

Wrocław, 31.05.2022 r.

STUDIA PODYPLOMOWE

**„Tester oprogramowania dla aplikacji mobilnych i
serwerowych”**

Wyższe Szkoły Bankowe
www.wsb.pl

Projekt:

**„Automatyzacja testów dla sklepu
castorama.pl przy wykorzystaniu
Selenium Webdriver”**



Autor:

Justyna Wieteska

Grupa 1

Spis treści

I.	Przypadki testowe dla aplikacji webowej.....	3
1.	Testy strony startowej castorama.pl.....	3
	ID 001	3
2.	Testy podstrony logowanie i rejestracja	3
	ID 101	3
	ID 102	4
	ID 103	4
	ID 104	5
	ID 105	5
3.	Testy dodania produktów do koszyka.....	6
	ID 201	6
	ID 202	6
II.	Skrypty testowe dla aplikacji mobilnej.....	8
III.	Uwagi końcowe	20

I. Przypadki testowe dla aplikacji webowej

1. Testy strony startowej castorama.pl

ID 001	
Tytuł	Załadowanie strony startowej https://www.castorama.pl
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Zainstalowana przeglądarka firefox. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę https://www.castorama.pl	Strona się wyświetla
2. Zaakceptuj pliki cookies.	Pasek cookies znika
3. Sprawdź czy jesteś na poprawnej stronie.	Użytkownik jest na poprawnej stronie
Przypadek pozytywny, test pozytywny.	

2. Testy podstrony logowanie i rejestracja

ID 101	
Tytuł	Przejście na stronę rejestracji.
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę https://www.castorama.pl	Strona się wyświetla
2. Kliknij „Logowanie i rejestracja”.	Użytkownik klika pole „Logowanie i rejestracja”
3. Sprawdź czy jesteś na poprawnej stronie.	Użytkownik jest na poprawnej stronie
Przypadek pozytywny, test pozytywny.	

ID 102	
Tytuł	Test komunikatu przy rejestracji nowego użytkownika używając błędnego adresu e-mail
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę rejestrowania ze strony głównej wybierając „Logowanie i rejestracja”.	Strona się wyświetla
2. Wpisz niepoprawny adres e-mail w sekcji rejestracji	Wpisano niepoprawny e-mail
3. Wpisz hasło zgodnie z wymaganiami	Wpisano poprawne hasło
4. Zaakceptuj regulamin	Regulamin zaakceptowany
5. Kliknij „Założ konto”	Użytkownik klika „Założ konto”
6. Pojawia się komunikat na temat błędnego adresu e-mail.	Komunikat jest zgodny z oczekiwanym.
REZULTAT: Użytkownik nie założył konta. Przypadek negatywny, test pozytywny.	

ID 103	
Tytuł	Test komunikatu przy rejestracji nowego użytkownika używając błędnego hasła
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę www.castorama.pl	Strona się wyświetla
2. Przejdź na podstronę rejestracji	Wyświetlono stronę rejestracji
3. Wpisz poprawny adres e-mail	Wpisano poprawny e-mail
4. Wpisz niepoprawne hasło	Wpisano niepoprawne hasło
5. Zaakceptuj regulamin	Regulamin zaakceptowany
6. Kliknij „Założ konto”	Użytkownik klika „Założ konto”
7. Pojawia się komunikat na temat błędnego hasła.	Komunikat jest zgodny z oczekiwanym.
REZULTAT: Użytkownik nie założył konta. Przypadek negatywny, test pozytywny.	

ID 104	
Tytuł	Test komunikatu przy rejestracji nowego użytkownika nie akceptując regulaminu.
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę www.castorama.pl	Strona się wyświetla
2. Przejdź na podstronę rejestracji	Wyświetlono stronę rejestracji
3. Wpisz poprawny adres e-mail	Wpisano poprawny e-mail
4. Wpisz poprawne hasło	Wpisano poprawne hasło
5. Nie akceptuj regulaminu	Regulamin nie został zaakceptowany
6. Kliknij „Założ konto”	Użytkownik klika „Założ konto”
7. Pojawia się komunikat na temat nieakceptacji regulaminu	Komunikat jest zgodny z oczekiwanym.
REZULTAT: Użytkownik nie założył konta. Przypadek negatywny, test pozytywny.	

