

Plan testów

Aplikacja dla biura turystycznego

Wersja 2.0.0

16 stycznia 2020

Autor

Dominik Wróbel

Spis treści

1 Historia wersji

Data	Wersja	Opis	Autor
14.01.2020	1.0.0	Pierwsza wersja dokumentu	Dominik Wróbel
15.01.2020	1.0.0	Testy jednostkowe i integracyjne	Dominik Wróbel
16.01.2020	2.0.0	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 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 <https://console.firebase.google.com/u/1/project/travel-agency-ddc82/overview>
- 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 panelu admina (admin)

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 czy 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.1.10.1	Dwie takie same wycieczki	Dodanie jednej wycieczki do koszyka	Dwie takie same wycieczki powinny sprawić, że w koszyku będzie przechowywana tylko jedna z liczbą rezerwacji równą 2
TC.1.10.2	Dwie różne wycieczki	Dodanie dwóch wycieczek do koszyka	Dwie różne wycieczki powinny sprawić, że w koszyku będą przechowywane dwie wycieczki

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.1.11.1	Wycieczka, której nie ma w koszyku	Brak reakcji ze strony koszyka	Aplikacja powinna być zabezpieczona przed nieoczekiwanymi danymi
TC.1.11.2	Wycieczka, która jest w koszyku	Usunięcie wycieczki z koszyka	Normalne działania aplikacji

6.12. TC.1.12 Filtrowanie wycieczek

6.12.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

6.12.2. Podsumowanie

Test sprawdza czy prezentowane dla użytkownika wycieczki są zgodne z kryteriami zaznaczonymi przez użytkownika.

6.12.3. Kroki

1. Użytkownik wyświetla wszystkie oferty wycieczek
2. Użytkownik otwiera panel do ustawiania preferencji
3. Użytkownik wybiera preferencje
4. Użytkownik akceptuje wprowadzone preferencje

6.12.4. Dane testowe TODO

Id	Dane	Oczekiwanie	Uzasadnienie
TC.1.12.1	Filtrowanie po cenie, cena minimalna 1400, cena maksymalna 5000	System wyświetla wycieczki w zadanym przedziale cenowym	Testowanie jednej z opcji filtrowania
TC.1.12.2	Filtrowanie po dacie, od daty 12/01/2019, do daty 01/01/2020	System wyświetla wycieczki w zadanym przedziale czasowym	Testowanie jednej z opcji filtrowania
TC.1.12.3	Filtrowanie po kraju, wycieczki do kraju 'Poland'	System wyświetla wycieczki do zadanego kraju	Testowanie jednej z opcji filtrowania
TC.1.12.4	Filtrowanie po ratingu, minimalny rating równy 3.0	System wyświetla wycieczki z ratingiem większym niż zadany	Testowanie jednej z opcji filtrowania

7 Testy systemowe - przypadki testowe

7.1. TC.2.1 Wyświetlanie wiadomości powitalnej

7.1.1. Warunki początkowe

- Brak

7.1.2. Podsumowanie

Test ma sprawdzić czy wiadomość powitalna na stronie domowej wyświetla się poprawnie.

7.1.3. Kroki

1. Użytkownik przechodzi na stronę domową

7.1.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.1.1	-	Wyświetlenie wiadomości powitalnej	-

7.2. TC.2.2 Wyświetlenie menu dla niezalogowanego użytkownika

7.2.1. Warunki początkowe

- Użytkownik nie jest zalogowany

7.2.2. Podsumowanie

Test ma sprawdzić czy linki dla niezalogowanego użytkownika są wyświetlane poprawnie.

7.2.3. Kroki

1. Użytkownik przechodzi na stronę domową

7.2.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.2.1	-	Wyświetlenie w menu linku do logowania i zakładania konta	-

7.3. TC.2.3 Logowanie użytkownika z rolą klienta i dostosowanie paska menu

7.3.1. Warunki początkowe

- Brak

7.3.2. Podsumowanie

Test ma sprawdzić czy system poprawnie loguje użytkownika oraz jak reaguje w przypadku nieprawidłowych danych logowania. Oprócz tego po poprawnym zalogowaniu system powinien dostosować pasek menu do użytkownika i jego roli.

