Obraz zawierający tekst

Opis wygenerowany automatycznie

|  |
| --- |
| Podstawy konteneryzacji i architektury mikroserwisów |
| System Edukujący Nowe Gusta Muzyczne |
| Projekt końcowy |

|  |
| --- |
| IO3 Marta Chadała 235842  2023-05-26 |

Spis treści

[**STRESZCZENIE** 2](#_Toc137049390)

[**KEYWORDS** 2](#_Toc137049391)

[I.WSTĘP 2](#_Toc137049392)

[A.Obszar / tło działań 2](#_Toc137049393)

[B.Cel Projektu 2](#_Toc137049394)

[II.OPIS ROZWIĄZANIA 2](#_Toc137049395)

[A. Opis techniczny rozwiązania 2](#_Toc137049396)

[1. Pseudokod 2](#_Toc137049397)

[2. Przykład obsługi 4](#_Toc137049398)

[3. Diagram stanów (STD) 4](#_Toc137049399)

[4. Diagram związków encji (ERD) 4](#_Toc137049400)

[5. Diagramy przepływu danych (DFD) 5](#_Toc137049401)

[B. Testy 5](#_Toc137049402)

[C. Wyniki i analiza 5](#_Toc137049403)

[III.WNIOSKI I PERSPEKTYWY ROZWOJU 5](#_Toc137049404)

[Bibliografia 6](#_Toc137049405)

[ZAŁĄCZNIKI 6](#_Toc137049406)

# STRESZCZENIE

W artykule przedstawiono metody działania programu do sugerowania utworów przez wyświetlanie linków do losowych utworów na Wikipedii (<https://en.wikipedia.org/wiki/Main_Page>) oraz Shazam (<https://www.shazam.com/pl>).

# KEYWORDS

Docker, kontener, losowość, muzyka, Shazam, Wikipedia

# I. WSTĘP

Niniejszy raport jest poświęcony wykorzystaniu kontenerów w generowaniu losowych linków do artykułów o piosenkach ze stron Wikipedii oraz Shazam.

Realizowany projekt obejmował zagadnienia dotyczące:

* Algorytmów
* Programowania skryptowego (w tym przypadku język Python)
* Konteneryzacji i mikroserwisów (w tym przypadku z użyciem Dockera)

W raporcie został zamieszczony opis działania aplikacji oraz jej wyniki oraz aktualny stan wiedzy na temat podobnych rozwiązań jakie można znaleźć na ryku wraz z ich krótkim opisem.

## A. Obszar / tło działań

Aktualnie na rynku istnieje wiele rozwiązań polecających muzykę na podstawie historii słuchanych utworów czy zadeklarowanych preferencji użytkownika. [Spotify](https://open.spotify.com/) używa licznych metryk by spróbować dopasować polecane utwory do gustów użytkownika (Pastukhov 2022). Istnieją też serwisy takie jak np. [randommer](https://randommer.io/random-songs) mogące generować linki do losowych utworów ale mają one swoje kryteria wykluczające piosenki z puli wybieranych takie jak np. ilość wyświetleń na YouTube (randommer 2023).

## B. Cel Projektu

W projekcie skupiono się na przedstawieniu informacji o losowo wybranych piosenkach zamiast ich słuchaniu. W tym celu piosenki są wybierane z tych na liście piosenek w angielskiej wersji artykułu [„Piosenki według roku powstania” na Wikipedii](https://en.wikipedia.org/wiki/Category:Songs_by_year) oraz strony Shazam.

# II. OPIS ROZWIĄZANIA

## A. Opis techniczny rozwiązania

W projekcie zastosowano konteneryzację programów napisanych w Pythonie oraz oficjalny obraz mongodDB (<https://hub.docker.com/_/mongo/>). Wykorzystano również biblioteki socket, requests, random, pymongo do łączenia z bazą danych NoSQL (mongo) i wikipediaapi w celu użycia API Wikipedii.

