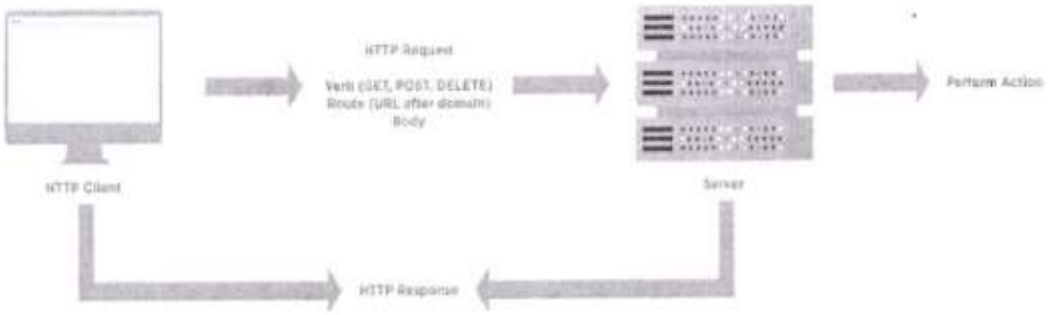
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 5*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

КОМПОНЕНТИ SELENIUM WEBDRIVER



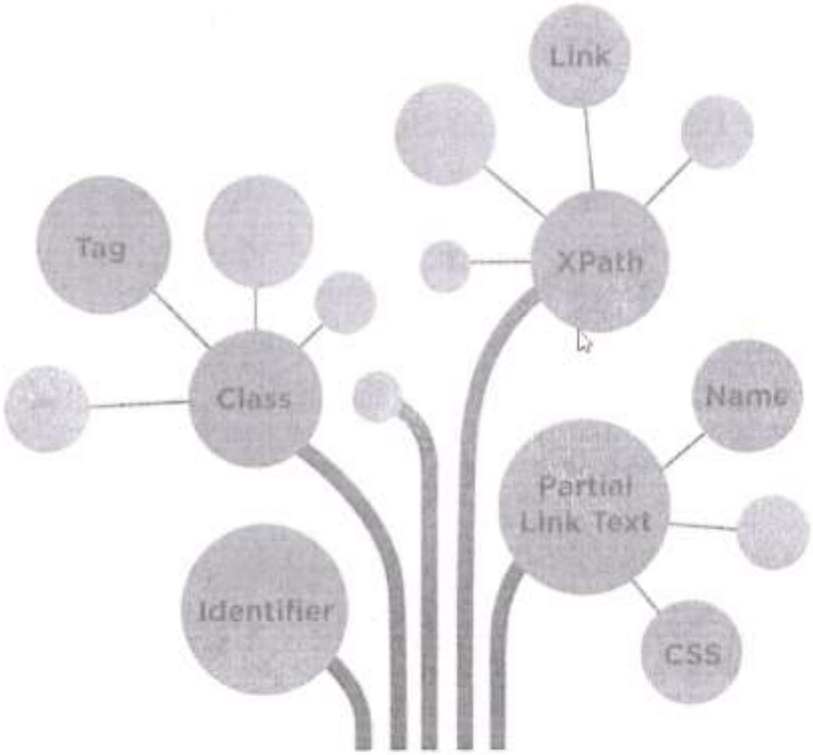
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 6*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РОБОЧИЙ ПРОЦЕС SELENIUM WEBDRIVER



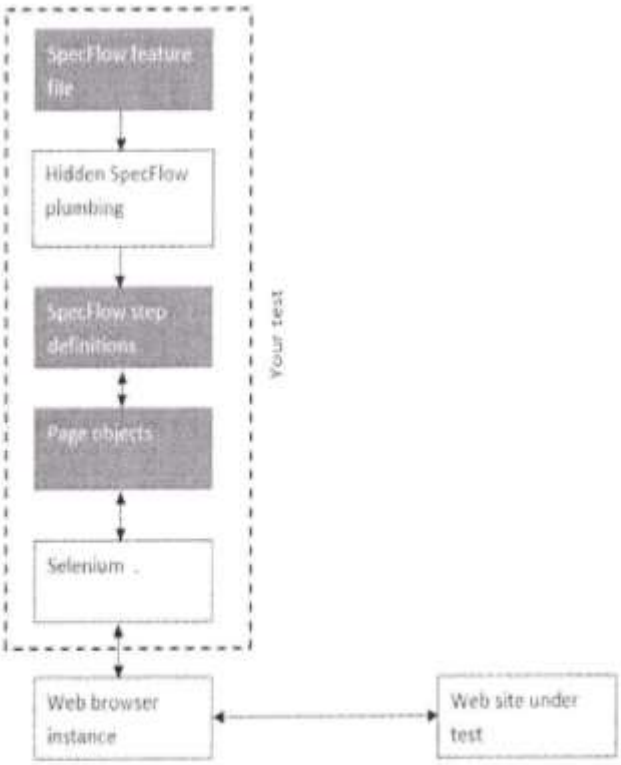
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 7*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ТИПИ ЛОКАТОРІВ



Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 8*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ВЗАЄМОДІЯ SPECFLOW ТА BDD З ВЕБ-ДОДАТКОМ



Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 9*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

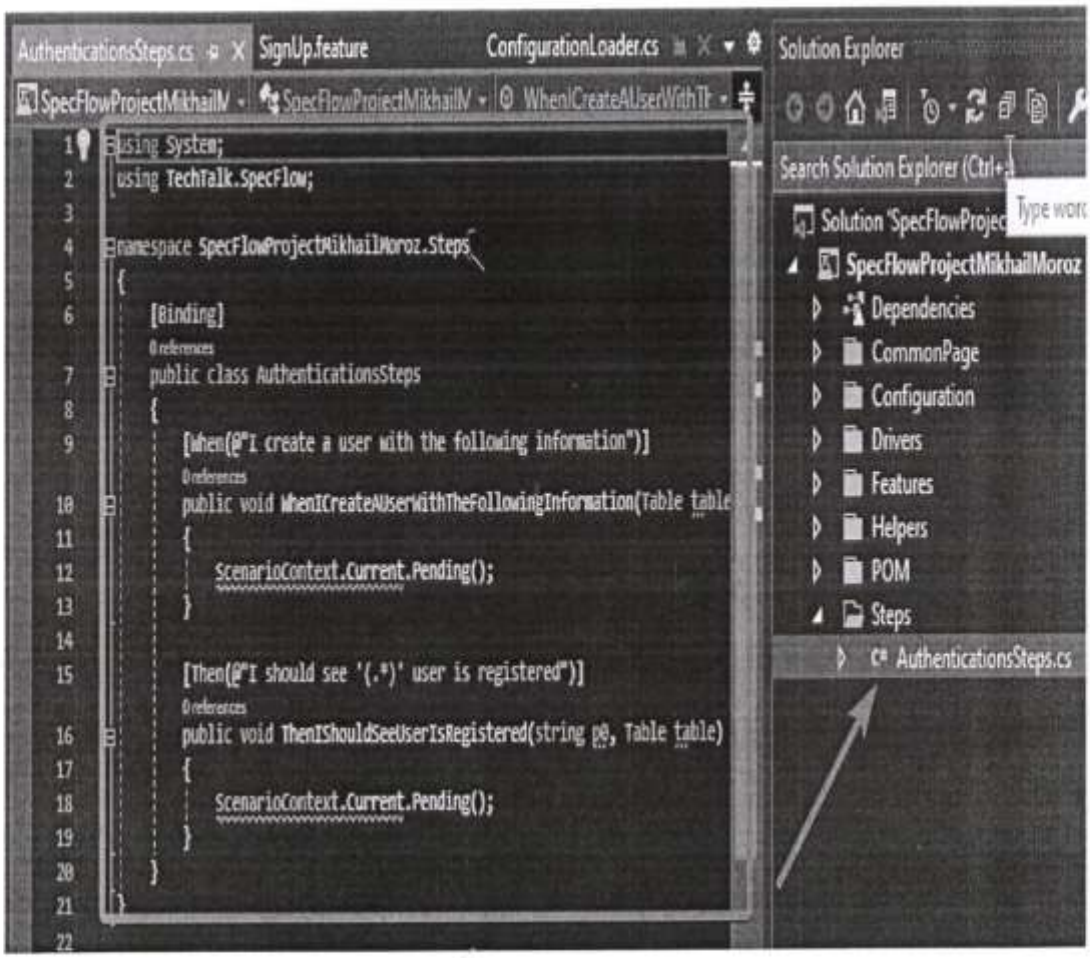
ТЕСТОВИЙ СЦЕНАРІЙ РЕЄСТРАЦІЇ У ВЕБ-ДОДАТОК

SignUp.feature

```
Feature: Sign Up
As a user
I want to be able to sign in
So that I have an ability to do shopping

Background:
    Given I navigate to the main page on the site

@Sign in
Scenario: As a user I want to be able to create an account
    When I navigate to the authentication page
    And I create a user with the following information
        | Field                | Value                  |
        | Email                | mikhailmoroz2@gmail.com |
        | Personal First Name  | Mikhail                |
        | Personal Last Name   | Moroz                  |
        | Password             | Qqwerty1!!             |
        | Day of Birth         | 18                     |
        | Month of Birth       | May                    |
        | Year of Birth        | 1999                   |
        | Address First Name   | Mikhail                |
        | Address Last Name    | Moroz                  |
        | Address              | St Main                |
        | City                 | Kharkiv                |
        | State                | California             |
        | Zip code             | 62002                  |
        | Phone                | 0991231232             |
    Then I should see 'Mikhail Moroz' user is registered
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 10*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ЗГЕНЕРОВАНИЙ КЛАС ВИЗНАЧЕНИХ КРОКІВ З ШАБЛОНАМИ МЕТОДІВ



Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 11*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ЛОКАТОРИ ЕЛЕМЕНТІВ ДЛЯ РЕЄСТРАЦІЇ

