Pineapple Music

Group 3

Alex Fleming, Ethan Brown, David Reese, Neel Patel

Spring 2019

**Planning Scheduling and Peer Evaluation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Assignee Name | Email | Task | Duration  (hours) | Dependency | Due Date | Evaluation |
| Alex Fleming | amfleming3861@gmail.com | Revise Problem Statement, Create Report and System Requirements | 2 hours | Needs other tasks finished first | 2/15 | 100 |
| David Reese | bunzssbm@gmail.com | Class Diagrams | 2 hours |  | 2/14 | 100 |
| Ethan Brown | ethanbrown78@gmail.com | Update To Do list, Case diagrams, Use Cases, and System Requirements | 2 hours |  | 2/14 | 100 |
| Neel Patel | np69450@gmail.com | Database Specification and Analysis | 2 hours |  | 2/14 | 100 |

Problem Statement

1. What is your product, on a high level? A platform for music artists to upload their music and for users to listen to that music.
2. Whom is it for? It is for clients using a web browser who want to share their music and those who want to listen to music. It is for people who do not want to pay a fee to share/listen to music.
3. What problem does it solve? It provides a free option for artists to share their music on a copyright protected platform. Users can discover new local artists; established bands will be deterred from the site due to no compensation, so artists looking to be discovered can thrive in this community.
4. What alternatives are available? Soundcloud. You can upload music for free and certainly stream music for free. However, SoundCloud offers paid subscriptions to artists in return for spotlighting their music, schedule their releases, and other perks. Our product will spotlight artists based on popularity, recommendation systems, and if they have an upcoming live show in the user’s city.
5. Why is this project compelling and worth developing? This project is compelling and worth developing because of the large number of artists with access to computers who aren’t well known and want to gain a following.
6. Describe the top-level objectives, differentiators, target customers, and scope of your product. Our objective is to create a platform for users to discover new music and keep track of artists in their area. Our service is free and recommends artists based on recommendation systems and local activity. Our target customers are music enthusiasts and small bands willing to forgo profit to gain a bigger following. Our platform focuses on just music, no videos. Users have accounts, including artists who can post about new music and upcoming shows.
7. What are the competitors and what is novel in your approach? Soundcloud and Bandcamp are our biggest competitors, though they charge their users fees for certain functionalities. Our entire service is free.
8. Make it clear that the system can be built, making good use of the available resources and technology. We have a team of 4 software engineering students. We will use java to support the back-end and a SQL database to store data.
9. What is interesting about this project from a technical point of view? We can use data mining techniques to create groups of people with similar interests and recommend songs that similar users like. With user’s location data, we can also recommend bands when they have an upcoming show in that user’s area.

****

**Use Cases**

**Use Case Number:**​ 1 ​**Use Case Name**​: Finding an Artist

**Actors**​: User

**Description**​: The user searches the account database for an artist either using an account name or tags. The account database is searched using the name or tags depending on set preferences. A list of matching results is then displayed to the user.

**Alternate Path**​: you will either be displayed the matching results or a message ‘no matching results’.

**Pre-Condition**​: Must specify searching for an artist and must specify if searching by name or by tags. Must be under 25 characters if by name and under 25 if by tags.

**Use Case Number:**​ 2 ​**Use Case Name**​: Finding a Song

**Actors:** User​

**Description**​: The user searches the music database for a song either using an song name or tags. The music database is searched using the name or tags depending on set preferences. A list of matching results is then displayed to the user.

**Alternate Path**​: A list of matching songs is displayed or a message ‘no matching results’ is shown.

**Pre-Conditions**​: must specify searching for a strong and whether they are searching by name or tags. Must be under 25 characters for finding by name and under 10 tags if by tags.

**Use Case Number**​: 3 ​**Use Case Name**​: Streaming a Song

**Actors**​: User

**Description**​: A user selects a song, and the music database selects the correct mp3 file using the ID and plays it.

**Alternate Path**​: If a song cannot be streamed due to connection issues, then an error code will be shown.

**Pre-Condition**​: They must have connection and must the song must be public to them. (If they don't then they can't see it).

