**WYŻSZA SZKOŁA INFORMATYKI I ZARZĄDZANIA**

**„COPERNICUS” WE WROCŁAWIU**

**WYDZIAŁ INFORMATYKI, ADMINISTRACJI I FIZJOTERAPII**

Kierunek: **Informatyka (INF)**

Specjalność: **Inżynieria Systemów Informatycznych (ISI)**

**PRACA DYPLOMOWA INŻYNIERSKA**

Marta Korzeniowska

**Projekt i wykonanie testów kluczowych funkcjonalności wybranej aplikacji webowej za pomocą Selenium WebDriver**

**Project and execution of automated tests for key functionalities in a selected web application using Selenium WebDriver**

**Ocena pracy**:………….…………....……………………………………

(ocena pracy dyplomowej, data, podpis promotora)

…..……………………………………………

(pieczątka uczelni)

**Promotor:**

**mg inż. Marek Michalski**

**WROCŁAW 2022**

**Spis treści**

1. **Wstęp** 3
   1. Motywacja 3
   2. Cel i zakres pracy 3
   3. Struktura pracy 4
2. **Specyfikacja wymagań** 5
   1. Opis rzeczywistości 5
   2. Słownik pojęć 6
   3. Wymagania funkcjonalne 7
   4. W 11
   5. Scenariusze testowe 11
3. **Zastosowane technologie** 14
   1. Platforma i środowisko 14
   2. Selenium WebDriver 15
   3. Python 18
   4. Pytest 19
   5. Narzędzia programistyczne 21
   6. Uzasadnienie wyboru technologii 21
4. **Projekt testów** 22
   1. Struktura projektu 22
   2. Page Object Model 24
   3. Napotkane problemy 26
   4. Uruchamianie 27
   5. Raportowanie 28
   6. Wybrane rozwiązania programistyczne 39
5. **Podsumowanie** 50
   1. Efekt prac 50
   2. Rozwój testów 50
6. **Bibliografia** 51
7. **Wstęp**
   1. **Motywacja**

Obecnie e-commerce jest jednym z najbardziej dynamicznie rozwijających się rynków. W dobie pandemii i globalnego lock downu konsumenci wybierali internetowe wersje sklepów w miejsce ich stacjonarnych odpowiedników. W 2020 roku światowy rynek e-commerce doświadczył ogromnej ekspansji. Według szacunków IBM pandemia przyspieszyła o ok.5 lat przechodzenie klientów ze strefy stacjonarnej do strefy online. Dla przykładu Polacy dokonali w ten sposób zakupów na ponad 15,3 mld EUR co oznacza wzrost o 31,4% w stosunku do roku poprzedniego. Dzięki temu polski rynek e-commerce jest jednym z najbardziej dynamicznie rozwijających się w Europie. Oznacza to rosnące zapotrzebowanie na aplikacje webowe dla sklepów internetowych, a co za tym idzie również na projekty testów, które sprawdzą poprawność działania takich aplikacji. W swojej pracy skupiam się na testowaniu ponieważ uważam je za bardzo ciekawe i niezwykle potrzebne zagadnienie. Niewłaściwie wyświetlona strona sklepu internetowego, niedziałający przycisk, problem z założeniem konta czy dodaniem produktu do koszyka, wszystko to może sprawić, że klient zniechęci się do zakupów, a to tylko niektóre z potencjalnych problemów jakie mogą wystąpić. Uważam zatem, że w procesie wytwarzania oprogramowania testowanie gotowej aplikacji jest niezwykle potrzebnym aspektem.

* 1. **Cel i zakres pracy**

Celem pracy jest stworzenie planu, opisu i wreszcie projektu testów automatycznych aplikacji webowej My Store ([www.automationpractice.com](http://www.automationpractice.com/)).

