



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII
BIOMEDYCZNEJ

KATEDRA INFORMATYKI STOSOWANEJ

Testowanie i walidacja oprogramowania

Raport z testów

Autor: Weronika Wisz

Kierunek studiów: Informatyka i Systemy Inteligentne

Kraków, 2022

Spis treści

1. Wprowadzenie	3
2. Testy jednostkowe	3
2.1. Testy warstwy serwera	4
2.1.1. Procedura uruchamiania testów	4
2.1.2. Testy serwisów	4
2.1.3. Testy kontrolerów	11
2.2. Testy warstwy klienta	15
2.2.1. Procedura uruchamiania testów	15
2.2.2. Testy serwisów	15
2.2.3. Testy komponentów	19
3. Testy integracyjne	26
3.1. Testy warstwy serwera	26
3.1.1. Testy integracji kontrolerów z serwisami	26
3.1.2. Testy integracji serwisów z bazą danych	29
3.2. Testy warstwy klienta	32
3.2.1. Testy integracji komponentów z serwisami	32
4. Testy systemowe	35
5. Testy bezpieczeństwa	46
6. Testy wydajnościowe	48

1. Wprowadzenie

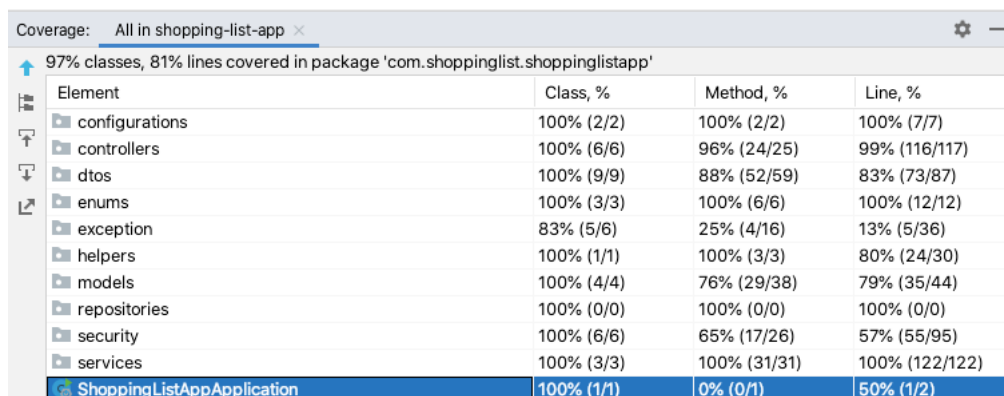
W ramach projektu powstała aplikacja do tworzenia listy zakupów. Zaprojektowana została ona zgodnie z modelem architektury trójwarstwowej typu klient-serwer. Do implementacji warstwy prezentacji wykorzystany został framework Angular. Warstwa aplikacji realizowana była za pomocą frameworku Spring Boot. Przy tworzeniu warstwy danych użyty został PostgreSQL.

Aplikacja składa się z trzech modułów:

- moduł zarządzania użytkownikami - logowanie, rejestracja
- moduł list - przeglądanie, dodawanie, edytowanie, usuwanie list
- moduł produktów - przeglądanie, dodawanie, edytowanie, usuwanie produktów

2. Testy jednostkowe

Testy jednostkowe przeprowadzono w warstwie serwera oraz klienta zaprojektowanej aplikacji. Pokrycie testami wynosi kolejno 81% i 91%. Nie testowano fragmentów kodu związanych z konfiguracją aplikacji, skupiono się na testach związanych z funkcjonalnością aplikacji. Wszystkie komponenty poza testowanymi były mockowane. Niektóre testy pominięto ze względu na ich znaczną ilość.



Element	Class, %	Method, %	Line, %
configurations	100% (2/2)	100% (2/2)	100% (7/7)
controllers	100% (6/6)	96% (24/25)	99% (116/117)
dtos	100% (9/9)	88% (52/59)	83% (73/87)
enums	100% (3/3)	100% (6/6)	100% (12/12)
exception	83% (5/6)	25% (4/16)	13% (5/36)
helpers	100% (1/1)	100% (3/3)	80% (24/30)
models	100% (4/4)	76% (29/38)	79% (35/44)
repositories	100% (0/0)	100% (0/0)	100% (0/0)
security	100% (6/6)	65% (17/26)	57% (55/95)
services	100% (3/3)	100% (31/31)	100% (122/122)
ShoppingListAppApplication	100% (1/1)	0% (0/1)	50% (1/2)

Rysunek 2.1: Pokrycie testami warstwy serwera

```
TOTAL: 87 SUCCESS

===== Coverage summary =====
Statements   : 90.88% ( 389/428 )
Branches     : 75.86% ( 132/174 )
Functions    : 85.9% ( 128/149 )
Lines        : 90.88% ( 339/373 )
=====
```

Rysunek 2.2: Pokrycie testami warstwy klienta

2.1. Testy warstwy serwera

Do testów użyte zostały JUnit oraz Mockito. Poniżej zaprezentowane zostaną przypadki testowe przeprowadzone w ramach testów jednostkowych warstwy serwera.

2.1.1. Procedura uruchamiania testów

Wymagania wstępne:

- Pobrać serwer aplikacji shopping-list-app.
- Java 11, Spring Framework, Maven i Tomcat - do uruchomienia części serwera.
- PostgreSQL - należy utworzyć bazę danych i w pliku application.properties części serwera podać do niej użytkownika, hasło oraz schemat.

Komenda do uruchomienia testów: `./mvnw test`

2.1.2. Testy serwisów

Nazwa	ShoppingListService - Pobranie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Pobranie list zakupów użytkownika
Warunki końcowe	Zwrócona lista, która zawiera listę zakupów o id 1

Tabela 2.1: ShoppingListService - Pobranie list zakupów użytkownika

Nazwa	ShoppingListService - Pobranie list zakupów użytkownika - brak list
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik jest zalogowany i nie ma żadnej listy zakupów
Procedura	Pobranie list zakupów użytkownika
Warunki końcowe	Zwrócona lista jest pusta

Tabela 2.2: ShoppingListService - Pobranie list zakupów użytkownika - brak list

Nazwa	ShoppingListService - Dodanie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Dodanie list zakupów (nazwa: lista, data: 2023-04-08)
Warunki końcowe	Wywołany jest na repozytorium zapis listy z danymi: nazwa: lista, data: 2023-04-08, użytkownik: username, status: oczekująca

Tabela 2.3: ShoppingListService - Dodanie listy zakupów użytkownika

Nazwa	ShoppingListService - Edycja listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Edycja listy zakupów o id 1 (nazwa: newTitle, data: 2023-05-15)
Warunki końcowe	Lista o id 1 ma nazwę zmienioną na newTitle i datę na 2023-05-15

Tabela 2.4: ShoppingListService - Edycja listy zakupów użytkownika

Nazwa	ShoppingListService - Zatwierdzenie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Zmiana statusu listy o id 1 na wykonaną
Warunki końcowe	Lista o id 1 ma ustawiony status wykonana

Tabela 2.5: ShoppingListService - Zatwierdzenie listy zakupów użytkownika

Nazwa	ShoppingListService - Zatwierdzenie listy zakupów i produktów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt 1 - status: wykonany, lista: 1 Produkt 2 - status: oczekujący, lista: 1
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Zmiana statusu listy o id 1 na wykonaną
Warunki końcowe	Produkty listy mają status wykonane

Tabela 2.6: ShoppingListService - Zatwierdzenie listy zakupów i produktów

Nazwa	ShoppingListService - Przywrócenie listy zakupów i produktów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: wykonana Produkt 1 - status: wykonany, lista: 1 Produkt 2 - status: oczekujący, lista: 1
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Zmiana statusu listy o id 1 na oczekującą
Warunki końcowe	Produkty listy mają status oczekująca

Tabela 2.7: ShoppingListService - Przywrócenie listy zakupów i produktów

Nazwa	ShoppingListService - Edycja listy zakupów innego użytkownika
Dane testowe	Użytkownik - nazwa: username2, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Edycja listy zakupów o id 1
Warunki końcowe	Rzucany jest wyjątek ApiForbiddenException

Tabela 2.8: ShoppingListService - Edycja listy zakupów innego użytkownika

Nazwa	ShoppingListService - Usunięcie listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Usunięcie listy zakupów o id 1
Warunki końcowe	Lista zakupów otrzymuje status usunięta

Tabela 2.9: ShoppingListService - Usunięcie listy zakupów

Nazwa	ShoppingListService - Usunięcie listy zakupów innego użytkownika
Dane testowe	Użytkownik - nazwa: username2, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Usunięcie listy zakupów o id 1
Warunki końcowe	Rzucany jest wyjątek ApiForbiddenException