```
using NoOi;

namespace SpecFlowProjectMikhailMoroz.POM
{
    23 references
    class AuthenticationPage
    {
        public static string emailAddressInputField = "//input[@id='email_create']";
        public static string createAnAccountButton = "//button[@id='SubmitCreate']";
        public static string genderMale = "//input[@id='id_gender1']";
        public static string customerFirstName = "//input[@id='customer_firstname']";
        public static string customerLastName = "//input[@id='customer_lastname']";
        public static string password = "//input[@id='passwd']";
        public static string addressFirstName = "//input[@id='firstname']";
        public static string addressLastName = "//input[@id='lastname']";
        public static string address = "//input[@id='address1']";
        public static string city = "//input[@id='city']";
        public static string zipCode = "//input[@id='postcode']";
        public static string country = "//select[@id='id_country']/option[text()='united states']";
        public static string mobilePhone = "//input[@id='phone_mobile']";
        public static string registerAccount = "//button[@id='submitAccount']";
        public static string daysCombobox = "//select[@id='days']";
        public static string monthCombobox = "//select[@id='months']";
        public static string yearsCombobox = "//select[@id='years']";
        public static string loginEmailField = "//form[@id='login_form']//input[@id='email']";
        public static string loginPasswordField = "//form[@id='login_form']//input[@id='passwd']";
        public static string loginSignInButton = "//form[@id='login_form']//button[@id='SubmitLogin']";
        1 reference
        public static string usernameOnHeader(string username) => $"//div[@class='header_user_info']/a/span[text()='{username}']";
        1 reference
        public static string dayOfBirth(string day) => $"//select[@id='days']/option[contains(text(),'{day}')]";
        1 reference
        public static string monthOfBirth(string month) => $"//select[@id='months']/option[contains(text(),'{month}')]";
        1 reference
        public static string yearOfBirth(string year) => $"//select[@id='years']/option[contains(text(),'{year}')]";
        1 reference
        public static string state(string state) => $"//select[@id='id_state']/option[contains(text(),'{state}')]";
```

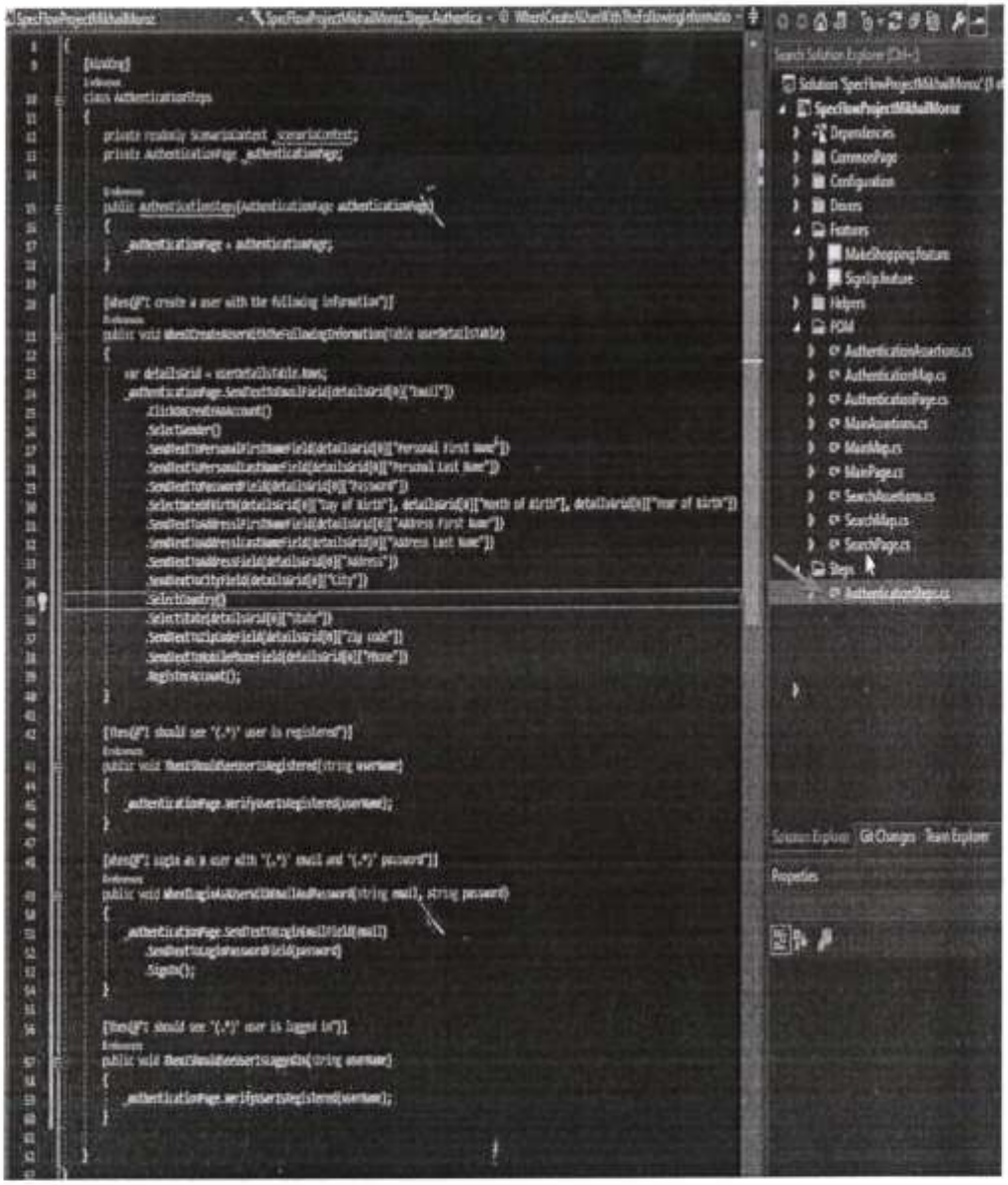
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 12*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗОВАНІ МЕТОДИ КЛАСУ AUTHENTICATIONPAGE

```
0 references
public AuthenticationPage SendTextToEmailField(string emailAddress)
{
    _authenticationMap.SendTextToInput(emailAddressInputField, emailAddress);
    return this;
}
0 references
public AuthenticationPage ClickOnCreateAnAccount()
{
    _authenticationMap.ClickButton(createAnAccountButton);
    return this;
}
0 references
public AuthenticationPage SelectGender()
{
    _authenticationMap.ClickButton(genderMale);
    return this;
}
0 references
public AuthenticationPage SendTextToPersonalFirstNameField(string firstName)
{
    _authenticationMap.SendTextToInput(customerFirstName, firstName);
    return this;
}
0 references
public AuthenticationPage SendTextToPersonalLastNameField(string lastName)
{
    _authenticationMap.SendTextToInput(customerLastName, lastName);
    return this;
}
0 references
public AuthenticationPage SendTextToPasswordField(string userPassword)
{
    _authenticationMap.SendTextToInput(password, userPassword);
    return this;
}

0 references
public AuthenticationPage SelectDateOfBirth(string day, string month, string year)
{
    _authenticationMap.ClickButton(dayOfBirth(day));
    _authenticationMap.ClickButton(monthOfBirth(month));
    _authenticationMap.ClickButton(yearOfBirth(year));
    return this;
}

0 references
public AuthenticationPage SendTextToAddressFirstNameField(string firstName)
{
    _authenticationMap.SendTextToInput(addressFirstName, firstName);
    return this;
}

0 references
public AuthenticationPage SendTextToAddressLastNameField(string lastName)
{
    _authenticationMap.SendTextToInput(addressLastName, lastName);
    return this;
}

