Resound.fm

by

Matt Bleifer

Computer Science Department

College of Engineering

California Polytechnic State University

2015

Contents

Abstract

The goal of this senior project was to create a social platform for people to share and discover the music they love. Current music streaming technologies lack an effective tool for helping users find new music. The personal element of music that used to exist in the days that people gathered around record players or shared mixtapes has been lost online and Resound.fm seeks to bring it back. Resound.fm is a Ruby on Rails web application, which uses some of the latest web technologies and design patterns to create a powerful new music discovery experience online. The project was completed over a period of seven months and is currently available for the public to use at www.resound.fm.

1. Introduction

In 1999, a peer-to-peer music sharing application known as Napster hit the web and caused the landscape of the music industry to forever be changed. It marked the end of an era, where consumers bought full length, physical albums in the form of CDs, cassettes, or vinyl records. It began a revolution that brought music out of the physical world, and into the virtual one. Since then, the music industry has only continued to change. The world saw the rise of the iTunes store which offered consumers the ability to download individual songs for only 99 cents. Now in recent years, another massive shift has begun to occur, as many consumers transition to music streaming services.

The concept is simple. Pay a monthly fee, and gain instant access to a catalog of millions of songs. Gone are the days of buying albums, and even individual songs. Gone are the days of needing to store music on a personal device. Users now have access to music that could only be dreamt of decades ago.

As of 2013, music streaming became a billion dollar industry. In the past year alone, music streaming has grown 34.7% and continues to grow every single year. Major companies have begun to realize the market opportunity and are making very large acquisitions. Apple acquired Beats Music for three billion dollars. Google acquired Songza for thirty-nine million dollars, and Rhapsody acquired Soundtracking and Ex.fm for undisclosed amounts. This industry has been growing so rapidly in recent years that it has become ripe with market opportunity.

1.1 The Problem

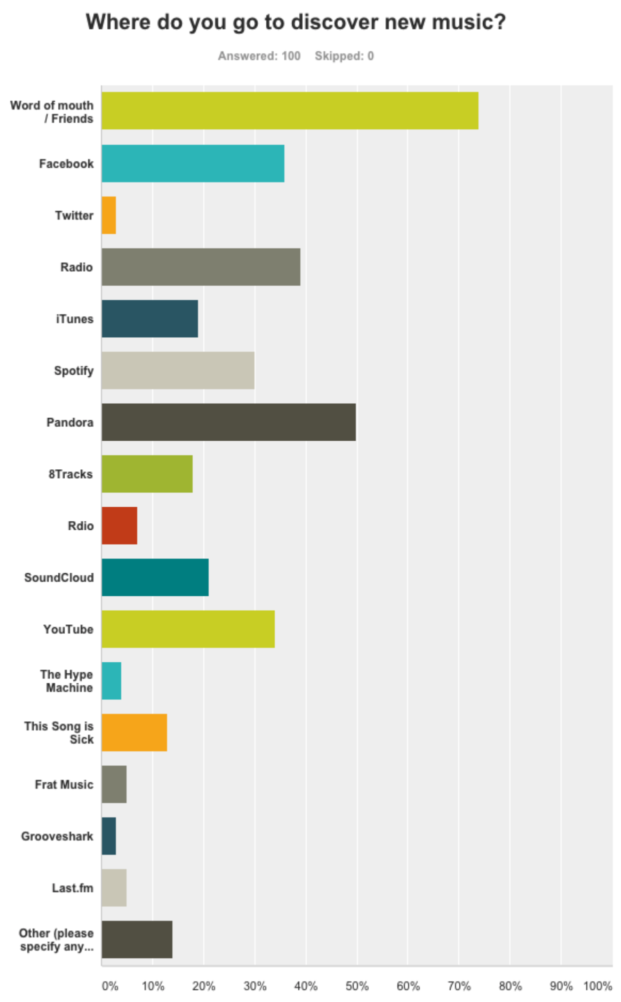
While the rise in music streaming services has opened consumers' access to music like never before, it has also created an issue that was never before present. Now that individuals can consume new music with such ease, they are consistently searching for new material. The problem is, now that they have access to millions of songs, it is harder to make decisions about what music to listen to. Currently, companies have taken two different approaches to solving the issue of music discovery. Services like Pandora and Spotify Radio use algorithms to try to identify a user's taste in music, and provide them with recommended songs accordingly. On the other hand, services like Beats Music and Songza use professionally curated playlists based on different moods or genres to help users identify new songs or artists that they might like.

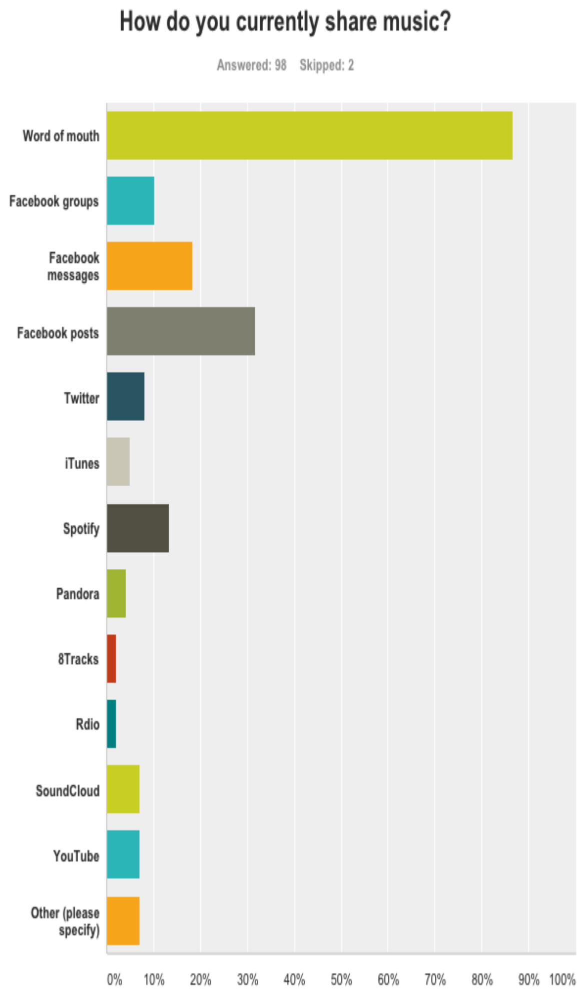
After conducting extensive customer development interviews with over fifty individuals, my team concluded that neither of these approaches were effective solutions to the issue of music discovery. The algorithms behind Pandora and Spotify radio tend to converge on similar artists and lack a sense of spontaneous differentiation when it comes to discovering entirely new genres. The curated playlists of Beats Music and Songza fail to relate to users on a personal level. The tastes of one critic may be very different from individual users, making it hard to consistently find good music. The results of these interviews proved that a new approach to music discovery was needed.

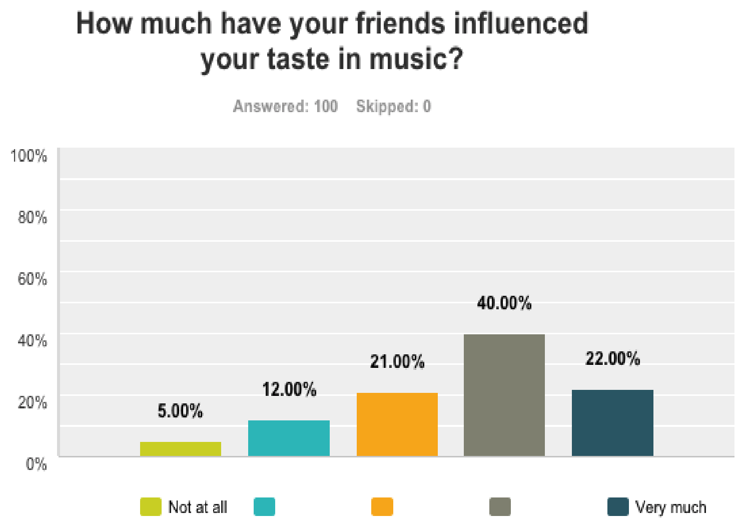
1.2 The Solution

With the creation of Resound.fm, we have taken a new approach to helping users discover music. We connect them with their friends. Back in the day, before the world of music streaming, people made mixtapes of the songs that they loved to share with the people that mattered most to them. Music was something that connected us all, and yet when the world of music went online, somehow that personal connection was forgotten. Resound.fm aims to bring it back.

In order to validate our hypothesis, my team surveyed roughly one-hundred individuals to learn about the ways in which they typically share and discover music. The following graphs show the results obtained through this survey.


Where do you go to discover new music?
Answered: 100 Skipped: 0
Word of mouth / Friends
Facebook
Twitter
Radio
iTunes
Spotify
Pandora
8Tracks
Rdio
SoundCloud
YouTube
The Hype Machine
This Song is Sick
Frat Music
Grooveshark
Last.fm
Other (please specify any...
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%


How do you currently share music?
Answered: 98 Skipped: 2
Word of mouth
Facebook groups
Facebook messages
Facebook posts
Twitter
iTunes
Spotify
Pandora
8Tracks
Rdio
SoundCloud
YouTube
Other (please specify)
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%


How much have your friends influenced your taste in music?
Answered: 100 Skipped: 0
100%
80%
60%
40%
20%
0%
5.00%
12.00%
21.00%
40.00%
22.00%
Not at all
Very much

The results of the survey were clear. People relied on their friends to share and discover new music; however they lacked an online platform that could effectively allow them to do so. In order to solve this issue, my team set out to build a social network in the image of Twitter and Instagram, which put people first and allowed them to share their favorite music with their friends, one song at a time.

2. Application Overview

2.1 Signing Up For Resound.fm

When a user first visits Resound.fm, they are greeted with an initial login page and have the ability to login through Facebook as seen in Figure 1. We initially contemplated using other login methods, but settled on Facebook because it allowed us to easily connect a user to their friends, as well as automatically gather their name and profile picture. We wanted to highlight the individual users as much as possible by giving them a recognizable identity on the platform.