**Use Case Number**​: 4 ​**Use Case Name**​: Uploading/Removing a Song

**Actors:**​ User and Account Database

**Description**​: The user uploads an mp3 file and specifies the name + tags. The account database sets their status to artist and the mp3 file and specified tags + name is added to the database. If they are removing a song then the account databases is accessed to make sure that they are the artist of the song and if yes then the song is removed from the music database. **Alternate Path**:​ If the file doesn't meet the requirements or if there is a connection problem an error code is shown.(For removing: If they are not the artist of the song then they aren't shown the option to remove the song.)

**Pre-Conditions**​: Must be an mp3 file type and under 30 minutes long. The name must be under 25 characters. There must be at least 1 tag and under 10. (For removing: Must be the artist of the song.)

**Use Case Number**​: 5 ​**Use Case Name**​: Creating/Deleting a Profile

**Actors:**​ User and possibly the music database if an artist account is removed

**Description:**​ The user will enter a name and tags that describe them. They will set their profile to public or private. They will also provide an email and password. This information will be added to the account database. A forgot password code will be generated as well. Their profile can be edited at anytime using a profile editing page. Once changes are made they will hit submit and the changes will be recorded in the account database. For deleting an account, they will simply submit remove account and all their information will be removed from the account database and all their music, (if they have any), will be removed from the music database. **Alternate paths**​: If an email or username is taken they will be notified that an account with that data already exists and be prompted for either a different name or to use their email to log in. **Pre-Conditions:**​ Must have an unused email and the password must be at least 7 characters long and under 25. The username must be at least 1 character and under 25. Must have at least 1 tag and under 10. (For editing an account they must first be logged in.) For removing the account they must already have one.

**Use Case Number:**​ 6 ​**Use Case Name:**​ Resetting a Password

**Actors:** User​

**Description:**​ The user will click forgot password, then use a reset code provided in the email to access their account and go to the reset password page where they will enter and confirm a new password. The password will be updated in the account database and the reset code will be regenerated and replaced.

**Alternate Paths:**​ If they are already logged in they will just navigate to the reset page without using an emailed code. If the password is the same it will be denied for already being in use and if the password is too short it will be denied for not meeting the proper criteria.

**Pre-Conditions:**​ They must have access to their accounts email, or know their old password. They must have a new password that is at least 7 characters long and under 25 characters.

**Use Case Number:** 7 ​ **Use Case Name:**​ Creating/Removing a Playlist​

**Actors:**​ User and Account Database

**Description:**​ The user inputs the name of the playlist and whether or not it will be public. If it is public, anyone can listen to/ add songs and albums to it, if it is private, only the creator can add songs and albums to the playlist. They must be the creator of a playlist to remove it. The playlist is then added to the music database.

**Alternate Path:**​If the preconditions aren't met they will be prompted to meet them.

**Pre-Conditions:**​ The name must be under 25 characters. There must be between 1 and 10 tags. There must be at least one song added to the playlist.

**Use Case Number**​: 8 ​**Use Case Name:**​ Text post

**Actors**​: User

**Description:**​ A user can post a message to their account which will send a notification to all accounts which follow that account. Here artists can post about upcoming projects and shows **Alternate Paths:**​ None

**Pre-Conditions:**​ post must be under 150 characters

**Use Case Number:** 9​ **Use Case Name:**​ Create/Removing an Album​

**Actors:**​ User

**Description:**​ The artist inputs the name of the playlist and whether or not it will be public. They must then select at least one song. The playlist is then added to the music database. The artist can set public or private access to playlist at anytime using a toggle. Their ownership of the playlist is authenticated via the account database and the status is adjusted in the music database. Albums can be removed in the same manner as removing songs.