ID 105	
Tytuł	Test komunikatu przy logowaniu nowego użytkownika wpisując błędne hasło.
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę www.castorama.pl	Strona się wyświetla
2. Przejdź na podstronę logowania	Wyświetlono stronę logowania
3. Wpisz poprawny adres e-mail	Wpisano poprawny e-mail
4. Wpisz niepoprawne hasło	Wpisano niepoprawne hasło.
5. Zaakceptuj regulamin	Regulamin został zaakceptowany
6. Kliknij „Zaloguj mnie”	Użytkownik klika „Zaloguj mnie”
7. Pojawia się komunikat na temat błędnego hasła	Komunikat jest zgodny z oczekiwanym.
REZULTAT: Użytkownik nie zostaje zalogowany. Przypadek negatywny, test pozytywny.	

3. Testy dodania produktów do koszyka

ID 201	
Tytuł	Dodanie do koszyka produktu
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę https://www.castorama.pl	Strona się wyświetla
2. Wyszukaj produkt.	Wyszukano produkt
3. Kliknij w produkt.	Kliknięto w produkt
4. Kliknij „Do koszyka”.	Kliknięto „Do koszyka”
5. Kliknij „Przejdź do koszyka”.	Wybrano „Przejdź do koszyka”
6. Sprawdź czy w koszyku jest produkt.	Sprawdzono czy produkt jest w koszyku
REZULTAT: Produkt został dodany do koszyka. Przypadek pozytywny, test pozytywny.	

ID 202	
Tytuł	Test komunikatu po wprowadzeniu błędnych danych (brak wpisanego imienia).
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę https://www.castorama.pl	Strona się wyświetla
2. Wyszukaj produkt	Wyszukano produkt
3. Wejdź w produkt	Kliknięto w produkt
4. Kliknij „Do koszyka”	Kliknięto „Do koszyka”
5. Kliknij „Przejdź do koszyka”	Wybrano „Przejdź do koszyka”
6. Kliknij „Wybierz sklep”	Wybrano „Wybierz sklep”
7. Wybierz „Wrocław Magnolia”	Wybrano „Wrocław Magnolia”
8. Kliknij „Przejdź do zamówienia”	Wybrano „Przejdź do zamówienia”
9. Kliknij „Kupuję jako gość”	Wybrano „Kupuję jako gość”
10. Uzupełnij pola: e-mail, nazwisko, kraj, kod pocztowy, ulicę, nr domu i telefon. Nie uzupełniaj imienia.	Uzupełniono pola: e-mail, nazwisko, kraj, kod pocztowy, ulicę, nr domu i telefon.

11. Wybierz opcję dostawy: odbiór osobisty w sklepie	Wybrano odbiór osobisty w sklepie
12. Kliknij „Wybierz sposób płatności”	Kliknięto „ Wybierz sposób płatności”
REZULTAT: Brak złożonego zamówienia. Pojawia się komunikat „Błąd danych do wysyłki” Przypadek negatywny, test pozytywny.	

ID 203	
Tytuł	Test komunikatu po wprowadzeniu błędnych danych (brak akceptacji regulaminu).
Środowisko	Firefox wersja 100.0.2, Ubuntu wersja 20.04.3 LTS
Warunki wstępne	Uruchomiona przeglądarka. Użytkownik nie jest zalogowany.

Kroki	Oczekiwany rezultat
1. Wejdź na stronę https://www.castorama.pl	Strona się wyświetla
2. Wyszukaj produkt	Wyszukano produkt
3. Wejdź w opis produktu	Kliknięto w produkt
4. Kliknij „Do koszyka”	Kliknięto „Do koszyka”
5. Kliknij „Przejdź do koszyka”	Wybrano „Przejdź do koszyka”
6. Kliknij „Wybierz sklep”	Wybrano „Wybierz sklep”
7. Wybierz „Wrocław Magnolia”	Wybrano „Wrocław Magnolia”
8. Kliknij „Przejdź do zamówienia”	Wybrano „Przejdź do zamówienia”
9. Kliknij „Kupuję jako gość”	Wybrano „Kupuję jako gość”
10. Uzupełnij pola: e-mail, imię, nazwisko, kraj, kod pocztowy, ulicę, nr domu i telefon.	Uzupełniono pola: e-mail, imię, nazwisko, kraj, kod pocztowy, ulicę, nr domu i telefon.
11. Wybierz opcję dostawy: kurier	Wybrano odbiór osobisty w sklepie
12. Kliknij „Wybierz sposób płatności”	Kliknięto „ Wybierz sposób płatności”
13. Wybierz sposób płatności BLIK	Wybrano sposób płatności BLIK
14. Kliknij „Kupuję z obowiązkiem zapłaty”	Kliknięto „Kupuję z obowiązkiem zapłaty”
REZULTAT: Brak złożonego zamówienia. Pojawia się komunikat „Nie zaznaczono wymaganej zgody”. Przypadek negatywny, test pozytywny.	

