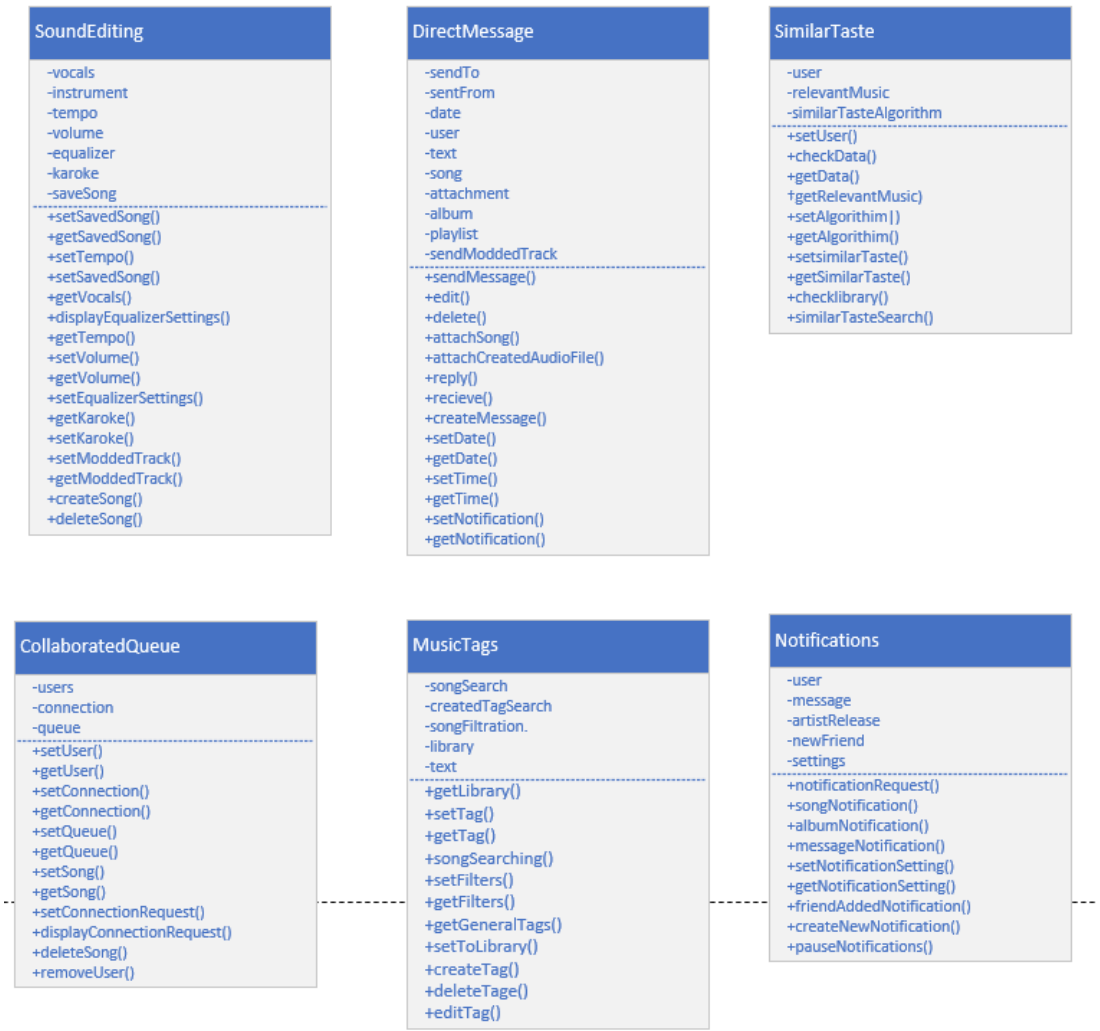
Team RYW

Ryan Catania, Yuness Rachidi, & Wyatt Gockeler
12/15/2023

System Design Document

Introduction (YR):
This document's purpose is to give detailed information regarding each class in Spotify+. The design level class diagram depicts the attributes, methods, and pseudocode for each respective class. The object state charts display each class' potential "states" and how each feature becomes activated. The system sequence diagram depicts the primary interactions between the user and system, detailing each method and its input, as well as the Spotify+'s response to each user input.

Design Level Class Diagram(YR):



Pseudocode:

SoundEditing (RC):

Choose a song;
Press play button;
Turn master volume up to preferred volume;
Scroll down to find volume faders;
Move volume faders up or down;
Press the save button to keep the volume fader position;

DirectMessage (YR):

Select user to message;
Input text message into text field;
Attach song/playlist/lyrics to send to a user;
Send button appears, available to input;
If (send button clicked)
{
    system sends the message to selected user;
}
Else
{
    save message draft;
}

SimilarTaste (WG):

Select similar taste section;
Select desired song;
Song stream, play/skip/scroll button and song duration;

CollaboratedQueue (RC):

Click on the current song that is playing;
Scroll down and press Create Lobby;
Choose how many users can join;
Press create button;
Press the invite button;
Type the user/users name in the textbox;
Press send button;
Wait for users to join;
Add songs to queue;

MusicTags (WG):

