

Seth Nute, Andrew Baehre, Tony DiMarco, Amit Ishairzay
Intermediate Software Development Group Project
Collaborative Music Player

Fully Dressed Use-Case Model

Use Case 1.) User creates an account

Primary Actor: User

Stakeholders and interests:

Site admins; interested because they care about user demographics

Preconditions:

User has navigated to the application

Success Guarantees:

User account is created with the credentials they have provided (username and password)

User has 10 coins in their account balance

Main success scenario:

- 1.) User navigates to signup page
- 2.) User enters desired username and password
- 3.) System validates username and password choice
- 4.) User account is created
- 5.) User account gets 10 coins added to it
- 6.) User is logged into their new account

Extensions:

- 3.) Validation fails
 - 3a.) Username already exists and they create a new one
 - 3b.) Password doesn't meet password requirements and they enter a new one
 - 3c.) Passwords don't match and they need to retype them

Special requirements: User data needs to be stored securely

Technology and data variation list:

- User interacts with the application through modern web browsers
+ mouse + keyboard

Frequency of occurrence: One time per user

Use Case 2.) User logs in to their account

Primary Actor: User

Stakeholders and interests:

Site admins; interested because they care about login data

Preconditions:

User has navigated to the application

User has already created an account

Success Guarantees:
User is logged into their account and has a session

Main success scenario:

- 1.) User navigates to login page
- 2.) User enters username and password
- 3.) System validates username and password
- 4.) User is logged into their account

Extensions:

- 3.) Validation fails
 - 3a.) Username doesn't exist and they need to retype it
 - 3b.) Password isn't correct for username and they need to reenter it
 - 3c.) User forgot username/password and needs to recover them

Special requirements: User data needs to be stored securely

Technology and data variation list:

- User interacts with the application through modern web browsers + mouse + keyboard

Frequency of occurrence: Up to one time per user per application visit

Use Case 3.) User spends a coin to add a song to a playlist

Primary Actor: User

Stakeholders and interests:

Site admins; interested because they care about user spending data

Music artists; interested because of data about users listening to their music

Preconditions:

- User has navigated to the application
- User has already created an account
- User is logged in

Success Guarantees:

- Song gets added to the bottom of a playlist
- User loses one coin from their account balance

Main success scenario:

- 1.) User navigates to the add song page
- 2.) User provides the API credential for the song they want (URL most likely)
- 3.) User picks what playlist they want to add to
- 4.) System validates if song is valid and the add is valid
- 5.) Song is added to playlist of choice
- 6.) User loses one coin from their account

Extensions:

- 1.) User doesn't have enough coins to add a song and redirections is blocked

4.) Validation fails
4a.) Song credential is invalid, user needs to reenter it
4b.) Playlist already contains song and it is rejected (no duplicates allowed)

Special requirements: User data needs to be stored securely

Technology and data variation list:

- User interacts with the application through modern web browsers
+ mouse + keyboard

Frequency of occurrence: Semi-frequently for each user (based on the number of coins we end up giving them when they create their account)

Use Case 4.) User spends a coin to add a song to a playlist from recommended songs

Primary Actor: User

Stakeholders and interests:

Site admins; interested because they care about user spending data

Music artists; interested because of data about users listening to their music

Preconditions:

User has navigated to the application

User has already created an account

User is logged in

Success Guarantees:

Song gets added to the bottom of a playlist

User loses one coin from their account balance

Main success scenario:

- 1.) User navigates to the add song from recommended page
- 2.) User selects the song they would like to add
- 3.) User picks what playlist they want to add to
- 4.) System validates if song is valid and the add is valid
- 5.) Song is added to playlist of choice
- 6.) User loses one coin from their account

Extensions:

1.) User doesn't have enough coins to add a song and redirections is blocked

4.) Validation fails

4a.) Playlist already contains song and it is rejected (no duplicates allowed)

Special requirements: User data needs to be stored securely

Technology and data variation list:

- User interacts with the application through modern web browsers
+ mouse + keyboard

Frequency of occurrence: Semi-frequently for each user (based on the number of coins we end up giving them when they create their account)

User Case 5.) User starts tuning into the playlist (listening to whatever is playing)

Primary Actor: User

Stakeholders and interests: Other members of the playlist and site admins.

Preconditions: User is logged in and the playlist they are tuning into exists.

Success Guarantees: User is listening to music from the playlist

Main Success scenario:

- 1.) User browses and finds a playlist they would like to tune into
- 2.) User then clicks the playlist and is synced with the current playlist.
- 3.) User is then on the page with the playlist and is listening to the song that is playing.