II. Skrypty testowe dla aplikacji mobilnej

ID 001

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By

#Dane testowe
strona_glowna = "Zakupy w Castorama"

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testow")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")
        self.driver.implicitly_wait(10)

    # Zamkniecie przegladarki
    def tearDown(self):
        print("Zamkniecie przegladarki")
        self.driver.quit()

    #Uruchomienie testow
    def testNoNameInput(self):
        print("Uruchomienie testu nr 001")
        driver = self.driver

    #Znajdz cookies na stronie startowej i je zaakceptuj
        cookies = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
        cookies.click()

    #Sprawdz czy jestes na poprawnej stronie
        test_strony = driver.find_element(By.CSS_SELECTOR, ".shopping-pictograms__title").text
        assert test_strony == strona_glowna

if __name__ == '__main__':
    unittest.main()
```

ID 101

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By

#Dane testowe
podstrona_rejestracji = "Rejestracja" \
    ""

class CastoramaTest(unittest.TestCase):
```



```

def setUp(self):
    print("Przygotowanie do testow")
    self.driver = webdriver.Firefox()
    self.driver.maximize_window()
    self.driver.get("http://www.castorama.pl")
    self.driver.implicitly_wait(10)
#Zamkniecie przegladarki
def tearDown(self):
    print("Zamkniecie przegladarki")
    self.driver.quit()
#Uruchomienie testow
def testNoNameInput(self):
    print("Uruchomienie testu nr 101")
    driver = self.driver
#Przejdź do strony logowania
    przejdź_do_logowania = driver.find_element(By.CSS_SELECTOR,"a.login-header__upper-link")
    przejdź_do_logowania.click ()
#Sprawdzenie poprawnosci strony
    test_podstrony = driver.find_element(By.CSS_SELECTOR,"div.section-base:nth-child(2) > h4:nth-
child(1)").text
    assert test_podstrony == podstrona_rejestracji

if __name__ == '__main__':
    unittest.main()

```

ID 102

```

import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep

```

```

#Dane testowe
email = "jonybrawo@xd"
haslo = "Lubieplacki123!"
komunikat = "Niepoprawny adres email"

```

```

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testow")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")
        self.driver.implicitly_wait(10)

```

```

#Zamkniecie przegladarki
def tearDown(self):
    print("Zamkniecie przegladarki")
    self.driver.quit()

```

```

#Uruchomienie testow
def testrejestracji(self):
    print("Uruchomienie testu nr 102")
    driver = self.driver
#Przejdź do strony rejestracji
    przejdź_do_rejestracji = driver.find_element(By.CSS_SELECTOR,"a.login-header__upper-link")
    przejdź_do_rejestracji.click ()
#Wpisz niepoprawny email
    wpisz_email = driver.find_element(By.ID,"signInSignUp")
    wpisz_email.send_keys (email)
    sleep (3)
#Wpisz poprawne hasło
    wpisz_hasło = driver.find_element(By.ID,"signUpPassword")
    wpisz_hasło.send_keys (hasło)
#Zaakceptowanie regulaminu
    akceptuj_regulamin = driver.find_element(By.ID,"regulationsAgreement")
    akceptuj_regulamin.click ()
    sleep (3)
#Cookie- by się zarejestrować należy zaakceptować cookie
    cookie = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
    cookie.click()
#Założenie konta
    załóż_konto = driver.find_element(By.CSS_SELECTOR,"button.button-base:nth-child(6)")
    załóż_konto.click ()
#Test komunikatu
    test_komunikat = driver.find_element(By.CSS_SELECTOR,"p.base-text").text
    assert test_komunikat == komunikat

if __name__ == '__main__':
    unittest.main()

```

ID 103

```

import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep

```

```

#Dane testowe
email = "justynatester2022@gmail.com"
hasło = "casto"
komunikat = "Hasło nie spełnia wymagań"

```

```

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testów")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")
        self.driver.implicitly_wait(10)

