Q: locally synchronized social music.

Yvan: Chief Systems Engineer, Objective C Specialist, Designer, iOS Developer (music playback specialist), Board Member, Director of Operations, CFO, Trigonometry Engineer, Lead Contrarian Theorist, Smoker of Herbs

Alec: UI/UX engineer, front end developer, head of iOS dev, designer, head of web dev, Chairman of the board, CTO, Manager Of Personnel, Chief Printer of Posters

Sam: Network Engineer, full stacks developer, design dispute resolver, Head of Graphic Design, “Humanities Major”, CEO, Supplier of Post Its, DJ,

Problem: People want an effortless way to discover new music from friends without feeling pressured to.

Solution: Q is a crowd-powered local radio queue, which synchronizes a shared musical experience across many speakers.

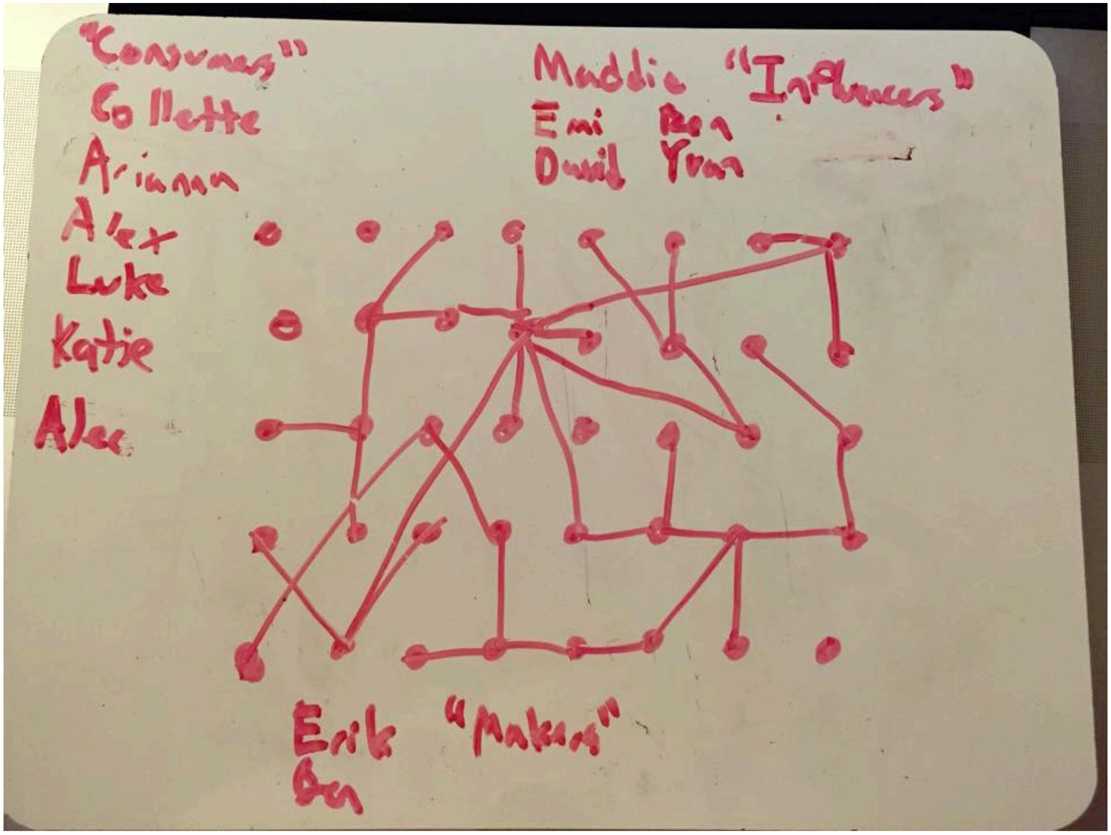
Music so far has been fairly autocratic. Whoever is playing is a dictator. Whoever isn’t chaffs under the yoke of being forced to bend their will to that of the autocrat. 3 hours of taylor swift on a car ride through central California inspired me to bring freedom and democracy to the land of music playback.