Tabela 2.10: ShoppingListService - Usunięcie listy zakupów innego użytkownika

Nazwa	ShoppingItemService - Pobranie produktów z list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Pobranie produktów z listy 1
Warunki końcowe	Zwrócona lista, która zawiera produkt o id 1

Tabela 2.11: ShoppingItemService - Pobranie produktów z list zakupów użytkownika

Nazwa	ShoppingItemService - Dodanie produktu do listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Dodanie produktu do listy 1 (nazwa: produkt, ilość 2, jednostka: sztuki, obraz: 1 bajt)
Warunki końcowe	Wywołany jest na repozytorium zapis produktu z danymi: nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący, obraz: 1 bajt

Tabela 2.12: ShoppingItemService - Dodanie produktu do listy zakupów

Nazwa	ShoppingItemService - Edycja produktu z listy użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Edycja produktu o id 1 (nazwa: newName, ilość: 8, jednostka: gramy, obraz: 1 bajt)
Warunki końcowe	Produkt o id 1 ma nazwę zmienioną na new Name, ilość na 8, jednostkę na gramy i ma dodany obraz

Tabela 2.13: ShoppingItemService - Edycja produktu z listy użytkownika

Nazwa	ShoppingItemService - Zatwierdzenie produktu z listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Zmiana statusu produktu o id 1 na wykonany
Warunki końcowe	Produkt o id 1 ma ustawiony status wykonany

Tabela 2.14: ShoppingItemService - Zatwierdzenie produktu z listy zakupów

Nazwa	ShoppingItemService - Zatwierdzenie produktu i listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Zmiana statusu produktu o id 1 na wykonany
Warunki końcowe	Lista o id 1 ma status wykonana

Tabela 2.15: ShoppingItemService - Zatwierdzenie produktu i listy zakupów

Nazwa	ShoppingItemService - Przywrócenie produktu i listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: wykonana Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: wykonany
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Zmiana statusu produktu o id 1 na oczekujący
Warunki końcowe	Lista o id 1 ma status oczekująca

Tabela 2.16: ShoppingItemService - Przywrócenie produktu i listy zakupów

Nazwa	ShoppingItemService - Edycja produktu innego użytkownika
Dane testowe	Użytkownik - nazwa: username2, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Edycja produktu o id 1
Warunki końcowe	Rzucany jest wyjątek ApiForbiddenException

Tabela 2.17: ShoppingItemService - Edycja produktu innego użytkownika

Nazwa	ShoppingItemService - Usunięcie produktu z listy zakupów
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Usunięcie produktu o id 1
Warunki końcowe	Produkt otrzymuje status usunięty

Tabela 2.18: ShoppingItemService - Usunięcie produktu z listy zakupów

Nazwa	ShoppingItemService - Usunięcie produktu innego użytkownika
Dane testowe	Użytkownik - nazwa: username2, hasło: password Lista zakupów - id: 1, nazwa: lista, właściciel: username, data: 2023-04-08, status: oczekująca Produkt - id: 1, nazwa: produkt, ilość 2, jednostka: sztuki, lista: 1, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Usunięcie produktu o id 1
Warunki końcowe	Rzucany jest wyjątek ApiForbiddenException

Tabela 2.19: ShoppingItemService - Usunięcie produktu innego użytkownika

2.1.3. Testy kontrolerów

Nazwa	AuthController - Logowanie użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik username istnieje
Procedura	Wywołanie zapytania HTTP POST /auth/signin body - nazwa: username, hasło: password
Warunki końcowe	Status OK Wywołanie metody uwierzytelniania na authenticationManager wygenerowanie tokena jwt

Tabela 2.20: AuthController - Logowanie użytkownika

Nazwa	AuthController - Rejestracja użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik username nie istnieje
Procedura	Wywołanie zapytania HTTP POST /auth/signup body - nazwa: username, hasło: password
Warunki końcowe	Status OK

Tabela 2.21: AuthController - Rejestracja użytkownika

Nazwa	AuthController - Rejestracja użytkownika nazwa istnieje
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik username istnieje
Procedura	Wywołanie zapytania HTTP POST /auth/signup body - nazwa: username, hasło: password
Warunki końcowe	Status BAD REQUEST

Tabela 2.22: AuthController - Rejestracja użytkownika

Nazwa	ShoppingListController - Pobranie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP GET /shopping-list/shopping-list
Warunki końcowe	Status OK Wywołana jest na serwisie metoda pobrania list Zwrócona lista, która zawiera listę zakupów o id 1

Tabela 2.23: ShoppingListController - Pobranie list zakupów użytkownika

Nazwa	ShoppingListController - Dodanie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Wywołanie zapytania HTTP POST /shopping-list/shopping-list body - nazwa: title
Warunki końcowe	Status OK Wywołana jest na serwisie metoda dodawania listy z podanymi wartościami

Tabela 2.24: ShoppingListController - Dodanie list zakupów użytkownika

Nazwa	ShoppingListController - Edycja listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP PUT /shopping-list/shopping-list/1 body - nazwa: title
Warunki końcowe	Status OK Wywołana jest na serwisie metoda edycji listy z podanymi wartościami

Tabela 2.25: ShoppingListController - Edycja listy zakupów użytkownika

Nazwa	ShoppingListController - Usunięcie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP DELETE /shopping-list/shopping-list/1
Warunki końcowe	Status OK, Wywołana jest na serwisie metoda usunięcia listy

Tabela 2.26: ShoppingListController - Usunięcie listy zakupów użytkownika

Nazwa	ShoppingItemController - Pobranie produktów z list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username Produkt - id: 1, lista: 1
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP GET /shopping-item/shopping-item/1
Warunki końcowe	Status OK Wywołana jest na serwisie metoda pobrania produktów z listy 1 Zwrócona lista, która zawiera produkt o id 1

Tabela 2.27: ShoppingItemController - Pobranie produktów z list zakupów użytkownika

Nazwa	ShoppingItemController - Dodanie produktu do listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP POST /shopping-item/shopping-item body - image: 1 bajt, info: nazwa: name, lista: 1
Warunki końcowe	Status OK Wywołana jest na serwisie metoda dodania produktu z podanymi wartościami

Tabela 2.28: ShoppingItemController - Dodanie produktu do listy zakupów użytkownika

Nazwa	Edycja produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username Produkt - id: 1, lista: 1
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP PUT /shopping-item/shopping-item/1 body - image: 1 bajt, info: nazwa: name
Warunki końcowe	Status OK Wywołana jest na serwisie metoda edycji produktu z podanymi wartościami

Tabela 2.29: ShoppingItemController - Edycja produktu z listy zakupów użytkownika

Nazwa	ShoppingItemController - Usunięcie produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP DELETE /shopping-list/shopping-list/1
Warunki końcowe	Status OK, Wywołana jest na serwisie metoda usunięcia produktu

Tabela 2.30: ShoppingItemController - Usunięcie produktu z listy zakupów użytkownika

2.2. Testy warstwy klienta

Do testów użyte zostały Jasmine oraz Karma. Poniżej zaprezentowane zostaną przypadki testowe przeprowadzone w ramach testów jednostkowych warstwy klienta.

2.2.1. Procedura uruchamiania testów

Wymagania wstępne:

- Pobrać klienta aplikacji shopping-list-gui.
- Node.js oraz npm package manager - do uruchomienia części klienta
- Pobrać zależności projektu npm install

Komenda do uruchomienia testów: ng test

2.2.2. Testy serwisów

Nazwa	TokenStorageService - Zapis tokena w sessionStorage
Dane testowe	Token - nazwa: sometoken
Warunki początkowe	Brak
Procedura	Zapis tokena w sessionStorage
Warunki końcowe	W sessionStorage znajduje się element auth-token z wartością sometoken

Tabela 2.31: TokenStorageService - Zapis tokena w sessionStorage

Nazwa	TokenStorageService - Pobranie tokena z sessionStorage
Dane testowe	Element w session storage: nazwa: auth-token, wartość: anothertoken
Warunki początkowe	W sessionStorage jest zapisany token
Procedura	Pobranie tokena z sessionStorage
Warunki końcowe	Pobrany token ma wartość anothertoken

Tabela 2.32: TokenStorageService - Pobranie tokena z sessionStorage

Nazwa	TokenStorageService - Zapis nazwy użytkownika w sessionStorage
Dane testowe	Użytkownik - nazwa: username
Warunki początkowe	Brak
Procedura	Zapis użytkownika w sessionStorage
Warunki końcowe	W sessionStorage znajduje się element auth-user z wartością username: username

