550 Project Outline - Spotify Songs

### **1. Motivation for the idea**

Our website's goal is to provide a genuinely distinctive and customized music discovery experience while overcoming the drawbacks of the widely used music apps available today. Our website enables users to search for music according to their own needs and create their own private playlists customized to their own interests, unlike many other platforms where users are limited to listening to playlists made by others. We understand that everyone has distinct tastes and that a one-size-fits-all playlist strategy frequently leaves consumers unsatisfied. We provide customers the power to create a customized library that precisely suits their tastes and moods by letting them browse music using various features and hear song samples before adding them. Our website aims to give users full control over their music journey, enhancing their discovery process in a way that feels entirely their own.

### **2. List of Features**

1. Homepage Suggestion: where show top popular songs/albums, and a daily song advice
2. Search bar: where the user can search the song’s name/album name/artist name
3. dropdown menu: where the user can find songs based on their genre
4. Another drop-down menu: where users can find songs based on their functions(e.g.dance, etc)
5. A filter: where users can choose how many results they want to see on one page

### **3. List of Features (Optional)**

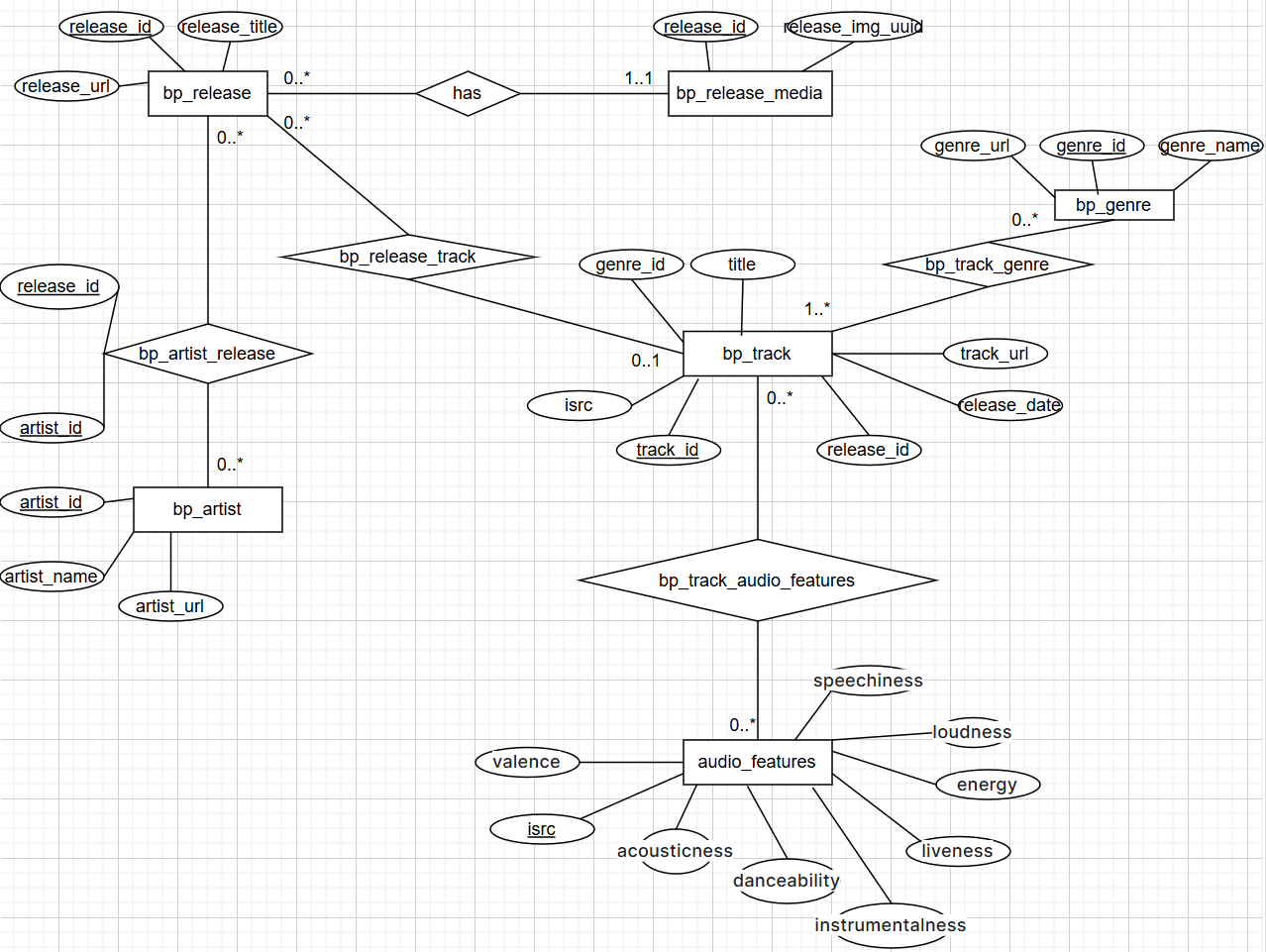
1. A login page with a Google/Facebook account
2. An add button allows the user to add songs to their own playlists.