I don’t think anyone intended to create an autocracy. It was a byproduct of shortsightedness and technological limitations, a legacy from an ear where freedom and human rights carried less weight. Fortunately, times have changed. Smartphones are ubiquitous. And music is growing in terms of diversity and quantity. However, over the course of the history of music playback the theory behind it has not changed much. Wax cylinders, vinyl, tapes, cds, and mp3s do not vary much. The user selects a song and plays it. There have been trends. Music has become more portable. Listening experiences more tailored (headphones to subwoofers) but the premise remains the same. This does not mean people have not tried. Apple’s Ping sought to make music more social. My Space and Napster can be viewed as attempts at reinventing how users interact with music. Spotify, Pandora, Shazam, Youtube, and Soundhound have all experienced degrees of success at become “social” but none of the three really offers much novelty in terms of playback. The most shocking thing about music consumption isn’t how much but how little is has changed.

We want to change that. Music can be done better. Something that all three of us cared very deeply about was experience. We wanted to find a way to share something unique. Music didn’t have to be an autocracy. Music does not have to be the cacophony of infinite that is Spotify or the cold, algorithmic playlists of Pandora. The limitations of the absolutist approach taken by many platforms, an approach intrinsic to the server model, where one solution or algorithm attempts to determine the most optimal treatment of every node, seems limited in the context of such a human area as music. We also wanted help connect people, breaking the feedback loops that are so common in modern society, (we tend to associate with people similar to ourselves, unintentionally diminishing the probability of encountering anything novel, and causing us to reflect even more the traits that define us). In the modern world, it is often easier to talk to someone on the other side of the world than talk to your neighbor, and chances are you will have more in common with the person across the planet. In terms of music, this is terrifying.

We started off our project by doing a very thorough study of the consumption of music. Interviews, questionnaires, and discussion filled many hours. Eventually, after a particularly productive afternoon we developed a model.

We spent about three hours discussing the roles of our various POV groups in terms of network dynamics and realized that a choice few “influencers” have a vast influence on what most people listen to (i.e. everyone’s “one friend” they get music from is shared by many others). So in effect, our solutions focused on ways to connect these people with as many “casuals” (people who passively absorb music) as possible. In addition to those two groups, there is another outlier of people who view music as an art and want to make the listening experience as pure as possible. This group tends to have more niche tastes, and also overlaps significantly with musicians. Keeping this in mind, another goal was to connect these “makers” with other listeners or musicians with similar niche tastes. Below is a rough visualization of this network (dots = users, lines = musical connection).



It was eerily consistent with our fear of feedback loops. People got music from a key specific friends and the evolution of their “tastes” seemed to over represent the influence of these key individuals. While this is not a bad thing in and of itself, it does fit the feedback loop model which does seem sub optimal.

Q: locally synchronized social music.

As we continued to study the results of our investigation we broke it down into three key use areas. Parties, discovery, and chilling. Parties are large, anonymous social events. As far as we can tell the only person who liked music at parties liked it only when she was “really fucking drunk”. This was really interesting. We think that people do not think its “cool” to like music at parties which warped the feedback some but our data still indicates a high degree of satisfaction. Discovery is the use case we were arguably most excited about. Finding new music is hard. For most people, it was either something that just “happened” or was a result of song recommendations (which had a surprisingly high fail rate). Discovery is very difficult for most of the people we asked. We also learned that we were relatively proactive as music discoverers. Chilling was our third and final use case. How can we make average music consumption better? There was incredibly diversity in this category among the people we interviewed. Some people worked to music. Some never really played their own. Others lived and breathed music. We decided to try to develop a solution that addressed all three of these use cases. We wanted to improve how players interacted with people.

For individuals, the key is flexibility. Music is extremely contextual. Playing it on headphones, on speakers, at parties, in the shower, while working. We really wanted to fix the social aspect of music. How can music, in the context of hanging out, be better?

For parties we wanted to find a way to improve the quality of music and turn an experience that is generally viewed as grating into an opportunity for discovery.

For discovery we wanted to facilitate accessibility to new music. We realized very quickly that there was a tradeoff in terms of cost, in terms of discomfort, and reward in terms of “new good music”. Listening to random music typically causes rates of discomfort that exceed the value of the new music acquired. Our goal was to find a way to alter the ratio to make music discovery less painful. The obvious way to do this was find some constraint besides similarity to use as a selector for what music to expose users to. We opted to use proximity, mostly because we are all fucking romantics and believe there is something special about the world around us. That there is something truly special in bringing beauty out of woodwork and using the digital world to enhance your perception the physical, rather than desaturate it.

