Groupify Problems Report

The Groupify Development Team

Updated as of 5/30/2021

Known Bugs and Incorrectly Working Functions

1. **Add or remove friend action will reload the webpage when adding a friend from the page, making for filtering/searching for users difficult**
    a. Input/action that causes failure:
        i. Adding or removing a friend immediately from the add_friend end page causes the page to reload, removing the search filters set in place to find users, and sending the user back to the top of the page.
    b. Location of fault
        i. Controllers.py addFriend() and deleteFriend() functions.
    c. Possible action for removal
        i. Refactor the action of adding or removing a friend from add_friend into vue.js as opposed to our Python Controller. This will store local variables and allow the page to store items like the filtered list, or the text entered into the filter search box, and the page may update without needing a reload. This would require an add_friend.js to be the bridge between add_friend.html and the add_friend() / deleteFriend() actions in controllers.py.

2. **Current logged-in user appears on the add_friend page.**
    a. Input/action that causes failure:
        i. Searching for your own username on add_friends.html, and clicking the add-friend button on your resulting tile. The button will not add yourself as a friend.
    b. Location of fault
        i. Add_friend.html.
    c. Possible action for removal
        i. While displaying all users in Groupify's database, add a conditional statement to skip over the current user's tile, as to not display their own profile on the page.

3. **Database is locked error, resulting in a 500 Error.**
    a. Input/action that causes failure:
        i. Groupify attempts to access a Py4web database that had been locked while cashing.
    b. Location of fault
        i. Py4web's databases directory. This fault lies with the py4web database structure, not Groupify's implementation of the database.

c. Possible action for removal
    i. Recovery is possible if host enters command rm databases/*. There is nothing the user can do to solve the issue, and must contact the owner/hoster.

4. **Accessing the “settings” page while not logged in does not redirect you to the “index” page, but gives a 500 error.**
    a. Input/action that causes failure:
        i. Pressing the back button to return to user’s setting page directly after signing out
    b. Location of fault
        i. Settings.html and Controller.py’s settings/userid path
    c. Possible action for removal
        i. Adding redirect code to index if there is no userID, similar to other pages

5. **Pictures for artists warp and stretch.**
    a. Input/action that causes failure:
        i. Zooming in or out too far and attempting to view the site on a very narrow screen.
    b. Location of fault
        i. artists.html
    c. Possible action for removal
        i. Removing the height parameter for images and constraining size a different way (removing it with no replacement resulted in very small images

6. **Pictures for playlists become wider than the enclosing tile.**
    a. Input/action that causes failure:
        i. Zooming in/out and attempting to view the site on a very narrow screen.
    b. Location of fault
        i. playlists.html
    c. Possible action for removal
        i. Adding a parameter that adjusts image size when changing size of display.

7. **Friend tiles in Group Listen Session cover share your link textbox.**
    a. Input/action that causes failure:
        i. Zooming in/out and attempting to view the site on a very narrow screen, or having too many friends in a group listen.
    b. Location of fault
        i. groupSession.html
    c. Possible action for removal

i. Changing CSS style conditions to assure that “share your link” div renders underneath the friends listening grid

8. **After hitting the browser back button to rejoin the group session page after leaving, the user is not displayed in the session.**
    a. Input/action that causes failure:
        i. This may have something to do with how browsers cache pages. Perhaps the necessary backend call is not sent because of this.
    b. Location of fault
        i. groupSession.html
    c. Possible action for removal
        i. Find a way to keep track of when a user has entered the group session page by hitting their browser’s back button, and then making a call to the database to display the user.

9. **The ordering of people in the groupSession can change on refresh.**
    a. Input/action that causes failure:
        i. A user hitting refresh deletes them from the database because it is seen as leaving the window. Then when the refresh completes they are back in the session but in a different order.
    b. Location of fault
        i. groupSession.html, groupSession.js line 365
    c. Possible action for removal
        i. Do not remove the user’s from the group session on refresh.
        ii. Display the names of the user’s in the group session alphabetically.

10. **Pausing when a user's spotify is already paused will return a 500 error to the server.**
    a. Input/action that causes failure:
        i. Spotify returns an error when an API call is made to pause a track when the instance of Spotify is already paused.
    b. Location of fault
        i. groupSession.html, groupSession.js playOrPause(), controllers.py playOrPauseTrack()
    c. Possible action for removal
        i. Check if the user’s Spotify is already paused, however, this will result in making one more expensive call to Spotify.

11. **Hitting the pause and play buttons repeatedly may make the current song disappear until another 5 seconds pass (for the host) or on refresh/synchronization (for the visitor)**
    a. Input/action that causes failure:

i. Each press of the button makes a call to the Spotify API. If multiple calls are sent out in a short period of time, an error is returned from Spotify.

b. Location of fault
   i. groupSession.html, groupSession.js playOrPause(), controllers.py playOrPauseTrack()

c. Possible action for removal
   i. Keep track of how many times the play and pause buttons have been pressed in a short amount of time. If they have been pressed repeatedly, temporarily lock them.

