DECEMBER 15, 2023

UPDATED SPOTIFY+ SYSTEM REQUIREMENTS DOCUMENT

TEAM RYW
RYAN CATANIA, YUNESS RACHIDI, & WYATT GOCKELER

TABLE OF CONTENTS

Introduction (WG)

The purpose of this requirements document is to be a guideline for the implementation of the features of Spotify+. The description models act somewhat like pseudocode for the features and explain the purpose and process of adding these features. The class diagrams provide a rough idea of the variables and methods needed to implement each feature. The use case diagram shows how each feature would be connected to each other, and the actions that the features share. The use case scenarios help with determining how the user will interact with the new features of the system, as well as serving as a guideline for how the system will be designed. Finally, the system sequence charts show the sequence of events that would take place between the user and the system when using the new features.

Description Model (YR)

Security: The user must create an account. This account must be linked to an email in case password recovery is ever needed. A password and username will be required. The user must also input their birth date to determine if explicit content filtration is needed. Once signed up/ logged in, the user will have access to Spotify + and all its respective features.

Similar Taste Section Requirements: A user must be on an account with at least 10 songs previously listened to, to begin building a music profile for the user. As a security measure, the entire section will remain locked until that criteria is met. Once 10 songs have been inputted and recorded by the system, the output will be the similar taste section unlocking to the user. The algorithm will produce and display to the user a page filled with songs, artists, and playlists that

are relevant to the user. “Relevance” is determined by the users previously listened to songs, which is why at least 10 songs should be listened to before building a profile for the user.

Socialization of Spotify Requirements : The social page of Spotify + allows the user to input text and attached songs/playlists to post on the page. Once a user clicks send, the output is the successful post by the system, displaying the post onto the feed for friends to see. For a user to begin sending direct messages, they must have a friend added, the message button remains locked until that requirement is met. For the user to add a friend, they must first search the user up by username. There is no confirmation needed from the searched user, anyone can simply add anyone as a friend. A “block user” option is always available to users. Once a user inputs by adding a friend, the system output is the “friend is added, and the direct message option unlocks.” After direct messaging unlocks the user can input text, with an attached song/playlist to send to a user. Once sent, the output by the system is the sent message to the selected friend.

Push notifications Requirements :

For notifications to be enabled, a user must be logged into an account. The system will ask the user for notification permission, if declined the user will receive no notifications. It is required by the system to accept notifications, to begin sending them. This following input and output requires the user to first follow the artist (if not followed, notifications won’t be sent): Once an artist inputs a new song/album to the system that is relevant to a user’s music profile, the system output is that user is notified of its release via notifications. Once a user inputs to the system a

direct message, the output is a notification sent to the user receiving the message. Once a user adds a friend, the added friend will receive a notification stating who exactly added them.

Music Tags Requirements :