Mesh networks offered an incredibly interesting opportunity to change the way music was played. One of the real appeals of mesh networks was how fundamentally different they were from most other technology. They offer very direct and very literal connections between people, with essentially no layers of abstractions. The lack of servers, websites, and credentials seems to hold a degree luddistic purity that really appealed to our mission to improve music.

Social is intrinsic to mesh networks, assuming the other nodes are people. Over the course of our research we often found ourselves reliving the timeless debate of distributed vs centralized power. Centralized power offers a much greater ability for an individual to expose others to “new” music but also takes away their ability to choose and increases the risk “bad” music. Distributed power, at its extreme, could block all new and therefore unknown music. Democracy, Republicanism, and Autocracy were all redefined in terms of music playback. We all agreed that music could not be worse than in its current system, relying the mandate of heaven or the loudest stereo system.

Q: locally synchronized social music.

Once we embraced mesh networks and democracy our real task became implementation. A common queue curated democratically playing crowd sourced music. This raised a variety of other questions: what music do we play? I personally rely almost entirely on Spotify for all my music. One of my roommates uses primarily amazon. How do we capture people's music? We decided to use local files simply because we feel that people are more emotionally connected to things they own. In a sense elements of our app are supposed to be anachronistic as we are exploring an alternative to the current status quo. We decided to build an alternative to the modern consumption culture around music, to turn music into a restaurant rather than a grocery store. In a sense modern music streaming is evacuative of Borges' story, the library of babel, where infinite information is meaningless. If a library contained all potential information it would contain no information. We think that this phenomena is damaging music for consumers. A variety of studies have indicated that choice actually lowers consumer satisfaction. So we decided to askew streaming services in favor of good old local files.

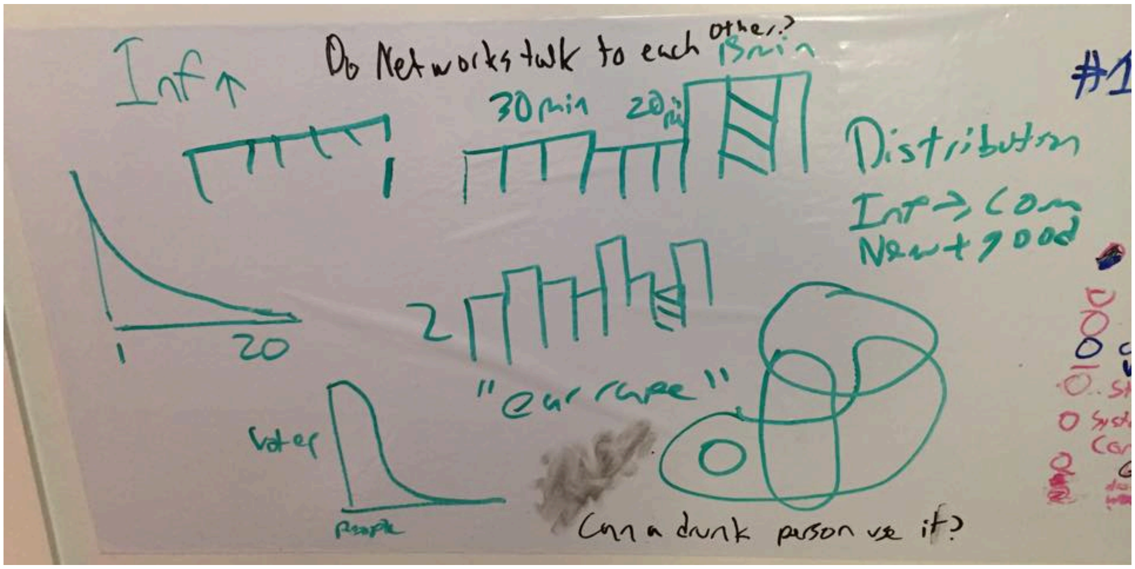
Our needfinding was extremely important in crystallizing what we tried to accomplish with Q. Our how might we's are extremely revelatory in this regard:
How might we connect musicians to each other and listeners around them? (Erik/Emi)
How might we make music listening more social and less labor intensive? (Erik/Colette)
How might we keep music from getting old? (Colette)

There was an idea that we played with in our need finding phase, that, while we opted not to pursue it, we spiritually incorporated into our final produce. In fact several of our key UI elements were first prototyped in this idea. "Rave" was an app for instantaneous spontaneous parties. Users could set a place and time and would get "rep" for people actually showing up. The compass, which evolved into a record, came from this project. Much of the social ideology, connecting people who would otherwise never interact, carried over as well.