0 references
public AuthenticationPage SendTextToAddressField(string addressName)
{
    _authenticationMap.SendTextToInput(address, addressName);
    return this;
```

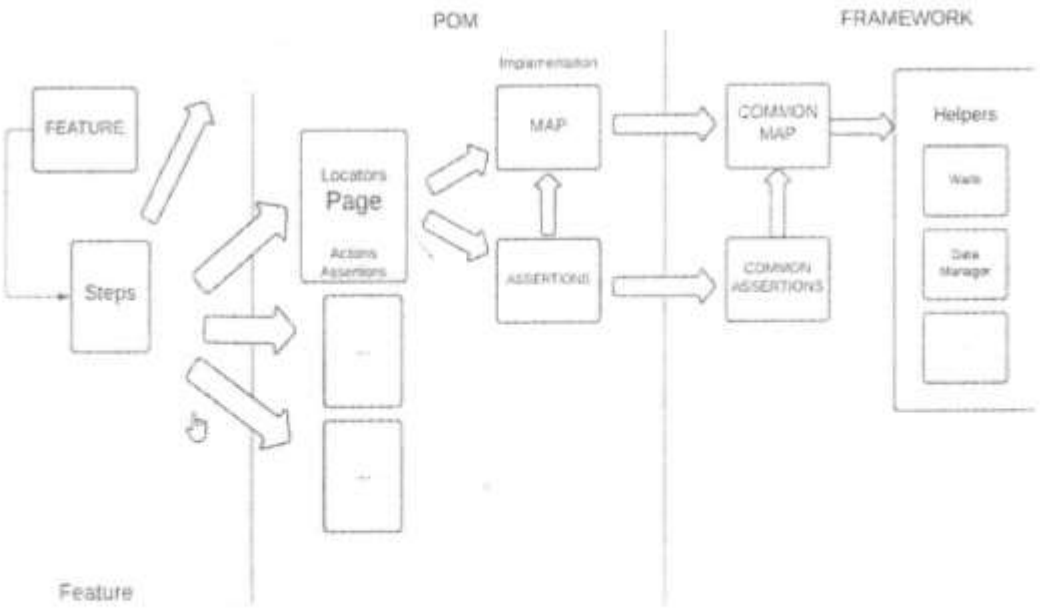
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 13*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗОВАНІ МЕТОДИ ВИЗНАЧЕННЯ КРОКІВ AUTHENTICATIONSTEPS



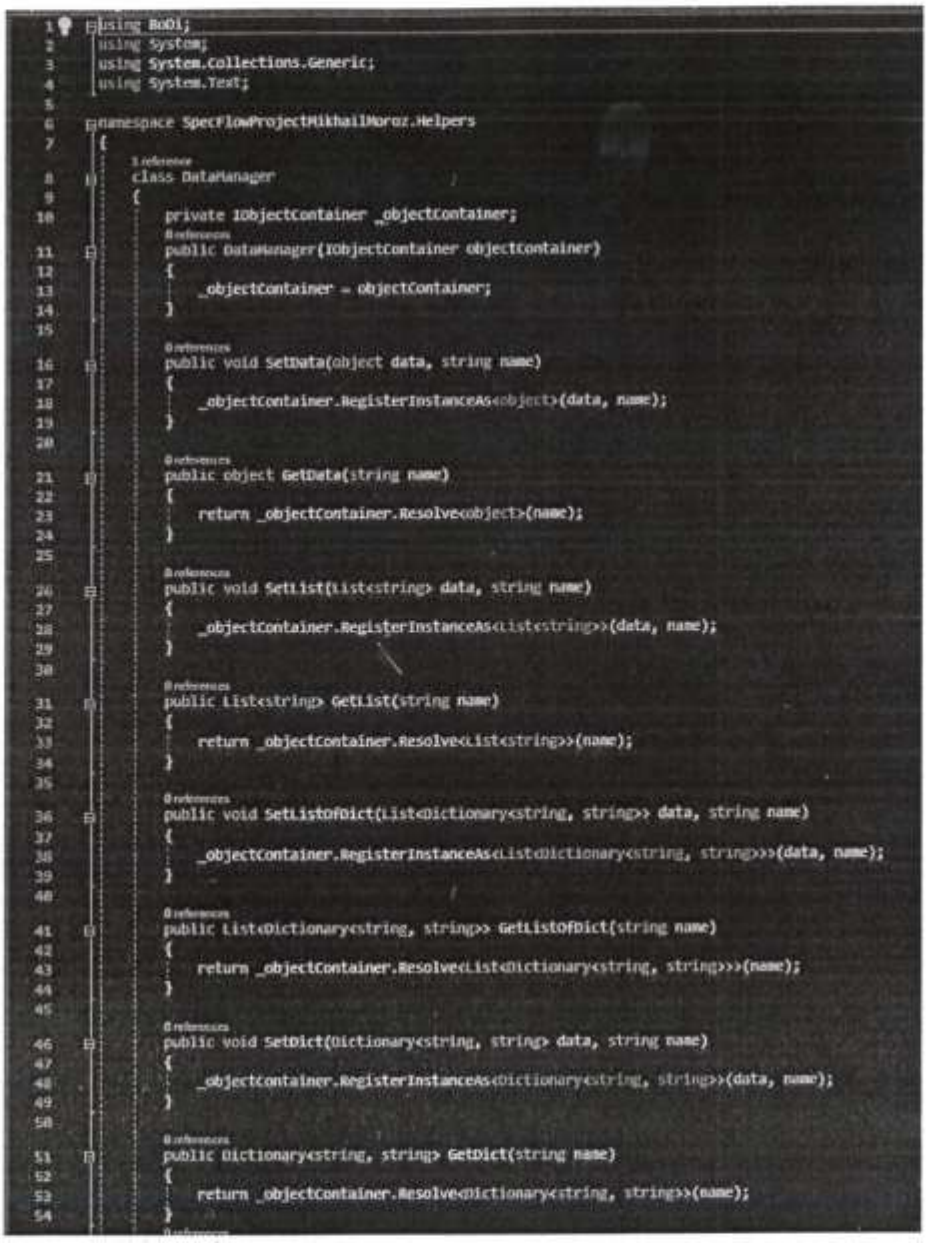
Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 14*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

ДІАГРАМА СТРУКТУРИ ТА ВЗАЄМОДІЇ КОМПОНЕНТІВ



Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 15*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

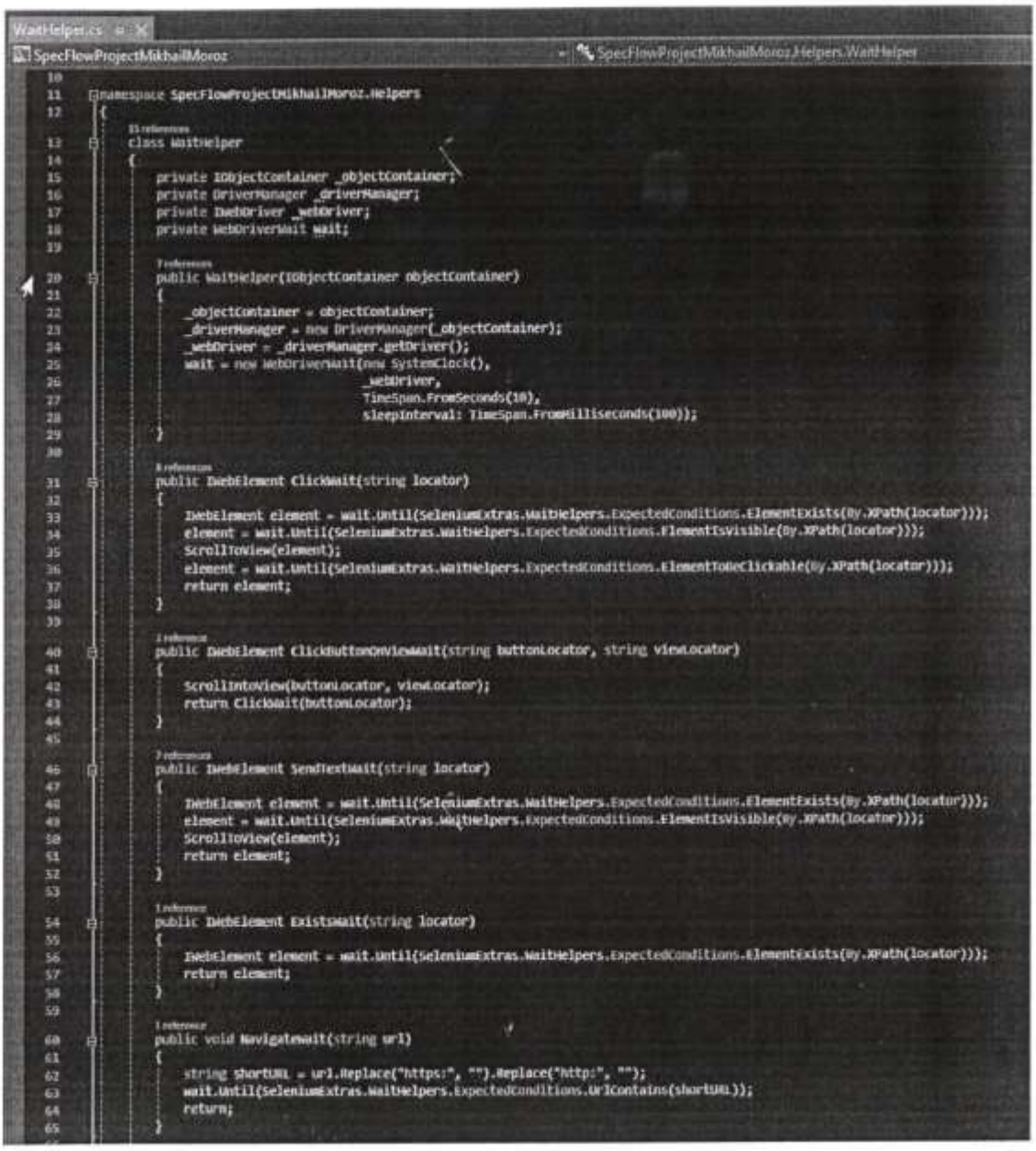
РЕАЛІЗАЦІЯ КЛАСУ DATAMANAGER



```
using BoDi;
using System;
using System.Collections.Generic;
using System.Text;

namespace SpecFlowProjectMikhailMoroz.Helpers
{
    1 reference
    class DataManager
    {
        private IObjectContainer _objectContainer;
        0 references
        public DataManager(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
        }

        0 references
        public void SetData(object data, string name)
        {
            _objectContainer.RegisterInstanceAs<object>(data, name);
        }

        0 references
        public object GetData(string name)
        {
            return _objectContainer.Resolve<object>(name);
        }

        0 references
        public void SetList(List<string> data, string name)
        {
            _objectContainer.RegisterInstanceAs<List<string>>(data, name);
        }

