CSCE 310 Final Project: KANM Student Radio

Team:

Charlotte Harrington

Elijah Sanders

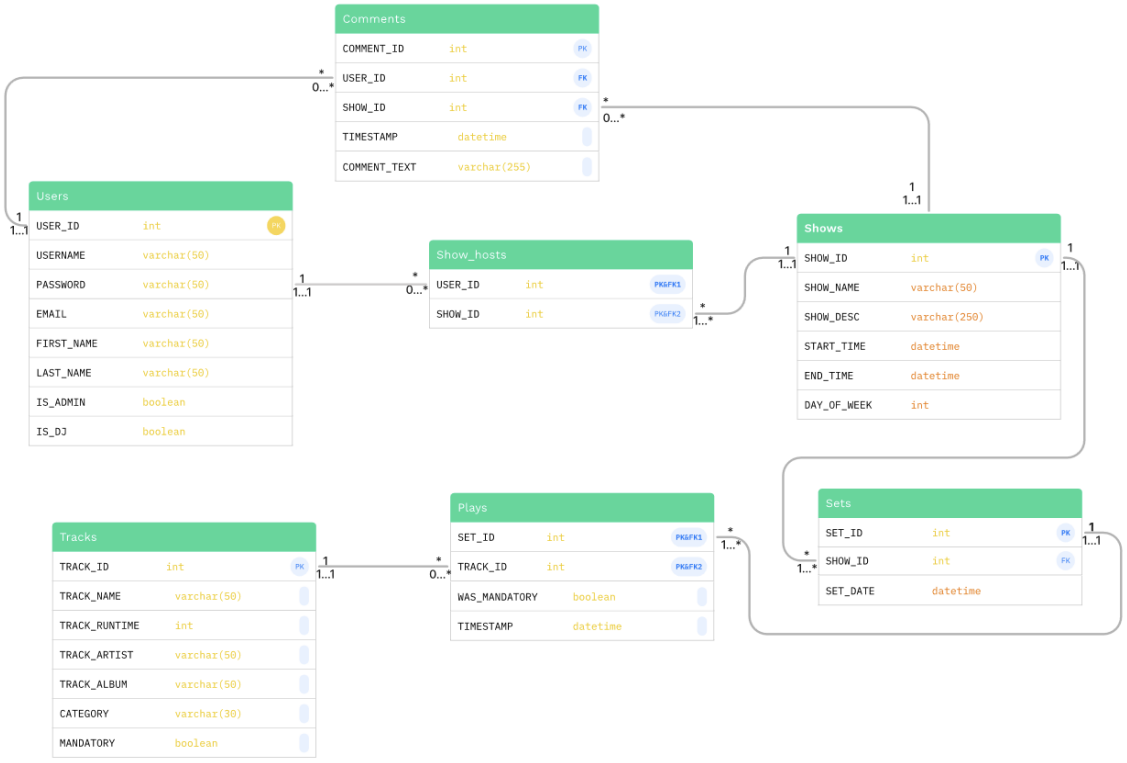
Liam Ramsey

Samuel Torres

Github Link:

https://github.com/samuel-torres-code/kanm-310

Entity Relationship Diagram



File Breakdown

Frontend (kanm310/react/src):

Header.tsx
: Header for the website with login stuff using Shows, Users, and Show_Hosts entities.
Sam

Shows.tsx
: Page for given show using Shows, Users, Show_Hosts, and Comments Entities.
Liam, Sam, and Elijah

UserManager.tsx
: Page showing all users in a table with CRUD functionality. Uses the Shows, Users, and Show_Hosts entities, with the members view.
Charlotte

UserProfile.tsx
: Page showing individual user with ability to edit their own information. Uses User entity.
Charlotte

ShowSchedule.tsx
: Page showing all shows and their attributes. Uses Show entity. Also used to manage DJs on a show and add new shows.
Liam and Elijah

Backend (kanm310/react/php):

createUser.php
: This creates a row in the *users* table, given all user data (save for DJ and Admin bool)
Charlotte

deleteUser.php
: This deletes a user row from the *users* table, given a user_id.
Charlotte

getShows.php
: This handles all the select statements for shows table and the user_shows view.
Liam, Sam, and Elijah

updateShow.php
: This handles all the update statements for shows table.

Sam

deleteShow.php
Removes a show from the shows table.
Elijah

createShowAndShowHost.php
This creates a new show with an initial show host.
Elijah

createShowHost.php
This creates a new show host for a show.
Elijah

deleteShowHost.php
This removes a host from a show.
Elijah

createComment.php
This handles the inserts for the comments table.
Liam

getComments.php
This retrieves all entries from the comments table by show.
Liam

updateComment.php
This handles all updates for the *comment_text* field in the *comments* table.
Liam

deleteComment.php
This handles deletions for comments in the *comments* table.
Liam

getUsers.php
This handles the select statements for the users table and members view.
Charlotte, Sam, and Elijah

updateUser.php
This handles all updates to user data in the *users* table.
Charlotte

Functionality Breakdown

Functionality 1: User Management, Charlotte Harrington

Notes:

Only admins can create or delete users, for security reasons: as a student organization, only officers should be able to add or remove members. Users can be deleted from the 'User Manager' page, accessible only to admins. A user is always created with first name, last name, username, password, and email. DJ and Admin bools are unset by default, but an admin can make another user an admin or a DJ (by associating them with a show and setting the DJ boolean). The three types of users are general member (referred to as a 'user'), admin, and DJ. Users can be both an admin and DJ, or neither.