### **4. List of pages the application**

1. Home Page: user can choose what they want to explore on our website
2. Search Page: user can search for a song based on the album name, track name, release year, artist name, etc., as well as choose how many songs they want to show on a single page
3. Functionality filter page: users choose functionalities of songs they want and filter them.
4. Create my own List page: The user can add their favorite songs to their private lists, as well as rate the song, categorize the song, and add some notes to the song. (Optional)
5. Login Page: users can create accounts and access their own databases, or as guests. (Optional)

### **5. Relational schema**

<https://www.kaggle.com/datasets/mcfurland/10-m-beatport-tracks-spotify-audio-features?resource=download&select=bp_genre.csv>



### **6. SQL DDL\_Ver 1**

**CREATE TABLE bp\_release** (

release\_id INT PRIMARY KEY,

release\_title VARCHAR(255) NOT NULL,

release\_url VARCHAR(255) );

**CREATE TABLE bp\_release\_media** (

release\_id INT PRIMARY KEY,

release\_img\_uuid INT NOT NULL,

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE );

**CREATE TABLE bp\_artist** (

artist\_id INT PRIMARY KEY,

artist\_name VARCHAR(255) NOT NULL,

artist\_url VARCHAR(255));

**CREATE TABLE bp\_artist\_release** (

artist\_id INT,

release\_id INT,

PRIMARY KEY (artist\_id, release\_id),

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE,

FOREIGN KEY (artist\_id) REFERENCES bp\_artist(artist\_id) ON DELETE CASCADE );

**CREATE TABLE bp\_genre** (

genre\_id INT PRIMARY KEY,

genre\_name VARCHAR(255) NOT NULL,

genre\_url VARCHAR(255) );

CREATE TABLE audio\_features (

isrc CHAR(12) PRIMARY KEY,

valence FLOAT,

acousticness FLOAT,

danceability FLOAT,

instrumentalness FLOAT,

liveness FLOAT,

energy FLOAT,

loudness FLOAT,

speechiness FLOAT

);

CREATE TABLE bp\_track (

track\_id INT PRIMARY KEY,

genre\_id INT,

title VARCHAR(255) NOT NULL,

isrc CHAR(12) UNIQUE,

track\_url VARCHAR(255),

release\_date DATE,

release\_id INT,

FOREIGN KEY (genre\_id) REFERENCES bp\_genre(genre\_id) ON DELETE SET NULL,

FOREIGN KEY (isrc) REFERENCES audio\_features(isrc) ON DELETE SET NULL,

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE

);

**CREATE TABLE bp\_track\_genre** (

(track\_id, genre\_id) PRIMARY KEY,

track\_id INT,

genre\_id INT,

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE,

FOREIGN KEY (genre\_id) REFERENCES bp\_genre(genre\_id) ON DELETE CASCADE );

**CREATE TABLE bp\_track\_audio\_features** (

(track\_id, isrc) PRIMARY KEY,

track\_id INT,

isrc CHAR(12),

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE,

FOREIGN KEY (isrc) REFERENCES audio\_features(isrc) ON DELETE CASCADE );

