Ludwik Madej

X 2025

Spis treści

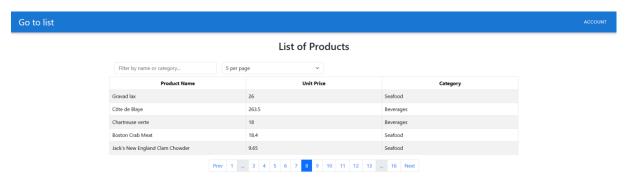
Opis aplikacji	1
Tabele do przeglądania	1
Dodatkowe uprawnienia użytkownika	2
Hierarchia użytkowników	3
Założenia poprawności danych	5
Przepływ danych	5

Opis aplikacji

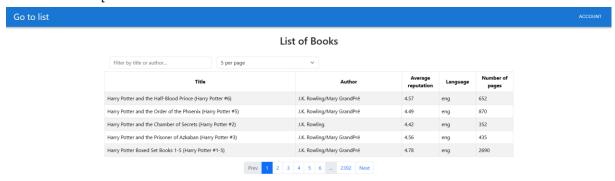
Tabele do przeglądania

Dla zwykłego użytkownika dostępne są dwie tabele:

1. Tabela produktów

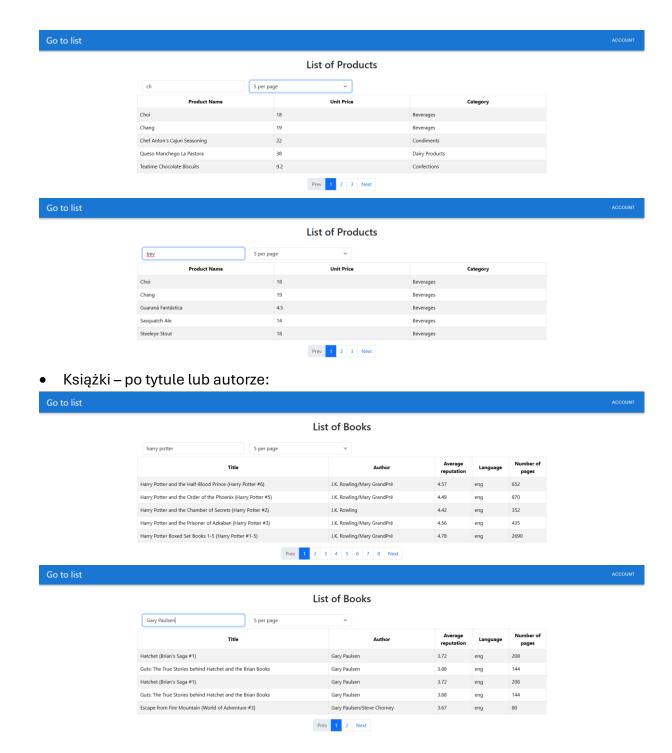


2. Tabela książek



Obie można filtrować w zależności od ich zawartości, tj.:

Produkty – po nazwie lub kategorii

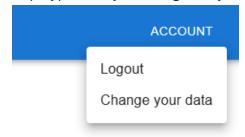


Na obu stronach możliwe jest wybranie ilość pozycji na stronie, które mają się wyświetlać, a po wpisaniu frazy w pole wyszukiwania automatycznie filtrowane są produkty i książki.

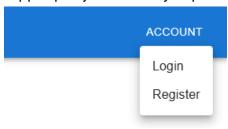
Dodatkowe uprawnienia użytkownika

W zależności od stanu zalogowania, użytkownik może:

W przypadku bycia zalogowanym – zmienić swoje dane lub się wylogować



• Wpp. – przejść do strony odpowiadającej za rejestracje lub logowanie.



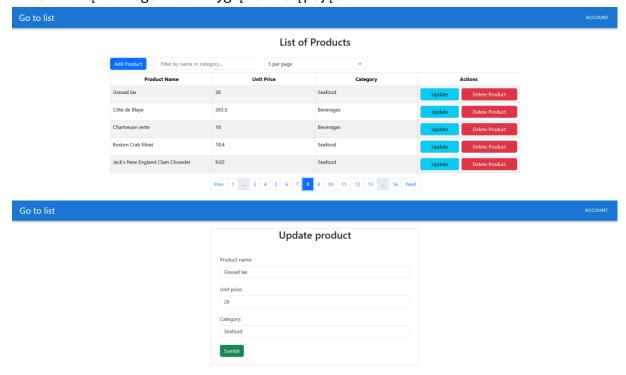
Hierarchia użytkowników

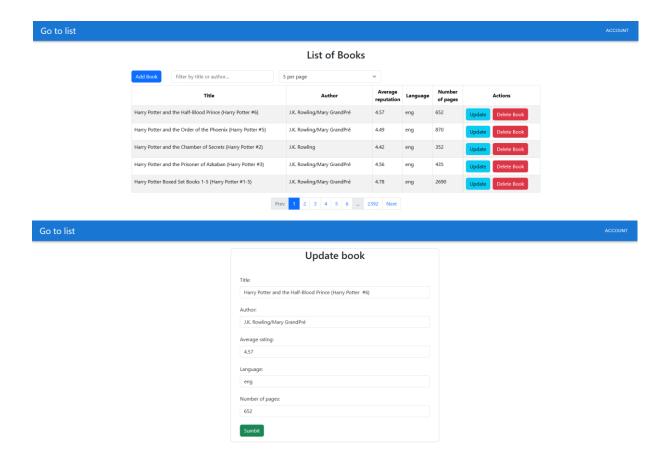
Dostępne są trzy role od których zależą akcje możliwe do wykonania na stronie:

1. **User** – może tylko przeglądać tabele i zmieniać swoje dane (zostawienie pustego pola odpowiadającemu nowemu hasłu i jego potwierdzeniu spowoduje zostawienie tego samego hasła użytkownika)

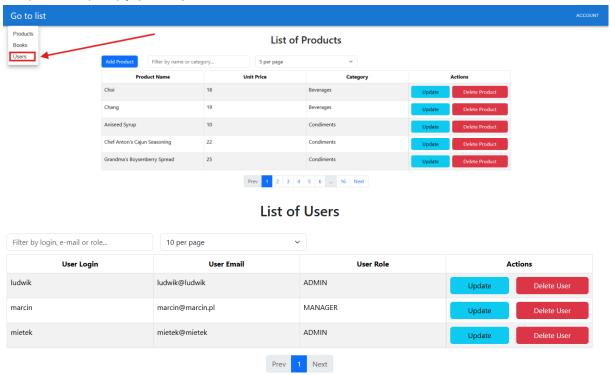


2. **Manager** – zyskuje dodatkowo możliwość dodawania, usuwania i modifkowania produktów oraz książek. Jego widok wygląda następująco:





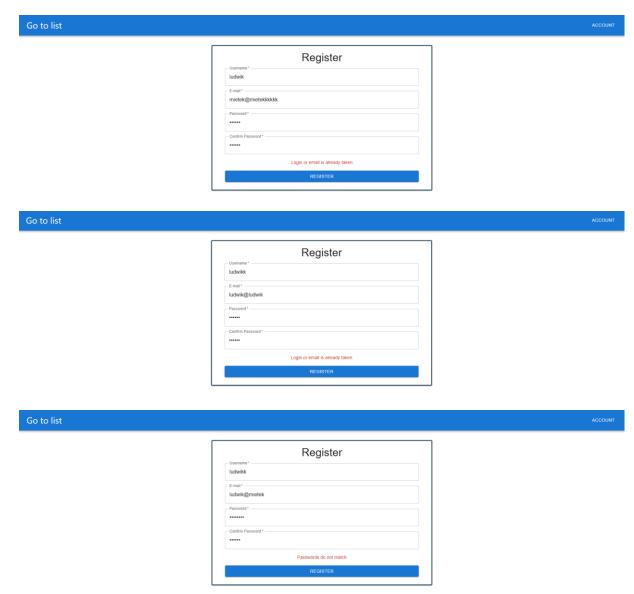
3. **Admin** – dostaje wgląd w listę użytkowników. Może modyfikować ich dane oraz usuwać bez konieczności używania ich hasła. Możliwa jest również filtracja po wszystkich możliwych polach opisujących użytkownika.



Podobnie jak w przypadku zmiany danych przez użytkownika – pozostawienie hasła pustego nie spowoduje jego zmiany.

Założenia poprawności danych

Jedyne obostrzenia dotyczą użytkowników – każdy z nich musi mieć unikalny login oraz e-mail. Dane te są walidowane w każdym z formularzy rejestracji oraz zmiany użytkownika.



Przepływ danych

Za obsługę oraz sprawdzanie poprawności danych odpowiada REST API, które komunikuje się z bazą danych. Użyłem w tym celu Spring Boot, który w znakomity sposób ułatwia funkcjonowanie tego systemu. Nie ma ograniczeń co do bazy danych – można podpiąć dowolną a aplikacja sama utworzy potrzebne tabele z określonymi funkcjami. Backend obsługuje 3 endpointy (nie chciałem zaśmiecać raportu wszystkimi, więc wrzuciłem te, od których wszystko wychodzi):

- http://localhost:8080/api/users/**
- http://localhost:8080/api/books/**
- http://localhost:8080/api/products/**

Za których pomocą dodajemy/modyfikujemy lub usuwamy produkty, książki oraz użytkowników. Poszczególne funkcjonalności dostępne są dla wyznaczonych ról, np. wgląd do książek i produktów mają tylko zalogowani użytkownicy, a modyfikować je mogę tylko Ci z rolami 'MANAGER' lub 'ADMIN'

Pokrycie kodu testami

Kod został pokryty w znacznej części, co potwierdza poniższy zrzut ekranu:

Coverage java in ems-backend $ imes$: -
電子で とで、				
Element	Class, % ^	Method, %	Line, %	Branch, %
✓ ■ all	84% (16/19)	75% (70/93)	86% (196/226)	100% (8/8)
ⓒ main	0% (0/1)	0% (0/1)	0% (0/2)	100% (0/0)
in net.javaguides.ems_backend	88% (16/18)	76% (70/92)	87% (196/224)	100% (8/8)
© EmsBackendApplication	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
> in security	66% (2/3)	46% (6/13)	75% (30/40)	100% (0/0)
> 🖻 repository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
> exception	100% (1/1)	100% (1/1)	100% (1/1)	100% (0/0)
> 🗈 dto	100% (1/1)	66% (8/12)	66% (8/12)	100% (0/0)
> in entity	100% (3/3)	69% (16/23)	66% (16/24)	100% (0/0)
> incontroller	100% (3/3)	81% (13/16)	83% (26/31)	100% (2/2)
> loservice	100% (3/3)	100% (20/20)	100% (81/81)	100% (6/6)
> • mapper	100% (3/3)	100% (6/6)	100% (34/34)	100% (0/0)