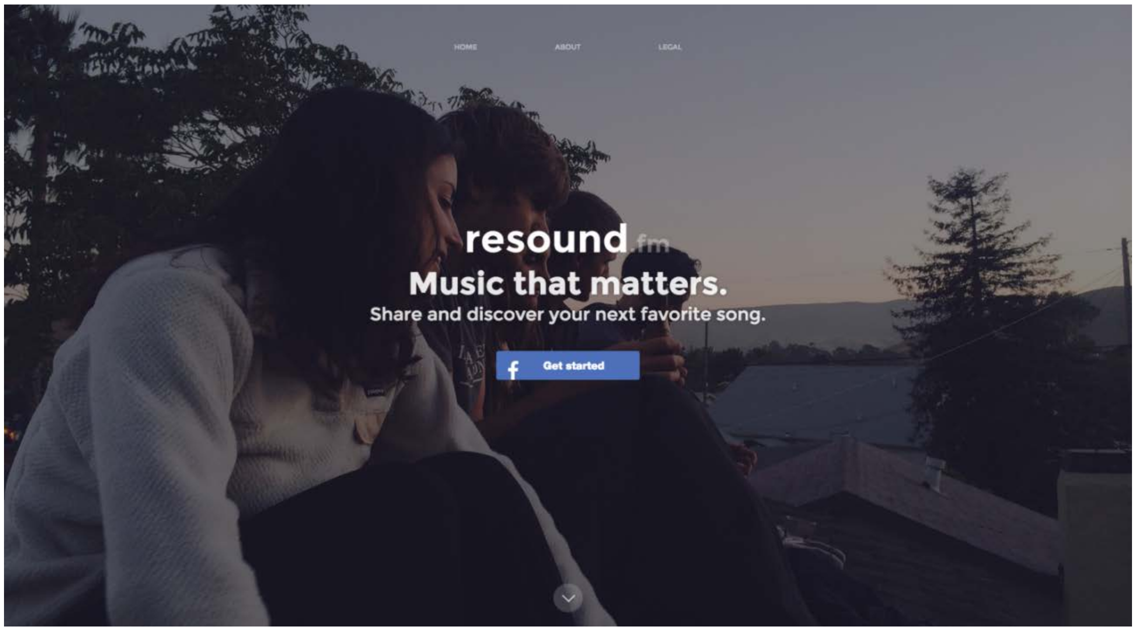
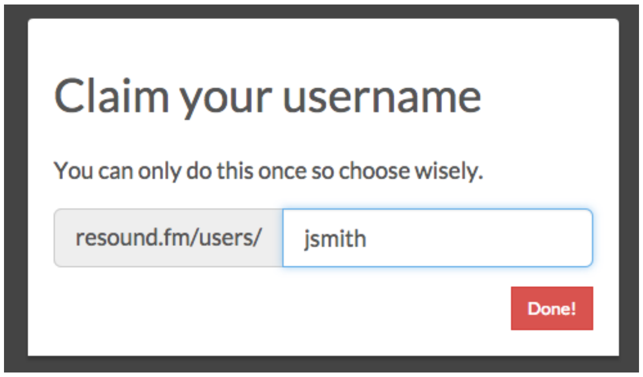


Figure 1: Resound.fm landing page.

2.2 The Onboarding Process

In designing our user onboarding process, my team studied the onboarding processes of various social networking applications in order to determine some best practices. The first step for a brand new user is to claim their own unique username (Figure 2) that allows them to share their Resound.fm profile page online using the link “resound.fm/{username}.” This was the first step to building a social network foundation that would allow our site to be spread through other social networking applications. It allows users to easily showcase their taste in music for others to see and listen to.

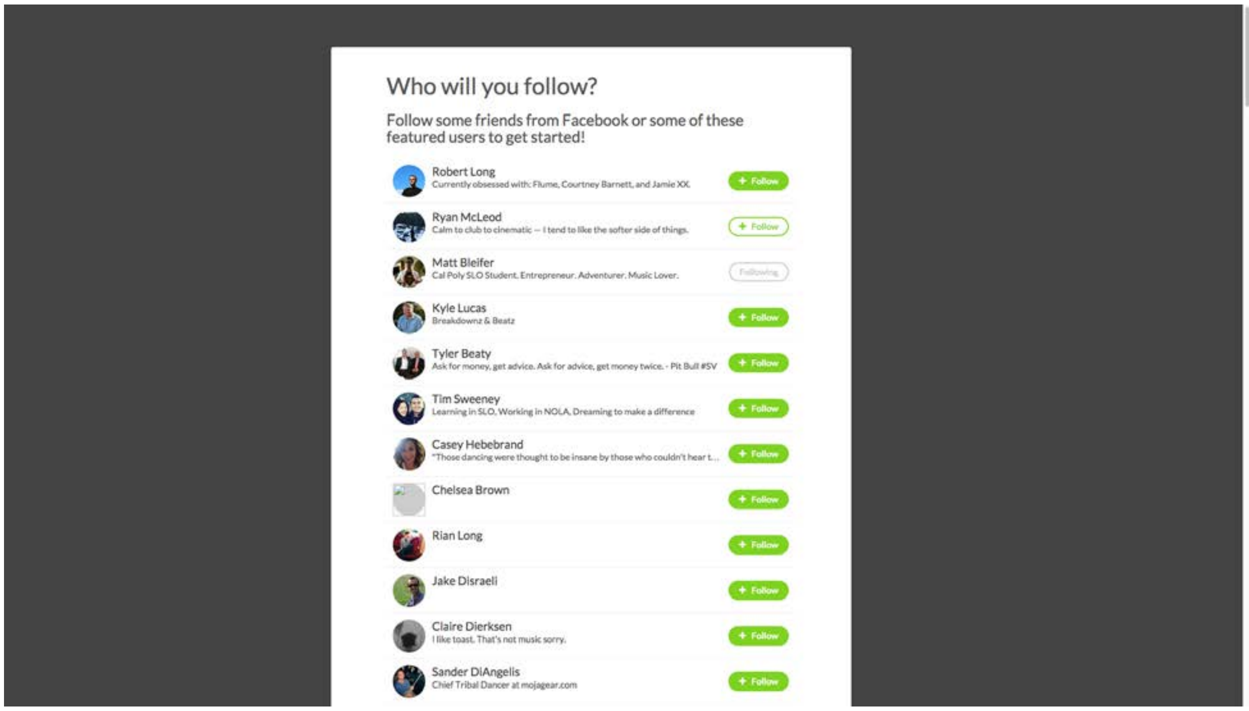

Claim your username

You can only do this once so choose wisely.

resound.fm/users/ jsmith

Figure 2: Claiming a username on Resound.fm.

The next step of the user onboarding process is undoubtedly the most important. This is the step where we connect users with their friends, which can be seen in Figure 3. We knew from the start that the crutch of our whole platform would lie in our ability to make these meaningful connections between users. In following the practices of viral applications like Snapchat, we aim to never bring a user onto the platform before first connecting them with as many of their friends as possible. In order to do so, we leverage the user’s Facebook information that they have previously agreed to provide us with. We compile a list of Resound.fm users that prioritizes first the user’s Facebook friends, then their friends of friends, and lastly the most popular Resound.fm users. Each user’s Resound.fm profile description is shown next to an easy, one-click “Follow” button. We hoped that this would make it as easy as possible for new users to rapidly connect with all their friends and move on to sharing their favorite songs with them.



Who will you follow?

Follow some friends from Facebook or some of these featured users to get started!

Robert Long
Currently obsessed with: Flume, Courtney Barnett, and Jamie XX.

Ryan McLeod
Calm to club to cinematic — I tend to like the softer side of things.

Matt Bleifer
Cal Poly SLO Student. Entrepreneur. Adventurer. Music Lover.

Kyle Lucas
Breakdownz & Beatz

Tyler Beaty
Ask for money, get advice. Ask for advice, get money twice. - Pit Bull #SV

Tim Sweeney
Learning in SLO. Working in NOLA. Dreaming to make a difference

Casey Hebebrand
"Those dancing were thought to be insane by those who couldn't hear t...

Chelsea Brown

Rian Long

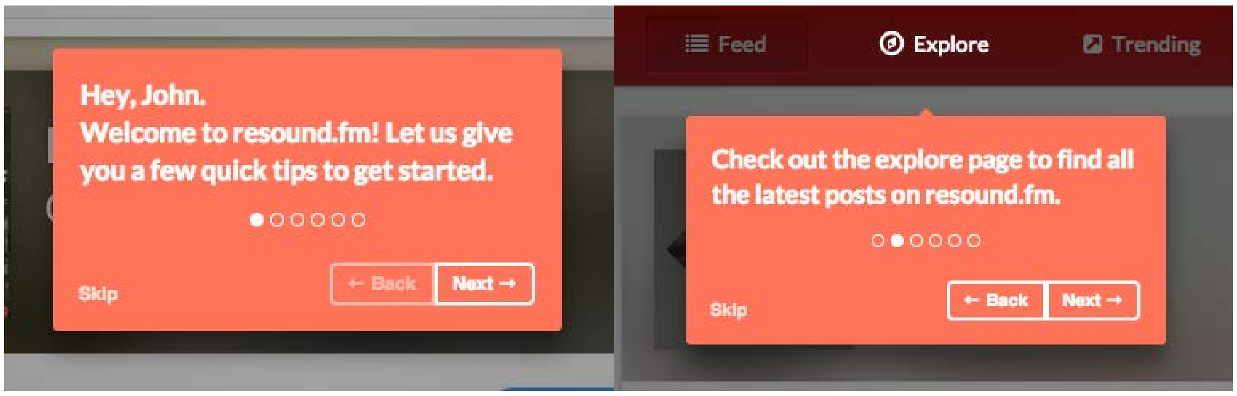
Jake Disraeli

Claire Dierksen
I like toast. That's not music sorry.

Sander DiAngelis
Chief Tribal Dancer at mojagear.com

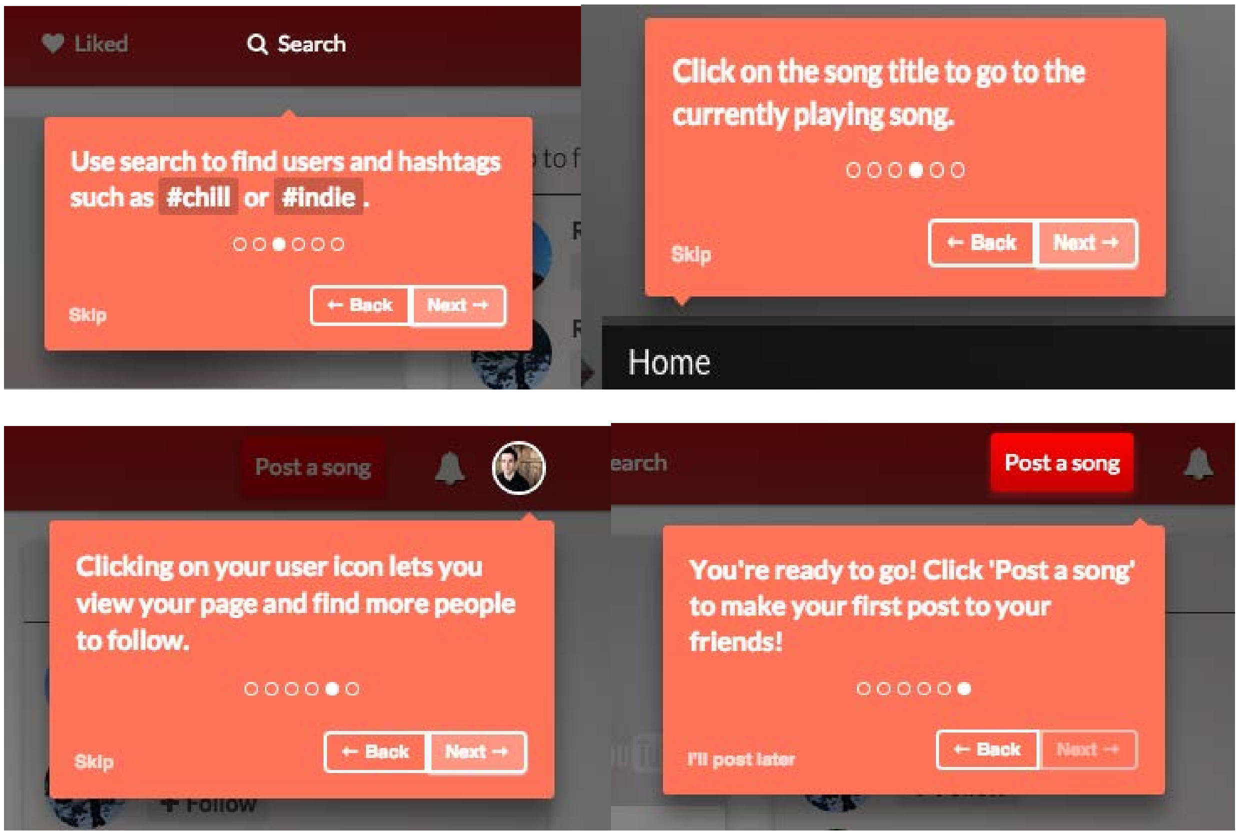
Figure 3: Finding initial users to follow on Resound.fm.

