Caroline Silva, Trevor Asbery, Arjun Bulusu,

Phillip Addei, Fahad Ahmed, Lu Wu

CSCI 201

Project Proposal

New music is released every day, its difficult to keep up with the endlessly evolving music industry. Each person has different music preferences – pop, rap, indie, rock n' roll, classical, this list is never-ending. Some people religiously follow certain artists and spend hours curating playlists, while others steal playlists and listen to any genre. By utilizing *Newsic*, a social media platform designed to bring new music to light, finding new music will never be an issue.

Newsic will be a website where users can create a personalized account linked with their music streaming platform of choice (Spotify/Apple Music/Sound Cloud). Users will be able to curate posts that include songs, albums, or artists that they want to share with their followers. This allows friends to share and discover new music. Users can leave a like and/or comment on posts to build connections and boost engagement. Each post will include a 10-30 second sound clip of the song to provide users with a preview, which they can then add to their playlists on their desired streaming platform. Based on which posts the user engages with there will be a page for song and artist recommendations, to enable the user to find new music based on their interests. Each user's profile will display their posts, most listened to songs and genres, and followers/following.

We wanted to create *Newsic* in order to help others discover new music and curate playlists easily. *Newsic* will bring together people with all tastes of music and create connections beyond each chorus.

Features:

-profile where you Spotify or apple music is linked

-feed where you can see posts (music) of who you follow

 -post individual songs or music or artists

-like and comments on posts

-preview of song

-add what people post to your playlists

-recommendation page

-profile with 3 most listened to songs

Caroline Silva, Trevor Asbery, Arjun Bulusu,

Phillip Addei, Fahad Ahmed, Lu Wu

CSCI 201

Newsic
Technical Specifications

Newsics goal is to make it easier to find new music and share it with friends to make it easier than ever to find new music, artists, and overall expand your music horizon. Through utilizng social media and connecting Newsic with your music streaming platform discovering new meusic will be revolutionized.

Features:

1. Landing/Home Page
   - Nav bar(fixed on all pages):
     - Logo which when clicked will redirect user to home page
     - Login & Register button which will redirect user to Login & Register page
     - Create post button (appear only once user is logged in)
     - Logout button (appear only once user is logged in)
   - Search bar:
     - Search other users to find their profiles, when clicking on a users name will redirect them to the Deatils/Profile page of that user
     - Search results will display list of profiles that match the search key word and display users profile photo, name, and top music genres
   - Current Top Songs:
     - Current top songs displaying image of songs cover, name, and artists
     - Plus button to add song to users library (if not logged in will direct user to login page)
2. Login & Register Page
   - Login box: asks for users username and password, will check this with the Users table in mySQL, if exists will log the user in and redirect them to the Feed Page, if the user does not have an account will remain on the page and display an error message
   - Users will also have the option to login with Spotify
   - Register box: asks for username, password, and then to confirm the password. Will store this information in mySQL users table.
     - Questions about users music preferences (Radio buttons)

       - Comes up with profiles that match the search results

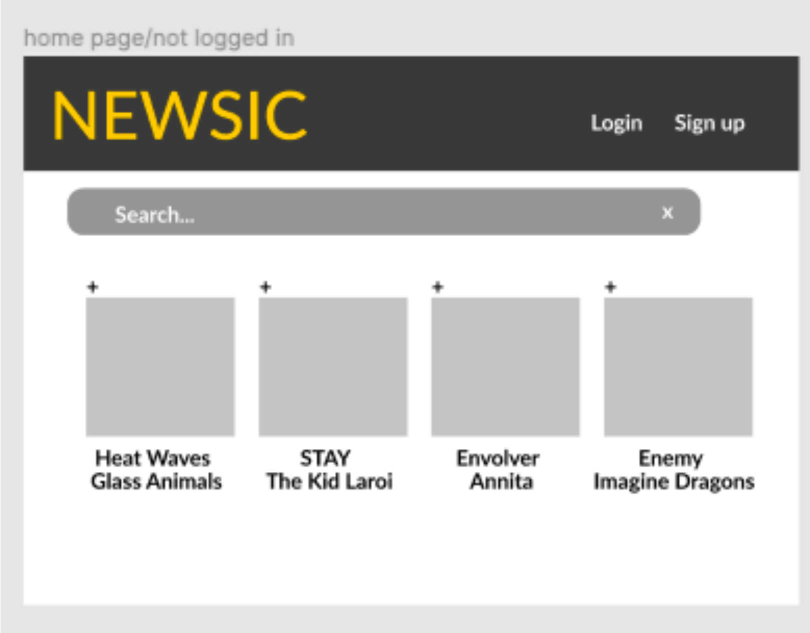
3. Feed Page
   - This page will contain the users friends posts, the user can scroll and view all posts in a most recently posted order
   - Each post will contain:

- User who posted (if clicked on will take to users profile page)
- Image of song or album cover
- Song or album name
- Artists
- Message by user
- Reaction buttons (thumbs up, thumbs down)
- 10-30 second preview of the song (play button)
- Pluss button to add to library

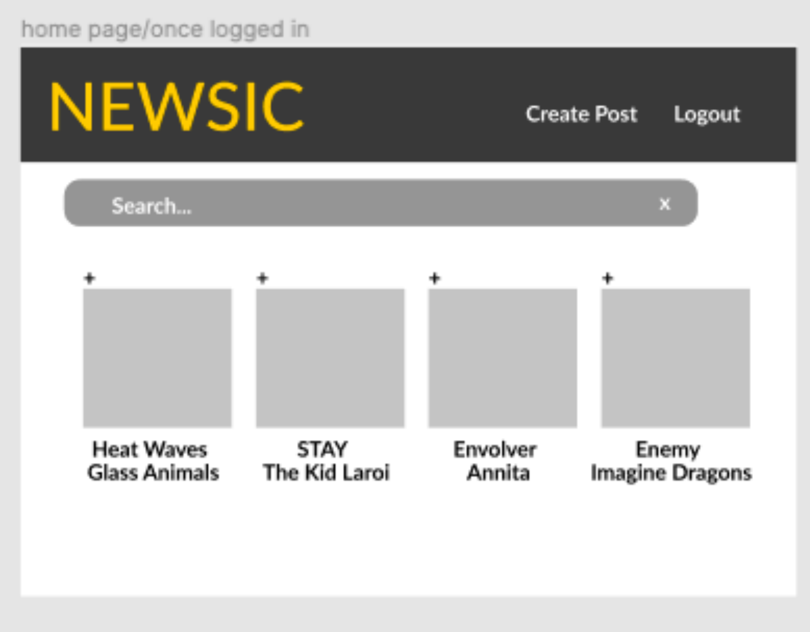
4. Details/Profile Page
    - This page is directed to from the search page or when a user clicks on the profile from their feed page
    - Contains users info:
        - User name
        - Full name
        - Number of followers
        - Follow button
        - Info about their music preferences (gathered from when they registered)
        - All the users posts
5. Create Post Page
    - Directed to from nav bar create post button
    - Connects to spotify through API and user will choose a song
    - Once a song is chosen will direct back to this page and user can add a message and post it
    - Once posted will be redirected to feed page

Detailed Design Document

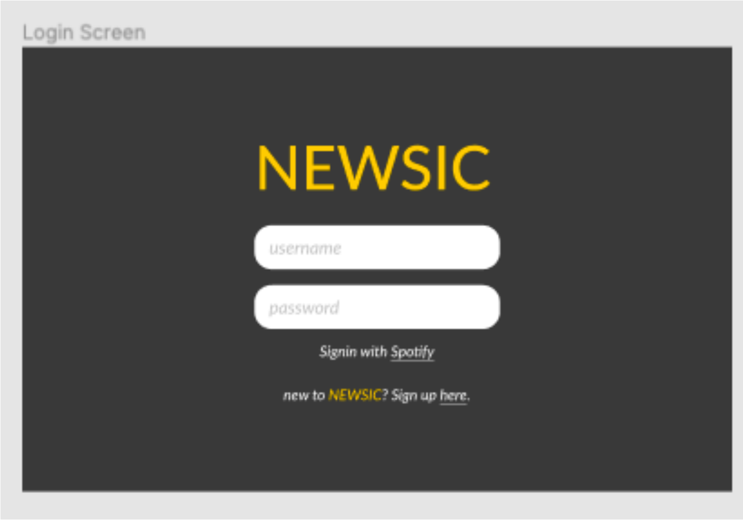
1. GUI Mockup & Technical Specification
    a. Home page
        i. Guest