Extensions:

- 1). User may click a playlist which does not have music within it. No playlist may have music in it, therefore no music will be played regardless of playlist clicked.
- 2.) User may lose internet connection while trying to retrieve the playlist and not be synced with the current state of the playlist.
- 3.) User's internet is not strong enough to load the music playing.

Special requirements:

- Song playback should be concurrent between all users
- Song playlist should buffer smoothly

Technology and data variation list: Use an API to retrieve and stream music to a user.

Frequency of occurrence: Very frequent, because all users will want to listen to the music.

Use case 6.) User stops tuning into the playlist

Primary Actor: User

Stakeholders and interests: site admin to make sure a user can stop tuning into a playlist.

Preconditions:

- User is logged in
- User is tuned into a playlist

Success Guarantees: User is no longer tuned into a playlist

Main Success scenario:

- 1.) User is tuned into a playlist
- 2.) User clicks button to stop listening to playlist
- 3.) User is redirected to main landing page with no playlist selected

Extensions:

- 1.) The user is not properly connected to the playlist, and so is not synced.
- 2.) User may click the wrong button and not stop listening to the playlist.
- 3.) User may fail to load the landing page.

Special requirements:

- Web app must run smoothly on all modern browsers.

Technology and data variation list: Provide button for stopping and exiting a playlist.

Frequency of occurrence: Whenever a user wants to stop listening to a playlist. Could be often.

Use case 7.) Song user added just played and user gets 1 coin added to their account

Primary Actor: User

Stakeholders and interests: Other users in the playlist and site admins.

Preconditions:

- User must be logged in
- User has a song in the playlist
- User's song plays.

Success Guarantees:

- The user must have one more coin than they had before.
- The user's song must have played.

Main Success scenario:

- 1.) User is tuned into a playlist.
- 2.) User's song is played on the playlist.
- 3.) User is refunded the coin that was used to add this song to the playlist.

Extensions:

- 1.) User's connection could be poor and the playlist is not synced.
- 2.) User's song was also added by another user, and so two users are refunded for the song playing
- 3.) Server error prevents user from being refunded their coin. User has one less coin than they should have.

Special Requirements: Song playback should be concurrent across all tuned in users.

Technology and data variation list:

- Back-end component to update the user's coin count by 1
- Front-end component to display to a user how many coins they currently have in order to make actions on the site.

Frequency of occurrence: Often. When a user has a song that they added played on the playlist.

Use case 8.) User can adjust volume

Primary Actor: User

Stakeholders and interest: site admins, interested because they care about the user's experience.

Preconditions:

- User is logged in
- User is listening to a playlist

Success Guarantees:

- User's volume is increased or decreased
- User has changed the volume to their desired level.

Main Success Scenario:

- 1.) User is on playlist page, tuned into the playlist.
- 2.) User changes the slider to control the volume of the music.
- 3.) The volume is increased or decreased based on the direction the slider is dragged.

Extensions:

- 1.) Playlist may not be buffering and may be out of sync for the user.
- 2.) User may move the slider in the wrong direction from what they desire.
- 3.) The volume may be changed too much or too little from what a user desired.

Special Requirements:

The web app must work well on all modern browsers

Technology and Data variation:

Front-end component for a user to slide and control the volume of the playlist that is playing.

Frequency of Occurrence: Often. When a user would like to change the volume of the music playing.

Use Case 9.) User can choose the genre of the playlist they are listening to

Primary Actor: The user

Stakeholders: Users and Site Admins

Preconditions: User is connected and logged in to the site

Success Guarantees: User will be in a playlist of the genre they chose from a predetermined list

Main success scenario:

1. User logs into site
2. User selects genre on main page
3. User is redirected to selected genre page

Extensions:

1. User could not log into or connect to the site correctly
2. User selects wrong genre and wants to go back
3. User doesn't connect to the genre page

Special Requirements: None

Frequency of Occurrence: As often as User(s) log in and/or want to switch genres

Use Case 10.) User can report a song or user for not following proper songs or etiquette

Primary Actor: Users

Stakeholders: Site Admins and Users

Preconditions: User is connected and logged into the site and has selected another users account or song in a playlist

Success Guarantees: Report will be submitted successfully to the admins and action will be taken

Main success scenario:

1. User selects song or other users account
2. User clicks report and selects reason for reporting
3. If song receives more than a set amount of reports it will be removed from the playlist
4. If a user receives a report it will be evaluated by the site admins who can decide action from there
5. User that receives a report will either receive nothing, a warning or a permanent ban

Extensions:

2. User accidentally reports a person or song that they wish not to
4. There are no site admins currently online to evaluate reports
5. User receives a ban and then makes another account

Special Requirements: None

Frequency of Occurrence: Users can only report another user one time and an added song one time

Use Case 11.) User votes on top 4 songs in the queue to increase their chance of being played

Primary Actor: Users

Stakeholders: Users

Preconditions: User is logged in and in a genre playlist, there are songs in the queue to be voted on

Success Guarantees: If there are X total votes and a song receives Y votes then the probability that that song gets randomly selected to be played next is Y/X . The chosen song will then play when the current song ends.

