

# Grooveshark—Chat Interface

## Context

Grooveshark is a music sharing website that will implement a new chat feature which allows artists and taste-makers to socialize and share music.

## User Needs

Multiple Users in the Focus Groups Indicated a Desire for the Following:

- Ability to send quick responses via emoticons or send lengthy responses via text
- Ability to go back and see prior conversations
- Ability to know if a user is on/offline or just busy
- Ability to block spam/mute other users
- Ability to easily share created playlists with friends
- Ability to easily advertise to many people
- Easily understandable user interface, something the user is already familiar with such as Facebook.

## Persona: Tastemaker

**Name:** Hannah

"I prefer Facebook, but I'm open to messaging through GrooveShark."

### Demographics:

Age: Early 20s

Occupation: Undergrad Student

Major: Marketing

### Device Usage:

Laptop Usage: 80 hrs/week

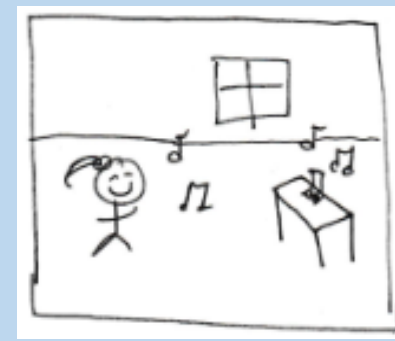
Hours Listened to Music: 60 hrs/week



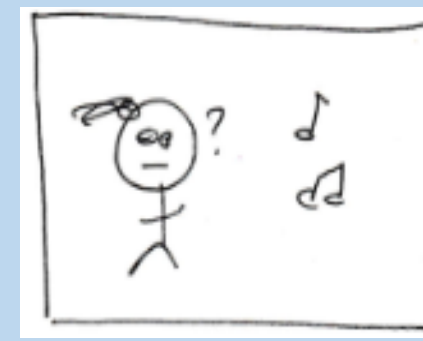
### Description:

Hannah often listens to music through Grooveshark on her laptop while studying. She currently uses Facebook Messaging to communicate with friends. She says it would be really cool to be able to connect with friends through Grooveshark. She also loves going to local concerts.

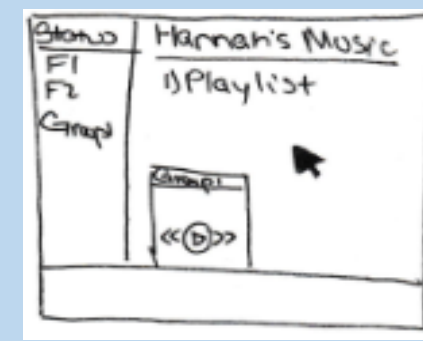
## Storyboard: Group Sharing



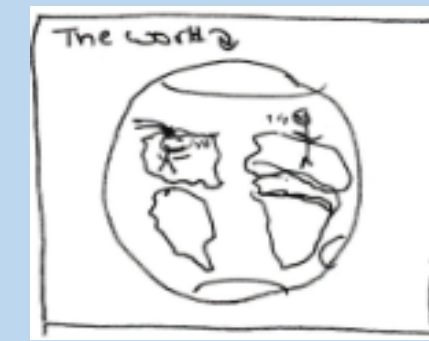
Hannah is jamming out to a playlist she created on Grooveshark!



Hannah is getting bored listening to the same old stuff...

