

ASSIGNMENT 3: GROOVESHARK CHAT INTERFACE

By

TIMMY CHANDY
HOHYUN JEON
ABBY MAFOOD
TANVIR TALUKDER

DOCUMENTATION FOR ASSIGNMENT 3
CIS4930/CIS6930 INTERACTION DESIGN
GROUP 9

UNIVERSITY OF FLORIDA

2014

TABLE OF CONTENTS

	<u>page</u>
CONTEXT.....	3
RESEARCH.....	4
Focus Groups	4
Personas	6
Ordinary users	6
Taste-making users.....	6
Artist Users.....	6
User Wants	7
User Needs	8
DESIGN PROCESS.....	8
User Scenarios.....	8
Storyboards	9
Chad Sends an Emoticon.....	9
Hannah Broadcasts a Playlist	9
Hannah Blocks a User	9
Rob Announces a Gig	10
Rob Checks Chat History and Shares His New Song	10
Key Features	10
Iterative Design Process	10
DESIGN SOLUTION	11
Task Flow Diagram.....	11
Wireframes.....	11
Announcement	11
Broadcasting.....	11
Change Status	12
Chat Settings	12
Chat Window Options	12
Group Chat and Playlist Sharing	13
Home Page	14
Removal Options.....	14
Share Song.....	15
Low Fidelity Prototype	15
Design	15
Feedback.....	16
High Fidelity Prototype.....	16
Design	16
Feedback.....	17

CONTEXT

Grooveshark is a music sharing website that wants to expand into the social media world by expanding their chat feature across their website. In order to implement a chat user interface that will be easy to understand and intuitive to use, we will be researching what users would like to see in the chat UI, break down false affordances in the original UI, brainstorm concepts for each part of the UI, create a low fidelity prototype, and then create a high fidelity prototype. During each step of the process we will also be receiving feedback from test users and analyze the good and the bad of each design in order to improve from the previous design.

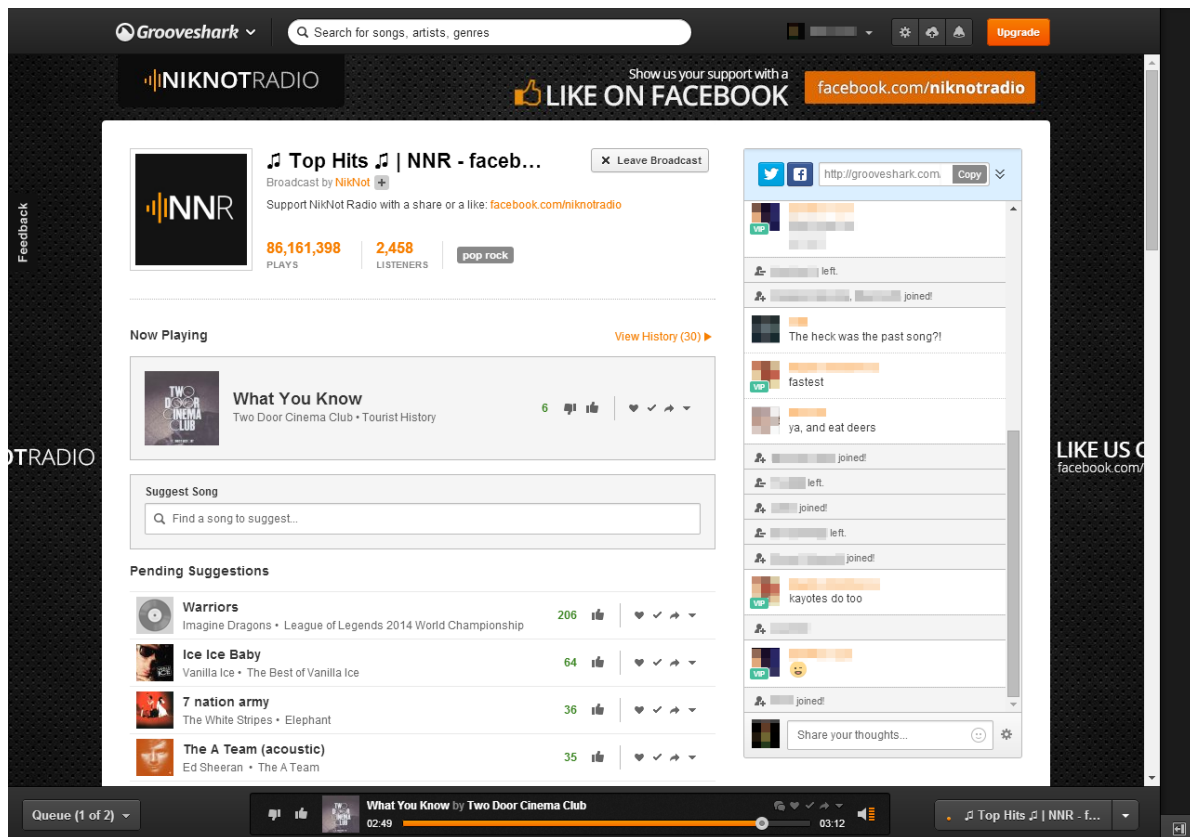


Figure 1. A screenshot of Grooveshark's current chat interface.

As you may see in Figure 1, the current chat UI design has several faults. Grooveshark's current interface is limited to the broadcast page. When the user leaves the page, they will also leave the chat. That means that the user can not be browsing for music and chatting at the same time. Each post made by a user in the chat does not have a timestamp. The chat log is constantly spammed by notices that a user joined or left. The chat log is limited to when the user logged in. Emoticons are limited to paid users. The settings menu only lets you toggle join/leave activity, song plays, and suggestions. It gives no options to

mute/report users. We created a list of topics that focuses on the different features that will tackle the issues.

Table 1. List of issues and topics for research.