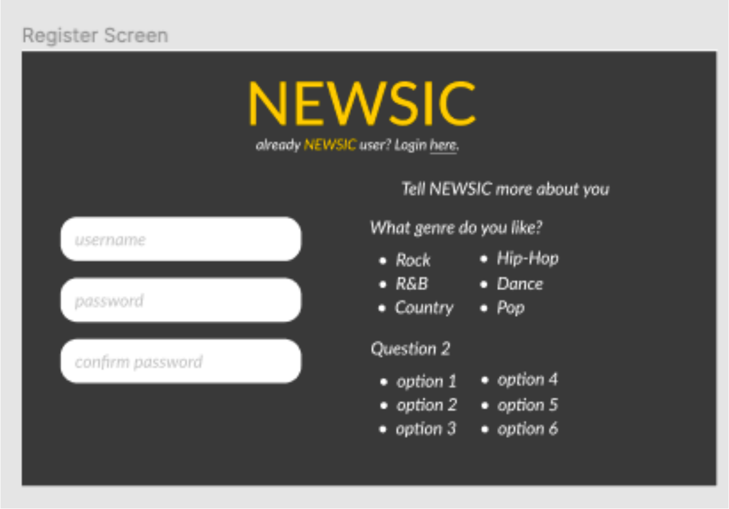


        ii. Logged in

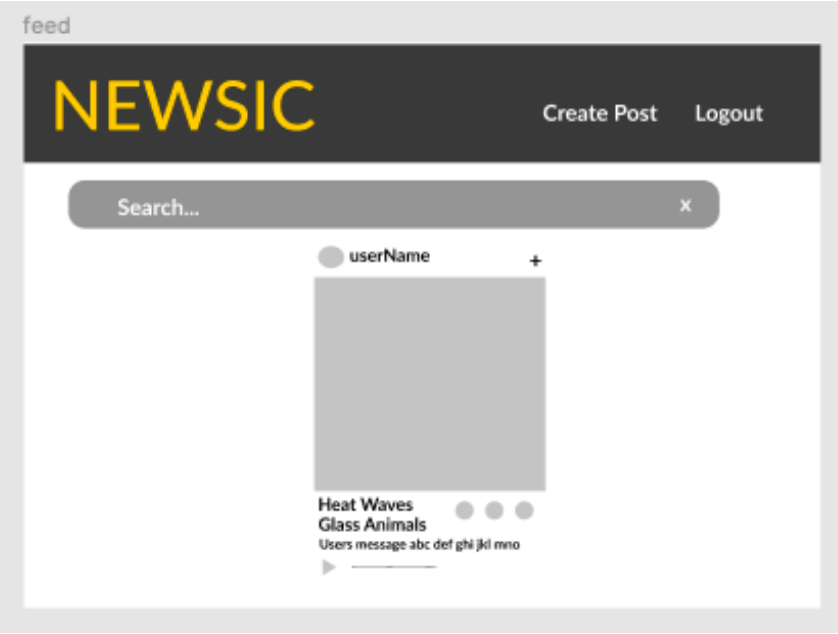


b. Login/Register:

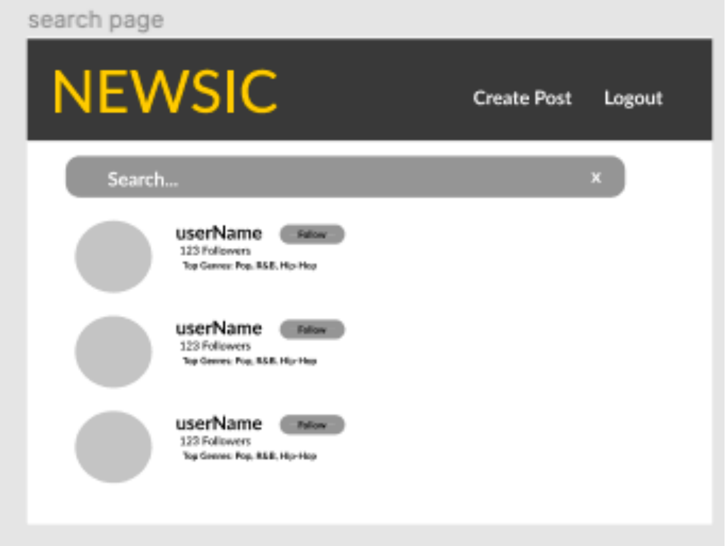


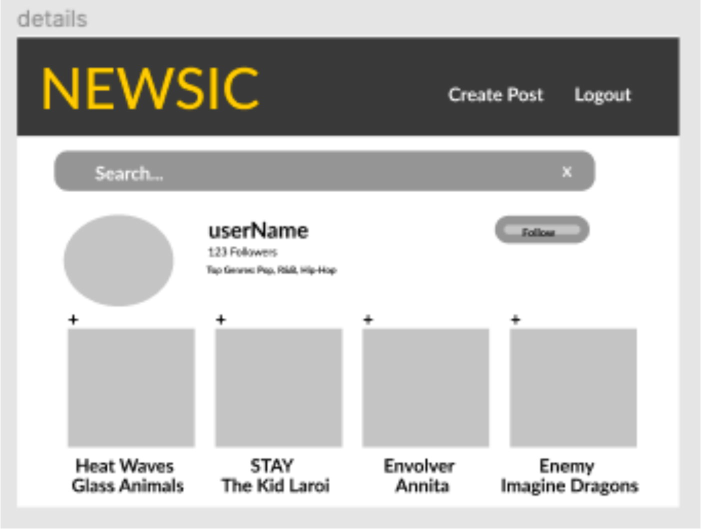


c. Feed

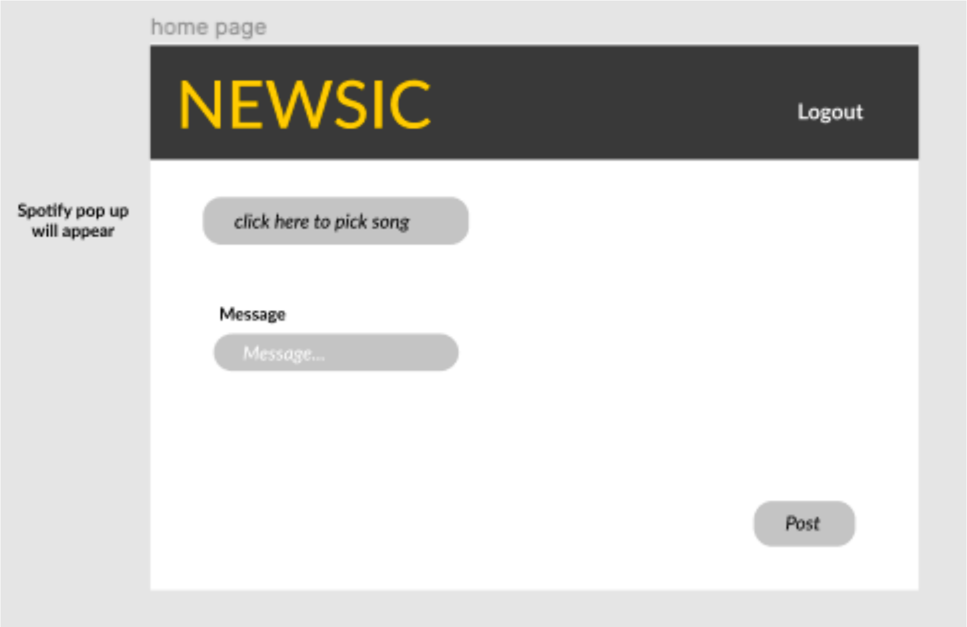


d. Search/Details

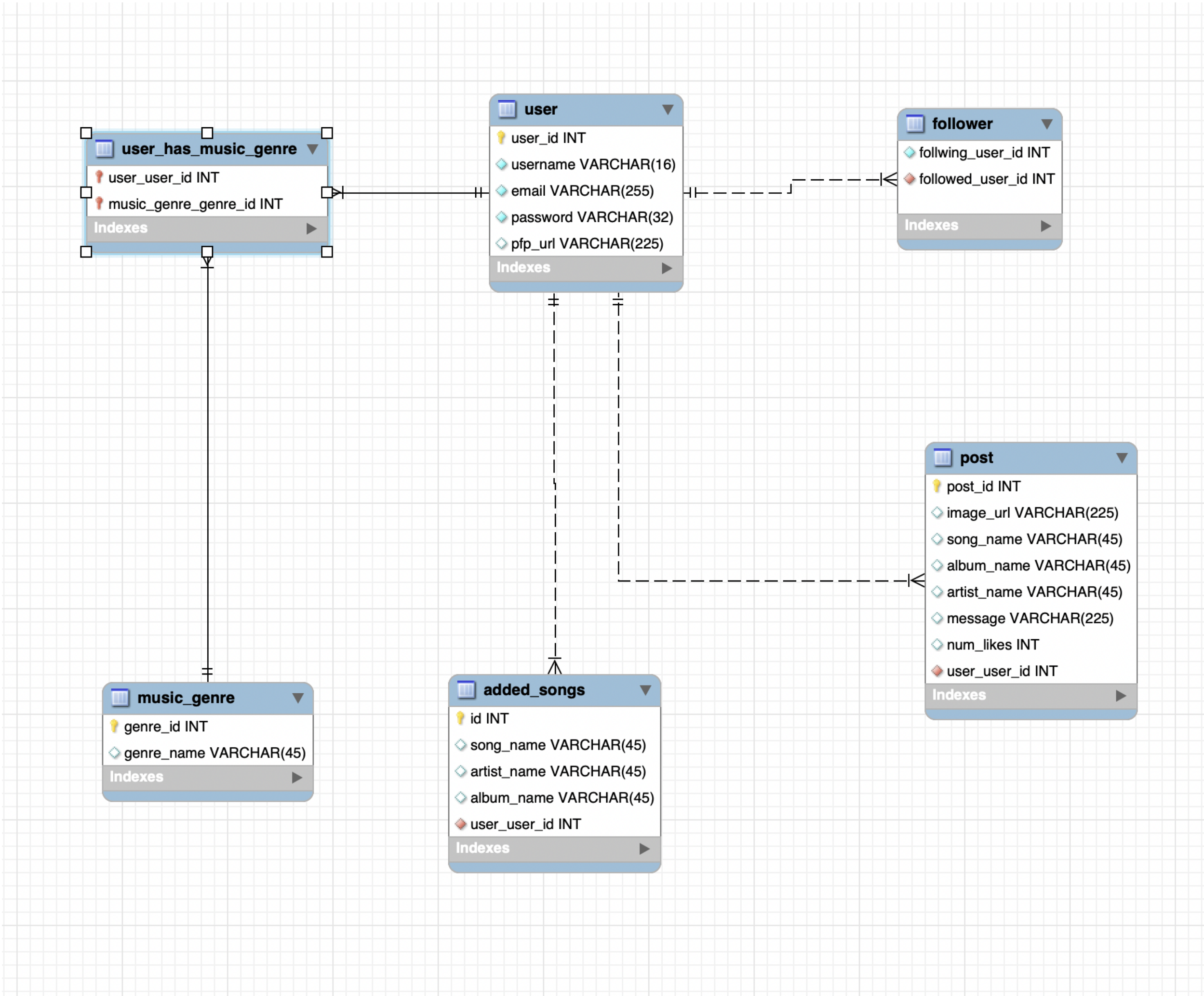


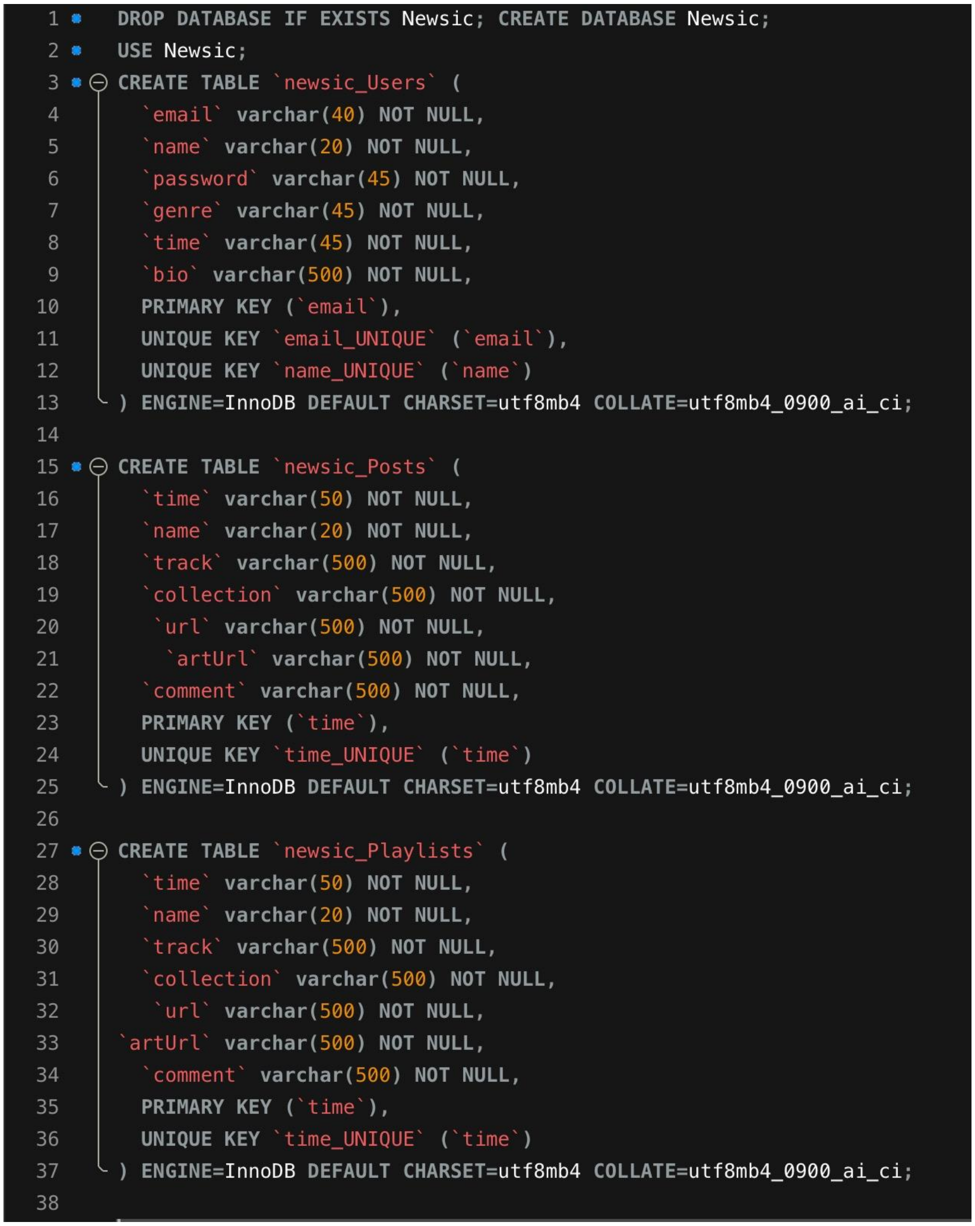


e. Create post



2. Database Schemas





```sql
DROP DATABASE IF EXISTS Newsic; CREATE DATABASE Newsic;
USE Newsic;
CREATE TABLE `newsic_Users` (
  `email` varchar(40) NOT NULL,
  `name` varchar(20) NOT NULL,
  `password` varchar(45) NOT NULL,
  `genre` varchar(45) NOT NULL,
  `time` varchar(45) NOT NULL,
  `bio` varchar(500) NOT NULL,
  PRIMARY KEY (`email`),
  UNIQUE KEY `email_UNIQUE` (`email`),
  UNIQUE KEY `name_UNIQUE` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `newsic_Posts` (
  `time` varchar(50) NOT NULL,
  `name` varchar(20) NOT NULL,
  `track` varchar(500) NOT NULL,
  `collection` varchar(500) NOT NULL,
   `url` varchar(500) NOT NULL,
    `artUrl` varchar(500) NOT NULL,
  `comment` varchar(500) NOT NULL,
  PRIMARY KEY (`time`),
  UNIQUE KEY `time_UNIQUE` (`time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `newsic_Playlists` (
  `time` varchar(50) NOT NULL,
  `name` varchar(20) NOT NULL,
  `track` varchar(500) NOT NULL,
  `collection` varchar(500) NOT NULL,
   `url` varchar(500) NOT NULL,