Tabela 2.33: TokenStorageService - Zapis nazwy użytkownika w sessionStorage

Nazwa	TokenStorageService - Pobranie nazwy użytkownika z sessionStorage
Dane testowe	Element w session storage: nazwa: auth-user, wartość: username: username
Warunki początkowe	W sessionStorage jest zapisany użytkownik
Procedura	Pobranie użytkownika z sessionStorage
Warunki końcowe	Pobrany użytkownika ma wartość username: username

Tabela 2.34: TokenStorageService - Pobranie nazwy użytkownika z sessionStorage

Nazwa	TokenStorageService - Wylogowanie czyści sessionStorage
Dane testowe	Element w session storage: nazwa: auth-user, wartość: username: username; nazwa: auth-token, wartość: anothertoken
Warunki początkowe	W sessionStorage jest zapisany token i użytkownik
Procedura	Wylogowanie użytkownika
Warunki końcowe	SessionStorage jest pusty

Tabela 2.35: TokenStorageService - Wylogowanie czyści sessionStorage

Nazwa	ShoppingListService - Pobranie list zakupów użytkownika
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Wywołanie metody pobierającej listy zakupów
Warunki końcowe	Wywołanie zapytania HTTP GET http://localhost:8080/shopping-list/shopping-list

Tabela 2.36: ShoppingListService - Pobranie list zakupów użytkownika

Nazwa	ShoppingListService - Dodanie list zakupów użytkownika
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Wywołanie metody dodającej listę zakupów data: bieżąca, status: oczekująca
Warunki końcowe	Wywołanie zapytania HTTP POST http://localhost:8080/shopping-list/shopping-list

Tabela 2.37: ShoppingListService - Dodanie list zakupów użytkownika

Nazwa	ShoppingListService - Edycja listy zakupów użytkownika
Dane testowe	Lista zakupów - id: 1, data: bieżąca, status: oczekująca
Warunki początkowe	Brak
Procedura	Wywołanie metody edytującej listę zakupów 1
Warunki końcowe	Wywołanie zapytania HTTP PUT http://localhost:8080/shopping-list/shopping-list/1

Tabela 2.38: ShoppingListService - Edycja listy zakupów użytkownika

Nazwa	ShoppingListService - Usunięcie listy zakupów użytkownika
Dane testowe	Lista zakupów - id: 1, data: bieżąca, status: oczekująca
Warunki początkowe	Brak
Procedura	Wywołanie metody usuwającej listę zakupów 1
Warunki końcowe	Wywołanie zapytania HTTP DELETE http://localhost:8080/shopping-list/shopping-list/1

Tabela 2.39: ShoppingListService - Usunięcie listy zakupów użytkownika

Nazwa	ShoppingItemService - Pobranie produktu z list zakupów użytkownika
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Wywołanie metody pobierającej produkty dla listy zakupów 1
Warunki końcowe	Wywołanie zapytania HTTP GET http://localhost:8080/shopping-item/shopping-item/1

Tabela 2.40: ShoppingItemService - Pobranie produktu z list zakupów użytkownika

Nazwa	ShoppingItemService - Dodanie produktu do list zakupów użytkownika
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Wywołanie metody dodającej produkt do listy zakupów 1 ilość: 0, jednostka: sztuki, status: oczekująca
Warunki końcowe	Wywołanie zapytania HTTP POST http://localhost:8080/shopping-item/shopping-item

Tabela 2.41: ShoppingItemService - Dodanie produktu do list zakupów użytkownika

Nazwa	ShoppingItemService - Edycja produktu z listy zakupów użytkownika
Dane testowe	Produkt - id: 1, ilość: 0, jednostka: sztuki, status: oczekująca
Warunki początkowe	Brak
Procedura	Wywołanie metody edytującej produkt 1
Warunki końcowe	Wywołanie zapytania HTTP PUT http://localhost:8080/shopping-item/shopping-item/1

Tabela 2.42: ShoppingItemService - Edycja produktu z listy zakupów użytkownika

Nazwa	ShoppingItemService - Usunięcie produktu z listy zakupów użytkownika
Dane testowe	Produkt - id: 1, ilość: 0, jednostka: sztuki, status: oczekująca
Warunki początkowe	Brak
Procedura	Wywołanie metody usuwającej produkt 1
Warunki końcowe	Wywołanie zapytania HTTP DELETE http://localhost:8080/shopping-item/shopping-item/1

Tabela 2.43: ShoppingItemService - Usunięcie produktu z listy zakupów użytkownika

2.2.3. Testy komponentów

Nazwa	LoginComponent - pobranie tokenu
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej token z TokenStorageService

Tabela 2.44: LoginComponent - pobranie tokenu

Nazwa	LoginComponent - pobranie użytkownika
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej użytkownika z TokenStorageService

Tabela 2.45: LoginComponent - pobranie użytkownika

Nazwa	LoginComponent - zalogowanie użytkownika powodzenie
Dane testowe	nazwa: username, hasło: passwordpassword
Warunki początkowe	Podana nazwa i hasło użytkownika
Procedura	Wywołanie logowania użytkownika
Warunki końcowe	Wywołanie metody logowania na AuthService

Tabela 2.46: LoginComponent - zalogowanie użytkownika powodzenie

Nazwa	LoginComponent - zalogowanie użytkownika niepowodzenie
Dane testowe	nazwa: username, hasło: passwordpassword AuthService zwraca błędne dane logowania
Warunki początkowe	Podana nazwa i hasło użytkownika
Procedura	Wywołanie logowania użytkownika
Warunki końcowe	Pola nazwa i hasło użytkownika wyświetlają się jako błędne

Tabela 2.47: LoginComponent - zalogowanie użytkownika niepowodzenie

Nazwa	RegisterComponent - pobranie tokenu
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej użytkownika z TokenStorageService

Tabela 2.48: RegisterComponent - pobranie tokenu

Nazwa	RegisterComponent - rejestracja użytkownika powodzenie
Dane testowe	nazwa: username, hasło: passwordpassword
Warunki początkowe	Podane obowiązkowe dane formularza rejestracji
Procedura	Wywołanie rejestracji użytkownika
Warunki końcowe	Wywołanie metody rejestracji na AuthService

Tabela 2.49: RegisterComponent - rejestracja użytkownika powodzenie

Nazwa	RegisterComponent - rejestracja użytkownika zajęta nazwa
Dane testowe	nazwa: username, hasło: passwordpassword AuthService zwraca błąd zajęta nazwa
Warunki początkowe	Podane obowiązkowe dane formularza rejestracji
Procedura	Wywołanie logowania użytkownika
Warunki końcowe	Pole nazwa użytkownika wyświetla się jako błędne

Tabela 2.50: RegisterComponent - rejestracja użytkownika zajęta nazwa

Nazwa	ShoppingListComponent - pobranie tokenu
Dane testowe	Brak
Warunki początkowe	Brak
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej token z TokenStorageService

Tabela 2.51: ShoppingListComponent - pobranie tokenu

Nazwa	ShoppingListComponent - pobranie list zakupów
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej listy zakupów z ShoppingListService

Tabela 2.52: ShoppingListComponent - pobranie list zakupów

Nazwa	ShoppingListComponent - dodanie formatki dla nowej listy
Dane testowe	Brak
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Użytkownik wybiera przycisk dodania nowej listy
Warunki końcowe	Wywołanie metody dodającej formatkę dla nowej listy

Tabela 2.53: ShoppingListComponent - dodanie formatki dla nowej listy

Nazwa	ShoppingListComponent - zmiana widoku listy na formatkę do edycji
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik wybiera przycisk edycji istniejącej listy
Warunki końcowe	Wywołanie metody zmieniającej widok listy na formularz edycji

Tabela 2.54: ShoppingListComponent - zmiana widoku listy na formatkę do edycji

Nazwa	ShoppingListComponent - usunięcie listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik wybiera przycisk usunięcia istniejącej listy
Warunki końcowe	Wywołanie metody usuwającej listę Usunięcie listy z komponentu

Tabela 2.55: ShoppingListComponent - usunięcie listy

Nazwa	ShoppingListComponent - poniechanie dodania listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka dodawania nowej listy
Procedura	Użytkownik wybiera przycisk poniechania wprowadzonych zmian
Warunki końcowe	Wywołanie metody kasującej listę Usunięcie formatki dodania listy z komponentu