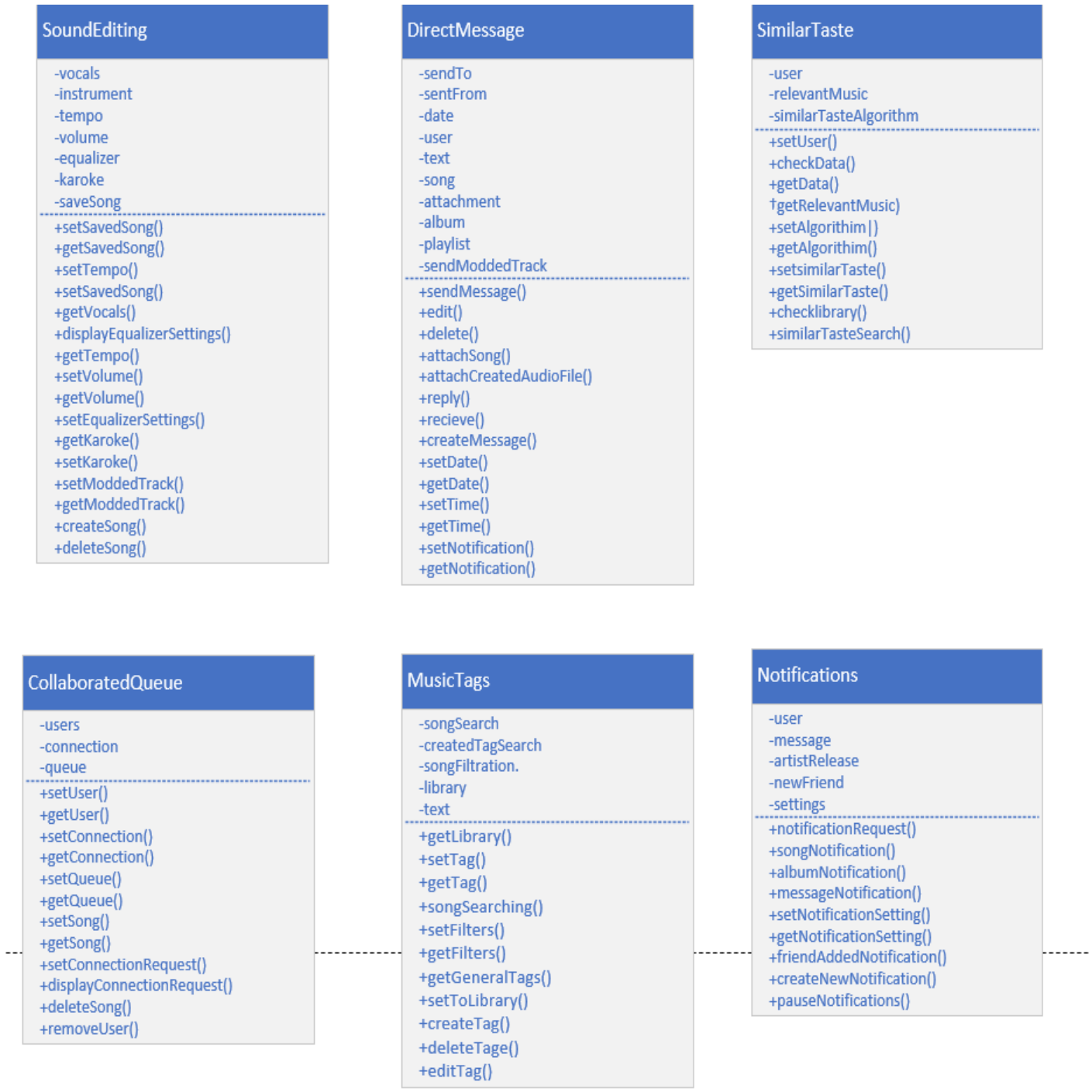
For the ability to set music tags, the user must first be logged into an account. It is required that the user first attains a minimum of 5 songs added in their library, before attempting to categorize them via tags. This is required, as to even begin categorizing you must have enough songs to categorize first, or else there is no point. Once criteria is met the following input/outputs unlock to the user: If the user decides to input by “tagging a song” the system output prompts the user to enter the name of a tag to add the song to. The user can then input a tag name by typing into a textbox, in which the system would output the created tag in the user’s library with the selected song.

Collaborated Queue Requirements :

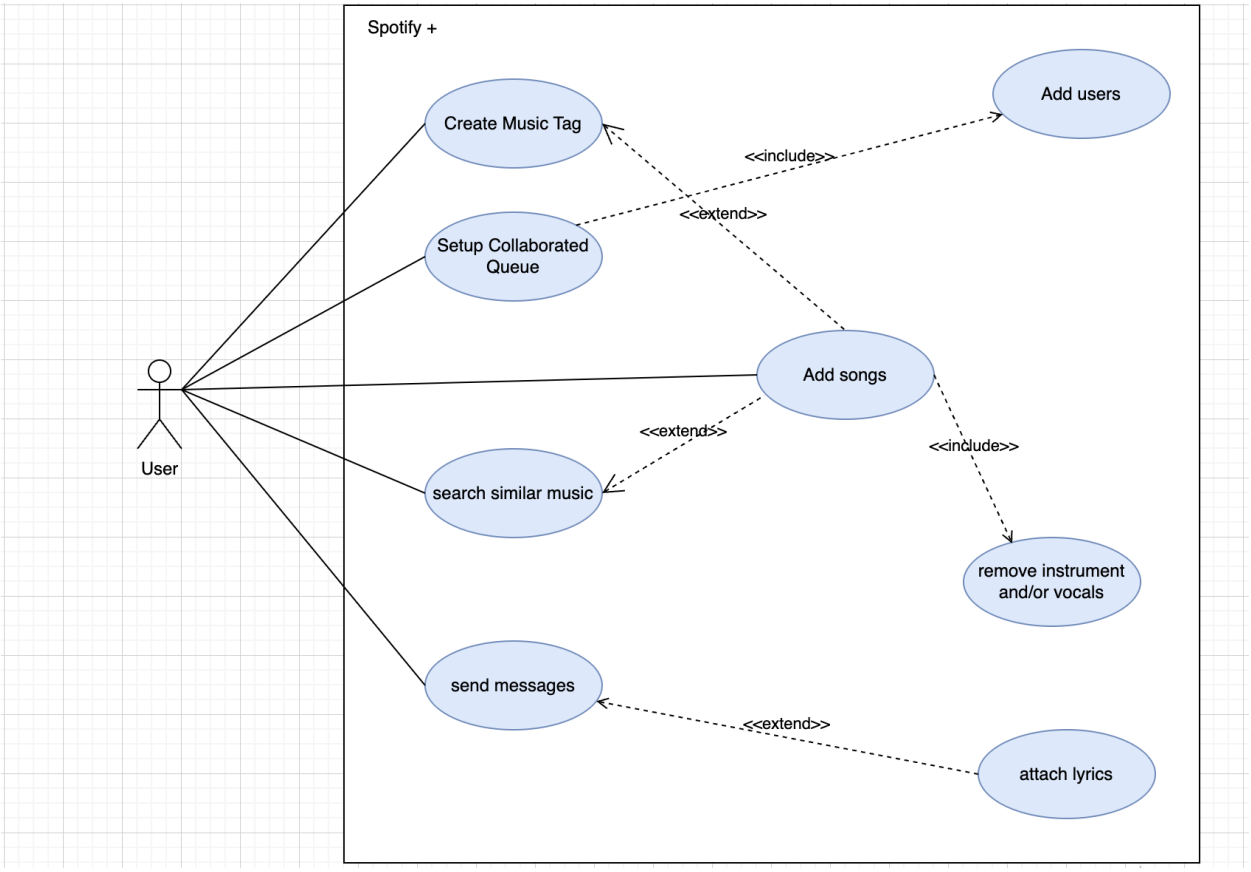
A user can input to the system by clicking on the collaborated queue button, the system outputs by allowing other users to connect. Once a user is connected they can input to the system a song to add to the queue, the output by the system is the song added to the queue’s host.

