

Software Requirements Specification (SRS) Document

Orchestra

12/10/2024

Version 2

Albert Lusk, Gabe Roberts, Matthew
Ingallinera

<https://github.com/SomeStars/Orchestra>

1. Project General Description

- a. The primary goal of our application is to promote collaboration between UNCG students, specifically in regards to music production. Our application will allow students to showcase and sell music they produce. And allow others on campus to contact them for the purpose of collaborations. There are a multitude of student run projects and clubs that incorporate music. Such as: Carmichael Studios, members of the UNCG game dev club, and other music producers on campus that are interested in collaborations.

2. Product Features

- a. The Orchestra application is designed to make collaboration between musicians easier by making it simple to share and view files that each user has created. Below are the key features that define the Orchestra application:

Profile creation: Each user can create their own profile with music that they have created or music that they have found from other users on Orchestra.

Content uploads: Orchestra will allow creators to upload their original music so that others may use it in their content or so that users may listen for their own enjoyment.

Content sharing: Orchestra users will also be able to share music that they have created or found on the platform with others.

Profile following: Orchestra users will also be able to follow their favorite creators on the platform so that they can see whenever that creator uploads something to their profile.

3. Functional Requirements

- a. FR0: Orchestra will allow the admin to view all reports.
- b. FR1: Orchestra will allow the admin to suspend a user.
- c. FR2: Orchestra will allow the admin to delete users.
- d. FR3: Orchestra will allow the admin to delete reports of users.
- e. FR4: Orchestra will allow the admin to manage the database.
- f. FR5: Orchestra will allow the provider to post music files.
- g. FR6: Orchestra will allow the provider to edit their personal music files.
- h. FR7: Orchestra will allow the provider to edit their personal profile page.
- i. FR8: Orchestra will allow the provider to reply to comments on their profile.
- j. FR9: Orchestra will allow the provider to create an account.
- k. FR10: Orchestra will allow the user to leave a review.
- l. FR11: Orchestra will allow the user to like or dislike a song.
- m. FR12: Orchestra will allow the user to subscribe to a music provider.
- n. FR13: Orchestra will allow the user to create their User account.
- o. FR14: Orchestra will allow the admin, provider, and user to log in.
- p. FR 15: Orchestra will allow the admin, provider, and user to log out.
- q. FR16: Orchestra will allow the admin, provider, and user to view content on the platform.

4. Non-functional Requirements

- a. NFR1: Orchestra will not allow the user or provider access to the admin panel.
- b. NFR2: Orchestra will allow admin access to the admin panel.

5. Scenarios

a. Admin - *Albert Lusk*

i. **View all reports**

1. **Initial Assumption:** The admin is able to click on the “view all reports” button in the admin panel and it will take them to a page displaying all reported songs/reviews while also being able to take action.
2. **Normal:** Admin will be looking at a list of reports and can decide what actions to take regarding the report.
3. **What Can Go Wrong:** The admin could misclick a button, so there will be a confirmation screen to confirm the actions.
4. **Other Activities:** Admin will be able to see actions happening as they ignore/take action on reports.
5. **System State on Completion:** Database will be updated with suspensions/deleted reports and flagged reports will no longer be flagged.

ii. **Delete report**

1. **Initial Assumption:** Admin will be able to delete a report from public view if it is unhelpful or inappropriate.
2. **Normal:** Admin can click on the “delete” button while looking over a reported review if they deem it necessary.
3. **What Can Go Wrong:** Admin could accidentally click the ‘ignore’ button which would keep the review up and delete the report. There will be a confirmation screen for this.
4. **Other Activities:** Admin can look into the entire post containing the reported review to look at all facts.
5. **System State on Completion:** Report will be deleted from public view and from the system database entirely.

iii. **Suspend user**

1. **Initial Assumption:** Users will have their account suspended until further notice.
2. **Normal:** Administrators will have the ability to suspend users by clicking on the suspend button if they did something to violate the web application’s terms of use.
3. **What Can Go Wrong:** Admin might accidentally suspend a user they didn’t mean to. There will be a confirmation screen to avoid this.
4. **Other Activities:** When enough time has passed, admins can unsuspend users to give them permissions back.