**Alternate Paths:**​ If there conditions aren’t met then they are prompted to meet the conditions. **Pre-Conditions:**​ The name must be under 25 characters. There must be between 1 and 10 tags. There must be at least one song added to the playlist. (For removing albums they must be the artist of that album)

**Use Case Number:**​ 10 ​**Use Case Name:**​ Liking a Song/Album

**Actors:**​ User and the account database

**Description:**​ A user can select ‘Like’ on a song and that song will have its ‘Like’ counter incremented. The song’s information will also be added to the account database section where liked songs are kept. The same process occurs if the user is liking an album but with the album replacing the song.

**Alternate Paths:**​ None

**Pre-Conditions:**​ User must have an account.

**Use Case Number:**​ 11 ​**Use Case Name:**​ Commenting on a Song/Album **Actors:**​ User

**Description:**​ A user can enter a comment to be added to a song or album. Their comment is added to the music database and is displayed to all other users.

**Alternate Paths:**​ If the pre-conditions are not met the user is prompted to meet them by the system.

**Pre-Conditions:**​ The comment must be under 50 characters long.

**Use Case Number:**​ 12 ​**Use Case Name:**​ Removing a Comment on a Song/Album

**Actors:** User and account database​

**Description:**​ Either the user who created the comment or the artist of the song has the option of removing a comment from a song or album so that it can no longer be seen. The account database is accessed so that their ownership is verified before being shown the option to remove the comment.

**Alternate Path:**​ None

**Pre-Conditions:**​ The user must either be the creator of the comment or the artist of the song.

**Use Case Number:**​ 13 ​**Use Case Name:**​ Following an Artist

**Actors:** ​User and music database

**Description:** A user can select “Follow” next to an artist name. The users account will then receive a notification whenever that artist makes a post of any kind(text post, uploading a song). They also have an option to unfollow at anytime.

**Alternate Paths:** None​

**Pre-Conditions:**​ Must have an account.

**Use Case Number:**​ 14 ​**Use Case Name:**​ Adding/Removing Songs for a Playlist

**Actors:**​ User

**Description:** ​The creator of a playlist can add any public song to their playlist using an option available on all public songs. The system will simply append or remove the song from the playlist.

**Alternate Paths:**​ If there is only one song in the playlist and the user is removing the song, they will be notified that if they do the playlist will be deleted and asked if they want to continue. (If they do then it will be deleted and if they don’t then it won’t). **Pre-Conditions:**​ They must be the owner of the playlist.

**Use Case Number:**​ 15 ​**Use Case Name:** Sharing a Song/Album/Playlist​

**Actors:** ​User and account database

**Description:**​ The user can search any public profile and then select an option to share either any public media on the site or any of their own media on the site to share with them. (Even if the media is private if the sender is the artist then whoever they share it with will be able to access it indefinitely. In that case the receiver is given a code that allow them to see the media in the music database). The receiver is given a message with the songs basic information from the music database.

**Alternate Paths:**​ none

**Pre-Conditions:** The sender must be either the owner of the material or the material must be​ public and the receiver must have a public profile.

**System Requirements**

**Requirement number:** 1    **Use Case number:** 1

**Introduction:**  Searching the account database for an artist via string

**Inputs:**  string between 1-25 characters

**Requirements Description:** account database compares input string to artist names

**Outputs:** links to the pages of the top 5 lexically similar matches of artist names in the database

**Requirement number:**2 **Use Case number:** 1

**Introduction:** Searching the account database for an artist via tags

**Inputs:**  1-25 hardcoded tags selected by the user

**Requirements Description:**  the account database finds artists who have the most matching tags

**Outputs:**  links to the pages of artists with all matching tags

**Requirement number:** 3    **Use Case number:** 2

**Introduction:**  Searching the music database for aa song via string

**Inputs:**  string between 1-25 characters

**Requirements Description:** music database compares input string to song names

**Outputs:** the song name and id of the top 5 lexically similar matches of song names in the database

**Requirement number:**4 **Use Case number:** 2

**Introduction:** Searching the music database for a song via tags

**Inputs:**  1-10 hard-coded tags selected by the user

**Requirements Description:**  the music database finds songs which have the most matching tags

**Outputs:**  the song name and id of songs with all matching tags

**Requirement number:** 5 **Use Case number:** 3

**Introduction:**  Selecting an mp3 file from the music database

**Inputs:**  The song id of the song clicked on by the user or the id of the next song in a playlist

**Requirements Description:**  The music database selects the song with the matching id of the desired song and returns the .mp3 file to the client