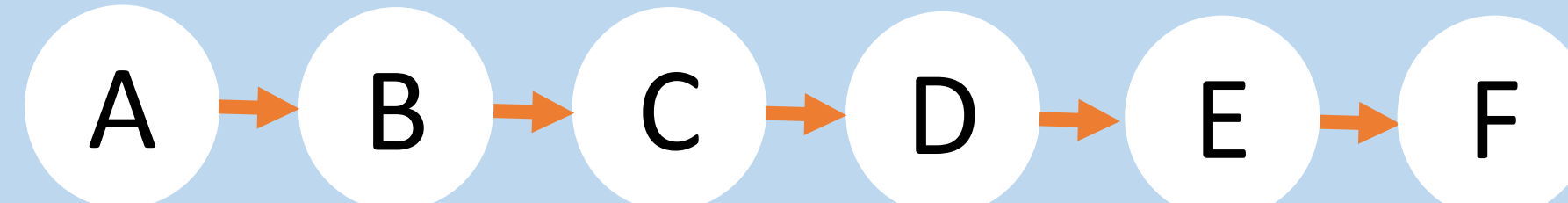


Hannah shares her playlist with her friends through drag & drop and gets their input.

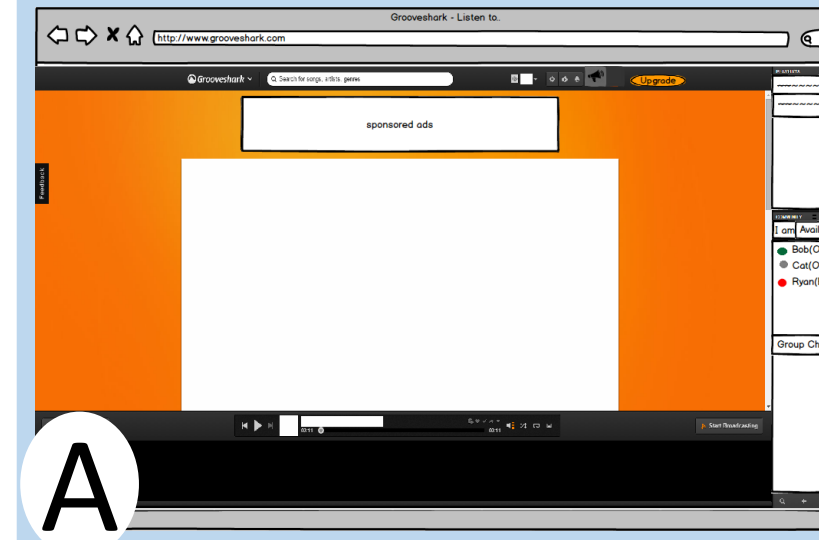


Hannah broadcasts her playlist to talk to people all over the world about what else she can add to it.

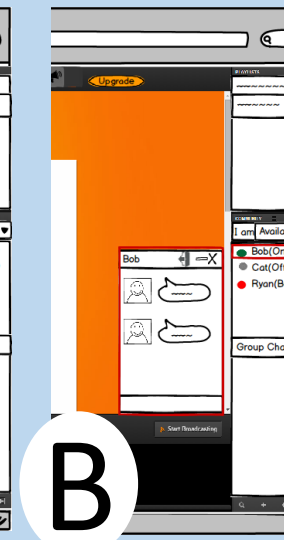
## Task Flow: Group Sharing



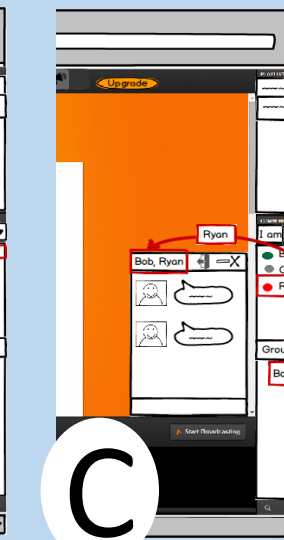
## Wireframes: Group Sharing



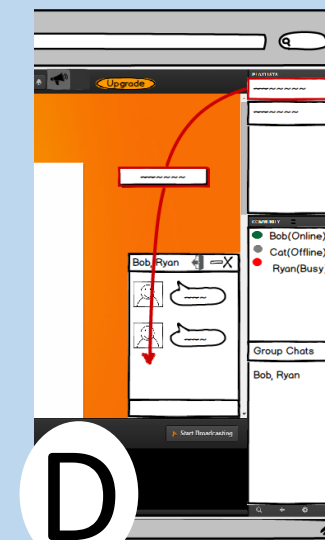
This is the homepage with the chat bar on the right instead of the left.



