

# PROJEKT ZESPOŁOWY

PLAYER YOUTUBE Z CHROME EXTENSION

Maciej Bagiński

Maja Bojarska

Damian Koper

Wiktor Pieklik

31 marca 2020

# Spis treści

<b>1</b>	<b>Wymagania</b>	<b>3</b>
1.1	Wymagania funkcjonalne . . . . .	3
1.2	Wymagania niefunkcjonalne . . . . .	4
<b>2</b>	<b>Architektura aplikacji i użyte technologie</b>	<b>5</b>
2.1	Baza danych . . . . .	5
2.2	Architektura i technologie . . . . .	5

# 1 Wymagania

## 1.1 Wymagania funkcjonalne

Użytkownik może:

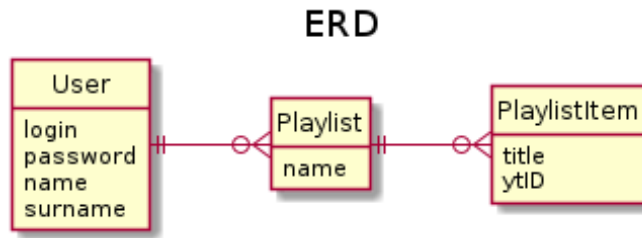
- utworzyć indywidualne konto, identyfikowane poprzez unikalny login, zabezpieczone hasłem,
- zalogować się do swojego konta w aplikacji,
- zapisywać (dodawać) utwory do swojego konta, na wybraną playlistę, za pomocą:
  - przycisku wyświetlonego pod filmem w domenie YouTube,
  - formularza dodawania utworu w aplikacji,
- usuwać utwory z wybranej playlisty,
- utworzyć wiele indywidualnych playlist, identyfikowanych nazwą,
- usunąć wybraną playlistę, wraz z zapisanymi na niej utworami,
- przenieść wybrany utwór do innej playlisty,
- odtwarzać utwory zapisane w wybranej playliste:
  - w kolejności losowej,
  - w kolejności według daty dodania:
    - od najnowszego,
    - od najstarszego,
- ustawić tryb zapętlenia na:
  - “powtarzaj jeden”
  - “powtarzaj playlistę”.

## 1.2 Wymagania niefunkcjonalne

- Aplikacja zapewnia bezpieczeństwo sesji użytkownika za pomocą JWT<sup>1</sup>.
- Aplikacja jest odporna na popularne ataki, typu “SQL Injection”.
- Interfejs graficzny aplikacji działa po stronie przeglądarki użytkownika i komunikuje się z jej serwerem za pomocą RESTful API.
- Architektura aplikacji musi umożliwiać szybką instalację wszystkich jej komponentów, serwisów i jej uruchomienie za pomocą przeglądarki opartej o projekt Chromium.
- Aplikacja funkcjonuje w przeglądarce opartej o Chromium, jako jej rozszerzenie (Chrome extension).

---

<sup>1</sup>JSON Web Token



Rysunek 1: Diagram encji

## 2 Architektura aplikacji i użyte technologie

### 2.1 Baza danych

W bazie danych znajdują się tabele wraz z danymi encji, których istnienie wymuszają przedstawione wymagania. Są nimi:

1. User
2. Playlist
3. PlaylistItem

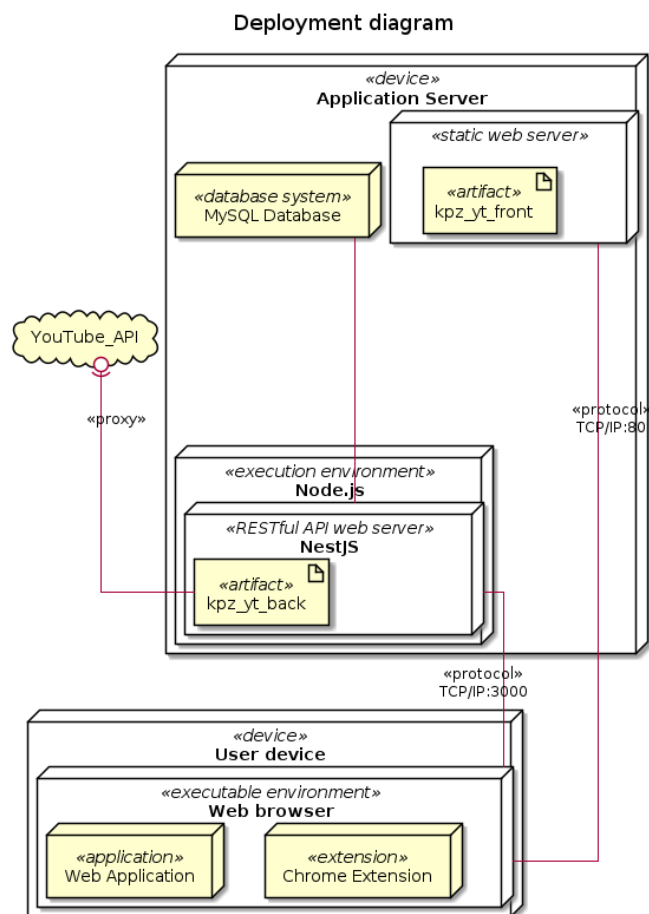
Relacje i pola niebędące definicją struktury bazy przedstawia diagram na rysunku 1. Mała liczba encji wynika z faktu, że potrzebne dane aplikacja pobiera z zewnętrznych źródeł, a główną jej funkcjonalnością jest manipulacja i prezentacja tych danych w interfejsie użytkownika.

### 2.2 Architektura i technologie

Baza danych aplikacji działa w oparciu o DBMS *MariaDB*[1], który bazuje na i jest kompatybilny z popularnym systemem *MySQL*.

Środowiskiem uruchomieniowym dla aplikacji udostępniającej RESTful API jest *NodeJS*[4]. Aplikacja ta oparta jest na frameworku *NestJS*[5], który pozwala na tworzenie wydajnych i skalowalnych aplikacji z użyciem języka *TypeScript*. *NestJS* współpracuje z frameworkiem *TypeORM*[6] obsługującym mapowanie obiektowo-relacyjne.

Frontend aplikacji, przesyłany do użytkownika za pomocą serwera plików statycznych, stworzony został z użyciem języka *TypeScript* i frameworka *VueJS*[7]. Za wygląd aplikacji odpowiada framework *Vuetify*[8], który zaopatruje w gotowy zestaw komponentów, które wyglądem zgodne są ze stylem *Material Design*[9].



**Rysunek 2:** Diagram wdrożenia

Rozszerzenie instalowane w przeglądarce użytkownika również oparte jest na frameworku *VueJS* z wykorzystaniem *Vuetify*. Zawiera ono element popup i obsługuje bezpośrednią interakcję ze strony serwisu YouTube.

Całość uruchamiana jest z użyciem wirtualizacji na poziomie systemu operacyjnego, z użyciem środowiska *Docker*[10]. Pozwala to zminimalizować liczbę kroków potrzebnych do konfiguracji środowiska na różnych maszynach oraz ujednoliciła je na maszynach deweloperów, co skutkuje minimalizacją błędów związanych z różnicą konfiguracji w różnych środowiskach. Diagram wdrożenia nieuwzględniający wirtualizacji przedstawiony jest na rysunku 2.

# Literatura

- [1] MariaDB: <https://mariadb.org/>
- [2] NGINX: <https://www.nginx.com/>
- [3] Apache: <https://httpd.apache.org/>
- [4] NodeJS: <https://nodejs.org/en/>
- [5] NestJS: <https://nestjs.com/>
- [6] TypeORM: <https://typeorm.io/>
- [7] VueJS: <https://vuejs.org/>
- [8] Vuetify: <https://vuetifyjs.com/en/>
- [9] Material Design: <https://material.io/design/>
- [10] Docker: <https://www.docker.com/>
- [11] MVVM: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>