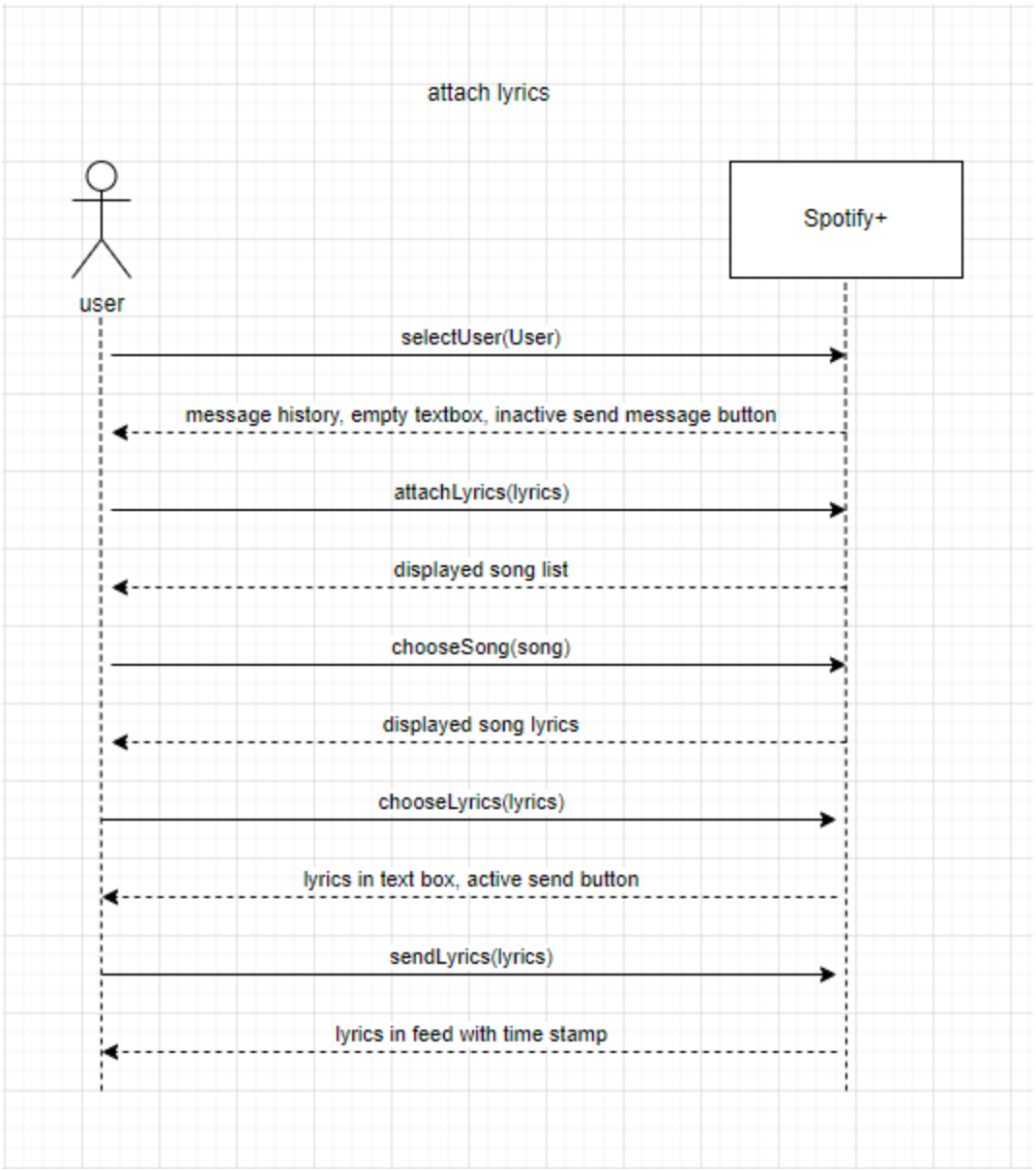
Choose a song;
Select “add tag” button;
Type a name into empty textbox;
Press activated save button;
Sort by created tags;

Notifications (YR):

