



Specyfikacja testowa

Projekt testowy: MrBuggy

Wydrukowany przez TestLink dnia 19.02.2023

2012 © TestLink Community

Spis treści

1. Interfejs Pierwszego Logowania

MB-7.0-2: Weryfikacja logiki panelu logowania

MB-7.0-1: Weryfikacja walidacji panelu logowania

Zakres

PLAN TESTÓW

Wersja: 0.1

Autor: Bartosz Adamus

Ostatnio aktualizowane 17.02.2023

Status: W trakcie implementacji

1. Wprowadzenie

1.1 Cel

Celem dokumentu jest określenie jakości aplikacji poprzez testy, które mają za zadanie znaleźć jak największą listę defektów. Testy będą bazować na dostarczonej dokumentacji MrBuggy-Change-Request-1-v1.pdf.

1.2 Przegląd projektu

Aplikacja odpowiada za efektywne zarządzanie change requestami dostawców.

Wersja aplikacji poddawanej testom jest MrBuggy 7.0 a a produktem końcowym będzie MrBuggy 7.1

W przygotowanym dokumencie zostały zebrane kluczowe informacje dot. działań testowych.

Zostały wyszczególnione wszystkie komponenty oprogramowania, które zostaną poddane weryfikacji, walidacji oraz typy testów jakie zostaną przeprowadzone.

1.3 Do kogo skierowany

Dokument jest skierowany do:

- testerów
 - interesariuszy
 - kierownika zespołu (team leadera)
-

2. Strategia Testów

2.1 Zakres testów

Realizowane typy testów:

- funkcjonalne
- jednostkowe
- API

Typy testów które nie będą realizowane, ze względu na ograniczony budżet:

- wydajnościowe
- bezpieczeństwa
- automatyczne

2.2 Przedmioty testów

komponentami poddawanymi testom będą:

- interfejs
- wprowadzenie i zapis danych
- export danych

2.3 Testowane cechy/funkcje:

- przegląd estymowanych kosztów per change request
- przegląd i edycja i dodawanie kosztów 1 MD
- wprowadzanie waluty - usuwanie kosztów
- wyświetlanie obliczonych kosztów
- notyfikacje

2.3 Kryteria zaliczenia/niezaliczenia testów

Kryteria zaliczenia oraz nie zaliczenia testu Testy zaliczane będą na podstawie zgodności z dokumentacją MrBuggy-Change-Request-1-v1 oraz na podstawie akceptacji przez klienta na przeprowadzonym co-dwutygodniowym demie.

3. Przeprowadzanie testów

3.1 Narzędzia potrzebne do przeprowadzania testów:

- Jira
- Testlink
- Postman

3.2 Kryteria wejścia

- działające środowisko testowe,
- zakończona faza implementacji aplikacji,

3.3 Kryteria wyjścia

- Wszystkie dostępne przypadki testowe zostały pomyślnie wykonane,
- Aplikacja spełnia wszystkie wymagania dokumentacji.

3.4 Harmonogram testów

1. Przeprowadzenie testów funkcjonalnych:

- weryfikacja funkcjonalności w oparciu o user story - 1h,
- wykonanie wcześniej zaprojektowanych przypadków testowych - 2h,
- weryfikacja warstwy backend.

2. Przeprowadzenie testów API:

- Weryfikacja ile wynosi średni czas odpowiedzi,
- Weryfikacja czy wszystkie uzyskane odpowiedzi są zgodne z dokumentacją.
- Weryfikacja czy wszystkie zapytania dostępne w dokumentacji wykonywane są poprawnie.

3.5 Śledzenie defektów & raportowanie

wszystkie defekty powinny być raportowane oraz dokumentowane poprzez osobę odpowiedzialną za testy.

4. Zarządzanie testami

4.1 Metodologia pracy:

Scrum

4.2 Narzędzie do zarządzania testami

Do zarządzania testami używana będzie platforma JIRA

4.3 Ryzyko

Największym ryzykiem jest krótki czas przeznaczony na testy.

Dodatkowym ryzykiem jest ograniczony budżet.

5. Środowisko testowe

Środowiskiem testowym będą:

- testowy serwer bazodanowy,
 - System operacyjny windows 10 64 bit.
-

6. Obowiązki oraz odpowiedzialności

Zespół liczy jedną osobę:

Tester/Test leader - Bartosz Adamus - testy funkcjonalne, jednostkowe, API

7. Zatwierdzenie planu

Plan potwierdził test leader Bartosz Adamus.