Next in the onboarding process is a quick tutorial (Figure 4) that helps show new users some key features of the application. We didn’t actually implement this tutorial until the application had launched to the public, but in tracking usage statistics we found that it instantly caused certain features of the site to be used more often.



Hey, John.
Welcome to resound.fm! Let us give you a few quick tips to get started.

Check out the explore page to find all the latest posts on resound.fm.



Use search to find users and hashtags such as #chill or #indie.

Click on the song title to go to the currently playing song.

Clicking on your user icon lets you view your page and find more people to follow.

You're ready to go! Click 'Post a song' to make your first post to your friends!

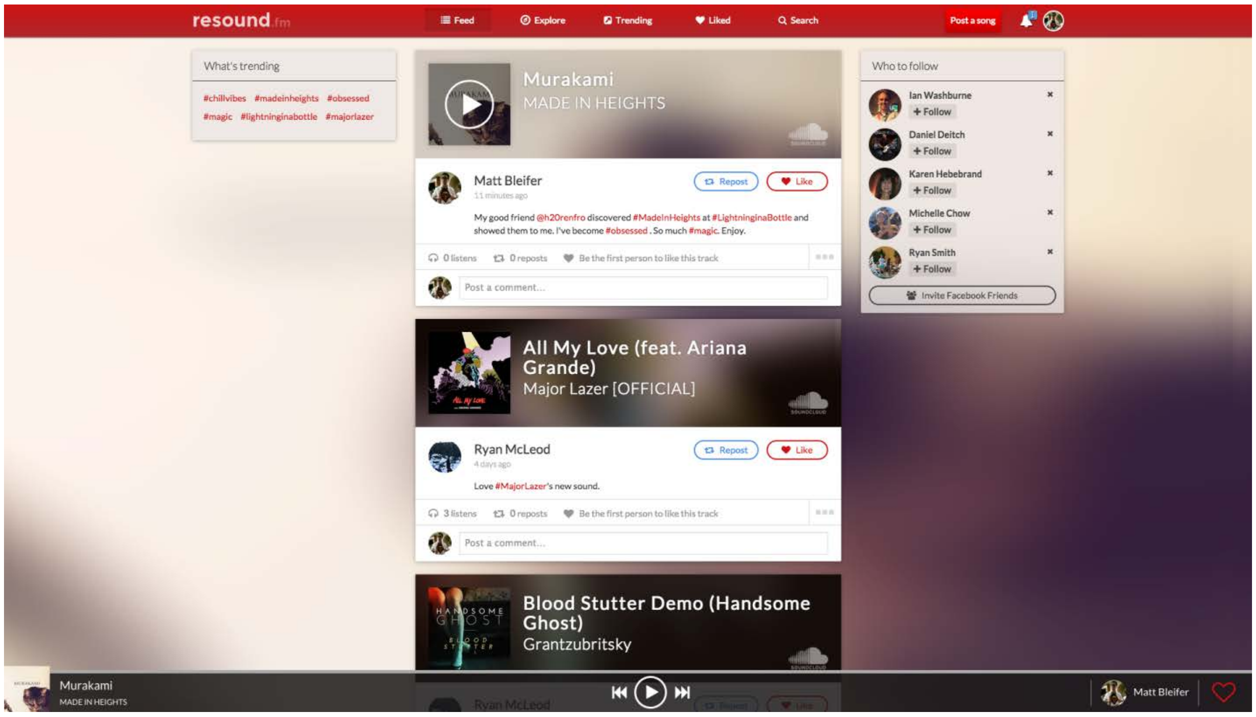
Figure 4: Initial Resound.fm tutorial.

As seen in Figure 4, the last thing we show a new user is how to post a song on the platform. Here we do something that is a little sneaky, and extremely effective. We intentionally disable the “next” button in the last step of the tutorial and instead push users to click the “Post a song” button in order to finish the tutorial.

In studying our usage statistics early on, we discovered that one of the largest factors that determined if a new user would convert to a returning user was whether or not they posted a song on their first session. Our theory, which we later confirmed through customer development, was that this caused a positive feedback loop for the user due to the notifications they received on that first song post. As with any social application, a large part of the value for an individual comes in the form of a positive boost in self-esteem. In interviewing our current and potential customers, we found that people hold a strong emotional tie to their personal taste in music. For many people, music relates directly to their personality or personal experiences they have had. When others express an approval for an individual's taste in music, that individual receives a boost in their personal self-esteem which in turn causes them to share more music. This positive feedback loop is something that we continued to incorporate as much as possible into the rest of the Resound.fm platform.

2.3 The Home Page

Once a user logs in to Resound.fm, they are presented with the home page shown in Figure 5. The following sections of this paper will explore the various pieces of functionality contained within the Resound.fm application.



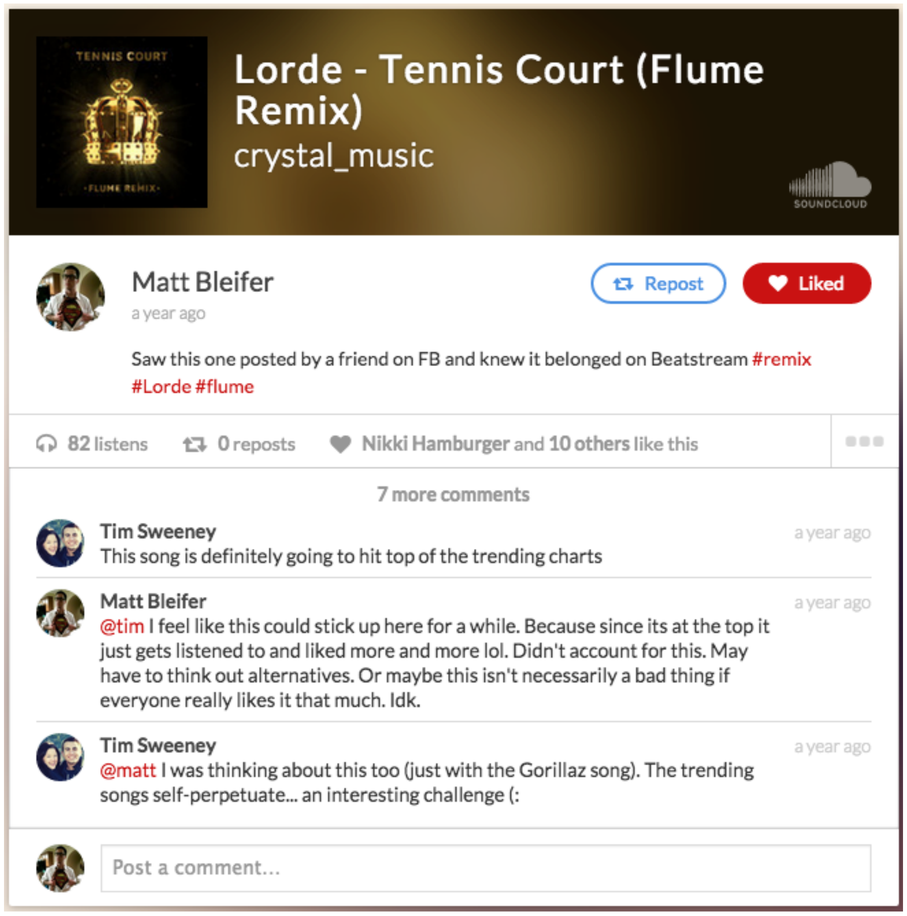
Matt Bleifer
11 minutes ago
My good friend @h20renfro discovered #MadeInHeights at #LightninginaBottle and showed them to me. I've become #obsessed . So much #magic. Enjoy.

Ryan McLeod
4 days ago
Love #MajorLazer's new sound.

Figure 5: Resound.fm home page.

2.4 A Song Post

Each post by a user on Resound.fm represents an individual song, as seen in Figure 6. These songs can be played by clicking on the album art of the song post. Each post contains the name and picture of the user who made the post, as well as the description that user gives to the song. Users can like the post, repost it to their followers, or comment on it. The total number of plays, reposts, and likes for each post are shown as well.



Lorde - Tennis Court (Flume Remix)
crystal_music

Matt Bleifer
a year ago

Saw this one posted by a friend on FB and knew it belonged on Beatstream #remix #Lorde #flume

82 listens 0 reposts Nikki Hamburger and 10 others like this

7 more comments

Tim Sweeney a year ago
This song is definitely going to hit top of the trending charts

Matt Bleifer a year ago
@tim I feel like this could stick up here for a while. Because since its at the top it just gets listened to and liked more and more lol. Didn't account for this. May have to think out alternatives. Or maybe this isn't necessarily a bad thing if everyone really likes it that much. Idk.

Tim Sweeney a year ago
@matt I was thinking about this too (just with the Gorillaz song). The trending songs self-perpetuate... an interesting challenge (:

Figure 6: Resound.fm song post.

In order to increase the likelihood of a user checking out any given song, we make sure to first list which of their friends, if any, like the song, and then allow them to hover over the "others" word in the statistics bar to see the full list of users who liked the post.

Additional options for deleting a post or obtaining a hyperlink to a specific post can be found in the menu bar to the right of the statistics bar in the post (Figure 7).

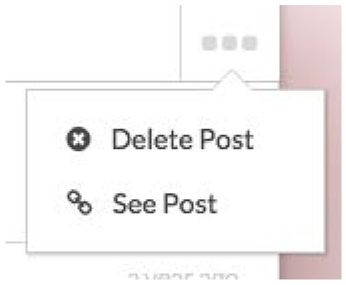


Figure 7: Additional options for a song post.

2.5 Posting a Song

In order to post a song on Resound.fm, a user clicks on the “Post a song” button in the top right corner of the application (Figure 8).

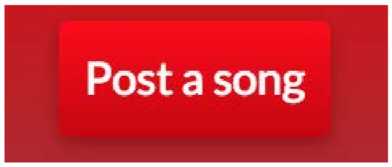


Figure 8: “Post a song” button.

A user is then presented with a search field where they can search for songs based on a song title or artist name (Figure 9). As the user types into the search field, a dropdown list continuously loads results for the search from YouTube and SoundCloud.

