# Plan testów dla skleptest.pl

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

**Wstęp**

Celem tworzonego planu testów jest szczegółowe przetestowanie funkcjonalności sklepu w oparciu o dostarczone wymagania.

Adres sklepu to <https://skleptest.pl/>

Wszystkie napotkane defekty zostaną poprawione i przekazane testom do testów potwierdzających.

**Zakres testów**

Poziomy testów objęte w planie testów:

* Testowanie modułowe
* Testowanie integracyjne
* Testowanie akceptacyjne

Typy testów ujęte w planie testów:

* Testy funkcjonalne - oprogramowanie zostanie zweryfikowane pod kątem dostarczonych specyfikacji i danych
* Testy niefunkcjonalne - oprogramowanie zostanie sprawdzone pod kątem wydajności, użyteczności oraz niezawodności
* Testy integracyjne - testowanie integralności z baza danych i interfejsów
* 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
* Testy weryfikacji dokumentów - brak przydzielonej dokumentacji do testów

**Przedmiot testów**

Przedmiotem testów jest sklep internetowy dostępny pod adresem <https://skleptest.pl/> wraz z jego wszystkimi funkcjonalnościami.

**Kryterium zaliczenia**

1. Testy wydajnościowe (niefunkcjonalne):
   1. Czas odpowiedzi serwera nie może być dłuższy niż 1000 ms
   2. Oprogramowanie musi wytrzymać obciążenie 500 użytkowników w tej samej chwili
   3. Obsługa 250 użytkowników bez utraty wydajności
   4. Zapewnienie, że wszystkie funkcje i operacje są dostępne i działają sprawnie podczas obciążenia
2. Testy funkcjonalne:
   1. Wszystkie funkcje aplikacji działają zgodnie z oczekiwaniami
   2. Wszystkie scenariusze testowe zostały wykonane, a wszystkie możliwe ścieżki przetestowane w celu zapewnienia pełnego pokrycia funkcjonalności
   3. Brak krytycznych błędów, które mogą uniemożliwić użytkownikom korzystanie z aplikacji
   4. Aplikacja spełnia określone wymagania biznesowe i celowościowe, umożliwiac osiągniecie założonych celów biznesowych
3. Testy automatyczne:
   1. Testy zostały podłączone do narzędzi CI/CD - Jenkins
   2. Zostały zautomatyzowane wszystkie przypadki testowe

**Kryterium niezaliczenia**

1. Testy wydajnościowe (niefunkcjonalne):
   1. Czas odpowiedzi dłuższy niż 1000 ms
   2. Aplikacja zawiesza się lub przestaje działać przy 500 użytkownikach lub większej ilości
   3. Realizacja zamówień większej liczbie osób niż 250 użytkowników wpływa znacząco na działanie aplikaacji
   4. Któraś z funkcji nie działa prawidłowo lub jest niedostępna dla zwykłego użytkownika strony przy zwiększonym obciążeniu aplikacji
2. Testy funkcjonalne:
   1. Któraś z funkcji nie działa prawidłowo lub jest niedostępna dla zwykłego użytkownika aplikacji
   2. Któryś ze scenariuszy nie przeszedł poprawnie testu
   3. Aplikacja posiada błędy wpływające na funkcjonowanie strony
   4. Nie został spełniony któryś z wymagań akceptacyjnych

**Kryteria wejścia**

* Wszystkie funkcje zaimplementowane i gotowe do testowania
* Środowiska testowe skonfigurowane i dostępne dla zespołu testowego
* Środowisko testowe musi być zbliżone do środowiska produkcyjnego
* Dostępność danych testowych
* Dostęp do urządzenia z systemem iOS

**Kryterium wyjścia**

* Wykonane wszystkie przypadki testowe
* Aplikacja spełnia wszystkie określone kryteria akceptacyjne
* Wszystkie znalezione defekty zostały znalezione, zarejestrowane i przekazane do zespołu deweloperskiego w celu naprawy
* Prace testerskie zostały ukończone zgodnie z harmonogramem

**Lista funkcjonalności do przetestowania**

* Rejestracja użytkownika
* Logowanie użytkownika
* Przypomnienie hasła
* Funkcjonowanie koszyka (dodawanie, usuwanie, aktualizowanie produktu w koszyku)
* Płatność za produkt z koszyka
* Kompatybilność aplikacji z różnymi przeglądarkami oraz urządzeniami mobilnymi
* Formularz kontaktowy

**Środowisko testowe**

Testy sklepu internetowe będą przeprowadzane na dedykowanym serwerze <https://skleptest.pl/>

Środowisko testowe jest zbliżone parametrami do środowiska produkcyjnego.