- 5. System State on Completion:** Users that have been suspended will not be able to log into the system, but their accounts will still exist.

iv. Delete user

- 1. Initial Assumption:** User will be deleted from the system database and unable to login or reset password.
- 2. Normal:** Administrators will have the ability to delete users by clicking on the delete user button if they did something to violate the web application's terms of use.
- 3. What Can Go Wrong:** Admin might accidentally click to delete a user they didn't mean to. There will be a confirmation screen to avoid this.
- 4. Other Activities:** Admins will not see a deleted user on the list of users when they want to manage all users.
- 5. System State on Completion:** Deleted users will be gone from the system database and will not be able to login with old username and password.

v. Manage website

- 1. Initial Assumption:** Website changes will be applied after clicking the save changes button.
- 2. Normal:** Administrators will have the ability to change the theme of the website, including colors and fonts, whenever they choose to do so.
- 3. What Can Go Wrong:** The administrator changes may not go through to the website.
- 4. Other Activities:** Administrators will be able to choose from a color wheel or something similar that lets them choose from all ranges of colors.
- 5. System State on Completion:** Website theme and fonts will be updated according to the administrator who changed them.

b. Provider - *Gabe Roberts*

i. **Personalize Profile**

1. Initial Assumption:

The music producer will be able to customize their profile by adding details such as a bio, profile picture, social media links, and contact information.

2. Normal:

The producer clicks on the "edit profile" button and can update their bio, upload a profile picture, add social links, and customize other personal details.

3. What Can Go Wrong:

The producer might upload the wrong picture or save incorrect details. There will be a confirmation screen before saving changes to verify the inputs.

4. Other Activities:

The producer can preview the profile changes before they are publicly visible.

5. System State on Completion:

The producer's profile will be updated with the new details, and the changes will be reflected when viewed by others.

ii. **Upload Music**

1. Initial Assumption:

The producer will be able to upload their tracks in supported formats (.MP3, .WAV) to the platform.

2. Normal:

The producer clicks on the "upload music" button, selects files, adds metadata (title, genre, tags), and submits the tracks.

3. What Can Go Wrong:

The producer might accidentally upload the wrong file or metadata, or encounter an error during the upload. A confirmation screen will verify the details before submission.

4. Other Activities:

The producer can choose whether the track is immediately published or saved as a draft for future edits.

5. System State on Completion:

The music track will be uploaded, and the producer will be able to manage it (edit details, delete, or publish). The track will be added to the database and made available for listeners or buyers.

iii. **Set Up Payment Methods**

1. Initial Assumption:
The producer will be able to link their payment methods (PayPal, bank account) to receive payments for music sales.
2. Normal:
The producer accesses the "payment settings" section, selects a payment method, and enters relevant details.
3. What Can Go Wrong:
The producer might enter incorrect payment details or choose an unsupported payment method. A confirmation screen will validate the details before saving.
4. Other Activities:
The producer can view and update their payment methods at any time.
5. System State on Completion:
The payment method will be successfully set up, and payments from music sales will be routed to the producer's chosen account.

iv. **View Purchase History**

1. Initial Assumption:
The producer will be able to see a detailed log of all their music sales, including buyer details and payment status.
2. Normal:
The producer navigates to the "purchase history" section, where all purchases are listed with dates, track names, and payment amounts.
3. What Can Go Wrong:
The purchase history might not load correctly due to system issues, or certain purchases may not appear. There will be an option to filter or refresh the data.
4. Other Activities:
The producer can download reports of their purchase history for their records.
5. System State on Completion:
The purchase history will be displayed, and the producer can review their earnings and transaction records.

v. **View Music Statistics**

1. Initial Assumption:
The producer will be able to see how their music is performing, including playtime and downloads

2. Normal:

The producer clicks on the "music statistics" section and views data on individual tracks, such as total plays, downloads, and listener geographic locations.

3. What Can Go Wrong:

The statistics might not be updated in real-time or may fail to load. There will be an option to refresh the data.

4. Other Activities:

The producer can filter statistics by date range or track.

5. System State on Completion:

The producer will be able to view up to date performance data for their tracks, helping them make informed decisions about their music strategy.

c. User - *Matthew Ingallinera*

I. Leave review

- A. Initial assumption: User will be able to leave a review of a song that they have found on the platform.
- B. Normal: The user will be able to leave a review of song by clicking on the review button and typing their review in the text box.
- C. What can go wrong: A user may want to edit their review of the song in which case they edit the review late by by clicking on edit review and editing the the text box once more.
- D. Other Activities: edit review button for updating previous reviews made by that users.
- E. System state on completion: The users review will be saved in a review section for that song.

II. Like/dislike song

- A. Initial assumption: The user will be able to like or dislike a song that they have found on the platform
- B. Normal: The users will be able to like or dislike the song by clicking on the like or dislike button next to the song that they have listened to.
- C. What can go wrong: A user might accidentally click on the wrong button (clicked like instead of dislike or vice versa) in which case they can click the other button to change to the correct state.
- D. Other Activities: The users can see all of the songs that they have liked through a liked songs list or something similar on their profile.
- E. System state on completion: The users input of a like or dislike of song will be saved and the user can see all of the songs they have liked.