### 1. Pseudokod

Program składa się z 4 kontenerów w tym jednego z bazą danych i jednego programu służącego jako interfejs dla użytkownika.  
User interface (ui.py)

1. Połącz się z łącznikiem
2. Poinformuj użytkownika, że może zacząć wpisywać
3. Otrzymaj tekst od użytkownika
4. Jeżeli ten tekst to „quit” wyślij do łącznika wiadomość „quit” i zakończ działanie  
   w przeciwnym razie wyślij do łącznika wiadomość „request”
5. Czekaj na odpowiedź od łącznika
6. Wyświetl użytkownikowi odpowiedź od łącznika
7. Wróć do punktu 3

#### Łącznik (connector.py w kontenerze)

1. Ustal własny adres IP
2. Zwiąż port 7777
3. Rozpocznij słuchanie
4. Zaakceptuj połączenie od kontenera losującego z Shazam
5. Zaakceptuj połączenie od kontenera losującego z Wikipedii
6. Zaakceptuj połączenie od programu interfejsu użytkownika
7. Czekaj na wiadomość od programu interfejsu użytkownika
8. Jeżeli wiadomość to „quit” prześlij wiadomość „quit” do kontenera losującego z Shazam i kontenera losującego z Wikipedii i zakończ działanie.  
   Jeżeli wiadomość to „request” przejdź do punktu 9  
   w przeciwnym razie przejdź do punktu 7
9. Wylosuj losową liczbę ze zbioru {1, 2}
10. Jeśli losowa liczba to 1 przejdź do punktu 11  
    w przeciwnym wypadku przejdź do punktu 15
11. Wyślij wiadomość „request” do kontenera losującego z Shazam
12. Czekaj na odpowiedź od kontenera losującego z Shazam
13. Wyślij do programu interfejsu użytkownika uzyskaną odpowiedź
14. Przejdź do punktu 7
15. Wyślij wiadomość „request” do kontenera losującego z Wikipedii
16. Czekaj na odpowiedź od kontenera losującego z Wikipedii
17. Wyślij do programu interfejsu użytkownika uzyskaną odpowiedź
18. Przejdź do punktu 7

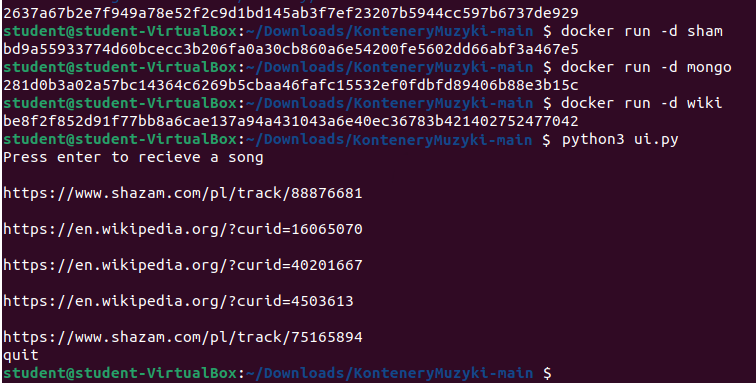
#### Losowanie z Shazam (shazam.py w kontenerze)

1. Połącz się z łącznikiem
2. Czekaj na wiadomość od łącznika
3. Jeżeli wiadomość to „quit” zakończ działanie.  
   Jeżeli wiadomość to „request” przejdź do punktu 4  
   w przeciwnym wypadku przejdź do punktu 2
4. Wygeneruj liczbę całkowitą z przedziału 10000000 do 99999999
5. Połącz tekst „https://www.shazam.com/pl/track/” z wygenerowaną liczbą by uzyskać potencjalny adres url
6. Pobierz tekst z potencjalnego url
7. Jeśli tekst „error-page panel-landing” jest zawarty w kekście z potencjalnego url wróć do punktu 4
8. Wyślij do łącznika potencjalny adres url
9. Wróć do punktu 2

#### Losowanie z Wikipedii (wiki.py w kontenerze)

1. Połącz się z bazą danych
2. Dla każdej kategorii na stronie <https://en.wikipedia.org/wiki/Category:Songs_by_year> stwórz wpis w bazie danych składający się z nazwy kategorii i wartości będącej liczbą wszystkich piosenek w poprzednio dodanych kategoriach
3. Czekaj na wiadomość od łącznika
4. Jeżeli wiadomość to „quit” zakończ działanie.  
   Jeżeli wiadomość to „request” przejdź do punktu 5  
   w przeciwnym wypadku przejdź do punktu 3
5. Wygeneruj liczbę całkowitą z przedziału 0 do liczby wszystkich piosenek we wszystkich kategoriach w bazie danych
6. Pobierz z bazy danych wpis z największą wartością będącą mniejszą lub równą do wylosowanej wartości
7. Odejmij wartość pobraną z bazy danych od wygenerowanej liczby
8. Zwróć stronę piosenki o indeksie obliczonym przez odejmowanie w kroku 7 w pobranej z bazy danych kategorii
9. Wyślij do łącznika stronę piosenki z punktu 8
10. Wróć do punktu 3