**Outputs:**  the .mp3 file of the desired song

**Requirement number: 6    Use Case number:** 4

**Introduction:**  uploading a song

**Inputs:**  .mp3 file under 30 mins, song name under 25 characters, 1-10 tags

**Requirements Description:**  the .mp3 file attributed the given name and tags in the music database, that created song id is added to the artists account. If this is the first song added to an account, that user becomes an artist

**Outputs:**  song id, new tuple in the music database

**Requirement number:** 7 **Use Case number:** 4

**Introduction:**  removing a song

**Inputs:**  song id

**Requirements Description:**  user clicks on a song to be removed (if they aren’t the artist of that song, they aren’t shown this option), the tuple with the matching song id is removed from the music database and the id is unassociated with the artists account

**Outputs:**  none

**Requirement number:**8 **Use Case number:** 5

**Introduction:**  creating a profile

**Inputs:**  username not already in database and having 1-25 characters, email not already in account database, password having 7-15 characters

**Requirements Description:**  the data is added to the account database, along with a randomly generated password reset code

**Outputs:**  a new tuple in the account database

**Requirement number:**9 **Use Case number:** 5

**Introduction:**  removing a profile

**Inputs:**  user select to delete profile

**Requirements Description:**  user must be logged into that profile to delete it

**Outputs:**  all the tuples in the music database associated with that account (if there are any) are deleted and the tuple in the account database associated with that account is deleted

**Requirement number:**10 **Use Case number:** 6

**Introduction:**  request resetting password for a user account

**Inputs:**  email

**Requirements Description:**  the user inputs their email in the login page and clicks “forgot password”. If the email exists in the account database, the password reset code associated with that account is sent to that email along with a link to the password reset page

**Outputs:**  link and password reset code sent to email

**Requirement number:**11 **Use Case number:** 6

**Introduction:**  password reset page

**Inputs:**  email, password reset code, new password having 7-15 characters

**Requirements Description:**  if the given password reset code matches the on in the account database associated with that email, the password is reset to the given password and a new password reset code is generated for that account. Otherwise an error is displayed

**Outputs:**  a new password reset code and new password are given to that user account

**Requirement number:**12 **Use Case number:** 7

**Introduction:** creating a playlist

**Inputs:** playlist name having 1-25 character

**Requirements Description:** the user creates a playlist to be associated with his account. The name of the playlist must be unique to the other playlists that user has created, but not necessarily unique to all playlists in the music database.  It will be added to the music database as an empty list. If the public option is chosen, other users can view/add to/remove from/ delete that playlist. If private, only the creator of the playlist can view/add to/ remove from/delete that playlist

**Outputs:**  empty list in music database with playlist id associated to the user account.

**Requirement number:** 13  **Use Case number:** 7

**Introduction:** adding a song to a playlist

**Inputs:** song id, valid playlist name

**Requirements Description:** If the playlist selected is public playlist, the song id is added to the list of songs for that playlist in the music database. If the playlist is a private playlist, the user must be the owner of the playlist in order to add to it.

**Outputs:**  append the song id to the playlist list

**Requirement number:**14 **Use Case number:** 7

**Introduction:** removing a song from a playlist

**Inputs:** selecting “remove from playlist” next to a song in a playlist

**Requirements Description:** The “remove from playlist” option will only be visible if the user is the creator of that playlist

**Outputs:**  the selected song id will be removed from the playlist

**Requirement number:** 15 **Use Case number:** 7

**Introduction:** deleting a playlist

**Inputs:** selecting “delete playlist” next to playlist name

**Requirements Description:** if the user is the owner of the specified playlist, the playlist will be removed from the music database

**Outputs:**  none

**Requirement number:**16 **Use Case number:** 8

**Introduction:** creating a post to be displayed to all followers

**Inputs:** A string of text having 1-150 characters

**Requirements Description:** the text post will be posted on the users page and any user following that user will receive a notification of the posts existence