Sound Editing: A user can input to the system a song they’d like to modify. The output by the system is an interface that allows the user to modify the song via editing sliders. The user can then input to the system by adjusting a slider, in which the system would output the song with the modifications applied.

CLASS DIAGRAM (YR)



Use Case Diagram (RC)



Use Case Scenarios (RC)

Use Case Title	Attach Lyrics					
Primaray Actor	User					
Level	Full Description					
Stakeholders:	Other Users					
Precondition:	Other users account added					
Postcondition	Lyrics is sent to other users					
Trigger:	Select User					
Main Success Scenario:	Actor	System				
1	Click on User	1.1 display user message history				
		1.2 open new message textbox with send message button inactive				
2	Click on Attach Lyrics	2.1 Song list displayed				
3	Choose Song you would like to extract ly	3.1 Display Song lyrics				
	Choose lyrics you would like to send	3.2 Lyrics attach to empty textbook, send message button active				
4	Press active send button	3.2 Draft message is deleted				
		3.3 Lyrics posted in feed with timestamp				
Exceptions						
3.2	If user decides to delete message, then send message button is deactivated, draft is deleted					
3.3	If user decides not to send the message, a. They can leave the unsent message in the channel and it gets saved as a draft, or, b. They can manually delete the text from the box					

Use Case Title	Create music tag			
Primaray Actor	User			
Level	Full Description			
Stakeholders:	Other Users			
Precondition:	Access to app, Songs in library			
Postcondition	Song added to group			
Trigger:	Select song			
Main Success Scenario:	Actor	System		
1	Click on song	1.1 display options to organize song		
2	Click on add tag	2.1 display empty textbox with save button inactive		
3	Type in tag name	3.1 Activate save button		
4	Save tag	4.1 song tag is displayed in library		
5				
Exceptions				
1.1	If song was not originally in library, then it cannot be organized and would have to be added to library			
2.1	If There is a tag already, then previous tag will be displayed with the option to add another			

