CS374 – Intro to Database Management

Application Development Project

Rubric for Second Deliverable

Group Member #1: Jacob Janzen

Group Member #2: Ethan Nicolay

Group Member #3: Noah Frost

Name	Requirements	Points	Awarded
Description of Application	• An overview of your application • System requirements (e.g. hardware, DBMS, other software) • A detailed description of your application • Are there features that will not be implemented? What are they, and why won't you fulfill them?	10	
Project Management - Schedule	• Detailed schedule of who will do what part of project, by when	5	
Logical Diagram	• Logical diagram in UML or E-R • Discussion of how your data model will satisfy the needs of your application. • Discussion of alternative designs that you did not do (and why)	15	
Queries Required	• Required queries in English (not SQL) • What entities and/or relationships are required for each query? • How will each query satisfy the needs of your application	15	
Grammar, punctuation, syntax, and references	• Follow rules from the Penguin handbook on writing. • References as appropriate (e.g. if you are modeling your application after an existing application, make note of that)	5	

I. Description of Application

a. Application Overview

The application will be a functioning UI for users to access their personal music libraries. It will allow users to add songs they like, search for new ones, create playlists and libraries, and access all of these features through an intuitive UI.

b. System Requirements

This is a full stack application. This application in particular will require access to a DBMS, such as MySQL, a functioning backend program that can access the necessary APIs as well as connect to the DBMS and UI, and a UI that makes the backend features accessible for the user. The UI will need to be hosted on a web hosting service.

c. Detailed Description

- Backend
  - Will be mainly written in Python.
    - If we run into performance issues, we can use embedded C++.
  - Will use MySQL Connector to connect the Python program to our DBMS.
  - Use APIs to connect to the UI and query the web if we choose to add the predictive algorithm described above.
  - Will implement consistent and frequent unit testing to ensure program works for all edge cases.
  - Readable and abundant commenting will allow for future development and understanding of the backend program.
- Frontend
  - Will be created using custom HTML, CSS, and JS
    - Could also look at using WordPress if time constraint becomes an issue down the line.
  - Will allow users to view their personal library songs, playlists, etc.
  - Will allow users to sort their playlists in a variety of ways, available through a simple drop-down menu next to the playlist title.
  - Could possibly create a generative algorithm to suggest music to users adjacent to their established preferences.
  - Will connect to the web using a hosting service.
    - Will most likely use Google Cloud as it is free for us as students.
- DBMS
  - Use MySQL to create a relational database that includes several related tables.
  - We will generate the tables using a Python program, so we don't have to manually create them in MySQL.
    - Users and all their required information.
    - Playlists
      - Playlist Table (<u>playlist_id</u>, user_id(foreign key), name, date_created)

- Songs Table (song_id, title, artist, album, genre, etc.)
- Relationship Table (playlist_id(foreign key), song_id(foreign key))

