

18.06.2020

Programowanie Zespołowe - Informatyka Stosowana

Wydział Fizyki i Informatyki Stosowanej

BookStore

Wykonawcy:

**Aleksandr Shaforostov
Michał Prokop
Bartłomiej Pitala
Katarzyna Cieślak
Michał Matuszewski
Paula Palcowska
Wojciech Franczuk
Karol Stasiak
Dominik Kaczmarczyk
Evgeniy Natalenko
Robert Gałat
Gabriel Górski
Eryk Jarocko**

Spis treści:

- 1. Opis projektu**
- 2. Interfejs**
- 3. Ogólny schemat ideowy**
- 4. Mikroserwisy**
- 5. Diagramy Use Case**
- 6. REST API**
- 7. Podział zadań**

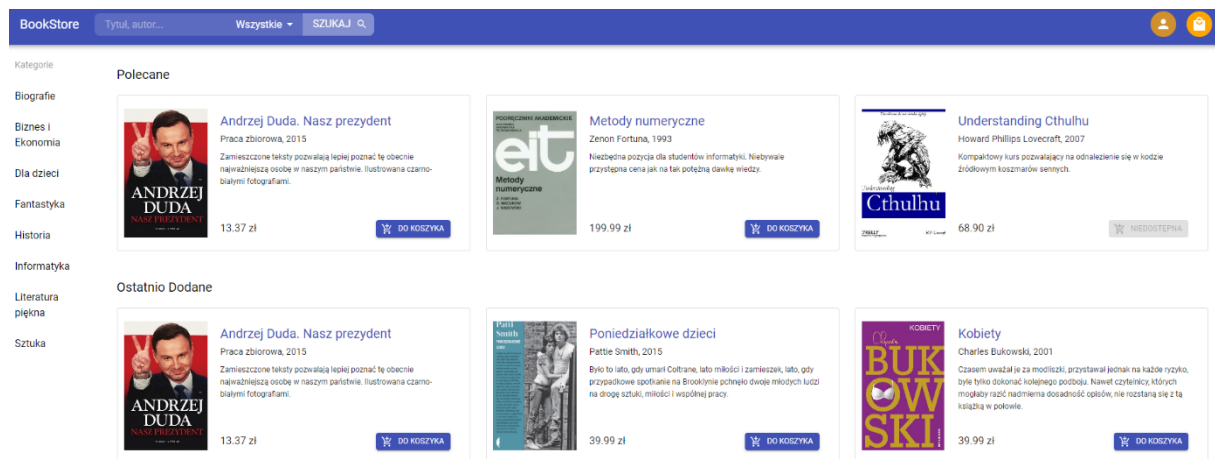
1. Opis projektu

Sklep internetowy – księgarnia „BookStore” z architekturą opartą na mikroservisach oraz REST API. Deployment wykonany z użyciem Kubernetesa i Dockera.

Użytkownikami sklepu mogą być klienci oraz administrator, który dysponuje osobnym panelem pozwalającym na edytowanie wystawionych produktów, klientów sklepu i zamówień.

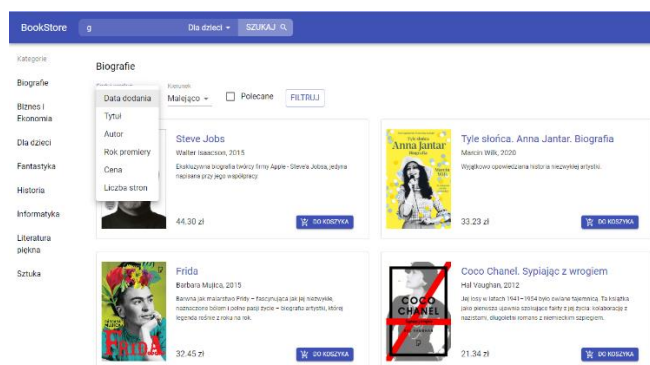
2. Interfejs

- Strona główna:



Po lewej stronie lista kategorii do wybrania. U góry search box z możliwością wyszukiwania książek po dowolnym ciągu znaków (wyszukiwanie ciągu w tytule, nazwisku i imieniu autora) w wybranej kategorii. U góry po prawej stronie dwa przyciski: koszyk i ustawienia konta. W części centralnej wyświetlane są proponowane książki wraz z opisami.

