Seeded Defect Log

Adam Johnston, Brian Duffy, Akhil Agrawal, Shivan Desai

Defect #	Issue	Suggested Fix	Location
1	Unauthenticated users can vote on tracks	Make sure the spotify user id is not the default of ‘unidentified_user’	home.component.ts/ upvote
2	Users can see the database without logging in (may need to elaborate)	In the html, insert *ngIf="spotifyUser.id != 'unidentified_user'" in elements that should not display	home.component.html
3	There are no upper and lower bounds to the vote count.	Set upper bounds to Number.MAX_SAFE_INTEGER and lower bounds to Number.MIN_SAFE_INTEGER	Home.component.ts /upvote and downvote
4	Upon pressing the login button, users are immediately redirected to the home page before they finish spotify authentication	Adding the keyword await before the call to in the async onSubmit()	login.component.ts/ onSubmit this.spotifyService.authenticate();
5	New browser opening, user must login twice.	Adding the keyword await before the call to in the async onSubmit()	login.component.ts/ onSubmit this.spotifyService.authenticate();
6	Sometimes users cannot login due to a defer timeout issue with the authentication window.	Increase the time of the timeout for the authentication window.	angular5-spotify.ts
7	Upon successfully logging in, users cannot play the playlist without going to another page then back to the home page.	Adding the keyword await before the call to in the async onSubmit()	login.component.ts/ onSubmit this.spotifyService.authenticate();
8	User can attempt to search without properly authenticating.	We added a check to see if the spotifyUser id was ‘unidentified_user’ if so we showed an error message and	Radio.component.ts /onSearch()

		terminated the search request.	
9	Users can search undefined and the request will go through normally and return songs related to ‘undefined’.	We added a check to see if the user had entered anything in the search bar. If they didn’t then we let them know through an alert dialog box and terminated the search request.	Radio.component.ts/ onSearch()
10	Users could type in a search term then delete it to where nothing is in the search bar. Their search term would go through as “” an empty string.	We added a check to see if the queryterm defined in radio.component.ts was an empty string. If so, we opened an alert dialog box and terminated the search request.	radio.component.ts/ onSearch()
11	Users can add duplicate tracks to the database which is unintended behavior.	We added a check that searched through the database track list and checked if a specific id of a track already existed, then we would terminate the add track request and alert them via dialog box.	Radio.component.ts / onAddTrack
12	When clicking the upvote button the vote amount decreases.	Make sure that the function that is associated with the upvote button is in fact the upvote method.	home.component.html
13	When clicking the downvote button the vote amount increases.	Make sure that the function that is associated with the downvote button is in fact the downvote method.	home.component.html
14	When clicking the search header tab, the user is redirected to the login page instead of the search page.	Make sure that the router navigation has a router link of “/radio” instead of “/login”.	header.component.html
15	When clicking the login header tab, the user is redirected to the search page instead of the login page.	Make sure that the router navigation has a router link of “/login” instead of “/radio”.	header.component.html
16	The Please Login button shows up even after a user logs in.	Add an *ngIf statement inside of the <p> checking if spotifyUser.id == ‘unidentified_user’	home.component.html

17	When a user is playing a song through spotify and they navigate to another page and then back to the home page, when they click play again, they are sent back to the first song.	A possible solution to this would be to store the current song's id so when the embedded player is loaded, it can play that song from the playlist instead of going back to the beginning.	home.component.ts
18	The currently playing banner only shows up after navigating back to the home page.	Add a check to see if the spotify player is currently playing a song. If so, update that variable so the correct information is displayed.	home.component.ts
19	Upon refresh of the page, the spotifyUser.id is no longer recognized. So they need to log back in again.	A possible fix to this would to store the user's spotify id into localstorage and upon the loading of a page, check that if they already have logged in before.	Home.component.ts / radio.component.ts
20	The currently playing banner does not update when a user skips to the next song.	We could possibly watch elements inside of the iframe. More specifically the song name element and store it locally and bind it to the currently playing banner with angular services.	home.component.ts
21	The ordering of songs in the playlist seems to sort alphabetically when the votes are equal to one another while the website has a different ordering than alphabetical.	Adjust the sorting algorithm to treat tracks with equal votes by sorting them alphabetically.	home.component.ts / login.component.ts
22	The button to go back a song doesn't work. It may have something to do with non premium listeners.	If the user is a non premium user we should alert them that they need premium to go back a song.	home.component.ts
23	When a user gets to the last song, they cannot skip to the next one. This means that they cannot listen to the playlist again from the beginning.	When no more tracks are being played and the last track has finished playing we could log them back in and this restarts the playlist. (Spotify doesn't support playlist repeating. This was a work around we discovered.)	Home.component.ts / login.component.ts / angular5-spotify.ts
24	Spotify sets a limit to the maximum number of tracks that	In the search/radio component we should check to see the size of	Radio.component.ts /

	can be in the embedded player’s playlist. That maximum being 200.	the database before adding another track and make sure that the user knows there is a maximum.	angular5-spotify.ts
25	The mobile version of the header component has a menu button that doesn’t collapse the menu after a user selects a page to navigate to.	We could add a button listener that collapses the menu when one of the choices is chosen	Header.component.ts / header.component.html