**Plan testów sklepu internetowego**

Spis treści:

1. Wstęp

2. Zakres testów

3. Przedmiot testów

4. Kryteria zaliczenia

5. Kryteria niezaliczenia

6. Kryteria wejścia

7. Kryteria wyjścia

8. Lista funkcjonalności do przetestowania

9. Środowisko testowe

10. Kategorie błędów

11. Miejsce testów

12. Harmonogram testów

13. Raporty z testów

14. Lista narzędzi

15. Zarządzanie incydentami

16. Role i odpowiedzialność

**1. Wstęp**

Celem tworzonego planu testów jest szczegółowe przetestowanie funkcjonalności sklepu internetowego w oparciu o dostarczone wymagania. Adres sklepu to test-testowanie.oprogramowania.pl

Wszystkie napotkane defekty będą na bieżąco raportowane, poprawiane i przekazane testerom do testów potwierdzających.

**2. Zakres testów**

Poziomy testów objęte w planie testów to:

- testowanie modułowe

- testowanie integracyjne

- testowanie akceptacyjne

Typy testów ujęte w planie testów:

- testy funkcjonalne – oprogramowanie zostanie zweryfikowane pod kątem dostarczonych specyfikacji

- testy integracyjne – testowanie komunikacji pomiędzy API np. wdrożonego modułu przelewy24

- testy automatyczne – implementacja testów automatycznych do kluczowych funkcjonalności systemu według dostarczonych przypadków testowych

Typy testów wykluczone z planu testów:

- testy bezpieczeństwa – brak kompetencji osób w zespole związanych z dziedziną pentestingu

**3. Przedmiot testów**

Przedmiotem testów jest sklep internetowy dostępny pod adresem test-testowanie.oprogramowania.pl wraz z jego wszystkimi funkcjonalnościami.

**4. Kryteria zaliczenia**

a) Testy wydajnościowe

- czas odpowiedzi z serwera nie może być większy niż 500 ms

- oprogramowanie musi wytrzymać obciążenie 1000 użytkowników w tej samej chwili

b) Testy funkcjonalne

- kryteria akceptacyjne są zgodne z obecną implementacją sklepu interentowego

c) Testy automatyczne

- zostały zautomatyzowane wszystkie przypadki testowe

- testy zostały podłączone do narzędzia CI/CD – Jenkins

**5. Kryteria niezaliczenia**

a) testy wydajnościowe

- czas odpowiedzi serwera jest dłuższy niż 500 ms

b) testy funkcjonalne

- zaimplementowane funkcjonalności nie są zgodne z wymaganiami

**6. Kryteria wejścia**

- uruchomione środowisko testowe

- środowisko testowe musi być zbliżone do środowiska produkcyjnego

- dostęp do urządzenia z systemem iOS

**7. Kryteria wyjścia**

- zostały wykonane wszystkie przypadki testów

- wszystkie typy i poziomy testów uwzględnione w planie testów zostały zrealizowane

- poprawione zostały wszystkie defekty

**8. Lista funkcjonalności do przetestowania**

- wyszukiwarka produktów – Przypadek testowy 001

- proces składania zamówienia – Przypadek testowy 002

- integracja z systemem przeelewy24 - Przypadek testowy 003

- tworzenie produktu - Przypadek testowy 004

- panel administracyjny - Przypadek testowy 005

**9. Środowisko testowe**

Testy sklepu internetowego będą przeprowadzane na dedykowanym serwerze testy-testowanie.oprogramowania.pl.

Środowisko testowe jest zbliżone parametrami do środowiska produkcyjnego.

Specyfikacja:

- Procesor: 2 GHz

- RAM: 4 GB

- Dysk: 240 GB

- System: Linux Debian 17

Testy będą przeprowadzane przy użyciu poniższych urządzeń:

1. Komputer PC

- Procesor Intel i4 3,6 GHz

- RAM 16 GB

2. Urządzenie mobilne z systemem iOS Iphone 13

- Procesor Apple A15 Bionic

- Pamięć RAM 4GB

- iOS wersja 16

**10. Kategorie błędów**

Wyróżniamy różne priorytety w naszym planie testów:

- awaria

- błąd blokujący

- błąd poważny

- błąd średni

- błąd trywialny

**11. Miejsce testów**

Testy będą przeprowadzane w siedzibie firmy wytwarzającej oprogramowanie

**12. Harmonogram testów**

a) testowanie statyczne

- weryfikacja przygotowanych dokumentacji

- weryfikacja historyjek użytkownika

Czas: 10h

b) testy funkcjonalne

- weryfikacja zaimplementowanych funkcjonalności zgodnie z przygotowanymi założeniami

- raportowanie defektów

- wykonanie przypadków testowych

Czas: 30h

c) testy wydajnościowe

- weryfikacja czasu odpowiedzi serwera

- weryfikacja obciążenia serwera użytkownikami

Czas: 14h

d) testy automatyczne

- przygotowanie struktury plików

- implementacja testów automatycznych w oparciu o dostarczone przypadki testowe

- podłączenie testów pod narzędzie Jenkins

- podłączenie testów pod narzędzie Grafana

Czas: 40h

**13. Raporty z testów**

- raporty o wykrytych defektach

- zaprojektowane przypadki testowe

- skrypty testów automatycznych

- metryki z testów wydajnościowych

**14. Lista narzędzi**

a) narzędzia do testów wydajnościowych

- Jmeter

b) narzędzia do raportowania defektów

- Jira

c) narzędzia do tworzenia przypadków testowych

- TestLink

- Xray

d) narzędzia do automatyzacji

- WebdriverIO

- SuperTest

- Allure

- Jenkins

- Grafana

e) narzędzia do tworzenia zrzutów ekranu i filmików

- PicPick

- ShareX

**15. Zarządzanie incydentami**

Po wykryciu defektu zostanie on zgłoszony do systemu zarządzania defektami jakim jest Jira. Taki defekt będzie posiadał priorytet wskazany w planie testów oraz osobę, która będzie zajmowała się poprawą błędu.

Gdy programista poprawi taki błąd to zostanie on następnie przekazany testerowi do przeprowadzenia testów potwierdzających.

W momencie poprawnego naprawienia defektu zgłoszenie zostanie zamknięte i przesunięte do kolumny na Gotowe.

W przypadku nieprawidłowego naprawienia defektu zgłoszenie zostanie na nowo przekazane do programisty.

**16. Role i odpowiedzialność**

Kierownik testów – nadzorowanie i organizowanie pracy całego zespołu testowego

Analityk testów – analizowanie podstawy testów, przygotowanie testaliów potrzebnych do testów, wspieranie kierownika testów.

Tester – realizuje zadania wyznaczone przez z kierownika testów, wykonuje testy i raportuje defekty

Administrator techniczny systemu – przygotowuje środowisko testowe i nadzoruje je podczas procesu testowego