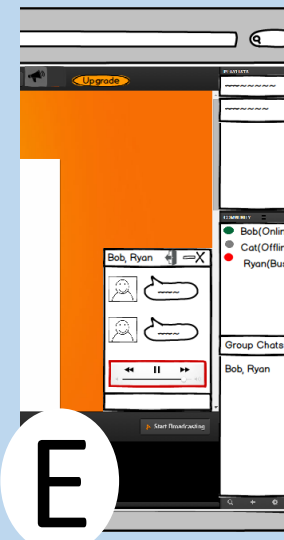
The user clicks on the name of their friend to open a chat window



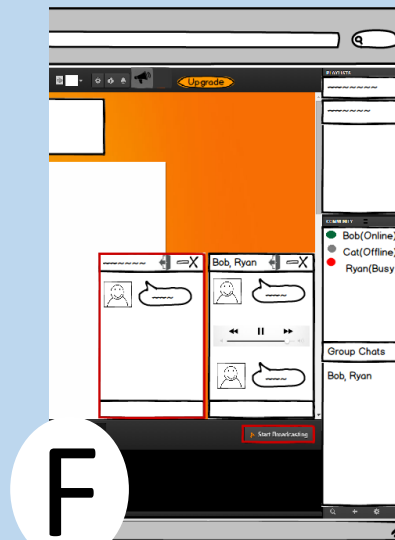
The user clicks and drags the name of another friend to create a group chat



The user would click and drag the playlist in the upper right corner straight into the group chat.



The users within the group chat can now listen to the shared playlist.



The users clicks on the broadcasting button in red to broadcast the playlist in the queue

**Difference from Original Design:** The layout of our design was changed per Grooveshark's feedback. Additionally, we added the avatar/pictures into the Chat Window as this was prominent in our Focus Groups

## Persona: Artist

**Name:** Rob

"I want everyone to hear my music"

### Demographics:

Age: Early 30s

Occupation: EDM Artist

### Device Usage:

Laptop Usage: 80 hrs/week

Hours Listened to Music: 60 hrs/week



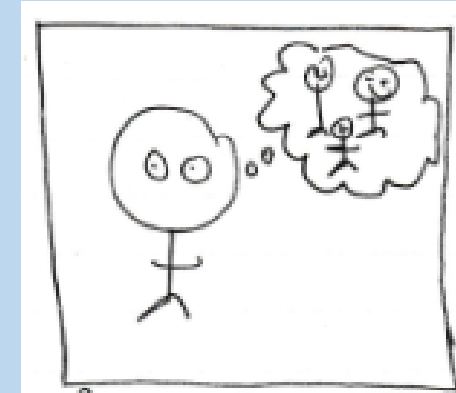
### Description:

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.

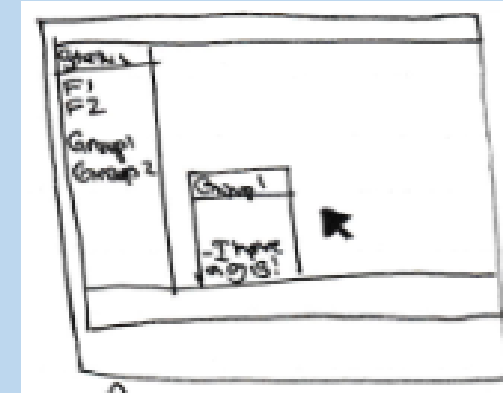
## Storyboard: Group Sharing & Announcements