```

```

#Zamkniecie przegladarki
def tearDown(self):
    print("Zamkniecie przegladarki")
    self.driver.quit()

#Uruchomienie testow
def testrejestracji(self):
    print("Uruchomienie testu nr 103")
    driver = self.driver
#Przejdź do strony rejestracji
    przejdź_do_rejestracji = driver.find_element(By.CSS_SELECTOR,"a.login-header__upper-link")
    przejdź_do_rejestracji.click ()
#Wpisz poprawny email
    wpisz_email = driver.find_element(By.ID,"signInSignUp")
    wpisz_email.send_keys (email)
#Wpisz niepoprawne hasło
    wpisz_hasło = driver.find_element(By.ID,"signUpPassword")
    wpisz_hasło.send_keys (hasło)
    sleep (3)
#Zaakceptowanie regulaminu
    akceptuj_regulamin = driver.find_element(By.ID,"regulationsAgreement")
    akceptuj_regulamin.click ()
    sleep (3)
#Cookie- by się zarejestrować należy zaakceptować cookie
    cookie = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
    cookie.click()
#Założenie konta
    zaloż_konto = driver.find_element(By.CSS_SELECTOR,"button.button-base:nth-child(6)")
    zaloż_konto.click()
#Test komunikatu
    test_komunikat = driver.find_element(By.CSS_SELECTOR,"p.base-text").text
    assert test_komunikat == komunikat

if __name__ == '__main__':
    unittest.main()

```

ID 104

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
```

```
#Dane testowe
```

```
email = "justynatester2022@gmail.com"
```

```
haslo = "Castor123!"
```

```
komunikat = "Nie zaznaczono wymaganej zgody"
```

```
class CastoramaTest(unittest.TestCase):
```

```
    def setUp(self):
```

```
        print("Przygotowanie do testow")
```

```
        self.driver = webdriver.Firefox()
```

```
        self.driver.maximize_window()
```

```
        self.driver.get("http://www.castorama.pl")
```

```
        self.driver.implicitly_wait(10)
```

```
#Zamkniecie przegladarki
```

```
    def tearDown(self):
```

```
        print("Zamkniecie przegladarki")
```

```
        self.driver.quit()
```

```
#Uruchomienie testow
```

```
    def testrejestracji(self):
```

```
        print("Uruchomienie testu nr 104")
```

```
        driver = self.driver
```

```
#Przejdź do strony rejestracji
```

```
        przejdź_do_rejestracji = driver.find_element(By.CSS_SELECTOR,"a.login-header__upper-link")
```

```
        przejdź_do_rejestracji.click()
```

```
#Wpisz poprawny email
```

```
        wpisz_email = driver.find_element(By.ID,"signInSignUp")
```

```
        wpisz_email.send_keys (email)
```

```
        sleep(2)
```

```
#Wpisz poprawne haslo
```

```
        wpisz_haslo = driver.find_element(By.ID,"signUpPassword")
```

```
        wpisz_haslo.send_keys (haslo)
```

```
        sleep(2)
```

```
#Cookie- by się zarejestrować należy zaakceptować cookie
```

```
        cookie = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
```

```
        cookie.click()
```

```
#Założenie konta
```

```
        załoz_konto = driver.find_element(By.CSS_SELECTOR,"button.button-base:nth-child(6)")
```

```
        załoz_konto.click()
```

```
#Komunikat
```

```
        logowanie = driver.find_element(By.CSS_SELECTOR,"p.base-text").text
```

assert logowanie == komunikat

```
if __name__ == '__main__':  
    unittest.main()
```

ID 105

```
import unittest  
from selenium import webdriver  
from selenium.webdriver.common.by import By  
from time import sleep
```

#Dane testowe

email = "justynatester2022@gmail.com"

haslo = "haslo123"

komunikat = "Hasło powinno składać się co najmniej z 8 znaków: dużej litery, małej litery, cyfry lub znaku specjalnego (!'^£\$%&*(){}@#~?><>,|=_+~-])."