III. Subscribe to a music provider:

- A. Initial assumption: The user will be able to click on a music providers profile and subscribe to them to get notified when that provider releases new music.
- B. Normal: The user will be able to subscribe to the provider by clicking on the subscribe button on the providers profile.
- C. What can go wrong: A user might accidentally click the subscribe button for a creator they don't want to be

subscribed to in which case they will also be able to unsubscribe from that provider.

- D. Other Activities: Users can view all of the providers that they have subscribed to as well be able to unsubscribe from any of those providers.
- E. System state on completion: Users subscription to a provider will be saved and viewable from a list of subscriptions.

IV. Create User Account:

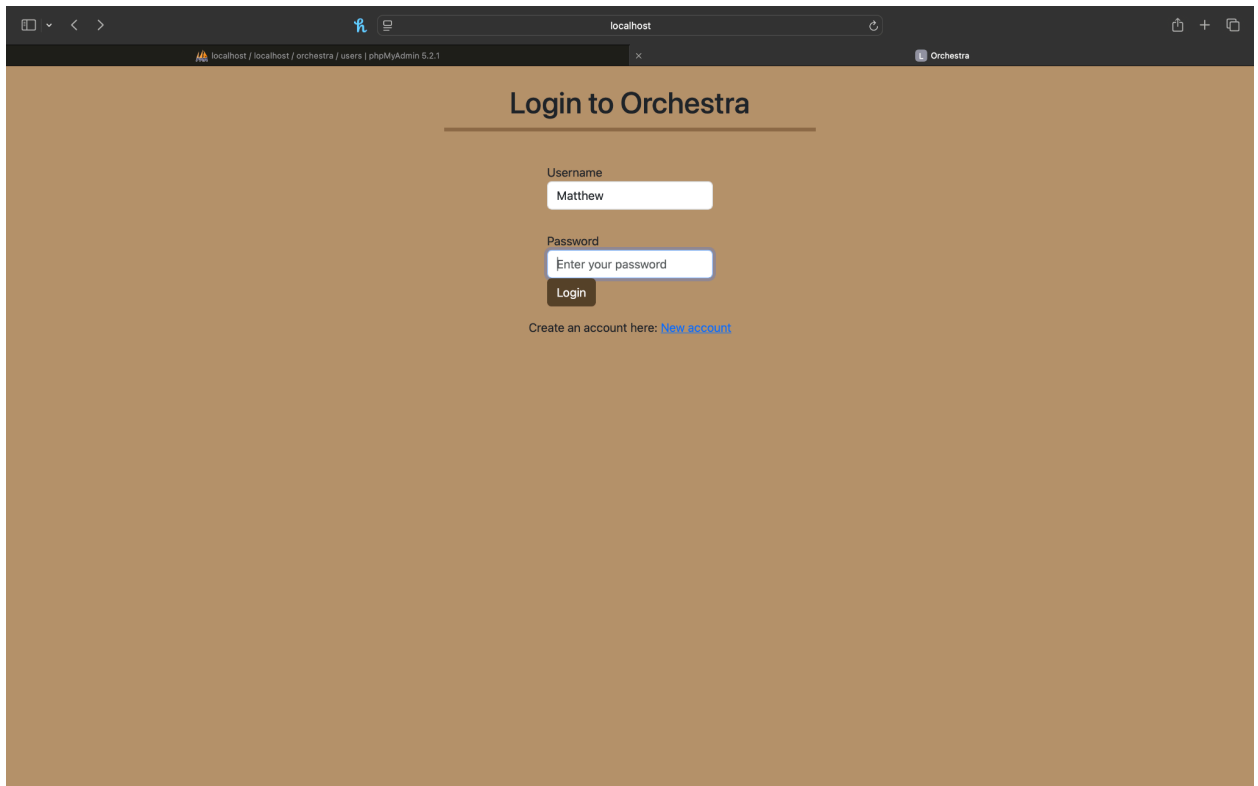
- A. Initial assumption: Users can create their account when they first access the site.
- B. Normal: Users create their account by inputting a username and password and clicking the create account button.
- C. What can go wrong: A user might forget their username or password needed to access the account so they can use the forgot username and forgot password buttons to reset these.
- D. Other Activities: Users can click forgot username or forgot password to reset forgotten accounts.
- E. System state on completion: The username and password will be saved and associated with an account. Now the user will be able to access the site.

