**Fully Dressed Use Cases**

This document outlines the 16 Fully Dressed Use Cases for the Collaborative Music Player

Use Case 1: Song Playback

**Primary Actor:**

User

**Stakeholders and interests:**

User, hear song with good music quality

**Preconditions:**

A song was queued

**Success guarantee:**

The song was played

**Main success scenario:**

1. User Uploads song to the queue

2. The system notifies the user the song was uploaded and added

3. Application will play the next queued song

4. The song is removed from the queue after completion

**Extensions:**

1a. Song upload failure

1. System notifies user of upload failure
2. The user may try again

3a. Song playback failure

1. System pulls the song from the queue
2. The next queued song is played

**Special requirements:**

**Technology and data variation List:**

1. Song format must be natively or convertible to a supported audio format
   1. Supported Audio formats are AAC, MP3

**Frequency of Occurrence:**

Use Case 2: User Registration

**Primary Actor:**

User

**Stakeholders and interests:**

User, wants to register to the system to add songs

System, wants to keep track of users’ credits

**Preconditions:**

User has a way to validate themselves

**Success Guarantee:**

User is registered to the system and given the correct number of credits

**Main Success Scenario:**

1. User accesses the webpage
2. System prompts the user to register
3. User enters account information
4. System checks if the information is valid
5. System sends a verification email
6. System prompts the user to check their email
7. User accesses verification email
8. System registers the user
9. System gives the user 10 credits

**Extensions:**

1. Invalid information
   1. System signals the error and rejects it
   2. System prompts the user to enter valid information
   3. User enters different information
2. Email entered is already registered
   1. System notifies the user that the email is already being used
      1. User logs in using the email
      2. User registers with a different email
3. There is no verification email from the system
   1. User requests another verification email

**Special Requirements:**

**Technology and Data Variations List:**

**Frequency of Occurrence:**

Once per user

Use Case 3: User Login

**Primary Actor:**

User

**Stakeholders and interests:**

User, wants to login to add songs

System, wants to authenticate that user

**Preconditions:**

The user is registered and account is verified

**Success Guarantee:**

User is authenticated and is given full application access permissions

**Main Success Scenario:**

1. The user access the login page
2. The user enters his/her account details
3. The system authenticates the user and and loads account details

**Extensions:**

2a. Bad login

1. System notifies user of failed login attempt
2. The ip address is blocked after 3 failed attempts

**Special requirements:**

**Technology and data variation List:**

1. User accounts are stored in SQL table as key value pairs

**Frequency of Occurrence:**

Use Case 4: No More Credits

**Primary Actor:**

Listener

**Stakeholders and interests:**

Listener wants to bump a song

**Preconditions:** User is logged in

Song(s) is/are queued

**Success guarantee:**

Song is not bumped up in queue

**Main success scenario:**

1. User tries to bump a song with 0 credits

2. The system doesn’t allow the user to bump the song--queue positions are not changed

**Extensions:**

1a.

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 5: Bumping a Song

**Primary Actor:**

User

**Stakeholders and interests:**

User bumps a song

**Preconditions:**

1. User is logged in
2. User has more than 0 credits
3. Song(s) is/are queued

**Success guarantee:**

Song is bumped up in queue

**Main success scenario:**

1. User identifies song in queue to bump

2. The user selects the song to bump

3. User selects the number of credits to use

4. The song is rearranged in the queue in order of descending credit value

**Extensions:**

3a. The user selects more credits than they have

1. The user can only add the number of credits they have available

4a. The song is at the top of the queue

1. The song acquires more credits but is not moved in position

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 6: Adding a Song to the Queue

**Primary Actor:**

Listener

**Stakeholders and interests:**

User wants their song played

**Preconditions:**

The user is logged in

**Success guarantee:**

The song was uploaded

**Main success scenario:**

1. User uploads a supported music file or submits a link for a song

2. The system notifies the user the song was uploaded and added

**Extensions:**

1a. Song upload failure

1. System notifies user of upload failure
2. The user may try again

3a. Unsupported song format

1. System notifies the user of upload failure

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 7: Credit Return

**Primary Actor:**

System

**Stakeholders and interests:**

System, returns credits to user accounts

User, receives credits

**Preconditions:**

The current playing song is completed

The song has a number of credits

**Success guarantee:**

All credits bumped into this song are returned to the respective user

**Main success scenario:**

1. Current playing song is finished

2. System attempts to redistribute the credits that were given to this song

3. Each user that put credits into the song receives that number of credits back

**Extensions:**

1a. The song had no credits

1. There will be no credits returned

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 8: Display Credits

**Primary Actor:**

System

**Stakeholders and interests:**

System, shows the credits for user convenience

User, wants to see how many credits they have

**Preconditions:**

The user is logged in

**Success guarantee:**

The credits are displayed properly for each user

**Main success scenario:**

1. System retrieves the number of credits that the user has
2. System creates a display to show the user their credits
3. User views their credits

