

Костянтин Кашарайло, Ігор Крошний*, Олег Габа*

студент 6-го курсу, НЛТУ України, Львів, Україна,
kasharaylo.k@nltu.lviv.ua

в.о. директора ННІ КНІТ НЛТУ України, Львів, Україна,
<https://orcid.org/0000-0003-0018-7998>. kroshny.igor@nltu.edu.ua.

асистент кафедри КН НЛТУ України, Львів, Україна,
<https://orcid.org/0009-0006-2140-4524>, oleh.haba@nltu.edu.ua.

Розроблення API модуля вебпорталу навчально-наукового інституту

Анотація - Використовуючи сучасні фреймворки і принципи створення Page-Object патернів для автоматизованих систем розроблено програму для тестування вебпорталу інституту. Були створені основні тестові сценарії готового функціоналу за допомогою мови програмування Ruby і бібліотеки Cucumber.

Ключові слова - Ruby, Cucumber, Chrome Driver, Page-Object, Selenium, Rubymine, автоматизація.

Автоматизоване тестування, на відміну від ручного, спрощує процес виявлення багів за допомогою спеціальних програм, чим скорочує витрати й час на цикл тестування. Іншими словами, це дозволяє отримати готовий програмний продукт без помилок в коротші терміни.

Одним з головних завдань впровадження автоматизації в процес тестування є підвищення ефективності, збільшення охоплення та прискорення тестування за умов постійного повтору тестових сценаріїв. Автотест можна запускати регулярно, в робочий і неробочий час.

Автоматизація потрібна для того, щоб [2, 5]:

- Замінити трудомістке ручне тестування робочих процесів і сценаріїв
- Виключити помилки внаслідок «людського фактору»
- Забезпечити прозорий контроль над процесом тестування
- Скоротити витрати на скрипти завдяки їх повторному використанню
- Скоротити час виходу готового продукту на ринок
- Поліпшити процес розробки (Continues Integration)

Предметом дослідження є інформаційний вебпортал та його функціональність, якою користуються студенти для ознайомлення з програмами та освіт-

німи програмами, розкладом занять, особовим складом інституту та кафедри. Ця інформація є ключовою при виборі вищого навчального закладу абітурієнтом при вступі.

Структуру проекту розроблено з використанням патерну page-object (Рис.1). Елементи які перевикористовуються на різних сторінках містяться в папці “pages”. Це дозволяє використовувати різні сценарії в папці “step_defenitions” багаторазово [1, 7].

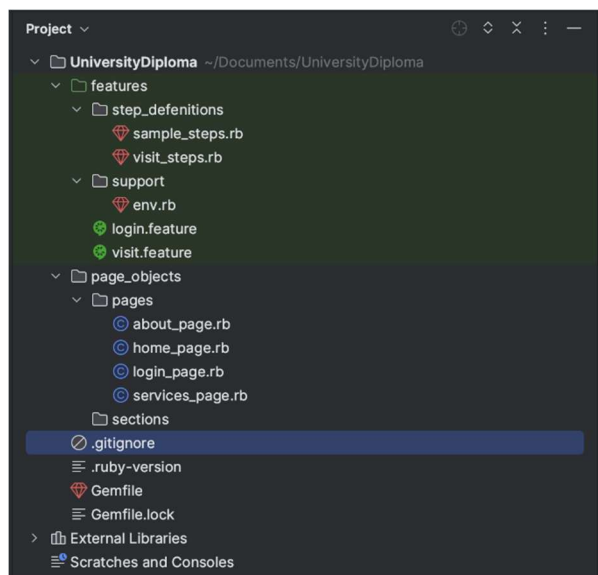


Рис. 1. Структура проекту з використанням патерну page-object

Частина тестових сценаріїв написана з використанням бібліотеки Cucumber (Рис. 2) [2, 4]. На даному прикладі відображено спосіб написання тестів який ввів у себе переваги Gherkin language і простоту мови, яку зрозуміє навіть нетехнічний спеціаліст.