Specyfikacja:

Procesor: 2 GHz

Ram: 4 GB

Dysk: 100 GB

System: Windows 11 Home 64-bitowy

Testy będą przeprowadzane przy użyciu poniższych urządzeń:

1. Laptop:
   1. System: Windows 10 Home 64-bitowy
   2. Procesor: Intel Core i5 1.6-3.4GHz
   3. Ram: 16 GB
   4. Karta graficzna: NVIDIA GeForce MX150
   5. Dysk: 104 GB
2. Urządzenie mobilne Oppo Reno 3 Pro:
   1. System: Android 10
   2. Procesor: Qualcomm Snapdragon 765G
   3. Ram: 12 GB
   4. Rozdzielczość ekranu: 2400 x 1080 pikseli

**Kategorie błędów**

Wyróżniamy różne priorytety w naszym planie testów:

* Błędy funkcjonalne
* Błędy interfejsu użytkownika
* Błędy wydajnościowe
* Błędy kompatybilności
* Błędy integracji
* Błędy danych
* Awaria
* Błąd blokujący
* Błąd średni
* Błąd poważny

**Miejsce testów**

Testy będą przeprowadzane w siedzibie firmy wytwarzającej oprogramowanie przez testera

**Harmonogram testów**

1. Przygotowanie do testów (Czas 15h):
   1. Opracowanie planu testów w tym strategii, zasobów, harmonogram i priorytety
   2. Przygotowanie środowiska:
   3. Konfiguracja testowa
2. Testy statyczne (Czas 15h):
   1. Weryfikacja historyjek użytkownika
3. Testy funkcjonalne (Czas 40h):
   1. Weryfikacja zaimplementowanych funkcjonalności zgodne z przygotowaniami założeniami
   2. Raportowanie defektów
   3. Wykonanie przypadków testowych
4. Testy wydajnościowe (Czas 20h):
   1. Weryfikacja czasu odpowiedzi serwera
   2. Weryfikacja obciążenia serwera użytkownikami
5. Testy automatyczne (Czas 40h):
   1. Przygotowanie struktury plików
   2. Implementowanie testów automatycznych w oparciu o dostarczone przypadki testowe
   3. Podłączenie testów pod narzędzie Jenkins
   4. Podłączenie testów pod narzędzie Grafana

**Raporty z testów**

* Raporty o wykrytych defektach
* Zaprojektowane przypadki testowe
* Skrypty testów automatycznych
* Metryki z testów wydajnościowych

**Lista narzędzi**

1. Narzędzia do raportowania defektów
   1. Jira
2. Narzędzia do tworzenia przypadków testowych
   1. TestLink
   2. XRay
   3. TestRail
3. Narzędzia do automatyzacji
   1. Jenkins
   2. Grafan
   3. WebDriverIO
4. Narzędzia do testów wydajnościowych
   1. Jmeter
5. Narzędzia do tworzenia zrzutów ekranu i filmików
   1. PicPick
   2. ShareX

**Zarządzanie incydentami**

Po wykryciu defektu tester taki przypadek zgłasza w programie Jira. Będzie ustalony priorytet oraz osobę, która będzie zajmować się poprawieniem takiego błędu.

Po poprawieniu takiego błędu programista przekaże testerowi stosowną informację do przeprowadzenia testów potwierdzających.

W momencie poprawnego naprawienia defektu, zgłoszenie zostanie oznaczone jako zamknięte i przeniesione do kolumny “Gotowe”.

W przypadku, gdy defekt ponownie się pojawi z powodu niepoprawnego naprawienia lub z innego podobnego powodu, zgłoszenie zostanie na nowo zostanie przekazane programiście.

**Role i odpowiedzialność**

1. Kierownik projektu:
   1. Nadzoruje proces testowy
   2. Ustala cele oraz priorytety
   3. Koordynuje prace zespołu testowego i deweloperskiego
2. Analityk testowy:
   1. Analizuje wymagania biznesowe i specyfikacje funkcjonalne
   2. Opracowanie przypadków testowych na podstawie wymagań
   3. Przygotowanie dokumentacji testowej, planu testów oraz scenariuszy testowych
3. Tester funkcjonalny:
   1. Przeprowadza testy funkcjonalne
   2. Raportuje znalezione błędy oraz uwagi
4. Tester wydajnościowy:
   1. Przeprowadza testy wydajnościowe
   2. Tworzenie raportów o wynikach wydajności oraz o problemach związanych z nimi
5. Tester integracyjny:
   1. Przeprowadza testy integracyjne
   2. Sprawdza czy integracja działa zgodnie z oczekiwaniami
   3. Raportowanie rozbieżności
6. Tester kompatybilności:
   1. Przeprowadza testy na różnych platformach, przeglądarkach i urządzeniach
   2. Raportowanie błędów wizualnych i funkcjonalnych
7. Koordynator testów:
   1. Nadzorowanie harmonogramu testów
   2. Przypisywanie zadań testerom
   3. Monitorowanie postępów
8. Inżynier testów automatycznych:
   1. Wykonanie testów automatycznych dla powtarzalnych scenariuszy testowych
   2. Utrzymanie i aktualizowanie zestawu testów automatycznych