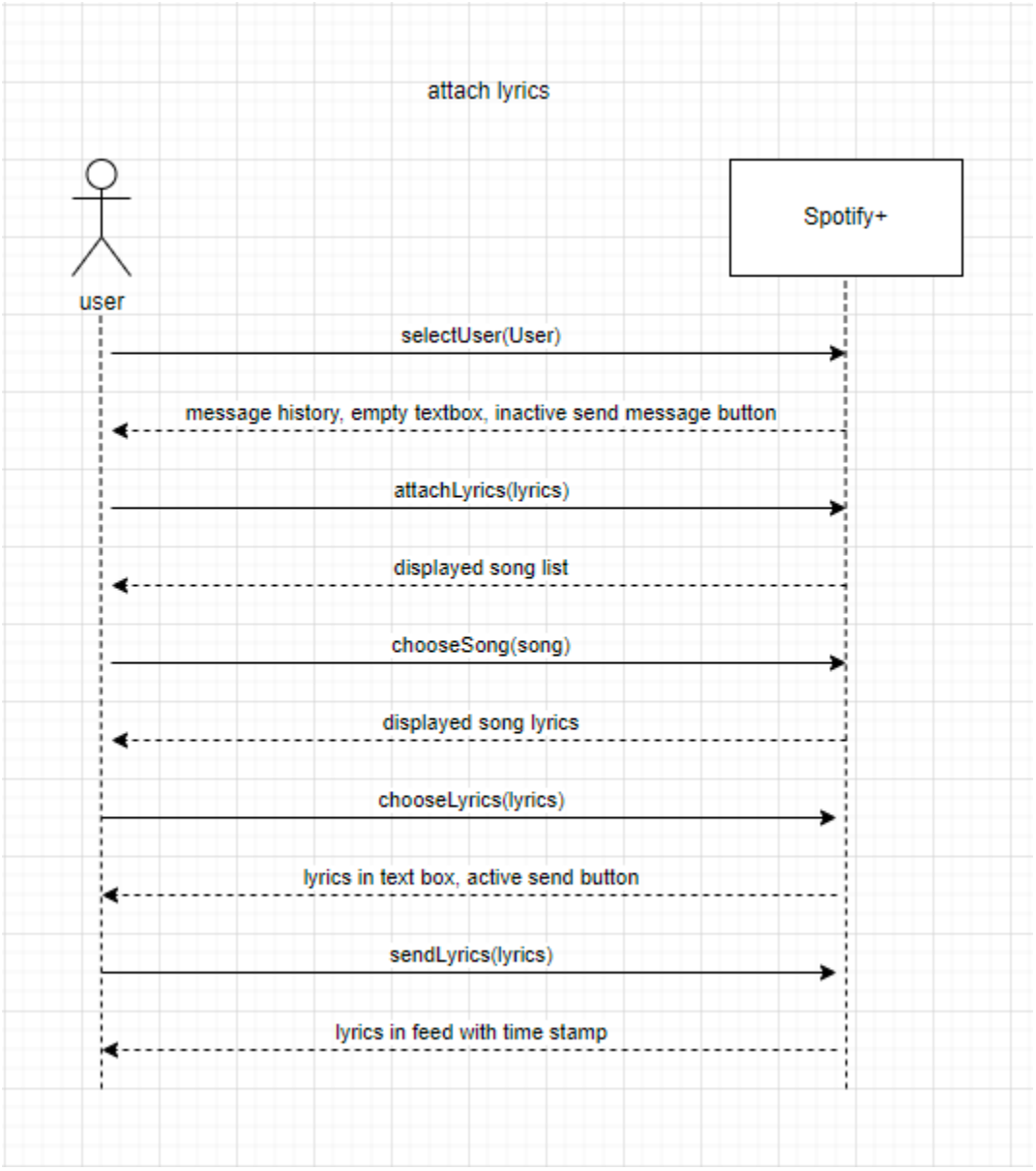
Use Case Title	Setup Collaborated Queue			
Primaray Actor	User			
Level	Full Description			
Stakeholders:	Other Users			
Precondition:	Access to app, Songs in library			
Postcondition	Lobby is created and users join			
Trigger:	Select collaborate queue			
Main Success Scenario:	Actor	System		
1	Select create lobby	1.1 show lobby configurations		
		1.2 user who created lobby is the only one that is in it		
	Call Add Users			
1	Click on add User	1.1 display textbox with deactivated send button		
2	Type in users username	2.1 display user suggestions based off of user input		
		2.2 activate send button when user clicks on name		
3	Send invite to users	3.1 display message that says that the user was invited		
		3.2 other user receives notification about invites		
Exceptions				
3.2	If User is not found, then send button is not activated			