Fault	Topic
Chat is in the broadcast page only even though songs in broadcast will persist when switching pages internally.	Persistent chat.
Chat log is limited	Full chat history
Emoticons are limited	Emoticon and non-text media in chat
No way invite friends internally	Friend invite to chat
No way to mute/flag users	Muting/Flagging users
Missing timestamp	Timestamp
Other user activity spamming chat	Move activity alert into a different format
Settings menu only controls what you see in terms of user activity.	What users would like to control in the settings menu

RESEARCH

Focus Groups

Using the topics listed in Table 1, an organized list of topics in Table 2 was created to cover all parts of the chat UI. We originally had three focus groups of users: potential, current, and expert users but replaced the list to ordinary, taste-making, and artist users to widen the scope of our research. Ordinary users use Groovespark to listen and chat. Taste-making users use Groovespark to create and share playlists. Artist users use Groovespark to upload and share their music. We presented the following topics to the focus groups.

Table 2. List of research topics

Topic	Description
1 User Related	
1.1 Avatar/Username	Does the user want to see the avatar and username in the chat?
2 Chat Related	
2.1 Text/html editor	ASCII text or HTML scripting in text editor?
2.2 emoticons	Make emoticons standard instead of a paid service
2.3 embedding	Embed videos, music, images, and playlists into chat.
2.4 chat history	Currently chat history is limited to when the user logged in. Does the user want to see the full chat history or have it limited?

Table 2. Continued

Topic	Description
2.5 show when others are typing	A toast or alert notice that allows the user to know that someone in the chat is typing.
2.6 Who saw message	A chat features that allows the user to see who saw which message.
3 Screen Space	
3.1 View multiple chat sessions at the same time	Facebook's chat UI allows the user to see two different chat sessions simultaneously. Does the user want to use that or focus on one chat session at a time?
3.2 full window chat or chat in corner	Should the chat UI take all the screenspace or should be limited to a small portion the Grooveshark UI?
3.3 separate window or embedded chat	Should the chat be independent from the rest of the UI or should it be integrated with the main UI?
4 Social Related	
4.1 mute/report users	How should abusive users be handled by the user.
4.2 hot swapping chat sessions	Can the user stay active in two or more chat sessions and switch between them?
4.3 Session merging	Should the user owner two chat sessions be able to fuse them together?
4.4 status	Should users have a user status? What statuses should be used?
4.5 offline chat to PM	Should messages to offline users be saved as a PM?

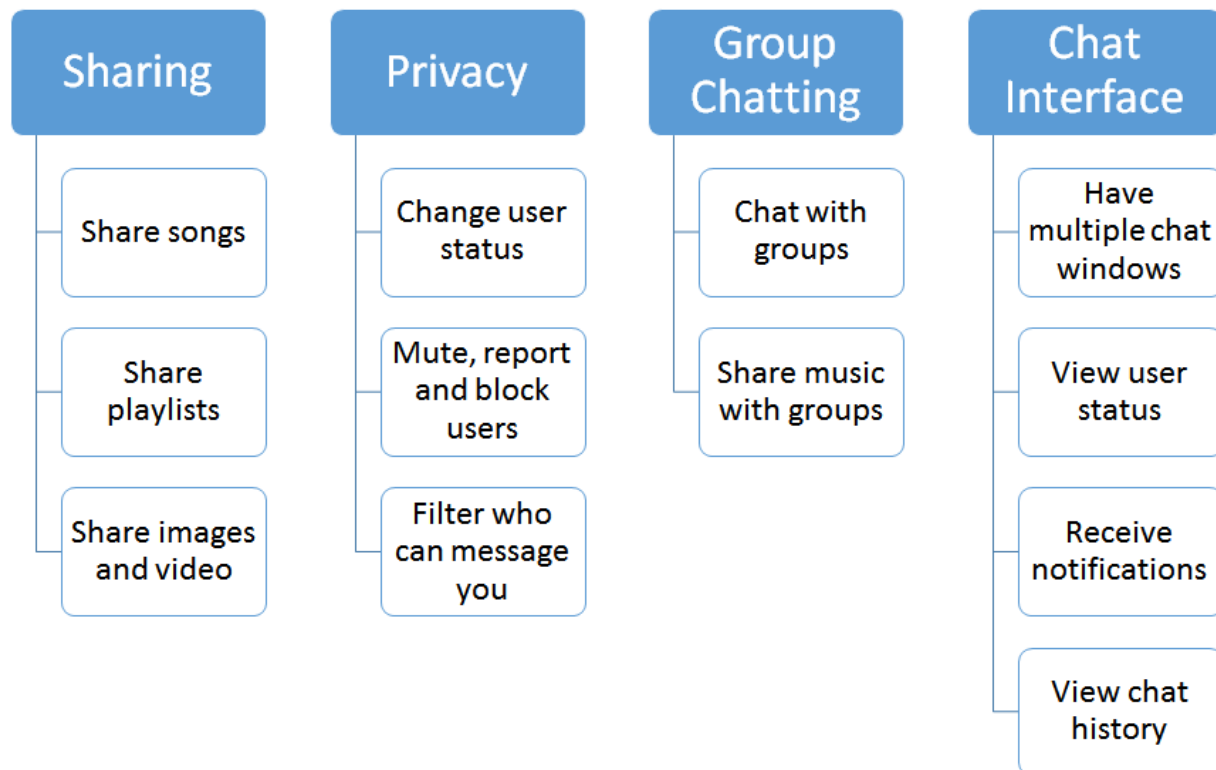


Figure 2. User need affinity diagram. The user research yielded the following user needs which were organized into the affinity diagram above.

Personas

Our personas were created based on the different characteristics of user needs that appeared in our focus group studies. Because the user needs we found were so interconnected with each persona, we will cover the results of the user needs under each persona.