Tabela 2.56: ShoppingListComponent - poniechanie dodania listy

Nazwa	ShoppingListComponent - zapisanie nowej listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wypełniona formatka dodawania nowej listy
Procedura	Użytkownik wybiera przycisk zapisania wprowadzonych zmian
Warunki końcowe	Wywołanie metody zapisującej listę Usunięcie formatki dodania listy z komponentu Dodanie nowej listy do widoku

Tabela 2.57: ShoppingListComponent - zapisanie nowej listy

Nazwa	ShoppingListComponent - poniechanie edycji listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji listy
Procedura	Użytkownik wybiera przycisk poniechania wprowadzonych zmian
Warunki końcowe	Wywołanie metody kasującej zmiany Usunięcie formatki edycji listy z komponentu

Tabela 2.58: ShoppingListComponent - poniechanie edycji listy

Nazwa	ShoppingListComponent - zatwierdzenie edycji listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji listy
Procedura	Użytkownik zmienia nazwę na Zakupy 2 wybiera przycisk zapisania zmian
Warunki końcowe	Wywołanie metody aktualizującej listę Zmiana nazwy listy na Zakupy 2

Tabela 2.59: ShoppingListComponent - zatwierdzenie edycji listy

Nazwa	ShoppingListComponent - zatwierdzenie wykonania listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik wybiera przycisk zatwierdzenia listy
Warunki końcowe	Wywołanie metody aktualizującej listę Zmiana statusu listy na wykonaną

Tabela 2.60: ShoppingListComponent - zatwierdzenie wykonania listy

Nazwa	ShoppingListComponent - przywrócenie listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: wykonana
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik odznacza przycisk zatwierdzenia listy
Warunki końcowe	Wywołanie metody aktualizującej listę Zmiana statusu listy na oczekującą

Tabela 2.61: ShoppingListComponent - przywrócenie listy

Nazwa	ShoppingListComponent - pobranie produktów z listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Użytkownik wybiera przycisk rozwinięcia produktów listy zakupów
Warunki końcowe	Wywołanie metody pobierającej produkty listy z ShoppingItemService

Tabela 2.62: ShoppingListComponent - pobranie produktów z listy

Nazwa	ShoppingListComponent - zmiana widoku produktu na formatkę do edycji
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Użytkownik wybiera przycisk edycji istniejącego produktu
Warunki końcowe	Wywołanie metody zmieniającej widok produktu na formularz edycji

Tabela 2.63: ShoppingListComponent - zmiana widoku produktu na formatkę do edycji

Nazwa	ShoppingListComponent - usunięcie produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Użytkownik wybiera przycisk usunięcia istniejącego produktu
Warunki końcowe	Wywołanie metody usuwającej produkt Usunięcie produktu z komponentu

Tabela 2.64: ShoppingListComponent - usunięcie produktu

Nazwa	ShoppingListComponent - poniechanie dodania produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka dodawania nowego produktu
Procedura	Użytkownik wybiera przycisk poniechania wprowadzonych zmian
Warunki końcowe	Wywołanie metody kasującej produkt Usunięcie formatki dodania produktu z komponentu

Tabela 2.65: ShoppingListComponent - poniechanie dodania produktu

Nazwa	ShoppingListComponent - dodanie formatki dla nowego produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik wybiera przycisk dodania nowego produktu do listy
Warunki końcowe	Wywołanie metody dodającej formatkę dla nowego produktu

Tabela 2.66: ShoppingListComponent - dodanie formatki dla nowego produktu

Nazwa	ShoppingListComponent - zapisanie nowego produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wypełniona formatka dodawania nowego produktu
Procedura	Użytkownik wybiera przycisk zapisania wprowadzonych zmian
Warunki końcowe	Wywołanie metody zapisującej produkt Usunięcie formatki dodania produktu z komponentu Dodanie nowego produktu do widoku

Tabela 2.67: ShoppingListComponent - zapisanie nowego produktu

Nazwa	ShoppingListComponent - poniechanie edycji produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji produktu
Procedura	Użytkownik wybiera przycisk poniechania wprowadzonych zmian
Warunki końcowe	Wywołanie metody kasującej zmiany Usunięcie formatki edycji produktu z komponentu

Tabela 2.68: ShoppingListComponent - poniechanie edycji produktu

Nazwa	ShoppingListComponent - zatwierdzenie edycji produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji produktu
Procedura	Użytkownik zmienia nazwę na Produkt 2, wybiera przycisk zapisania zmian
Warunki końcowe	Wywołanie metody aktualizującej produkt Zmiana nazwy produktu na Produkt 2

Tabela 2.69: ShoppingListComponent - zatwierdzenie edycji produktu

Nazwa	ShoppingListComponent - zatwierdzenie wykonania produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Użytkownik wybiera przycisk zatwierdzenia produktu
Warunki końcowe	Wywołanie metody aktualizującej produkt Zmiana statusu produktu na wykonany

Tabela 2.70: ShoppingListComponent - zatwierdzenie wykonania produktu

Nazwa	ShoppingListComponent - przywrócenie produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: wykonana Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: wykonany
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Użytkownik odznacza przycisk zatwierdzenia produktu
Warunki końcowe	Wywołanie metody aktualizującej produkt Zmiana statusu produktu na oczekujący

Tabela 2.71: ShoppingListComponent - przywrócenie produktu

3. Testy integracyjne

Testy integracyjne przeprowadzone zostały w warstwie serwera oraz klienta. Testowana była integracja pomiędzy kontrolerami i serwisami oraz serwisami i bazą danych na serwerze oraz integracja komponentów z serwisami klienta. Wszystkie komponenty poza parą, której integracja była testowana, były mockowane. Niektóre testy pominięto ze względu na ich znaczną ilość. Testy integracyjne uruchamiane są razem z testami jednostkowymi (procedura uruchamiania opisana wcześniej).

3.1. Testy warstwy serwera

Do testów użyte zostały JUnit oraz Mockito. Poniżej zaprezentowane zostaną przypadki testowe przeprowadzone w ramach testów integracyjnych warstwy serwera.

3.1.1. Testy integracji kontrolerów z serwisami

Nazwa	ShoppingListController + ShoppingListService Pobranie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, użytkownika: username
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP GET /shopping-list/shopping-list
Warunki końcowe	Status OK Zwrócona lista, która zawiera listę zakupów o id 1

Tabela 3.1: ShoppingListController + ShoppingListService - Pobranie list zakupów użytkownika

Nazwa	ShoppingListController + ShoppingListService Dodanie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Wywołanie zapytania HTTP POST /shopping-list/shopping-list body - nazwa: title
Warunki końcowe	Status OK W repozytorium znajduje się lista użytkownika

Tabela 3.2: ShoppingListController + ShoppingListService - Dodanie list zakupów użytkownika

Nazwa	ShoppingListController + ShoppingListService Edycja listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP PUT /shopping-list/shopping-list/1 body - nazwa: newTitle, status: oczekująca
Warunki końcowe	Status OK Nazwa i status listy zostały odpowiednio zmienione

Tabela 3.3: ShoppingListController + ShoppingListService - Edycja list zakupów użytkownika

Nazwa	ShoppingListController + ShoppingListService Usunięcie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP DELETE /shopping-list/shopping-list/1
Warunki końcowe	Status OK, Status listy został zmieniony na usunięta

Tabela 3.4: ShoppingListController + ShoppingListService - Usunięcie listy zakupów użytkownika

Nazwa	ShoppingItemController + ShoppingItemService Pobranie produktów z list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username Produkt - id: 1, lista: 1
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP GET /shopping-item/shopping-item/1
Warunki końcowe	Status OK Zwrócona lista, która zawiera produkt o id 1

Tabela 3.5: ShoppingItemController + ShoppingItemService - Pobranie produktów z list zakupów użytkownika

Nazwa	ShoppingItemController + ShoppingItemService Dodanie produktu do listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Wywołanie zapytania HTTP POST /shopping-item/shopping-item body - image: 1 bajt, info: nazwa: name, lista: 1
Warunki końcowe	Status OK W repozytorium znajduje się produkt z listy 1 użytkownika

Tabela 3.6: ShoppingItemController + ShoppingItemService - Dodanie produktu do listy zakupów użytkownika

Nazwa	ShoppingItemController + ShoppingItemService Edycja produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, właściciel: username Produkt - id: 1, nazwa: oldName, lista: 1, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP PUT /shopping-item/shopping-item/1 body - image: 1 bajt, info: nazwa: newName, status: wykonany
Warunki końcowe	Status OK Nazwa i status produktu zostały odpowiednio zmienione

Tabela 3.7: ShoppingItemController + ShoppingItemService - Edycja produktu z listy zakupów użytkownika

