Outline: Orpheus

Seika Mahmud & John Zontos

URL: https://orpheus-cs340.herokuapp.com

Overview

Problem: There are plenty of social media platforms out there, but none that are tailored specifically to music discovery. This is exactly what “Orpheus” the web application (and maybe one-day mobile app) is set out to accomplish. What makes Orpheus different from other social media platforms is that it is structured in a way that makes sharing your music and discovering new music easy and fun.

Spotify has 217 million monthly active users, but it does not give users the ability to post the music they like and comment on that music. It also lacks the ability to send songs directly to your friends. Our platform would give users that ability. Also, finding new music on traditional streaming services can be cumbersome. In addition to showing you what your friends and favorite artists are listening to, Orpheus will feature a discover feed. This feed will populate with content that users you don’t follow have posted related to the content you have shown interest in. The final feature that makes us unique is the ability to repost and react or collaborate with other users’ posts.

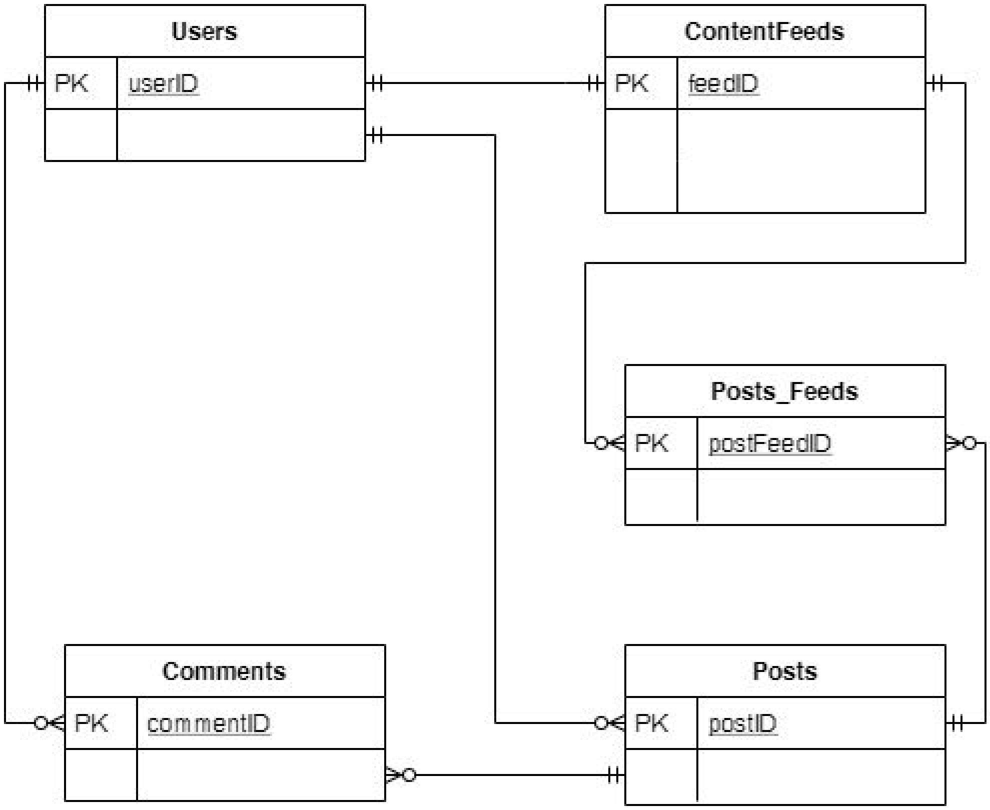
The Database for this app will store content from the users as posts and display them to the user in a “Content Feed.” For each post, we will need to host the media files uploaded by the author of the post. In the database, we will store the file path to the media. In addition, we will give the user the ability to “repost” or embed the content of another post in their post. To do this we made “embedded post” an attribute of Post.

Database Outline

- **Users**: stores information relating to specific user accounts used to populate their profile page.
  - userID: int, auto_increment, unique, not NULL, PK