**CREATE TABLE bp\_release\_track** (

(release\_id, track\_id) PRIMARY KEY,

release\_id INT,

track\_id INT,

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE,

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE );

### **7. SQL DDL\_Ver 2** (with correct order of tables)**:**

CREATE TABLE bp\_release (

release\_id INT PRIMARY KEY,

release\_title VARCHAR(255) NOT NULL,

release\_url VARCHAR(255)

);

CREATE TABLE bp\_artist (

artist\_id INT PRIMARY KEY,

artist\_name VARCHAR(255) NOT NULL,

artist\_url VARCHAR(255)

);

CREATE TABLE bp\_genre (

genre\_id INT PRIMARY KEY,

genre\_name VARCHAR(255) NOT NULL,

genre\_url VARCHAR(255)

);

CREATE TABLE audio\_features (

isrc CHAR(12) PRIMARY KEY,

valence FLOAT,

acousticness FLOAT,

danceability FLOAT,

instrumentalness FLOAT,

liveness FLOAT,

energy FLOAT,

loudness FLOAT,

speechiness FLOAT

);

CREATE TABLE bp\_track (

track\_id INT PRIMARY KEY,

genre\_id INT,

title VARCHAR(255) NOT NULL,

isrc CHAR(12) UNIQUE,

track\_url VARCHAR(255),

release\_date DATE,

release\_id INT,

FOREIGN KEY (genre\_id) REFERENCES bp\_genre(genre\_id) ON DELETE SET NULL,

FOREIGN KEY (isrc) REFERENCES audio\_features(isrc) ON DELETE SET NULL,

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE

);

CREATE TABLE bp\_release\_media (

release\_id INT PRIMARY KEY,

release\_img\_uuid INT NOT NULL,

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE

);

CREATE TABLE bp\_artist\_release (

artist\_id INT,

release\_id INT,

PRIMARY KEY (artist\_id, release\_id),

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE,

FOREIGN KEY (artist\_id) REFERENCES bp\_artist(artist\_id) ON DELETE CASCADE

);

CREATE TABLE bp\_track\_genre (

track\_id INT,

genre\_id INT,

PRIMARY KEY (track\_id, genre\_id),

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE,

FOREIGN KEY (genre\_id) REFERENCES bp\_genre(genre\_id) ON DELETE CASCADE

);

CREATE TABLE bp\_track\_audio\_features (

track\_id INT,

isrc CHAR(12),

PRIMARY KEY (track\_id, isrc),

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE,

FOREIGN KEY (isrc) REFERENCES audio\_features(isrc) ON DELETE CASCADE

);

CREATE TABLE bp\_release\_track (

release\_id INT,

track\_id INT,

PRIMARY KEY (release\_id, track\_id),

FOREIGN KEY (release\_id) REFERENCES bp\_release(release\_id) ON DELETE CASCADE,

FOREIGN KEY (track\_id) REFERENCES bp\_track(track\_id) ON DELETE CASCADE

);

### **8. Clean and Pre-process**

Delete the rows with null values; Drop columns we do not need; Entity Resolution (change column names); Replace Categorical Variables with Indicators (genre\_id for genres); Standardize feature audio\_feature(loudness) to [0,1] (from mostly negative to small positive decimal right now)

### **9. List of Technologies**

* Front-end: HTML; CSS; React.js
* Back-end: Node.js; PostgreSQL Database; SQL in PostgreSQL for queries; AWS RDS; Javascript; Github; (> 80% test for backends planned)
* Others for EDA and Integration (Planned):

Colab and Bing search; Integrated with APIs to fetch streaming data

### **10. Each Member’s Role**

Yiting: Front-end development of the website.