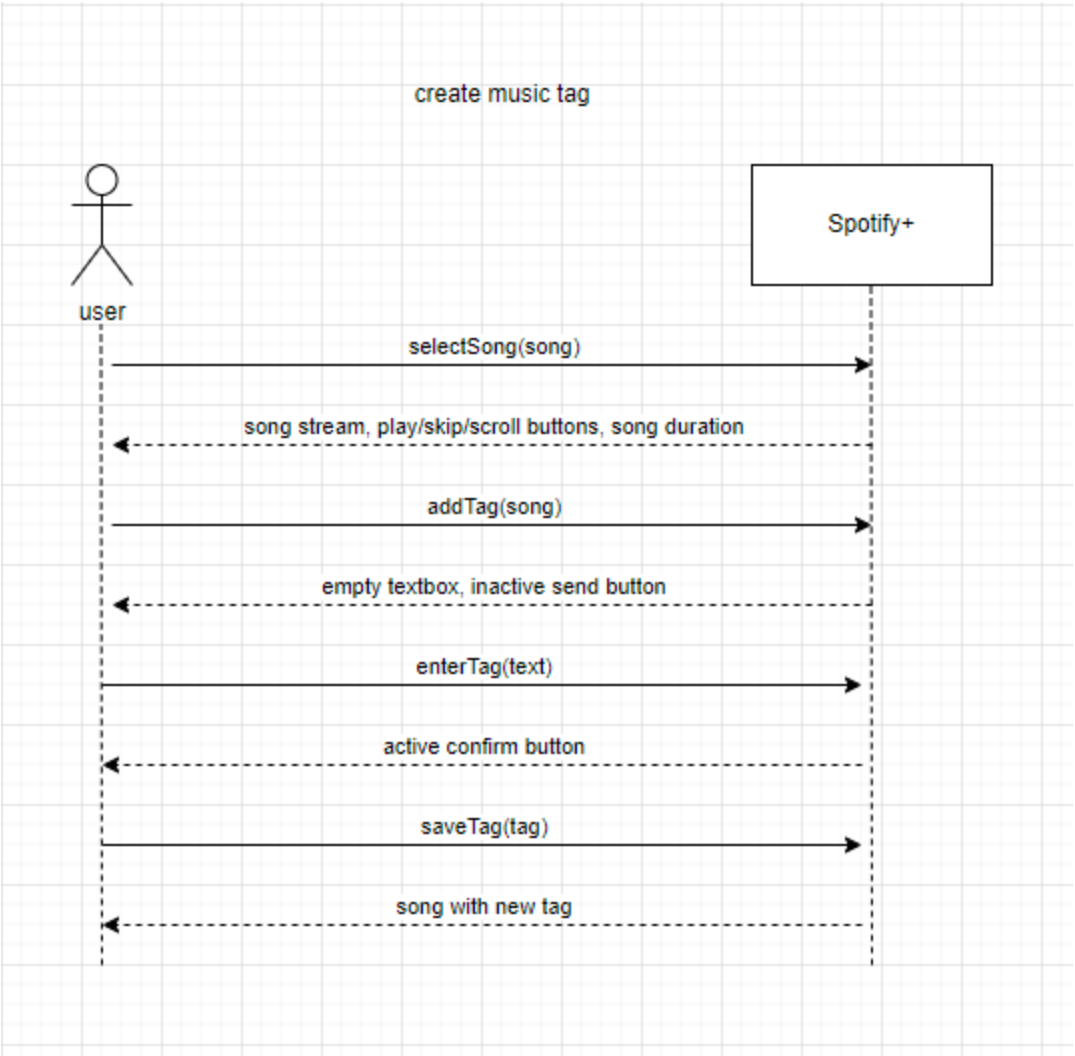
{
Ask if notifications should be pushed;
If (push allowed)
{
Direct message notifications pushed;
Added friend notifications pushed;
Collaborated queue request pushed;
}
Else
{
Don’t push notifications;
}

System Sequence Diagrams:

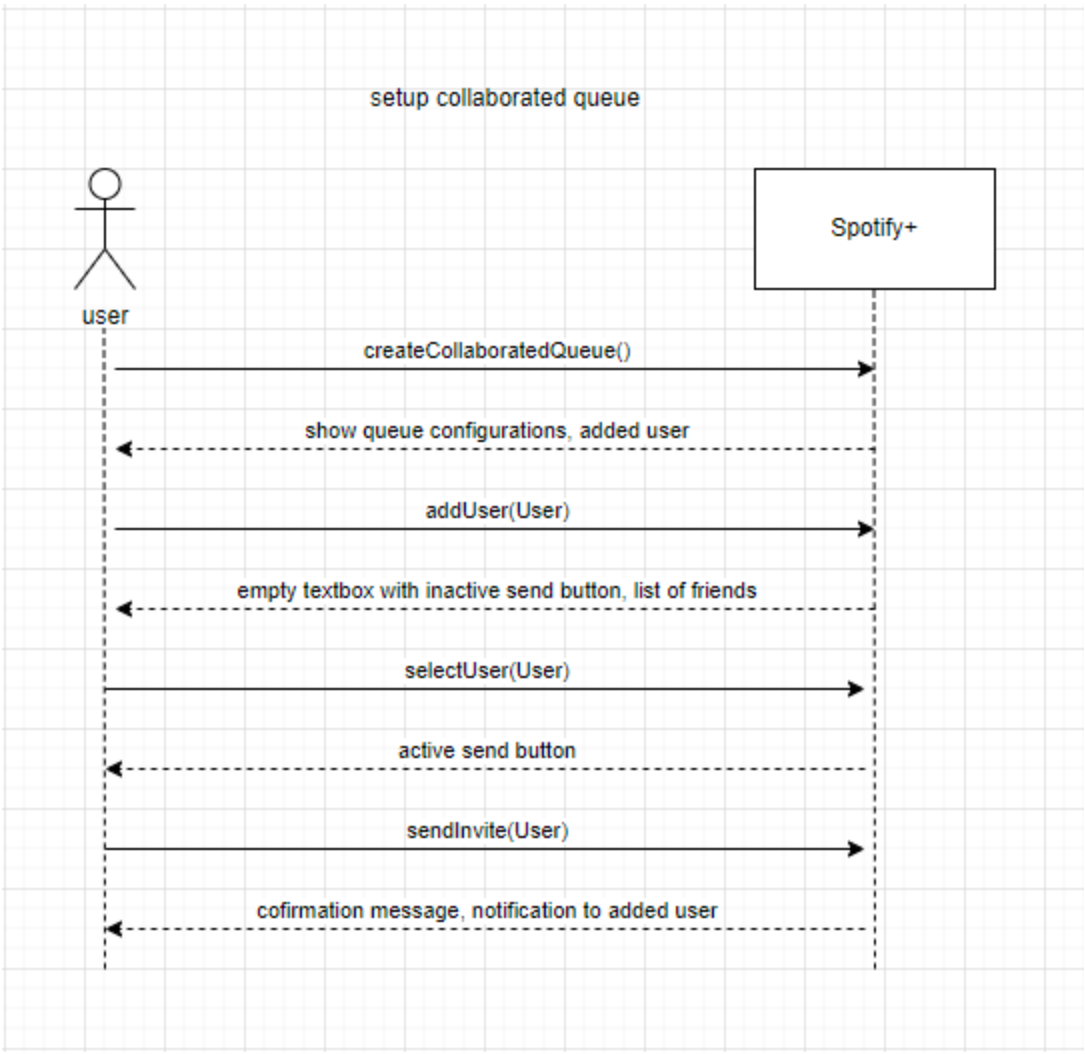
Team RYW


attach lyrics
user
Spotify+
selectUser(User)
message history, empty textbox, inactive send message button
attachLyrics(lyrics)
displayed song list
chooseSong(song)
displayed song lyrics
chooseLyrics(lyrics)
lyrics in text box, active send button
sendLyrics(lyrics)
lyrics in feed with time stamp

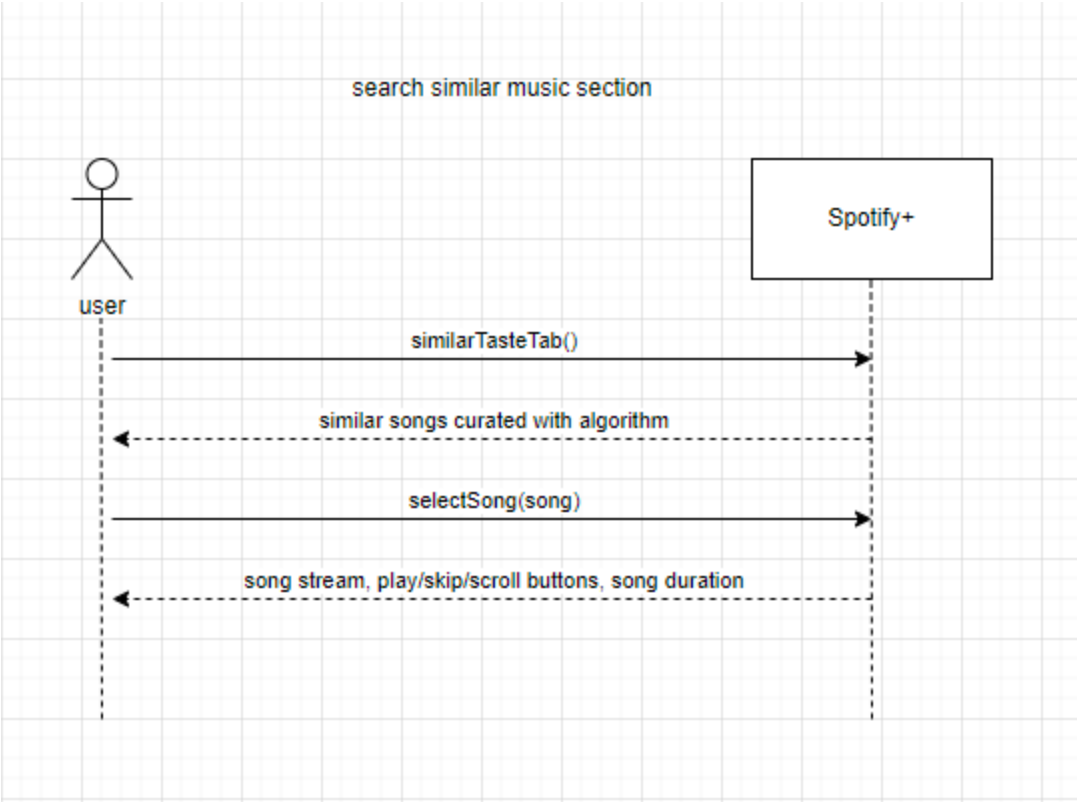
Team RYW


create music tag
user
Spotify+
selectSong(song)
song stream, play/skip/scroll buttons, song duration
addTag(song)
empty textbox, inactive send button
enterTag(text)
active confirm button
saveTag(tag)
song with new tag