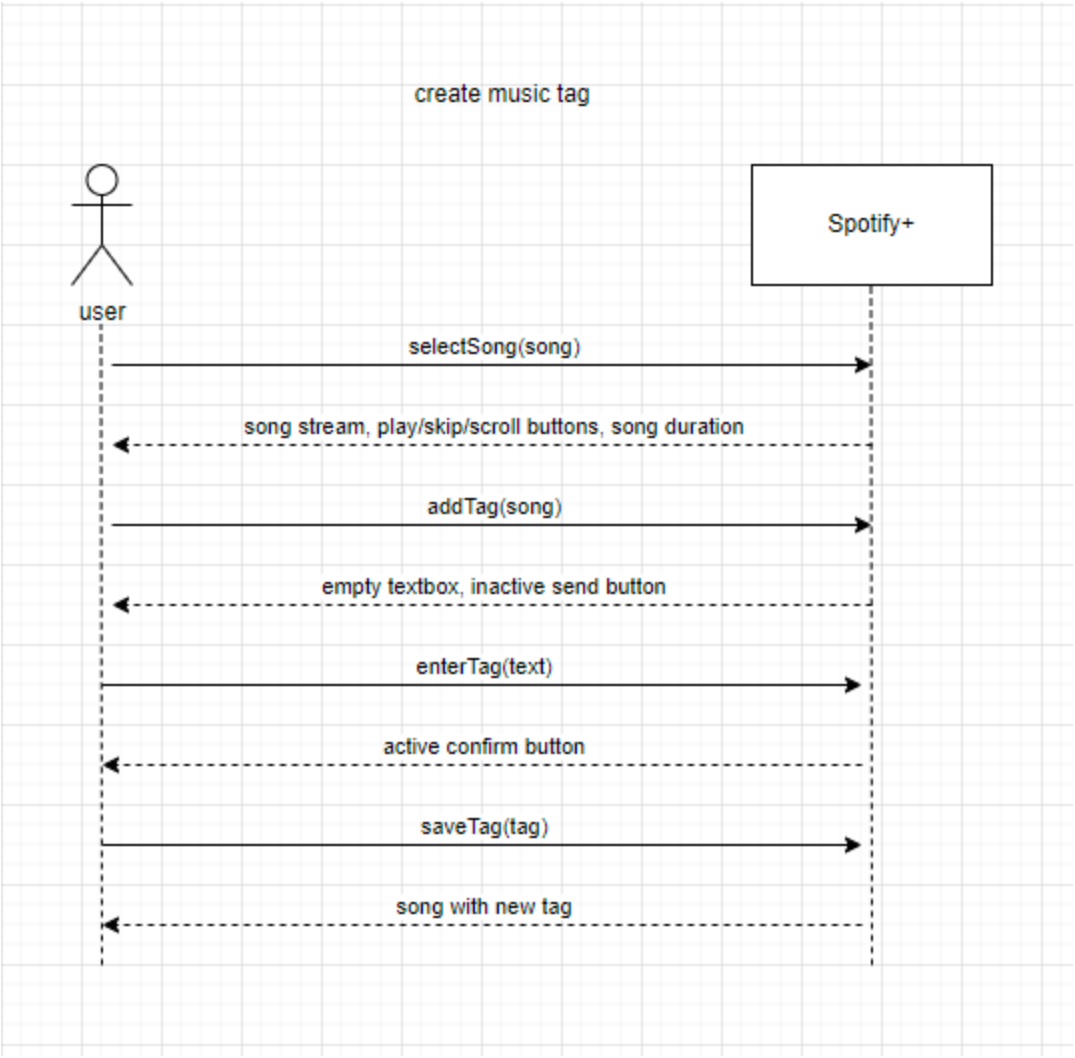
Use Case Title	Search Similar Music section			
Primaray Actor	User			
Level	Full Description			
Stakeholders:	Other Users			
Precondition:	10 Songs previously listened to			
Postcondition	New material discovered, song downloaded			
Trigger:	Select browse tab			
Main Success Scenario:	Actor	System		
1	Click on browse similar taste	1.1 display songs chosen from algorithm		
Exceptions				
1.1	If user has not listened to 10 songs, then the system cannot gather enough data to create a section			