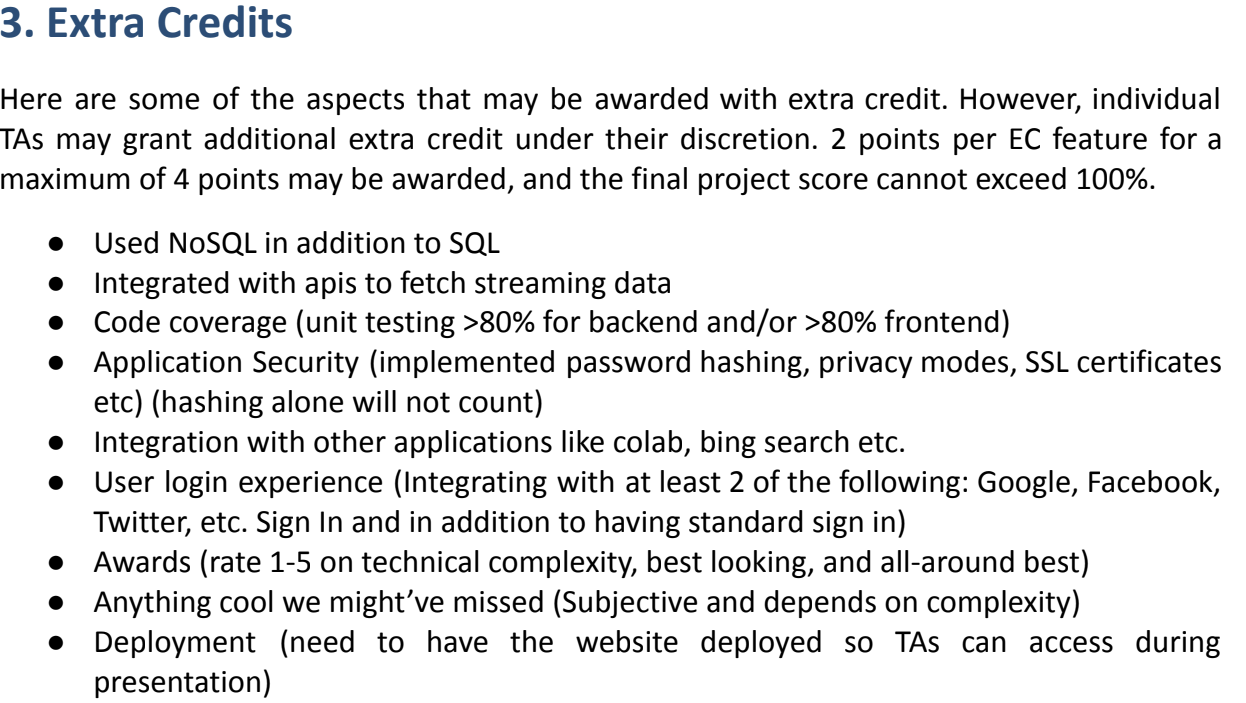
Haorui: Back-end like DB and/or .js files

Jason Pan: Front-end development of the website

Kris Zhang: Back-end like DB and/or .js files

(And we think we will support each other when we need help/discussions)

**Extra Parts - Not Submit, as the 4th Page**



Bp\_release (release\_id, release\_title, release\_url)

Release\_id: primary key

Bp\_release\_media (release\_id, release\_img\_uuid)

Release\_id: primary key

Release\_id: foreign key reference Bp\_release(release\_id)

Bp\_artist\_release (artist\_id, release\_id)

Primary key: (artist\_id, release\_id)

Release\_id: foreign key reference Bp\_release(release\_id)

Artist\_id: foreign key references Bp\_artist(artist\_id)

Bp\_artist (artist\_id, artist\_name, artist\_url)

Artist\_id: primary key

Bp\_track (track\_id, genre\_id, title, isrc, track\_url, release\_date, release\_id)

Track\_id: primary key

Genre\_id: foreign key reference Bp\_genre(genre\_id)

Isrc: foreign key references audio\_features(isrc)

Release\_id: foreign key references Bp\_release(release\_id)

Bp\_genre (genre\_id, genre\_name, genre\_url)

Genre\_id: primary key

Audio\_features (isrc, valence, acousticness, danceability, instrumentalness, liveness, energy, loudness, speechiness)

isrc: primary key

### 

### 

### 