

Plan testów

Aplikacja dla biura turystycznego

Wersja 1.0.2

15 stycznia 2020

Autor

Dominik Wróbel

Spis treści

1	Historia wersji	2
2	Wstęp	2
3	Opis systemu	2
3.1	Najważniejsze funkcje systemu i role użytkowników	2
4	Zakres testów i dokumentu	2
5	Środowisko testowe	3
5.1	Dane ogólne	3
5.2	Biblioteki i narzędzia	3
6	Testy integracyjne - przypadki testowe	4
6.1	TC.1.1 Wyświetlanie modułu admina	4
6.2	TC.1.2 Usuwanie wycieczki (admin)	5
6.3	TC.1.3 Dodawanie wycieczki (admin)	6
6.4	TC.1.4 Tworzenie paska menu dla niezalogowanego użytkownika	8
6.5	TC.1.5 Tworzenie paska menu dla użytkownika zalogowanego z rolą klienta	9
6.6	TC.1.6 Tworzenie paska menu dla użytkownika zalogowanego z rolą admina	10
6.7	TC.1.7 Obliczanie minimalnej i maksymalnej ceny wycieczek	11
6.8	TC.1.8 Obliczanie liczby wszystkich zarezerwowanych wycieczek	12
6.9	TC.1.9 Wyświetlenie panelu filtrowania wycieczek	13
6.10	TC.1.10 Dodawanie wycieczki do koszyka i aktualizacja zarezerwowanych wycieczek . .	14
6.11	TC.1.11 Usunięcie wycieczki z koszyka i aktualizacja zarezerwowanych wycieczek	15
7	Testy systemowe - przypadki testowe	15
8	Raport z testów	15
9	Zalecenia	15
10	Szablon przypadku testowego	15
10.1	TC.1.2 Usuwanie wycieczki	16

1 Historia wersji

Data	Wersja	Opis	Autor
14.01.2020	1.0.0	Pierwsza wersja dokumentu	Dominik Wróbel
15.01.2020	2.0.0	Testy jednostkowe i integracyjne	Dominik Wróbel
16.01.2020	3.0.0	Testy jednostkowe, integracyjne i systemowe	Dominik Wróbel

2 Wstęp

Celem tego dokumentu jest opis przypadków testowych dla systemu biura turystycznego *DreamTrip* (<https://github.com/software123456789/travel-agency>).

3 Opis systemu

System agencji *DreamTrip* to aplikacja webowa wykonana w architekturze *Single Page Application* w technologii Typescript i Angular.

3.1. Najważniejsze funkcje systemu i role użytkowników

- Rola: klient
 - Logowanie, rejestrowanie
 - Wyświetlanie wycieczek
 - Rezerwowanie wycieczek
 - Dodawanie wycieczek do koszyka
 - Usuwanie wycieczek z koszyka
 - Ocenianie wycieczek
 - Komentowanie wycieczek
- Rola: admin
 - Dodawanie wycieczek
 - Usuwanie wycieczek

4 Zakres testów i dokumentu

W zakres tego dokumentu wchodzi opis testów:

- testy jednostkowe
- testy integracyjne
- testy systemowe

Dokument zawiera również analizę raportu z testów oraz zalecenia dla zespołu developerskiego.

5 Środowisko testowe

5.1. Dane ogólne

- System operacyjny: Windows 10
- Baza danych: Firebase
- Przeglądarka: Google Chrome 79
- Środowisko programistyczne: Visual Studio Code 1.41
- Server: Node JS v12.13.0
- npm: v6.12.0
- Angular 8

5.2. Biblioteki i narzędzia

- Jasmine - framework javascript umożliwiający pisanie testów jednostkowych i integracyjnych
- Karma - umożliwia wykonywanie testów, połączenie z przeglądarką
- Protractor - framework pozwalający na pisanie testów systemowych

6 Testy integracyjne - przypadki testowe

6.1. TC.1.1 Wyświetlanie modułu admina

6.1.1. Warunki początkowe

- Zalogowany użytkownik z rolą admin

6.1.2. Podsumowanie

Test ma sprawdzić czy moduł admina działa poprawnie w zakresie wyświetlania informacji raz komunikacji z serwisami zasilającymi go w dane.

6.1.3. Kroki

1. Admin przechodzi do panelu admina

6.1.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.1.1	Poprawna lista wycieczek	Panel admina poprawnie wyświetla wycieczki	-

6.2. TC.1.2 Usuwanie wycieczki (admin)

6.2.1. Warunki początkowe

- Zalogowany użytkownik z rolą admin

6.2.2. Podsumowanie

Test ma sprawdzić czy admin może poprawnie usunąć wycieczkę z systemu. Test sprawdza interakcję komponentów panelu admina oraz serwisu zasilającego te komponenty w dane. Komponent odpowiedzialny za usuwanie danych ma emitować event w przypadku wybrania wycieczki oraz pobierać dane o wycieczce od komponentu rodzica.