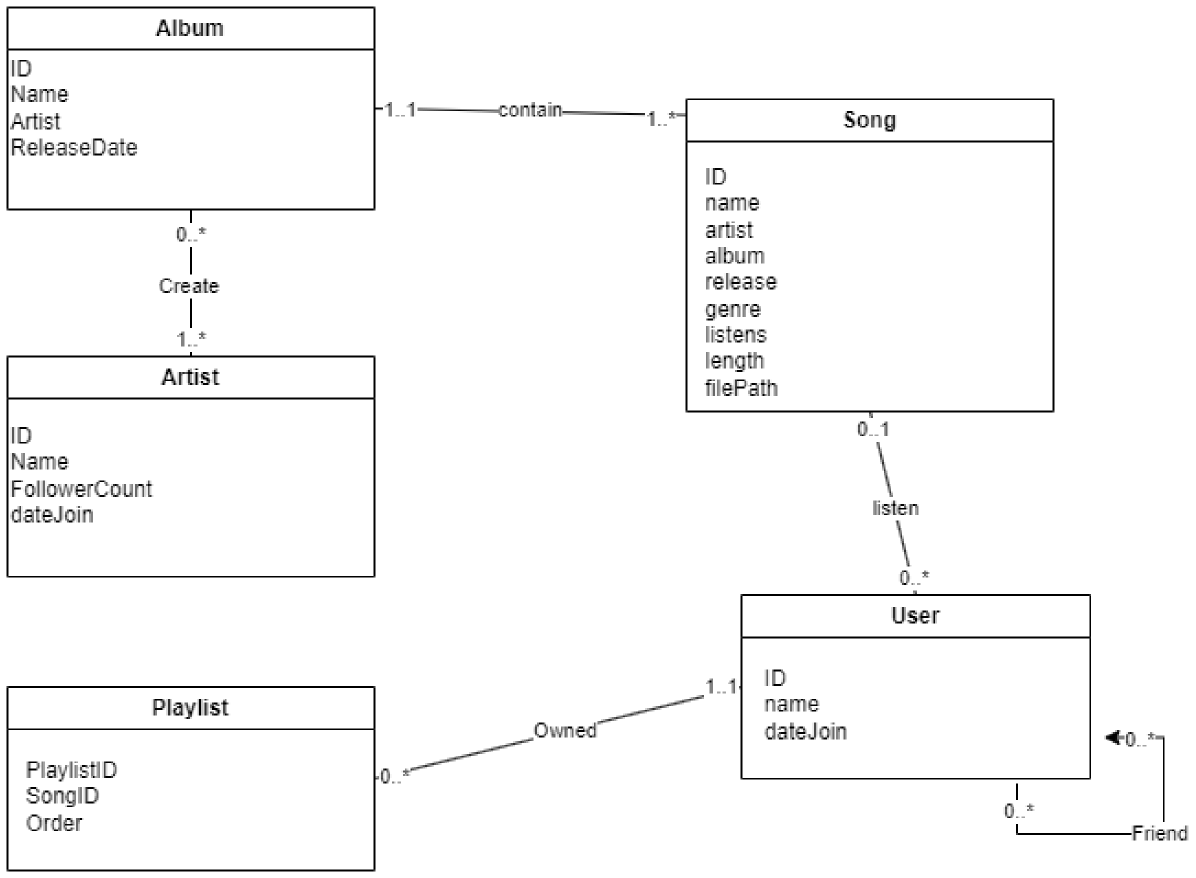
d. Unused Features

At the moment we have no features that we originally planned for that will go unused. This is subject to change as we continue to develop the application and see what is possible given the time constraint. There is the potential that we may add a predictive algorithm that could recommend music similar to the user's established taste. This is the only feature that we are considering cutting if time prohibits us from fully developing it.

II. Project Management Schedule

Final Project Management Schedule	Column1	Column2
Jobs	Team Member	Deadline
Finalize UML/EER	Noah	11/16/23
Write code to generate database tables	Jacob	11/24/23
Write program to query database from IDE	Noah	11/24/23
Design UI layout	Ethan, Jacob	11/28/23
Execute UI layout(WordPress or HTML?)	Ethan, Jacob	11/28/23
Design application backend	Jacob, Noah	11/28/23
Write program to query database from IDE	Noah	11/28/23
Connect backend to web (REST API)	Ethan	12/2/23
Create final presentation	All	12/13/23

III. Logical Diagram



Some of the decisions we made when designing this database included requiring that every song must be in an album, and every playlist has one owner and exactly one owner. Deciding that every song is in an album is pretty arbitrary and could change. We also decided that every playlist has exactly one owner for the sake of simplicity in making certain playlists visible to certain people. By making playlists have exactly one owner, we avoid complications with other users being able to edit or view playlists. Another important decision was to store the actual song itself as a file path as opposed to a sound file stored in the database. This avoids complications of the size of the data, and it will hopefully allow for easier integration with other apps that have audio streaming.

IV. Queries Required

- Get the user's information.
  - This will allow our application to display the users name and their playlists.
- Get the user's playlists.
  - The user table will have a playlist because the playlists will be stored in another table. The user can have multiple playlists, but a playlist can only have one user.
  - This query will allow the application to display the songs in the user's playlist.

- Get the songs in the user's playlists.
  - The song information will be stored in a table separate from playlists. The playlist data will store a list of song IDs. Songs can have multiple playlists and playlists can have multiple songs.
  - This will allow us to list the songs in the user's playlist.
- Get the songs information.
  - Getting the song names, genre, and other information will allow the application to display the songs and all their information.

V. References

No external references were used in the creation of this document.