

Book shop

Kacper Wójcik, Marcin Stefaonwicz, Paweł Nowak

June 13, 2023

Abstract

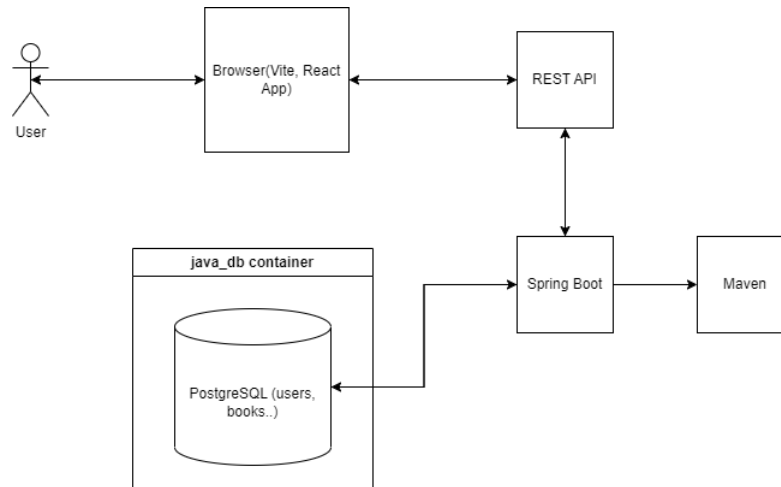
Book shop jest projektem sklepu internetowego z książkami. Głównym założeniem było stworzenie platformy przy użyciu Spring Boota z Mavenem, która pomoże użytkownikom w prowadzeniu sklepu internetowego. Przy użyciu dockera powstał kontener zawierający bazę danych PostgreSQL. React razem z Vite, odpowiedzialny jest za warstwę, którą bezpośrednio odbiera klient tj. warstwę frontu, z której dane są odbierane od klienta i wysyłane po przez REST API na część backendową, oraz do której są wysyłane informacje z serwera.

Inicjalizacja

Aby móc uruchomić projekt trzeba:

- zainstalować dockera
- sklonować zdalne repozytorium <https://github.com/KWojcik243/SKE-FB-Book-shop>
- wejść do folderu z projektem
- użyć komendy **docker-compose up --build**, aby zbudować pustą bazę
- w programie do uruchamiania kodu, odpalić plik "SKE-FB-Book-shop\src\main\java\com\example\demo\SkeFbBookShopApplication.java"
- zainstalować nodeJS
- w folderze SKE-FB-Book-shop\fronten\bookshop-frontend w terminalu użyć: "npm install" a następnie "npm run dev"

1 Projekt systemu



Rys. 1: Ogólny projekt systemu

1.1 Spring App

Przy użyciu spring boota budujemy główną aplikację i inicjujemy możliwość przepływu danych po przez np. REST API. Dodanie tokena JWT umożliwiło nam w sposób bezpieczny przesyłanie na do przeglądarki informacji o tym jaki użytkownik jest zalogowany, oraz jego prawa dostępu.

W tym projekcie został użyty Maven służący do kompilacji i budowania projektu, oraz do dodawania nowych zależności do frameworku.

Przy użyciu Spring Boota w wersji 3.0.6 został stworzony projekt ze starterów `m.in. jdbc, webmvc-ui`.

API posiada wiele endpointów, które używane są do tego, aby umożliwić odbiór informacji z frontu, oraz aby móc wysłać na niego niezbędne dane. Wszystkie endpointy dostępne są do sprawdzenia po odpaleniu projektu na stronie <http://localhost:8080/swagger-ui/index.html>

Używanie swaggera ułatwia pracę na wielu punktach końcowych, które przy skali projektu i ich ilości mogły by być ciężkie do zrozumienia bez odpowiedniej dokumentacji, a nawet wtedy swagger jest nie zastąpionym narzędziem do testowania tych endpointów.