6.2.3. Kroki

1. Admin wyświetla panel admina
2. Admin usuwa wycieczkę poprzez wybranie wycieczki z tabeli

6.2.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.2.1	Lista poprawnych wycieczek	Usunięcie z listy wybranej wycieczki	-

6.3. TC.1.3 Dodawanie wycieczki (admin)

6.3.1. Warunki początkowe

- Zalogowany użytkownik z rolą admin

6.3.2. Podsumowanie

Test ma sprawdzić czy admin może poprawnie dodać wycieczkę do systemu. Test skupia się na testowaniu poprawności danych wprowadzanych przez użytkownika oraz sposobie obsługi tych danych przez system. Wymagania wobec danych:

- name, minimum 3 znaki, maksimum 200, pole wymagane
- country, minimum 3 znaki, maksimum 200, pole wymagane
- cena, minimum 1, maksimum 1000000, pole wymagane, liczba
- startDate, data w formacie dd/mm/rrrr, pole wymagane
- endDate, data w formacie dd/mm/rrrr, pole wymagane
- maxAvailableTrips, minimum 1, maksimum 200, pole wymagane, liczba
- tripsDescription, minimum 10, maksimum 500, pole wymagane

6.3.3. Kroki

1. Admin wyświetla panel admina
2. Admin wypełnia formularz i dodaje wycieczkę

6.3.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.3.1	Brak danych	Walidacja zawodzi, przycisk do wysyłania deaktywowany	Testowanie zachowania przy braku wprowadzonych danych
TC.1.3.2	Poprawne dane	Walidacja pomyślna, przycisk do wysyłania aktywny	Testowanie zachowania przy braku wprowadzonych danych
TC.1.3.3	Data w niepoprawnym formacie	Walidacja niepomyślna, komunikat o niepoprawnej dacie	Test formatu danych
TC.1.3.4	Przekroczenie maksymalnej liczby znaków	Walidacja niepomyślna, komunikat o wymaganej liczbie znaków	Testowanie walidatora odpowiedzialnego za długość danych
TC.1.3.5	Zbyt mała liczba	Walidacja niepomyślna, komunikat o błędnej wartości	Testowanie walidatora odpowiedzialnego za zakres liczb
TC.1.3.6	Zbyt duża liczba	Walidacja niepomyślna, komunikat o błędnej wartości	Testowanie walidatora odpowiedzialnego za zakres liczb
TC.1.3.7	Znaki zamiast liczb	Walidacja niepomyślna, komunikat o błędnych danych	Testowanie walidatora odpowiedzialnego za liczby
TC.1.3.8	Wartości graniczne	Walidacja niepomyślna, komunikat o błędnych danych	Testowanie działania dla wartości granicznych walidatorów

6.4. TC.1.4 Tworzenie paska menu dla niezalogowanego użytkownika

6.4.1. Warunki początkowe

- Użytkownik nie jest zalogowany

6.4.2. Podsumowanie

Test ma sprawdzić czy menu paska nawigacyjnego widoczne w sekcji *header* jest poprawnie tworzone dla niezalogowanego użytkownika, tj. zawiera przyciski do logowania i rejestracji. Test sprawdza integrację komponentu nagłówka z serwisem autoryzacji.

6.4.3. Kroki

1. Użytkownik otwiera stronę domową aplikacji

6.4.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.4.1	-	Lista menu wyświetla przyciski do logowania i rejestracji	-

6.5. TC.1.5 Tworzenie paska menu dla użytkownika zalogowanego z rolą klienta

6.5.1. Warunki początkowe

- Użytkownik nie jest zalogowany

6.5.2. Podsumowanie

Test ma sprawdzić czy menu paska nawigacyjnego widoczne w sekcji *header* jest poprawnie wyświetlane dla zalogowanego użytkownika z rolą klienta, tj. zawiera przyciski do wyświetlania wycieczek, koszyka, historii i wylogowania. Test sprawdza także integrację komponentu nagłówka z serwisem autoryzacji.

6.5.3. Kroki

1. Użytkownik przechodzi na stronę domową
2. Użytkownik loguje się do systemu z rolą klienta

6.5.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.5.1	Model użytkownika z rolą klienta i poprawnym emailem oraz hasłem	Wyświetlenie 4 przycisków w pasku menu	Po poprawnym zalogowaniu dla użytkownika z rolą klient widoczne są 4 przyciski

6.6. TC.1.6 Tworzenie paska menu dla użytkownika zalogowanego z rolą admina

6.6.1. Warunki początkowe

- Użytkownik nie jest zalogowany

6.6.2. Podsumowanie

Test ma sprawdzić czy menu paska nawigacyjnego widoczne w sekcji *header* jest poprawnie wyświetlane dla zalogowanego użytkownika z rolą admina, tj. zawiera przyciski do wyświetlania wycieczek, koszyka, historii, wylogowania oraz dodatkowo panel admina. Test sprawdza także integrację komponentu nagłówka z serwisem autoryzacji.