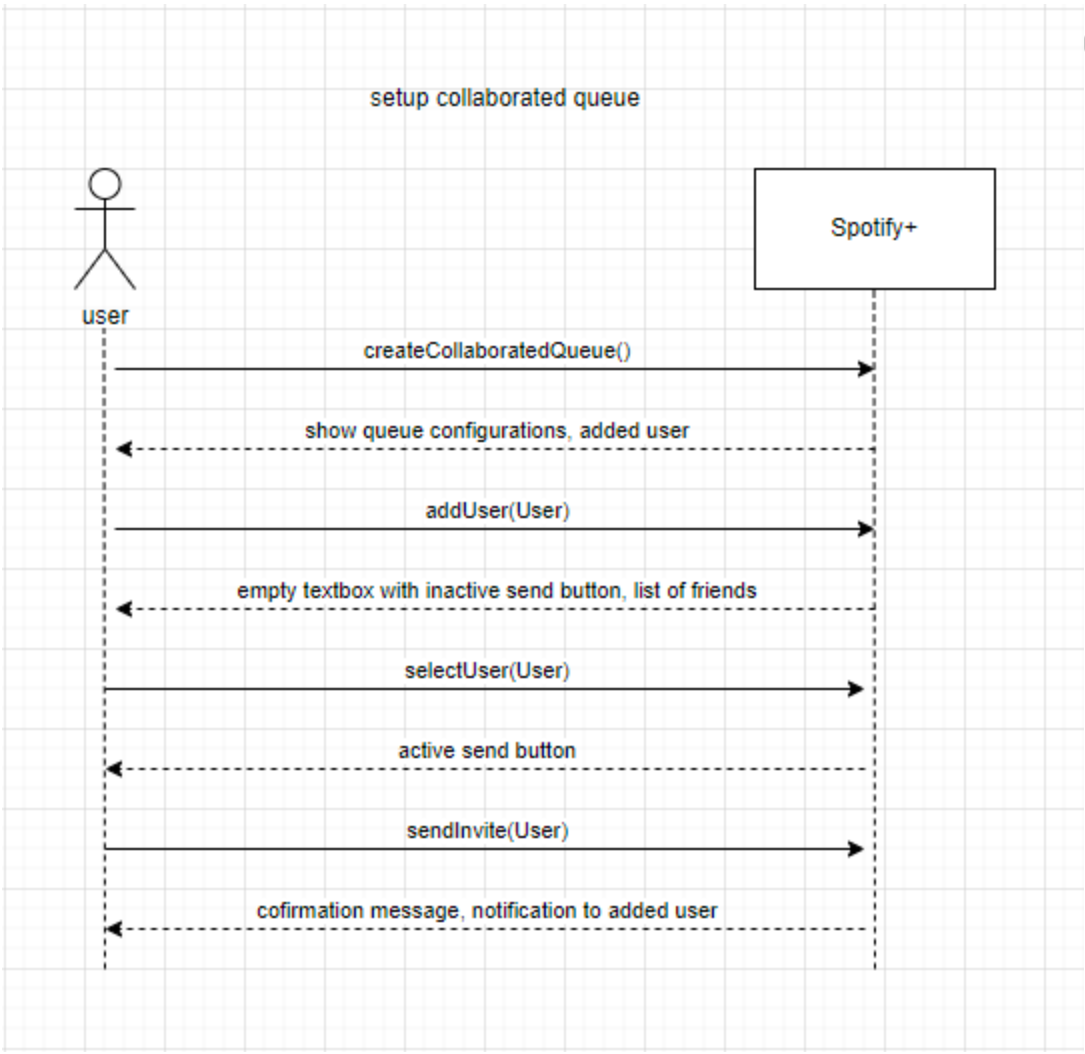
Use Case Title	Send Messages					
Primaray Actor	User					
Level	Full Description					
Stakeholders:	Other Users					
Precondition:	Other users account added					
Postcondition	Message is sent to other users					
Trigger:	Select user					
Main Success Scenario:	Actor	System				
1	Click on user	1.1 display user message history				
		1.2 open new message textbox with send message button inactive				
2	Type in message	2.1 activate send message button				
3	Click on send message button or press e	3.1 post message in feed with timestamp				
		3.2 Draft message is deleted				
		3.3 Notifications sent to user				
Exceptions						
2.1	If user decides to delete message, send message button is deactivated, draft is deleted					
3.1	If user decides not to send the message, a. They can leave the unsent message in the channel and it gets saved as a draft, or, b. They can manually delete the text from the box					

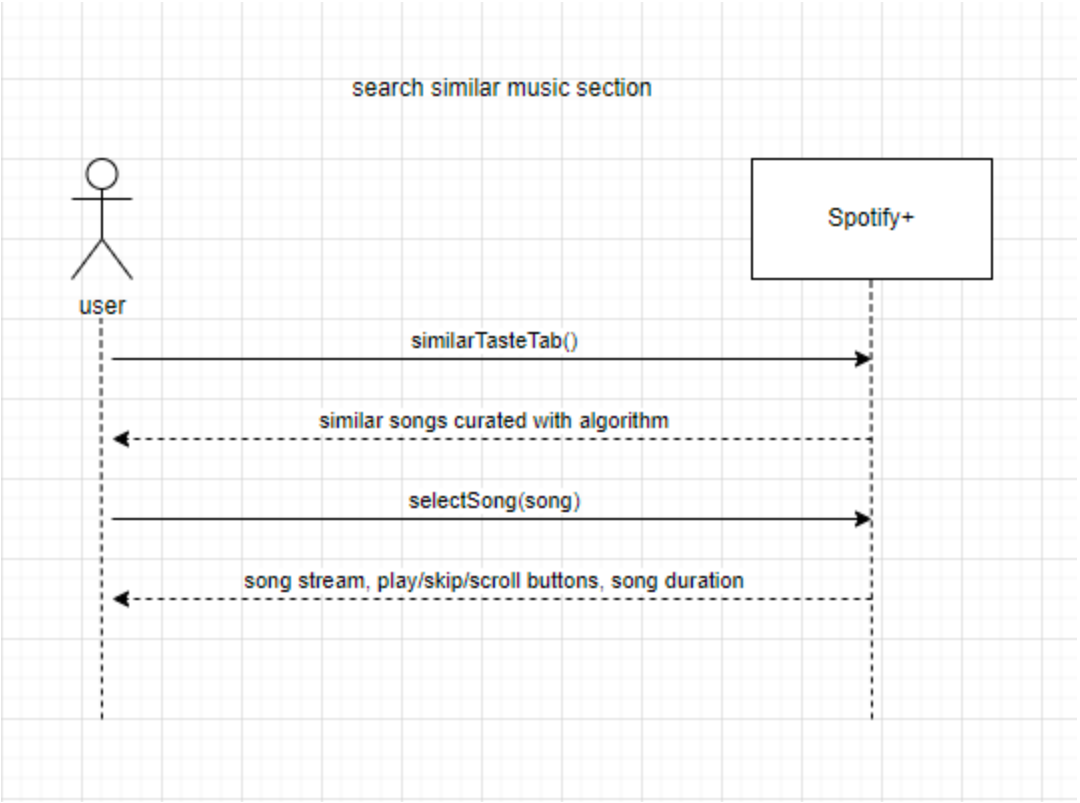
Use Case Title	Add Songs				
Primaray Actor	User				
Level	Full Description				
Stakeholders:	User				
Precondition:	10 Songs saved in library				
Postcondition	Configured song is saved				
Trigger:	Select song				
Main Success Scenario:	Actor	System			
1	Click on song	1.1 display pause, skip, rewind, and song time			
	Click on add to library	1.2 added to library notification displayed			
	Call Remove instrument and/or Vocals				
2	Scroll down to volume faders	2.1 display all volume faders see to 0db			
3	Configure volume faders	3.1 display new db level of fader configured			
		3.2 song is configured to users liking			
		3.3 faders are saved in this position			
Exceptions					
3.3	If User wants to reset faders, then they must press reset to put faders back to original configuration				