- wyniki wyszukiwania:



Istnieje możliwość sortowania książek po: dacie dodania, tytule, autorze, roku premiery, cenie, liczbie stron.

- Panel administratora

BookStore

g

Dla dzieci

SZUKAJ

PANEL ADMINISTRATORA

A

Zarządzaj

Zamówienia

Zamówienia

Książki

Użytkownicy

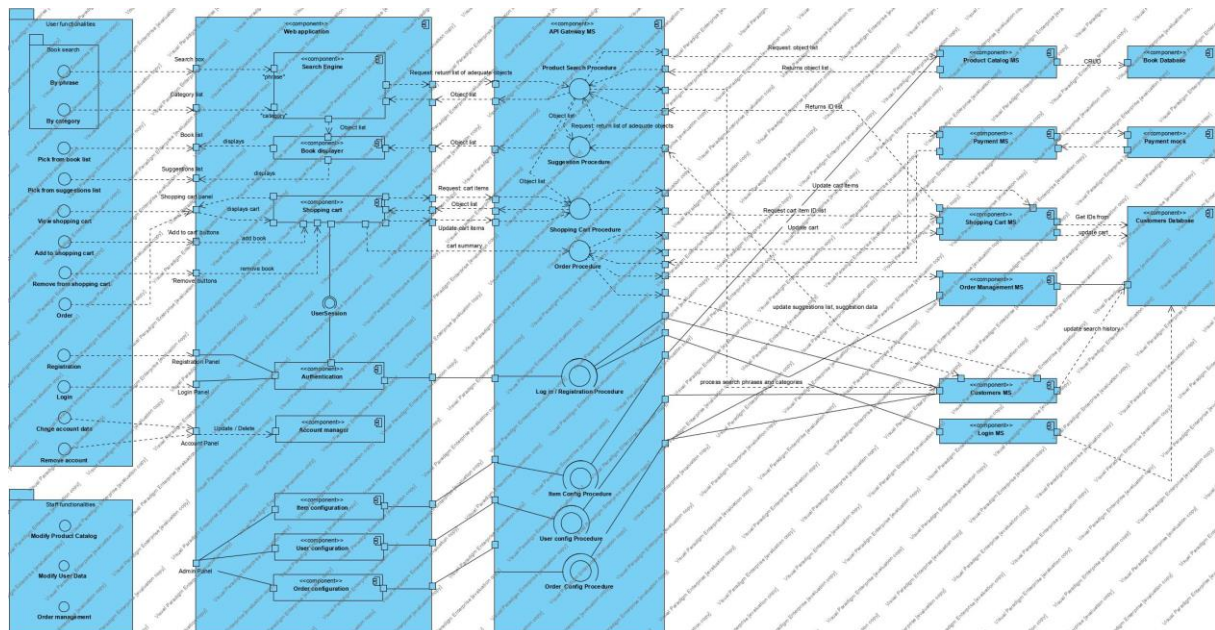
Kategorie

Kupony

ID	ID Koszyka	Adres dostawy	Data dostawy	Status	Akcje
2	120	Ulicowa, 1234, Krakow, Poland	2020-06-20	DISAPPROVED	... <div><div></div><div></div><div></div></div>
3	139	ulica 1, miasto , 30-057, Kraków, Poland	2020-06-16	PLACED	... <div><div></div><div></div><div></div></div>
4	105	Ulica, 123, Krakow, Poland	2020-06-20	PLACED	... <div><div></div><div></div><div></div></div>
5	157	asdf, 123, krk, Poland	2020-06-20	DELIVERED	... <div><div></div><div></div><div></div></div>
6	159	asdf, 213, Krakow, Poland	2020-06-20	PLACED	... <div><div></div><div></div><div></div></div>
7	160	tests, 123, Krakow, Poland	2020-06-20	APPROVED	... <div><div></div><div></div><div></div></div>
8	162	abc, 213, 123, Poland	2020-06-20	PLACED	... <div><div></div><div></div><div></div></div>
9	166	123, 123, 123, Poland	2020-06-20	DELIVERED	... <div><div></div><div></div><div></div></div>

Administrator w swoim panelu ma możliwość zarządzania zamówieniami i kontami użytkowników, książkami i kategoriami książek w sklepie oraz kuponami promocyjnymi.

