

# **INTERNETOWY KATALOG MUZYCZNY**

# SCHEMAT MODELU DANYCH

Ostatnia modyfikacja: 01.05.2020



<b>1</b>	<b>Przedstawienie danych w postaci klas.....</b>	<b>3</b>
<b>2</b>	<b>modelu klas danych i powiązań między nimi.....</b>	<b>3</b>
<b>3</b>	<b>Użytkownicy.....</b>	<b>4</b>
3.1	Klasa User.....	4
3.2	Przechowywane dane .....	4
<b>4</b>	<b>Wykonawcy.....</b>	<b>5</b>
4.1	Klasy.....	5
4.2	Przechowywane dane .....	5
<b>5</b>	<b>Utwory muzyczne .....</b>	<b>6</b>
5.1	Klasy.....	6
5.2	Przechowywane dane .....	6
<b>6</b>	<b>Albumy i recenzje.....</b>	<b>7</b>
6.1	Klasy.....	7
6.2	Przechowywane dane .....	7
<b>7</b>	<b>Listy utworów .....</b>	<b>8</b>
7.1	Klasy.....	8
7.2	Przechowywane dane .....	8

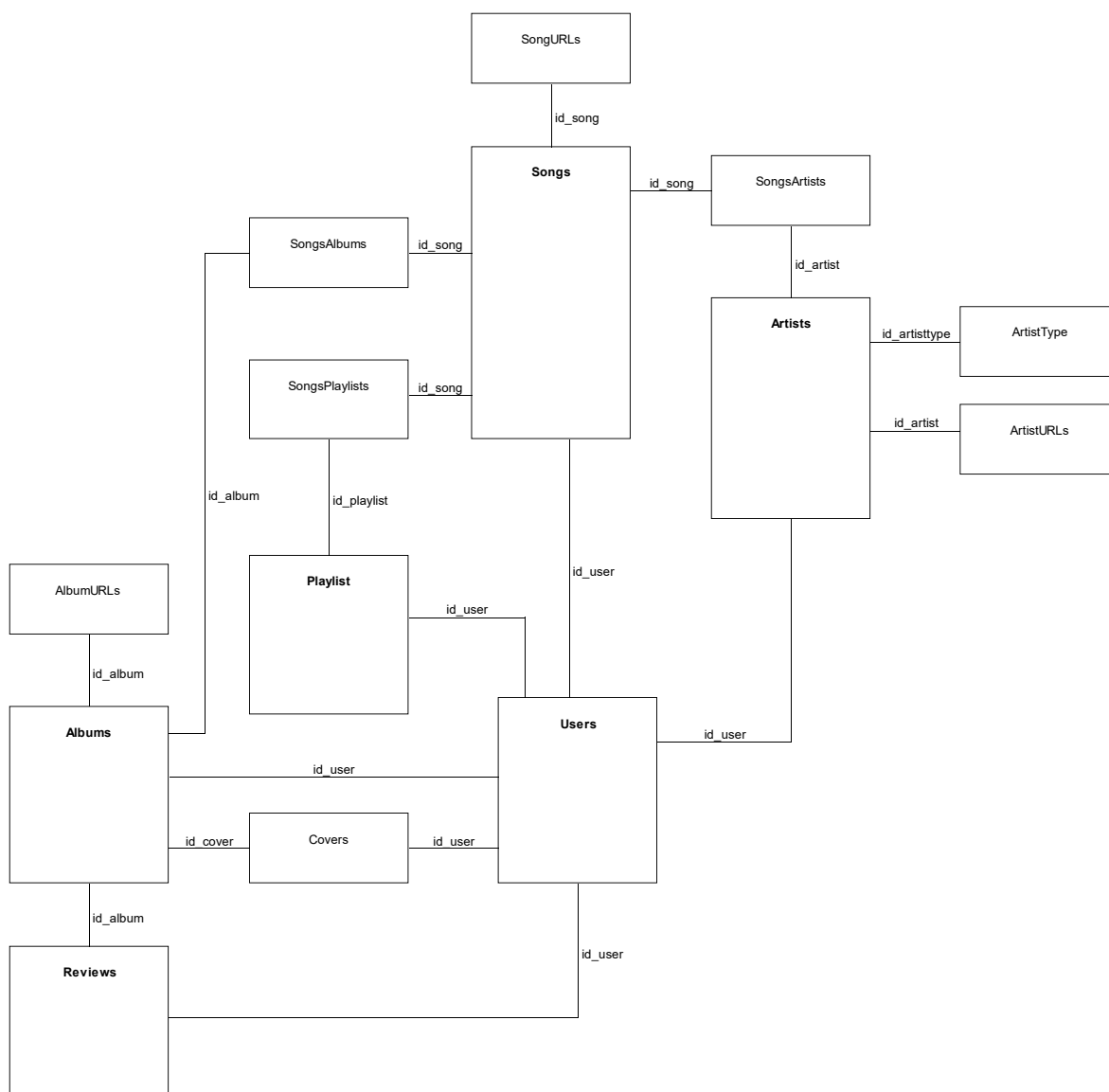