### 2. Przykład obsługi



Linki w formie tekstowej:  
<https://www.shazam.com/pl/track/88876681>  
<https://en.wikipedia.org/?curid=16065070>

<https://en.wikipedia.org/?curid=40201667>

<https://en.wikipedia.org/?curid=4503613>

<https://www.shazam.com/pl/track/75165894>

### 3. Diagram stanów (STD)

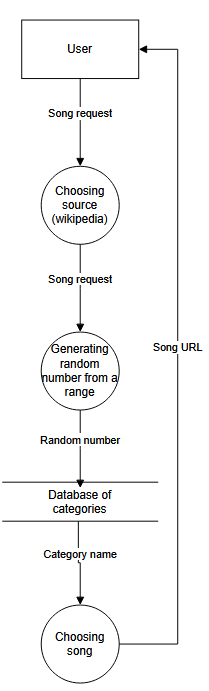
Rysunek 1 Diagram stanów programu

### 4. Diagram związków encji (ERD)

### 5. Diagramy przepływu danych (DFD)

Rysunek 2 Diagram przepływu danych przy losowaniu piosenki z Wikipedii

Rysunek 3 Diagram przepływu danych przy losowaniu piosenki z Shazam



## B. Testy

Istnieją niedoskonałości w działaniu aplikacji spowodowane odnoszeniem się do zasobów w internecie. Dla przykładu posłużę się utworem wylosowanym przez moduł losujący z Wikipedii dla losowej wartości 93884. Wygenerowany link „<https://en.wikipedia.org/?curid=19404136>” prowadzi do strony definiującej anioła. Po przeanalizowaniu odpowiedniego wpisu w kategorii piosenki z roku 2019 można znaleźć piosenki zatytułowane „Angel”, które nie przekierowują do oddzielnych artykułów przez co wyszukanie ich przez API Wikipedii nakierowuje na artykuł o istocie anioła a nie piosence o tytule Anioł.

## C. Wyniki i analiza

Program bez wątpienia generuje linki z piosenkami ale nie jest idealnym rozwiązaniem ze względu na brak obiektywnego zbioru wszystkich piosenek.

# III. WNIOSKI I PERSPEKTYWY ROZWOJU

Program zdecydowanie jest narzędziem dzięki, któremu można poszerzać swoją wiedzę o muzyce. Jednocześnie warto zauważyć, że można w łatwy sposób go rozwijać dodając nowe źródła losowania piosenek.

# Bibliografia

Dmitry Pastukhov. 2022. Music Tomorrow. 9 Luty. Data uzyskania dostępu: czerwiec 6, 2023. https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022.

2023. randommer. Data uzyskania dostępu: Czerwiec 6, 2023. https://randommer.io/random-songs.

# ZAŁĄCZNIKI

Zał. 1 – Kod udostępniony na platformie GitHub - <https://github.com/Ewangelina/KonteneryMuzyki>   
Zał. 2 – Instrukcja uruchamiania programu

Instrukcja zakłada, że na systemie zainstalowany jest Docker i python3.

1. Pobierz i rozpakuj kod z <https://github.com/Ewangelina/KonteneryMuzyki>
2. Otwórz terminal w folderze głównym programu
3. Wpisz:  
   docker build -t con ./connector

docker build -t sham ./shazam

docker build -t mongo ./database

docker build -t wiki ./wikipedia

1. Wpisz:  
   docker container run -d -p 8080:7777 con  
   lub  
   docker container run -p 8080:7777 con  
   by zobaczyć informacje drukowane przez łącznik (następne komendy będą musiały być wpisywane w nowym oknie terminala)
2. Odczekaj 5 sekund
3. Wpisz:  
   docker run -d sham
4. Odczekaj 5 sekund
5. Wpisz:

docker run mongo  
Baza danych nie jest zamykana automatycznie

1. Otwórz nowe okno terminala w tym samym miejscu
2. Jeśli okno z punktu 8 przestało wypisywać nowe informacje wpisz:  
   docker run -d wiki  
   lub  
   docker run wiki  
   by zobaczyć informacje drukowane przez kontener losujący z Wikipedii (następne komendy będą musiały być wpisywane w nowym oknie terminala)
3. Odczekaj 5 sekund
4. Wpisz:  
   python3 ui.py