

PLAN TESTÓW

1. Identyfikator planu testów

PTA_practicesoftwaretesting.com_2023.04

Plan Testów Automatycznych, testowana aplikacja webowa - practicesoftwaretesting, data tworzenia planu testów - kwiecień 2023.

2. Wprowadzenie

Obiekt testów - aplikacja webowa, która jest typową stroną e-commerce i znajduje się pod adresem: [<https://practicesoftwaretesting.com/#/>] (<https://practicesoftwaretesting.com/#/>). Jest to sklep internetowy zajmujący się sprzedażą narzędzi. Dana aplikacja bazuje na funkcjonalnościach wspólnych dla większości sklepów internetowych. Główne z nich to rejestracja i logowanie, sortowanie oraz filtrowanie produktów według różnych charakterystyk, dodawanie produktów do koszyka, płatność i realizacja zamówienia.

Cele testowania:

- sprawdzenie zgodności działania kluczowych funkcjonalności aplikacji webowej z oczekiwaniami użytkowników;
- weryfikacja spełnienia kluczowymi funkcjonalnościami aplikacji wymagań technicznych i biznesowych;
- weryfikacja stabilności działania procesów rejestracji oraz logowania poprzez wielokrotne przeprowadzenie testów zautomatyzowanych na danym obszarze;
- weryfikacja kompatybilności aplikacji webowej z dwoma najpopularniejszymi przeglądarkami (google chrome oraz firefox mozilla), poprzez przeprowadzenie na nich testów funkcjonalności kluczowych;
- wykonanie wszystkich testów manualnych i automatycznych;
- wykrycie i zaraportowanie maksymalnej liczby błędów na testowanych obszarach, jeśli takie zaistniały.

3. Zakres testów

Planowane jest przetestowanie następujących funkcjonalnych składników aplikacji :

- proces rejestracji nowego użytkownika
- proces logowania
- dodawanie produktów do koszyka
- proces płatności i realizacji zamówienia
- spójność wyświetlanych informacji

4. Wyłączenia z zakresu

Ze względu na mniejszą wagę priorytetową, na danym etapie testowania nie zostaną przetestowane następne składniki funkcjonalne:

- sortowanie oraz filtrowanie produktów
- rozdział "Contact"
- rozdział "Favorites"
- rozdział "Profile"
- rozdział "Invoices"
- rozdział "Messages"

5. Podejście do testowania

W związku z zdefiniowanymi celami, testowanie będzie przeprowadzane na systemowym poziomie, z wykorzystaniem głównie funkcjonalnego typu testów. Pozostałe typy testów (takie jak testowanie atrybutów нефункциональных, testowanie struktury/architektury, testowanie związane ze zmianami) nie będą przeprowadzane w danym przedsięwzięciu testowym.

Testy będą wykonywane w sposób manualny. Także zaplanowana jest automatyzacja tych testów, wielokrotne wykonywanie w sposób manualny których, zajęłoby dużo czasu. W taki sposób zostanie osiągnięta optymalizacja procesu testowania oraz zmniejszenie jego czasochłonności. Automatyzacja testów będzie realizowana za pomocą Selenium IDE oraz Cucumber i Selenium WebDriver

W celu skutecznego wyszukiwania błędów, będzie zaprojektowana duża liczba negatywnych przypadków testowych.

Przy realizacji danego projektu stosowane jest podejście scrumowe. Do wizualizacji zadań oraz skutecznego zarządzania nimi przez zespół będzie wykorzystane narzędzie trello. Przypadki testowe będą prowadzone za pośrednictwem narzędzia Testlink. W celu łatwości dostępu każdego członka zespołu do backlogu danego testowania, prowadzony on będzie w specjalnie stworzonym repozytorium GitHub.

