COMS 4111 Project 1

MusicParty is a recommendation and media-sharing application that allows users to form groups, also known as parties, to watch videos together. Users can share Spotift songs within parties, listen to them together, and vote for what kinds of videos they want to share with others. The application's main entities are **users**, **parties**, **songs**, and **interests**. On the app, users can create a party by choosing a song to start playing and **tagging** the party with interests that describe what kind of music they want to share (i.e. R&B, Eminem, or study music). A party must have at least one **participant** to be active and continue **playing** songs. The app will recommend a **playlist** of songs that are **generated** using the keywords from a party's interests. Unlike live streams or other music-sharing apps, parties do not have a host or DJ; the songs played will be entirely based on **number of votes** for the songs in the recommended playlist. Data collected from song plays such as the most popular songs associated with certain interests will be pushed to the database to influence future song recommendations for other parties with similar interests. To help grow communities within parties, users can also post comments, which **belong to** the user who authors it and the song playing when the comment is written. With data on how users interact through comments, we can see how users are responding to songs, how they like to discuss music, and how much they are enjoying a truly democratic music-sharing platform.

ER-Diagram Assumptions:

- A "song play" represents a session of time within a party's time interval that a specific song is being played. Comments belong to the "song play" (in addition to a user who authors it) rather than the song or party itself because comment messages can be specific to either the song played or a discussion within the party or both.
- In the Playlist-Contains-Songs relationship set, we chose to represent user votes as an attribute because we are not interested in storing who voted for what songs, only how many votes each song contained in the playlist received.
- In order to use the Spotify API to play songs, users will have to connect their email with a valid Spotify account. There was not a simple way of representing this in the ER Diagram and is more application-specific than part of the conceptual database design.
- A playlist can be generated by multiple interests, one interest, or without any interests. If it is, the playlist will be randomly populated with the top 50 songs. On the other side of the Generates relationship set, an interest is not constrained by the number of playlists it generates; it can generate many different playlists based on the keyword to the best of its ability. In the real-world implementation of the app, this could translate into a "refresh" button that generates a new playlist based on the interests.

Created by Stanley Yu (sy2751) and Yang He (yh2825). Made for COMS 4111 (Fall '18).

Contingency Plan

The contingency plan simplifies the MusicParty platform to only involve **users**, **parties**, and **songs**. Users will simply **participate** in parties and vote for which songs they want to play without specifying interests. As before, parties will only continue to play songs as long as it has at least one participant, and parties will be able to track how long participants stay in the session and what song was playing during that session.