        0 references
        public List<string> GetList(string name)
        {
            return _objectContainer.Resolve<List<string>>(name);
        }

        0 references
        public void SetListOfDict(List<Dictionary<string, string>> data, string name)
        {
            _objectContainer.RegisterInstanceAs<List<Dictionary<string, string>>>(data, name);
        }

        0 references
        public List<Dictionary<string, string>> GetListOfDict(string name)
        {
            return _objectContainer.Resolve<List<Dictionary<string, string>>>(name);
        }

        0 references
        public void SetDict(Dictionary<string, string> data, string name)
        {
            _objectContainer.RegisterInstanceAs<Dictionary<string, string>>(data, name);
        }

        0 references
        public Dictionary<string, string> GetDict(string name)
        {
            return _objectContainer.Resolve<Dictionary<string, string>>(name);
        }
        0 references
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 16*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

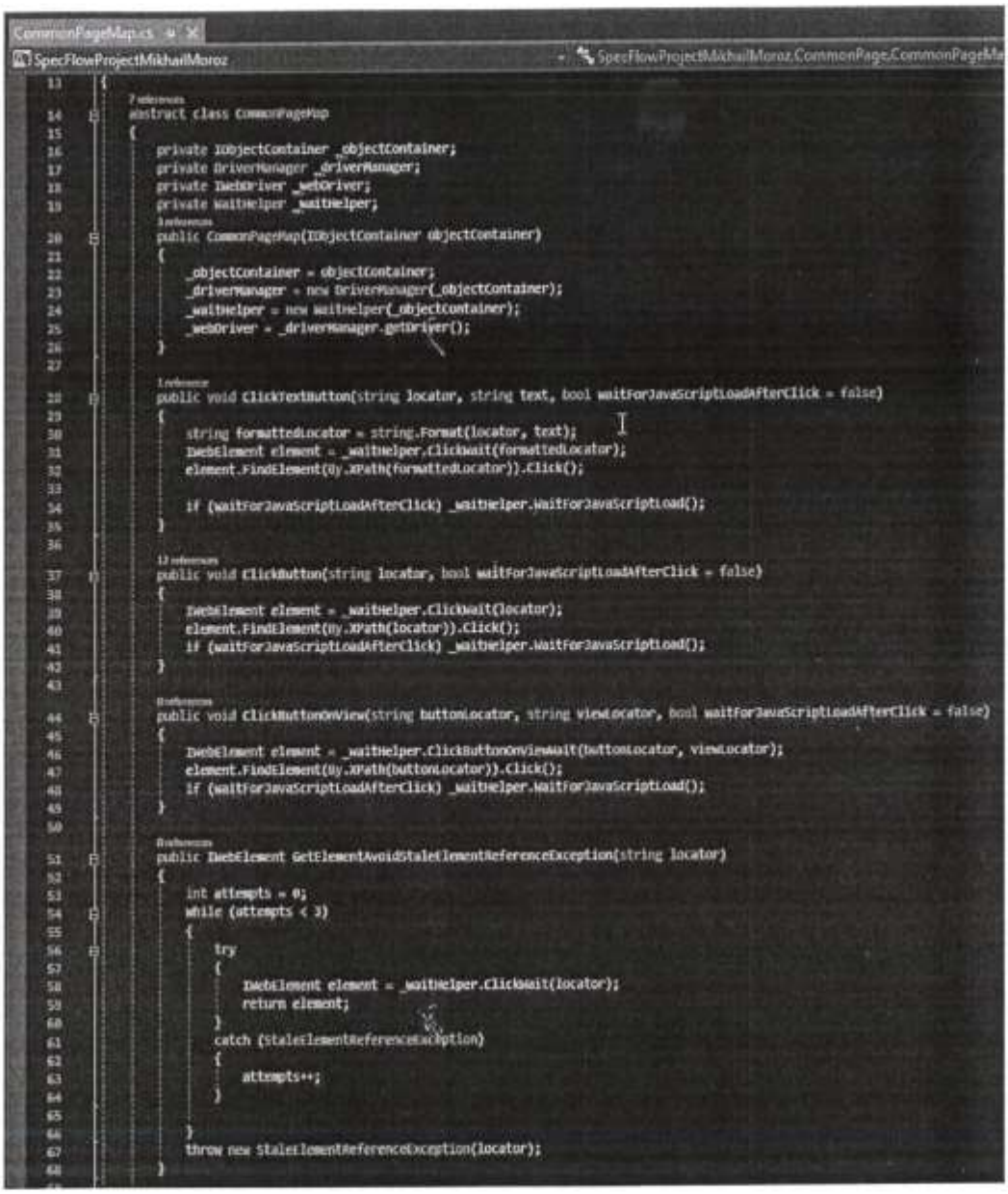
РЕАЛІЗАЦІЯ КЛАСУ WAITHELPER



```
WaitHelper.cs
SpecFlowProjectMikhailMoroz                    SpecFlowProjectMikhailMoroz.Helpers.WaitHelper

namespace SpecFlowProjectMikhailMoroz.Helpers
{
    0 references
    class WaitHelper
    {
        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;
        private WebDriverWait wait;

        1 reference
        public WaitHelper(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            _driverManager = new DriverManager(_objectContainer);
            _webDriver = _driverManager.getDriver();
            wait = new WebDriverWait(new SystemClock(),
                                     _webDriver,
                                     TimeSpan.FromSeconds(10),
                                     sleepInterval: TimeSpan.FromMilliseconds(100));
        }

        0 references
        public IWebElement ClickWait(string locator)
        {
            IWebElement element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator)));
            element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.XPath(locator)));
            ScrollToView(element);
            element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementToBeClickable(By.XPath(locator)));
            return element;
        }

        1 reference
        public IWebElement ClickButtonOnViewWait(string buttonLocator, string viewLocator)
        {
            ScrollIntoView(buttonLocator, viewLocator);
            return ClickWait(buttonLocator);
        }

        1 reference
        public IWebElement SendTextWait(string locator)
        {
            IWebElement element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator)));
            element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.XPath(locator)));
            ScrollToView(element);
            return element;
        }

        1 reference
        public IWebElement ExistsWait(string locator)
        {
            IWebElement element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator)));
            return element;
        }

        1 reference
        public void NavigateWait(string url)
        {
            string shortURL = url.Replace("https:", "").Replace("http:", "");
            wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.UrlContains(shortURL));
            return;
        }
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 17*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

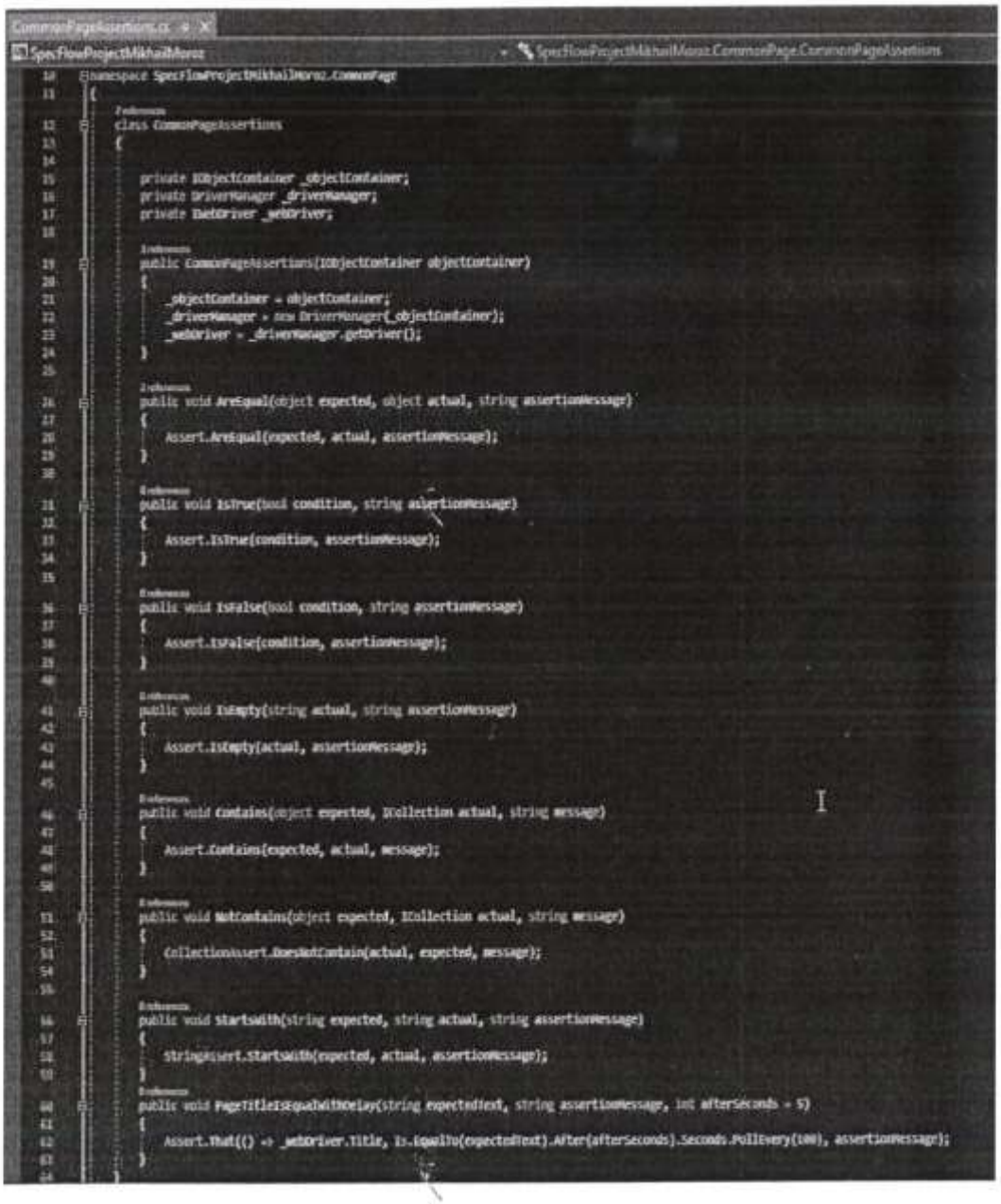
РЕАЛІЗАЦІЯ КЛАСУ COMMONPAGEMAP