## 1 PRZEDSTAWIENIE DANYCH W POSTACI KLAS

Warstwy serwera aplikacji internetowej wykorzystuje mapowanie obiektowo relacyjne do przechowywania danych w bazie danych. Dokładna struktura bazy danych zależy od samych algorytmów mapowania i w niniejszym dokumencie został przedstawiony model danych na wyższym poziomie abstrakcji, za pomocą klas. Niektóre dodatkowe dane przechowywane w implementacji nie zostały opisane w tym dokumencie.

## 2 MODELU KLAS DANYCH I POWIĄZAŃ MIĘDZY NIMI

Poniższy schemat przedstawia klasy danych w modelu aplikacji oraz połączenia relacyjne między nimi.



### 3 UŻYTKOWNICY

#### 3.1 KLASA USER

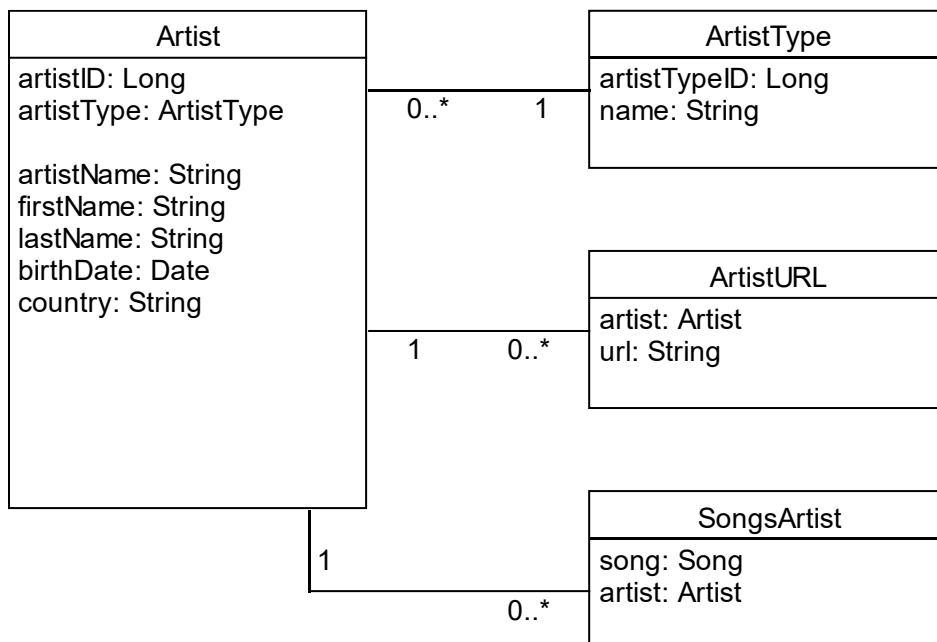
User
userID: Long name: String email: String password: String superUser: Boolean banned: Boolean resetCode: String resetDate: Date