The tasks we chose to represent our mission of facilitating music sharing in all forms has undergone many iterations, but has mainly centered around the fundamentals of making this system work: joining a network and voting on a song in the queue (both one button actions) were our simple tasks at different points in the design process, while uploading your own song was the medium task the whole time, as it requires going into your library and picking a song, while many people will only listen to the network passively. Our third (complex) task has fluctuated wildly based upon where our current design trajectory in considering really far-out ideas. At this point in the design process, our third task was "as a dj, share a song," as we were toying with implementing "God Mode" for social party polling. We thought it would really cool to use smartphones as a direct bridge between listeners and performers, actively getting opinions on what the dj is playing.

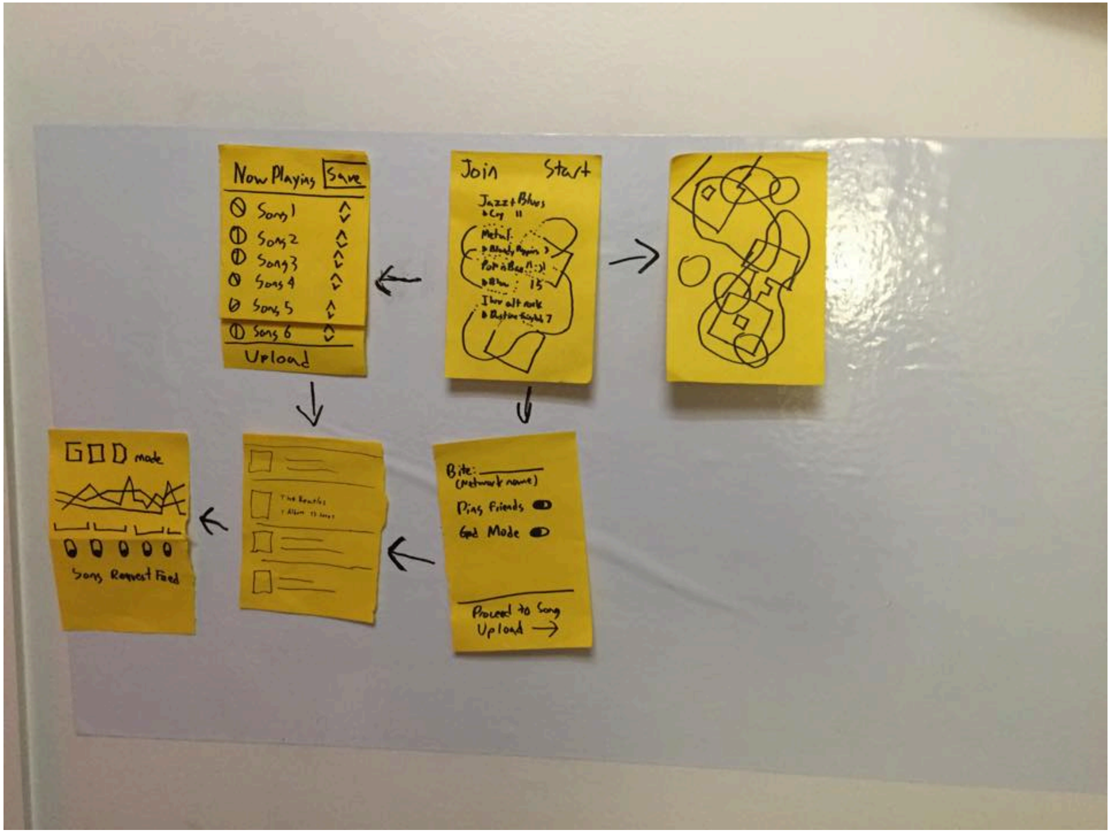
We looked at an incredibly diverse set of design ideas. The key central challenges with our design were representing the networks and voting. We wanted voting to facilitate the discovery of new music while also representing the opinions of our constituents.

Q: locally synchronized social music.