```
class CastoramaTest(unittest.TestCase):  
    def setUp(self):  
        print("Przygotowanie do testow")  
        self.driver = webdriver.Firefox()  
        self.driver.maximize_window()  
        self.driver.get("http://www.castorama.pl")  
        self.driver.implicitly_wait(10)
```

#Zamkniecie przegladarki

```
    def tearDown(self):  
        print("Zamkniecie przegladarki")  
        self.driver.quit()
```

#Uruchomienie testow

```
    def testlogowania(self):  
        print("Uruchomienie testu nr 105")  
        driver = self.driver
```

#Przejdź do strony rejestracji

```
        przejdź_do_rejestracji = driver.find_element(By.CSS_SELECTOR,"a.login-header__upper-link")  
        przejdź_do_rejestracji.click ()
```

#Wpisz poprawny email

```
        wpisz_email = driver.find_element(By.ID,"signInEmail")  
        wpisz_email.send_keys (email)  
        sleep (2)
```

#wpisz niepoprawne haslo

```
        wpisz_haslo = driver.find_element(By.ID,"signInPassword")  
        wpisz_haslo.send_keys (haslo)  
        sleep (2)
```

#Zalogowanie

```

        zaloz_konto = driver.find_element(By.CSS_SELECTOR,"button.button-base:nth-child(5)")
        zaloz_konto.click ()
#Komunikat
        logowanie = driver.find_element(By.CSS_SELECTOR,".text-warn").text
        assert logowanie == komunikat

if __name__ == '__main__':
    unittest.main()

```

ID 201

```

import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By

```

#Dane testowe

```

produkt = "Wiertarko-wkrętarka udarowa Makita DHP485 18 V"

```

```

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testow")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")
        self.driver.implicitly_wait(10)

```

#Zamknięcie przeglądarki

```

    def tearDown(self):
        print("Zamknięcie przeglądarki")
        self.driver.quit()

```

#Uruchomienie testów

```

    def testkoszyka(self):
        print("Uruchomienie testu nr 201")
        driver = self.driver

```

#Wyszukanie produktu

```

        wyszukaj_produkt = driver.find_element(By.ID,"search-input-desktop")
        wyszukaj_produkt.send_keys (produkt)
        kliknij_lupke = driver.find_element(By.CSS_SELECTOR,"#search_mini_formdesktop > button:nth-
child(3)")
        kliknij_lupke.click ()
        znajdz_produkt = driver.find_element(By.CSS_SELECTOR,"#product-i-1150747 > div:nth-child(1)
> div:nth-child(1) > a:nth-child(1) > img:nth-child(1)")
        znajdz_produkt.click ()

```

#Dodaj do koszyka

```

        dodaj_do_koszyka = driver.find_element(By.CSS_SELECTOR,"button.add-to-cart-button:nth-
child(2)")

```

```

        dodaj_do_koszyka.click ()
#Przejdź do koszyka
    przejdź_do_koszyka = driver.find_element(By.CSS_SELECTOR, ".button")
    przejdź_do_koszyka.click ()
#Test produktu w koszyku
    produkt_w_koszyku = driver.find_element(By.CSS_SELECTOR, ".product-info__product-name-link").text
    assert produkt_w_koszyku == produkt

if __name__ == '__main__':
    unittest.main()

```

ID 202

```

import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep

#Dane testowe
produkt = "Wiertarko-wkrętarka udarowa Makita DHP485 18 V"
mail = "justynatester2022@gmail.com"
last_name = "Bravo"
first_name = "John"
country = "Polska"
zipcode = "54-104"
street = "Maslicka"
number = "12345"
phone = "123456789"
expected_message = "Błąd danych do wysyłki"

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testów")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")
        self.driver.implicitly_wait(10)
#Zamknięcie przeglądarki
    def tearDown(self):
        print("Zamknięcie przeglądarki")
        self.driver.quit()

#Uruchomienie testów
    def testzamowienia(self):
        print("Uruchomienie testu nr 202")
        driver = self.driver

```

#Wyszukanie produktu

```
wyszukaj_produkt = driver.find_element(By.ID, "search-input-desktop")
wyszukaj_produkt.send_keys (produkt)
kliknij_lupke = driver.find_element(By.CSS_SELECTOR, "#search_mini_formdesktop > button:nth-child(3)")
kliknij_lupke.click ()
znajdz_produkt = driver.find_element(By.CSS_SELECTOR, "#product-i-1150747 > div:nth-child(1) > div:nth-child(1) > a:nth-child(1) > img:nth-child(1)")
znajdz_produkt.click ()
```

#Dodaj do koszyka

```
dodaj_do_koszyka = driver.find_element(By.CSS_SELECTOR, "button.add-to-cart-button:nth-child(2)")
dodaj_do_koszyka.click ()
```

#Przejdź do koszyka