login.feature

Feature: Test site functionality

Scenario: Test login into admin panel

Given I navigate to the admin page

When I enter the email and password

And I click the login button

Then I should be logged in successfully

Рис. 2. Тестові сценарії з використанням бібліотеки Cucumber

Інша частина тестових сценаріїв реалізована з використанням Selenium [3, 6]. Приклад написання і виконання кроків тестового сценарію зображено на рисунку 3.

sample_steps.rb

```
Given("I navigate to the admin page") do
  @home_page.visit_admin_page
end
When("I enter the email and password") do |email, password|
  sleep 2
  @login_page.enter_email(email)
  @login_page.enter_password(password)
  sleep 2
end
And("I click the login button") do
  @login_page.click_login_button
end
Then("I should be logged in successfully") do
  # Add assertions to verify successful login
  # For example, check for the presence of a logged-in user element or a welcome message
  # You can use appropriate Selenium methods to locate and assert the elements
  # Example:
  logged_in_user_element = @driver.find_element(:id, 'logged_in_user')
  expect(logged_in_user_element.displayed?).to be true
end
```

Рис. 3. Написання і виконання кроків тестового сценарію з використанням Selenium

Висновок. В результаті виконання роботи із автоматизації було проведено ознайомлення з предметною областю, пошуку аналогів рішень, які використовують для автоматизації вебдодатків, було обрано технології, за допомогою яких можна вирішити поставлені завдання, та запроектовано фреймворк який здатний покривати основні кейси тестування вебпорталу інституту та кафедр. Як результат, отриманий фреймворк можна перевикористовувати та наповнювати новими сценаріями в майбутній реалізації функціональності. Даний продукт описаний, протестований та запущений в тестовому режимі. Користуватись і розвивати даний фреймворк дуже зручно, оскільки система представляє зручний інструмент у вигляді автоматизованої програми в яку можна додавати сценарії спеціалістом автоматизованного тестування.

Список використаних літературних джерел

- [1] Ruby documentation: <https://ruby-doc.org/>.
- [2] Cucumber official website: <https://cucumber.io/>.

- [3] Page Object model with Ruby and Selenium:
https://www.selenium.dev/documentation/en/guidelines_and_recommendations/page_object_models/.
- [4] Page Object model in Cucumber:
<https://automationpanda.com/2017/01/23/behavior-driven-python-getting-started-with-page-object-models/>.
- [5] Selenium official website: <https://www.selenium.dev/>.
- [6] Selenium WebDriver with Ruby:
<https://www.selenium.dev/selenium/docs/api/rb/>.
- [7] WebDriver architecture and implementation details:
https://www.selenium.dev/documentation/en/webdriver/driver_requirements/.

Олег Левченко, Ірина Борецька*

студент 6-го курсу, НЛТУ України, Львів, Україна. o.levtschen@gmail.com

* к.т.н., доцент кафедри КН, НЛТУ України, Львів, Україна.

<https://orcid.org/0000-0002-6767-104X>, boretska@nltu.edu.ua

Автоматизація процесів розгортання програмних сервісів за допомогою системи керування життєвим циклом Keptn

Анотація - Створено конвеєр, що дозволяє в автоматичному режимі виконати процес розгортання програмного сервісу послідовно у всіх запрограмованих оточеннях. Конвеєр використовує ворота контролю якості для оцінки якості розгортання у кожній фазі та приймає рішення про перехід до наступної, використовуючи цілі рівня обслуговування (SLO). Результати дослідження можуть бути використані для побудови систем управління процесами розгортання програмних сервісів та автоматизації операцій різних рівнів складності.

Ключові слова - конвеєр, розгортання сервісів, автоматизація операцій, ворота контролю якості, цілі рівня обслуговування, оркестрація.

Розвиток Інтернету та технологій сприяв удосконаленню програмних сервісів та способів їх розміщення на серверах. З розвитком віртуалізації ефективність використання фізичних серверів підвищилась завдяки запуску кількох віртуальних машин паралельно на одному сервері. Поява контейнеризації збі-