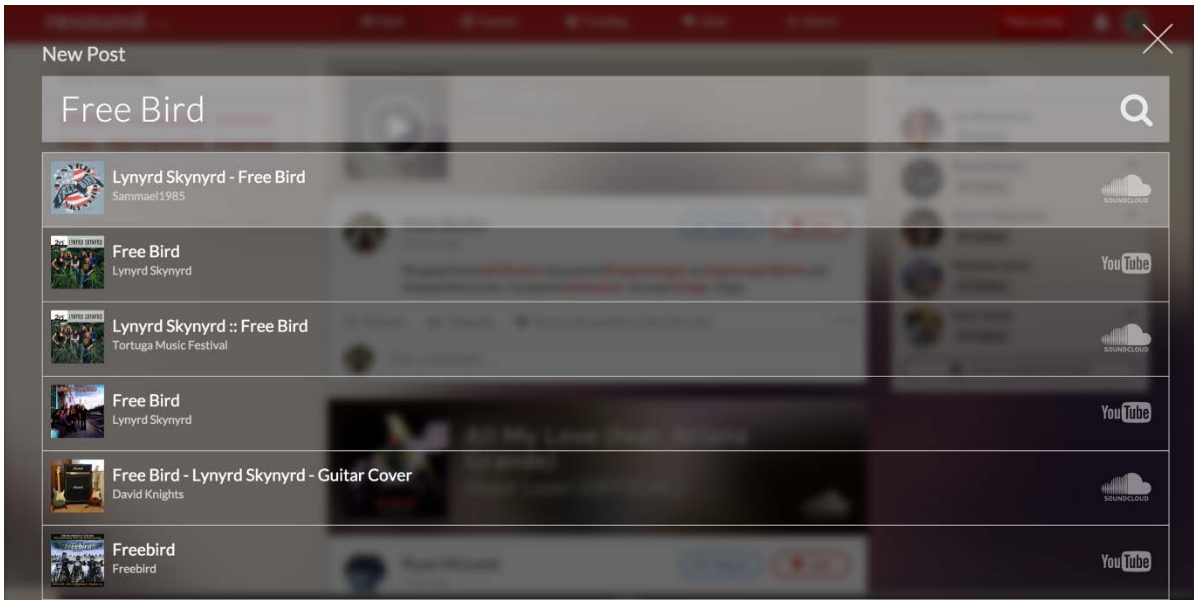


Figure 9: Searching for a song.

Once a user selects a song, they are taken to a screen where they can preview the song and add their description, including hashtags and user profile tags as they please (Figure 10). If the song source is from YouTube, five alternative YouTube sources are loaded for the user to swap the current track with to make sure they are posting the right version of the song.

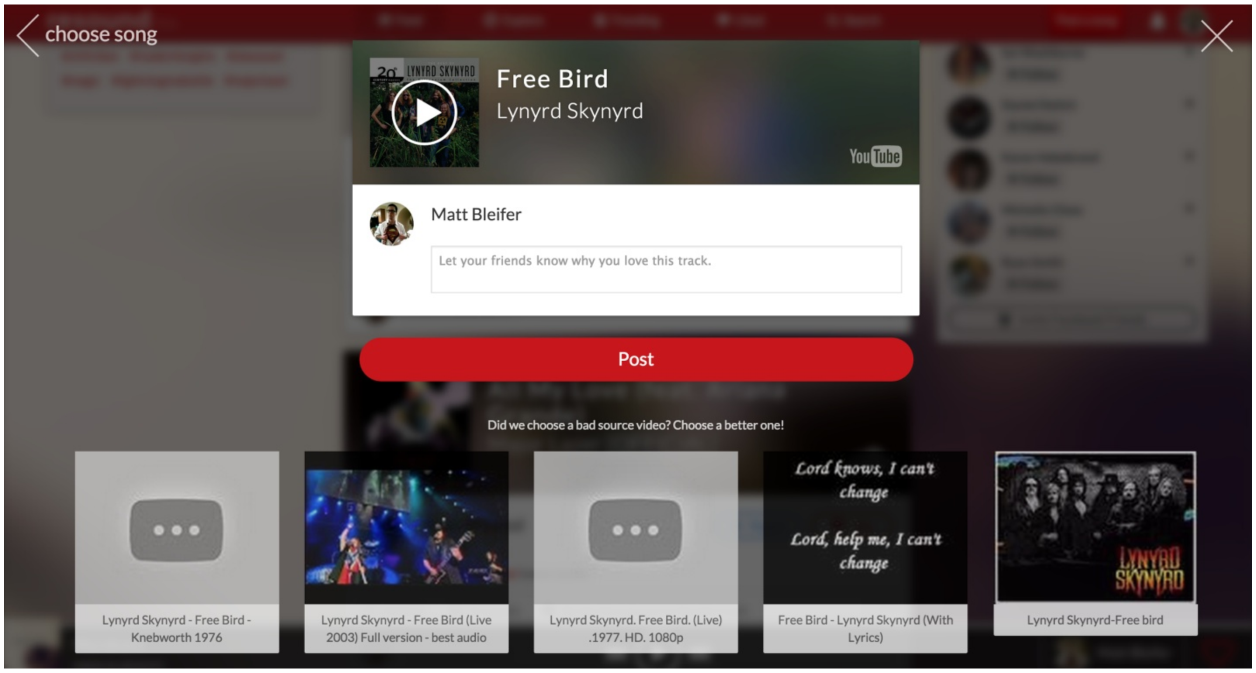

choose song
Free Bird
Lynyrd Skynyrd
YouTube
Matt Bleifer
Let your friends know why you love this track.
Post
Did we choose a bad source video? Choose a better one!
Lynyrd Skynyrd - Free Bird - Knebworth 1976
Lynyrd Skynyrd - Free Bird (Live 2003) Full version - best audio
Lynyrd Skynyrd. Free Bird. (Live) .1977. HD. 1080p
Free Bird - Lynyrd Skynyrd (With Lyrics)
Lynyrd Skynyrd-Free bird

Figure 10: Previewing a song post and adding a description.

Because the value in Resound.fm relies on users posting songs, we had to make sure that the process to post a song was as simple and effective as possible. Several factors had to be taken into account when creating this process. First we had to find a way to obtain access to a nearly unlimited collection of songs at no cost to the user. Then we had to design a way for a user to search and access these songs without leaving the application. Lastly, we wanted to make sure to grab the correct album artwork, title, and artist name for every song. Meeting these demands was no easy challenge, but we eventually came up with a solution.

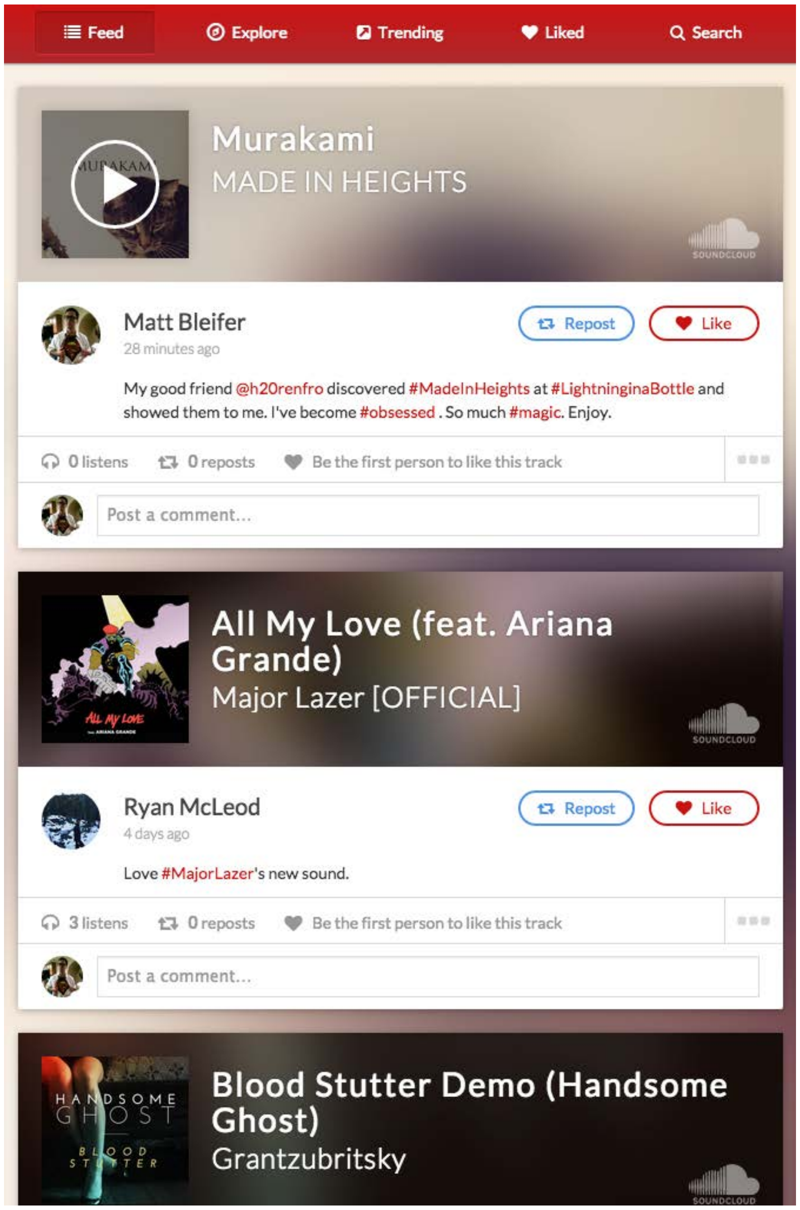
In order to gain access to a nearly limitless collection of music at no cost to the user, we sourced all of our tracks from SoundCloud and YouTube using their API's. We determined that these two sources together contained the vast majority of songs that users wished to post. We were able to load the combined search results from these API's in our application, and allow users to preview and post the songs without ever leaving Resound.fm. The tricky part however, was gaining the correct album art, song title, and artist name for the YouTube tracks.

We wanted to maintain a certain level of professionalism on the site, and not clutter it with YouTube videos containing inconsistent titles. To do this, we actually first return search results to the user from SoundCloud and iTunes. Loading this data from iTunes allows us to grab the correct album art, song title, and artist name for each track. Then, if a user selects one of the iTunes sources, we use the given song and artist names from iTunes as a string by which to search YouTube's API for the correct song. We then apply an algorithm we created that helps us determine which of the results are most likely to be the correct song source. We return the top choice to the user by default, and use the next 5 as the alternate sources that a user could select from.

The algorithm for determining which YouTube track is likely the best audio source takes many factors into account. It first filters out any results whose titles contain the words "cover" or "acoustic." It then weighs in factors such as how close the video duration is to the listed iTunes song duration, the number of likes and views that the video has, and how high in the YouTube search rankings the video is. All of these factors combined allow us to achieve a pretty strong accuracy rate for sourcing songs from YouTube. We then hide the actual YouTube video in the background of the site, while allowing the audio to play for users to enjoy.

2.6 The Feeds

Resound.fm is made up of a variety of "feeds" containing a series of filtered song posts. Each of these song posts are a playable audio track. When one song is over, the song from the next post in the feed will begin playing. This allows users to listen to Resound.fm as a continuous media player, filtering the type of music they listen to by the content of the different feeds. Figure 11 shows an individual feed on Resound.fm.



Murakami
MADE IN HEIGHTS

Matt Bleifer
28 minutes ago

My good friend @h20renfro discovered #MadeInHeights at #LightninginaBottle and showed them to me. I've become #obsessed . So much #magic. Enjoy.

0 listens 0 reposts Be the first person to like this track

Post a comment...

All My Love (feat. Ariana Grande)
Major Lazer [OFFICIAL]

Ryan McLeod
4 days ago

Love #MajorLazer's new sound.

3 listens 0 reposts Be the first person to like this track

Post a comment...