We removed the UIN data requirement because users were uniquely identifiable by a 'user_id' field.

Users are uniquely identified by their user_id, which is auto-generated when they are inserted into the database. This serves as the primary key for the update and delete operations (update can be done by users, delete by admin), and is not editable by users or admins.

Data:

1. First Name
    a. Admins can create or update a user with a given name.
    b. Users can change their own first name
2. Last Name
    a. Admins can create or update a user with a given name
    b. Users can change their own last name
3. Username
    a. Admins can create or update a user with a given username
    b. Users can change their own username
4. Password
    a. Admins can create or update a user with a given password
    b. Users can change their own passwords
5. DJ Bool
    a. Unset by default, set if an admin makes a user a DJ. A user cannot make themselves a DJ (again, for security reasons), but once a DJ, a user can change their own show data.
6. Admin Bool
    a. Unset by default, set if an admin makes a user an admin. A user cannot make themselves an admin if they are not already.
7. Email
    a. Admins can create or update a user with a given email

b. Users can change their own emails

Views:

- Members
  - Joins the *users* and *shows* tables using the *show_hosts* bridge table, to display show information next to DJs (users who have shows). Users without shows are also shown in this view, with blank columns for show data.

Indexes:

- Show_hosts
  - Two foreign key constraints/indexes (show_id to shows and user_id to user)
  - These fields are used in joins and where clauses.
- users
  - Primary key constraint on user_id

Functionality 2: Scheduling, Elijah Sanders

Data:

1. Date
   a. Admins can create a show for a given date.
   b. DJs can edit the date for their show.
2. Time
   a. Admins can create a show for a given hour and duration.
   b. DJs can edit the hour and duration for their show.
3. Associated DJ
   a. Admins can easily add/remove DJs (show hosts) from shows. A new show is created with the user who created it set as the first show host.
   b. A show is always created with a DJ (the one who created it)
4. Associated Show
   a. Admins can create a show for a given time and date with details that the user sets.
   b. Admins can delete shows.
   c. DJs can edit their own show.

Views:

- User_show
  - Joins the users, show_hosts, and shows tables to get all the show information for DJs

Indexes:

- Show_hosts

- Two foreign key constraints/indexes (show_id to shows and user_id to user)
- These fields are used in joins and where clauses.

Functionality 3: Items (Shows), Samuel Torres

Data:

1. Show Name
    a. Show hosts and admin can change their show name
    b. Everyone else can just view
2. DJ Name(s)
    a. DJ Names can be changes and edited in the user profile
    b. Admins can also change DJ names
    c. Everyone else can just view
    d. DJs assignments for each show can be edited by admins only.
3. Tracklist
    a. Tracklist for different broadcasts can be viewed by everyone
    b. This should not be able to be edited to keep integrity of the stream data
        i. Stream data is pulled externally (from the station itself)
4. Show Picture
    a. Show hosts and Admins can update the Show Picture on the show page
    b. Everyone else can just view

Views:

- Set_info
    - Joins the tracks, plays, sets and shows table to get information about what songs were played on specific broadcasts of a show (these broadcasts are called sets)
- User_show
    - Joins the users, show_hosts, and shows tables to get all the show information for DJs

Indexes:

- Show_hosts
    - Two foreign key constraints/indexes (show_id to shows and user_id to user)
    - These fields are used in joins and where clauses.
- Set
    - One foreign key constraints/indexes (show_id to shows)

    - This field is used in joins and where clauses.
- Plays
    - Two foreign key constraints/indexes (set_id to sets and track_id to tracks)
    - These fields are used in joins.

Functionality 4: Comment/Status, Liam Ramsey

Data:

1. Comment Author
    a. Displays the author of the comment by username
    b. Cannot be changed
2. Timestamp
    a. Displays when the original comment was made (date and time)
    b. Cannot be changed
3. Comment Text
    a. Displays the content of the comment
    b. Comment authors can edit or delete their own comments
    c. Admins can edit or delete all comments
4. Show commented on
    a. Comments of a show display under the show information
    b. Different comments display under different shows

Views:

- User_comments
    - Joins the *comments* and *users* tables in order to link usernames to comments and allow for proper comment editing permissions.

Indexes:

- Comments
    - Two foreign key constraints/indexes (show_id to shows and user_id to user)
    - These fields are used in joins and where clauses.