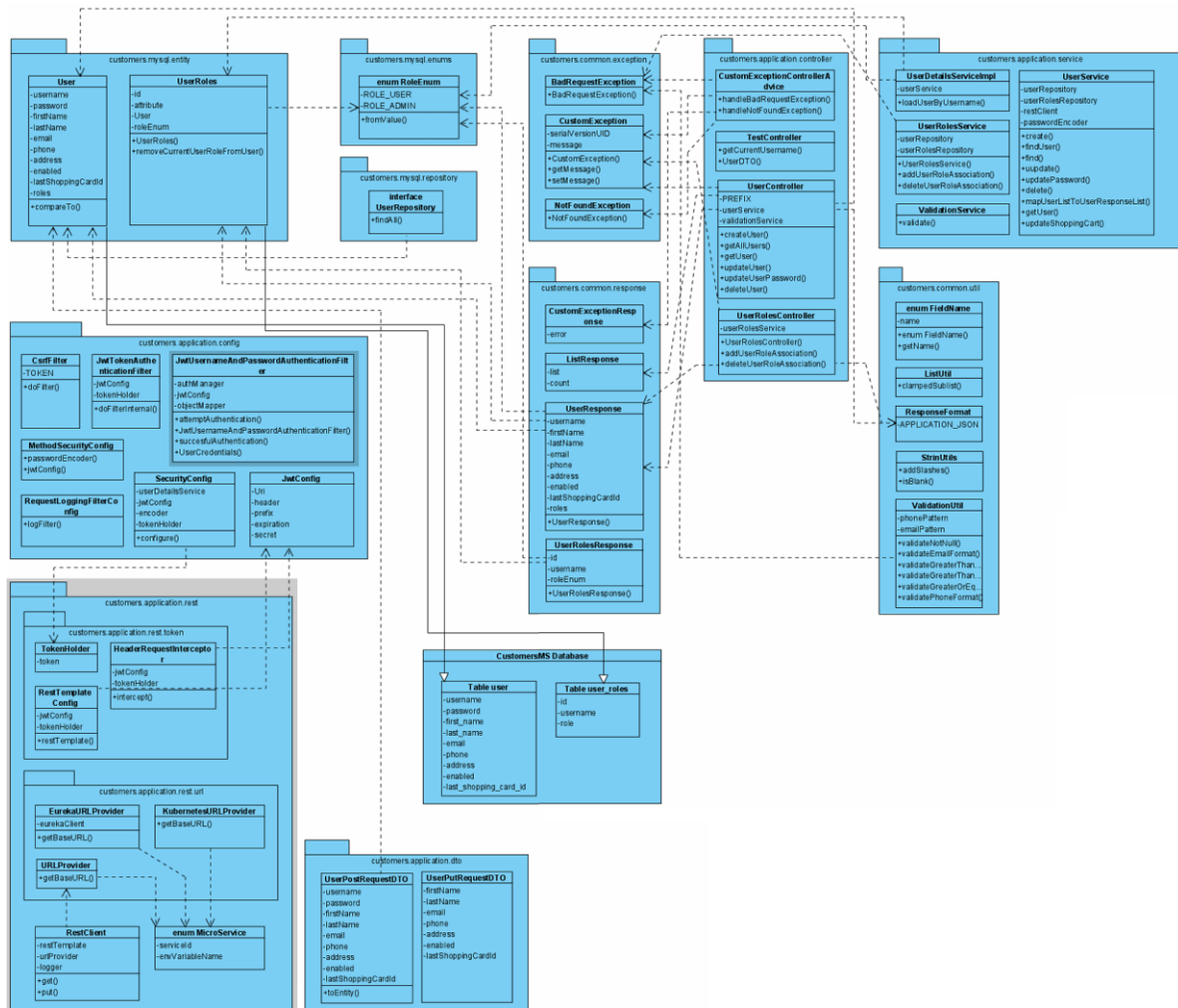
3. Ogólny schemat ideowy



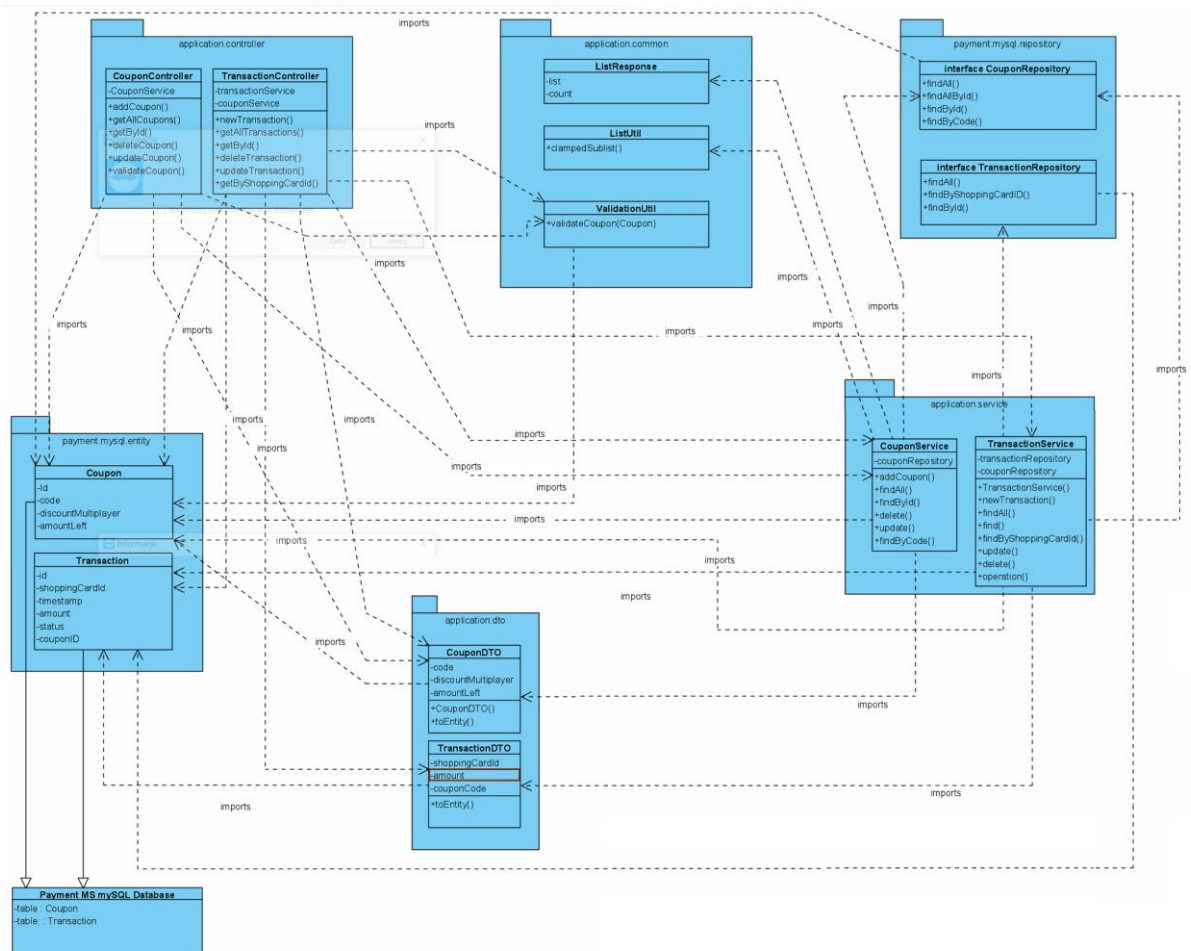
Wszystkie mikroserwisy komunikują się przez API Gateway i posiadają własne bazy danych.