Ordinary users



Name: Chad

“Chatting through Grooveshark should be convenient”

Demographics:

Age: Early 20s

Occupation: Undergrad Student

Chad uses Grooveshark to listen to music for free. Chad is familiar with using Facebook Messaging and Google Plus to connect with friends.

Taste-making users



Name: Hannah

“I want to know what people think of my music tastes”

Demographics:

Age: Mid 20s

Occupation: Music Critic/Journalist

Hannah is a tastemaker on Grooveshark -- A person who creates playlists and broadcasts them so others can listen. She currently uses Facebook Messaging and GroupMe to communicate with friends who follow her broadcasts, but would prefer to have a better chat interface on Grooveshark to interact with them on the website.

Artist Users



Name: Rob

“I want everyone to hear my music”

Demographics:

Age: Early 30s

Occupation: Electronic Dance Music (EDM) Artist

Rob is an EDM artist who uses Grooveshark to share his music. He uses Twitter to announce new releases and frequents Grooveshark/Twitter for feedback from his fans. Because feedback is important to him, he wants to be able to interact with his fans easily.

User Wants

Using these personas we organized what the focus groups said they wanted to do in the Grooveshark chat. This research data was organized with each persona so we can describe what each persona wants to do in the Grooveshark chat. We went into detail from these user wants to specific user needs for each part of the user interface. This information can be found in Table 3.

- Chad wants to send messages to a group of people.
- Chad wants to discuss songs with other fans.
- Hannah wants to be notified when one of the artists she is following sends a message.
- Hannah wants control chat content in her broadcast
- Rob wants to announce a new song release.
- Rob wants to chat with his fans.

Table 3. User needs by persona based on user need topics covered in Table 2.

Topic	Chad	Hannah	Rob
1 User			
1.1 Avatar/Username	Yes	Yes	Yes
2 Chat			
2.1 Text Editor	standard text editor	standard text editor	HTML text editor
2.2 Emoticons	Yes	Yes	Yes
2.3 Embedding	Playlist	Music/Playlist	Video/Music/Image
2.4 Chat History	Full History	Full History	Full History
2.5 Show when others are typing	Yes	No	No
2.6 Show if someone saw their message	For artists only	Yes	No
3 Screen Space			
3.1 View multiple chat sessions at the same time	One at a time	Multiple chat sessions	Multiple chat sessions
3.2 Full window chat or chat in corner	Both, depending on the situation	Chat in corner	Chat in corner
3.3 Separate window or embedded chat	Both, depending on the situation	Embedded chat	Separate window
4 Social			
4.1 Mute/report users	Yes	Yes	Yes
4.2 Hot swapping chat sessions	Yes	Yes	Yes
4.3 Chat session merging	No	No	No
4.4 Status	Online/Offline/Busy	Online/Offline	Online/Offline
4.5 Offline chat to PM	No	Yes	Yes

User Needs

Based on the response on each topic, the list of user needs was re-built into the following list. From this list, we asked our focus groups again how and what they would look for to fulfill each user need. To be concise, we compiled the expectations for each focus group into one overall expectation per user need in Table 4.

Table 4. User needs and their expectations.

User Need	Expectation
See avatar and username in post	Look for the avatar and username inline with the user post.
Emoticons and text-based communication	Click in the text box to begin typing. Look for a smiley face to open a emoticon menu.
Embedded video,image,music, and playlist sharing	Drag and drop media or insert link that will automatically embed media into post.
View full chat history	Scroll up to view history
Other user chat activity	A toast that will show if one or more people are typing. An animated icon will work as well.
Integrated chat interface with the main UI that takes up an appropriate amount of space.	Expect an icon or a menu to begin chatting. Would also expect some way to hide and unhide interface. Logically in an area where UI for other features does not already take place.
View one chat session at a time	Expect one window to do chatting.
User muting/reporting	Look for icons below avatar or username to report or mute user.
Chat hot swapping	Look for a side pane, tabbed menu, or a list that shows how many chat sessions are open. Choose one chat session to switch.
User status (Online/Offline/Busy)	Look for an icon next to username to see if the user's status. Change self-status by looking for a settings menu.

DESIGN PROCESS

User Scenarios

The following list describes what each persona would do in a particular scenario.

- Chad discovers a new song and wants to discuss them with his friends on Grooveshark.
- Chad is using Grooveshark to listen to music when his friend tries to chat but he does not want to chat.
- Hannah created a new playlist and wants to know what people think of it.
- Hannah sees a user that is talking badly about her playlist and she wants to ban him.

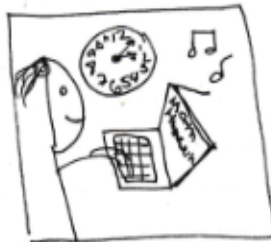
- Rob has a new gig and wants to announce the news on Grooveshark to see what his fans think about it.
- Rob uploaded a song and wants to know what others think about it.

Storyboards

Chad Sends an Emoticon



Chad is studying and listening to music on Grooveshark.



Chad doesn't have much time and receives a message from a friend

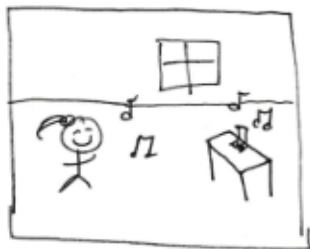


Chad doesn't have much time to chat right now so he changes his status to busy



Chad's friend understands that Chad can't chat because his status is busy.

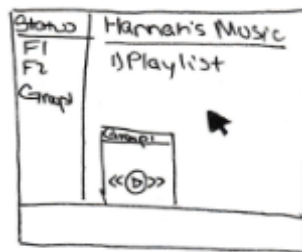
Hannah Broadcasts a Playlist



Hannah is jamming out to her own playlist she created on Grooveshark.



Hannah is getting bored listening to the same old stuff.



Hannah shares her playlist with her friends by dragging and dropping it into the group chat.