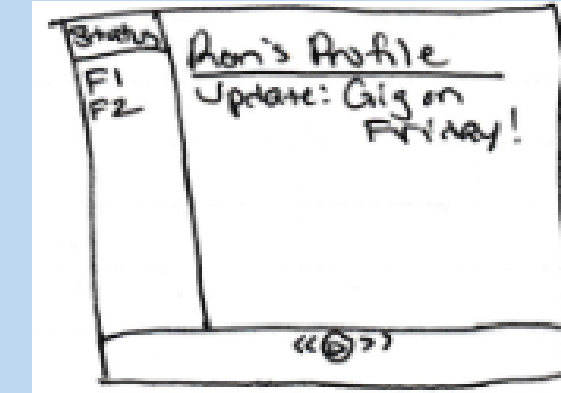
Rob receives a call about a new gig!



Rob wants all his buddies to know!

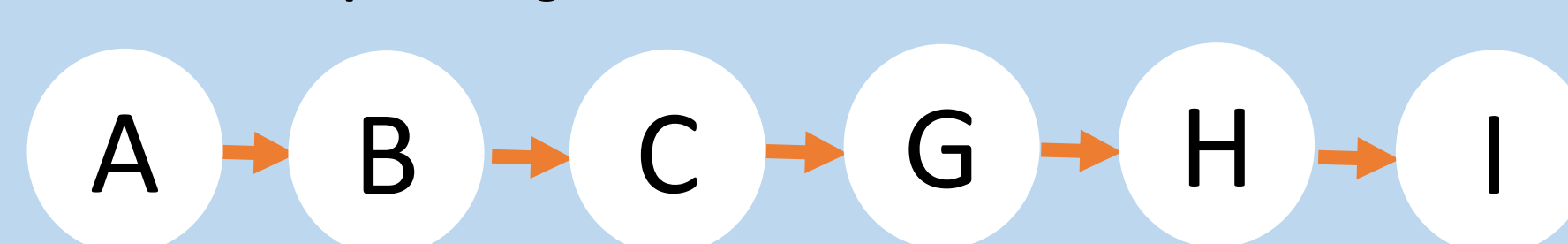


Rob clicks open the group message he has with his buddies and tells them the news!

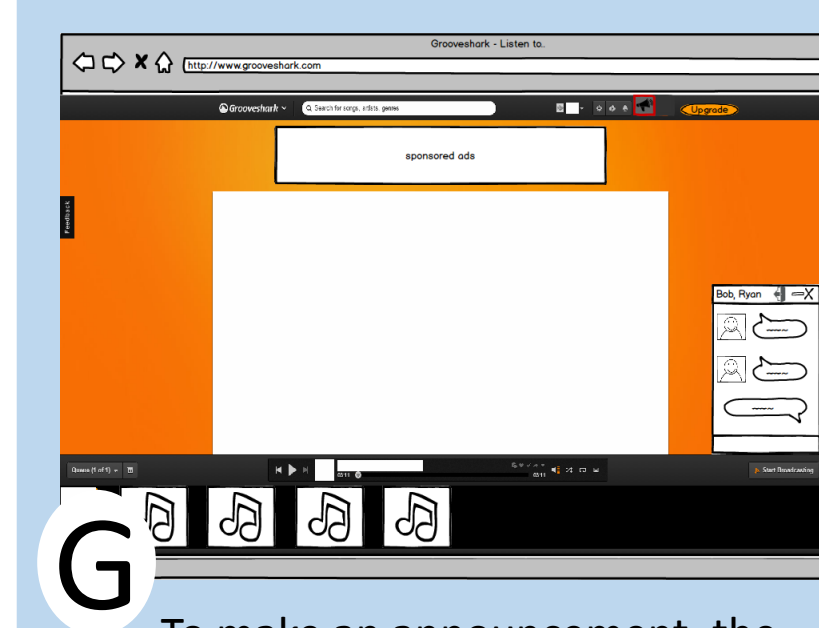


Rob also decides to advertise to fans by announcing it on his profile where fans will see