```
przejdź_do_koszyka = driver.find_element(By.CSS_SELECTOR, ".button")
przejdź_do_koszyka.click ()
```

#Wybierz sklep

```
wybor_sklepu = driver.find_element(By.CSS_SELECTOR, ".page-checkout-cart__choose-store-monit > button:nth-child(2)")
wybor_sklepu.click ()
wybor_magnolia = driver.find_element(By.CSS_SELECTOR, "div.popup-choose-availability-market-item:nth-child(85) > button:nth-child(2)")
wybor_magnolia.click ()
sleep(3)
```

#Przejdź do zamowienia

```
zamowienie = driver.find_element(By.CSS_SELECTOR, ".checkout-cart-summary-old__go-to-order-button")
zamowienie.click ()
sleep(3)
```

#Odklikaj pojawiające się okienko - reklama

```
castomat = driver.find_element(By.ID, "dws-shop-popup-mask")
castomat.click()
```

#Kupuj jako gość

```
gosc = driver.find_element(By.CSS_SELECTOR, ".login-page__link")
gosc.click()
```

#Uzupełnij dane zamawiającego

```
sleep (4)
wprowadz_mail = driver.find_element(By.ID, "email")
wprowadz_mail.send_keys(mail[0:1])
wprowadz_mail.send_keys(mail[1:len(mail)])
nazwisko = driver.find_element(By.ID, "lastname")
nazwisko.send_keys(last_name[0:1])
nazwisko.send_keys(last_name[1:len(last_name)])
kod_pocztowy = driver.find_element(By.ID, "zipCode")
kod_pocztowy.send_keys(zipcode)
```

#Cookie- by uzupełnić ulicę należy zaakceptować cookie

```
cookie = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
cookie.click()
ulica = driver.find_element(By.ID, "vs4__combobox")
ulica.click()
wybierz_ulice = driver.find_element(By.ID, "vs4__option-1")
wybierz_ulice.click()
```



```

numer_domu = driver.find_element(By.ID, "buildingNumber")
numer_domu.send_keys(number[0:1])
numer_domu.send_keys(number[1:len(number)])
telefon = driver.find_element(By.ID, "areaCodeWithPhone")
telefon.send_keys(phone)
#Wybierz opcje dostawy - odbior osobisty
odbior = driver.find_element(By.CSS_SELECTOR,"div.multistep-checkout-shipping-methods-
form__shipping-option-wrapper:nth-child(1) > div:nth-child(1) > div:nth-child(1) > div:nth-child(1) >
label:nth-child(2)")
odbior.click()
#Wybierz sposob platnosci
platnosc = driver.find_element(By.CSS_SELECTOR,"button.button-base--blue:nth-child(1)")
platnosc.click()

#Test komunikatu
sleep(2)
komunikat = driver.find_element(By.CSS_SELECTOR,"p.base-text:nth-child(1)").text
assert komunikat == expected_message

if __name__ == '__main__':
    unittest.main()

```

ID 203

```

import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep

# Dane testowe
produkt = "Wiertarko-wkrętarka udarowa Makita DHP485 18 V"
mail = "justynatester2022@gmail.com"
last_name = "Bravo"
first_name = "John"
country = "Polska"
zipcode = "54-104"
street = "Maslicka"
number = "12345"
phone = "123456789"
expected_message = "Nie zaznaczono wymaganej zgody"

```

```

class CastoramaTest(unittest.TestCase):
    def setUp(self):
        print("Przygotowanie do testow")
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("http://www.castorama.pl")

```