4. Mikroserwisy (MS)

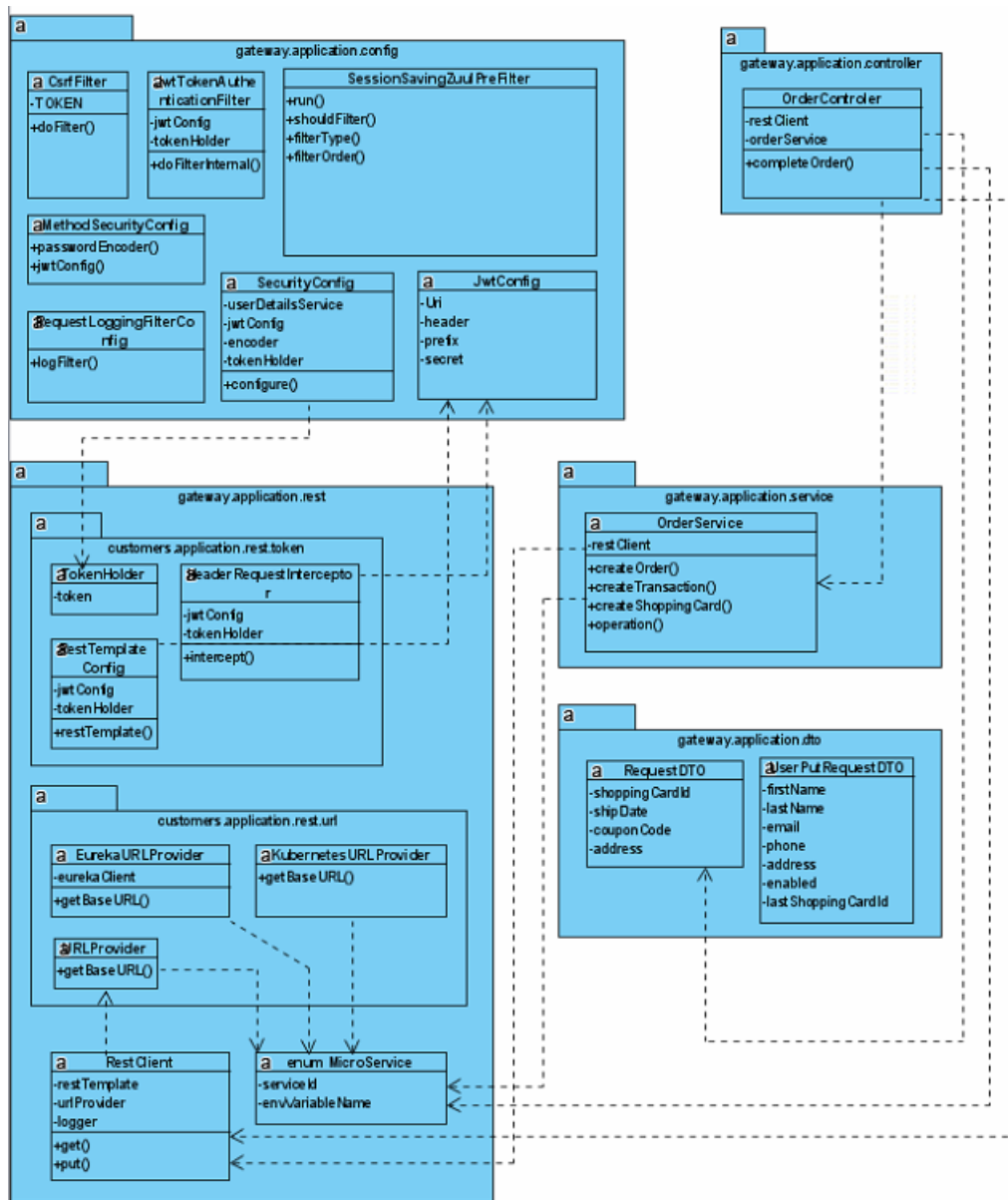
a. Customers MS



b. Payment MS



c. APIGateway MS



5. Diagramy Use Case