6. Kryteria

Kryteria rozpoczęcia testowania:

- testowana aplikacja jest dostępna
- środowisko testowe jest skonfigurowane i gotowe
- skończona i gotowa do wykorzystania specyfikacja w postaci przypadków testowych, co najmniej do jednego obszaru testowego

Kryteria zakończenia testowania:

- wykonane wszystkie manualne i automatyczne przypadki testowe wszystkich priorytetów na obu przeglądawkach
- zakończenie czasu przeznaczanego na przeprowadzenie testowania

7. Czynności i zadania testowania

Czynności wykonywane w ramach realizacji danego przedsięwzięcia, zdefiniowane są w postaci harmonogramu, podzielonego przez członków zespołu na tygodniowe sprinty, po wykonaniu każdego z których będzie przeprowadzona retrospektywa.

Harmonogram:

Sprint I 30.03.2023:

- wybór obiektu testowania
- podział ról i odpowiedzialności w zespole
- stworzenie repozytorium GitHub
- stworzenie tablicy zadań w Trello
- przygotowanie środowiska testowego

Sprint II 06.04.2023:

- analiza funkcjonalności testowanej aplikacji
- przygotowanie planu testów
- przygotowanie scenariuszy testowych
- przygotowanie przypadków testowych

Sprint III 13.04.2023:

- automatyzacja testów
- wykonywanie testów
- sporządzenie raportów z wykonanych testów
- projekt prezentacji z postępów prac

Zakończenie prac oraz ostatnia retrospektywa 20.04.2023

8. Środowiska testowe

Urządzenia na których będą wykonywane testy:

- 4 komputery z systemem Windows 10

Przeglądarki w których będzie testowana aplikacja webowa:

- Google Chrome wersja 112.0.5615.137
- Mozilla Firefox wersja 112.0.1

Narzędzia dla prowadzenie specyfikacji testowej:

- Testlink 1.9.20 [DEV]

Narzędzia do automatyzacji testów:

- Selenium IDE wersja 3.17.2
- Cucumber
- Selenium WebDriver

Prowadzenie backlogu testowania:

- Repozytorium GitHub (projekt_zespolowy)

Narzędzia wspomagające:

- Microsoft office word 365
- Microsoft office excel 365
- trello.com

9. Role i odpowiedzialności

Oleksandr Tarasenko (analityk, tester automatyzujący):

- napisanie scenariuszów testowych;
- napisanie przypadków testowych;
- tworzenie i wspólne edytowanie planu testów
- automatyzacja testów za pomocą Cucumber oraz Selenium WebDriver

Kasper Szczęch

Dominika Żołyński

- wykonywanie testów manualnych i automatycznych;
- raportowanie wyników testów;
- tworzenie i wspólne edytowanie dokumentacji;

Daniel Zalewski

- wykonywanie testów manualnych;
- raportowanie wyników testów;
- tworzenie i wspólne edytowanie dokumentacji;

10. Metryki

- Rezultaty według priorytetów;
- Rezultaty według słów kluczowych;
- Rezultaty według zestawów testowych;
- Czas wykonania testów.

11. Ryzyka i plany awaryjne

Ryzyka projektowe:

- Opóźnienie w harmonogramie projektu, spowodowane brakiem dostępności członków zespołu na czas. Rozwiązanie - wykonanie wyłącznie najwyższych za priorytetem zadań wynikających z celów testowania.
- Brak wymagań funkcjonalnych - może prowadzić do popełnieniu błędów w testowaniu. Będzie stosowane doświadczenia testerów dla wiedzy o tym jak ma działać aplikacja webowa.
- Nieznajomość narzędzi i technologii użytych w projekcie. Wyszukiwanie materiałów w Internecie w celu nauki zastosowania wykorzystywanych narzędzi.

Ryzyka środowiskowe:

- Awaria sprzętu lub oprogramowania w środowisku testowym. Zaangażowanie dodatkowego wspomagającego sprzętu oraz oprogramowania
- Niekompatybilność przeglądarek internetowych z aplikacją. Poszukiwanie kompatybilnych wersji przeglądarek.