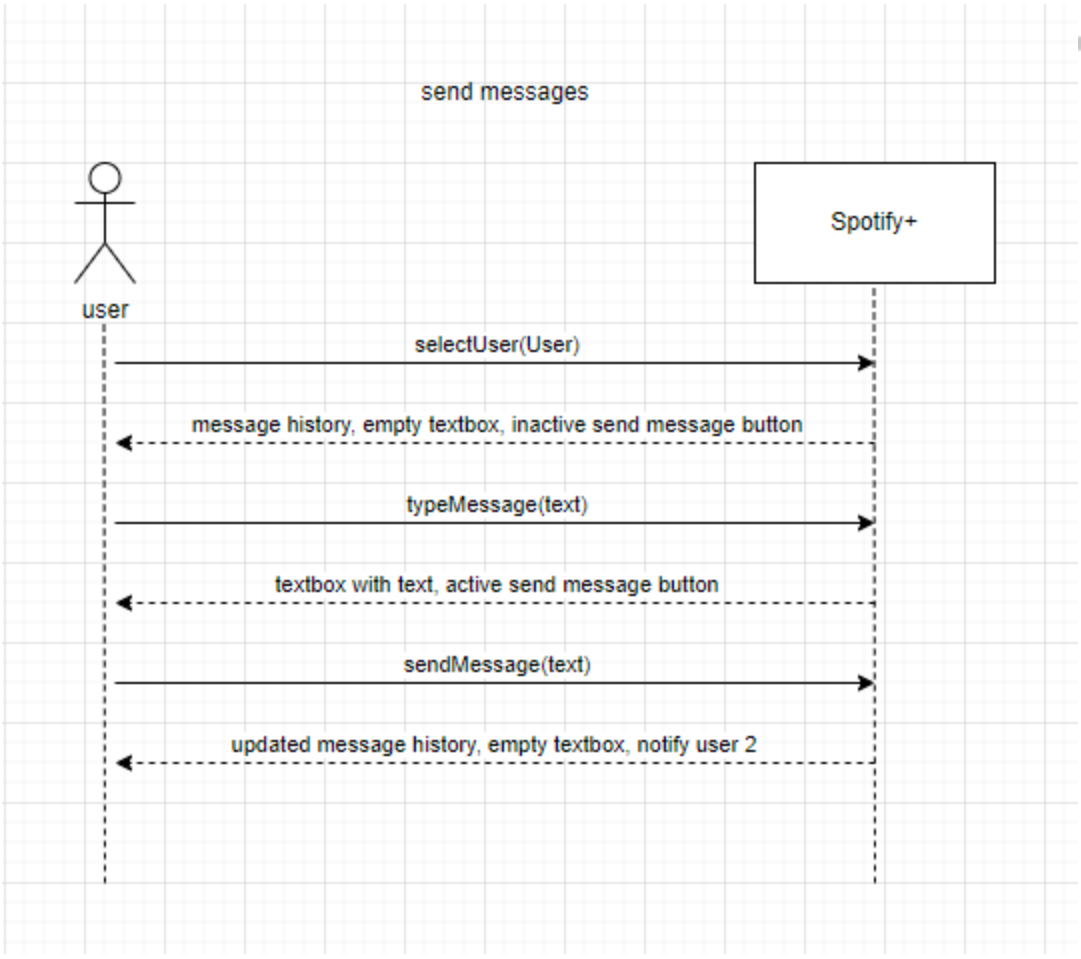
System Sequence Charts (WG)

