

# 楽曲管理・配信サービスの概念設計

---

計算機科学実験及演習 4: データベース 課題 1 レポート 京都大学工学部情報学科 計算機科学コース 3 年生  
王篤遥 学生番号: 1029332225, 提出日: 2023-10-06

## 楽曲管理・配信サービス「OtoCircle」の概要

音楽クリエイターと作曲系サークルのための楽曲管理・配信サービスである。

### OtoCircle の 3 つのフェーズ

サービスとしての成長を期待して、OtoCircle は音楽配信サービスとして以下の 3 つのフェーズを経て成長することを目標とする。

- 音楽クリエイターと作曲系サークルのための楽曲管理・配信サービス
- インターネットインディーズ音楽の分散型データベース・配信サービス
- すべてのクリエイターとリスナーを繋げる、分散型音楽ストリーミングサービス

これらの目標のうち、本実験では第 1 フェーズとして「作曲系サークル用の楽曲管理・配信サービス」のデータベース設計およびアプリケーション開発を行う。

## OtoCircle の機能

OtoCircle は、以下の機能を持つアプリケーションとなる予定である。

- アカウント機能とサークル機能
  - ユーザーの作成・検索・削除、プロフィールの閲覧・編集
  - サークルの作成・検索・削除、プロフィールの閲覧・編集
  - ユーザーのサークルへの所属関係の設定・解除
- 音楽ストリーミングアプリとしての機能
  - 音楽の検索
  - おすすめや新作のレコメンド
  - 音楽のストリーミング再生
  - 音楽の情報の閲覧
  - (発展) 外部 API を利用して大手ストリーミングサービスの音楽を取得
- 音楽投稿サイト、楽曲データベースとしての機能
  - 音楽の投稿・削除、情報の編集
  - プレイリストへの追加・閲覧・編集・削除
  - 音楽・プレイリストの公開範囲 (自分だけ/サークル/グローバル) 設定
- 音楽を介したユーザー同士のインタラクション機能
  - (発展) お気に入りへの追加・閲覧・編集・削除
  - コメントの追加・閲覧・編集・削除
  - (発展) ユーザー間のフォロー・フォロワー関係
  - (発展) 投票やクイズなど定期的なイベントの開催
- (発展) 通知機能
  - (発展) フォローしたユーザーの更新通知
  - (発展) サークル内ユーザーの更新通知

- (発展) システム通知

(発展) と付したものは本実験で実装する見込みはないが、余裕があれば実装し、より豊富な機能を持った Web アプリケーションを目指す。

## ユーザーのロールについて

OtoCircle の第 1 フェーズとして想定するユーザーは 「**作曲系サークルに所属する音楽クリエイター**」 である。音楽のクリエイターとリスナーという非対称な関係をフラットにして、新しいインタラクションを促進するという観点から、ユーザーをクリエイター・リスナーというロールに分けていない。ユーザーのロールは、サークルやサービス全体の権限管理のために以下の通り用意する。

### システム管理者 (System Administrator)

システム全体に対し制御権限を持つ。管理画面を用意し、アカウントの作成・閲覧・更新・削除 (CRUD)、ロールの設定などの一括管理や、その他諸エンティティの CRUD を一元的に行えるようにする。(発展) システム通知を発行する権限も持つ。

### サークル管理者 (Circle Administrator)

自サークルに対する CRUD を制御する権限を持つ。他ユーザーをサークルに追加・解除することができる。サークル公式としてのプレイリストの CRUD を行うこともできる。

### ユーザー (User)

以上の権限を持たないが、その他のアカウント機能・ストリーミング機能・投稿機能・インタラクション機能を持つ。

## ロールごとの機能について

### システム管理者 (System Administrator)

以下の機能を特権的に持つ。

- アカウント機能とサークル機能
  - **自分以外**の ユーザーの作成・削除、プロフィールの編集
  - **自サークル以外**の サークルの作成・削除、プロフィールの編集
- 音楽ストリーミングアプリとしての機能
  - (発展) 外部 API を利用して大手ストリーミングサービスの音楽を取得
- 音楽投稿サイト、楽曲データベースとしての機能
  - **自分以外**の 音楽の削除、情報の編集
  - **自分以外**の プレイリストへの追加・編集・削除
  - **自分以外**の 音楽・プレイリストの公開範囲設定
- 音楽を介したユーザー同士のインタラクション機能
  - (発展) **自分以外**の お気に入りへの追加・編集・削除
  - **自分以外**の コメントの追加・閲覧・編集・削除
  - (発展) **自分以外**の ユーザー間のフォロー・フォローワー関係
  - (発展) 投票やクイズなど定期的なイベントの開催
- (発展) 通知機能