Diagram „Use Case” – wyszukiwanie książki w katalogu

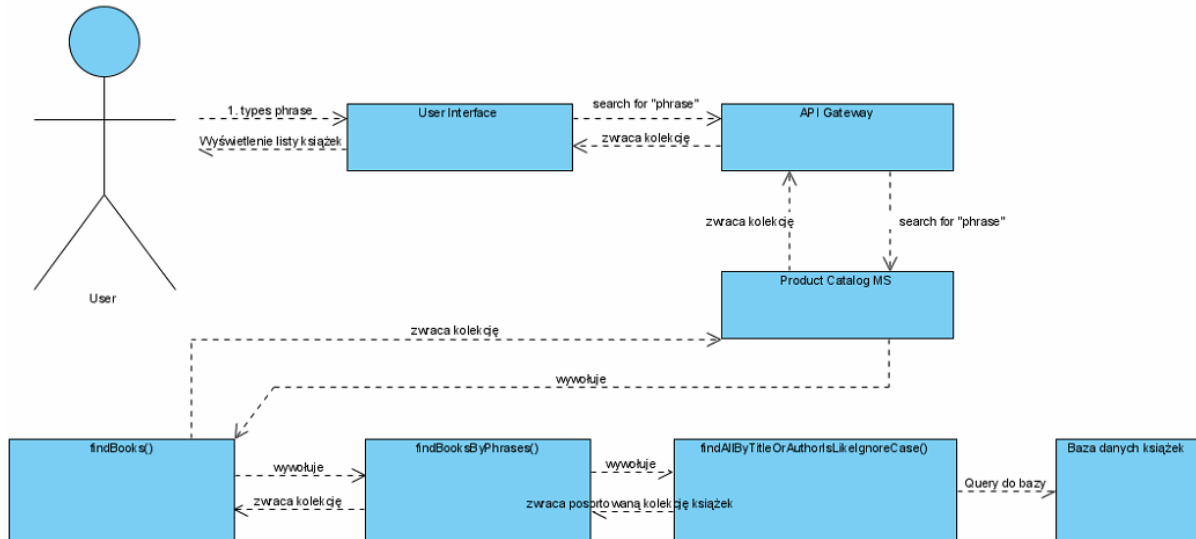
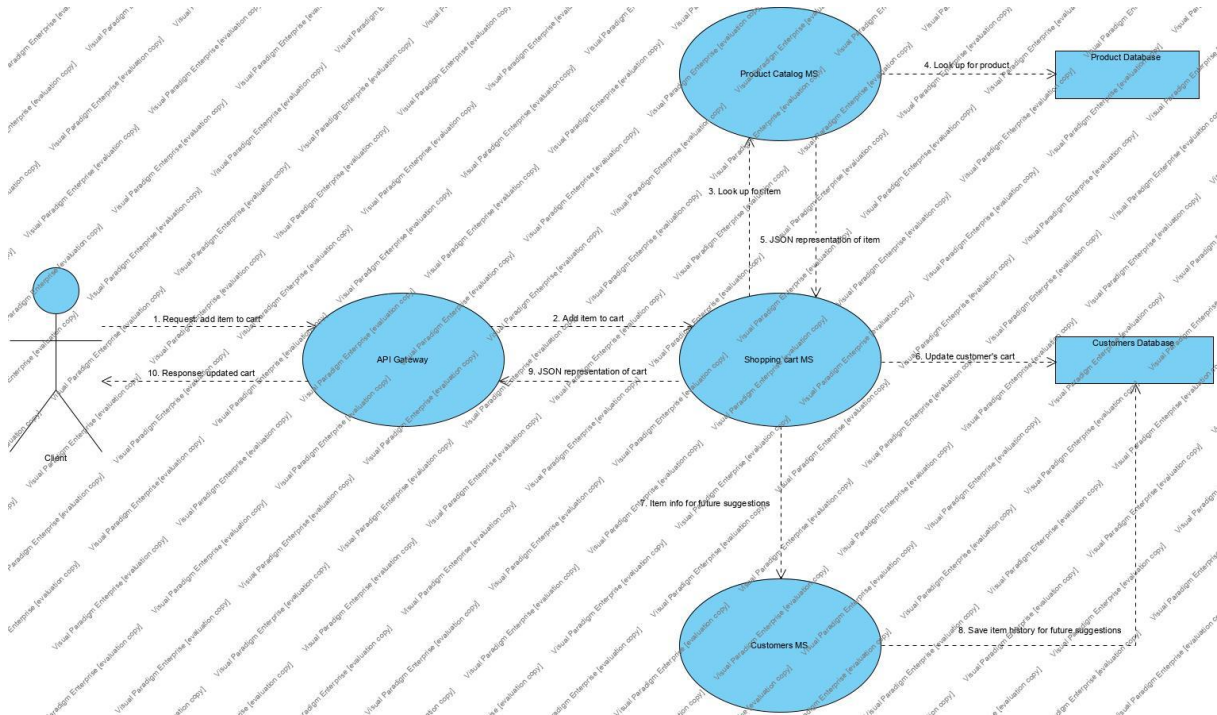