Hannah broadcasts her playlist to talk to people all over the world about what else she can add to it.

Hannah Blocks a User



Hannah walks to school while listening to music on Grooveshark.



Hannah hates receiving messages from her crazy ex.

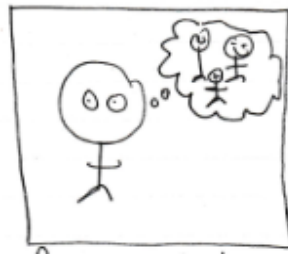


Hannah uses the privacy settings on Grooveshark to block him.

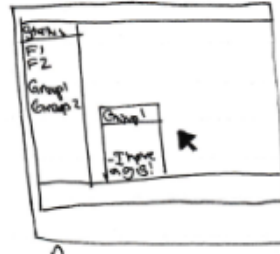
Rob Announces a Gig



Rob receives a call about a new gig.



Rob wants all of his buddies to know

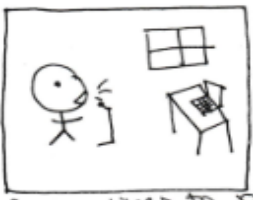


Rob clicks open the group message and tells them the news.

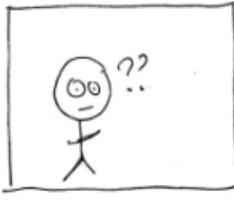


Rob advertises to his fans by announcing the news on his profile.

Rob Checks Chat History and Shares His New Song



Rob just finished recording and uploading his new song.



He can't remember if his buddy asked him to send the playlist directly.



Rob looks through their chat history to see whether his buddy made that request.



Rob clicks and drags his newly created playlist into their chat box.

Key Features

1. Join a chat session and send messages in the chat window -- including using related parts of the chat interface such as emoticons, status, layout of website & chat windows
2. Privacy Settings--Blocking/muting Users and private chat
3. Group Sharing: Drag & Drop Playlists/Music into chat window to suggest songs in the broadcast

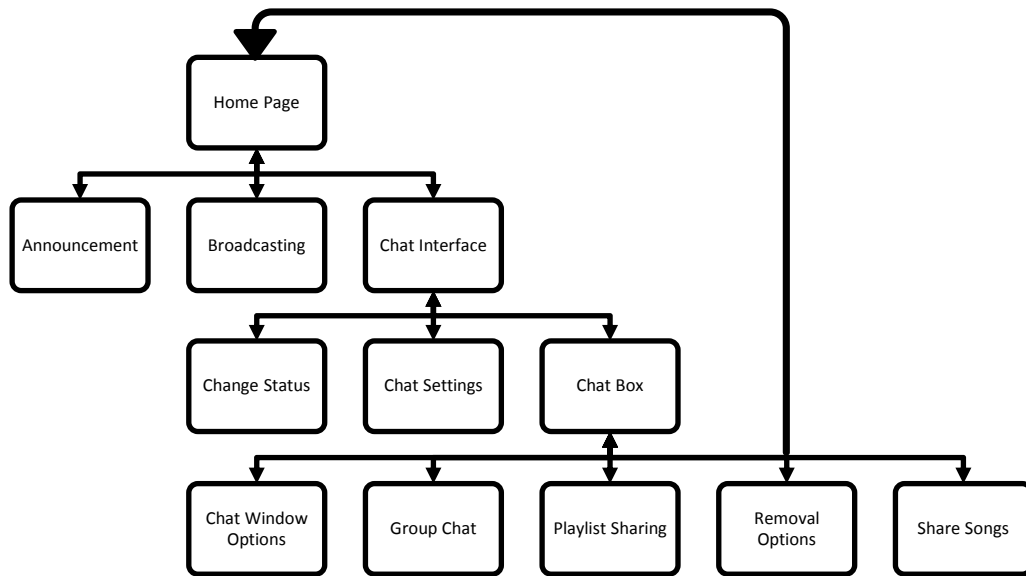
Iterative Design Process

We went through several iterations of the design. For the sake of brevity we will quickly discuss what we went through and show the final design.

Initially, we wanted put the entire chat interface in a left sidebar with each chat session contained there with a list of friends. However our focus groups found the chat too cluttered/constrained and we decided to separate the friend list and the chat box into separate parts. This design is very similar to the Facebook chat interface and we did this because most of our focus groups were used to using a Facebook style chat interface. After the midpoint evaluation, we realized that we missed a feature that already exists in Grooveshark -- the community list. This was equivalent to our current friend list so we moved the left sidebar and integrated it to the already existing community section in the right sidebar. From this process we arrived at the following design.