- (発展) システム通知の発行

## サークル管理者 (Circle Administrator)

以上の機能は持たず、以下の機能の特権的に持つ。

- アカウント機能とサークル機能
  - **自サークルの** サークルの作成・削除、プロフィールの編集
  - ユーザーの **自サークルへの** 所属関係の設定・解除
- 音楽投稿サイト、楽曲データベースとしての機能
  - **自サークルの** プレイリストへの追加・閲覧・編集・削除
  - **自サークルの** 音楽・プレイリストの公開範囲設定

## ユーザー (User)

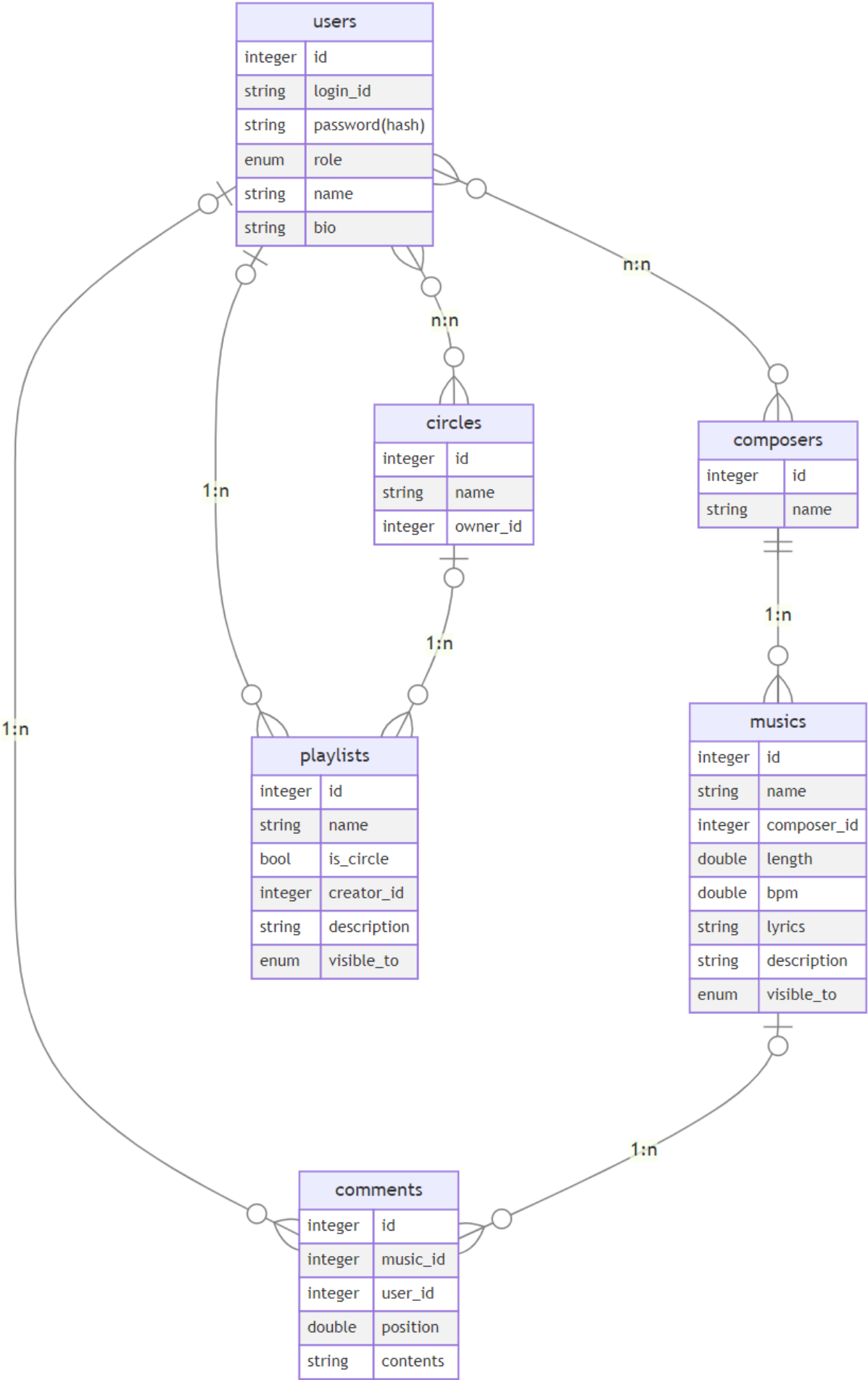
以上の機能は持たず、以下の機能を持つ。

- アカウント機能とサークル機能
  - **自分自身の** ユーザーの作成・削除、プロフィールの編集
  - ユーザーの検索、プロフィールの閲覧
  - サークルの検索、プロフィールの閲覧
  - **自分自身の** ユーザーのサークルへの所属関係の解除
- 音楽ストーリーミングアプリとしての機能
  - 音楽の検索
  - おすすめや新作のレコメンド
  - 音楽のストーリーミング再生
  - 音楽の情報の閲覧
- 音楽投稿サイト、楽曲データベースとしての機能
  - **自分自身の** 音楽の投稿・削除、情報の編集
  - **自分自身の** プレイリストへの追加・編集・削除
  - プレイリストの検索・閲覧 (公開範囲設定が合っている場合)
  - **自分自身の** 音楽・プレイリストの公開範囲設定
- 音楽を介したユーザー同士のインタラクション機能
  - (発展) **自分自身の** お気に入りへの追加・閲覧・編集・削除
  - **自分自身の** コメントの追加・閲覧・編集・削除
  - (発展) **自分自身の** ユーザー間のフォロー・フォローワー関係
- (発展) 通知機能
  - (発展) フォローしたユーザーの更新通知を受け取る
  - (発展) サークル内ユーザーの更新通知を受け取る
  - (発展) システム通知を受け取る

## OtoCircle の技術

ER 図

概念モデル



## ユーザー (users)

- `id`: ユーザーの識別 ID
- `login_id`: ログイン ID
- `password`: ログインパスワード。ハッシュ化されるはず
- `role`: ロール (システム管理者 / サークル管理者 / ユーザー)
- `name`: ニックネーム