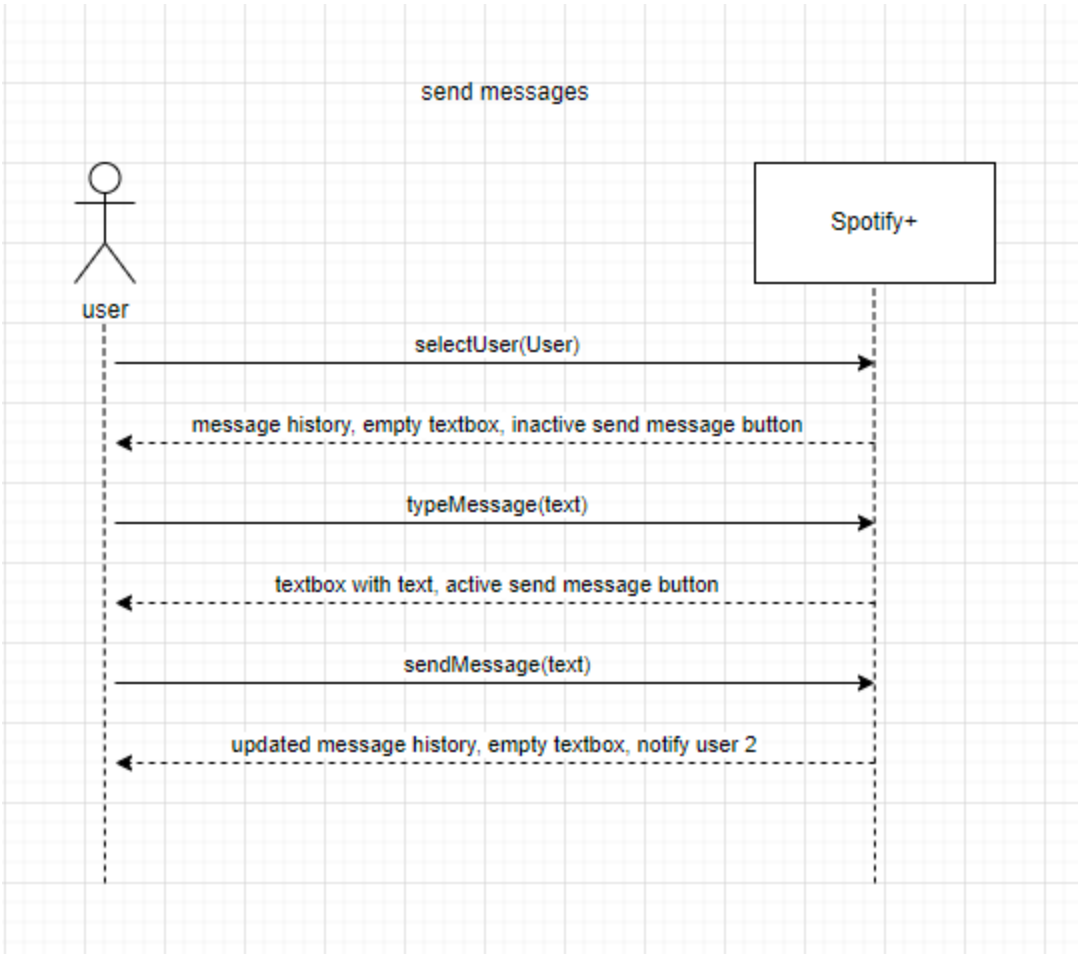
Team RYW


setup collaborated queue
user
Spotify+
createCollaboratedQueue()
show queue configurations, added user
addUser(User)
empty textbox with inactive send button, list of friends
selectUser(User)
active send button
sendInvite(User)
cofirmation message, notification to added user