DESIGN SOLUTION

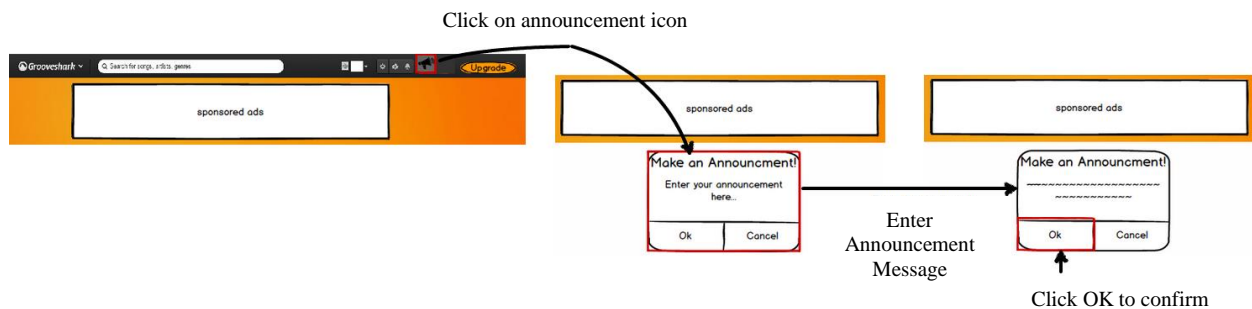
Task Flow Diagram



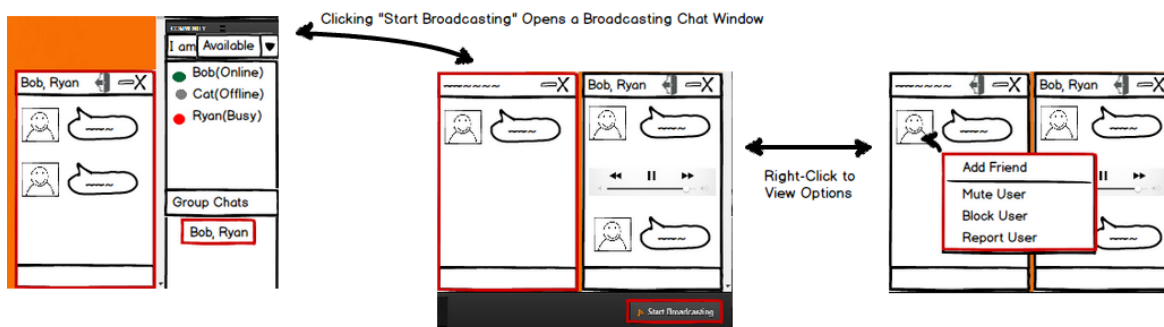
Wireframes

The following wireframes will go into detail how each part of the task flow diagram is designed. For convenience, the wireframes are organized into flowing diagrams for each task and placed in alphabetical order.