Nazwa	ShoppingItemController + ShoppingItemService Usunięcie produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - id: 1, nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Wywołanie zapytania HTTP DELETE /shopping-list/shopping-list/1
Warunki końcowe	Status OK, Status produktu został zmieniony na usunięty

Tabela 3.8: ShoppingItemController + ShoppingItemService - Usunięcie produktu z listy zakupów użytkownika

3.1.2. Testy integracji serwisów z bazą danych

Nazwa	ShoppingListService + ShoppingListRepository Pobranie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, użytkownik: username
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Pobranie list zakupów
Warunki końcowe	Zwrócona lista, która zawiera listę zakupów o nazwie lista

Tabela 3.9: ShoppingListService + ShoppingListRepository - Pobranie list zakupów użytkownika

Nazwa	ShoppingListService + ShoppingListRepository Dodanie listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Dodanie listy zakupów nazwa: lista, data: 2023-04-08
Warunki końcowe	W bazie znajduje się lista użytkownika

Tabela 3.10: ShoppingListService + ShoppingListRepository - Dodanie list zakupów użytkownika

Nazwa	ShoppingListService + ShoppingListRepository Edycja listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Edycja listy zakupów nazwa: newTitle, data: 2023-04-10
Warunki końcowe	W bazie zapisane są zmienione wartości listy

Tabela 3.11: ShoppingListService + ShoppingListRepository - Edycja listy zakupów użytkownika

Nazwa	ShoppingListService + ShoppingListRepository Usunięcie list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: title, właściciel: username, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Usunięcie listy o nazwie title po znalezionym id
Warunki końcowe	W bazie zapisany jest status listy usunięta

Tabela 3.12: ShoppingListService + ShoppingListRepository - Usunięcie listy zakupów użytkownika

Nazwa	ShoppingItemService + ShoppingItemRepository Pobranie produktów z list zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, status: oczekująca Produkt - nazwa: name, jednostka: sztuki, ilość: 1, lista: lista, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Pobranie produktów z listy o nazwie lista (po znalezionym id)
Warunki końcowe	Zwrócona lista, która zawiera produkt o nazwie nazwa

Tabela 3.13: ShoppingItemService + ShoppingItemRepository - Pobranie produktów z list zakupów użytkownika

Nazwa	ShoppingItemService + ShoppingItemRepository Dodanie produktu do listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Dodanie produktu nazwa: name do listy o nazwie lista
Warunki końcowe	W bazie znajduje się produkt w liście użytkownika

Tabela 3.14: ShoppingItemService + ShoppingItemRepository - Dodanie produktu do listy zakupów użytkownika

Nazwa	ShoppingItemService + ShoppingItemRepository Edycja produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, status: oczekująca Produkt - nazwa: name, jednostka: sztuki, ilość: 1, lista: lista, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Edycja produktu nazwa: newName, jednostka: litry, ilość: 2
Warunki końcowe	W bazie zapisane są zmienione wartości produktu

Tabela 3.15: ShoppingItemService + ShoppingItemRepository - Edycja produktu z listy zakupów użytkownika

Nazwa	ShoppingItemService + ShoppingItemRepository Usunięcie produktu z listy zakupów użytkownika
Dane testowe	Użytkownik - nazwa: username, hasło: password Lista zakupów - nazwa: lista, data: 2023-04-08, status: oczekująca Produkt - nazwa: name, jednostka: sztuki, ilość: 1, lista: lista, status: oczekujący
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Usunięcie produktu o nazwie name po znalezionym id
Warunki końcowe	W bazie zapisany jest status produktu usunięty

Tabela 3.16: ShoppingItemService + ShoppingItemRepository - Usunięcie produktu z listy zakupów użytkownika

3.2. Testy warstwy klienta

Do testów użyte zostały Jasmine oraz Karma. Poniżej zaprezentowane zostaną przypadki testowe przeprowadzone w ramach testów integracyjnych warstwy klienta.

3.2.1. Testy integracji komponentów z serwisami

Nazwa	LoginComponent + AuthService Zalogowanie użytkownika
Dane testowe	nazwa: username, hasło: passwordpassword
Warunki początkowe	Podana nazwa i hasło użytkownika
Procedura	Wywołanie logowania użytkownika
Warunki końcowe	Wywołanie metody logowania na AuthService

Tabela 3.17: LoginComponent + AuthService - zalogowanie użytkownika powodzenie

Nazwa	RegisterComponent + AuthService Rejestracja użytkownika
Warunki początkowe	Podane obowiązkowe dane formularza rejestracji
Procedura	Wywołanie rejestracji użytkownika
Warunki końcowe	Wywołanie metody rejestracji na AuthService

Tabela 3.18: RegisterComponent + AuthService - rejestracja użytkownika powodzenie

Nazwa	ShoppingListComponent + ShoppingListService Pobranie list zakupów
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany
Procedura	Utworzenie komponentu
Warunki końcowe	Wywołanie metody pobierającej listy zakupów z ShoppingListService

Tabela 3.19: ShoppingListComponent + ShoppingListService - pobranie list zakupów

Nazwa	ShoppingListComponent + ShoppingListService Usunięcie listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów
Procedura	Użytkownik wybiera przycisk usunięcia istniejącej listy
Warunki końcowe	Wywołanie metody usuwającej listę z ShoppingListService

Tabela 3.20: ShoppingListComponent + ShoppingListService - usunięcie listy

Nazwa	ShoppingListComponent + ShoppingListService Zapisanie nowej listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wypełniona formatka dodawania nowej listy
Procedura	Użytkownik wybiera przycisk zapisania wprowadzonych zmian
Warunki końcowe	Wywołanie metody zapisującej listę z ShoppingListService

Tabela 3.21: ShoppingListComponent + ShoppingListService - zapisanie nowej listy

Nazwa	ShoppingListComponent + ShoppingListService Zatwierdzenie edycji listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji listy
Procedura	Użytkownik wybiera przycisk zapisania zmian
Warunki końcowe	Wywołanie metody aktualizującej listę z ShoppingListService

Tabela 3.22: ShoppingListComponent + ShoppingListService - zatwierdzenie edycji listy

Nazwa	ShoppingListComponent + ShoppingItemService Pobranie produktów z listy
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany i ma listę zakupów
Procedura	Użytkownik wybiera przycisk rozwinięcia produktów listy zakupów
Warunki końcowe	Wywołanie metody pobierającej produkty listy z ShoppingItemService

Tabela 3.23: ShoppingListComponent + ShoppingItemService - pobranie produktów z listy

Nazwa	ShoppingListComponent + ShoppingItemService Usunięcie produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt na liście
Procedura	Użytkownik wybiera przycisk usunięcia istniejącego produktu
Warunki końcowe	Wywołanie metody usuwającej produkt z ShoppingItemService

Tabela 3.24: ShoppingListComponent + ShoppingItemService - usunięcie produktu

Nazwa	ShoppingListComponent + ShoppingItemService Zapisanie nowego produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wypełniona formatka dodawania nowego produktu
Procedura	Użytkownik wybiera przycisk zapisania wprowadzonych zmian
Warunki końcowe	Wywołanie metody zapisującej produkt z ShoppingItemService

Tabela 3.25: ShoppingListComponent + ShoppingItemService - zapisanie nowego produktu

Nazwa	ShoppingListComponent + ShoppingItemService Zatwierdzenie edycji produktu
Dane testowe	Lista - id : 1, nazwa : Zakupy 1, data : bieżąca, status: oczekująca Produkt - id : 1, nazwa : Produkt 1, ilość: 2, jednostka: sztuki, status: oczekująca
Warunki początkowe	Użytkownik jest zalogowany, wyświetlona formatka edycji produktu
Procedura	Użytkownik wybiera przycisk zapisania zmian
Warunki końcowe	Wywołanie metody aktualizującej produkt z ShoppingItemService

Tabela 3.26: ShoppingListComponent + ShoppingItemService - zatwierdzenie edycji produktu

4. Testy systemowe

Testy systemowe przeprowadzone zostały przy użyciu narzędzie Cypress. Testy te napisane i uruchamiane są po stronie warstwy klienta, ale testują działanie całej aplikacji (również warstwę serwera).

Testy uruchamiane są komendą: `ng e2e`. Konieczne jest uruchomienie warstwy serwera do poprawnego działania.