Blood Stutter Demo (Handsome Ghost)
Grantzubritsky

Figure 11: A feed on Resound.fm.

The different feeds on Resound.fm can be accessed in three ways: by using the navigation bar at the top of the screen, by clicking red colored links on the website, or by visiting user profiles.

2.6.1 The Main Feed

The main feed is the default feed that is loaded when a user first logs in to Resound.fm. It can also be accessed by clicking the “Feed” button in the navigation bar (Figure 12).

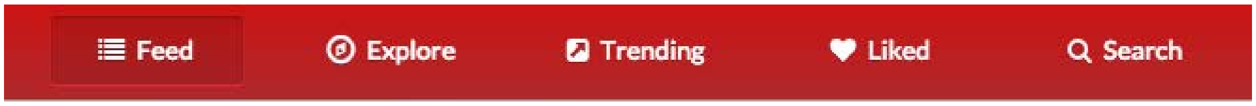


Figure 12: Accessing the main feed.

The main feed contains all of the songs posted by users that an individual is following. These song posts are listed in chronological order in order to make sure that the user is always seeing the latest music that all of their friends are listening to and talking about.

2.6.2 The Explore Feed

The “Explore” feed can be accessed by clicking on the “Explore” button in the navigation bar, as seen in Figure 13.

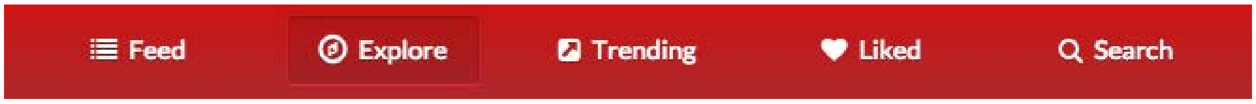


Figure 13: Accessing the “Explore” feed.

The “Explore” feed contains song posts from every user on Resound.fm. Originally we named this feed “Global,” but changed the name to “Explore” in order to give ourselves the freedom in the future to further customize this feed to fit the tastes of the user. We anticipated that if one day there were millions of users on the platform, the “Global” feed would fail to help users find relevant users to follow.

On the “Explore” feed, posts from users that an individual is not following contain a green “Follow” button (Figure 14) that allows the individual to easily follow the user with just one click.

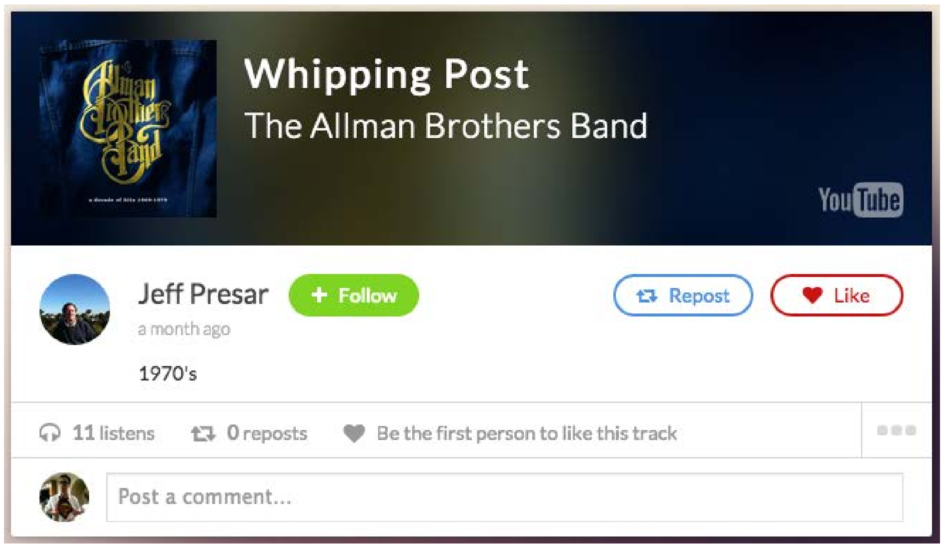


Figure 14: Following a new user on Resound.fm.

2.6.3 The Trending Feed

The “Trending” feed can be accessed by clicking on the “Trending” button in the navigation bar, as seen in Figure 15.

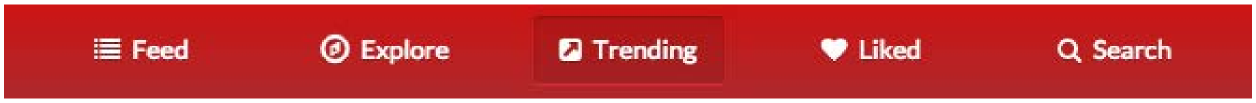
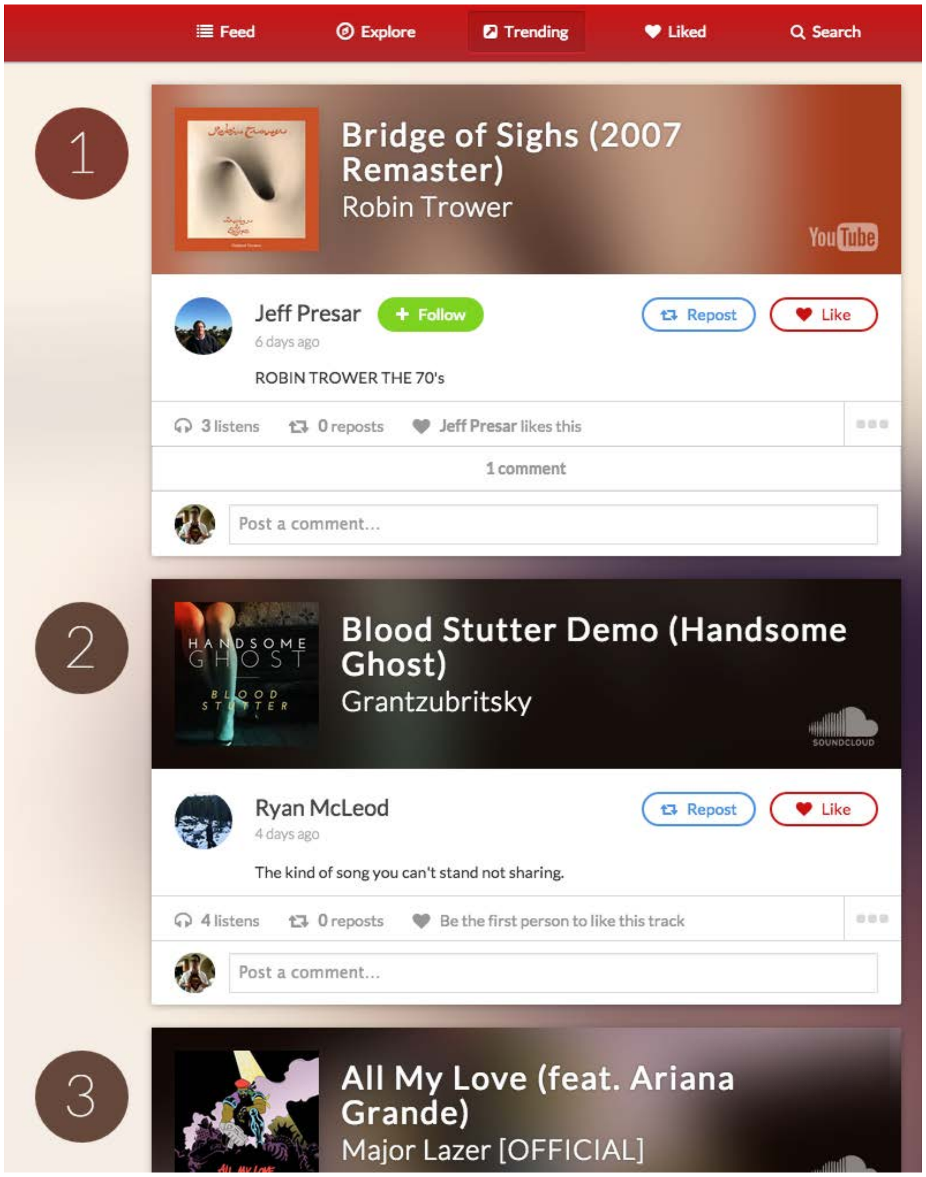


Figure 15: Accessing the “Trending” feed.

The “Trending” feed shows all of the current trending song posts on Resound.fm. Each post includes a number next to it that represents the current rank of the song post (Figure 16).



1

Bridge of Sighs (2007 Remaster)
Robin Trower

Jeff Presar
6 days ago

ROBIN TROWER THE 70's

3 listens 0 reposts Jeff Presar likes this

1 comment

Post a comment...

2

Blood Stutter Demo (Handsome Ghost)
Grantzubritsky

Ryan McLeod
4 days ago

The kind of song you can't stand not sharing.

4 listens 0 reposts Be the first person to like this track

Post a comment...

3

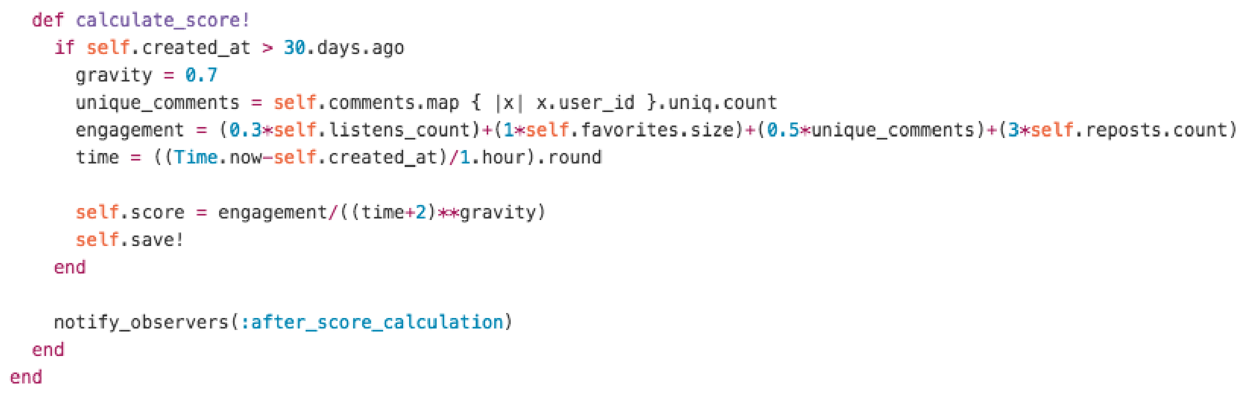
All My Love (feat. Ariana Grande)
Major Lazer [OFFICIAL]

Figure 16: Rankings on the “Trending” feed.

Introducing these rankings greatly increased user activity on the platform. My team found that the simple addition of the numbers next to the posts helped create another positive social feedback loop for users. We witnessed that users began talking to each other and bragging about the rankings that their songs achieved on the “Trending” feed. In order to increase this feedback loop even more, we added a feature to the application that notifies users when their songs reach one of the top ten spots on the trending feed.

