

1.Purpose / cel raportu

Celem projektu było udokumentowanie procesu testowego dla aplikacji sklepu internetowego KOMPUTRONIK.PL- funkcjonalności dotyczące dodawania produktu do koszyka, wyświetlania oraz sprawdzenie funkcjonalności koszyka z produktami

2.Application overview / Opis projektu

Projekt obejmował sprawdzenie zgodności założeń - dokumentacji projektowej z aplikacją - sklep internetowy Komputronik.pl uruchomioną w środowisku produkcyjnym, Zakres procesu testowego obejmował:

- sprawdzenie możliwości dodawania produktów do koszyka z poziomu karty produktu,
- strony wizualnej koszyka
- funkcjonalności koszyka związanej z limitami kwotowymi i ilościowymi
- możliwości dodawania produktów na stronie koszyka
- przenoszenia produktów do schowka
- udostępnianie koszyka

Proces testowy obejmował testy systemowe i testy eksploracyjne.

Przypadki testowe zostały utworzone, opisane i wykonane w testrailu.

Defekty zostały utworzone i przekazane dalej do zespołów developerskich poprzez utworzone (tickety)"bugi" w Jira

3.Testing scope / Zakres testu

Proces testowy przeprowadzono na różnych systemach operacyjnych Windows, MacOS oraz urządzeniach mobilnych - Android i iOS

W ramach systemów operacyjnych Windows i MacOS przeprowadzono na przeglądarkach Chrome, Firefox, Safari, Opera

4.Metrics / Metryki (Statystyki)

Test cases planned / Zaplanowane przypadki testowe	Test cases executed / Wykonane przypadki testowe	Test cases passed / Zaliczone przypadki testowe	Test cases failed / Niezaliczone przypadki testowe
19	19	6	13

5. Types of tests performed / Typy przeprowadzonych testów

5.1 Exploratory testing / Testy eksploracyjne

Testowanie eksploracyjne polega na projektowaniu, wykonywaniu i rejestrowaniu testów nieformalnych (tj. testów, które nie zostały wcześniej zaprojektowane) oraz dokonywaniu ich oceny w sposób dynamiczny podczas ich wykonywania. Rezultaty testów dostarczają wiedzy na temat modułu lub systemu, a także pomagają tworzyć testy dotyczące obszarów wymagających dalszego przetestowania.

Testy wykonane zostały w założonych obszarach aplikacji tj dodawanie produktów do koszyka oraz działania dotyczące funkcjonalności koszyka.

Testy przeprowadzono w celu zapoznania się z ogólnym odbiorem funkcjonalności, użyta kolorystyką, zbadaniu intuicyjności w poruszaniu się po koszyku.

5.2 Testy systemowe

Testy systemowe zaprojektowane zostały zgodnie z przygotowaną dokumentacją projektową i ich celem było zbadanie zgodności przygotowanej aplikacji z dokumentacją projektową. Przygotowano i wykonano 19 przypadków testowych w testrailu,

5.3 Acceptance testing / Testy akceptacyjne - aplikacja została przekazana do t. akc. po przeprowadzeniu t. system.

6. Test environment & tools / Środowisko testowe & narzędziowe

6.1 Browsers

Chrome, firefox, Opera, safari

6.2 Devices

desktop ThinkPad, ihpne11, Samsung Galaxy S20, dodatkowo wykorzystano dostęp do wirtualnych urządzeń mobilnych

6.3 OS

Windows 11 pro, iOS16.1.2,

6.4 Backend environment

środowisko produkcyjne

7. General tests results / Rezultaty testów

-> słowny opis przedstawiający poziom jakości

Zgodnie z przeprowadzonymi testami poziom testów zaliczonych do ogólnej liczby przeprowadzonych testów jest bardzo niski. Testy systemowe wskazują na duże rozbieżności aplikacji z dokumentacją. Założenia dotyczące limitów kwotowych oraz ilościowych produktów w koszyku są niezgodne z dokumentacją.

