

Dokumentacja projektu

Przetwarzanie danych w chmurach obliczeniowych

Karol Szeliga

WFIS AGH

28.12.2021

Link do strony:

<https://basic-online-shop.herokuapp.com/>

Link do repozytorium:

<https://github.com/KarolSzeliga4/projekt-Chmury-Obliczeniowe>

1. Główna funkcjonalność aplikacji

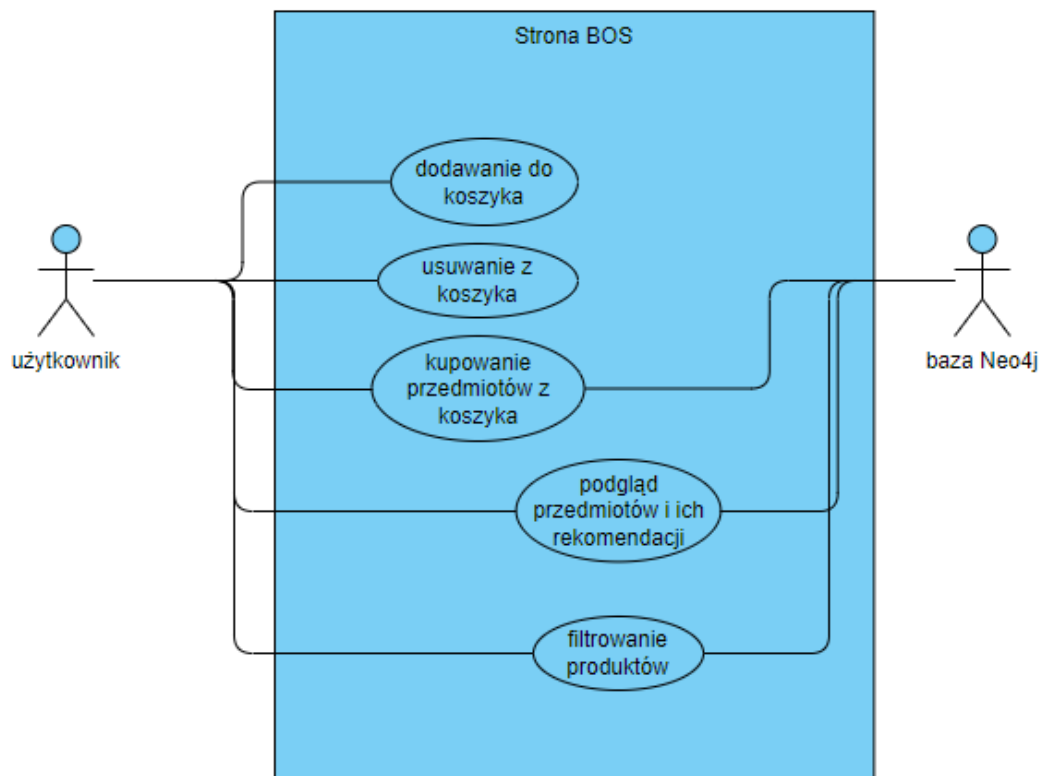
Strona to uproszczony sklep internetowy, która dzięki grafowej bazie danych jest w stanie szybko egzekwować rekomendacje produktów bazując na produktach zakupionych wcześniej w zestawie lub posiadające tę samą markę (markę można porównać z jednym sprzedawcą, co oznacza tańszy koszt przesyłki).

2. Dostępne funkcje aplikacji

Do udostępnionych funkcji zaliczają się:

- przeszukiwanie produktów poprzez odpowiednie filtry;
- podgląd produktu wraz z rekomendowanymi do niego produktami;
- dodawanie produktów do koszyka;
- usuwanie produktów z koszyka;
- podgląd koszyka;
- kupowanie zestawu produktów z koszyka.

Wszystkie funkcjonalności w uproszczonej formie, tylko po to, aby umożliwić sprawdzenie efektów zastosowania rekomendacji.



Rys.1 UC diagram.

3. Opis implementacji

Główny opis:

Strona internetowa napisana we frameworku Flask (Python3) hostowana na platformie Heroku, komunikująca się z grafową bazą danych z serwisu Neo4J.

W bazie danych znajdują się produkty możliwe do kupienia w sklepie. Każde kupno towarów w zestawie tworzy między nimi relacje. Tworzenie rekomendacji dla danego artykułu polega na policzeniu posiadanych przez niego połączeń z innymi przedmiotami. Im więcej relacji, tym wyżej na liście rekomendacji będzie produkt. Poza tym, rekomendowane są też produkty o tej samej marce.

Produkty:

Aby lepiej przetestować działanie funkcjonalności, w projekcie posłużono się przykładową bazą rowerów z historią zakupionych zestawów. Każdy produkt ma nazwę, cenę, markę, kategorie i rok produkcji. Zastosowane grupowania umożliwiają sprawdzenie poprawności działania rekomendacji. Intuicyjnie razem będą kupowane przedmioty droższe, z jednej marki, czy te z kategorii dziecięcej często w zestawie z miejskimi itd.

Połączenie z bazą danych:

Połączenie z instancją bazy danych z serwisu Neo4J zostało zrealizowane poprzez wykorzystanie sterownika z biblioteki Python-owej neo4j. Tworzy on sesję transakcji umożliwiającą czytanie i zapisywanie do bazy. Zapytania zostały napisane w języku Cypher.

5. Podsumowanie

Dzięki zastosowaniu grafowej bazy danych udało się stworzyć szybko działający system rekomendacji dla sklepu internetowego. Szczególnie pomocna była przy zapytaniu o relacje między produktami (w innej bazie danych trzeba by przeszukiwać całą historię zamówień). Niestety w wielu przypadkach rekomendacje okazały się nie mieć jasnej tendencji do intuicyjnego grupowania się. Może być to spowodowane niedostrzegalną tendencją klientów lub użyta baza mogła posiadać fałszywe informacje o historii zamówień. Poza tym przy ewentualnym znacznym zwiększeniu ilości produktów w bazie nieuniknione byłoby przekształcenie kilku pętli w kodzie strony internetowej, aby utrzymać akceptowalny czas ładowania strony.

6. Źródła

- [użyta przykładowa baza rowerów z historią zakupionych zestawów](#);
- [platforma heroku](#);
- [serwis Neo4J](#);
- [większość problemów rozwiązywana z pomocą stackoverflow :\)](#)