6.6.3. Kroki

1. Użytkownik przechodzi na stronę domową
2. Użytkownik loguje się do systemu z rolą admina

6.6.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.6.1	Model użytkownika z rolą admina i poprawnym emailem oraz hasłem	Wyświetlenie 5 przycisków w pasku menu	Po poprawnym zalogowaniu dla użytkownika z rolą admina widoczne jest 5 przycisków

6.7. TC.1.7 Obliczanie minimalnej i maksymalnej ceny wycieczek

6.7.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

6.7.2. Podsumowanie

Test ma sprawdzić czy najmniejsza i największa cena wycieczek, które komponent otrzymuje z serwisu jest poprawnie wyznaczana. Na podstawie tych wartości są dodawane style do kart wycieczek.

6.7.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek.

6.7.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.7.1	Lista poprawnych wycieczek	Obliczenie największej i najmniejszej ceny wycieczki	-

6.8. TC.1.8 Obliczanie liczby wszystkich zarezerwowanych wycieczek

6.8.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

6.8.2. Podsumowanie

Test ma sprawdzić czy poprawnie obliczana jest liczba wszystkich zarezerwowanych wycieczek. Na podstawie tych wartości do strony dodawane są style.

6.8.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek.

6.8.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.8.1	Lista poprawnych wycieczek	Obliczenie liczby wszystkich zarezerwowanych wycieczek	-

6.9. TC.1.9 Wyświetlenie panelu filtrowania wycieczek

6.9.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

6.9.2. Podsumowanie

Test ma sprawdzić komponent wycieczek poprawnie wyświetla panel do filtrowania wycieczek po kliknięciu na przycisk pokazania panelu.

6.9.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek
2. Użytkownik otwiera panel filtrowania wycieczek

6.9.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.9.1	-	Wyświetlenie panelu filtrowania wycieczek	-

6.10. TC.1.10 Dodawanie wycieczki do koszyka i aktualizacja zarezerwowanych wycieczek

6.10.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie
- Oferta wycieczki jest dostępna do zarezerwowania

6.10.2. Podsumowanie

Test sprawdza czy oferta wycieczki została poprawnie dodana do koszyka, w koszyku przechowywana jest liczba ofert dla danej wycieczki, dodanie dwa razy tej samej oferty powinno zwiększyć liczbę rezerwacji dla danej oferty, ale pozostawić tylko jedną ofertę dla wycieczki w koszyku. Dodanie wycieczki do koszyka powinno spowodować aktualizację liczby zarezerwowanych wycieczek.

6.10.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek
2. Użytkownik dodaje wybraną wycieczkę do koszyka

6.10.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.9.1	-	Dodanie wycieczki do koszyka	-

6.11. TC.1.11 Usunięcie wycieczki z koszyka i aktualizacja zarezerwowanych wycieczek

6.11.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie
- Użytkownik dodał przynajmniej jedną ofertę do koszyka

6.11.2. Podsumowanie

Test sprawdza czy oferta wycieczki została poprawnie dodana do koszyka, w koszyku przechowywana jest liczba ofert dla danej wycieczki, dodanie dwa razy tej samej oferty powinno zwiększyć liczbę rezerwacji dla danej oferty, ale pozostawić tylko jedną ofertę dla wycieczki w koszyku. Usunięcie wycieczki z koszyka powinno spowodować aktualizację liczby zarezerwowanych wycieczek.

6.11.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek
2. Użytkownik dodaje wybraną wycieczkę do koszyka

6.11.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.9.1	-	Dodanie wycieczki do koszyka	-

7 Testy systemowe - przypadki testowe

8 Raport z testów

9 Zalecenia

10 Szablon przypadku testowego

10.1. TC.1.2 Usuwanie wycieczki

10.1.1. Warunki początkowe

- Zalogowany użytkownik z rolą admin

10.1.2. Podsumowanie

Test ma sprawdzić czy admin może poprawnie usunąć wycieczkę z systemu.

10.1.3. Kroki

1. Admin wyświetla panel admina
2. Admin usuwa wycieczkę poprzez wybranie wycieczki z tabeli

10.1.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.1.1	Lista poprawnych wycieczek	Usunięcie z listy wybranej wycieczki	-