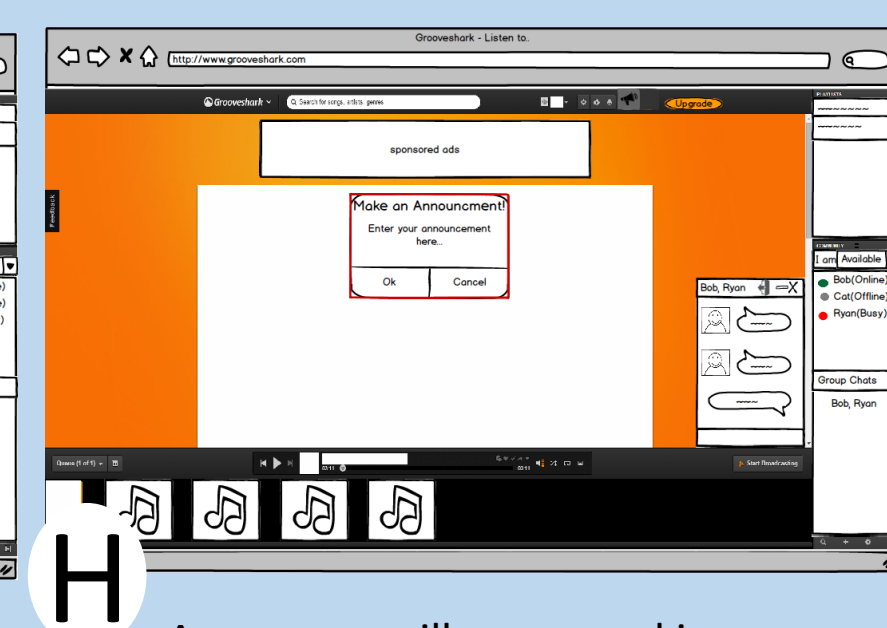
## Task Flow: Group Sharing & Announcements



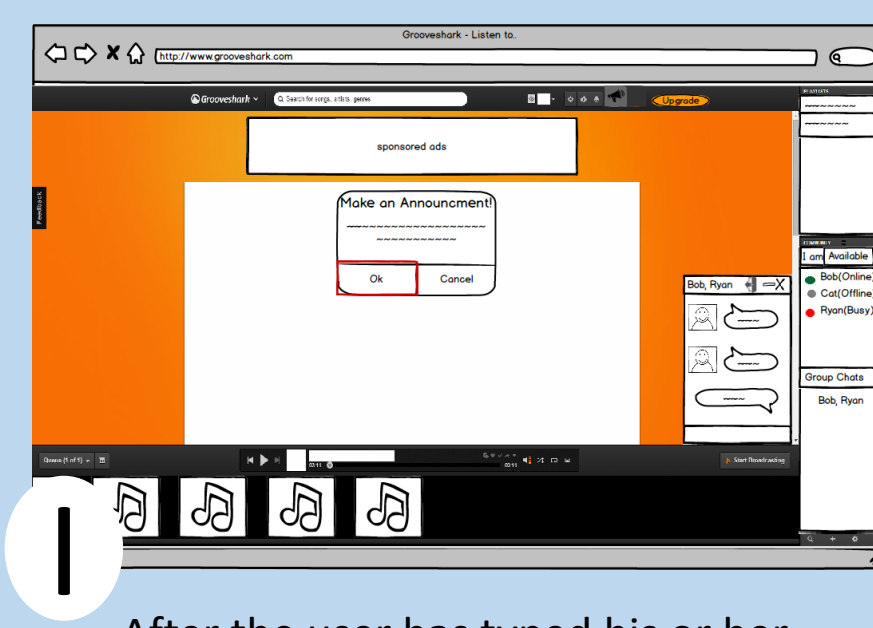
## Wireframes: Announcements



To make an announcement, the user would click on the Announcement button which is circled in red above.



A message will pop up asking the user to type his or her announcement.