V. View Content:

- A. Initial assumption: The user can view all content on the platform on the main page of the site.
- B. Normal: All content is visible on the main page of the site so that the user can listen to everything that has been uploaded.
- C. What can go wrong: Users may not have refreshed the site since more content was added so some content may not be visible until the user has refreshed.
- D. Other Activities: Users can view all of the content by one creator by looking at that individual creator's page.
- E. System state on completion:

6. Scenarios with Screenshots

1. Login functionality:



A screenshot of a web browser displaying the 'Login to Orchestra' page. The browser's address bar shows 'localhost' and the page title is 'Orchestra'. The login form is centered on a brown background. It includes a 'Username' field with the text 'Matthew', a 'Password' field with the placeholder 'Enter your password', and a 'Login' button. Below the form, there is a link that says 'Create an account here: [New account](#)'.

localhost

localhost / localhost / orchestra / users | phpMyAdmin 5.2.1

Login to Orchestra

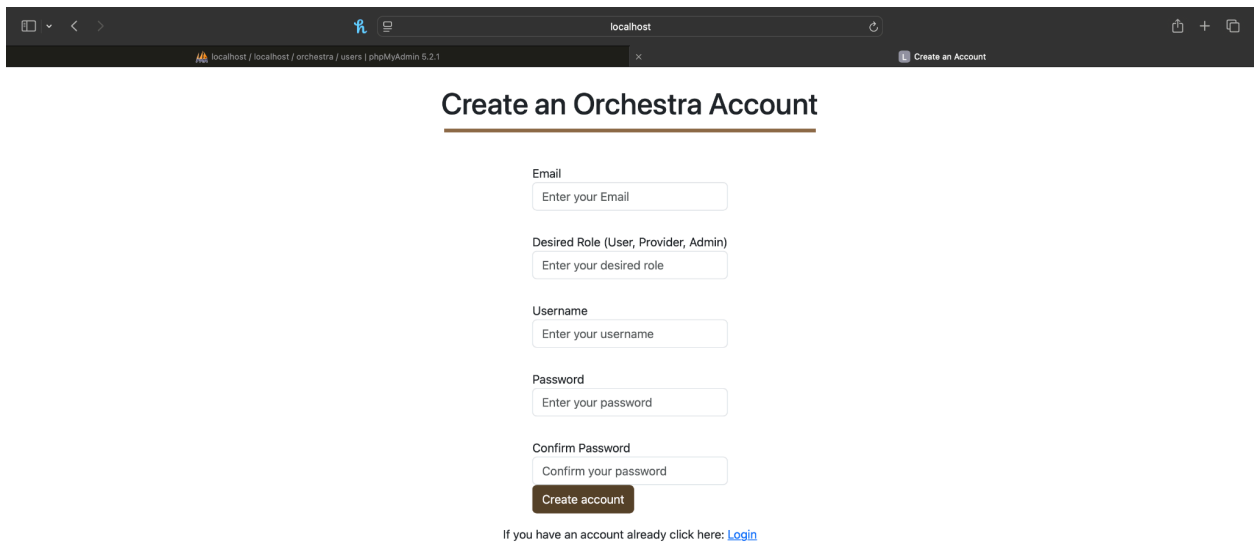
Username
Matthew

Password
Enter your password

Login

Create an account here: [New account](#)

2. Create Account functionality:



A screenshot of a web browser displaying the 'Create an Orchestra Account' page. The browser's address bar shows 'localhost' and the page title is 'Create an Account'. The form is centered on a white background. It includes fields for 'Email' (placeholder: 'Enter your Email'), 'Desired Role (User, Provider, Admin)' (placeholder: 'Enter your desired role'), 'Username' (placeholder: 'Enter your username'), 'Password' (placeholder: 'Enter your password'), and 'Confirm Password' (placeholder: 'Confirm your password'). A 'Create account' button is at the bottom of the form. Below the form, there is a link that says 'If you have an account already click here: [Login](#)'.

localhost

localhost / localhost / orchestra / users | phpMyAdmin 5.2.1

Create an Orchestra Account

Email
Enter your Email

Desired Role (User, Provider, Admin)
Enter your desired role

Username
Enter your username

Password
Enter your password

Confirm Password
Confirm your password

Create account

If you have an account already click here: [Login](#)