**Outputs:**  the string will be displayed on the users public profile

**Requirement number:**17 **Use Case number:** 9

**Introduction:** creating an album

**Inputs:** a list of songs which belong to the user, album title having 1-25 characters, 1-25 tags

**Requirements Description:** if a user has more than 2 or more song uploads associated with their account, they will have the option in their profile to create an album. An album is a special case of a playlist in which it is public but may not be edited by users other than the creator. All if the selected songs must belong to that user. They will be stored as a list of song ids in the music database with tags as an attribute.

**Outputs:**  a list of songs in the music database

**Requirement number:** 18 **Use Case number:** 10

**Introduction:** Liking/Unliking a Song/Album

**Inputs:** Button Click through UI

**Requirements Description:** Any user has the option to like a song or album via a button in the UI. If clicked then a ‘like’ will be added to the song/album count in the database. The same thing occurs if the button is re-clicked but rather than incrementing the count it will be decremented.

**Outputs:** The songs like count will be incremented and the button for like will be toggled to un-like. If it is clicked while in this state a ‘like’ will be removed and the button will be toggled to like.

**Requirement number:** 19 **Use Case number:** 11

**Introduction:** Commenting on a song/album

**Inputs:** a user will click a button to activate the option to enter text, and then must enter text with length being between 1 and 50 characters.

**Requirements Description:** The account database will be accessed to get the writer’s account name and it will add this to the music database where the comments are stored.

**Outputs:** The comment and writer’s name will be available under the song/album for any users to see.

**Requirement number:** 20 **Use Case number:** 12

**Introduction:** Removing a comment on a song/album

**Inputs:** Button click through UI

**Requirements Description:** a user, (either the writer or the artist), will have the option to click a button in the UI that will remove the comment from the music database where it is stored. (Thus it will no longer be visible)

**Outputs:** The button and comment will no longer be visible in the UI and the comment will be removed from the music database.

**Requirement number:** 21 **Use Case number:** 13

**Introduction:** Following/Unfollowing an Artist

**Inputs:** Button click through UI

**Requirements Description:** a user will have the option to click a button in the UI that will add them to the artist’s follow list so that whenever the artist uploads a song, every account on that list will be sent an email alerting them to the new song.  The button will toggle to unfollow and if clicked while like that the user’s info will be removed from the artist’s follow list.

**Outputs:** The button will be toggled to unfollow when clicked on in the follow state. It will then email the user whenever the artist uploads a song. If clicked on in the unfollow state it will convert to follow and the user won’t receive email notifications from that artist.

**Requirement number:** 22 **Use Case number:** 14

**Introduction:** Adding/Removing Songs from a Playlist

**Inputs:** Button click through UI

**Requirements Description:** The creator of a playlist can add any public song to their playlist using an option available on all public songs using a UI button. The system will then simply append or remove the songs info to the playlist. (The playlist is treated like a collection of links to songs so that if a song is removed by an artist it will no longer appear in the playlist).

**Outputs:** The button will no longer be visible in the UI and the songs information will appear in the playlist and will play in the playlist.