2.6.4 The Trending Feed Algorithm

Perfecting the algorithm that ranks the individual posts took a long time and involved a lot of trial and error. We needed to make sure that song posts would not stay near the top of the "Trending" feed for too long, or else other users' posts would never have a chance to rise in rank. Additionally, the songs that were supposed to be currently trending ran the risk of becoming outdated. On the flip side, we couldn't let posts fall out of the trending feed so quickly that they couldn't continue to amass popularity, or give the users who posted those songs a chance to experience the glory of making it to the top. To tackle this issue we developed the following algorithm shown in Figure 17.



```
  def calculate_score!
    if self.created_at > 30.days.ago
      gravity = 0.7
      unique_comments = self.comments.map { |x| x.user_id }.uniq.count
      engagement = (0.3*self.listens_count)+(1*self.favorites.size)+(0.5*unique_comments)+(3*self.reposts.count)
      time = ((Time.now-self.created_at)/1.hour).round

      self.score = engagement/((time+2)**gravity)
      self.save!
    end

    notify_observers(:after_score_calculation)
  end
end
```

Figure 17: The "Trending" feed algorithm.

In this model, each song post on the entire Resound.fm site is assigned its own unique "score" value, by which the "Trending" feed is sorted any time it is loaded by a user. Every time a user interacts with a song post in any way, the "calculate_score" method shown in Figure 17 calculates the new score for that post and updates the database accordingly.

As can be seen in the "calculate_score" method in Figure 17, the trending algorithm first checks to see if the post is less than thirty days old. This is because posts older than 30 days are no longer considered to be candidates for the "Trending" feed. It then evaluates the number of unique users that have commented on that song post. Next it gives the post an "engagement" value, representing how popular that post is. This "engagement" value is determined by multiplying the total number of plays, likes, unique comments, and reposts by individual weight factors and totalling the results. These weights were determined subjectively by my team based on what we viewed as the most significant or valuable actions a user could make on a post. The algorithm then evaluates the final score for the song post by dividing the engagement value by an exponential time decay value. This causes songs to drop exponentially lower in the feed as they get older. My team can easily increase or decrease the speed of this decay by increasing or decreasing the hard-coded "gravity" value in the "calculate_score" method. Through enough trial and error, we eventually settled on a "gravity" value of 0.7 as a reasonable decay strength.

2.6.5 The Liked Feed

The "Liked" feed can be accessed by clicking on the "Liked" button in the navigation bar, as seen in Figure 18.

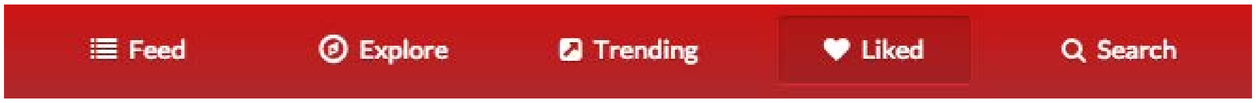
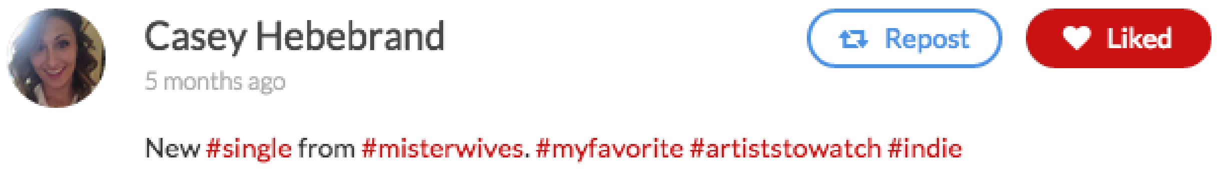


Figure 18: Accessing the "Liked" feed.

The "Liked" feed contains all the songs that a user has liked on Resound.fm. This is where a user can go to easily listen to their favorite new tracks. The songs in this feed are listed in chronological order so that the user can always be experiencing and becoming more familiar with their new favorite tracks.

2.6.6 Hashtag Feeds

Like many modern social networking platforms, Resound.fm incorporates the use of hashtags in posts (Figure 19). Users can view dynamic feeds of song posts that contain a given hashtag in their description or comments. This allows users to listen to music based on anything from an event (e.g. #Coachella), to a mood (e.g. #happy), to a genre (e.g. #rock). The possibilities of hashtag feeds are endless. In order to view hashtag feeds, users can click on red hashtag links, or search for hashtags directly by clicking the "Search" button in the navigation bar (Figure 20).



New #single from #misterwives. #myfavorite #artiststowatch #indie

Figure 19: Hashtags in a song post.

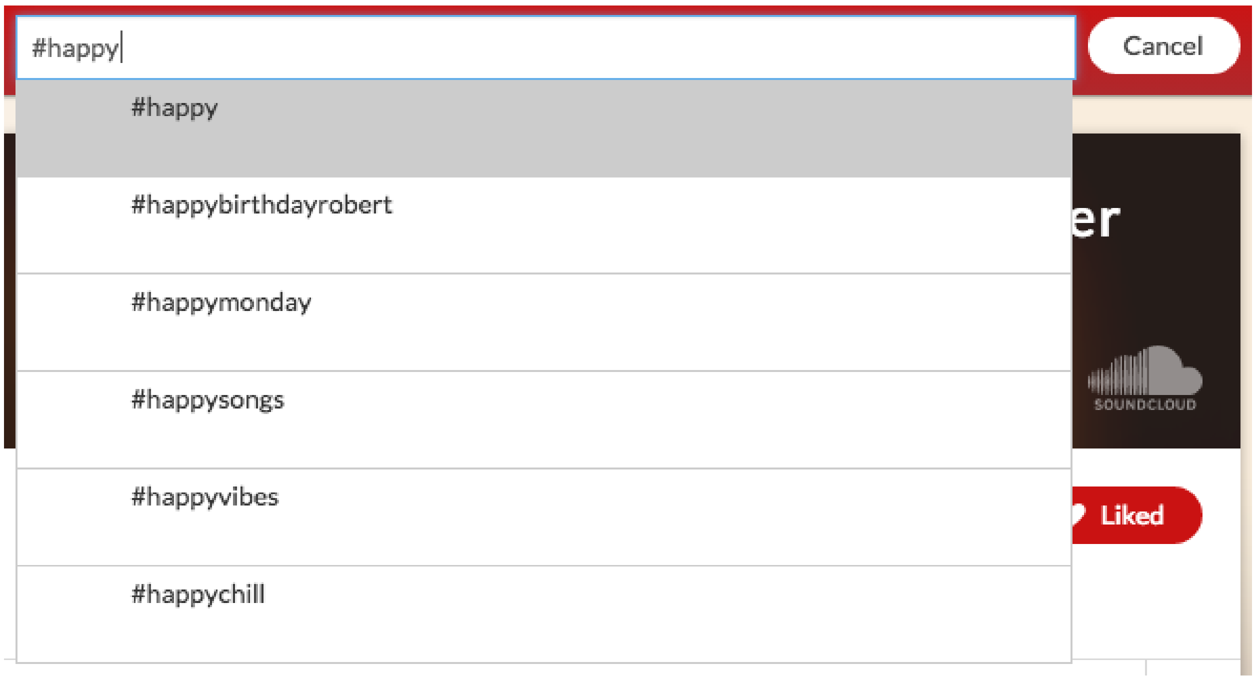


Figure 20: Searching for a hashtag feed.

When a user is viewing a particular hashtag feed, the title of that feed is listed in the top left of the site beneath the Resound.fm logo, as shown in Figure 21.

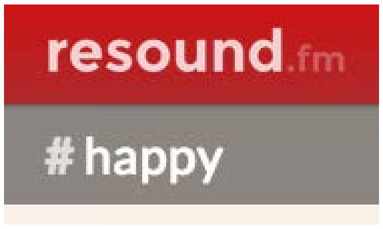


Figure 21: The current hashtag feed.

2.6.7 User Profile Feeds

The last two types of feeds that a user can view are a particular user's profile feeds. These feeds are accessed on a given user's profile page and contain all of the songs that user has posted, and all of the songs they have liked. User profile pages can be found by clicking on a user's profile picture or name, or by using the "Search" feature in the navigation bar as shown in Figure 22.

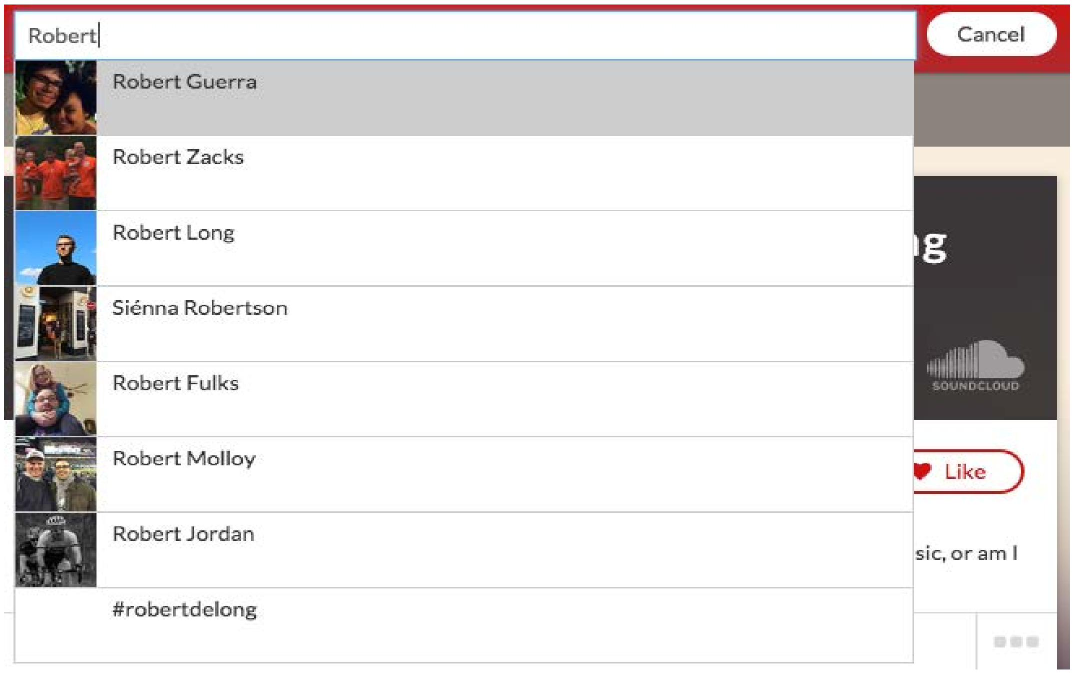


Figure 22: Searching for a user.

2.7 User Profile Pages

Individual user profile pages (Figure 23) are meant to showcase an individual and their taste in music. At the top they contain the name and picture of the user, their personal description, and links to view the songs a user has posted, the songs they have liked, the people that follow them, and the people they follow. Below this is where the song cards or user listings are displayed.