Testy wykonywane są kolejno po sobie, ponieważ są od siebie zależne. Zastosowano takie podejście, ponieważ aby np. edytować czy usunąć listę trzeba ją najpierw utworzyć. Również logowanie konieczne byłoby przed zrobieniem wszystkich innych operacji. Aby nie powielać testów są one wykonywane następująco:

TS1: Logowanie użytkownika

TS2: Dodanie listy zakupów

TS3: Edycja listy zakupów

TS4: Zaznaczanie listy zakupów jako wykonanej

TS5: Odznaczanie listy zakupów jako niewykonanej

TS6: Dodanie produktu do listy

TS7: Edycja produktu

TS8: Zaznaczanie produktu jako wykonanego

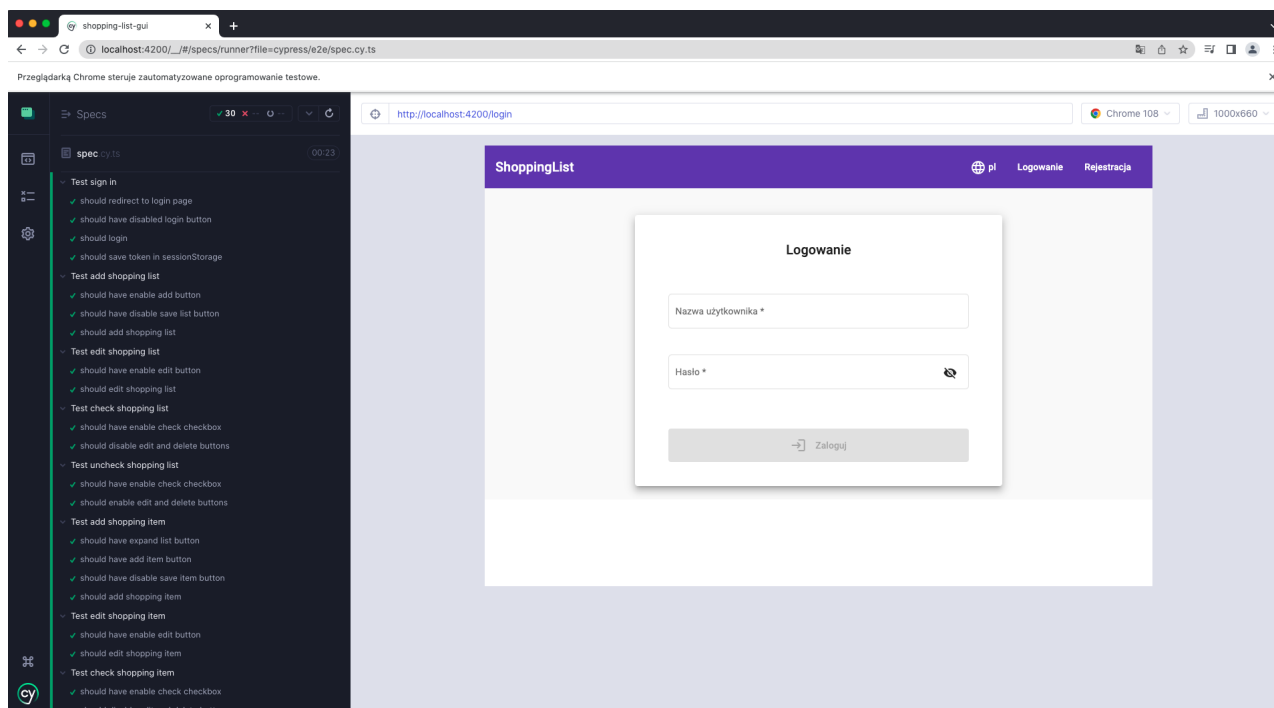
TS9: Odznaczanie produktu jako niewykonanego

TS10: Usunięcie produktu

TS11: Usunięcie listy

TS12: Wylogowanie

Jak można zauważyć na rysunku 4.1 testowane są dostępności przycisków, istnienie elementów (przy dodawaniu, usuwaniu), zmiany wartości elementów (przy edycji), przekierowania (przy logowaniu i wylogowaniu).



Rysunek 4.1: Testy systemowe

Nazwa	TS1. Logowanie	
Opis	Ten test sprawdza możliwość zalogowania się w aplikacji	
Dane testowe	Użytkownik: jankowal, hasło: aaaaaaaaaa	
Warunki początkowe	Użytkownik nie jest zalogowany	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona logowania zostaje wyświetlona	Strona logowania wyświetliła się
2. Sprawdź czy przycisk logowania zablokowany	Przycisk logowania zablokowany	Przycisk logowania był zablokowany
3. Wypełnij pola: - nazwa użytkownika - hasło	Pola: - nazwa użytkownika - hasło zostają wypełnione	Pola: - nazwa użytkownika - hasło zostały wypełnione
4. Sprawdź czy przycisk logowania dostępny	Przycisk logowania dostępny	Przycisk logowania był dostępny
5. Kliknij przycisk logowania	Użytkownik zostaje zalogowany	Użytkownik został zalogowany

Tabela 4.1: Logowanie

Nazwa	TS2. Dodanie listy zakupów	
Opis	Ten test sprawdza możliwość dodania listy zakupów	
Dane testowe	Lista: nazwa: Shopping List 1, data: 02/01/2023	
Warunki początkowe	Użytkownik jest zalogowany	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk dodania listy dostępny	Przycisk dodania listy dostępny	Przycisk dodania listy był dostępny
3. Kliknij przycisk dodania listy	Pojawia się formatka dodania listy	Pojawiła się formatka dodania listy
4. Sprawdź czy przycisk zapisu listy zablokowany	Przycisk zapisu listy zablokowany	Przycisk zapisu listy był zablokowany
5. Wypełnij pola: - tytuł - data wykonania	Pola: - tytuł - data wykonania zostają wypełnione	Pola: - tytuł - data wykonania zostały wypełnione
6. Sprawdź czy przycisk zapisu listy dostępny	Przycisk zapisu listy dostępny	Przycisk zapisu listy był dostępny
7. Kliknij przycisk zapisu listy	Lista zostaje dodana	Lista została dodana

Tabela 4.2: Dodanie listy zakupów

Nazwa	TS3. Edycja listy zakupów	
Opis	Ten test sprawdza możliwość edytowania listy zakupów	
Dane testowe	Lista - nazwa: Shopping List 1, data: 02/01/2023 Edycja listy - nazwa: Shopping List edited	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk edycji listy dostępny	Przycisk edycji listy dostępny	Przycisk edycji listy był dostępny
3. Kliknij przycisk edycji listy	Pojawia się formatka edycji listy	Pojawiła się formatka edycji listy
4. Zmień wartości w polach: - tytuł	Pole: - tytuł zostaje zmienione	Pole: - tytuł zostało zmienione
5. Sprawdź czy przycisk zapisu zmian dostępny	Przycisk zapisu zmian dostępny	Przycisk zapisu zmian był dostępny
6. Kliknij przycisk zapisu zmian	Lista zostaje zmieniona	Lista została zmieniona

Tabela 4.3: Edycja listy zakupów

Nazwa	TS4. Zaznaczanie listy zakupów jako wykonanej	
Opis	Ten test sprawdza możliwość zaznaczenia listy zakupów jako wykonanej	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów do wykonania	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk zatwierdzenia listy dostępny	Przycisk zatwierdzenia listy dostępny	Przycisk zatwierdzenia listy był dostępny
3. Kliknij przycisk zatwierdzenia listy	Lista zostaje przekreślona Przycisk edycji listy zablokowany	Lista została przekreślona Przycisk edycji listy był zablokowany

Tabela 4.4: Zaznaczanie listy zakupów jako wykonanej

Nazwa	TS5. Odznaczanie listy zakupów jako niewykonanej	
Opis	Ten test sprawdza możliwość odznaczenia listy zakupów jako niewykonanej	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów wykonaną	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk odznaczenia listy dostępny	Przycisk odznaczenia listy dostępny	Przycisk odznaczenia listy był dostępny
3. Kliknij przycisk odznaczenia listy	Lista nie jest przekreślona Przycisk edycji listy dostępny	Lista przestała być przekreślona Przycisk edycji listy był dostępny

Tabela 4.5: Odznaczanie listy zakupów jako niewykonanej

Nazwa	TS6. Dodanie produktu do listy zakupów	
Opis	Ten test sprawdza możliwość dodania produktu do listy zakupów	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023 Produkt: nazwa: Shopping item 1, ilość: 1	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy był dostępny
3. Kliknij przycisk rozwinięcia produktów	Pojawia się lista produktów	Pojawiła się lista produktów
4. Sprawdź czy przycisk dodania produktu dostępny	Przycisk dodania produktu dostępny	Przycisk dodania produktu był dostępny
5. Kliknij przycisk dodania produktu	Pojawia się formatka dodania produktu	Pojawiła się formatka dodania produktu
6. Sprawdź czy przycisk zapisu produktu zablokowany	Przycisk zapisu produktu zablokowany	Przycisk zapisu produktu był zablokowany
7. Wypełnij pola: - nazwa - ilość	Pola: - nazwa - ilość zostają wypełnione	Pola: - nazwa - ilość zostały wypełnione
8. Sprawdź czy przycisk zapisu produktu dostępny	Przycisk zapisu produktu dostępny	Przycisk zapisu produktu był dostępny
9. Kliknij przycisk zapisu produktu	Produkt zostaje dodany	Produkt został dodany