Main success scenario:

1. Voting for next song begins 30 seconds before the current song ends with the first 4 songs in the queue as choices
2. Votes next to songs are displayed and users may select one song
3. Voting ends 5 seconds before the next song plays and a song is randomly chosen

4. The chosen song will be played

Extensions:

1. For some reason a song is less than 30 seconds long
2. User votes and wants to switch vote to a different song
3. There are zero total votes for any song, a random song

from the 4 songs is chosen

Special Requirements:

-Voting must take place at the same time for all users

Frequency of Occurrence: Every time a song ends and a new one begins or at the beginning of a stream

Use Case 12.) Songs that do not get voted on after 10 cycles are removed from the playlist and user is refunded a coin

Primary Actor: Users

Stakeholders: Users

Preconditions: A song has remained in the front of the play queue without being voted on for 10 voting cycles

Success Guarantees: The song will be removed from the playlist and the user that added it will get a refunded coin back into their account

Main success scenario:

1. A song has been in the front (top 4) of the queue for 10 voting cycles
2. After the 10th voting cycle of not "winning" the song will be removed from the playlist entirely
3. The user that voted for the song will receive 1 coin back

Extensions:

2. The song is the only song in the queue, the stream stops until another is added

3. The user uses the refunded coin to add the removed song back and process repeats

Special Requirements: None

Frequency of Occurrence: Anytime a song has been through a voting cycle 10 times without being selected

Use Case 13.) Users can type messages to others listening to the same playlist

Primary Actor: User

Stakeholders and interests: System Admin, other users in the playlist

Preconditions: User must be logged in, must be in a playlist,

Success Guarantees: All users in the playlist must be able to see the original message sent

Main success scenario:

1. User is logged in
2. User connects to a playlist
3. User types a message in that playlist's chat

4. User's message is shown in the chat box to every user in the playlist

Extensions:

- 4. Message is shown only to some users, not all users
 - 4a. Internet connection failure
 - 4b. Chat is not updated properly

Special requirements: Chat is updated as messages come in

Technology and data variation list: User has mouse and keyboard to send messages

Frequency of occurrence: often depending on user discretion

Use Case 14.) Users can click another user's name from chat or leaderboard to go to their profile

Primary Actor: User

Stakeholders and interests: User

Preconditions: User must be logged in and on the leaderboard page or playlist page

Success Guarantees: The user can see the profile of the user clicked

Main success scenario:

- 1. User is logged in and on leaderboard page or playlist page
- 2. User clicks a user name from that page
- 3. The profile of the clicked user is shown

Extensions: n/a

Special requirements: n/a

Technology and data variation list: mouse for clicking username

Frequency of occurrence: often at user discretion

Use Case 15.) Users can look at leaderboard for top statistics amongst users

Primary Actor: User

Stakeholders and interests: Site admin, user

Preconditions: User must be logged in.

Success Guarantees: User can see leaderboard for users with top statistics

Main success scenario:

1. User is logged in
2. User clicks leaderboard button
3. User is shown leaderboards

Extensions: n/a

Special requirements: n/a

Technology and data variation list: mouse for clicking the leaderboard button

Frequency of occurrence: often based on user discretion

Use Case 16.) Users can filter the leaderboards

Primary Actor: User

Stakeholders and interests: User

Preconditions: User must be logged in and on the leaderboard page

Success Guarantees: The leaderboard is filtered based on user preference

Main success scenario:

1. User is logged in and on leaderboard page
2. User clicks a filter option for the leaderboard
3. The leaderboard is filtered based on that option

Extensions: n/a

Special requirements: n/a

Technology and data variation list: mouse for clicking leaderboard filter option

Frequency of occurrence: often at user discretion

Use Case 17.) User logs out of their account

Primary Actor: User

Stakeholders and interests:

Site admins; interested because they care about user data

Preconditions:

User has navigated to the application

User has already created an account

User is logged into their account

Success Guarantees:

User gets logged out and their session ends

Main success scenario:

- 1.) User navigates to logout page
- 2.) User is logged out of their account

Extensions:

Special requirements: User data needs to be stored securely

Technology and data variation list:

- User interacts with the application through modern web browsers
- + mouse + keyboard

Frequency of occurrence: Less than one time per user per application visit