

Database Final Project Requirements

Group Members: Malachi Bieker, Zach Majorowicz, Jacob Sandish

Title: Tapedeck

Description of the project: Tapedeck is like letterboxd for music. Users can archive, rate, and tag music that they've listened to and put that on their profile. Users can also create playlists alone or with friends to curate the specific vibe they are looking for in their next listening session. Tapedeck puts heavy emphasis on attributing moods and feelings with the music.

Requirements:

1. User can add songs to their list
 - Songs are added to the Song table by the add_song function.
2. User can assign rating, notes, or mood to a song in their list
 - Ratings, notes, and moods can be added to the SongRating table by the rate_song function.
3. User can add a rated song to a playlist
 - Supported by the playlist_songs many-to-many junction table. This table uses the create_playlist, add_to_playlist, and remove_song_from_playlist functions.
4. User can collaborate with other users on a playlist
 - Supported by the playlist_manager junction table. The add_collaborator function interacts with this table.
5. User can filter songs based on rating/mood/genre/artist
 - The songs() function creates a query on the join of the Song, Artist, and Album tables. It uses sqlalchemy functions to filter by any of the above criteria.
6. User can edit existing song notes and ratings and adjust them
 - Ratings can be edited through the rate-song() function.
7. User can add custom moods to their mood table.

- The add_mood and delete_mood functions allow users to create custom moods to be added to the Mood Table
8. User can edit their email/profile information.
 - Account creation is supported through the register function while editing is supported through the profile function.

Tables:

1. Table for songs
2. Table for users
3. Table for user song ratings
4. Table for playlist
5. Table for playlist managers
6. Table for moods