Incremental Testing and Regression Testing Sprint 2

Product: SpotiBot

Inspectors: Jimmy Carlson, Janka Gal, Puja Mittal, Jason Shipp, David Worley

1. Classification of Components

1.1 Define all components

Here we will define all components that were not defined in Sprint 1. To reference those components, please see the first Incremental Testing and Regression Testing document.
app.js - handles backend for the bot: user authentication and communication

- createPlaylistForCategory() - creates playlists based on different categories: mood, genres, artists, songs, and playlists.
- handleTopPlaylist() - helper method to call node-spotify-api, gets the users top tracks using getTopSongs based on how many songs the user requests (1-50), creates a new playlist with a formatted name, then passes in the songs retrieved before sending back the playlist url to the user.
- getTopSongs() - gets the list of 1 - 50 top songs from a user in a specific time range (short - four weeks, medium - 6 months, long - a few years)

1.2 Which form of incremental testing did you follow

For this sprint, we decided to continue following bottom up incremental testing and tested our code in smaller components. Our bot works by giving the user access to numerous smaller commands and functions so it makes sense to test them as they are being written and to write specific test cases for individual functions before they are combined with the rest of the application.

2. Incremental and Regression Testing

app.js

Module - User Flow

Incremental Testing

Defect No.	Description	Severity	How To Correct
1	Users would be able to access commands that they should not have been able to according to our flow chart	2	We switched from a case statement format to an if-else format
2	All the byop [...] commands were not getting to the proper place in the code and errored out	3	Modified how the input validation was occuring for byop commands

Regression Testing

Defect No.	Description	Severity	How To Correct
3	When typing ‘byop ?’ we would get an error message rather than the informational message	2	Added a separate ‘byop ?’ else if
4	When typing ‘?’ both the proper message and the error message would pop up	1	Added an else if case for just ‘?’
5	We realized after our edits that if a user were to _slightly_ type our commands for byop incorrectly then the system will most likely error out on an uncaught error based upon our reliance on the existence of the "' character, as we split the	3	Create pseudo redundant code which is able to deal with the existence or non—existence of that ascii character and make sure to pass it through a series of if chains

	string based upon its existence.		which are comfortable dealing with each situation. Users should not have to worry about their inputs being very similar and still failing.

Module - Create Playlist Based on Songs

Incremental Testing

Defect No.	Description	Severity	How To Correct
6	When a playlist based on a song is created, sometimes the wrong song names are used to generate a playlist because different songs can have the same name	2	We decided to fix this by requiring users to type in the format “[song name] by [artist name]”. This way we could make sure to find the exact song the user wanted to find.

Regression Testing

Defect No.	Description	Severity	How To Correct
7	This change in format caused a bug where the playlist generated based on songs crashed.	1	We realized this was because the “[song name] by [artist name]” format was not being parsed correctly. Once we fixed the parsing error, we were able to accurately find songs by artists.

Module - Create Playlist Based on Genres

Incremental Testing

Defect No.	Description	Severity	How To Correct
8	Inputting multiple genres didn't work because it parsed them as one genre	2	We changed the way we spliced the genres so that they were read as individual genres and could be passed as a category clearly.

Regression Testing

Defect No.	Description	Severity	How To Correct
10	If entering multiple word genres (ie. "pop rock") it would parse them as multiple genres	1	Specify that in multiple word genres to add a hyphen (ie. "pop-rock)

Module - Create Playlist Based on Mood

Incremental Testing

Defect No.	Description	Severity	How To Correct
11	Our process for curating songs based on moods was originally based on users' songs which was extremely inaccurate.	3	We pulled from Spotify's mood playlists instead and gave more accurate results.

Regression Testing

Defect No.	Description	Severity	How To Correct

12	If you enter multiple mood queries, and one does not work the whole search quits	3	When queries fail, don’t kill the search.

Module - mongdb database

Incremental Testing

Defect No.	Description	Severity	How To Correct
13	An error was found in our current mongodb driver in which the order and amount of documents returned by the query was suedo random based solely upon whether or not the string was semi contained in the id.	3	Instead of trying to access the pseudo—primary key generated by mongodb for the document title (users._id), we instead should and now query the proper key which is registered to the user,"sender_psid": user.id, through the mongo database driver.
14	The mongodb driver failed to return a properly jsonified object to through its promise.	3	Instead of using mongo's native find function on a collection, we first pass it through their personal version of JS mapping which is their toArray function which returns its own resolvable promise in the form of a javascript object.

Regression Testing

Defect No.	Description	Severity	How To Correct
15	Once we started testing our database, we realized the mongodb driver lacked the ability to be called in a synchronous fashion for the proper data flow to to pass without account crossover for any of the functions allowed by the chat bot.	3	We created asynchronous versions of the exported driver functions then use awaits and chained promise callbacks (.then()) to ensure the exact execution of the code flow. In this way we can assure that the oauth flow properly calls the spotify api with the correct and verified refresh and access token that is required by the spotify backend as well as executing the api call we actually require.

Module - Create Playlist Based on Playlists

Incremental Testing

Defect No.	Description	Severity	How To Correct
16	Playlist privacy variation affects the bots ability to forward pass the uri. If a playlist is private, there is more work required.	2	We needed to unpack the tracks in each playlist and pass it as a list of tracks as opposed to each playlist.
17	Creating a playlist based on playlist name was not returning anything.	3	We realized the playlist names were not being parsed

			correctly and fixed that by making sure they were retrieved with no additional white spaces at the ends.

Regression Testing

Defect No.	Description	Severity	How To Correct
18	The command still failed sometimes and we realized it's because users need to input the playlist name exactly otherwise the API cannot find it.	3	We decided to add a comment telling users to copy-paste the playlist name to make sure that the playlist name was inputted exactly as it appears in spotify..