Design Document

Orchestra

12/11/2024

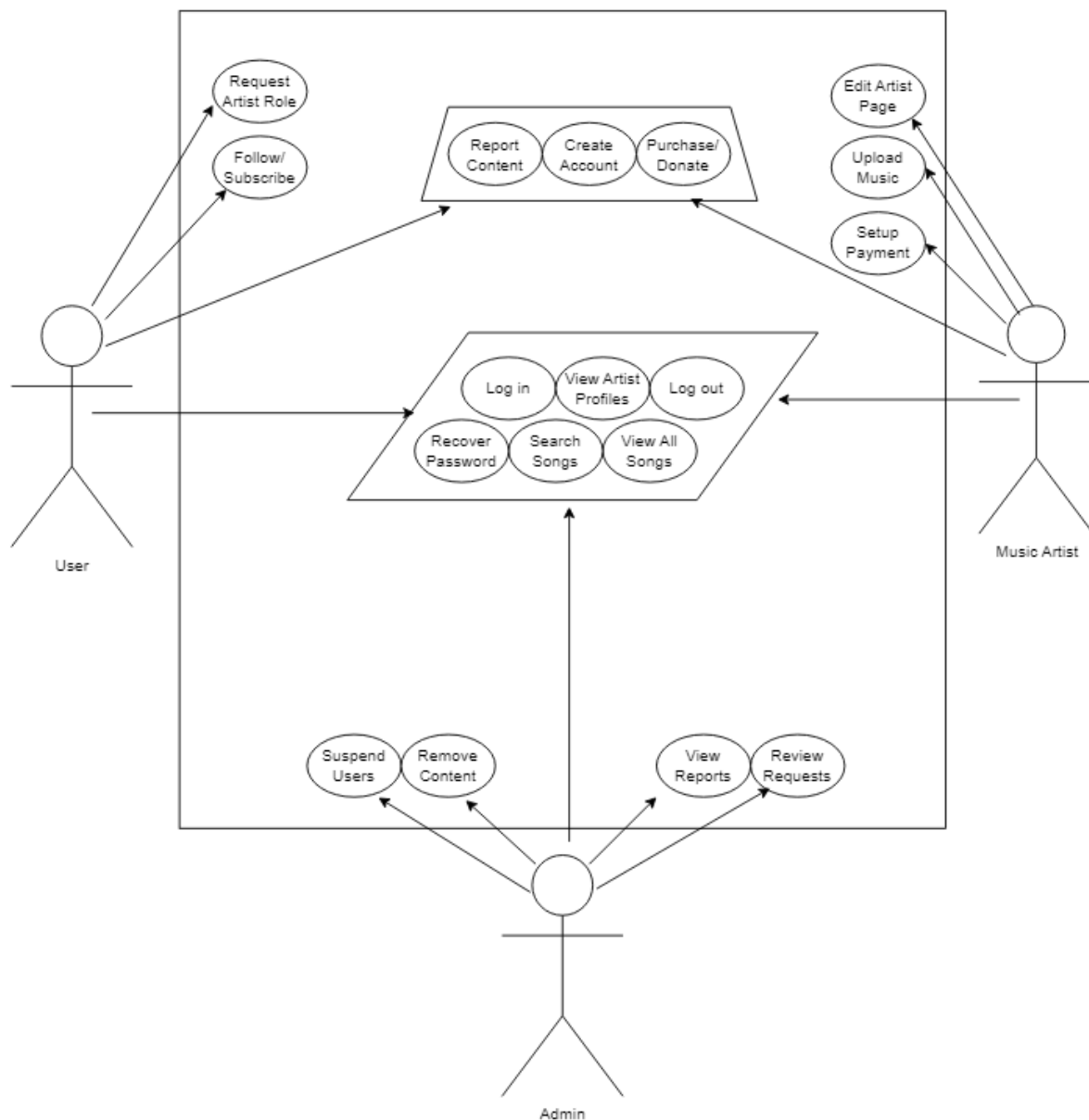
Version 2

Matthew Ingallinera, Gabe Roberts, Albert Lusk

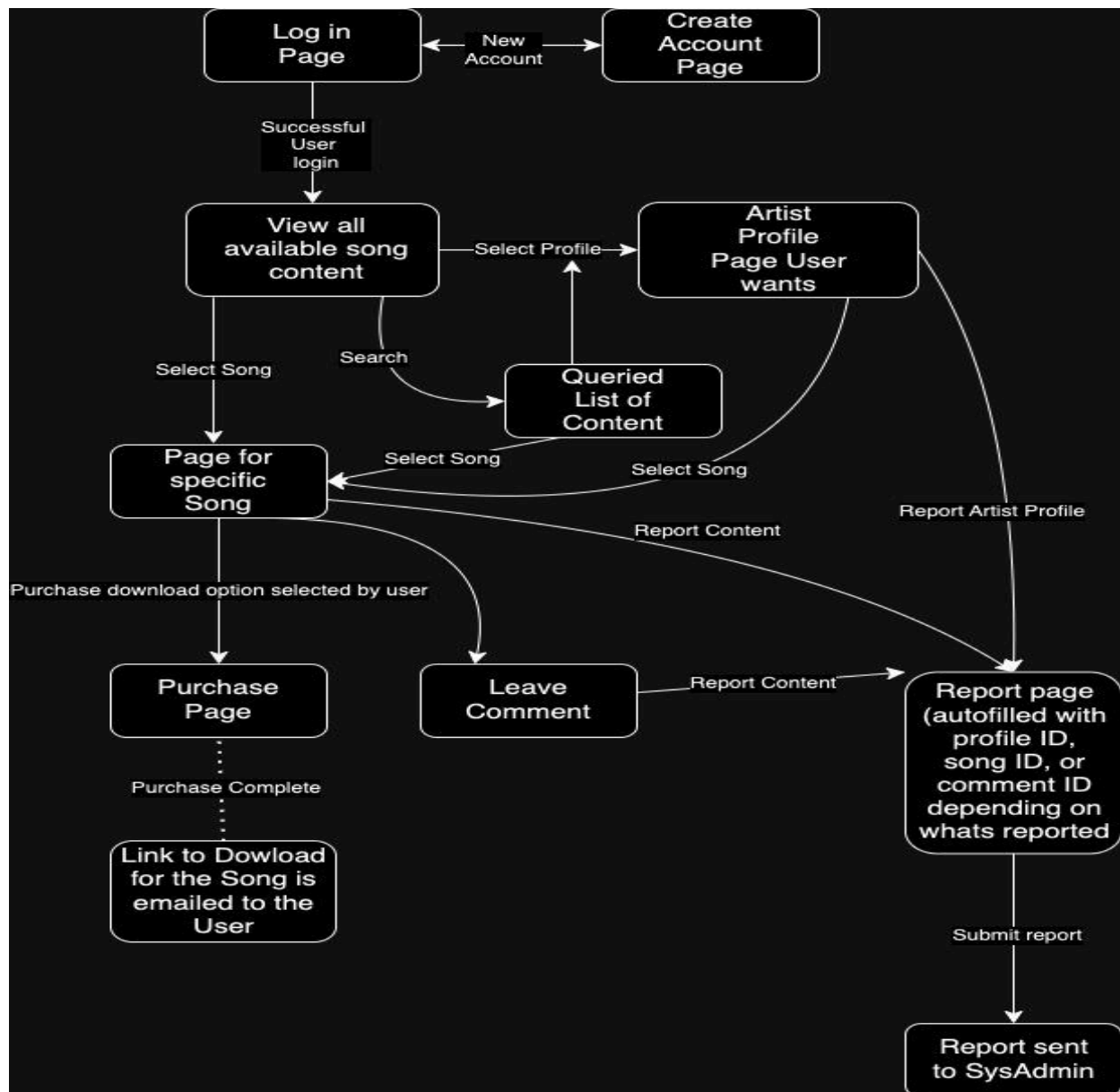
1. Project Overview:

The primary goal of our application is to promote collaboration between UNCG students, specifically in regards to music production. Our application will allow students to showcase and sell music they produce. And allow others on campus to contact them for the purpose of collaborations. There are a multitude of student run projects and clubs that incorporate music. Such as: Carmichael Studios, members of the UNCG game dev club, and other music producers on campus that are interested in collaborations.