`artUrl` varchar(500) NOT NULL,
  `comment` varchar(500) NOT NULL,
  PRIMARY KEY (`time`),
  UNIQUE KEY `time_UNIQUE` (`time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

3. Software Requirements

   Programing languages:

   - Front end: HTML, CSS, Bootstrap
   - Back end: Java, JavaScript
   - Databases: mySQL

   Software:

   - GitHub
   - IDE: latest version of Eclipse
   - Server: Apache TomCat version 9

4. Class Diagram

   Data is stored and retrieved from the database; therefore, our project does not need a complex inheritance hierarchy.

(1). User registration/login/logout (Login, Register and Logout Servlets)
(2). Post creation (Create Post Servlet)
(3). Post sorting (Feed Servlet)
(4). Add songs to library/ Remove songs from library
(5). Managing Spotify API (login with Spotify, spotify play back, metadata, user tate)
(6). Notification
(see list of servlets below)

5. Further Details
    a. MultiThreading: We will create a thread pool to notify the users when someone they follow creates a new post
    b. Servlets
        i. Login Servlet
            - Track cookie (what user is signed in)
            - Checks the entered email is entered in the appropriate pattern (using Regex)
            - Checks if username and password that is entered into login form already exists in the userInfo table in SQL
        ii. Register Servlet
            - Process info imputed into fom and validate it then add it to userInfo SQL table
            - Validation:
                - Check if email already exists
                - Check if passwords match
                - Check email entered is correct format (using Regex)
            - Track cookie
            - Will sign in user once register is complete
        iii. Spotify Login Sevlet
            - Connects to Spotify API to enable users to login with their spotify accounts
        iv. Logout Servlet
            - Removes cookie and loges out of spotidy account
        v. Search Servlet
            - Get information corresponding to search key word from SQl table containing user info (using Spotify user taste feature)
        vi. Feed Servlet
            - Gets info of users following and displays their post recent posts
            - Get following user info from SQL table and then gets their corresponding information from posts table
            - Using Spotify API: 30 second song playback, access song info using metadata feature)
        vii. Create Post Servlet
            - Take information from the form that the user inputs post information into and adds that info a a post SQL table to store info
        viii. Notification Servlet

- Responsible for multithreading, check if a new post has been created and will notify users that a new post has been posted

ix. Add to Songs

- Able to add songs to your own playlists
- Use the Playlist feature from Spotify API

Testing Documentation

Assuming home.jsp is deployed

Test Step	Input	Expected Output
Click on signup	Name, email, password, confirm password, bio (all strings), genre, time (both radio buttons)	Redirect back to home page with name of user on top of the st
Click on create post	Artist name or album (String)	Displays lists of songs by artist
Type in search bar under album cover	Post comment (string)	Show your post with comment below
Type in search bar (in home screen/ after creating post)	Name of user (string)	See all posts made by specified user
Select album cover of posts made by user	Click on image and then click add song to playlist	Will go to playlist page and display playlist
Click on playlist button	Nothing, just click on text	Will go to playlist page and display playlist
Click logout	Nothing, just click on text	Will take you to home page with all posts
After creating post	N/A	Multi-threaded notification will appear