```
CommonPageMap.cs
SpecFlowProjectMikhailMoroz
SpecFlowProjectMikhailMoroz.CommonPage.CommonPageMa
{
    abstract class CommonPageMap
    {
        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;
        private WaitHelper _waitHelper;
        public CommonPageMap(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            _driverManager = new DriverManager(_objectContainer);
            _waitHelper = new WaitHelper(_objectContainer);
            _webDriver = _driverManager.getDriver();
        }

        public void ClickTextButton(string locator, string text, bool waitForJavaScriptLoadAfterClick = false)
        {
            string formattedLocator = string.Format(locator, text);
            IWebElement element = _waitHelper.ClickWait(formattedLocator);
            element.FindElement(By.XPath(formattedLocator)).Click();

            if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
        }

        public void ClickButton(string locator, bool waitForJavaScriptLoadAfterClick = false)
        {
            IWebElement element = _waitHelper.ClickWait(locator);
            element.FindElement(By.XPath(locator)).Click();
            if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
        }

        public void ClickButtonOnView(string buttonLocator, string viewLocator, bool waitForJavaScriptLoadAfterClick = false)
        {
            IWebElement element = _waitHelper.ClickButtonOnViewWait(buttonLocator, viewLocator);
            element.FindElement(By.XPath(buttonLocator)).Click();
            if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
        }

        public IWebElement GetElementAvoidStaleElementReferenceException(string locator)
        {
            int attempts = 0;
            while (attempts < 3)
            {
                try
                {
                    IWebElement element = _waitHelper.ClickWait(locator);
                    return element;
                }
                catch (StaleElementReferenceException)
                {
                    attempts++;
                }

            }
            throw new StaleElementReferenceException(locator);
        }
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 18*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗАЦІЯ КЛАСУ COMMONPAGEASSERTIONS



```
namespace SpecFlowProjectMikhailMoroz.CommonPage
{
    class CommonPageAssertions
    {
        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;

        public CommonPageAssertions(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            _driverManager = new DriverManager(_objectContainer);
            _webDriver = _driverManager.getDriver();
        }

        public void AreEqual(object expected, object actual, string assertionMessage)
        {
            Assert.AreEqual(expected, actual, assertionMessage);
        }

        public void IsTrue(bool condition, string assertionMessage)
        {
            Assert.IsTrue(condition, assertionMessage);
        }

        public void IsFalse(bool condition, string assertionMessage)
        {
            Assert.IsFalse(condition, assertionMessage);
        }

        public void IsEmpty(string actual, string assertionMessage)
        {
            Assert.IsEmpty(actual, assertionMessage);
        }

        public void Contains(object expected, ICollection actual, string message)
        {
            Assert.Contains(expected, actual, message);
        }

        public void NotContains(object expected, ICollection actual, string message)
        {
            CollectionAssert.DoesNotContain(actual, expected, message);
        }

        public void StartsWith(string expected, string actual, string assertionMessage)
        {
            StringAssert.StartsWith(expected, actual, assertionMessage);
        }
        public void PageTitleIsEqualWithDelay(string expectedText, string assertionMessage, int afterSeconds = 5)
        {
            Assert.That(() => _webDriver.Title, Is.EqualTo(expectedText).After(afterSeconds).Seconds.PollEvery(100), assertionMessage);
        }
    }
}
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 19*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗАЦІЯ КЛАСУ DRIVERMANAGER

DriverManager.cs

SpecFlowProjectMikhailMoroz

```
using System.IO;
using NUnit.Framework;
using OpenQA.Selenium.Remote;

[assembly: Parallelizable(ParallelScope.Fixtures)]
[assembly: LevelOfParallelism(1)]
namespace SpecFlowProjectMikhailMoroz.Drivers
{
    [Binding]
    class DriverManager
    {
        private readonly IObjectContainer _objectContainer;
        private IWebDriver _driver;
        private DriverFactory driverFactory;
        private static string _browser;

        public DriverManager(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            driverFactory = new DriverFactory();
        }

        [BeforeTestRun]
        public static void setEnvironment()
        {
            var settings = ConfigurationLoader.Settings;
            _browser = settings.Browser;
        }

        [BeforeScenario]
        public void SelectBrowser()
        {
            Initialize();
        }

        public void Initialize()
        {

            _driver = driverFactory.InitLocalDriver(_browser);
            IAllowsFileDetection allowsDetection = (IAllowsFileDetection)_driver;
            allowsDetection.FileDetector = new LocalFileDetector();
            _objectContainer.RegisterInstanceAs<IWebDriver>(_driver, "driver");
        }

        public IWebDriver getDriver()
        {
            _driver = _objectContainer.Resolve<IWebDriver>("driver");
            return _driver;
        }

        [AfterScenario]
        public void CleanUp()
        {
            _driver.Quit();
        }
    }
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 20*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗАЦІЯ КЛАСУ DRIVERFACTORY

DriverFactory.cs

SpecFlowProjectMikhailMoroz — SpecFlowProjectMikhailMoroz.Drivers.DriverFactory

```
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Edge;
using OpenQA.Selenium.Fi
using System;
using System.Collections.Generic;
using System.IO;
using System.Text;

namespace SpecFlowProjectMikhailMoroz.Drivers
{
    2 references
    class DriverFactory
    {
        private IWebDriver _driver;
        private OptionsFactory options = new OptionsFactory();
        1 reference
        public IWebDriver InitLocalDriver(string browser)
        {
            var webDriverServerPath = Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
            switch (browser)
            {
                case "chrome":
                    _driver = new ChromeDriver(webDriverServerPath, options.getChromeOptions());
                    _driver.Manage().Window.Maximize();
                    return _driver;
                case "firefox":
                    _driver = new FirefoxDriver(webDriverServerPath, options.getFirefoxOptions());
                    _driver.Manage().Window.Maximize();
                    return _driver;
                case "edge":
                    _driver = new EdgeDriver(webDriverServerPath, options.getEdgeOptions());
                    _driver.Manage().Window.Maximize();
                    return _driver;
                default:
                    _driver = new ChromeDriver(webDriverServerPath, options.getChromeOptions());
                    _driver.Manage().Window.Maximize();
                    return _driver;
            }
        }
    }
}
```

{} namespace OpenQA.Selenium.Edge

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 21*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

РЕАЛІЗАЦІЯ КЛАСУ OPTIONSFACTORY

OptionsFactory.cs

SpecFlowProjectMikhailMoroz

```
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Edge;
using OpenQA.Selenium.Firefox;
using System;
using System.Collections.Generic;
using System.Text;

namespace SpecFlowProjectMikhailMoroz.Drivers
{
    2 references
    class OptionsFactory
    {
        2 references
        public ChromeOptions getChromeOptions()
        {
            string[] arguments = {"--start-maximized", "--ignore-certificate-errors",
                "--disable-popup-blocking", "--no-sandbox" , "--disable-dev-shm-usage"};
            ChromeOptions chromeOptions = new ChromeOptions();
            chromeOptions.AcceptInsecureCertificates = true;
            chromeOptions.AddArguments(arguments);

            return chromeOptions;
        }
        1 reference
        public FirefoxOptions getFirefoxOptions()
        {
            FirefoxOptions firefoxOptions = new FirefoxOptions();
            firefoxOptions.AcceptInsecureCertificates = true;

            return firefoxOptions;
        }
        1 reference
        public EdgeOptions getEdgeOptions()
        {
            EdgeOptions edgeOptions = new EdgeOptions();
            edgeOptions.UseChromium = true;
            edgeOptions.AcceptInsecureCertificates = true;

            return edgeOptions;
        }
    }
}
```

Розроб.	Мороз М.Ю.			Дослідження процесів контролю якості взаємодії користувачів з веб-додатком інтернет-магазину	
Перевір.	Мінухін С.В.				
Н. Контр.	Мінухін С.В.				
				СПРм-20-1	*Лист 22*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 22*

Додаток Б

Текст програми

ГЮИК.501610.008 – 01 12 01

(позначення документу)

ЗАТВЕРДЖЕНО
ГЮИК.501610.008 – 01 12 01 – ЛЗ

Дослідження процесів контролю якості взаємодії користувачів з веб-додатком Інтернет-магазину

Текст програми

ГЮИК.501610.008 – 01 12 01 – ЛЗ

ЛИСТІВ 21

2021 р.

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник кваліфікаційної роботи
проф. Мінухін С. В.

Дослідження процесів контролю якості взаємодії користувачів з веб-додатком Інтернет-магазину

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.501610.008 – 01 12 01 – ЛЗ

РОЗРОБИВ:
ст. гр. СПРм-20-1
Мороз М.Ю.