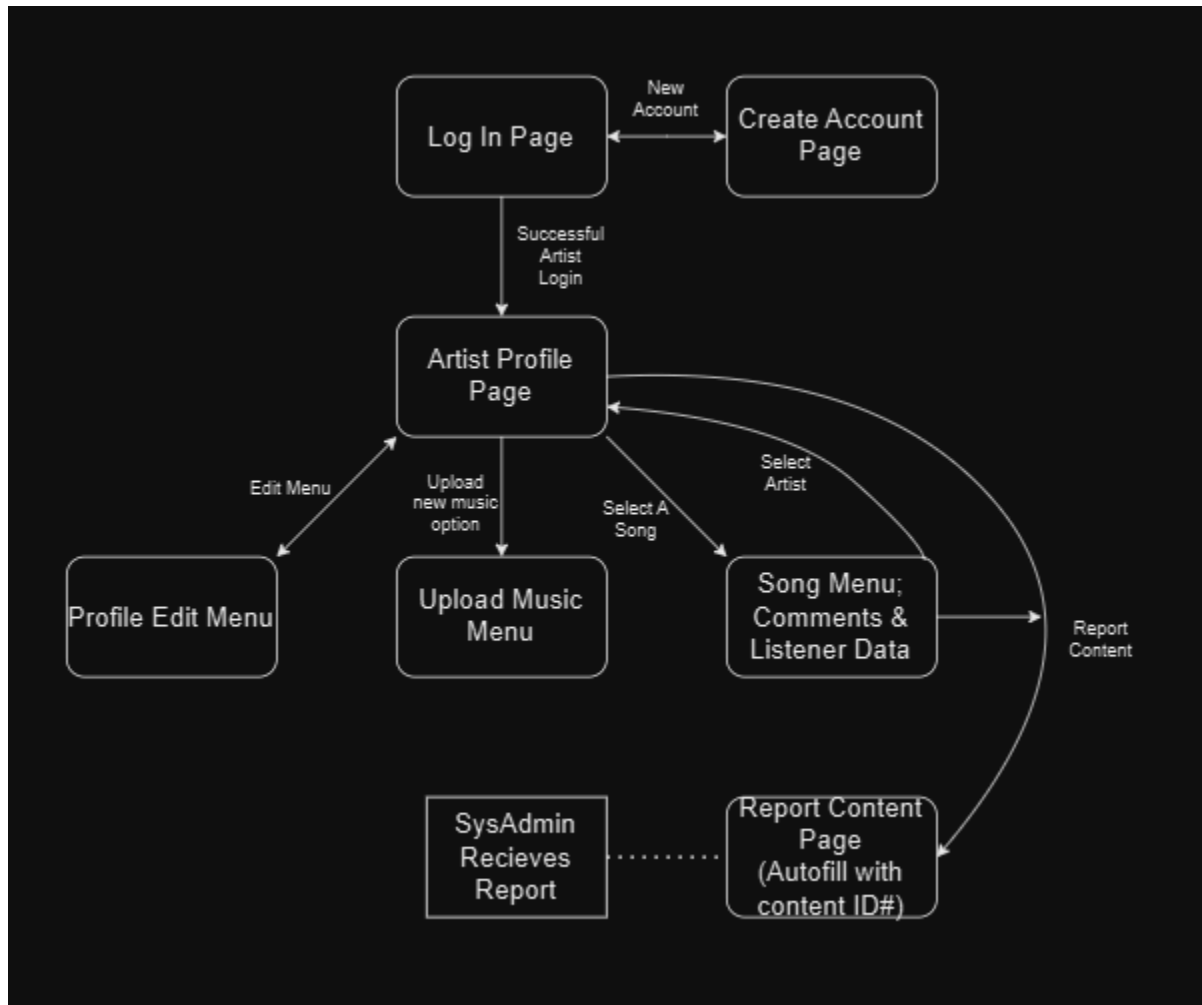
2. Use-Case Model



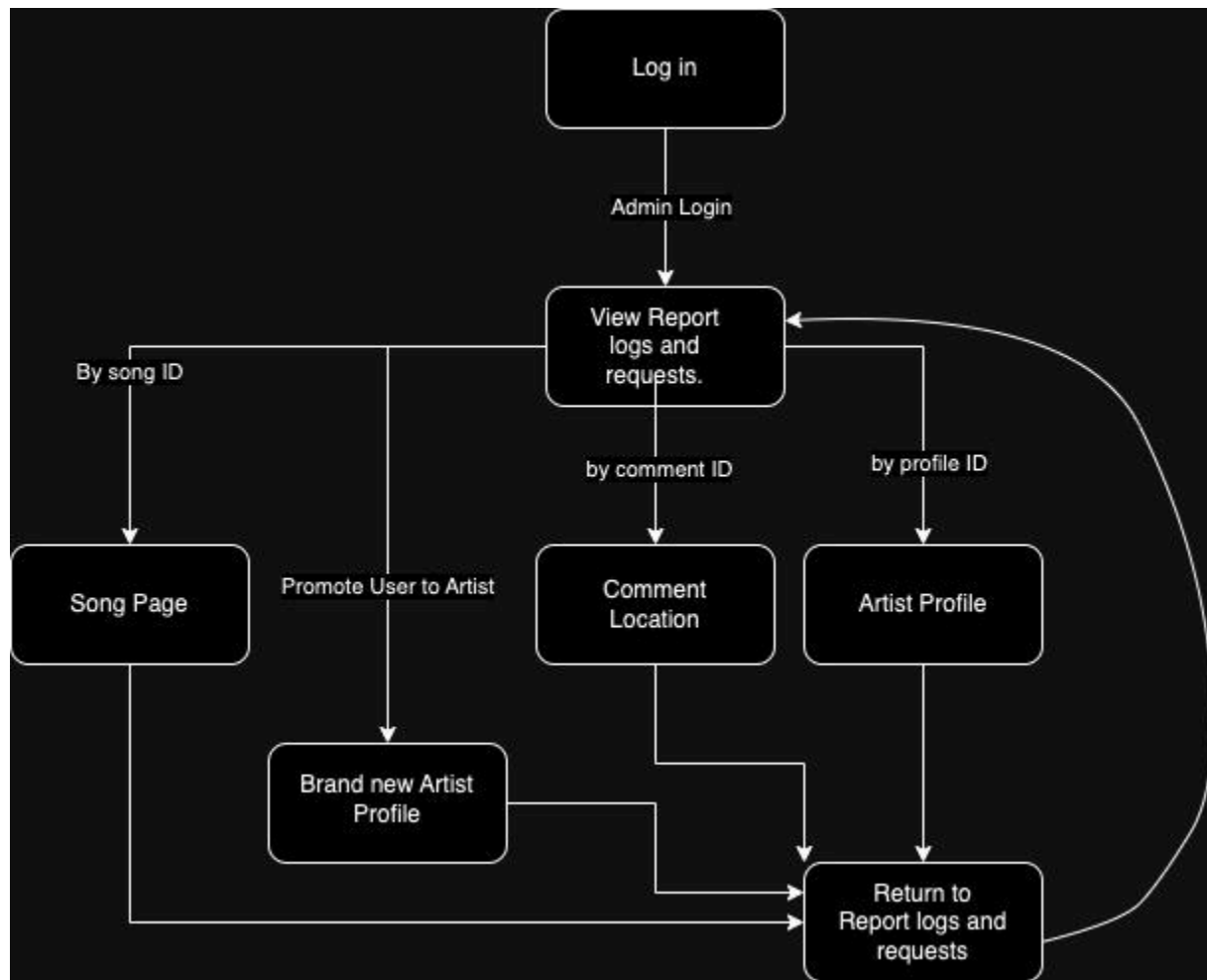
3. State Machine Diagrams:
User: Matthew Ingallinera