1.2 React

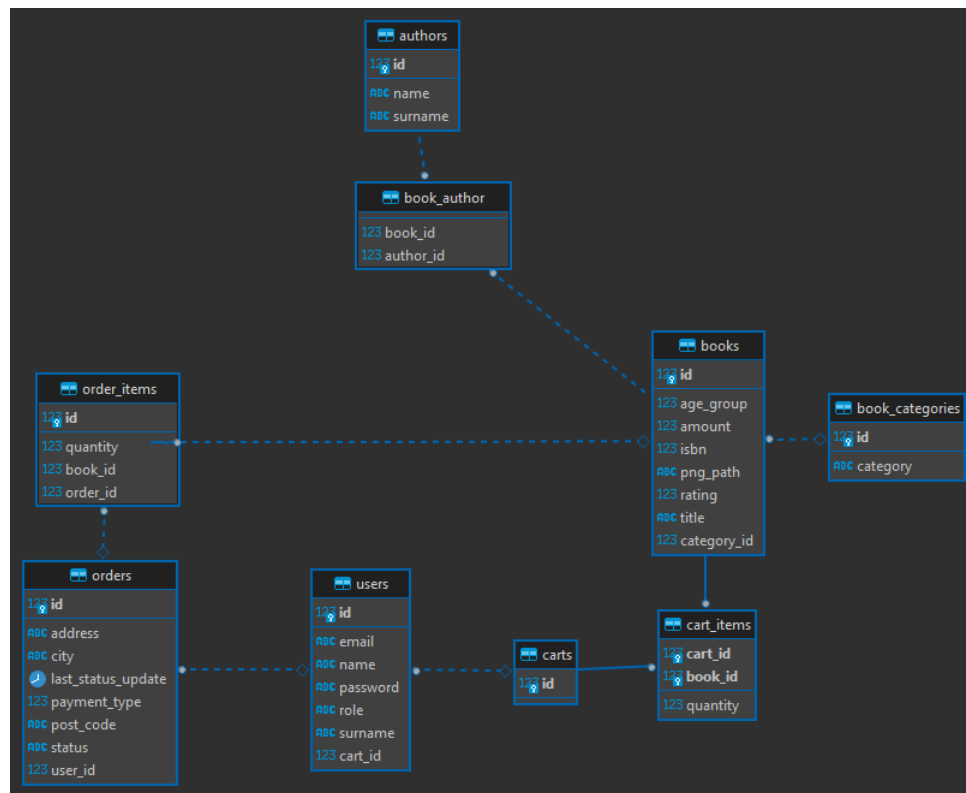
Aplikacja którą odbiera klient została napisana w reactie przy pomocy Vite. Najważniejszymi stronami są:

- <http://127.0.0.1:5173/dashboard>, gdzie Admin może dodawać książki, edytować je, usuwać, dodawać adminów itp
- <http://127.0.0.1:5173/catalog>, pozwala przeglądać książki i dodawać je do koszyka
- <http://127.0.0.1:5173/basket>, pozwala na składanie zamówień.

1.3 PostgreSQL

Kontener stworzony w docker-compose przechowuje i pozwala zarządzać bazą danych PostgreSQL. Została wystawiona na port 5433, aby umożliwić wymianę danych między nią i spring bootem, a jej zabezpieczenia np. możliwość stworzenia indywidualnych kont, oraz haseł pozwala zachować nam względne bezpieczeństwo. PostgreSQL został wybrany ze względu na jego popularność, szybkość działania, oraz to, że jest oparty na modelu relacyjnym. Baza danych zawiera w aktualnej formie 8 tabel, którymi są:

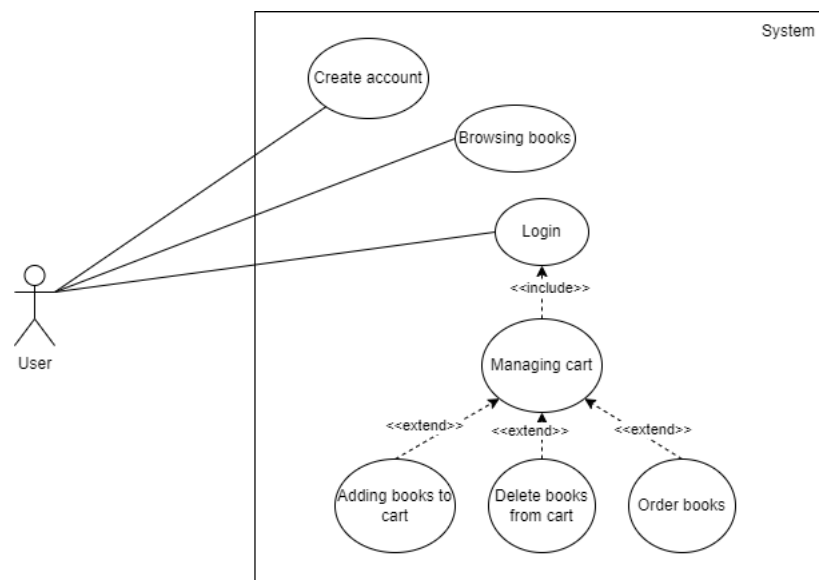
- users
- authors
- orders
- order_items
- books
- book_categories
- cart_items
- cart