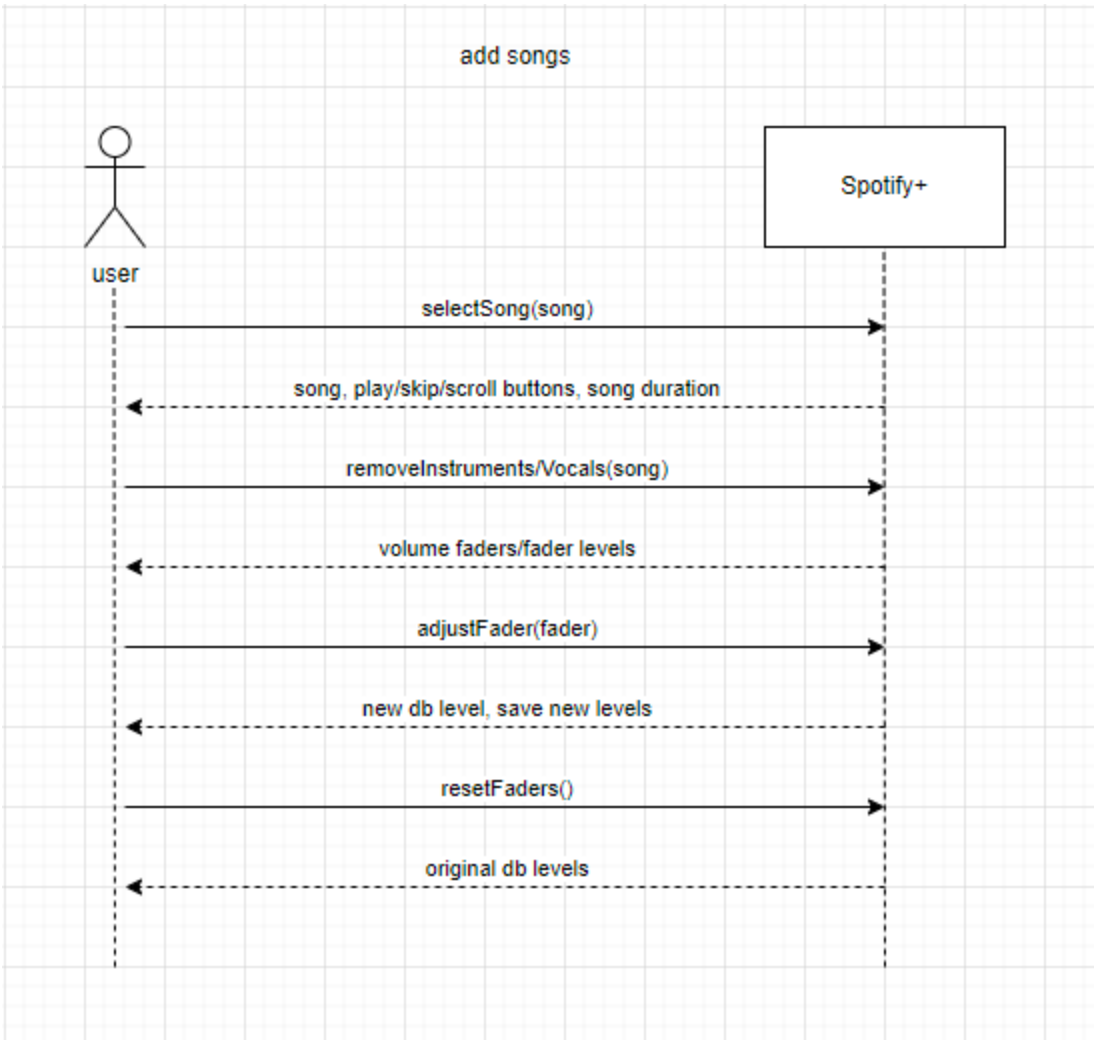



create music tag
user
Spotify+
selectSong(song)
song stream, play/skip/scroll buttons, song duration
addTag(song)
empty textbox, inactive send button
enterTag(text)
active confirm button
saveTag(tag)
song with new tag


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


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


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


add songs
Spotify+
user
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