7.3.3. Kroki

1. Użytkownik loguje się do systemu niepoprawnymi danymi
2. Użytkownik loguje się do systemu poprawnymi danymi z rolą klient
3. Użytkownik przechodzi na stronę z ofertami

7.3.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.3.1	Niepoprane dane logowania	System pozostaje na stronie domowej, brak zmiany paska menu	Testowanie działania dla niepoprawnych danych
TC.2.3.2	Poprawne dane logowania	System przechodzi na stronę z wycieczkami, dostosowanie paska menu	Testowanie działania dla poprawnych danych

7.4. TC.2.4 Logowanie użytkownika z rolą admina i dostosowanie paska menu

7.4.1. Warunki początkowe

- Brak

7.4.2. Podsumowanie

Test ma sprawdzić czy system poprawnie loguje użytkownika oraz jak reaguje w przypadku nieprawidłowych danych logowania. Oprócz tego po poprawnym zalogowaniu system powinien dostosować pasek menu do roli admina.

7.4.3. Kroki

1. Użytkownik loguje się do systemu niepoprawnymi danymi
2. Użytkownik loguje się do systemu poprawnymi danymi z rolą admin
3. Użytkownik przechodzi na stronę z ofertami

7.4.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.3.1	Niepoprane dane logowania	System pozostaje na stronie domowej, brak zmiany paska menu	Testowanie działania dla niepoprawnych danych
TC.2.3.2	Poprawne dane logowania	System przechodzi na stronę z wycieczkami, dostosowanie paska menu	Testowanie działania dla poprawnych danych

7.5. TC.2.5 Wyświetlanie kryteriów filtrowania

7.5.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

7.5.2. Podsumowanie

Test ma sprawdzić czy system poprawnie prezentuje panel do filtrowania wycieczek.

7.5.3. Kroki

1. Użytkownik przechodzi na stronę z ofertami wycieczek
2. Użytkownik wybiera przycisk panelu filtrowania

7.5.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.5.1	-	Poprawne wyświetlenie panelu filtrowania	-

7.6. TC.2.6 Dodawania i usuwanie wycieczki z koszyka

7.6.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

7.6.2. Podsumowanie

Test ma sprawdzić czy system poprawie wyświetla elementy, które służą do dodawania i usuwania elementu z koszyka. Przycisk dodawania jest dostępny po zalogowaniu o ile wycieczka jest dostępna. Przycisk usuwania jest dostępny jeśli użytkownik dodał wybraną wycieczkę do koszyka.

7.6.3. Kroki

1. Użytkownik przechodzi na stronę z ofertami wycieczek
2. Użytkownik dodaje wycieczkę do koszyka
3. Użytkownik usuwa wycieczkę z koszyka

7.6.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.6.1	-	System wyświetla przycisk dodawania wycieczki, system wyświetla przycisk usuwania wycieczki po dodaniu wycieczki do koszyka	-

7.7. TC.2.7 Oznaczanie wycieczki z małą liczbą dostępnych ofert

7.7.1. Warunki początkowe

- Użytkownik jest zalogowany w systemie

7.7.2. Podsumowanie

Test ma sprawdzić czy wycieczki z liczbą dostępnych ofert mniejszą niż 4 są poprawnie wyświetlane, tj. czy mają odpowiednią wartość koloru dla tła.