Rys. 2: Projekt bazy danych

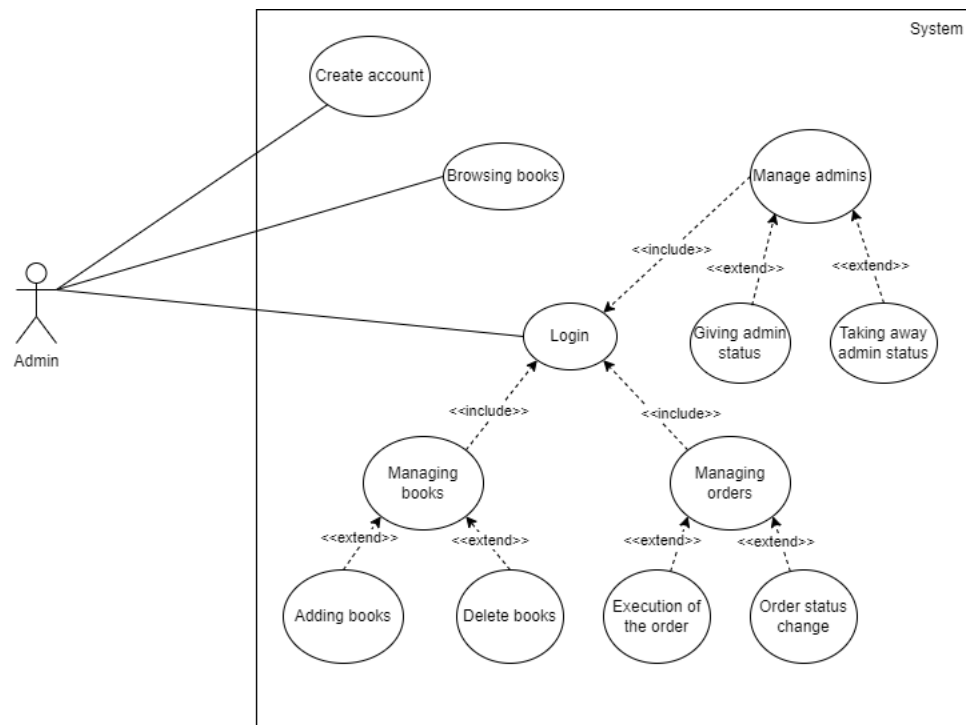
1.4 Przypadki użycia

Stworzony diagram przypadków użycia (Rys.3), przedstawia w sposób ogólny interakcje z aktorem(Userem), ramy systemu, oraz przypadki użycia. Użytkownik może stworzyć konto, zalogować się, przeglądać książki, a następnie po zalogowaniu zarządzać koszykiem tj. dodawać i usuwać książki z niego, lub przystąpić do zakupu.



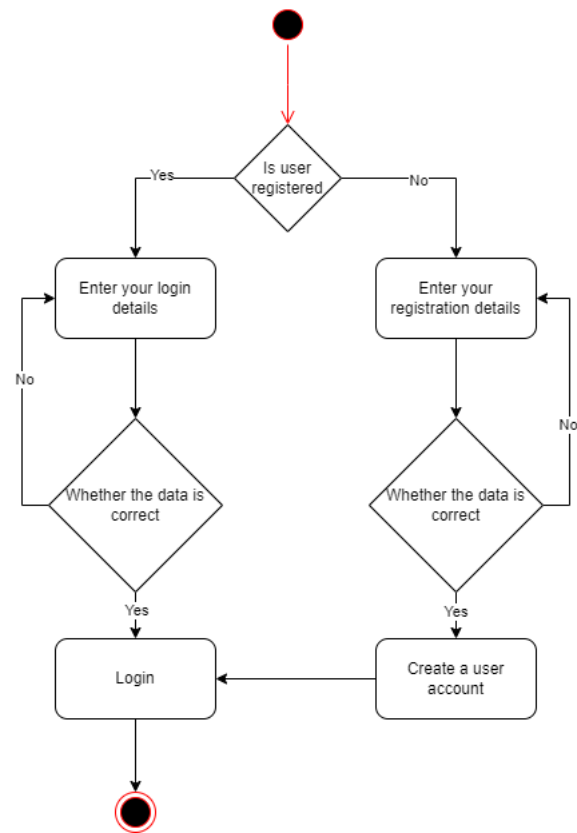
Rys. 3: Diagram przypadków użycia - User