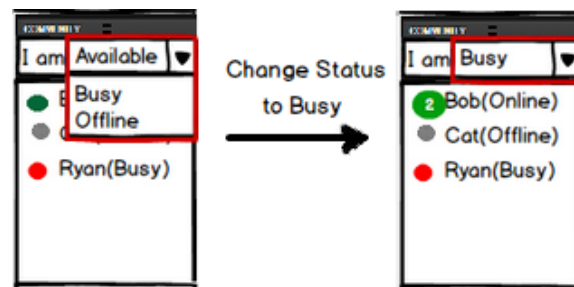
Announcement



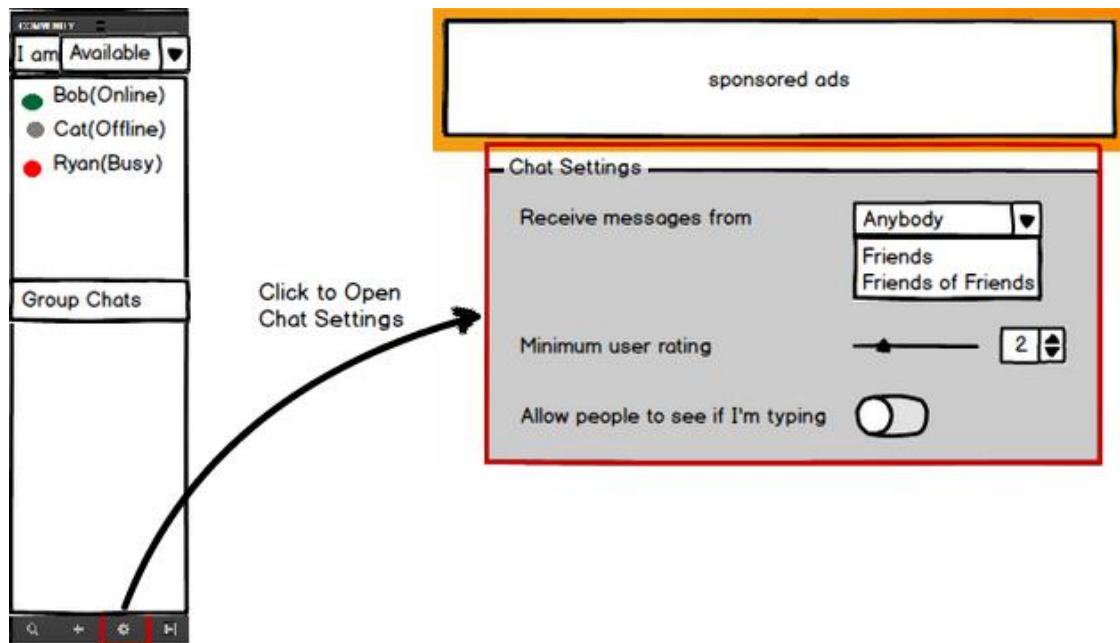
Broadcasting



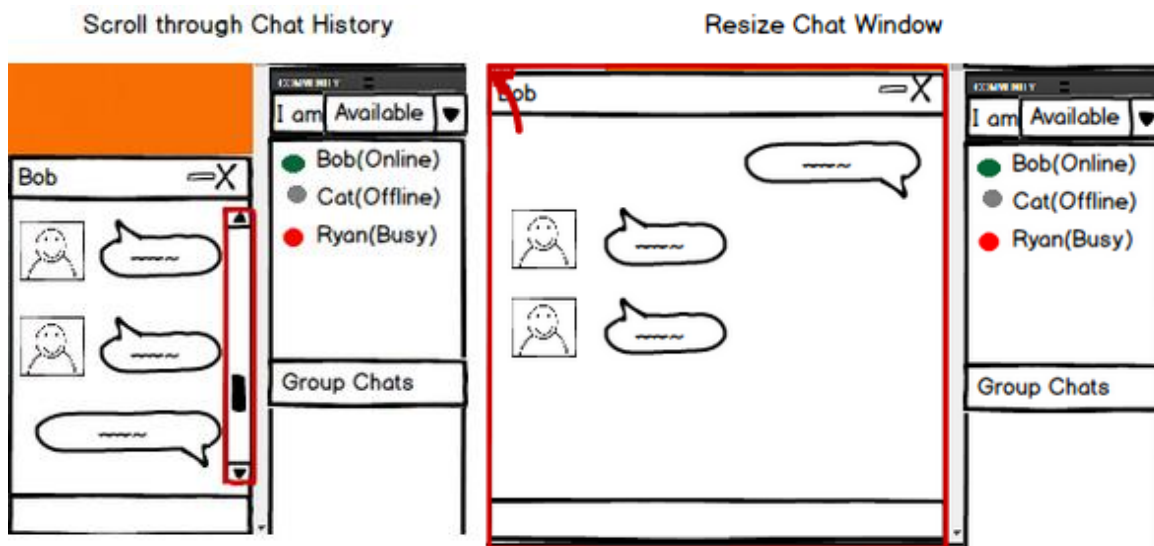
Change Status

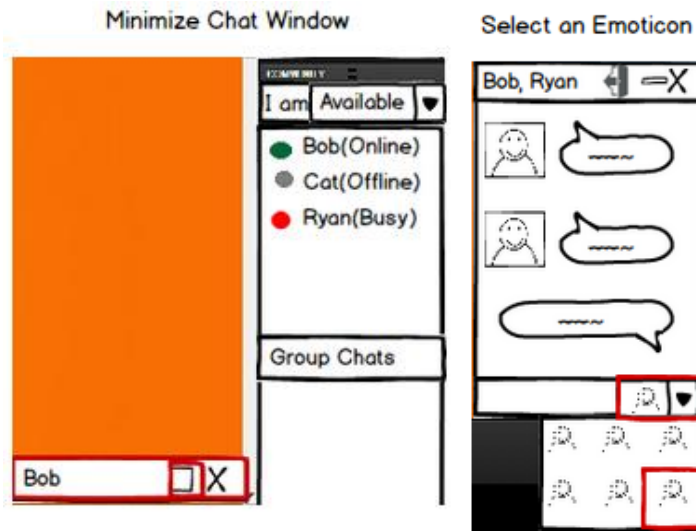


Chat Settings

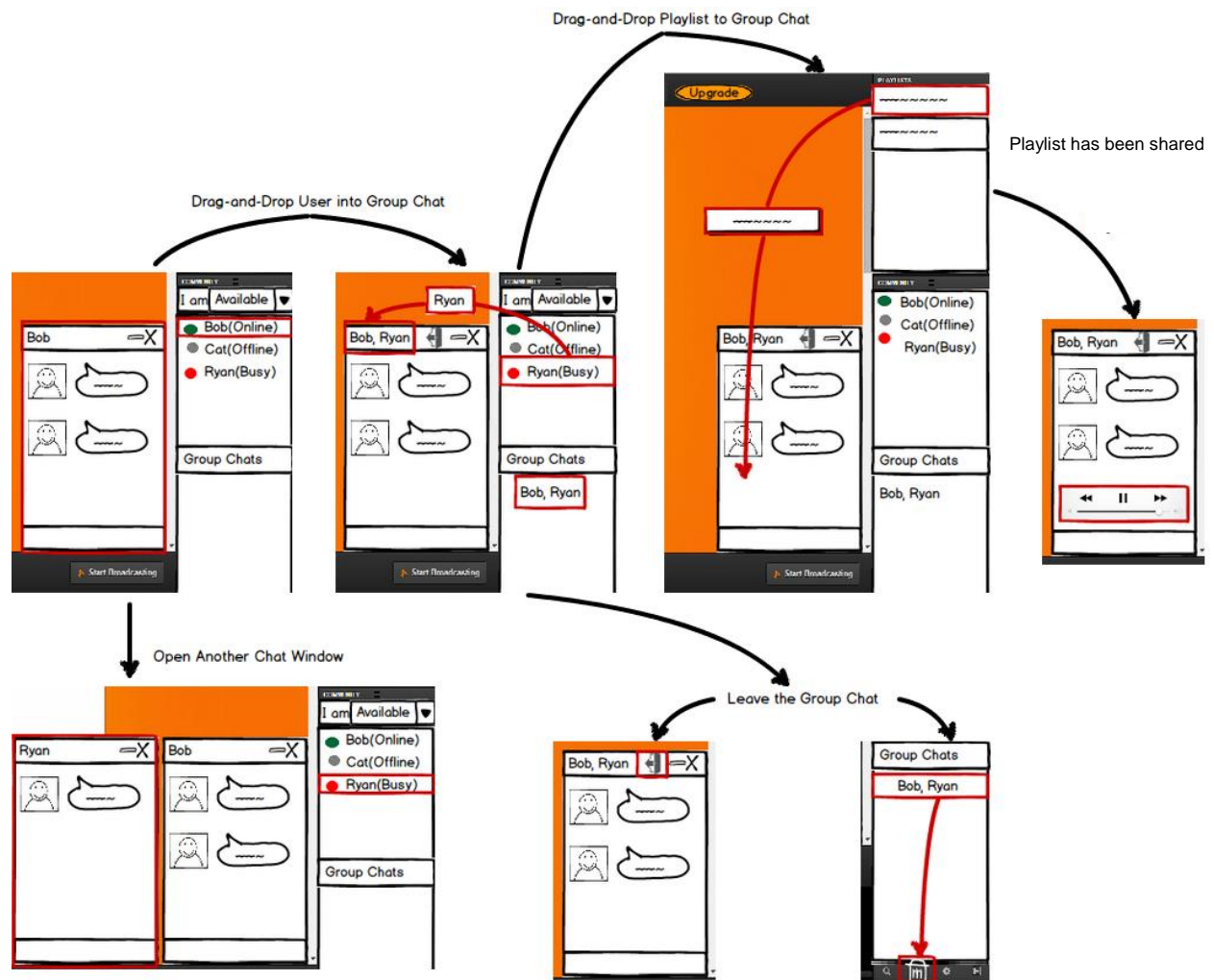


Chat Window Options

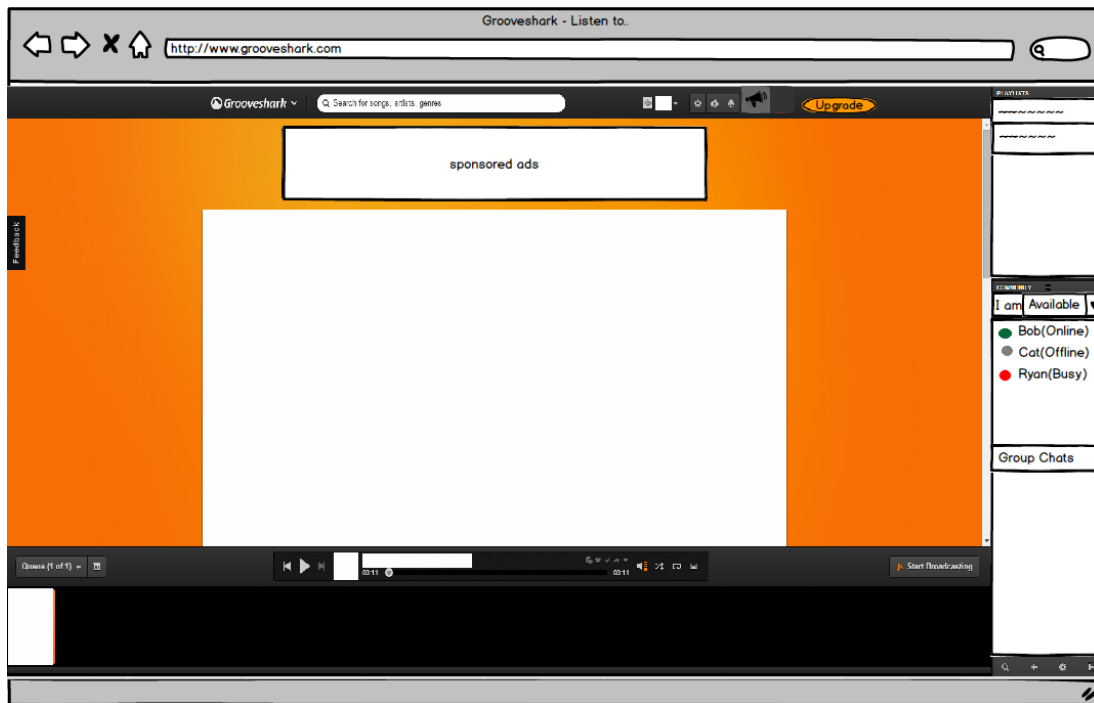




Group Chat and Playlist Sharing



Home Page

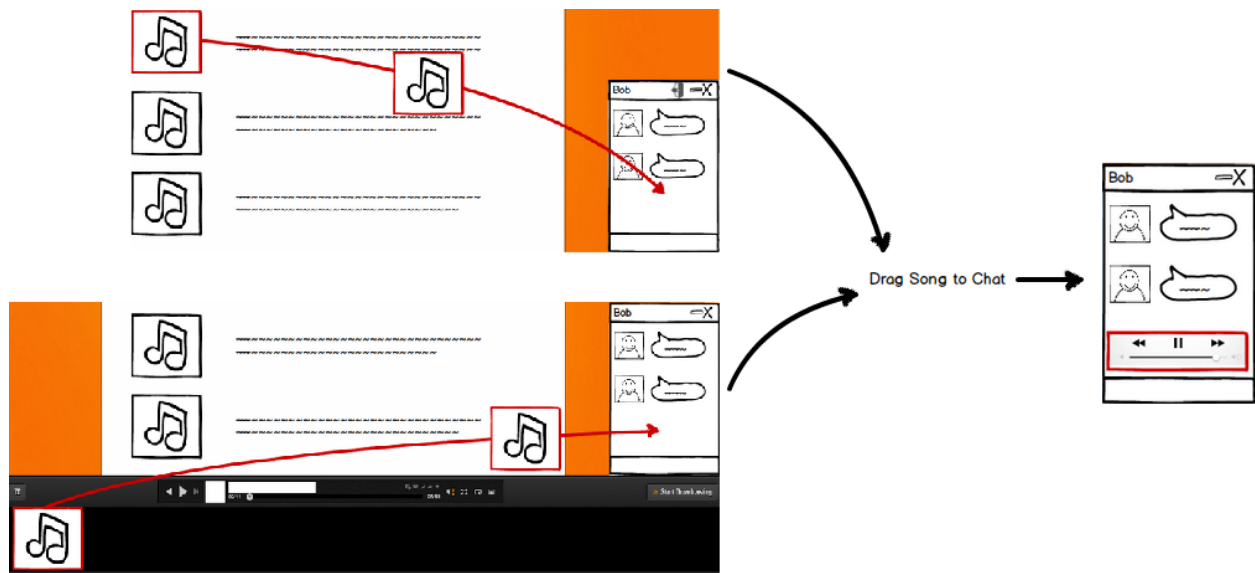


Removal Options



When Dragging, the "+" Icon at the Bottom Changes to a Trash Icon

Share Song



Low Fidelity Prototype

Design

The design for the low fidelity prototype was the same as the wireframes presented in the previous section. The low fidelity prototype was tested by printing out the wireframes and letting our test users attempt to complete the user scenarios by going from wireframe to wireframe.