Tabela 4.6: Dodanie produktu do listy zakupów

Nazwa	TS7. Edycja produktu	
Opis	Ten test sprawdza możliwość edytowania produktu	
Dane testowe	Lista - nazwa: Shopping List 1, data: 02/01/2023 Produkt - nazwa: Shopping item 1, ilość: 1 Edycja produktu - nazwa: Shopping item 1 edited, ilość: 2	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów o produkt	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy był dostępny
3. Kliknij przycisk rozwinięcia produktów	Pojawia się lista produktów	Pojawiła się lista produktów
4. Sprawdź czy przycisk edycji produktu dostępny	Przycisk edycji produktu dostępny	Przycisk edycji produktu był dostępny
5. Kliknij przycisk edycji produktu	Pojawia się formatka edycji produktu	Pojawiła się formatka edycji produktu
6. Zmień wartości w polach: - nazwa - ilość	Pola: - nazwa - ilość zostają zmienione	Pola: - nazwa - ilość zostały zmienione
7. Sprawdź czy przycisk zapisu zmian dostępny	Przycisk zapisu zmian dostępny	Przycisk zapisu zmian był dostępny
8. Kliknij przycisk zapisu zmian	Produkt zostaje zmieniony	Produkt został zmieniony

Tabela 4.7: Edycja produktu

Nazwa	TS8. Zaznaczanie produktu jako wykonanego	
Opis	Ten test sprawdza możliwość zaznaczenia produktu jako wykonanego	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023 Produkt: nazwa: Shopping item 1 edited, ilość: 2	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt do wykonania	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy był dostępny
3. Kliknij przycisk rozwinięcia produktów	Pojawia się lista produktów	Pojawiła się lista produktów
4. Sprawdź czy przycisk zatwierdzenia produktu dostępny	Przycisk zatwierdzenia produktu dostępny	Przycisk zatwierdzenia produktu był dostępny
5. Kliknij przycisk zatwierdzenia produktu	Produkt zostaje przekreślony Przycisk edycji produktu zablokowany	Produkt został przekreślony Przycisk edycji produktu był zablokowany

Tabela 4.8: Zaznaczanie produktu jako wykonanego

Nazwa	TS9. Odznaczanie produktu jako niewykonanego	
Opis	Ten test sprawdza możliwość odznaczenia produktu jako niewykonanego	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023 Produkt: nazwa: Shopping item 1 edited, ilość: 2	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt wykonane	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy był dostępny
3. Kliknij przycisk rozwinięcia produktów	Pojawia się lista produktów	Pojawiła się lista produktów
4. Sprawdź czy przycisk odznaczenia produktu dostępny	Przycisk odznaczenia produktu dostępny	Przycisk odznaczenia produktu był dostępny
5. Kliknij przycisk odznaczenia produktu	Produkt nie jest przekreślony Przycisk edycji produktu dostępny	Produkt przestał być przekreślony Przycisk edycji produktu był dostępny

Tabela 4.9: Odznaczanie produktu jako niewykonanego

Nazwa	TS10. Usunięcie produktu	
Opis	Ten test sprawdza możliwość usunięcia produktu z listy zakupów	
Dane testowe	Lista: nazwa: Shopping List edited, data: 02/01/2023 Produkt: nazwa: Shopping item 1 edited, ilość: 2	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów i produkt	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy dostępny	Przycisk rozwinięcia produktów listy był dostępny
3. Kliknij przycisk rozwinięcia produktów	Pojawia się lista produktów	Pojawiła się lista produktów
4. Sprawdź czy przycisk usunięcia produktu dostępny	Przycisk usunięcia produktu dostępny	Przycisk usunięcia produktu był dostępny
5. Kliknij przycisk usunięcia produktu	Produkt zostaje usunięty	Produkt został usunięty

Tabela 4.10: Usunięcie produktu

Nazwa	TS11. Usunięcie listy	
Opis	Ten test sprawdza możliwość usunięcia listy zakupów	
Dane testowe	Lista - nazwa: Shopping List edited, data: 02/01/2023	
Warunki początkowe	Użytkownik jest zalogowany, ma listę zakupów	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk usunięcia listy dostępny	Przycisk usunięcia listy dostępny	Przycisk usunięcia listy był dostępny
3. Kliknij przycisk usunięcia listy	Lista zostaje usunięta	Lista została usunięta

Tabela 4.11: Usunięcie listy

Nazwa	TS12. Wylogowanie	
Opis	Ten test sprawdza możliwość wylogowania się z aplikacji	
Dane testowe	Brak	
Warunki początkowe	Użytkownik jest zalogowany	
Typ testu	pozytywny	
Opis kroku	Oczekiwany rezultat	Rzeczywisty rezultat
1. Wejdź na stronę główną localhost:4200	Strona główna zostaje wyświetlona	Strona główna wyświetliła się
2. Sprawdź czy przycisk profilu dostępny	Przycisk profilu dostępny	Przycisk profilu był dostępny
3. Kliknij przycisk profilu	Pojawia się przycisk wyloguj	Pojawił się przycisk wyloguj
4. Sprawdź czy przycisk wylogowania dostępny	Przycisk wylogowania dostępny	Przycisk wylogowania był dostępny
5. Kliknij przycisk wylogowania	Użytkownik wylogowany Pojawia się strona logowania	Użytkownik został wylogowany Pojawiła się strona logowania

Tabela 4.12: Wylogowanie

5. Testy bezpieczeństwa

Testy bezpieczeństwa przeprowadzone zostały z wykorzystaniem narzędzie OWASP ZAP. Szczegółowy raport zamieszczono w repozytorium projektu w katalogu security-tests.

Do przeprowadzenia testów konieczne było uruchomienie aplikacji. W narzędziu OWASP ZAP podano URL aplikacji i przeprowadzono testy automatyczne oraz manualne. Podczas testów manualnych wykonano następujące funkcje:

1. Logowanie użytkownika
2. Dodanie listy zakupów
3. Edycja listy zakupów
4. Dodanie produktu do listy
5. Edycja produktu
6. Wyświetlenie obrazka
7. Usunięcie produktu
8. Usunięcie listy
9. Wylogowanie

Wyniki testów automatycznych i manualnych wykazały te same rezultaty, dlatego dołączono tylko jeden raport. Na rysunku 5.1 przedstawiono wykryte zagrożenia aplikacji. Jak widać wykrytych zostało 10 zagrożeń, w tym 3 o stopniu ryzyka średnim, 4 niskim i 3 informacyjne.

W pierwszej kolejności należałoby zająć się naprawą pierwszego z nich: **Content Security Policy (CSP) Header Not Set** (w przedstawionej tabeli tego nie widać, ale w raporcie można znaleźć informację, że ma on wysokie prawdopodobieństwo wystąpienia). Content Security Policy to dodatkowa warstwa zabezpieczeń, która pomaga wykrywać i ograniczać niektóre rodzaje ataków, w tym ataki Cross Site Scripting (XSS) i wstrzykiwanie danych. Należałoby skonfigurować serwer tak, aby ustawić nagłówek Content-Security-Policy.

Później warto przeanalizować dwa kolejne zagrożenia (mają średnie prawdopodobieństwo wystąpienia). **Cross-Domain Misconfiguration** - należałoby upewnić się, że wrażliwe dane nie są dostępne w sposób niewierzytelny. Skonfigurować nagłówek HTTP „Access-Control-Allow-Origin” na bardziej restrykcyjny zestaw domen lub całkowicie usunąć wszystkie nagłówki CORS, aby umożliwić

przeglądarce internetowej egzekwowanie zasad Same Origin Policy (SOP) w bardziej restrykcyjny sposób. **Missing Anti-clickjacking Header** - odpowiedź nie obejmuje ani Content-Security-Policy z dyrektywą „frame-ancestors”, ani X-Frame-Options w celu ochrony przed atakami „ClickJacking”. Należy upewnić się, że jeden z nich jest ustawiony na wszystkich stronach internetowych zwracanych przez aplikację.