Diagram przypadków użycia(Rys.4 Admin), przedstawia interakcje z aktorem(Adminem). Aktor może założyć konto, przeglądać książki. Po zalogowaniu dostaje możliwość zarządzania zamówieniami tj. zmiana statusu, realizacja itp., zarządzanie książkami, oraz nadawanie i odbieranie statusu admina innym użytkownikom.



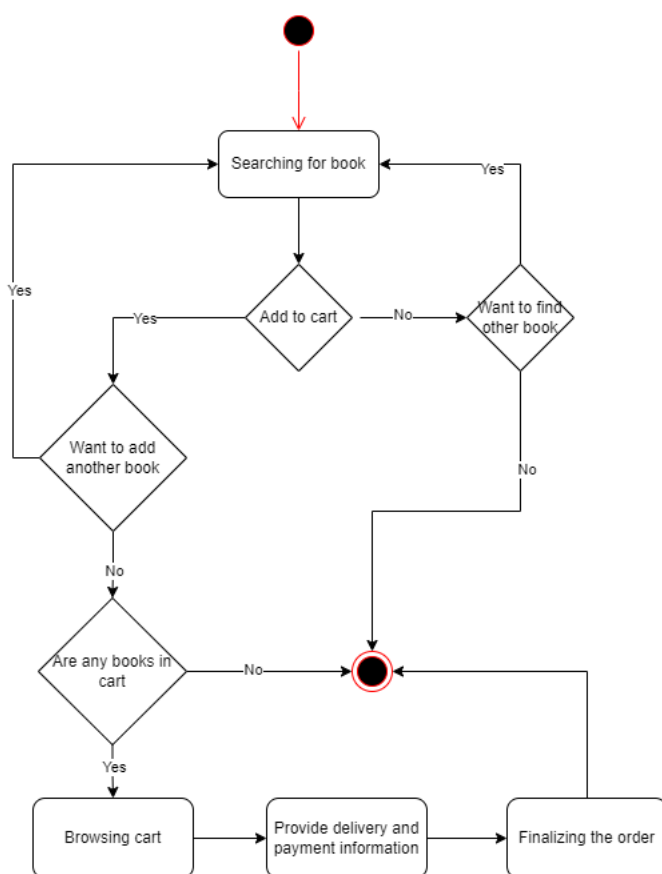
Rys. 4: Diagram przypadków użycia - Admin

Diagram aktywności logowanie/rejestracja(Rys.5) pokazuje proces rejestracji nowego użytkownika oraz proces logowania dla istniejących użytkowników. Zawiera weryfikację danych, oraz tworzenie konta użytkownika.



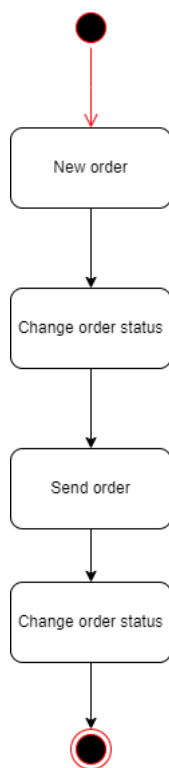
Rys. 5: Diagram aktywności - logowanie/rejestracja

Diagram aktywności zakup książek(Rys.6) obejmuje kroki, które użytkownik podejmuje, aby dokonać zakupu książki. Obejmuje to wyszukiwanie książki, dodawanie jej do koszyka, przeglądanie koszyka, podawanie informacji o dostawie, wybór metody płatności, finalizację zamówienia itp.



Rys. 6: Diagram aktywności - zakup książek

Diagram aktywności obsługa zamówień(Rys.7), skupia się na obsłudze zamówień po ich złożeniu. Obejmuje głównie akcje zmiany statusu, oraz wysyłki.



Rys. 7: Diagram aktywności - obsługa zamówień