* 1. **Struktura pracy**

1. **Specyfikacja wymagań**
   1. **Opis rzeczywistości.**

My Store jest projektem sklepu internetowego stworzonym przez Selenium Framework umożliwiającym naukę i ćwiczenie testów automatycznych w realnym środowisku webowym. Posiada wszystkie kluczowe funkcjonalności sklepu w tym m.in: podział produktów na kategorie, wyświetlanie karty produktu, zakładanie konta użytkownika, logowanie, dodawanie produktów do koszyka, aż po finalizację zakupów z wyłączeniem prawdziwych płatności. Stanowi zatem bezpieczne środowisko do nauki automatyzacji testów oraz wdrażania przykładowych scenariuszy testowych na wybranych przeglądarkach internetowych, odzwierciedlając przy tym realne zagadnienia i problemy. Z uwagi na ogólną dostępność strony twórcy My Store zastrzegli jedynie, by nie spamować strony harmonogramem testów uruchamianych w sposób ciągły. Dlatego w swojej pracy nie korzystam z podobnego rozwiązania uruchamiania testów.

* 1. **Słownik pojęć**

Wybrane terminy testowe:

**Historyjka użytkownika (user story):** Wymóg użytkownika lub biznesowy składający się z jednego zdania wyrażonego w języku codziennym lub biznesowym, określający funkcjonalność której potrzebuje użytkownik, uzasadnienie, wszelkie kryteria niefunkcjonalne, a także kryteria akceptacji.

**Plan testów (test plan):** Dokumentacja opisująca cele testowe do osiągnięcia oraz środki i harmonogram ich realizacji, zorganizowane tak, by koordynować czynności testowe.

**Przypadek testowy (test case):** Zestaw warunków wstępnych, danych wejściowych, akcji (w stosownych przypadkach), oczekiwanych rezultatów i warunków końcowych opracowany w oparciu o warunki testowe.

**Test (test):** Zestaw jednego lub więcej przypadków testowych.

**Test dymny (smoke test):** Zestaw testów obejmujący główną funkcjonalność modułu lub systemu sprawdzający, czy działa ona poprawnie przed rozpoczęciem planowanych testów.

**Testowanie funkcjonalne:** Testowanie wykonywane by ocenić czy moduł lub system spełnia wymagania funkcjonalne.

**Testowanie GUI (od ang. Graphical User Interface):** Testowanie wykonywane przez współdziałanie z testowanym oprogramowaniem poprzez graficzny interfejs użytkownika.

**Testowanie oparte na przypadkach użycia:** Czarnoskrzynkowa technika testowania, w której przypadki testowe są projektowane w celu wykonywania zachowań przypadków użycia.

**Testowanie w oparciu o historyjki użytkownika (user story testing):** Czarnoskrzynkowa technika projektowania testów, w której przypadki testowe są projektowane w oparciu o historyjki użytkownika, by zweryfikować poprawność ich implementacji.

**Ukończenie testów (test completion):** Czynność obejmująca udostępnianie testaliów dla późniejszego użycia, pozostawianie środowisk testowych w zadowalającym stanie i komunikowanie wyników testowania odpowiednim interesariuszom.

**Warunek testowy (test condition):** Testowalna własność modułu lub systemu zidentyfikowana jako podstawa do testowania.

**Warunek wstępny (precondition):** Wymagany stan elementu testowego i jego środowiska przed wykonaniem przypadku testowego.

**Warunek wyjściowy (postcondition):** Oczekiwany stan elementu testowego i jego środowiska po zakończeniu wykonywania przypadku testowego.

* 1. **Wymagania funkcjonalne**

* 1. **Wymagania niefunkcjonalne**
  2. **Przypadki użycia**

1. **Zastosowane technologie**
   1. **Platforma i środowisko**
   2. **Selenium WebDriver**

Wprowadzenie

Architektura WebDriver

Zgodność z przeglądarkami

Wsparcie językowe

Prędkość

Metody i klasy - ficzery

* 1. **Języki programowania - Python**
  2. **Pytest**

**4. Projekt testów**

**5.**

**7. Bibliografia**

1. Słownik terminów testowych ISTQB® wersja 3.4 (2020) International Software Testing Qualifications Board® oraz Stowarzyszenie Jakości Systemów Informatycznych