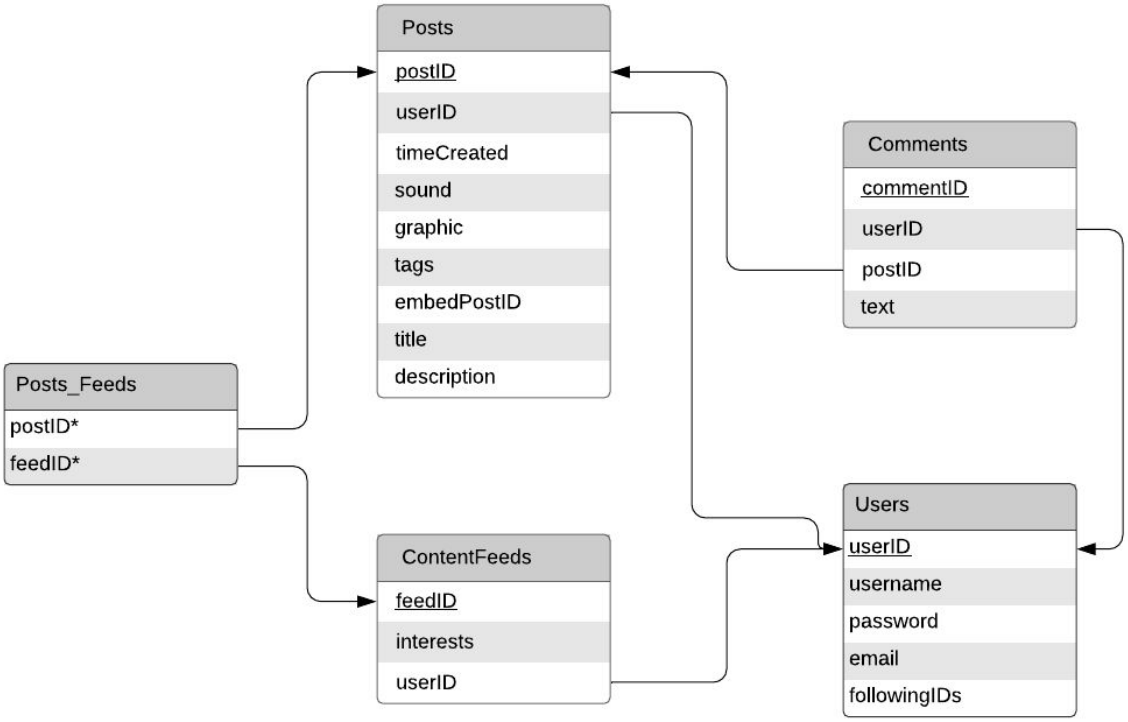
  - username: varchar, unique, not NULL, max 255
  - password: varchar, not NULL, max 255
  - email: varchar, unique, not NULLRelationships:
    - 1:M with Posts, FK stored in Posts
    - 1:1 with ContentFeeds, FK stored in Users
    - M:M with Users
- **Comments**: stores comment text
  - commentID: int, auto_increment, unique, not NULL, PK
  - postID: int, unique, not NULL, FK
  - text: varchar, not Null, max 255
  - Relationships:
    - M:1 with Users, FK stored in Comment
    - M:1 with Posts, FK stored in Comment
- **ContentFeeds**: stores info about the posts displayed to each user
  - feedID: int, unique, not NULL, PK
  - userID: int, unique, not NULL, FK
  - interests: varchar, max 255
  - Relationships:
    - 1:1 with Users, FK stored in Users
    - M:M with Posts
- **Posts**: holds info for each post
  - postID: int, auto_increment, unique, not NULL, PK
  - userID: int, unique, not NULL, FK
  - timeCreated: datetime
  - sound: filepat
  - graphic: filepath
  - tags: varchar, max 255
  - embedPostID: int
  - title: varchar, max 255
  - description: text
  - Relationships:
    - M:1 with Users, FK stored in Posts
    - M:M with ContentFeeds, FK stored in Content Feeds

Entity-Relationship Diagram:



Make Users-Comments relationship optional in both directions.

Schema:


Posts
postID
userID
timeCreated
sound
graphic
tags
embedPostID
title
description
Comments
commentID
userID
postID
text
Posts_Feeds
postID*
feedID*
ContentFeeds
feedID
interests
userID
Users
userID
username
password
email
followingIDs

Fixes based on Feedback from Previous Steps

TA feedback: Good job! Feel free to visit one of the TA office hours if you continue to have problems

Feedback:

Do all the items the team indicated as part of What works, in your estimate work according to the project guide?

DDQ would not load into PHPMYADMIN for me. The DMQ looks to be syntactically and logically correct. The homepage shows up for me, but any of the other pages are not functioning yet.

Was having trouble with the DDM file, it seems like there are syntactical errors. For instance, your SELECT query should be Posts.postID not post.postID when selecting from Posts. Home page does show, if what is shown when clicked, but no other pages work.

The sql dump file looks pretty much correct, there's only one insert that doesn't work. The website just takes us to the raw html, so none of the functionality works.

Syntax looks okay for the most part in the queries but there seem to be some errors. Website displays fine but doesn't seem to have functionality i.e. clicking on any navigation takes you a page showing the raw HTML.

Do you have suggestions to assist on What doesn’t work? or Where/why you are blocked?

It seems like you may need more time to write/modify code from the example node.js project. For example, in users.js, I notice there are function calls for getPeople and getPlanets, which I believe is from the demo code. Not sure of other suggestions at this point besides continue modifying your code from the example project.

Definitely improve your SQL queries and fix what is necessary for them to work. As for the page, I suggest you start by creating tables for each page. That way, you can start to work with your queries and import the data to your page to display and manipulate.

It might be too late, but we used Flask and it was really nice. Everything is written in python, so no JS, asynchronous calls, etc. You'd probably have to rewrite everything tho.

Just seems like there hasn't been much/enough time put into the project. If JS is a bit difficult to work with, Python w/ Flask is very easy to work with the and the provided starter code sets everything up for you. Not sure if you would want to switch over at this point though.

