

PLAN TESTÓW

MrBuggy 7

Aplikacja: **MrBuggy 7**

URL: <http://mrbuggy.pl/mrbuggy7/>

Wersja: 1.0.0.1

Data: 25/3/2024

Autor: Gabriel Chorzępa

Historia zmian

Wersja	Data	Autor	Komentarz
1.0.01	25.03.2024	Gabriel Chorzępa	Utworzenie dokumentu

Metryka

Nazwa dokumentu	Plan testów aplikacji MrBuggy 7
Autor	Gabriel Chorzępa
Opracowany na potrzeby	http://mrbuggy.pl/
ID	MB7-25032024
Wersja	1.0.0.1
Data	25.03.2024
Liczba stron	10

Spis treści

1. Cel dokumentu.....	3
2. Zakres testów.....	3
3. Przedmiot testów.....	4
4. Kryteria zaliczenia.....	4
5. Kryteria niezaliczenia.....	5
6. Kryteria wejścia.....	5
7. Kryteria wyjścia.....	5
8. Lista funkcjonalności do przetestowania.....	5
9. Środowisko testowe.....	6
10. Kategorie błędów.....	7
11. Miejsce testów.....	7
12. Harmonogram testów.....	7
13. Raporty z testów.....	8
14. Lista narzędzi.....	8
15. Zarządzanie incydentami.....	8
16. Role i odpowiedzialność.....	9

1. Cel dokumentu

Celem niniejszego planu testów jest przetestowanie funkcjonalności aplikacji MrBuggy 7 w oparciu o wymagania dostarczone przez autora. Adres strony z aplikacją oraz wymaganiami do pobrania to <http://mrbuggy.pl/mrbuggy7/>. Ponadto w dokumencie zostaną zawarte informacje dotyczące:

- zakresu testów
- opisu przedmiotu testów
- kryteriów zaliczenia testów
- kryteriów niezaliczenia testów
- kryteriów wejścia
- kryteriów wyjścia
- listy funkcjonalności do przetestowania
- środowiska testowego
- kategorii błędów
- miejsca przeprowadzania testów
- harmonogramu testów
- raportów z testów
- listy narzędzi użytych do przeprowadzenia testów
- zarządzania incydentami
- ról i odpowiedzialności

Wszystkie odnalezione defekty zostaną naprawione i przekazane do zespołu testerów w celu wykonania testów potwierdzających.

Dokument został przygotowany przez wykonawcę i przekazany do zamawiającego do zatwierdzenia.

2. Zakres testów

Poziomy testów uwzględnione w planie testów:

- Testowanie integracyjne
- Testowanie systemowe
- Testowanie akceptacyjne

Typy testów uwzględnione w planie testów:

- Testy funkcjonalne - weryfikacja oprogramowania w odniesieniu do dostarczonej specyfikacji

- Testy API - weryfikacja dostępności i działania endpointów API
- Testy czarnoskrzynkowe - testy eksploracyjne
- Retesty oraz testy regresji

Testy nieujęte w planie testów ze względu na krótki czas na przetestowanie aplikacji oraz brak testerów z dostatecznymi kompetencjami:

- Testy bezpieczeństwa
- Testy wydajnościowe

3. Przedmiot testów

Przedmiotem testów jest aplikacja MrBuggy 7, w zakres których wchodzi przetestowanie interfejsu użytkownika, wprowadzanie danych, zapis danych oraz eksport danych.

4. Kryteria zaliczenia

- Zarządzanie żadaniami zmian (CR - Change Request):
 - pomyślne logowanie i wylogowywanie użytkowników
 - tworzenie nowych CR w stanie "Draft" i ich edycja
 - zmiana statusu CR na "New" po oznaczeniu jako gotowe
 - przeglądanie CR przez pracowników i adminów z odpowiednimi uprawnieniami
 - dodawanie publicznych i prywatnych komentarzy do CR
 - możliwość modyfikacji, akceptacji, odrzucenia i archiwizowania CR
- Zarządzanie użytkownikami
 - tworzenie nowych użytkowników
 - przypisywanie użytkowników do dostawców (Providers)
 - blokowanie i odblokowywanie użytkowników
- Zarządzanie słownikami:
 - zarządzanie słownikiem dostawców i jednostek
 - definiowanie ceny za 1 MD (man-day) dla każdego dostawcy
- Eksport danych:
 - eksport CR do plików JSON i XML

- automatyczne przetwarzanie eksportowanych CR do odpowiednich stanów
- API:
 - dostępność funkcjonalności aplikacji poprzez API
 - prawidłowe działanie endpointów API zgodnie ze specyfikacją
- Automatyczne archiwizowanie:
 - automatyczne przenoszenie CR od stanu “Archived” po zdefiniowanym czasie
 -
- Interfejs użytkownika:
 - poprawne wyświetlanie formularzy i list zgodnie z rolami użytkowników
 - prawidłowe działanie filtrów i przycisków w interfejsie użytkownika

5. Kryteria niezaliczenia

- zaimplementowane funkcjonalności nie są zgodne z warunkami przedstawionymi w wymaganiach
- brak akceptacji przez klienta

6. Kryteria wejścia

- środowisko testowe skonfigurowane zgodnie z wymaganiami do przeprowadzania testów
- środowisko testowe zbliżone do środowiska produkcyjnego
- zakończona faza implementacji dla funkcjonalności podlegającej testowaniu
- zespół posiadający odpowiednie kompetencje do poprawnego przeprowadzenia testów

7. Kryteria wyjścia

- wszystkie przypadki testowe zostały wykonane
- wszystkie typy i poziomy testów zawarte w planie testów zostały zrealizowane
- wszystkie znalezione defekty zostały poprawione