2021 р.

DriverFactory.cs

```
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Edge;
using OpenQA.Selenium.Firefox;
using System.IO;

namespace SpecFlowProjectMikhailMoroz.Drivers
{
   class DriverFactory
   {
      private IWebDriver _driver;
      private OptionsFactory options = new OptionsFactory();
      public IWebDriver InitLocalDriver(string browser)
      {
         var webDriverServerPath =
Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
         switch (browser)
         {
            case "chrome":
               _driver = new ChromeDriver(webDriverServerPath,
options.getChromeOptions());
               _driver.Manage().Window.Maximize();
               return _driver;
            case "firefox":
               _driver = new FirefoxDriver(webDriverServerPath, options.getFirefoxOptions());
               _driver.Manage().Window.Maximize();
               return _driver;
            case "edge":
               _driver = new EdgeDriver(webDriverServerPath, options.getEdgeOptions());
               _driver.Manage().Window.Maximize();
               return _driver;
            default:
               _driver = new ChromeDriver(webDriverServerPath,
options.getChromeOptions());
               _driver.Manage().Window.Maximize();
               return _driver;
         }
      }
   }
}
```

DriverManager.cs

```
using TechTalk.SpecFlow;
using BoDi;
using OpenQA.Selenium;
using NUnit.Framework;
using OpenQA.Selenium.Remote;

[assembly: Parallelizable(ParallelScope.Fixtures)]
[assembly: LevelOfParallelism(1)]
namespace SpecFlowProjectMikhailMoroz.Drivers
{
    [Binding]
    class DriverManager
    {
        private readonly IObjectContainer _objectContainer;
        private IWebDriver _driver;
        private DriverFactory driverFactory;
        private static string _browser;

        public DriverManager(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            driverFactory = new DriverFactory();
        }

        [BeforeTestRun]
        public static void setEnvironment()
        {
            var settings = ConfigurationLoader.Settings;
            _browser = settings.Browser;
        }

        [BeforeScenario]
        public void SelectBrowser()
        {
            Initialize();
        }
        public void Initialize()
        {
            _driver = driverFactory.InitLocalDriver(_browser);
            IAllowsFileDetection allowsDetection = (IAllowsFileDetection)_driver;
            allowsDetection.FileDetector = new LocalFileDetector();
            _objectContainer.RegisterInstanceAs<IWebDriver>(_driver, "driver");
        }
```

```
        public IWebDriver getDriver()
        {
            _driver = _objectContainer.Resolve<IWebDriver>("driver");
            return _driver;
        }

        [AfterScenario]
        public void CleanUp()
        {
            _driver.Quit();
        }

    }
}
```

OptionsFactory.cs

```
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Edge;
using OpenQA.Selenium.Firefox;

namespace SpecFlowProjectMikhailMoroz.Drivers
{
    class OptionsFactory
    {
        public ChromeOptions getChromeOptions()
        {
            string[] arguments = {"--start-maximized", "--ignore-certificate-errors",
                "--disable-popup-blocking", "--no-sandbox" , "--disable-dev-shm-usage"};
            ChromeOptions chromeOptions = new ChromeOptions();
            chromeOptions.AcceptInsecureCertificates = true;
            chromeOptions.AddArguments(arguments);

            return chromeOptions;
        }
        public FirefoxOptions getFirefoxOptions()
        {
            FirefoxOptions firefoxOptions = new FirefoxOptions();
            firefoxOptions.AcceptInsecureCertificates = true;

            return firefoxOptions;
        }
        public EdgeOptions getEdgeOptions()
        {
            EdgeOptions edgeOptions = new EdgeOptions();
            edgeOptions.UseChromium = true;
            edgeOptions.AcceptInsecureCertificates = true;

            return edgeOptions;
```

```
        }
    }
}
```

CommonPageAssertions.cs

```
using BoDi;
using NUnit.Framework;
using OpenQA.Selenium;
using SpecFlowProjectMikhailMoroz.Drivers;
using System.Collections;

namespace SpecFlowProjectMikhailMoroz.CommonPage
{
    class CommonPageAssertions
    {

        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;

        public CommonPageAssertions(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            _driverManager = new DriverManager(_objectContainer);
            _webDriver = _driverManager.getDriver();
        }

        public void AreEqual(object expected, object actual, string assertionMessage)
        {
            Assert.AreEqual(expected, actual, assertionMessage);
        }

        public void IsTrue(bool condition, string assertionMessage)
        {
            Assert.IsTrue(condition, assertionMessage);
        }

        public void IsFalse(bool condition, string assertionMessage)
        {
            Assert.IsFalse(condition, assertionMessage);
        }
        public void IsEmpty(string actual, string assertionMessage)
        {
            Assert.IsEmpty(actual, assertionMessage);
        }
```

```
        public void Contains(object expected, ICollection actual, string message)
        {
            Assert.Contains(expected, actual, message);
        }

        public void NotContains(object expected, ICollection actual, string message)
        {
            CollectionAssert.DoesNotContain(actual, expected, message);
        }

        public void StartsWith(string expected, string actual, string assertionMessage)
        {
            StringAssert.StartsWith(expected, actual, assertionMessage);
        }
        public void PageTitleIsEqualWithDelay(string expectedText, string assertionMessage, int
afterSeconds = 5)
        {
            Assert.That(() => _webDriver.Title,
Is.EqualTo(expectedText).After(afterSeconds).Seconds.PollEvery(100), assertionMessage);
        }
    }
}
```

CommonPageMap.cs

```
using BoDi;
using OpenQA.Selenium;
using SpecFlowProjectMikhailMoroz.Drivers;
using SpecFlowProjectMikhailMoroz.Helpers;
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;

namespace SpecFlowProjectMikhailMoroz.CommonPage
{
    abstract class CommonPageMap
    {
        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;
        private WaitHelper _waitHelper;
        public CommonPageMap(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
            _driverManager = new DriverManager(_objectContainer);
            _waitHelper = new WaitHelper(_objectContainer);
            _webDriver = _driverManager.getDriver();
        }
```

```
        public void ClickTextButton(string locator, string text, bool
waitForJavaScriptLoadAfterClick = false)
    {
        string formattedLocator = string.Format(locator, text);
        IWebElement element = _waitHelper.ClickWait(formattedLocator);
        element.FindElement(By.XPath(formattedLocator)).Click();

        if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
    }

    public void ClickButton(string locator, bool waitForJavaScriptLoadAfterClick = false)
    {
        IWebElement element = _waitHelper.ClickWait(locator);
        element.FindElement(By.XPath(locator)).Click();
        if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
    }

    public void ClickButtonOnView(string buttonLocator, string viewLocator, bool
waitForJavaScriptLoadAfterClick = false)
    {
        IWebElement element = _waitHelper.ClickButtonOnViewWait(buttonLocator,
viewLocator);
        element.FindElement(By.XPath(buttonLocator)).Click();
        if (waitForJavaScriptLoadAfterClick) _waitHelper.WaitForJavaScriptLoad();
    }

    public IWebElement GetElementAvoidStaleElementReferenceException(string locator)
    {
        int attempts = 0;
        while (attempts < 3)
        {
            try
            {
                IWebElement element = _waitHelper.ClickWait(locator);
                return element;
            }
            catch (StaleElementReferenceException)
            {
                attempts++;
            }

        }
        throw new StaleElementReferenceException(locator);
    }
```

```
        public void SendTextToInput(string locator, string text, bool
waitForJavaScriptLoadAfterInput = false)
    {
        IWebElement element = _waitHelper.SendTextWait(locator);
        element.FindElement(By.XPath(locator)).SendKeys(text);
        if (waitForJavaScriptLoadAfterInput) _waitHelper.WaitForJavaScriptLoad();

    }

    public void ClearInputAndSendText(string locator, string text)
    {
        IWebElement element = _waitHelper.SendTextWait(locator);
        element.SendKeys(Keys.Control + "a");
        element.SendKeys(text);
    }

    public string GetText(string locator)
    {
        IWebElement element = _waitHelper.SendTextWait(locator);
        return element.FindElement(By.XPath(locator)).Text;
    }

    public void Navigate(string url, bool isWaitNeeded = true)
    {
        _webDriver.Navigate().GoToUrl(url);
        if (isWaitNeeded) _waitHelper.NavigateWait(url);
    }

    public void Refresh()
    {
        _webDriver.Navigate().Refresh();
    }

    public IWebElement GetElement(string locator)
    {
        return _waitHelper.GetTextWait(locator).FindElement(By.XPath(locator));
    }

    public IWebElement GetInvisibleElement(string locator)
    {
        return _waitHelper.ExistsWait(locator).FindElement(By.XPath(locator));
    }

    public ReadOnlyCollection<IWebElement> FindElementsInFrame(string locatorFrame,
string locatorElement)
    {
        _waitHelper.WaitForJavaScriptLoad();
```

```
        IWebElement frame =
_waitHelper.GetTextWait(locatorFrame).FindElement(By.XPath(locatorFrame));
        return
_webDriver.SwitchTo().Frame(frame).FindElements(By.XPath(locatorElement));
    }
    public IWebElement GetElementByFormatedLocator(string locator, string text)
    {
        string formattedLocator = string.Format(locator, text);
        return
_waitHelper.GetTextWait(formattedLocator).FindElement(By.XPath(formattedLocator));
    }

    public string GetElementValue(string locator)
    {
        return
_waitHelper.GetTextWait(locator).FindElement(By.XPath(locator)).GetAttribute("value");
    }

    public ReadOnlyCollection<IWebElement> GetElements(string locator)
    {
        _waitHelper.GetTextWait(locator);
        return _webDriver.FindElements(By.XPath(locator));
    }

    public IWebElement GetElementWithoutWait(string locator)
    {
        return _webDriver.FindElement(By.XPath(locator));
    }

    public ReadOnlyCollection<IWebElement> GetElementsWithoutWait(string locator)
    {
        return _webDriver.FindElements(By.XPath(locator));
    }

    public string GetPageUrl()
    {
        return _webDriver.Url;
    }

    public List<String> GetListOfValues(string locator)
    {
        return _webDriver.FindElements(By.XPath(locator)).Select(e =>
e.GetAttribute("value")).ToList();
    }

    public List<String> GetListOfTitles(string locator)
    {
```

```
        return _webDriver.FindElements(By.XPath(locator)).Select(e =>
e.GetAttribute("title")).ToList();
    }

    public List<String> GetListOfTexts(string locator)
    {
        return _webDriver.FindElements(By.XPath(locator)).Select(e => e.Text).ToList();
    }
```