After the user has typed his or her announcement, he or she will click "OK." Now all his or her followers will receive a notification for the announcement.

**Difference from Original Design:** An announcements button was added for user convenience when making an announcement

## Persona: Ordinary User

**Name:** Chad

"Chatting through Grooveshark should be convenient."

### Demographics:

Age: Early 20s

Occupation: Undergrad Student

Major: Engineering

### Device Usage:

Laptop Usage: 80 hrs/week

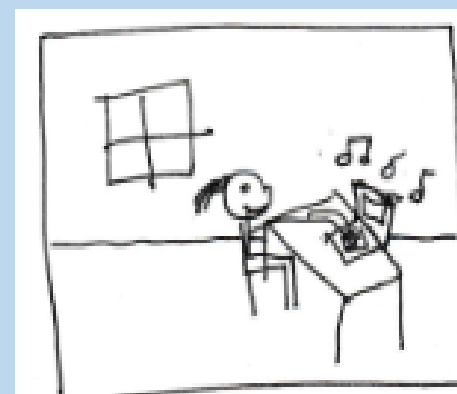
Hours Listened to Music: 25 hrs/week



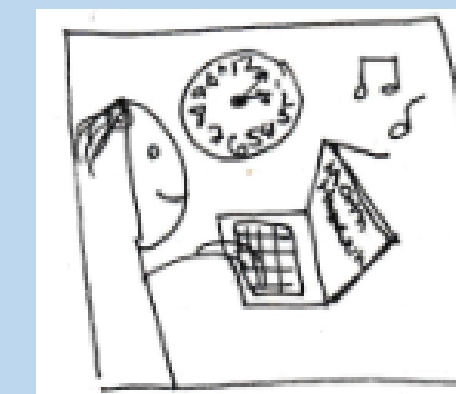
### Description:

Chad uses Pandora to listen to music for free. He usually uses his mobile device and the Pandora App. Chad uses Facebook Messaging and Google Plus to connect with friends

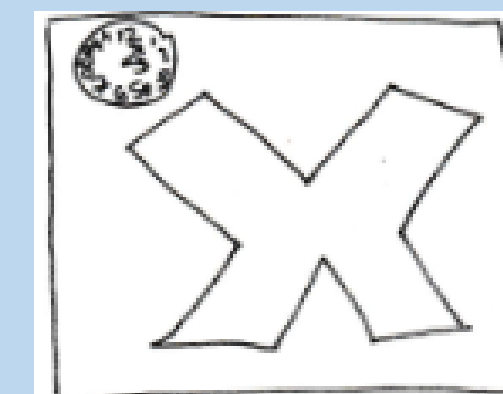
## Storyboard: User Status



Chad is listening to music and editing his resume while looking for a part time job.



Chad looks at the clock and sees he doesn't have much time. He then receives a message from a friend asking him to go hiking.

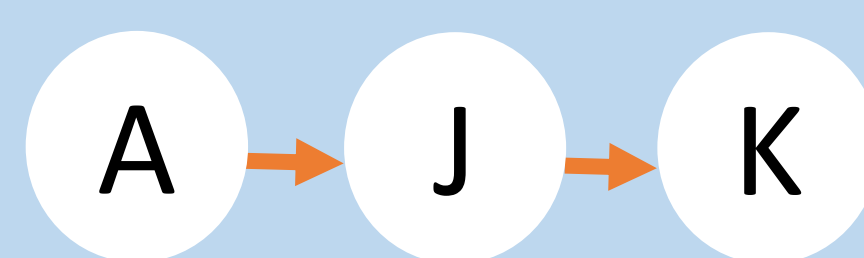


Chad doesn't have much time to chat right now so he changes his status to "busy."