Diagram „Use Case” – dodawanie książki do koszyka



6. REST API

Architektura REST API została zaimplementowana w Swaggerze i jest dostępna pod adresem:

<https://app.swaggerhub.com/apis/karsta26/Bookstore/1.9.2?fbclid=IwAR2-r8cwX4IO9IDaODCFK8QvgTO9fli5A-pNtjDMAW0yA5NbBRgJ-1raVm4#/>

Prawa dostępu – uwierzytelnianie użytkowników:

books

POST/books	admin
GET/books	każdy
GET/books/{id}	każdy
PUT/books/{id}	admin
DELETE/books/{id}	admin
PATCH/books/{id}	admin

categories

POST/categories	admin
GET/categories	każdy
GET/categories/{id}	każdy
DELETE/categories/{id}	admin

orders

GET/api/order-management-ms/orders	zalogowany użytkownik
POST/api/order-management-ms/orders	zalogowany użytkownik
GET/api/order-management-ms/orders/{orderId}	zalogowany użytkownik
DELETE/api/order-management-ms/orders/{orderId}	zalogowany użytkownik
PUT/api/order-management-ms/orders/{orderId}	admin

users

POST/users	dodawać userów może tylko admin
POST/.../users/{username}/roles/{role}	admin

DELETE/.../users/{username}/roles/{role}	admin
GET/users	admin
GET/users/{username}	admin i user jeśli {username}=zalogowany_user
PUT/users/{username}	admin i user jeśli {username}=zalogowany_user
DELETE/users/{username}	admin
PATCH/users/{username}	admin i user jeśli {username}=zalogowany_user

shoppingCart

POST/shoppingCards	jeśli ShoppingCartRequestDTO.username jest nullem to każdy, jeśli jest tam username to tylko on i admin
GET/shoppingCards/{id}	jeśli ShoppingCartResponseDTO.username jest nullem to każdy, jeśli jest tam username to tylko on i admin
GET/shoppingCards	admin
PUT/shoppingCards	jeśli ShoppingCartRequestDTO.username jest nullem to każdy, jeśli jest tam username to tylko on i admin
DELETE/shoppingCards/{id}	właściciel koszyka (jeśli jest, jak nie to każdy) i admin
POST/shoppingCards/items	właściciel koszyka (jeśli jest, jak nie to każdy) i admin
GET/shoppingCards/items/{id}	właściciel koszyka (jeśli jest, jak nie to każdy) i admin
PUT/shoppingCards/items/{id}	właściciel koszyka (jeśli jest, jak nie to każdy) i admin
DELETE/shoppingCards/items/{id}	właściciel koszyka (jeśli jest, jak nie to każdy) i admin
GET/.../shoppingCards/{shoppingCardId}/items	właściciel koszyka (jeśli jest, jak nie to każdy) i admin

7. Podział Zadań

Michał Prokop (Tech lead) :

- przygotowanie designu architektury systemu
- diagramów
- API
- dokumentacji do projektu.

Bartłomiej Pitala(Backend) :

- uzupełnienie braków w implementacji
- dodanie filtrowania i sortowania do product-ms
- dodanie liczby stron, oprawy, daty - do książki
- tworzenie dockerfiles.