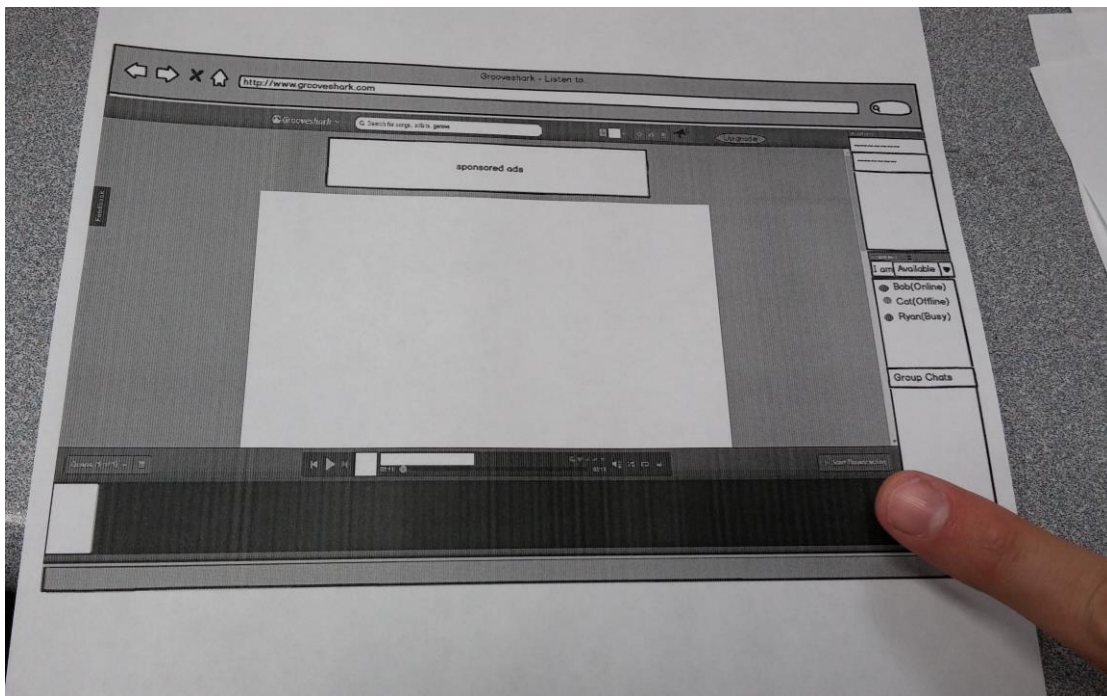


Figure 3. An image of a user testing a low fidelity prototype. Image courtesy of authors.

Feedback

The general feedback from the low fidelity prototype was that drag and drop system for chat creation, adding users to chat, and sharing music was intuitive and easy to do. However, some users were often lost at what they needed to do and blamed the cause on small user interface (icons and text). Responding to this feedback, we tried to cut out lengthy text and increased icon sizes.

High Fidelity Prototype

Design

We used axshare to create our high fidelity prototype. The current version can be found here: <http://igzwoy.axshare.com/>. We made changes to the design based on the feedback from the low fidelity prototype. These changes increased emphasis on important parts of the chat interface such as the notifications and user status menu.

There are several features we could not show in this prototype. We cannot show how to type text and make it show up in the chat box, we cannot show chat history scrolling, and we cannot show how to resize the chat box. This limitation is because Axshare works by creating images then linking parts of it to another image. Because of this fact we can either show an inflexible text field and make the prototype more complicated or show the concept of the feature for the clients. We chose to keep the prototype simple and show the clients the intended feature.

All the main features that we determined to be important were added and a user can go through and accomplish each task the feature is meant to do.

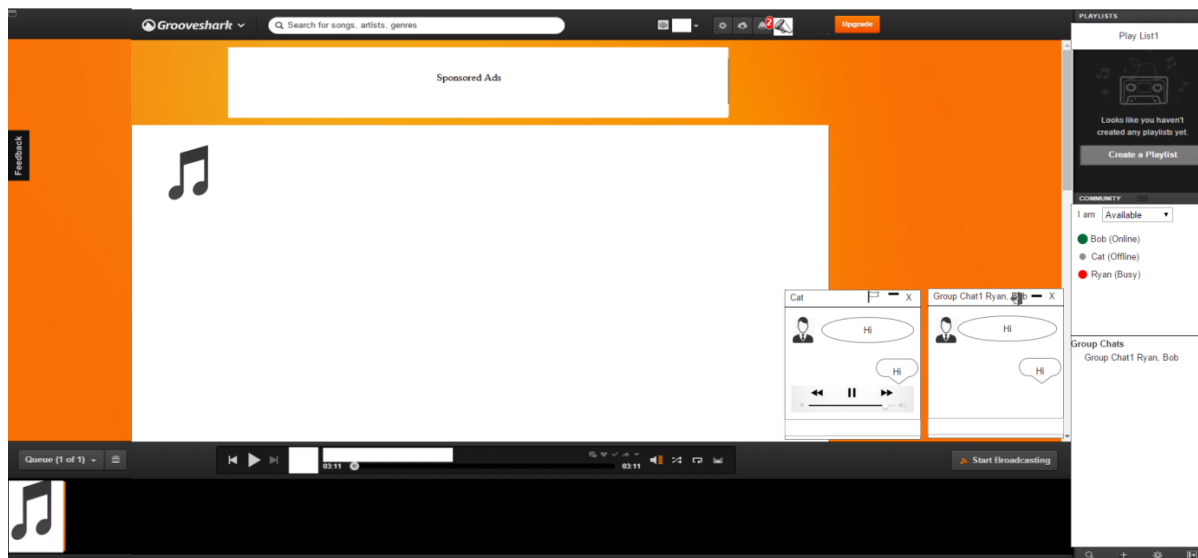


Figure 4. Group chat and music sharing features in the high fidelity prototype.

Feedback

Generally the responses to the current UI were positive. People understood what each icon and label meant and was able to process each task without a problem. There were some technical issues in the first feedback test because some links between tasks were missing. For example users could not close the notifications menu because the link to close it was missing. These issues were fixed in the subsequent feedback tests.