Nie została utworzona możliwość dodawania produktów do schowka przez niezalogowanych użytkowników. Również udostępnianie produktów jest niemożliwe dla niezalogowanego użytkownika

Testy eksploracyjne wykazały mało intuicyjne poruszanie się po aplikacji, duża ilość wyświetlanych ekranów podczas dodawania produktu do koszyka, proces dodawania produktu do koszyka wymaga przejścia kilku ekranów aby dość do karty produktu i dodać produkt do koszyka co może być zniechęcające dla potencjalnego użytkownika

→ lista defektów (linki do bugów w Jirze)

<https://sdatestingbootcamp.atlassian.net/browse/PK-67>

<https://sdatestingbootcamp.atlassian.net/browse/PK-72>

<https://sdatestingbootcamp.atlassian.net/browse/PK-86>

-> grafy (zrzuty z ekranu) z Testraila z wynikami test run'ów



<input type="checkbox"/> T1	Mozliwosci intereakcji na ekranie "produkt dodany do koszyka"	Passed ▾
<input type="checkbox"/> T2	desktop - Mozliwosci intereakcji na ekranie "produkt dodany do koszyka"	Failed ▾
<input type="checkbox"/> T3	desktop - dodawanie produktu do koszyka z poziomu karty produktu	Passed ▾
<input type="checkbox"/> T4	desktop - sprawdzenie funkcjonalności ekranu "produkt dodany do koszyka"	Passed ▾
<input type="checkbox"/> T5	desktop - 1 - sprawdzenie zawartości ekranu Koszyka	Passed ▾
<input type="checkbox"/> T6	desktop - 2 sprawdzenie zawartości ekranu Koszyka - elementy listy produktów	Passed ▾
<input type="checkbox"/> T7	desktop - 3 sprawdzenie funkcjonalności ekranu koszyka - pole liczba	Failed ▾
<input type="checkbox"/> T8	desktop - sprawdzenie usuwania produktu z koszyka	Failed ▾
<input type="checkbox"/> T9	desktop - sprawdzenie limitu kwotowego koszyka	Failed ▾
<input type="checkbox"/> T10	desktop - sprawdzenie limitu ilościowego różnych produktów w koszyku	Failed ▾
<input type="checkbox"/> T11	desktop - przenoszenie produktu do schowka z poziomu koszyka	Failed ▾
<input type="checkbox"/> T12	desktop - przechowywanie produktu w schowku dla niezalogowanego użytkownika	Failed ▾
<input type="checkbox"/> T13	desktop - sprawdzenie naliczenia darmowej dostawy	Failed ▾
<input type="checkbox"/> T14	desktop - sprawdzenie inf. o brakującej kwoty do naliczenia darmowej dostawy	Failed ▾
<input type="checkbox"/> T15	desktop - sprawdzenie możliwości udostępnić swój koszyk	Failed ▾
<input type="checkbox"/> T16	desktop - sprawdzenie powrotu z trybu udostępniania koszyka	Failed ▾
<input type="checkbox"/> T17	sprawdzenie prawidłowej klasy równoważności dla pola Liczba	Passed ▾
<input type="checkbox"/> T18	sprawdzenie klasy równoważności dla pola Liczba w koszyku	Failed ▾
<input type="checkbox"/> T32	spoarwdzenie wartości maksymalnej zamówieni 40000	Failed ▾

1. Risks / Ryzyka

1. duża ilość błędów w różnych miejscach aplikacji może spowodować, że podczas usuwania błędów pojawią się nowe
2. konieczność ścisłej współpracy dwóch zespołów developerskich w dwóch lokalizacjach i różnych krajach powoduje brak właściwego przepływu informacji
3. rozproszony zespół developerski - konieczność ścisłej współpracy przy retestach i testach regresji

2. Recommendations / Rekomendacje

1. uproszczenie procesu dodania produktu do koszyka
2. większe skupienie się na testowaniu tego obszaru
3. wykonanie dodatkowych testów