- `bio`: 自己紹介

## 作曲者名義 (composers)

ユーザーとは別に持っている。一人のユーザーが複数の名義で曲を投稿したり、合同名義を作ることを許容するため。

- `id`: 作曲者名義の識別 ID
- `name`: 作曲者名義

## サークル (circles)

ユーザーの集まりであり、音楽の公開範囲の一つの尺度となっている。

- `id`: サークルの識別 ID
- `name`: サークル名
- `owner_id`: サークル代表のユーザー ID。サークル管理者が少なくとも 1 人存在することを担保する。

## 音楽 (musics)

- `id`: 音楽の識別 ID
- `name`: 曲名
- `composer_id`: 作曲者の作曲者名義 ID
- `length`: 曲の長さ (後述するコメントの位置に利用)
- `bpm`: 曲の BPM (DJ などのユースケースに便利)
- `lyrics`: 曲の歌詞
- `description`: 曲の概要
- `visible_to`: 公開範囲 (自分のみ / サークル / グローバル)

## プレイリスト (playlists)

音楽をリストで集めたもの。サークルのアルバムとしての扱いや、各個人の愛聴盤をまとめたり、作曲の際のリファレンスをまとめたり、DJ のセットリストとして使える。

- `id`: プレイリストの識別 ID
- `name`: プレイリスト名
- `is_circle`: `true` ならサークル名義でのプレイリスト、`false` なら個人作成のプレイリスト
- `creator_id`: `is_circle == true` ならサークル ID、`is_circle == false` ならユーザー ID
- `description`: プレイリストの説明
- `visible_to`: 公開範囲 (自分のみ / サークル / グローバル)

## コメント (comments)

音楽の特定の位置にコメントをすることができる。ニコニコ動画のコメントのようにリアルタイムで表示できることを目指す。

- `id`: コメントの識別 ID
- `music_id`: コメントの付いている音楽の ID
- `user_id`: コメントをしたユーザーの ID
- `position`: コメントの曲での位置。音楽の `length` より短い
- `contents`: コメントの内容

## 論理モデル

課題 2, 3 にて構築する。