**Extensions:**

1. System accesses the wrong users credits
   1. The user sees the wrong number of credits
   2. >??

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 9: Song Progress

**Primary Actor:**

System

**Stakeholders and interests:**

System, shows progress of song

User, wants to know the progress of the song

**Preconditions:**

Their is a song being played

**Success guarantee:**

The progress of the song being played song is displayed

**Main success scenario:**

1. A user access the queue page

2. The current song being played has is progression displayed in minutes:seconds

**Extensions:**

2a. There is no song being played

1. No progress is indicated

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 10: User Joins a Room

**Primary Actor:**

System

User

**Stakeholders and interests:**

System, has multiple playback rooms available to join

User, wants to join a room

**Preconditions:**

There is a room that can be joined

**Success guarantee:**

The user is given a list of rooms to join and is allowed to join them.

**Main success scenario:**

1. A user wants to join a room

2. The user selects to join a room and is redirected to it

**Extensions:**

1a. There are no rooms to join

1. The user cannot join a room

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 11: Account Details

**Primary Actor:**

System

User

**Stakeholders and interests:**

System, retrieves user’s account details

User, wants to view their account details

**Preconditions:**

The user is logged in

**Success guarantee:**

The user is shown their account details and may edit them

**Main success scenario:**

1. A user is logged in

2. The user wishes to edit their account details

3. The user can access his/her account details page

4. The user can update their username, password, email

**Extensions:**

4a. The user changes password

1. Must confirm and match a password
2. Must provide security

4b. The user updates an email

1. Must confirm and match email
2. Must provide security

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 12: Parallel Playback

**Primary Actor:**

System

**Stakeholders and interests:**

Listeners want to listen to the same songs at the same time

**Preconditions:**

A song is playing

**Success guarantee:**

System plays the same song on all connected devices.

**Main success scenario:**

1. System saves and loads the current song state

2. When a user accesses the webpage, the song that is currently being played on the system is played in the user’s browser.

**Extensions:**

1a.

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 13: Adding a Friend

**Primary Actor:**

System

User

**Stakeholders and interests:**

System, connects two accounts together

User, adds a friends to their friends

**Preconditions:**

The user is logged in

Their is another user that can be “friended”

**Success guarantee:**

Both the user and friendee become friends

**Main success scenario:**

1. A user wishes to add a friend

2. The user searches for their friends username

3. The server returns a list of matching usernames

4. The user can select any of these users and send a friend request

5. The request can be accepted or rejected by the other party

6. If accepted the two users are now friends

**Extensions:**

2a. The friends username is not found

1. The user does not exist

5a. The request is rejected

1. The user can request an infinite number of times

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 14: Removing a Friend

**Primary Actor:**

User

**Stakeholders and interests:**

System, end the connection between two users

User, wants to remove a friend from their friends

**Preconditions:**

The user is logged in. User has a friend to remove.

**Success guarantee:**

The user’s friends list should be updated with the other user removed.

**Main success scenario:**

1. User accesses their account information
2. User accesses their friends list
3. User selects one of their friends
4. User removes their friend
5. System prompts the user to make sure they want to remove their friend
6. User confirms
7. System removes the connection from the user’s friends list

**Extensions:**

1. User selects the wrong friend
   1. User removes their friend
   2. System prompts the user to make sure they want to remove their friend
      1. User sees they’re removing the wrong friend
         1. User stops and chooses the friend they meant to remove
      2. User confirms
         1. System removes the friend

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 15: Join a Friends Room

**Primary Actor:**

System

User

**Stakeholders and interests:**

System, moves a user to room

User, wants to join a friends room

**Preconditions:**

The user is logged in

The user has a friend in a different room

**Success guarantee:**

The user joins the friends room

**Main success scenario:**

1. A user is logged in

2. The user wishes to join a friends room

3. The user, from their friends list, selects to join a friends room

4. The user joins the friends room

**Extensions:**

2a. The user and friend are already in the same room

1. The user is not moved

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**

Use Case 16: Delete Account

**Primary Actor:**

System

User

**Stakeholders and interests:**

System, removes an account

User, wishes to delete their account

**Preconditions:**

The user is logged in

**Success guarantee:**

The user is able to delete their account from the account details page

**Main success scenario:**

1. A user is logged in

2. The user wishes to delete their account

3. The user can access his/her account page and request to delete it

4. The system asks for confirmation

5. The user confirms and the account is deleted

**Extensions:**

4a. The user cancels the request

1. The account is not deleted

**Special requirements:**

**Technology and data variation List:**

**Frequency of Occurrence:**