**Requirement number:** 23 **Use Case number:** 15

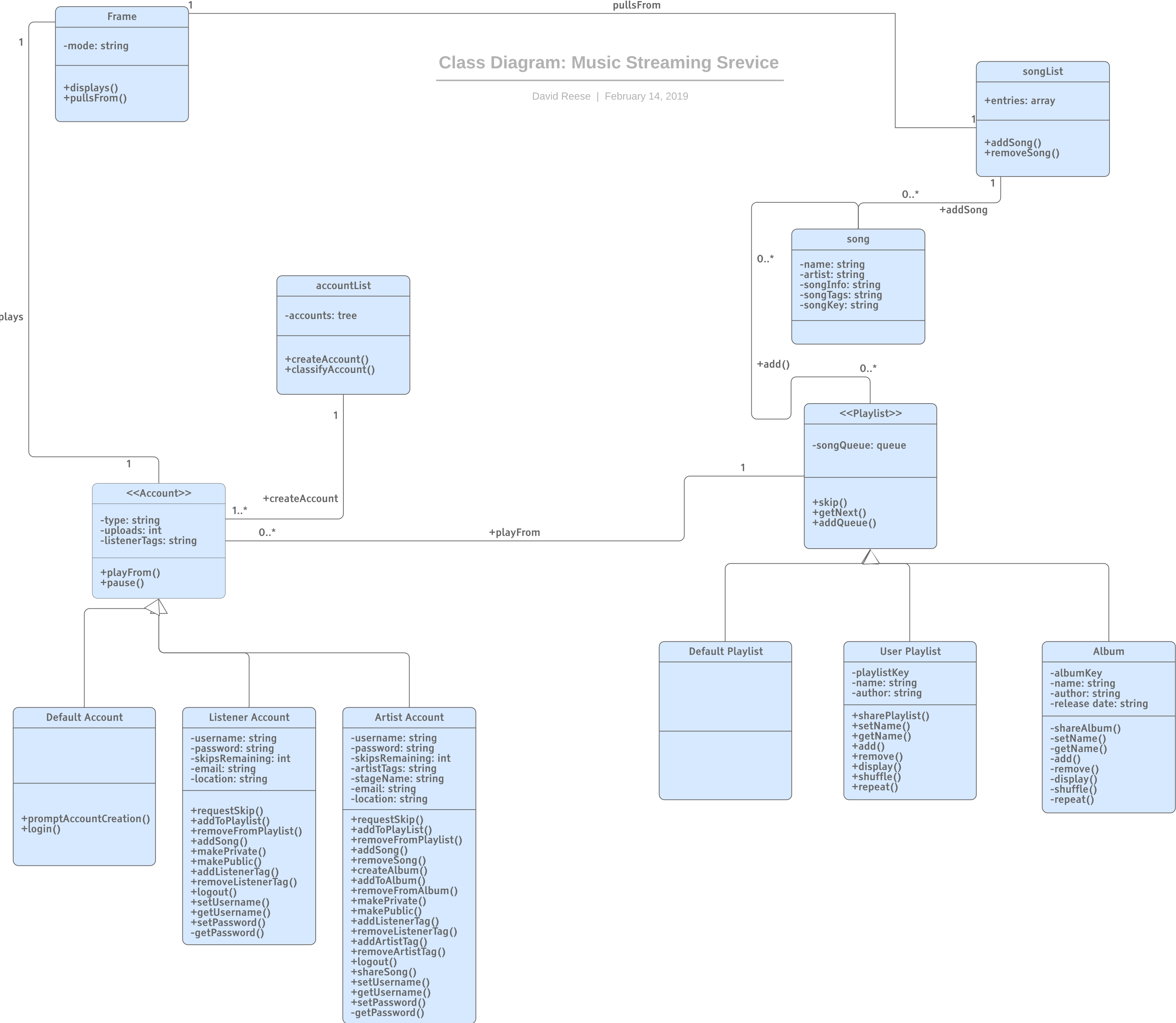
**Introduction:** Sharing a Song/Album/Playlist

**Inputs:** Button click through UI

**Requirements Description:** The user can search any public profile and then select an option to share either any public media on the site or any of their own media on the site to share with them using a button in the UI. Even if the media is private if the sender is the artist then whoever they share it with will be able to access it indefinitely. In that case the receiver is given a code, (similar to the way song links appear in playlists), that allow them to see the media in the music database.

**Outputs:** The receiver is given a message with the songs basic information from the music database. Using this information they can pull up and listen to the music.

System Modeling



Database specification and analysis

* We will be using a case diagram. We will be using the login as a Primary Keys because both user and account database can access. On the other hand, Tags, Name, Add/Remove/ Song List is the Foreign Keys due to the User being the only one to access.
* Java, Python, HTML, CLI-Unix, SQL, C, Beta-testing, Analyzing/ transposing code are the language we will be using.

MySQL and Oracle will also be used for our music streaming service.