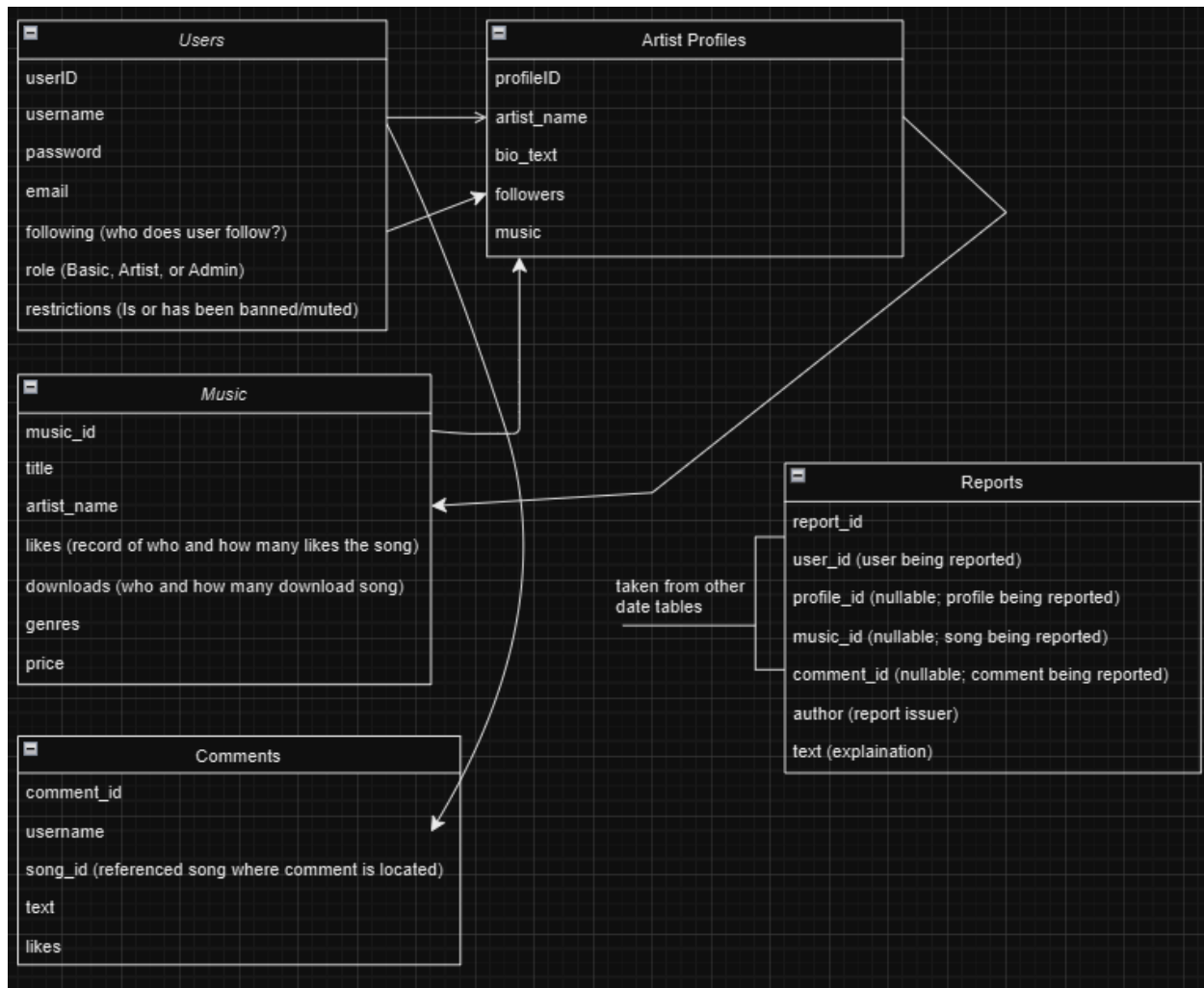
Artist: Gabe Roberts



Admin: Albert Lusk



4. Database Schema:



5. Software Architecture - MVC