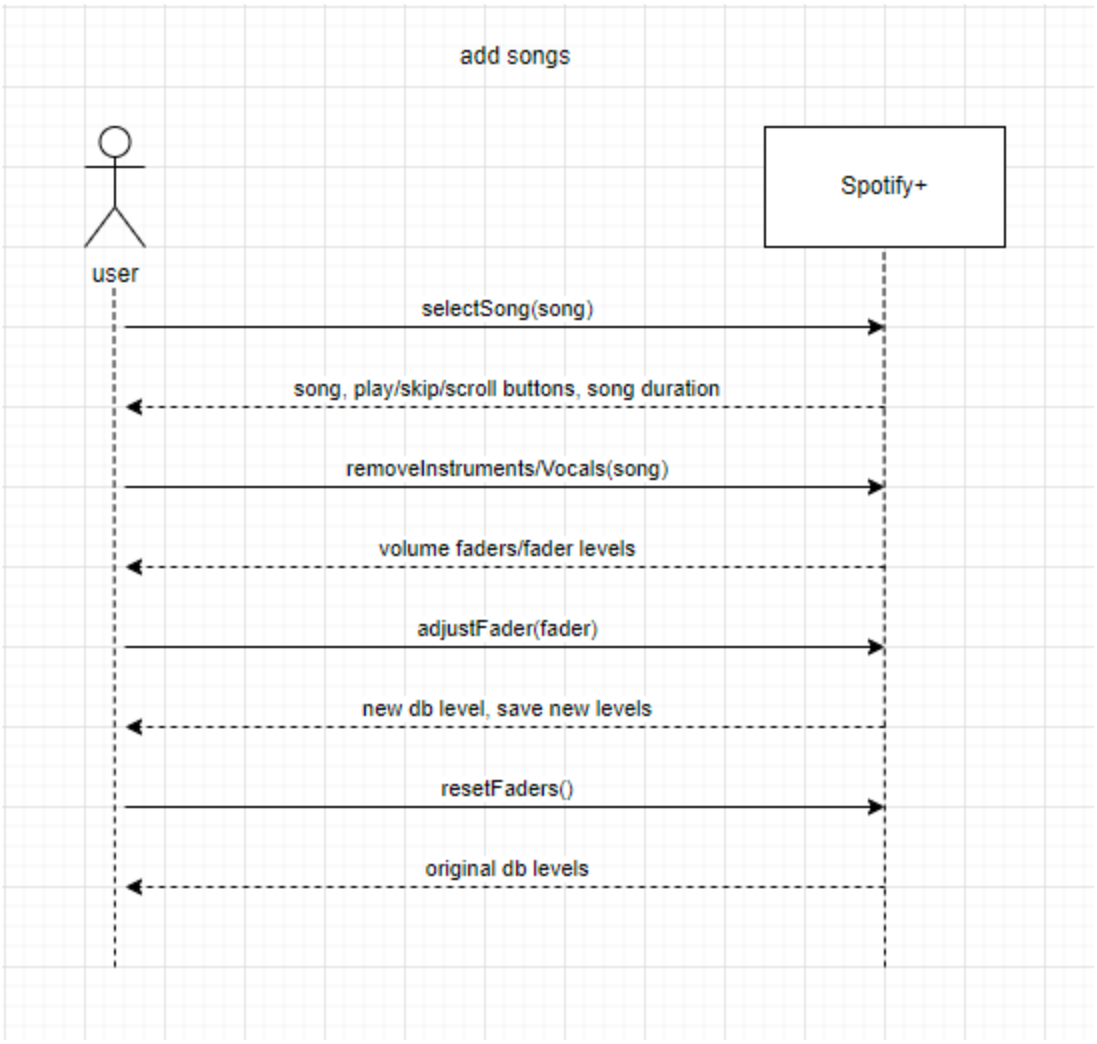
Team RYW


search similar music section
Spotify+
user
similarTasteTab()
similar songs curated with algorithm
selectSong(song)
song stream, play/skip/scroll buttons, song duration

Team RYW

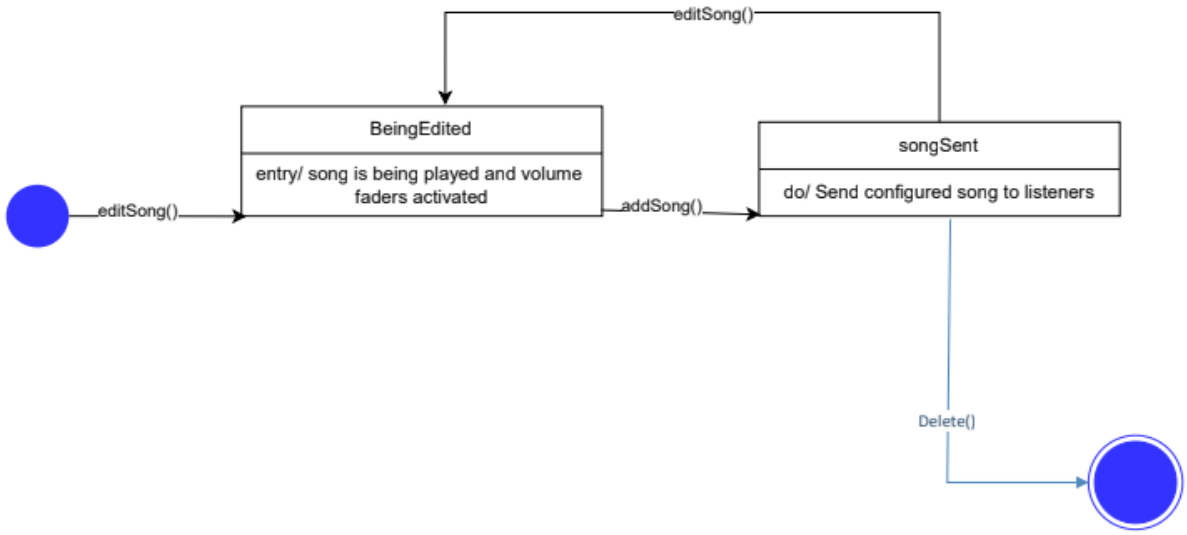

send messages
user
Spotify+
selectUser(User)
message history, empty textbox, inactive send message button
typeMessage(text)
textbox with text, active send message button
sendMessage(text)
updated message history, empty textbox, notify user 2

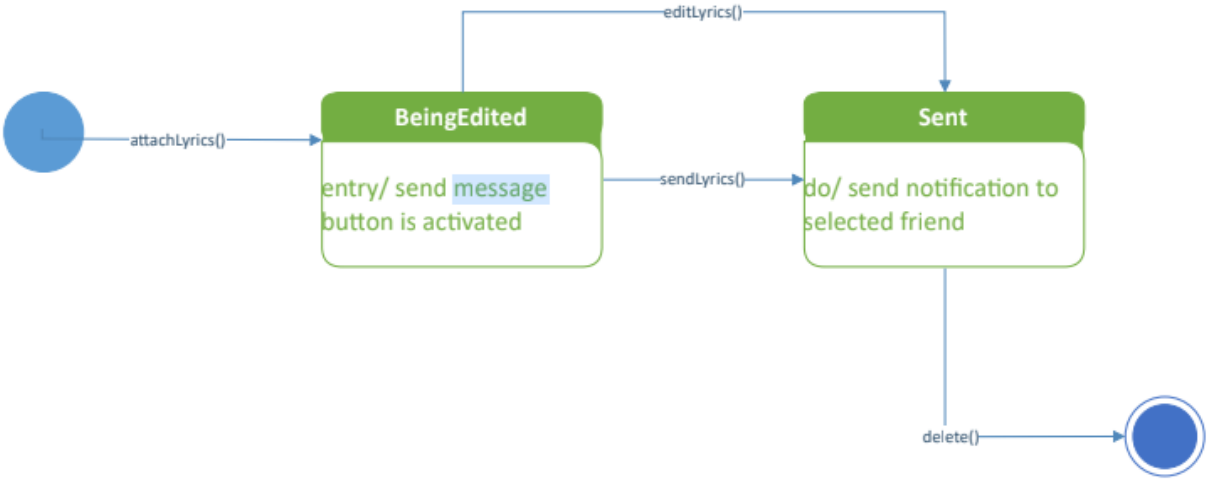
Team RYW


add songs
user
Spotify+
selectSong(song)
song, play/skip/scroll buttons, song duration
removeInstruments/Vocals(song)
volume faders/fader levels
adjustFader(fader)
new db level, save new levels
resetFaders()
original db levels

