

Tematyką projektu było stworzenie serwisu z artystami i zespołami muzycznymi wraz z ich albumami, piosenkami oraz tekstami tychże piosenek. Dodatkowo chcieliśmy umożliwić użytkownikom trochę więcej rzeczy niż czytanie tekstów piosenek, w tym:

- Dodawanie adnotacji do tekstu, które potem użytkownicy mogą oceniać (+1, -1) i komentować
- Dodawanie recenzji do piosenek, albumów, aliasów, zespołów i artystów z szczególnym wyróżnieniem userów z specjalną flagą krytyka
- Ciekawostki nt. piosenek, albumów, aliasów, zespołów i artystów
- Dostęp do opisów piosenek, albumów, aliasów, zespołów i artystów
- Dostęp do tagów dot. danego artysty, zespołu, albumu, piosenki, aliasu
- Dostęp do członków danego zespołu
- Dostęp do aliasów danego artysty
- Łatwe i intuicyjne chodzenie po grafie związków między artystami, zespołami, albumami, piosenkami i aliasami

Głównym problemem był wytknięty nam problem związany z wieloma nieciekawymi sytuacjami jakie mogą zajść w grafie relacji pomiędzy artystami, zespołami, aliasami, albumami i piosenkami (np. nie chcemy aby *Daria Zawiałow* należała do projektu *A* i zarówno *A* jak i *Daria* byli wpisani jako wykonawcy danej piosenki etc. etc.). Rozwiązaliśmy ten problem poprzez wprowadzenie pięciu warstw grafu, kolejno: **artist**, **alias**, **project**, **album**, **song**.

Okazuje się że teraz jedyne czego chcemy zabronić to: krawędzi wewnątrz danej warstwy, krawędzi w stronę wcześniejszej warstwy oraz tego, że istnieją z danego wierzchołka do jakiegokolwiek innego dwie ścieżki które się różnią co najmniej jedną krawędzią.

Sądzimy, że schemat bazy jest dosyć intuicyjny i nie wymaga dodatkowych komentarzy.

Do użytkowania aplikacji są potrzebne: serwer http (np. apache) oraz php. Po wypakowaniu zawartości pliku .tar na stronę oraz ustawienia poprawnych danych do logowania w pliku base.php (oznaczonych przez !!) wszystko powinno działać poprawnie.

Tworzenie najważniejszych rzeczy zostawiliśmy adminom i służą do tego funkcje:

- `create_entity(artist/alias/project/album/song, nazwa, ilosc wyswietlen, opis, link do zdjęcia)`
- `create_user(username, plaintext password, admin flag, critic flag)`
- `grant_permission(user id, entity id)`

Warto nadmienić że hasła użytkowników są hashowane. Teraz, po udzieleniu danemu userowi uprawnień może on wejść na stronę `ip/login.php`, zalogować się i zacząć dodawać albumy do artystów do których ma uprawnienia, piosenki do albumów do których ma uprawnienia (ważne: założyliśmy, że jeżeli user ma uprawnienia do np. danego artysty to ma też uprawnienia do wszystkiego co jest „pod nim”, czyli np. do wszystkich jego albumów, aliasów, projektów czy też piosenek).

Inne możliwości to:

- Linkowanie artysty do projektu.
- Dodawanie adnotacji do piosenek.
- Dodawanie tagów do danego entity.
- Dodawanie ciekawostek do danego entity.
- Dodawanie recenzji do danego entity (uwaga: każdy może to robić, nie ma żadnych wymagań dot. uprawnień)
- Edytowanie opisu/zdjęcia danego entity.

- Edytowanie swoich adnotacji.

Przykładowym użytkownikiem na stronie jest daria:zawialow oraz jej profil artysty: Daria Zawiałow.

Uważamy, że opisanie wszystkich możliwości aplikacji w tym miejscu jest dosyć męczące i niekoniecznie może odnieść zamierzony zamiar, więc zachęcamy do obejrzenia krótkiego (bo 5-minutowego) pokazu możliwości tegoż programu. <https://www.youtube.com/watch?v=jdBnKvHYRS8>. Warto wspomnieć, że delay przy dodawaniu komentarzy (widoczny w 1:22) był niezamierzony i został usunięty. Reszta założeń dot. bazy i aplikacji nie zmieniła się w stosunku do prezentacji projektu na zajęciach.