```

        self.driver.implicitly_wait(10)
#Zamknięcie przeglądarki
def tearDown (self):
    print("Zamkniecie przeglądarki")
    self.driver.quit()
#Uruchomienie testów
def testzamownienia(self):
    print("Uruchomienie testu nr 203")
    driver = self.driver
#Wyszukanie produktu
    wyszukaj_produkt = driver.find_element(By.ID, "search-input-desktop")
    wyszukaj_produkt.send_keys(produkt)
    kliknij_lupke = driver.find_element(By.CSS_SELECTOR, "#search_mini_formdesktop > button:nth-child(3)")
    kliknij_lupke.click()
    znajdz_produkt = driver.find_element(By.CSS_SELECTOR,
        "#product-i-1150747 > div:nth-child(1) > div:nth-child(1) > a:nth-child(1) > img:nth-child(1)")
    znajdz_produkt.click()
#Dodaj do koszyka
    dodaj_do_koszyka = driver.find_element(By.CSS_SELECTOR, "button.add-to-cart-button:nth-child(2)")
    dodaj_do_koszyka.click()
#Przejdź do koszyka
    przejdź_do_koszyka = driver.find_element(By.CSS_SELECTOR, ".button")
    przejdź_do_koszyka.click()
#Wybierz sklep
    wybor_sklepu = driver.find_element(By.CSS_SELECTOR,
        ".page-checkout-cart__choose-store-monit > button:nth-child(2)")
    wybor_sklepu.click()
    wybor_magnolia = driver.find_element(By.CSS_SELECTOR,
        "div.popup-choose-availability-market-item:nth-child(85) > button:nth-child(2)")
    wybor_magnolia.click()
    sleep(3)
#Przejdź do zamownienia
    zamownienie = driver.find_element(By.CSS_SELECTOR, ".checkout-cart-summary-old__go-to-order-button")
    zamownienie.click()
    sleep(3)
#Odklikaj pojawiające się okienko - reklama
    castomat = driver.find_element(By.ID, "dws-shop-popup-mask")
    castomat.click()
#Kup jako gość
    gosc = driver.find_element(By.CSS_SELECTOR, ".login-page__link")
    gosc.click()
#Uzupenij dane zamawiającego
    sleep(4)
    wprowadz_mail = driver.find_element(By.ID, "email")
    wprowadz_mail.send_keys(mail[0:1])
    wprowadz_mail.send_keys(mail[1:len(mail)])
    imie = driver.find_element(By.ID, "firstname")
    imie.send_keys(first_name[0:1])
    imie.send_keys(first_name[1:len(first_name)])

```

```

nazwisko = driver.find_element(By.ID, "lastname")
nazwisko.send_keys(last_name[0:1])
nazwisko.send_keys(last_name[1:len(last_name)])
kod_pocztowy = driver.find_element(By.ID, "zipCode")
kod_pocztowy.send_keys(zipcode)
#Cookie- by uzupełnic ulice należy zaakceptować cookie
cookie = driver.find_element(By.CSS_SELECTOR, ".cookie-bar-close-button")
cookie.click()
ulica = driver.find_element(By.ID, "vs4__combobox")
ulica.click()
wybierz_ulice = driver.find_element(By.ID, "vs4__option-1")
wybierz_ulice.click()
numer_domu = driver.find_element(By.ID, "buildingNumber")
numer_domu.send_keys(number[0:1])
numer_domu.send_keys(number[1:len(number)])
telefon = driver.find_element(By.ID, "areaCodeWithPhone")
telefon.send_keys(phone)
#Wybierz opcje dostawy - kurier
odbior = driver.find_element(By.CSS_SELECTOR,
    "div.col-12:nth-child(3) > div:nth-child(1) > div:nth-child(1) > div:nth-child(1) > label:nth-
child(2)")
odbior.click()
# wybierz sposob platnosci
platnosc = driver.find_element(By.CSS_SELECTOR, "button.button-base--blue:nth-child(1)")
platnosc.click()
#Wybierz platnosc blik
blik = driver.find_element(By.ID, "paymentMethodCode[bold_p24_blik]")
blik.click()
#Podsumuj zakupy
zakupy = driver.find_element(By.CSS_SELECTOR, ".button-base--full-width")
zakupy.click()
#Kup z obowiązkiem zapłaty
sleep(2)
kup = driver.find_element(By.CSS_SELECTOR, ".button-base--blue")
kup.click()

#Test komunikatu
sleep(2)
komunikat = driver.find_element(By.CSS_SELECTOR, ".base-text--line-height-normal").text
assert komunikat == expected_message

if __name__ == '__main__':
    unittest.main()

```

III. Uwagi końcowe

Automatyzacja przypadku testowego powiodła się.

Testy mogą być wrażliwe na zmianę struktury strony ze względu na brak bieżącej informacji dla autora testów o zmianach zachodzących na stronie.

Otwieranie okna przeglądarki dla każdego testu indywidualnie istotnie wpływa na czas trwania testu i jego wynik. Słabsze parametry łącza internetowego, wzmożony ruch na stronie oraz uruchomienie mechanizmów zabezpieczających stronę może spowodować, że limit czasowy zostanie przekroczony i test zakończy się błędem.