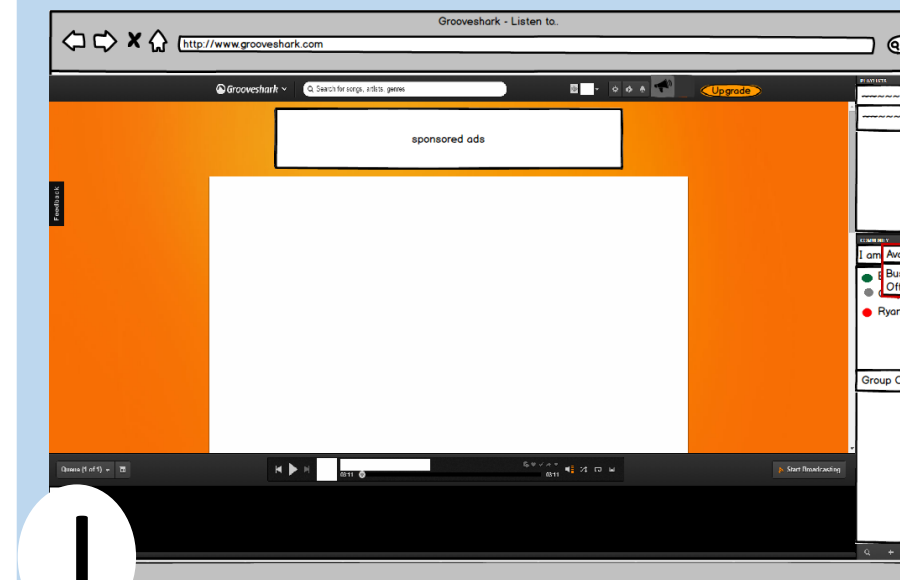


Chad's friend isn't offended that Chad isn't responding because he now sees that Chad is "busy."

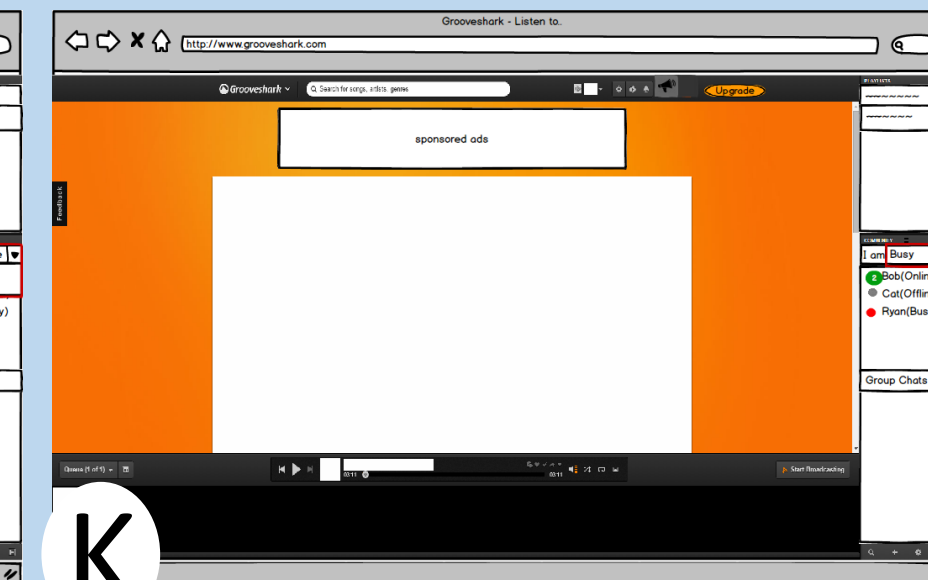
## Task Flow: User Status



## Wireframes: User Status



The user will click on the drop down menu circled in red to change his or her status



After selecting "Busy," the user's status will appear as "Busy" to all friends. If a friend tries to message the busy user, the message will not pop up as a chat window. Instead, a number will appear next to the friend's name with how many messages he or she sent

**Difference from Original Design:** The new design features a "Status" button/display so the user can easily see and change what his or her status is (Available/Busy/Offline) in one location instead of going into the Settings Menu.