Do the READ functionalities allow user to display data from the database?

No, not currently implemented.

Do the CREATE functionalities allow users to add new entries to the entities and relationships?

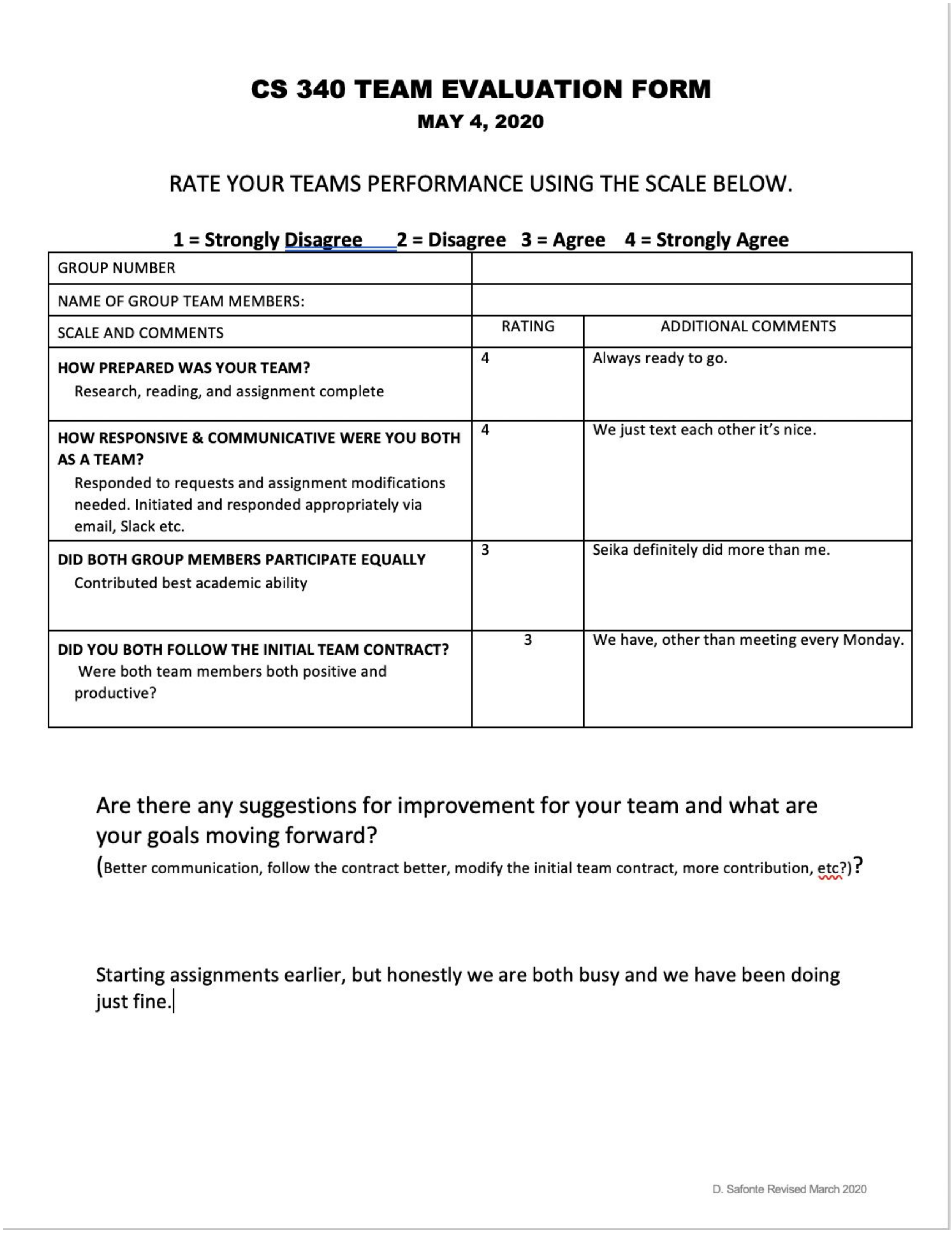
No, not currently implemented.

Does server side code use the DML queries and store the data in the database?

No, server side code is not fully implemented.

FIxes:

A. Re implemented using flask because of feedback recommending it.
B. Seems to be a lot easier but we are now a little bit behind because of this
C. We will continue to work on this this week.



CS 340 TEAM EVALUATION FORM

MAY 4, 2020

RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

1 = Strongly Disagree 2 = Disagree 3 = Agree 4 = Strongly Agree

GROUP NUMBER		
NAME OF GROUP TEAM MEMBERS:		
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	Always ready to go.
HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	We just text each other it's nice.
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY Contributed best academic ability	3	Seika definitely did more than me.
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? Were both team members both positive and productive?	3	We have, other than meeting every Monday.

Are there any suggestions for improvement for your team and what are your goals moving forward?

(Better communication, follow the contract better, modify the initial team contract, more contribution, etc?)?

Starting assignments earlier, but honestly we are both busy and we have been doing just fine.

D. Safonte Revised March 2020