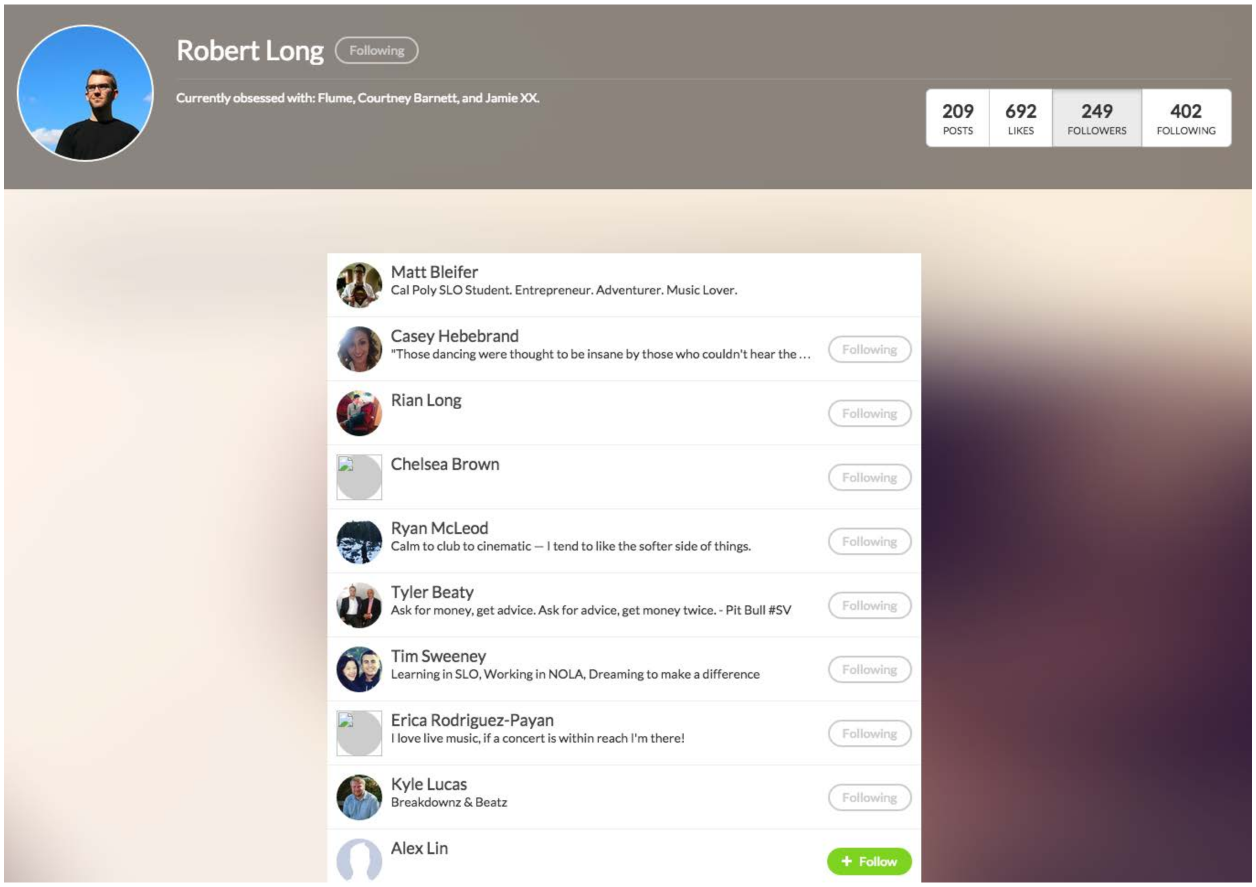
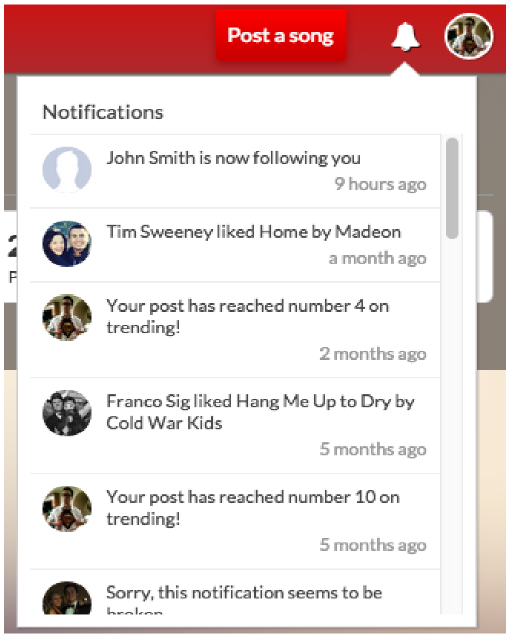

Robert Long
Following
Currently obsessed with: Flume, Courtney Barnett, and Jamie XX.
209 POSTS
692 LIKES
249 FOLLOWERS
402 FOLLOWING
Matt Bleifer
Cal Poly SLO Student. Entrepreneur. Adventurer. Music Lover.
Casey Hebebrand
"Those dancing were thought to be insane by those who couldn't hear the …
Following
Rian Long
Following
Chelsea Brown
Following
Ryan McLeod
Calm to club to cinematic — I tend to like the softer side of things.
Following
Tyler Beaty
Ask for money, get advice. Ask for advice, get money twice. - Pit Bull #SV
Following
Tim Sweeney
Learning in SLO, Working in NOLA, Dreaming to make a difference
Following
Erica Rodriguez-Payan
I love live music, if a concert is within reach I'm there!
Following
Kyle Lucas
Breakdownz & Beatz
Following
Alex Lin
+ Follow

Figure 23: A user profile page.

2.8 The Notification Center

In the top right corner of the site, there is a bell icon which the user can click to view their notifications (Figure 24). The number of unread notifications is signified by a blue number value on top of the bell icon.



John Smith is now following you
9 hours ago

Tim Sweeney liked Home by Madeon
a month ago

Your post has reached number 4 on trending!
2 months ago

Franco Sig liked Hang Me Up to Dry by Cold War Kids
5 months ago

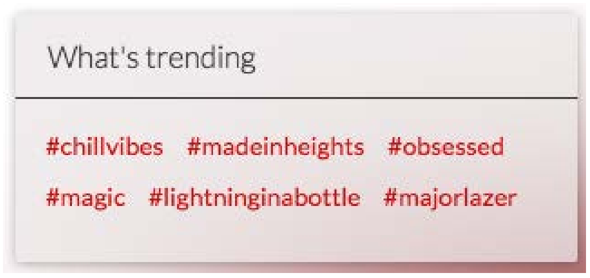
Your post has reached number 10 on trending!
5 months ago

Sorry, this notification seems to be

Figure 24: The notification center.

2.9 What's Trending

The currently trending hashtags on Resound.fm are listed on the left side of the site, as seen in Figure 25. This section is populated by calculating the most commonly used hashtags in the last thirty days.



What's trending

#chillvibes #madeinheights #obsessed
#magic #lightninginabottle #majorlazer

Figure 25: Currently trending hashtags.

2.10 Who to Follow

On the right side of the site, users are shown suggestions of who they might want to follow on Resound.fm (Figure 26). These suggestions are populated by first looking for the user's Facebook friends that they aren't already following on Resound.fm, then their friends of friends, and lastly the top Resound.fm users.

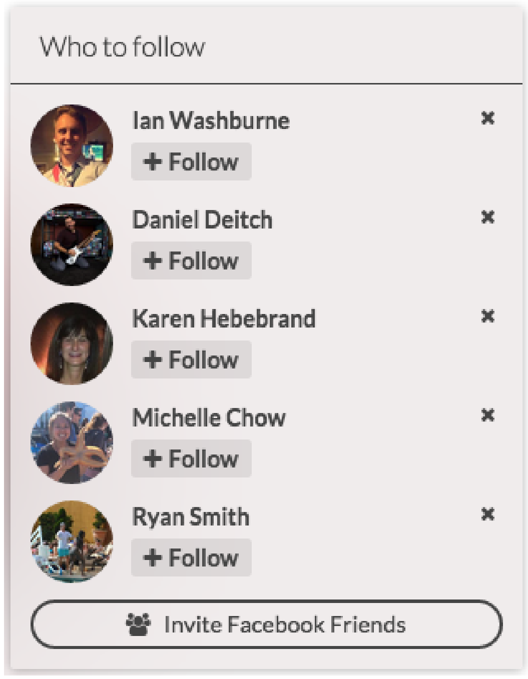


Figure 26: Suggested users to follow.

2.11 User Menu

By clicking on their profile picture icon in the top right of the application (Figure 27), users can view and edit their own profile, explore an extended list of suggested users to follow, or logout of the application.

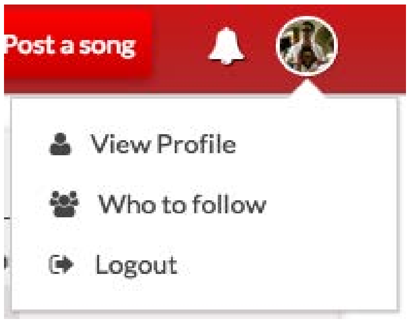


Figure 27: The user menu.

2.12 The Media Player

At the bottom of the application is the media player that controls the currently playing tracks on Resound.fm (Figure 28). Users can click on the song title or album art on the left to be taken to the song post for the currently playing track. They can also see who posted the song, and like the song on the right hand side. Lastly, users can click the red scroll bar to instantly go to any point in the track.

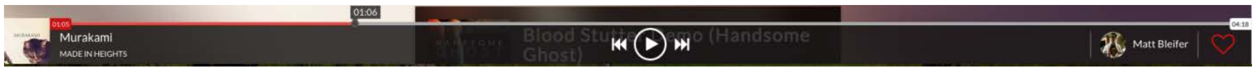


Figure 28: The media player.

3. Application Architecture

At its core, Resound.fm is built as a Ruby on Rails web application, using a PostgreSQL database, and follows the Model-View-Controller design paradigm. However, unlike many traditional web sites, Resound.fm functions as a single page web application. When a user first makes a request to our servers under any route, the server delivers all of the Model-View-Controller logic to the client's browser. When a user then takes an action on the site that requires data to be changed, an asynchronous function call is made to the API on our servers, and upon completion, the views on the site which are bound to that data are re-rendered as necessary. This makes it so that the page never needs to be fully refreshed, and the application can run much quicker during the duration of a user session. This also allows us to keep the audio tracks always playing through the client's browser so long as the website is open. Essentially our server side code is nothing more than a controller that serves a single view, and hosts a large API.

3.1 React.js

React.js is a JavaScript library developed by Facebook that helps developers build user interfaces in web applications. It is most commonly used as the View part of a Model-View-Controller design. React.js allows developers to easily update DOM elements efficiently based on changes to the models. To do this, React.js builds a virtual DOM between the application and the DOM (Figure 29). It then handles DOM updates in batches, calculating a nearly-minimal diff and applying it to the DOM in as few steps as possible. React.js helped us ensure that the media controller element at the bottom of our application would never get changed unnecessarily and cause music to stop playing.