Pozostałe zagrożenia mimo, że mają mniejsze ryzyko również należałoby przeanalizować i pozbyć się ich w miarę możliwości. Im jednak wyższe ryzyko i prawdopodobieństwo wystąpienia, tym wyższy priorytet powinniśmy nadawać usunięciu takiego zagrożenia.

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Średni	6 (60,0%)
Cross-Domain Misconfiguration	Średni	30 (300,0%)
Missing Anti-clickjacking Header	Średni	6 (60,0%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Niski	12 (120,0%)
Strict-Transport-Security Header Not Set	Niski	5 (50,0%)
Timestamp Disclosure - Unix	Niski	1 (10,0%)
X-Content-Type-Options Header Missing	Niski	28 (280,0%)
Information Disclosure - Suspicious Comments	Informacyjny	21 (210,0%)
Modern Web Application	Informacyjny	8 (80,0%)
Retrieved from Cache	Informacyjny	10 (100,0%)
Total		10

Rysunek 5.1: Testy bezpieczeństwa

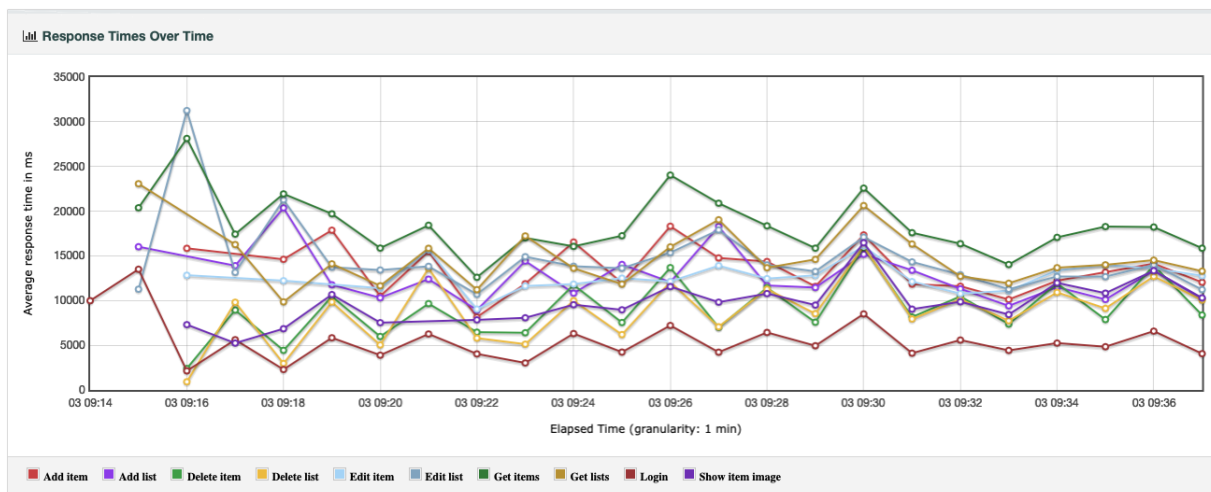
6. Testy wydajnościowe

Testy wydajnościowe przeprowadzone zostały z wykorzystaniem narzędzi JMeter - do przeprowadzania testów oraz pomocniczo z BlazeMeter - do nagrywania testów. Szczegółowy raport zamieszczono w repozytorium projektu w katalogu performance-tests/Wyniki, plik index.html. Podczas testów wykonano następujące funkcje:

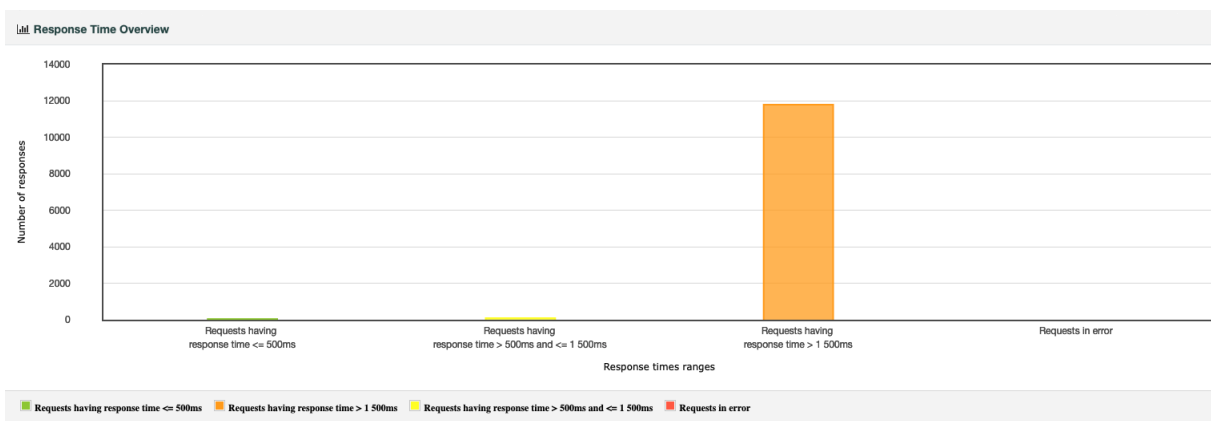
1. Logowanie użytkownika
2. Dodanie listy zakupów
3. Edycja listy zakupów
4. Dodanie produktu do listy
5. Edycja produktu
6. Wyświetlenie obrazka
7. Usunięcie produktu
8. Usunięcie listy

Testy nagrane przy pomocy BlazeMeter załadowano do Jmeter (plik shopping-list-performance-test.jmx). Należało zmienić testy, w których zapytania wykorzystywały multipart/form-data - nagrały się niepoprawnie. Ustawione zostały parametry na 100 użytkowników i 500 iteracji, testy przerwano po około pół godziny. Wyniki testów zapisywane były do pliku wyniki.jtl w katalogu performance-test. Z tak przeprowadzonych testów wygenerowano raport, którego zawartość zapisana została w katalogu performance-test/Wyniki.

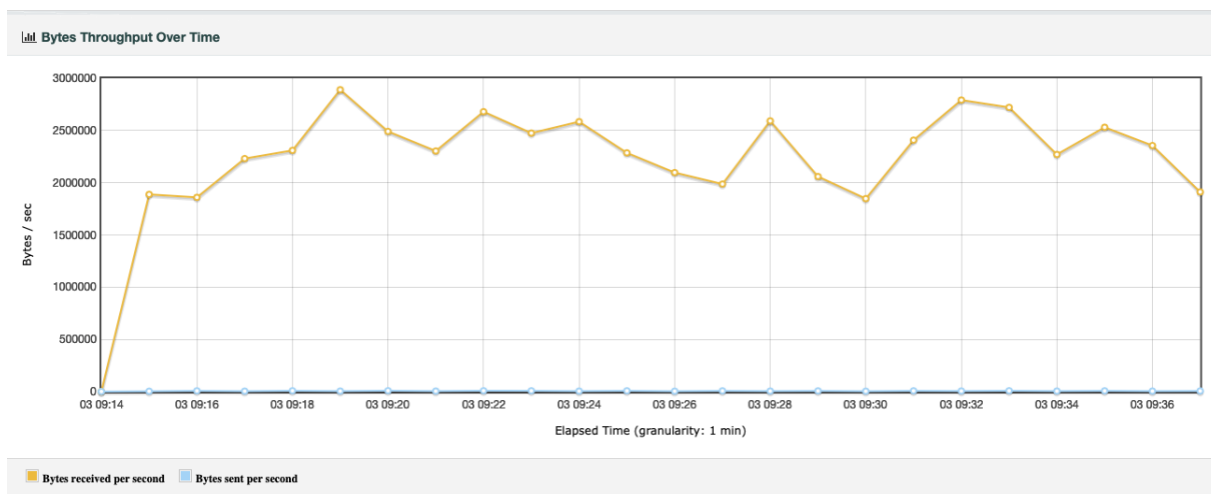
Jak widać na załączonych wykresach (6.1, 6.2, 6.3) działanie aplikacji wydaje się być w miarę stabilne przy zwiększonym obciążeniu. Czas odpowiedzi i przepustowość łączy oscylują w stałych granicach. Czas odpowiedzi jest dosyć długi, jednak testy te zależą od sprzętu na którym są uruchamiane, więc wyniki te nie są wiążące - dopiero po uruchomieniu ich na przygotowanym środowisku produkcyjnym można byłoby stwierdzić, czy wyniki nas zadowolają.



Rysunek 6.1: Czas odpowiedzi



Rysunek 6.2: Przegląd czasów odpowiedzi



Rysunek 6.3: Przepustowość łącza w czasie