12. It is possible that a person may enter a session and see a person is already there, they will be removed the next time a user asks for who is in the session.

a. Input/action that causes failure:
   i. Checking when a user was active in order to remove them only occurs when there is one person in the session. If there is only one person remaining in the session, and they decide to leave by closing the tab or browser, then they will still be marked as in the session in the database. Once a different user joins that same session, they will be removed because the checking of active times will commence again.

b. Location of fault
   i. groupSession.html, groupSession.js getPeopleInSession(), controllers.py getPeopleInSession()

c. Possible action for removal
   i. Periodically check the database for sessions with only one user, then check their last active time. If it has been a long time since they were last active, remove them.

13. It is possible that the removal of a person based on inactivity triggers an error due to a person leaving at the same time they are being removed or joining as they are being removed.

a. Input/action that causes failure:
   i. Checking when a user was active in order to remove them only occurs when there is one person in the session. If there is only one person remaining in the session, and they decide to leave by closing the tab or browser, then they will still be marked as in the session in the database. Once a different user joins that same session, they will be removed because the checking of active times will commence again.

b. Location of fault
   i. groupSession.html, controllers.py lines 578-980

c. Possible action for removal
   i. Place a lock during any modifications to the specific groupSessionPeople database entry.

14. **Users can join multiple group sessions at the same time.**
    a. Input/action that causes failure:
        i. Multiple group sessions can be opened in different tabs.
    b. Location of fault
        i. groupSession.html, controllers.py lines 578-980
    c. Possible action for removal
        i. Keep track of how many sessions a user is in and then remove them from one session or block them from entering a new session.

15. **Closing a spotify window will still allow users to pause and play**
    a. Input/action that causes failure:
        i. After a user closes their Spotify, the functions have no way of knowing the device ID is expired.
    b. Location of fault
        i. groupSession.html, groupSession.js getPlayingTrack() synchronizeVisitor()
    c. Possible action for removal
        i. Constantly check and update the user's device ID, but this may cause performance issues.

16. **Highlighting text and typing in active status does not make save and cancel buttons appear**
    a. Input/action that causes failure:
        i. Highlighting text in the active status box and typing new input
    b. Location of fault
        i. user.html
    c. Possible action for removal
        i. A keyup listener could have been added

17. **Active status has no limit, gets cut off but stores more text than is displayed**
    a. Input/action that causes failure:
        i. Typing a long active status
    b. Location of fault
        i. user.html
    c. Possible action for removal
        i. A character limit could have been defined

18. **The last person on the Add Friend page is cut off on smaller screens**
    a. Input/action that causes failure:
        i. Scrolling to the bottom of the list on the Add Friend page
    b. Location of fault

    i. addfriend.html
c. Possible action for removal
    i. The size of the panel could have been adjusted, or empty space could have been added to the bottom to allow more room to scroll

19. Profile pictures on the friends list stretch slightly on small screens

a. Input/action that causes failure:
    i. Using the site on a small screen
b. Location of fault
    i. user.html
c. Possible action for removal
    i. The images could have been set to scale rather than stick to a specific height/width

20. Leaving the banner editor without saving only gives a warning if the user clicks the “Exit” button

a. Input/action that causes failure:
    i. Exiting the banner editor (without saving) by doing something other than clicking the “Exit” button.
b. Location of fault
    i. search.html, search.js
c. Possible action for removal
    i. search.js should check if the user leaves the page, rather than whether the user specifically clicks the “Exit” button

21. The state of the cursor in the banner editor does not always reflect the actions that can be taken

a. Input/action that causes failure:
    i. The cursor almost always indicates that an element can be dropped when a user drags it even if it cannot be dropped on our site, e.g. highlighted text and album results from the search
b. Location of fault
    i. search.html, search.js
c. Possible action for removal
    i. Extensive checking for the type of element being dragged would be needed, and this could control how the cursor displays.

22. Overflow is supposed to limit text to two lines, but sometimes allows three

a. Input/action that causes failure:
    i. Our code determines when to use the overflow css class based on a character limit. There are some combinations of characters that fall under the character limits (for titles and descriptions) but are long enough to require a third line.

b. Location of fault
    i. user.html, playlists.html, groupify_style.css
c. Possible action for removal
    i. Determine whether the text overflows using the height of the section rather than number of characters

23. Tracks and artists with long names cause the image to disappear on some screens

a. Input/action that causes failure:
    i. Viewing the user page where a track or artist has a long name on a small screen
b. Location of fault
    i. user.html
c. Possible action for removal
    i. Limiting the width of the title column

24. Banner Editor sometimes changes the highlighted album after reordering

a. Input/action that causes failure:
    i. Selecting album at index 11 (last on the banner), dragging to index 10, and dragging an album at index <= 9 to index 10
b. Location of fault
    i. search.js
c. Possible action for removal
    i. Debugging the drag and drop functions, adjusting the conditions for when to change the selected album's index

25. Layout and drag and drop do not work correctly on mobile devices

a. Input/action that causes failure:
    i. Visiting the site and dragging album covers on a smartphone. Note: drag and drop does work on iPad, but are confirmed to not work on iPhone
b. Location of fault
    i. All html files, search.js
c. Possible action for removal
    i. Creating an alternate layout and drag and drop functions for mobile

NEEDED ON DOC

- List of functions not working correctly (can lose up to 20 points)
    - List all problems with your software that your team has identified.
        - the input/action that causes failure;
        - location of fault (if known);
        - possible action for removal of fault.

You will not lose points for items on the list; you will lose points for failing to put items on the list if your TA or the prof finds a problem during the acceptance test.

Plus: List of suggested user stories/acceptance criteria for actual acceptance test/project review