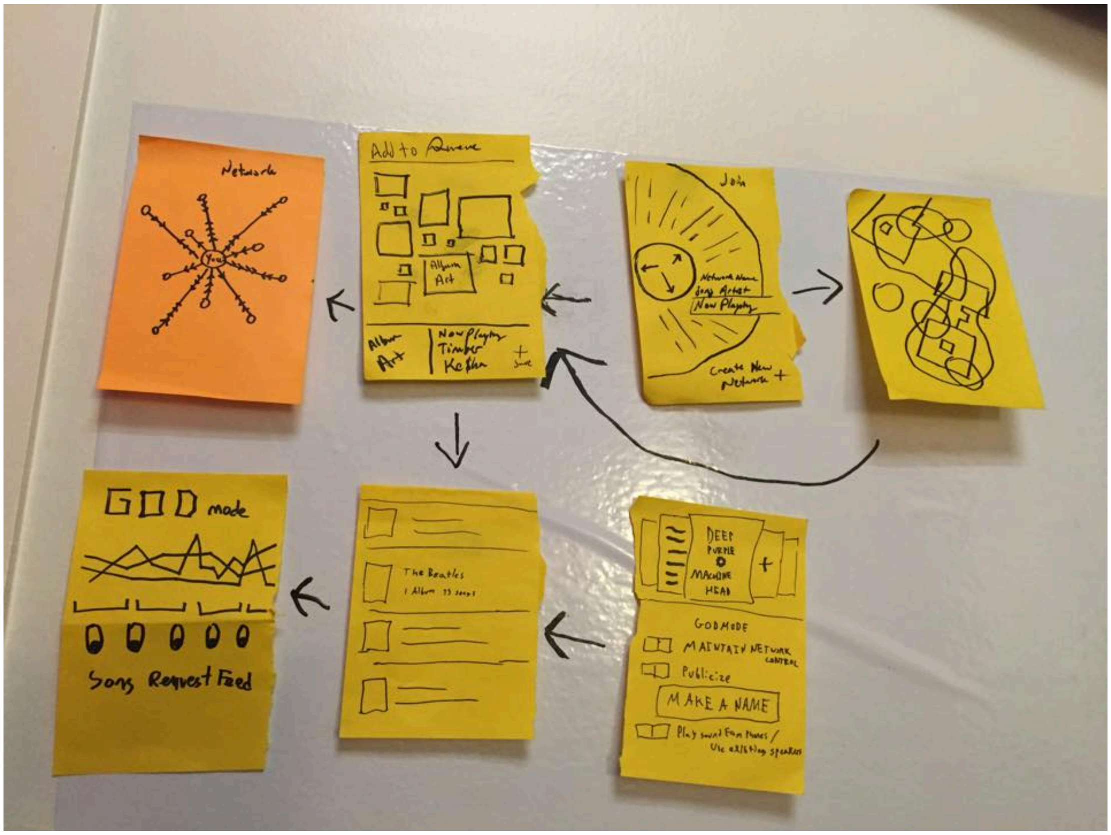
We drew up a multitude of design ideas to address the problems we outlined above and realized that two broad categories that our brainstorm sketches fell into were designs that were more familiar to us and designs that were more experimental. We thus chose to pursue these two directions independently.

Our two selected design directions:



Our first design went for an intuitive and functional direction

Q: locally synchronized social music.



Our second direction explored interface options more and gave the user a bit more control

We then thought about the pros and cons of both and realized that we really do not have enough information to select only one of these directions. We decided to prototype both design trees and use actual user testing data to make our decision.

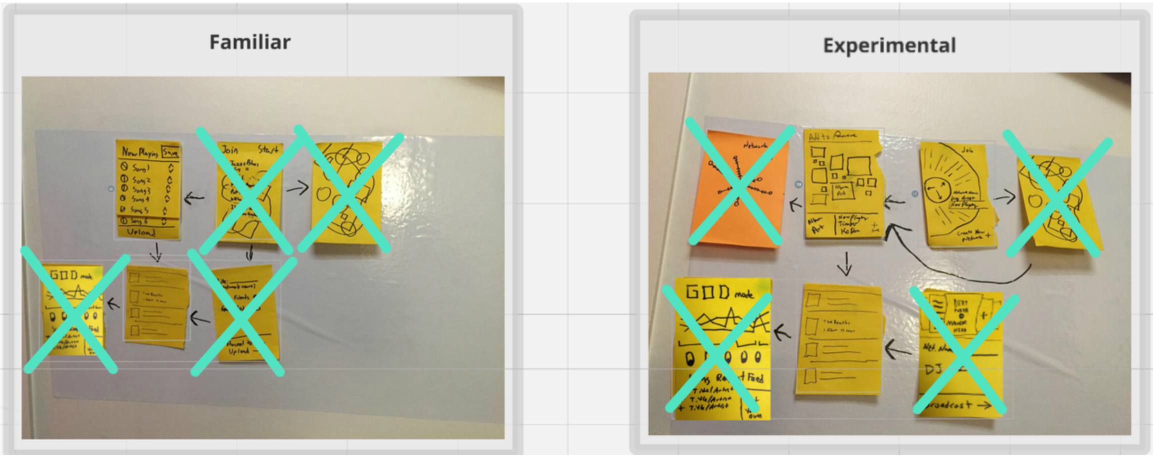
As a reminder, our three tasks were as follows:

- "Please vote on a song to show a preference"
- "Share a song of your own with the network"
- "As a DJ, select a requested song to play"

To make sure that we got a variety of user testers, we thought to our main use cases: casual listening and parties. To this end, we got a music influencer to one of our previous interviewees who also occasionally DJs small events as our first participant, referred to as subject 1. We chose him because we wanted someone who would be considered more of a power user and had a familiarity with DJing and music sharing in general. Subject 1 was interviewed in his dorm.

From testing, it became relatively obvious that both approaches had their own strengths and weaknesses, and we agreed very quickly to mate our two models to produce more viable offspring. As the map was apparently less of an asset that we had originally perceived, we agreed to ditch the functional join screen in favor of the scroll wheel that had elicited such positive feedback. The functional queue seems much simpler to implement and is much less of a barrier to usage. While the alternative is visually compelling, major barriers exist to a successful implementation that seem, without essentially training users specifically too it, insurmountable. The cost in the learning curve of implementing it could prove a significant barrier to an overall positive user experience. We will shelve the idea, but the technical difficulty of developing this view from the ground up is likely too much for our team, even if we knew how to fix the design issues.

Our other main takeaway centered around god mode. We originally designed the god mode screen as modular components that could be customized to the dj's preferences. We proposed a variety of solutions to barrier it posed to the fluidity of use. A pay wall, making it inaccessible to anyone besides dedicated users was one option. A simpler god mode was also considered. The solution that received the most support from our team was creating a computer program for god mode as it would work better with both the modular theme and a dj's typical setup for mixing (a laptop). This will eliminate much of the confusion surrounding creating a network, as it eliminates the need for a screen, and establishes djing as a separate, specialized use case that simply depends on audience members having the app, and doesn't add any complication to the typical use case.



Once we combined the experimental join screen with the intuitive queue screen and cut down on all superfluous functions (read: God mode) the app really began to come together, and it was simply a matter of iterating the design. Unfortunately, cutting out our complex task entirely meant we had to find something to replace it. We tried implementing a horizontal view for the queue screen that would allow for drag and drop rearrangement, sort of a pseudo-God mode, but we ended up abandoning it after heuristic evaluations made it abundantly clear that its functionality seemed redundant. It presented too much of a usability challenge and didn't have the reward of increased functionality, so we cut it out entirely. Once again faced without a complex task, we decided to make starring songs a centerpoint of the app, as discovering new music was so central a goal. We revamped our starred songs screen and made it so users could buy songs on itunes or even see the location where they saved the song.