7.7.3. Kroki

1. Użytkownik przechodzi na stronę z wycieczkami
2. Użytkownik dodaje do koszyka 4 z 5 dostępnych wycieczek

7.7.4. Dane testowe

Id	Dane	Oczekiwanie	Uzasadnienie
TC.2.7.1	-	Wycieczka wyświetlania jest na innym kolorze tła	-

8 Raport z testów

```
1 ng test
2 ...
3 Chrome 79.0.3945 (Windows 10.0.0): Executed 27 of 27 SUCCESS (9.044 secs / 8.962 secs)
4 TOTAL: 27 SUCCESS
5 TOTAL: 27 SUCCESS
```

```
1 ng e2e
2 ...
3 Executed 7 of 7 specs (2 FAILED) in 51 secs.
4 [18:42:16] I/launcher - 0 instance(s) of WebDriver still running
5 [18:42:16] I/launcher - chrome #01 failed 2 test(s)
6 [18:42:17] I/launcher - overall: 2 failed spec(s)
7 [18:42:17] E/launcher - Process exited with error code 1
```

Karma v4.4.1 - connected

Chrome 79.0.3945 (Windows 10.0.0) is idle

Jasmine 2.99.0

.....

27 specs, 0 failures

Admin Add Component form validation
 should validate empty form
 should disable submit button
 should validate name field
 should validate date field
 should validate price field
 should validate submit

Admin Remove Component Integration
 should display trip to be removed
 should trigger event emitter on remove

Admin Component and Trips Service
 should initialize all trips
 should receive all trips from trips service
 should add new trip
 should remove selected trip
 should display trips in the DOM

Header Component and Auth Service
 should create component
 should display menu for anathorised users
 should display menu for logged admin
 should display menu for logged user
 should display menu after user logout

Trips Filtering and Trips Service
 should create filtering component
 should filter by low and high price

Trips Component and Trips Service
 should create trips component
 should find min and max trips price
 should find all booked trips number
 should show filtering criteria panel
 should add trip to user basket
 should remove trip to user basket
 should remove trip to user basket

```

1 ng test --watch=false --code-coverage
2
3 ===== Coverage summary =====
4 Statements : 37.13% ( 186/501 )
5 Branches   : 14.55% ( 8/55 )
6 Functions  : 27.37% ( 49/179 )
7 Lines      : 33.62% ( 155/461 )
8 =====

```

File		Statements	Branches	Functions	Lines
src	<div></div>	100%	3/3	100%	0/0
src/app	<div></div>	15%	24/160	0%	0/26
src/app/admin/admincomponent	<div></div>	88.89%	8/9	100%	0/0
src/app/admin/admincomponent/admin-add-trip	<div></div>	100%	18/18	100%	0/0
src/app/admin/admincomponent/admin-remove-trip	<div></div>	100%	6/6	100%	0/0
src/app/app-view	<div></div>	100%	2/2	100%	0/0
src/app/auth	<div></div>	12.24%	6/49	0%	0/4
src/app/auth/login	<div></div>	20%	2/10	100%	0/0
src/app/auth/signup	<div></div>	20%	2/10	100%	0/0
src/app/footer	<div></div>	100%	2/2	100%	0/0
src/app/header	<div></div>	100%	14/14	100%	2/2
src/app/models	<div></div>	100%	12/12	100%	0/0
src/app/trips	<div></div>	97.37%	37/38	100%	6/6
src/app/trips/trip-detailed-view	<div></div>	4.17%	3/72	0%	0/17
src/app/trips/tripcarddetails	<div></div>	70.83%	17/24	100%	0/0
src/app/trips/trips-filter-criteria	<div></div>	80.65%	25/31	100%	0/0
src/app/user-basket	<div></div>	6.45%	2/31	100%	0/0
src/app/user-history	<div></div>	22.22%	2/9	100%	0/0
src/environments	<div></div>	100%	1/1	100%	0/0

Kod testów:

- <https://github.com/software123456789/travel-agency-tests.git>

9 Zalecenia

- Raport pokazuje zadowalające pokrycie testami jednostkowymi i integracyjnymi dla komponentów prezentacyjnych, zaleca się jednak wykonanie większej ilości testów dla serwisów
- Naprawy wymaga przycisk do wyświetlania panelu filtrowania (T.C.2.4), który poprawnie otwiera panel, ale przestaje działać po próbie ponownego otworzenia

Komentarz: mała liczba testów jednostkowych i integracyjnych dla serwisów wynika z konieczności mockowania dla tych serwisów serwisu, który zapewniany jest przez moduł angular firestore.