

# Test plan BG

## 1. INTRODUCTION

### 1.1 Purpose

Ten dokument jest planem testowym dla aplikacji e-commerce z funkcjonalności koszyka. Celem tego planu jest zapewnienie e aplikacji działa prawidłowo i spełnia wymagania.

### 1.2 Project overview

#### 1.3 Audience

-> aplikacja internetowa z obszaru...tak tak, tego pewnie już się domyślasz ;) ecommerce.

-> aplikacja ma umożliwić użytkownikowi dodawanie produktu do koszyka, edytowanie jego zawartości oraz przechodzenie do etapu finalizacji zamówienia

-> aplikacja ma być dostosowana do trybu desktop oraz mobile -> budżet projektu - na bogato, a co tam: \* zespół QA: 3 MT (manual tester, 2 junior, 1 mid), 1 AT (junior automation tester), 1 TM (Test Manager) \* zespół dev: 3 FE (frontend dev), 2 BE (backend dev), 1 PO (Product Owner), 1 SM (Scrum Master), 1 PM (Project Manager) \* szeroki dostęp do urządzeń testowych + licencja Browserstack

-> zespół międzynarodowy z oddziałami w krajach UE zlokalizowany w Polsce: część zespołu pracuje z Polski, część z Niemiec i Czech. QA podzielone między 3 lokalizacje

-> zależności między systemami: "nasz" zespół tworzy front, ale testy mają zapewnić także jako integracyjnych; za warstwę BE odpowiada firma zewnętrzna ze Skandynawii

-> PO/biznes pracuje w niemieckiej kulturze pracy: przywiązanie do standardów, procedur, dokumentacji -> praca w metodzie Scrum

-> stack narzędziowy: Jira & rodzina -> zespół ma w planach stosować innowacyjne architektury tworzenia oprogramowania

-> odbiorcami aplikacji/testów będzie PO oraz zarząd firmy zlecającej projekt

### 2.1 Tests goal

1. Zapewnienie e aplikacji działa zgodnie z oczekiwaniami PO.
2. Sprawdzenie czy dodawanie, usuwanie w koszyku działa poprawnie.
3. Sprawdzenie czy cena produktów z koszyka jest podawana dokładnie z aktualnymi cenami w sklepie.

### 2.2 Test Assumptions

1. Sprawdzenie czy produkty są poprawnie dodawane i wyświetlane na stronie.
2. Sprawdzenie aktualizacji systemów oprogramowania oraz narzędzi do potrzeb testów.

### 2.3 Levels and types of testing

#### • *Testowanie funkcjonalności koszyka*

1. Dodawanie produktów do koszyka:
  - a. Dodawanie produktu do koszyka powinno powodować jego dodanie do listy produktów w koszyku.
  - b. Po dodaniu produktu do koszyka powinna pojawić się informacja o dodaniu produktu do koszyka.
  - c. Możliwość dodania kilku produktów do koszyka i wyświetlenie ich na liście.
2. Edytowanie zawartości koszyka
  - a. Dodawanie lub odejmowanie produktów z koszyka powinno powodować odpowiednie zmiany w cenie.
  - b. Edytowanie ilości produktów powinno być możliwe tylko dla liczb całkowitych.
3. Przechodzenie do realizacji zamówienia
  - a. Dokonanie zakupów powinno być możliwe po dodaniu produktów do koszyka oraz możliwości wyboru kliku walut np: EUR, USD, PLN, BTC, PLN oraz przewalutowanie.
  - b. W etapie realizacji zamówienia powinna być wyświetlana lista produktów oraz cena.
  - c. Wybór sposobu płatności oraz dostawy.

#### • *Testowanie Kompatybilności:*

1. Testowanie przeglądarek: przetestowanie aplikacji na różnych popularnych przeglądarkach aby upewnić się e działa poprawnie na każdej z nich.
2. Testowanie na różnych systemach operacyjnych takich jak WIN, MacOS oraz Linux.

### 2.4 Test deliverables

### 2.5 Test effort estimation

QA activity	Test effort (MD)
Dokumentacja	3
Coding	5
Design	4
Testowanie	8

## 1. EXECUTION STRATEGY

### 3.1 Entry and exit criteria

Entry criteria: Moliwo dodawania produktów do koszyka

Exit criteria: Obsługa zwrotów i reklamacji

### 3.2 Test cycles

### 3.3 Validation and defect management

Walidacja:

1. Sprawdzenie i zapoznanie si z dokumentacj
2. Wykonanie testów jednostkowych
3. Wykonanie testów integracyjnych
4. Wykonanie testów akceptacyjnych

Zarządzanie bdami

- Zgaszanie bów
- Ocena bów
- Naprawa wystpujących bów
- Kontrola jakości

### 3.4. Test Metrics

Metric	Formula

## 1. TEST MANAGEMENT PROCESS

### 4.1 Test management tool

Jira and all plugins.

### 4.2 Test design process

1. Analiza wymaga
2. Planowanie testów
3. Tworzenie przypadków testowych
4. Implementacja testów
5. Wykonanie testów
6. Raportowanie
7. Kontrola jakości

### 4.3 Test execution process

1. Przygotowanie rodowiska
2. Testy jednostkowe
3. Testy integracyjne
4. Testy funkcjonalne
5. Testy wydajnościowe
6. Testy bezpieczeństwa

7. Raportowanie
8. Po dokonaniu napraw wykonujemy Testy Regresyjne

#### 4.4 Test risks and mitigation factors

Risk	Probability	Impact	Mitigation plan
<ul style="list-style-type: none"> <li>• Brak dostpności potrzebnych narzędzi</li> <li>• Niedotrzymanie terminów</li> <li>• Brak dowiadczczenia zespołu</li> <li>• Bdy spowodowane czynnikiem ludzkim</li> </ul>	<ul style="list-style-type: none"> <li>• Moliwo awarii systemu</li> <li>• Nieprawidłowe działanie aplikacji</li> </ul>	<ul style="list-style-type: none"> <li>•</li> </ul>	

#### 4.5 Test responsibility

- **Test Responsywności:**

1. Przeprowadzenie tstów na różnych rozdzielczociach ekranu np:800x600 i inne aby upewni si e aplikacja jest odpowiednia na róne urzdzenia.
2. Testowanie manualne na różnych urzdzeniach takich jak mobile oraz desktopy.

##### 4.5.1 QA team

QA: 3 MT (manual tester, 2 junior, 1 mid), 1 AT (junior automation tester), 1 TM (Test Manager)

##### 4.5.2 Development team

dev: 3 FE (frontend dev), 2 BE (backend dev), 1 PO (Product Owner), 1 SM (Scrum Master), 1 PM (Project Manager)

QA: 3 MT (manual tester, 2 junior, 1 mid), 1 AT (junior automation tester), 1 TM (Test Manager)

1. TEST ENVIRONMENT : komputery stacjonarne z różnymi systemami, przeglądarkami. Tablety oraz smartfony z różnymi systemami operacyjnymi/
2. TESTING TOOLS : selenium, bdd, devtools, postman.