#### 3.2 PRZECHOWYWANE DANE

- userID – identyfikator użytkownika
- name – nazwa użytkownika
- email – adres e-mail
- password – bezpiecznie przechowywane hasło (w postaci hash)
- superUser – czy użytkownik ma uprawnienia administratora
- banned – czy konto jest zablokowane
- resetCode – aktualny kod umożliwiający zmianę hasła
- resetDate – data utworzenia kodu do resetu hasła

## 4 WYKONAWCY

### 4.1 KLASY

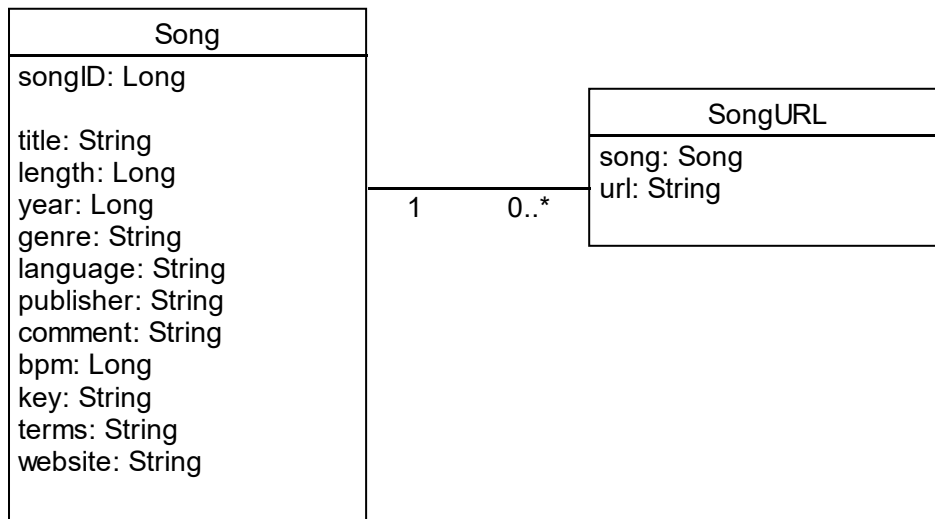


### 4.2 PRZECHOWYWANE DANE

- Klasa **Artist**
  - `artistID` – identyfikator wykonawcy
  - `artistType` – rodzaj działalności muzycznej (np. wokalista)
  - `artistName` – pseudonim artystyczny lub nazwa zespołu
  - `firstName` – imię
  - `lastName` – nazwisko
  - `birthDate` – data założenia zespołu bądź data urodzenia
  - `country` – kraj pochodzenia
  - `created_date` – data utworzenia rekordu
  - `created_id_user` – identyfikator użytkownika, który utworzył rekord
  - `modified_date` – data ostatniej modyfikacji rekordu
  - `modified_id_user` – identyfikator użytkownika, który zmodyfikował rekord
- Klasa **ArtistType**
  - `artistTypeID` – identyfikator
  - `name` – nazwa typu działalności muzycznej (np. wokalista, perkusista)
- Klasa **ArtistURL**
  - `artist` – wykonawca, do którego przypisany jest URL
  - `url` – adres URL np. strony internetowej zespołu muzycznego
- Klasa **SongsArtist**
  - `song` – piosenka
  - `artist` – wykonawca