- rezultaty przeprowadzonych testów pokrywają się z wymaganiami zawartymi w dokumentacji

8. Lista funkcjonalności do przetestowania

- Logowanie i wylogowywanie użytkowników:
 - logowanie dla różnych ról użytkowników
 - wylogowywanie i weryfikacja stanu sesji
- Zarządzanie profilami użytkowników:
 - zmiana hasła
 - edycja profilu użytkownika
- Zarządzanie żądaniami zmian (CR):
 - tworzenie nowych CR w stanie "Draft"
 - edycja i zapisywanie zmian w CR
 - zmiana statusu CR na "New"
 - przeglądanie CR przez pracowników i adminów
 - dodawanie komentarzy publicznych i prywatnych
 - modyfikacja, akceptacja, odrzucenie i archiwizacja CR
- Zarządzanie słownikami:
 - zarządzanie słownikiem dostawców
 - zarządzanie słownikiem jednostek
 - definiowanie ceny za 1 MD dla dostawców
- Eksport danych:
 - eksport do plików JSON i XML
 - automatyczne przetwarzanie eksportowanych CR
- API:
 - testowanie dostępności i działania endpointów API
- Automatyczna archiwizacja:
 - automatyczne przenoszenie CR do stanu "Archived"
- Interfejs użytkownika:
 - testowanie poprawności wyświetlania formularzy i list
 - weryfikacja działania filtrów i przycisków
- Zarządzanie użytkownikami:
 - tworzenie nowych użytkowników

- przypisywanie użytkowników do dostawców
- blokowanie i odblokowywanie użytkowników \
- Zarządzanie CR:
 - klonowanie CR dla różnych dostawców
 - zarządzanie statusami CR (np. "Approved", "Rejected")
- Komentarze i historia CR:
 - dodawanie komentarzy do CR
 - przeglądanie historii CR
- Funkcjonalności dostępne w GUI i API:
 - sprawdzanie czy funkcjonalności dostępne w GUI są również dostępne w API

9. Środowisko testowe

Przedmiotem testów jest aplikacja instalowana bezpośrednio na urządzeniu użytkownika, do przeprowadzenia testów przygotowane środowisko testowe jest zbliżone parametrami do środowiska produkcyjnego:

Środowisko testowe

Nazwa komputera	GAB
System operacyjny	Windows 10 Pro 64-bit (10.0.19045 Build 19045)
Język	Angielski
Producent systemu	Lenovo
Model systemu	80NV
BIOS	CDCN25WW (type: Legacy)
Procesor	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.60 GHz
Pamięć	16.0 GB
Dostępna pamięć	15.8 GB
Plik stronicowania	16789MB użytych, 6334MB dostępnych
Wersja DirectX	DirectX 12
Wersja DxDiag	DxDiag 10.00.19041.3636 64-bit Unicode

10. Kategorie błędów

W planie testów wyróżniamy różne priorytety błędów:

- awaria
- błąd blokujący
- błąd poważny
- błąd średni
- błąd trywialny

11. Miejsce testów

Testy zostaną przeprowadzone przez wyznaczonych testerów w siedzibie firmy.

12. Harmonogram testów

1. Testowanie statyczne:

- weryfikacja dokumentacji
- zaprojektowanie przypadków testowych

Czas: 2 dni

2. Testy funkcjonalne:

- testy jednostkowe
- testy integracyjne
- testy systemowe
- testy akceptacyjne użytkownika (UAT)

Czas: 5 dni

3. Testy eksploracyjne:

- testy eksploracyjne
- analiza heurystyczna
- testowanie kreatywne

Czas: 3 dni

4. Retesty:

- retesty zgłoszonych błędów
- weryfikacja poprawek

Czas: 2 dni

5. Testy regresji:

- ręczne testy regresji
- automatyczne testy regresji

Czas: 4 dni

6. Raportowanie:

- przygotowanie raportów z testów
- analiza wyników testów
- rekomendacje

Czas: 1 dzień

7. Przygotowanie do produkcji:

- finalne testy regresji
- sprawdzenie gotowości do wdrożenia

Czas: 2 dni

13. Raporty z testów

- raporty o wykrytych defektach
- zaprojektowane przypadki testowe

Raport zostanie wygenerowany za pomocą narzędzia TestLink.

14. Lista narzędzi

1. Narzędzia do raportowania defektów:

- Jira

2. Narzędzia do tworzenia przypadków testowych:

- TestLink

3. Narzędzia do testowania API:

- Postman
4. Narzędzia do tworzenia zrzutów ekranu i filmików:
- PicPick
 - ShareX

15. Zarządzanie incydentami

Wykryty defekt zostanie zgłoszony do systemu zarządzania błędami Jira. Następnym krokiem będzie ustalenie priorytetu oraz wyznaczenie osoby, która będzie odpowiedzialna za poprawę błędu.

Po naprawieniu błędu przez programistę zostanie on przekazany testerowi do przeprowadzenia testów potwierdzających.

Jeśli defekt nie zostanie poprawnie naprawiony, zostanie ponownie przekazany do programisty.

Gdy błąd zostanie poprawnie naprawiony zgłoszenie zostanie zamknięte i przesunięte do kolumny "Gotowe".

16. Role i odpowiedzialność

- **kierownik testów** - nadzór i organizacja pracy całego zespołu testowego
- **analityk testów** - analiza podstawy testów, przygotowanie tekstyliów potrzebnych do przeprowadzenia testów, wspieranie kierownika testów
- **tester** - realizacja zadań wyznaczonych przez kierownika testów, wykonanie testów, raportowanie defektów
- **administrator techniczny** - przygotowanie środowiska testowego oraz jego nadzór podczas procesu testowego