Katarzyna Cieślak(Backend) :

- dodanie JWT do product-ms,
- dodanie praw admina dla endpointów
- autoryzacja endpointów
- testowanie

Michał Matuszewski(Backend) :

- tworzenie całych Product MS
- API Gateway MS
- Customers MS
- Login MS.
- wykonanie roli team leadera backendu.
- postawienie EurekaServiceDiscovery

Paula Palcowska(Backend) :

- tworzenie mikroserwisu Shopping Cart MS
- testy do tego mikroserwisu;
- schema bazy danych shopping

Wojciech Franczuk(Backend) :

- tworzenie mikroserwisu Payment ms
- testy do tego mikroserwisu
- dodanie API w payment-ms do obsługi kuponów
- w Api Gateway stworzenie endpointu łączącego finalizowanie transakcji
- wyliczanie całkowitej ceny w koszyku

Karol Stasiak(Backend) :

- tworzenie Order Management MS
- dodanie komunikacji między MS
- zmiany w shopping cart (nowy shape danych)
- wykonanie roli team leader backendu.
- definiowanie zadań i ich rozdzielanie
- opieka nad kodem, w tym code review
- tworzenie uwierzytelniania z użyciem JWT
- dodawanie kodu do komunikacji między mikroserwisami
- opieka nad bazą danych
- drobne zmiany i aktualizacji w mikroserwisach

Dominik Kaczmarczyk(Frontend) :

- wykonanie roli team leadera dla frontendu
- architektura frontendu (diagram UML komponentów)
- prototyp strony w Adobe XD
- strona główna
- strona kategorii
- admin panel (listy, dodawanie, edycja niektórych elementów)
- logika i strona wyszukiwania
- paginacja
- navbar stylowanie strony w Material UI

Evgeniy Natalenko(Frontend) :

- strona książki (admin panel)
- strona profilu usera (admin panel)

- strona zamówienia (admin panel)
- edycja zamówienia (admin panel)
- dodawanie zamówienia (admin panel)
- edycja usera (admin panel)
- strona pojedynczej książki (user)
- strona profilu użytkownika (user)
- edycja danych profilu (user)
- pomoc z dockerfiles

Robert Gałat (Backend) :

- dodanie last koszyk id do usera
- dodanie adresu do order oraz ceny do shopping cart item;
- dodanie actuator (albo inne logowanie) do każdego ms
- security do tego

Gabriel Górski(DevOps) :

- przygotowanie repozytorium dla projektu na github
- stworzenie manifestów travisowych per projekt wraz z ustawieniem zmiennych i sekretów
- stworzenie manifestów k8s per projekt
- stworzenie skryptu Python'owego do podmieniania tokenów na zasadzie wzorca na potrzeby pipeline
- stworzenie projektu w Google Cloud, postawienie klastra k8s + bazy MySql
- stworzenie użytkownika serwisowego z ograniczonymi uprawnieniami na potrzeby trawisa
- nadanie uprawnień podglądania stanu klastra k8s programistom do potrzeb deweloperskich / debuggowania
- stworzenie infrastruktury CI + CD w oparciu o wymienione technologie
- zabezpieczenie repozytoriów poprzez uprawnienia i podzielnie całego zespołu na podgrupki (Frontend, Backend)
- model integracji oparty na Pull Requestach które muszą przejść poprawny build, unit testy & review co najmniej jednej osoby
- zablokowanie możliwości pusha do mastera na każdym z repozytoriów
- zmigrowanie projektu na nowe konto Google po wyczerpaniu się darmowego kredytu \$300
- założenie konta i repozytoriów artefaktów per projekt w hub.docker.com

Eryk Jarocko(DevOps) :

- tworzenie plików Dockerowych
- testowanie łączenia mikroservisów (docker-compose tool)
- dodawanie wersji produkcyjne profili w aplikacjach backendowych
- postawienie od zera podstawkę pod projekt wraz z ustawieniami tego projektu pod kątem deploymentu
- walidacja kodu
- widok koszyka
- widok zamówienia
- strona logowania i rejestracji
- poprawianie błędów
- robienie code review