DataManager.cs

```
using BoDi;
using System.Collections.Generic;

namespace SpecFlowProjectMikhailMoroz.Helpers
{
    class DataManager
    {
        private IObjectContainer _objectContainer;
        public DataManager(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
        }

        public void SetData(object data, string name)
        {
            _objectContainer.RegisterInstanceAs<object>(data, name);
        }

        public object GetData(string name)
        {
            return _objectContainer.Resolve<object>(name);
        }

        public void SetList(List<string> data, string name)
        {
            _objectContainer.RegisterInstanceAs<List<string>>(data, name);
        }

        public List<string> GetList(string name)
        {
            return _objectContainer.Resolve<List<string>>(name);
        }
        public void SetListOfDict(List<Dictionary<string, string>> data, string name)
        {
            _objectContainer.RegisterInstanceAs<List<Dictionary<string, string>>>(data, name);
        }
```

```
        public List<Dictionary<string, string>> GetListOfDict(string name)
        {
            return _objectContainer.Resolve<List<Dictionary<string, string>>>(name);
        }

        public void SetDict(Dictionary<string, string> data, string name)
        {
            _objectContainer.RegisterInstanceAs<Dictionary<string, string>>(data, name);
        }

        public Dictionary<string, string> GetDict(string name)
        {
            return _objectContainer.Resolve<Dictionary<string, string>>(name);
        }
        public void SetListInt(List<int> data, string name)
        {
            _objectContainer.RegisterInstanceAs<List<int>>(data, name);
        }

        public List<int> GetListInt(string name)
        {
            return _objectContainer.Resolve<List<int>>(name);
        }

    }
}
```

WaitHelper.cs

```
using BoDi;
using OpenQA.Selenium;
using OpenQA.Selenium.Interactions;
using OpenQA.Selenium.Support.UI;
using SpecFlowProjectMikhailMoroz.Drivers;
using System;
using System.Threading;

namespace SpecFlowProjectMikhailMoroz.Helpers
{
    class WaitHelper
    {
        private IObjectContainer _objectContainer;
        private DriverManager _driverManager;
        private IWebDriver _webDriver;
        private WebDriverWait wait;

        public WaitHelper(IObjectContainer objectContainer)
        {
            _objectContainer = objectContainer;
```

```
        _driverManager = new DriverManager(_objectContainer);
        _webDriver = _driverManager.getDriver();
        wait = new WebDriverWait(new SystemClock(),
                        _webDriver,
                        TimeSpan.FromSeconds(10),
                        sleepInterval: TimeSpan.FromMilliseconds(100));
    }

    public IWebElement ClickWait(string locator)
    {
        IWebElement element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator
)));
        element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.XPath(loca
tor)));
        ScrollToView(element);
        element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementToBeClickable(By.XPat
h(locator)));
        return element;
    }

    public IWebElement ClickButtonOnViewWait(string buttonLocator, string viewLocator)
    {
        ScrollIntoView(buttonLocator, viewLocator);
        return ClickWait(buttonLocator);
    }

    public IWebElement SendTextWait(string locator)
    {
        IWebElement element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator
)));
        element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.XPath(loca
tor)));
        ScrollToView(element);
        return element;
    }

    public IWebElement ExistsWait(string locator)
    {
        IWebElement element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator
)));
        return element;
```

```
        }

    public void NavigateWait(string url)
    {
        string shortURL = url.Replace("https:", "").Replace("http:", "");
        wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.UrlContains(shortURL));
        return;
    }

    public IWebElement GetTextWait(string locator)
    {
        IWebElement element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.XPath(locator
)));
        element =
wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.XPath(loca
tor)));
        ScrollToView(element);
        return element;
    }
```

AuthenticationPage.cs

```
using BoDi;

namespace SpecFlowProjectMikhailMoroz.POM
{
   class AuthenticationPage
   {
      public static string emailAddressInputField = "//input[@id='email_create']";
      public static string createAnAccountButton = "//button[@id='SubmitCreate']";
      public static string genderMale = "//input[@id='id_gender1']";
      public static string customerFirstName = "//input[@id='customer_firstname']";
      public static string customerLastName = "//input[@id='customer_lastname']";
      public static string password = "//input[@id='passwd']";
      public static string addressFirstName = "//input[@id='firstname']";
      public static string addressLastName = "//input[@id='lastname']";
      public static string address = "//input[@id='address1']";
      public static string city = "//input[@id='city']";
      public static string zipCode = "//input[@id='postcode']";
      public static string country = "//select[@id='id_country']/option[text()='United States']";
      public static string mobilePhone = "//input[@id='phone_mobile']";
      public static string registerAccount = "//button[@id='submitAccount']";
      public static string daysCombobox = "//select[@id='days']";
      public static string monthCombobox = "//select[@id='months']";
      public static string yearsCombobox = "//select[@id='years']";
      public static string loginEmailField = "//form[@id='login_form']//input[@id='email']";
      public static string loginPasswordField =
"//form[@id='login_form']//input[@id='passwd']";
```

```
        public static string loginSignInButton =
"//form[@id='login_form']//button[@id='SubmitLogin']";
      public static string usernameOnHeader(string username) =>
$"//div[@class='header_user_info']/a/span[text()='{username}']";
      public static string dayOfBirth(string day) =>
$"//select[@id='days']/option[contains(text(),'{day}')]";
      public static string monthOfBirth(string month) =>
$"//select[@id='months']/option[contains(text(),'{month}')]";
      public static string yearOfBirth(string year) =>
$"//select[@id='years']/option[contains(text(),'{year}')]";
      public static string state(string state) =>
$"//select[@id='id_state']/option[contains(text(),'{state}')]";



      private AuthenticationMap _authenticationMap;
      private AuthenticationAssertions _authenticationAssertions;

      public AuthenticationPage(IObjectContainer objectContainere, AuthenticationAssertions
authenticationAssertions)
      {
         _authenticationMap = new AuthenticationMap(objectContainere);
         _authenticationAssertions = authenticationAssertions;
      }

      public AuthenticationPage SendTextToEmailField(string emailAddress)
      {
         _authenticationMap.SendTextToInput(emailAddressInputField, emailAddress);
         return this;
      }
      public AuthenticationPage ClickOnCreateAnAccount()
      {
         _authenticationMap.ClickButton(createAnAccountButton);
         return this;
      }
      public AuthenticationPage SelectGender()
      {
         _authenticationMap.ClickButton(genderMale);
         return this;
      }
      public AuthenticationPage SendTextToPersonalFirstNameField(string firstName)
      {
         _authenticationMap.SendTextToInput(customerFirstName, firstName);
         return this;
      }
      public AuthenticationPage SendTextToPersonalLastNameField(string lastName)
      {
```

```
        _authenticationMap.SendTextToInput(customerLastName, lastName);
        return this;
    }
    public AuthenticationPage SendTextToPasswordField(string userPassword)
    {
        _authenticationMap.SendTextToInput(password, userPassword);
        return this;
    }

    public AuthenticationPage SelectDateOfBirth(string day, string month, string year)
    {
        _authenticationMap.ClickButton(dayOfBirth(day));
        _authenticationMap.ClickButton(monthOfBirth(month));
        _authenticationMap.ClickButton(yearOfBirth(year));
        return this;
    }

    public AuthenticationPage SendTextToAddresslFirstNameField(string firstName)
    {
        _authenticationMap.SendTextToInput(addressFirstName, firstName);
        return this;
    }

    public AuthenticationPage SendTextToAddresslLastNameField(string lastName)
    {
        _authenticationMap.SendTextToInput(addressLastName, lastName);
        return this;
    }

    public AuthenticationPage SendTextToAddressField(string addressName)
    {
        _authenticationMap.SendTextToInput(address, addressName);
        return this;
    }
    public AuthenticationPage SendTextToCityField(string cityName)
    {
        _authenticationMap.SendTextToInput(city, cityName);
        return this;
    }

    public AuthenticationPage SelectCountry()
    {
        _authenticationMap.ClickButton(country);
        return this;
    }
    public AuthenticationPage SelectState(string stateName)
    {
```

```
        _authenticationMap.ClickButton(state(stateName));
        return this;
    }

    public AuthenticationPage SendTextToZipCodeField(string zipCodeName)
    {
        _authenticationMap.SendTextToInput(zipCode, zipCodeName);
        return this;
    }

    public AuthenticationPage SendTextToMobilePhoneField(string mobilePhoneName)
    {
        _authenticationMap.SendTextToInput(mobilePhone, mobilePhoneName);
        return this;
    }

    public AuthenticationPage RegisterAccount()
    {
        _authenticationMap.ClickButton(registerAccount);
        return this;
    }

    public AuthenticationPage VerifyUserIsRegistered(string userName)
    {
        _authenticationAssertions.AreEqual(userName,
_authenticationMap.GetText(usernameOnHeader(userName)), $"{userName} is incorrect");
        return this;
    }

    public AuthenticationPage SendTextToLoginEmailField(string email)
    {
        _authenticationMap.SendTextToInput(loginEmailField, email);
        return this;
    }

    public AuthenticationPage SendTextToLoginPasswordField(string password)
    {
        _authenticationMap.SendTextToInput(loginPasswordField, password);
        return this;
    }
    public AuthenticationPage SignIn()
    {
        _authenticationMap.ClickButton(loginSignInButton);
        return this;
    }
  }
}
```