1.Zestaw testów : Interfejs Pierwszego Logowania

Zestaw będzie zawierał listę przypadków testowych dotyczących interfejsu aplikacji MrBuggy.

| | | |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Przypadek testowy MB-7.0-2: Weryfikacja logiki panelu logowania [Wersja : 1] | | |
| Autor przypadku: | | badamus |
| Cel testu: | | |
| Weryfikacja logiki panelu logowania | | |
| Warunki początkowe: | | |
| 1. Zaimplementowany system logowania | | |
| 2. Zintegrowanie z testowym serwerem bazodanowym. | | |
| #: | Opis operacji: | Spodziewane rezultaty: |
| 1 | Po wpisaniu nieprawidłowych danych w pola panelu rejestracji: 1. "USERNAME" 2. "PASSWORD" 3. "CONFIRM PASSWORD" Zgodnych z założeniami: ad1. Nazwa użytkownika musi zawierać minimum 5 znaków i maksymalnie 20 znaków. ad2. Hasło musi zawierać przynajmniej jeden znak specjalny (!@#\$%?.) i min 6 znaków a maksymalnie 20. | Nie można stworzyć konta administratora. |
| 2 | Po wpisaniu prawidłowych danych w pola panelu rejestracji: 1. "USERNAME" 2. "PASSWORD" 3. "CONFIRM PASSWORD" Zgodnych z założeniami: ad1. Nazwa użytkownika musi zawierać minimum 5 znaków i maksymalnie 20 znaków. ad2. Hasło musi zawierać przynajmniej jeden znak specjalny (!@#\$%?.) i min 6 znaków a maksymalnie 20. | Poprawne stworzenie konta administratora. |
| 3 | Po wpisaniu nieprawidłowych danych w pola panelu logowania: 1. "USERNAME" 2. "PASSWORD" i kliknięciu przycisku "LOGIN" | Nie można się zalogować. |
| 4 | Po wpisaniu prawidłowych danych w pola panelu logowania: 1. "USERNAME" 2. "PASSWORD" i kliknięciu przycisku "LOGIN" | Logowanie powiedzione, przeniesienie do panelu użytkownika |
| Typ wykonania: | | manualne |
| Szacowany czas wykonania (min): | | 60.00 |
| Waga: | | Wysoka |

| | |
|------------------------|------|
| <u>Wymagania</u> | Brak |
| <u>Słowa kluczowe:</u> | Brak |

Przypadek testowy MB-7.0-1: Weryfikacja walidacji panelu logowania [Wersja : 1]Autor przypadku: badamusCel testu:

Weryfikacja poprawności walidacji systemu logowania z dokumentacją.

Warunki początkowe:

1. Zaimplementowany system logowania
2. Zintegrowanie z testowym serwerem bazodanowym.

| <u>#:</u> | <u>Opis operacji:</u> | <u>Spodziewane rezultaty:</u> |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Uruchamiamy aplikacje. | 1. Aplikacja wczytała się pomyślnie, 2. Pojawił się interfejs tworzenia konta administratora. |
| 2 | Klikamy przycisk "SAVE" bez uzupełniania pól "USERNAME", "PASSWORD", "CONFIRM PASSWORD". | Pod każdym polem powinien pojawić się komunikat walidacyjny "This field is required". |
| 3 | Na przemian uzupełniamy pola: 1. Błędnie pole "USERNAME" - Nazwa użytkownika musi zawierać minimum 5 znaków i maksymalnie 20 znaków. 2. Błędnie pole "PASSWORD" - Hasło musi zawierać przynajmniej jeden znak specjalny (!@#\$%?.) i min 6 znaków a maksymalnie 20. 3. Błędnie pole "CONFIRM PASSWORD" - powtarzamy hasło. | 1. Pod polem "USERNAME" powinien pojawić się komunikat walidacyjny "The field must contain from 5 to 20 characters." 2. Pod polem "PASSWORD" powinien pojawić się komunikat walidacyjny "The password must have at least one special character (!@#\$%?.) and number." 3. Pod polem "CONFIRM PASSWORD" powinien pojawić się komunikat walidacyjny "the passwords are not the same" |
| 4 | Poprawnie uzupełniamy pola: 1. "USERNAME", 2. "PASSWORD", 3. "CONFIRM PASSWORD" | Poprawne stworzenie konta administratora i przekierowanie do strony logowania. |
| 5 | W panelu logowania nie uzupełniamy żadnego pól: 1. "LOGIN" 2. "PASSWORD" Klikamy przycisk "LOGIN". | Pod polami: 1. "LOGIN" powinien pojawić się komunikat walidacyjny "This field is required." 2. "Password" powinien pojawić się komunikat walidacyjny "This field is required." 3. Pod przyciskiem "LOGIN" powinien pojawić się komunikat walidacyjny "Incorrect login and / or password." |
| 6 | W panelu logowania na przemian oraz łącznie: 1. Błędnie uzupełniamy pole "USERNAME" do wcześniej stworzonego konta 2. Błędnie uzupełniamy pole "Password" do wcześniej stworzonego konta | Pod przyciskiem "LOGIN" powinien pojawić się komunikat walidacyjny "Incorrect login and / or password." |
| 7 | W panelu logowania wpisujemy poprawne dane do stworzonego wcześniej konta w | Poprawne logowanie oraz przeniesienie do interfejsu użytkownika |

| | | |
|----------------------------------------|-------------------------------------------|--|
| | polach: 1. "USERNAME" 2. "PASSWORD" | |
| <u>Typ wykonania:</u> | manualne | |
| <u>Szacowany czas wykonania (min):</u> | 60.00 | |
| <u>Waga:</u> | Niska | |
| <u>Wymagania</u> | Brak | |
| <u>Słowa kluczowe:</u> | Brak | |