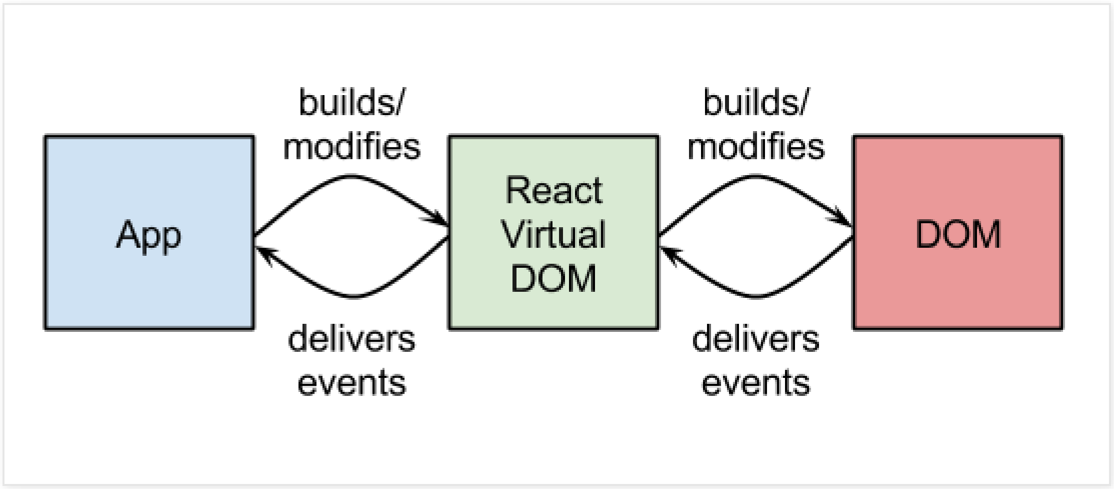
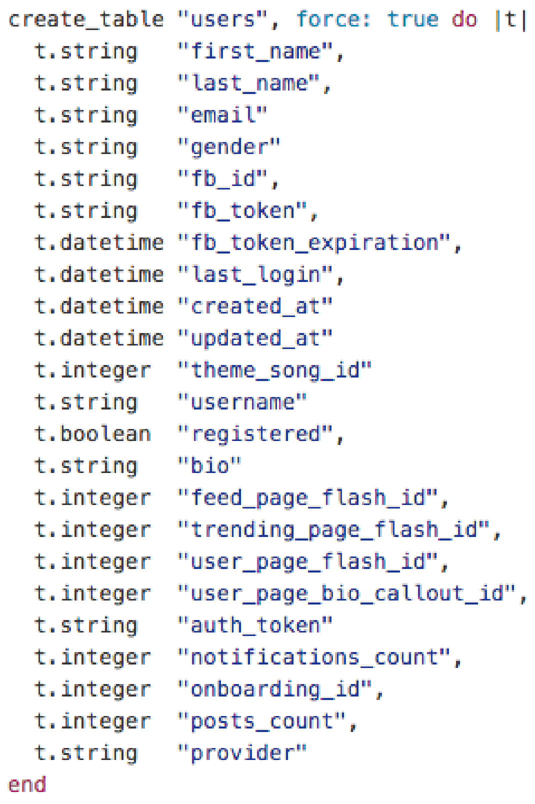


Figure 29: The React.js virtual DOM.

3.2 The User Model

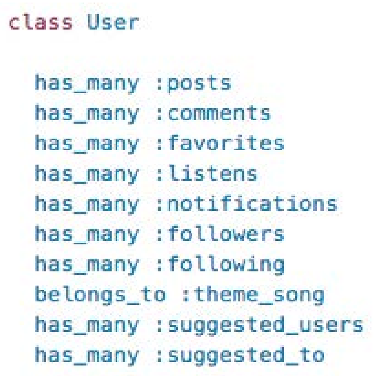
The “User” model contains all of the information for a given user on Resound.fm. It has fields that include the user’s first name, last name, email, gender, bio, and more. Figure 30 shows how the “users” table is created in our database with the complete list of fields for a user.



```
create_table "users", force: true do |t|
  t.string   "first_name",
  t.string   "last_name",
  t.string   "email"
  t.string   "gender"
  t.string   "fb_id",
  t.string   "fb_token",
  t.datetime "fb_token_expiration",
  t.datetime "last_login",
  t.datetime "created_at"
  t.datetime "updated_at"
  t.integer  "theme_song_id"
  t.string   "username"
  t.boolean  "registered",
  t.string   "bio"
  t.integer  "feed_page_flash_id",
  t.integer  "trending_page_flash_id",
  t.integer  "user_page_flash_id",
  t.integer  "user_page_bio_callout_id",
  t.string   "auth_token"
  t.integer  "notifications_count",
  t.integer  "onboarding_id",
  t.integer  "posts_count",
  t.string   "provider"
end
```

Figure 30: The “users” table.

The “User” model also contains many associations to other models in the Resound.fm schema. These associations include references to the posts that a user has made, the posts they have favorited, the other users they are following and more. The full list of associations is shown in the code in Figure 31 which comes directly from the “User” model.



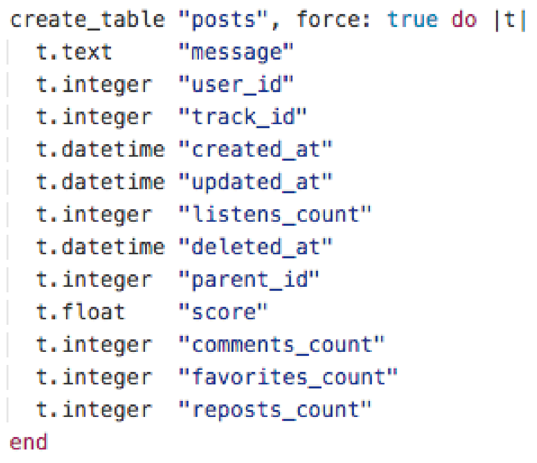
```
class User

  has_many :posts
  has_many :comments
  has_many :favorites
  has_many :listens
  has_many :notifications
  has_many :followers
  has_many :following
  belongs_to :theme_song
  has_many :suggested_users
  has_many :suggested_to
```

Figure 31: “User” model associations.

3.3 The Post Model

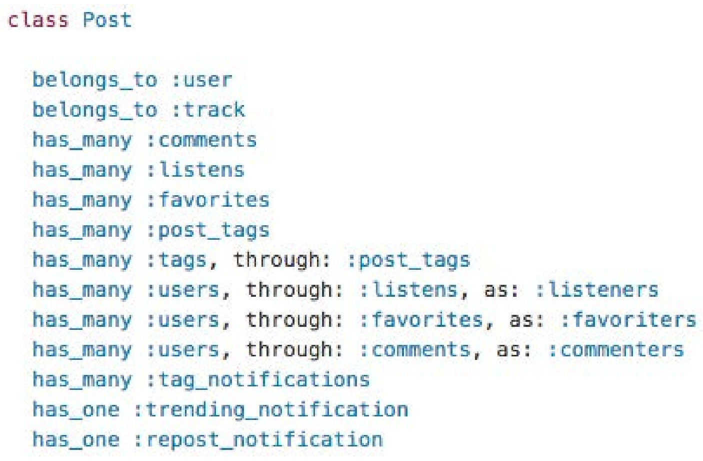
The “Post” model contains all of the necessary information for a single song post on Resound.fm. It includes fields representing the description message, comments count, score and more. Figure 32 shows how the “posts” table is created in our database with the complete list of fields for a post.



```
create_table "posts", force: true do |t|
  t.text     "message"
  t.integer  "user_id"
  t.integer  "track_id"
  t.datetime "created_at"
  t.datetime "updated_at"
  t.integer  "listens_count"
  t.datetime "deleted_at"
  t.integer  "parent_id"
  t.float    "score"
  t.integer  "comments_count"
  t.integer  "favorites_count"
  t.integer  "reposts_count"
end
```

Figure 32: The “posts” table.

The “Post” model also contains associations to other key models in the Resound.fm schema. These associations are shown in the code in Figure 33 which comes from the “Post” model. As can be seen in Figure 33, a post has associations to the user that made the post, the song track it belongs to, the comments it has, and more.



```
class Post

  belongs_to :user
  belongs_to :track
  has_many :comments
  has_many :listens
  has_many :favorites
  has_many :post_tags
  has_many :tags, through: :post_tags
  has_many :users, through: :listens, as: :listeners
  has_many :users, through: :favorites, as: :favoriters
  has_many :users, through: :comments, as: :commenters
  has_many :tag_notifications
  has_one :trending_notification
  has_one :repost_notification
```

Figure 33: The “Post” model associations.

4. Lessons Learned

The journey of building Resound.fm was not always an easy one. As with any large scale project mistakes were made and lessons learned. This section will outline some of the key lessons I took away from this experience.

4.1 Teamwork Is Everything

There is a common phrase in the startup world that “an A team with a B idea is better than a B team with an A idea.” I have found this to most certainly be true. The potential of a given software project is limited not by the technical skill of the developers, but by how well developers can work together. If communication does not flow properly in a team, these issues will be reflected in the quality of the product. In order to combat this, one must build a team that cooperates well, and establish communication expectations and standards from the beginning.

4.2 Avoid Bikeshedding

The term “bikeshedding,” otherwise known as Parkinson’s Law of Triviality, has become a common term in product management. It refers to C. Northcote Parkinson's 1957 argument that organisations give disproportionate weight to trivial issues. Sometimes without realizing it, teams can spend far too much time worrying about small issues in a product because those issues are easier to grasp. This can cause major issues to go completely unaddressed. Resound.fm was no exception to this issue. My team often spent far too much time worrying about tiny design details instead of paying attention to bigger issues that were on the horizon. Even though the details of a product do matter, it is important not to let perfection get in the way of productivity.

4.3 Address Problems Early

Another major lesson I took away from my Resound.fm experience was to address development issues early. Towards the start of our development process we ran into some technical issues in sourcing the songs for the site and building a media player around it. However, because there were so many other things to worry about, we continued to sweep this issue under the rug. Over time it led to a lot of bugs in the site that we continued to try to patch one by one. It wasn’t until much later that we came to our senses and decided to do a full rewrite of the media player, only after specifying all the details and having all the developers review and agree with the development plan. This mistake cost us about a week and a half of development time, which in a startup might as well be years. This taught me that it is crucial to address the most painful parts of the application first and to do things the right way instead of constantly patching up code.

5. The Team

Resound.fm would not have been made possible without the help of my three incredible co-founders: Robert Long, Ryan McLeod, and Kyle Lucas. All three of these members contributed an immense amount to the development and design of the application, as well as the development of the business. Robert Long served as our CTO and worked on all parts of Resound.fm, ensuring all the pieces of the puzzle worked together properly. Ryan McLeod was our Lead Frontend Developer and made great strides in revamping the design of the application in order to deliver a polished product. Kyle Lucas was our Lead Backend Developer and made sure that Resound.fm could scale for many users while still functioning at great speeds. Lastly, I served as our CEO and focused primarily on product management and business development. These three individuals are some of the smartest people that I have ever had the pleasure of working with, and without them Resound.fm would not be everything it is today.

The Resound.fm team from left to right:

Ryan McLeod, Kyle Lucas, Robert Long, Matt Bleifer

6. Conclusion

The experience of building Resound.fm was one of the greatest learning experiences of my life. I gained immense knowledge in the area of web development and got a chance to explore some of the latest web development technologies. I also got a chance to try my hand at product management and learned what it takes to effectively manage a team of developers. I spent a great deal of time interacting with end users and drafting requirements for the system. I also drove the larger vision of the product, while diving deep into the details of the design. There are many things I would do differently in the future, but overall I would call Resound.fm a success.

Today Resound.fm is live for the public to use and has over one thousand users on board who have posted over twenty-thousand songs. In the future we may decide to implement a mobile application to complement the web app and continue to drive user growth. I think that Resound.fm truly has the potential to disrupt the music streaming industry today and create a powerful new music sharing and discovery experience.

7. References

1. Resound.fm

Available at http://www.resound.fm

2. Ruby on Rails

Available at http://rubyonrails.org/

3. PostgreSQL

Available at http://www.postgresql.org/

4. React.js

Available at https://facebook.github.io/react/