AuthenticationMap.cs

```
using BoDi;
using OpenQA.Selenium;
using SpecFlowProjectMikhailMoroz.CommonPage;
using SpecFlowProjectMikhailMoroz.Drivers;
using SpecFlowProjectMikhailMoroz.Helpers;

namespace SpecFlowProjectMikhailMoroz.POM
{
   class AuthenticationMap : CommonPageMap
   {
      IObjectContainer _objectContainer;
      private DriverManager _driverManager;
      private IWebDriver _webDriver;
      private WaitHelper waitHelper;
      public AuthenticationMap(IObjectContainer objectContainer) : base(objectContainer)
      {
         _objectContainer = objectContainer;
         _driverManager = new DriverManager(_objectContainer);
         waitHelper = new WaitHelper(_objectContainer);
         _webDriver = _driverManager.getDriver();
      }

      public void ChooseElementInComboBoxList(string ValueInComboBoxListField, string
value)
      {
         ClickTextButton(ValueInComboBoxListField, value);
      }
   }
}
```

AuthenticationAssertions.cs

```
using BoDi;
using OpenQA.Selenium;
using SpecFlowProjectMikhailMoroz.CommonPage;
using SpecFlowProjectMikhailMoroz.Drivers;
using SpecFlowProjectMikhailMoroz.Helpers;
using System;
using System.Collections.Generic;
using System.Text;

namespace SpecFlowProjectMikhailMoroz.POM
{
   class AuthenticationAssertions : CommonPageAssertions
   {
      private IObjectContainer _objectContainer;
      private DriverManager _driverManager;
```

```
        private IWebDriver _webDriver;
      private WaitHelper waitHelper;
      public AuthenticationAssertions(IObjectContainer objectContainer) :
base(objectContainer)
      {
         _objectContainer = objectContainer;
         _driverManager = new DriverManager(_objectContainer);
         waitHelper = new WaitHelper(_objectContainer);
         _webDriver = _driverManager.getDriver();
      }
   }
}
```

AuthenticationSteps.cs

```
using SpecFlowProjectMikhailMoroz.POM;
using TechTalk.SpecFlow;

namespace SpecFlowProjectMikhailMoroz.Steps
{
   [Binding]
   class AuthenticationSteps
   {
      private readonly ScenarioContext _scenarioContext;
      private AuthenticationPage _authenticationPage;

      public AuthenticationSteps(AuthenticationPage authenticationPage)
      {
         _authenticationPage = authenticationPage;
      }

      [When(@"I create a user with the following information")]
      public void WhenICreateAUserWithTheFollowingInformation(Table userDetailsTable)
      {
         var detailsGrid = userDetailsTable.Rows;
         _authenticationPage.SendTextToEmailField(detailsGrid[0]["Email"])
            .ClickOnCreateAnAccount()
            .SelectGender()
            .SendTextToPersonalFirstNameField(detailsGrid[0]["Personal First Name"])
            .SendTextToPersonalLastNameField(detailsGrid[0]["Personal Last Name"])
            .SendTextToPasswordField(detailsGrid[0]["Password"])
            .SelectDateOfBirth(detailsGrid[0]["Day of Birth"], detailsGrid[0]["Month of
Birth"], detailsGrid[0]["Year of Birth"])
            .SendTextToAddresslFirstNameField(detailsGrid[0]["Address First Name"])
            .SendTextToAddresslLastNameField(detailsGrid[0]["Address Last Name"])
            .SendTextToAddressField(detailsGrid[0]["Address"])
            .SendTextToCityField(detailsGrid[0]["City"])
```

```
            .SelectCountry()
            .SelectState(detailsGrid[0]["State"])
            .SendTextToZipCodeField(detailsGrid[0]["Zip code"])
            .SendTextToMobilePhoneField(detailsGrid[0]["Phone"])
            .RegisterAccount();
        }

        [Then(@"I should see '(.*)' user is registered")]
        public void ThenIShouldSeeUserIsRegistered(string userName)
        {
            _authenticationPage.VerifyUserIsRegistered(userName);
        }

        [When(@"I Login as a user with '(.*)' email and '(.*)' password")]
        public void WhenILoginAsAUserWithEmailAndPassword(string email, string password)
        {
            _authenticationPage.SendTextToLoginEmailField(email)
                .SendTextToLoginPasswordField(password)
                .SignIn();
        }

        [Then(@"I should see '(.*)' user is logged in")]
        public void ThenIShouldSeeUserIsLoggedIn(string userName)
        {
            _authenticationPage.VerifyUserIsRegistered(userName);
        }

    }
}
```

appsettings.json

```
{
  "Settings": {
    "Browser": "chrome"
  }
}
```

Feature: Sign Up
As a user
I want to be able to sign in
So that I have an ability to do shopping

Background:
Given I navigate to the main page on the site

@Sign in
Scenario: As a user I want to be able to create an account
When I navigate to the authentication page
And I create a user with the following information
| *Email* | *Personal First Name* | *Personal Last Name* | *Password* | *Day of Birth* | *Month of Birth* | *Year of Birth* | *Address First Name* | *Address Last Name* | *Address* | *City* | *State* | *Zip code* | *Phone* |
| mikhailmoroz2@gmail.com | Mikhail | Moroz | Qqwerty1!! | 18 | May | 1999 | Mikhail | Moroz | St Main | Kharkiv | California | 62002 | 0991231232 |
Then I should see '*Mikhail Moroz*' user is registered

Додаток В

Відомість кваліфікаційної роботи

ГЮИК.501610.008

(позначення документу)

№	Позначення	Назва	Дод. відомості
		Текстові документи	
1.	**ГЮИК.501610.008 ПЗ**	Пояснювальна записка	113 с.
2.	**ГЮИК.501610.008-01 12 01**	Текст програми	19 с.
3.	**ГЮИК.501610.008**	Специфікація	2 с.
		Графічні документи	
4.	**ГЮИК.501610.008 С10**	Життєвий цикл розробки	Включено до ПЗ
		програмного забезпечення	
5.		Модель якості програмного	Включено до ПЗ
		забезпечення	
6.		Процес забезпечення якості	Включено до ПЗ
7.		Структура веб-додатку	Включено до ПЗ
8.		Компоненти Selenium	Включено до ПЗ
9.		Архітектурні компоненти Selenium	Включено до ПЗ
10.		Робочий процес Selenium WebDriver	Включено до ПЗ
11.		Типи локаторів	Включено до ПЗ
12.		Взаємодія SpecFlow та BDD з Веб-додатком	Включено до ПЗ
13.		Тестовий сценарій реєстрації у Веб-додаток	Включено до ПЗ
14.		Згенерований клас визначених кроків з шаблонами методів	Включено до ПЗ
15.		Локатори елементів для реєстрації	Включено до ПЗ
16.		Реалізовані методи класу AuthenticationPage	Включено до ПЗ

					ГЮИК.504310.008 ДЗ			
Зм.	*Лист*	*№ докум*	*Підп.*	*Дата*				
Розроб.		Мороз М.Ю.			*Дослідження процесі контролю якості взаємодії користувачів з веб-додатком інтернет-магазину.*		Лист	Листів
Перевірив.		Мінухін С.В.					*1*	*2*
Н. Контр.		Мінухін С.В.				*ХНУРЕ Кафедра СТ*		
Затверд.		*Гребеннік І.В.*						

17.	**ГЮИК.501610.008 С10**	Реалізовані методи визначення кроків AuthenticationSteps	Включено до ПЗ
18.		Діаграма структури та взаємодії компонентів	Включено до ПЗ
19.		Реалізація класу DataManager	Включено до ПЗ
20.		Реалізація класу WaitHelper	Включено до ПЗ
21.		Реалізація класу CommonPage Map	Включено до ПЗ
22.		Реалізація класу CommonPageAssertions	Включено до ПЗ
23.		Реалізація класу DriverManager	Включено до ПЗ
24.		Реалізація класу DriverFactory	Включено до ПЗ
25.		Реалізація класу OptionsFactory	Включено до ПЗ

Зм.	Лист	№ докум	Підп.	Дата	ГЮИК.501610.008 Д3		
Розроб.		Мороз М.Ю.			*Дослідження процесі контролю якості взаємодії користувачів з веб-додатком інтернет-магазину.*	Лист	Листів
Перевірив.		Мінухін С.В..				2	2
Н. Контр.		Мінухін С.В.				*ХНУРЕ Кафедра СТ*	
Затверд.		*Гребеннік І.В.*					