## 5 UTWORY MUZYCZNE

### 5.1 KLASY

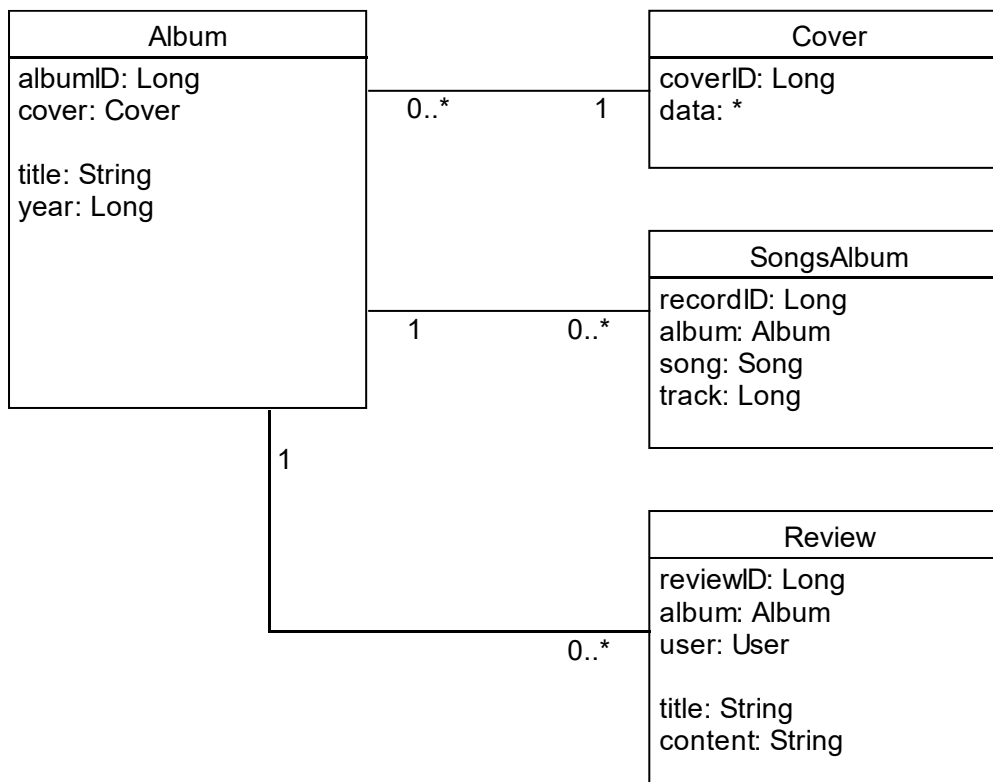


### 5.2 PRZECHOWYWANE DANE

- Klasa **Song** - przechowuje szczegółowe informacje o utworach muzycznych.
- Klasa **SongURL**
  - `song` – piosenka, do której przydzielony jest adres
  - `url` – adres URL powiązany z piosenką np. możliwość jej zakupu

## 6 ALBUMY I RECENZJE

### 6.1 KLASY

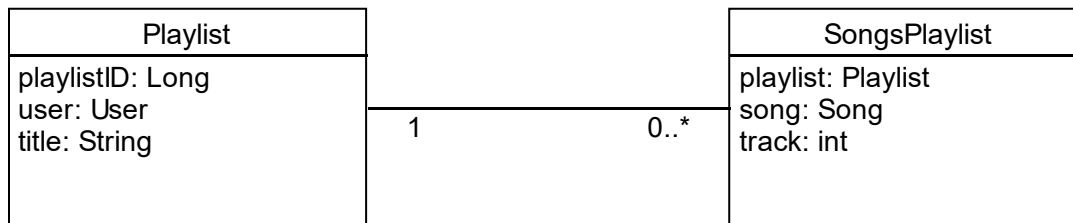


### 6.2 PRZECHOWYWANE DANE

- Klasa Album
  - albumID – identyfikator albumu
  - cover – odnośnik do okładki albumu
  - title – tytuł albumu
  - year – rok wydania albumu muzycznego
- Klasa Cover
  - coverID – identyfikator okładki
  - data – dane obrazu
- Klasa SongsAlbum – wpisy do bazy danych przypisujące piosenki do albumów muzycznych
- Klasa Review
  - reviewID – identyfikator recenzji
  - album – odnośnik do albumu
  - user – użytkownik, który umieścił recenzję
  - title – tytuł wpisu
  - content – zawartość recenzji

## 7 LISTY UTWORÓW

### 7.1 KLASY



### 7.2 PRZECHOWYWANE DANE

- Klasa **Playlist**
  - `playlistID` – identyfikator listy utworów
  - `user` – użytkownik, który utworzył listę
  - `title` – tytuł listy utworów
- Klasa **SongsPlaylist**
  - `playlist` – lista utworów
  - `song` – piosenka
  - `track` – numer elementu na liście