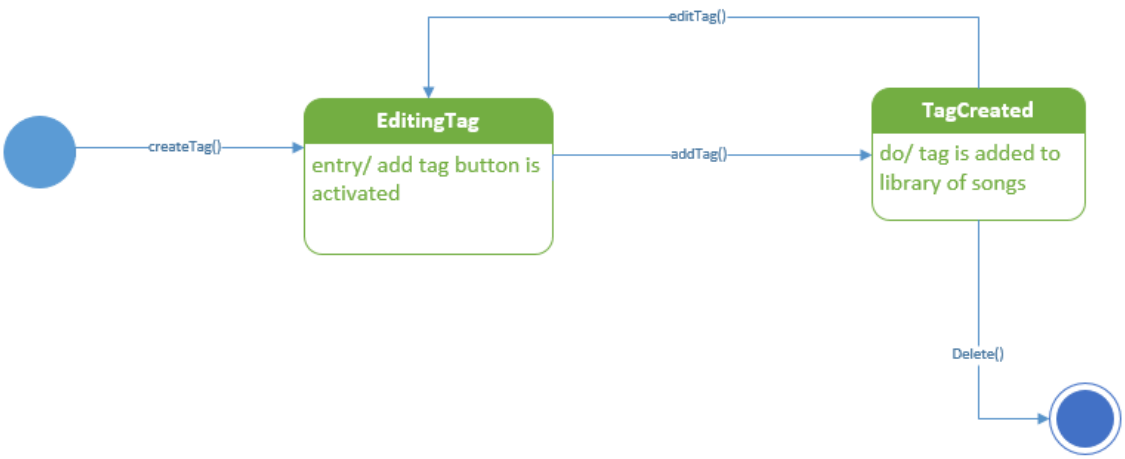
Team RYW

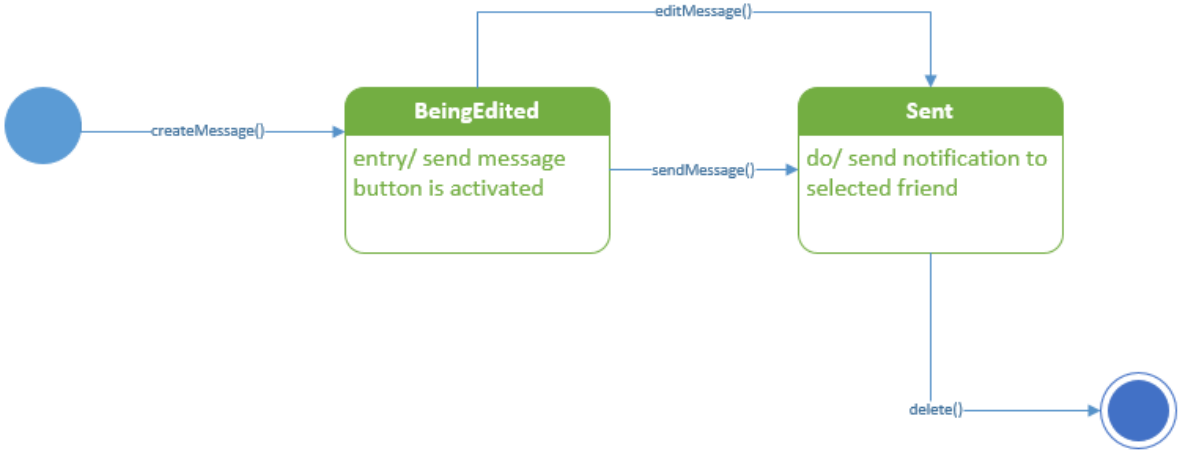
State Diagrams:



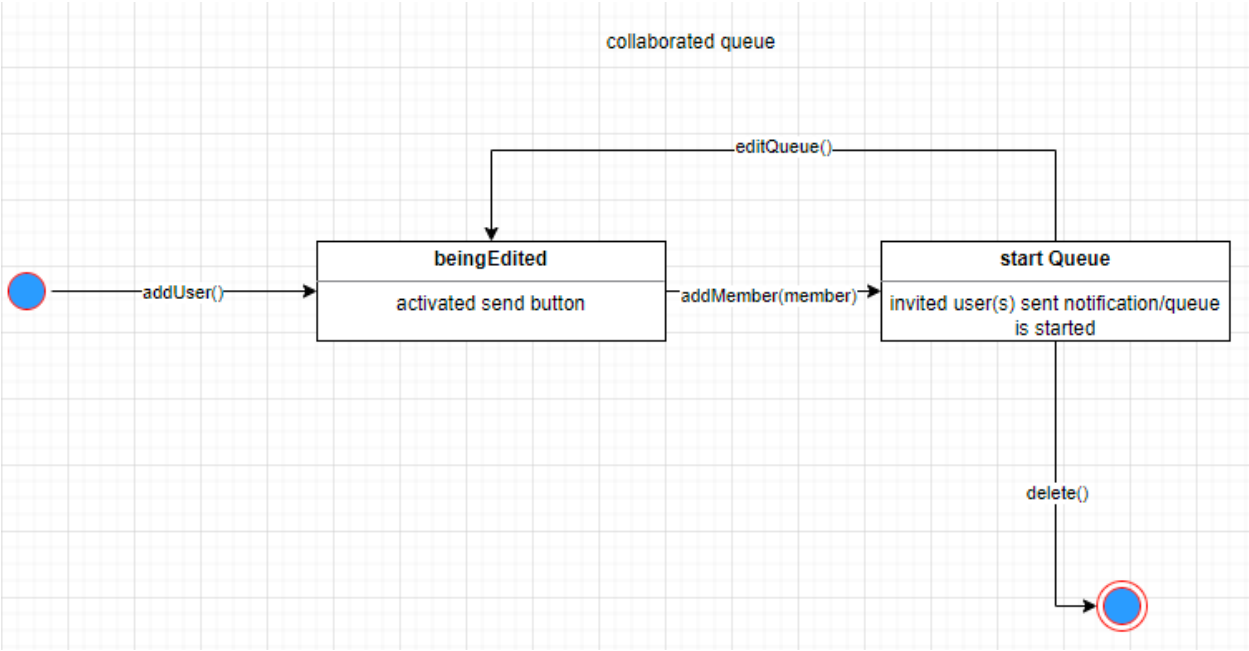


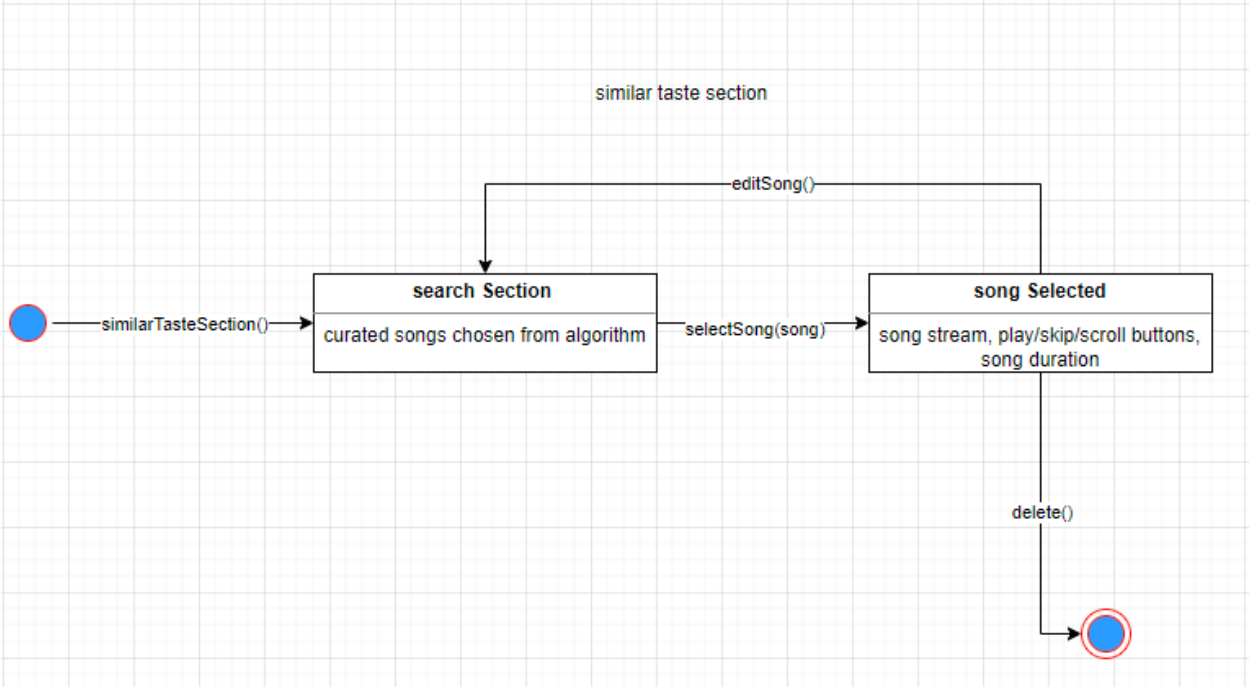
Team RYW


editTag()
createTag()
EditingTag
entry/ add tag button is activated
addTag()
TagCreated
do/ tag is added to library of songs
Delete()


editMessage()
createMessage()
BeingEdited
entry/ send message button is activated
sendMessage()
Sent
do/ send notification to selected friend
delete()

Team RYW


collaborated queue
editQueue()
addUser()
beingEdited
activated send button
addMember(member)
start Queue
invited user(s) sent notification/queue is started
delete()


similar taste section
editSong()
similarTasteSection()
search Section
curated songs chosen from algorithm
selectSong(song)
song Selected
song stream, play/skip/scroll buttons, song duration
delete()