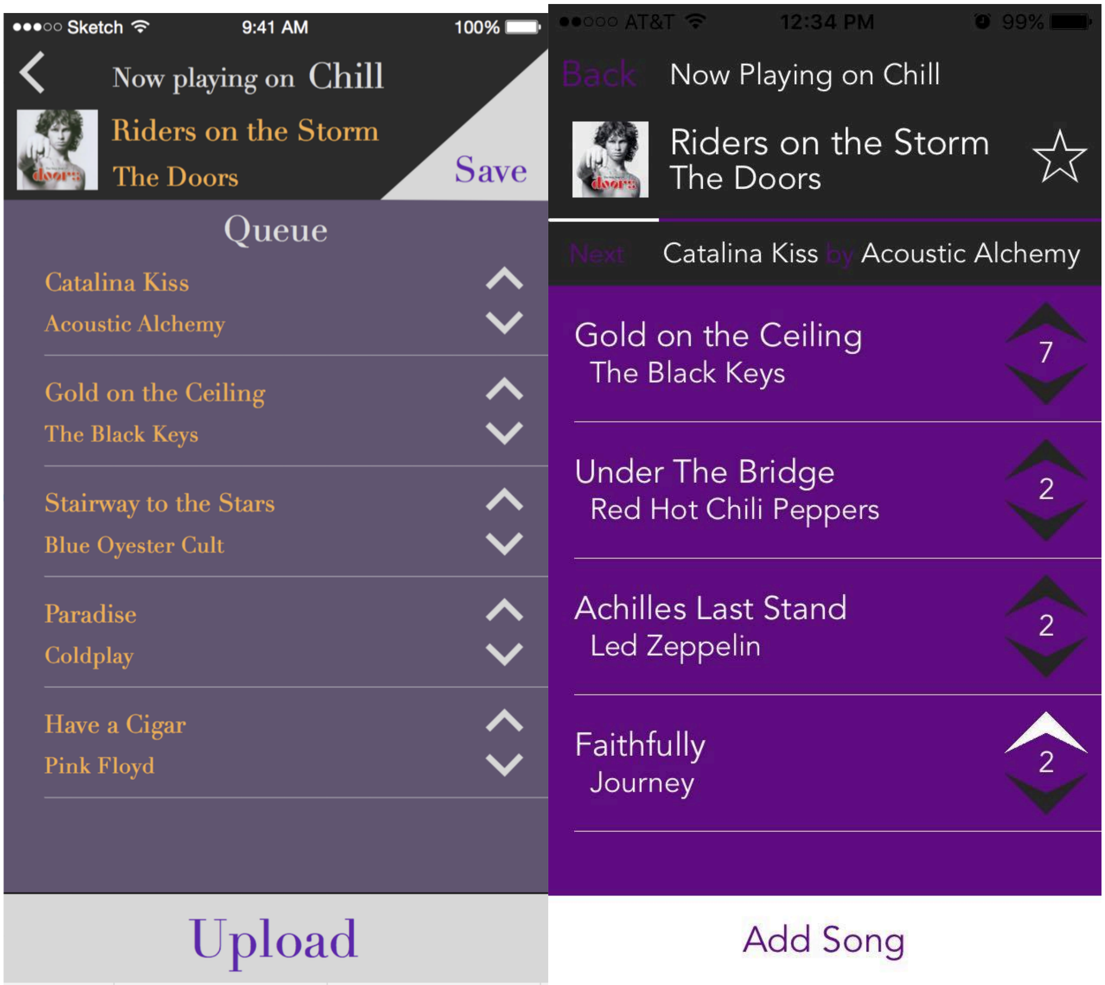
Below are the heuristic criticisms from before implementing the new and improved starred songs, sorted by screen and with planned changes outlined at the beginning of each section. Interestingly, we had almost no complaints about the queue screen, so very little changed there:

FONT & COLOR – Switched whole app to Avenir font (same as logo)

Q: locally synchronized social music.

[H2-8: Aesthetic & Minimalist Design][Severity 1][Found by A, C]
The font used throughout the app is boring and ugly. Users might stop using the app because it looks boring/under-developed. Use a less boring font that is still legible.

NOW PLAYING - (present in both HOME SCREEN & QUEUE SCREEN)
ADD SONG PROGRESS, UP NEXT & SAVING STAR, organize so it doesnt feel cramped
[H2-1: Visibility of System Status][Severity 3][Found by C]
There is never any indication of how far through a song the listener is. Users might want to know that information. Show a progress bar for the song in the Now Playing section.
[H2-4: Consistency & Standards][Severity 1][Found by C]
The artist and song title are formatted the same way on the home page and the now playing section of the network page. The user might not know which is the title and which is the artist. Make the artist smaller font or not bold (conventional)
[H2-1: Visibility of System Status][Severity 3][Found by C]
At the top of a network page it says “Now playing on Chill”. This takes up the full span on the window. If the network title were longer that 5 or 6 letters it would always be cut off. Have the name of the network on a new line.



●●●○○ Sketch 9:41 AM 100%

Now playing on Chill

Riders on the Storm
The Doors

Save

Queue

Catalina Kiss
Acoustic Alchemy

Gold on the Ceiling
The Black Keys

Stairway to the Stars
Blue Oyester Cult

Paradise
Coldplay

Have a Cigar
Pink Floyd

Upload

●●○○○ AT&T 12:34 PM 99%

Back Now Playing on Chill

Riders on the Storm
The Doors

Next Catalina Kiss by Acoustic Alchemy

Gold on the Ceiling
The Black Keys
7

Under The Bridge
Red Hot Chili Peppers
2

Achilles Last Stand
Led Zeppelin
2

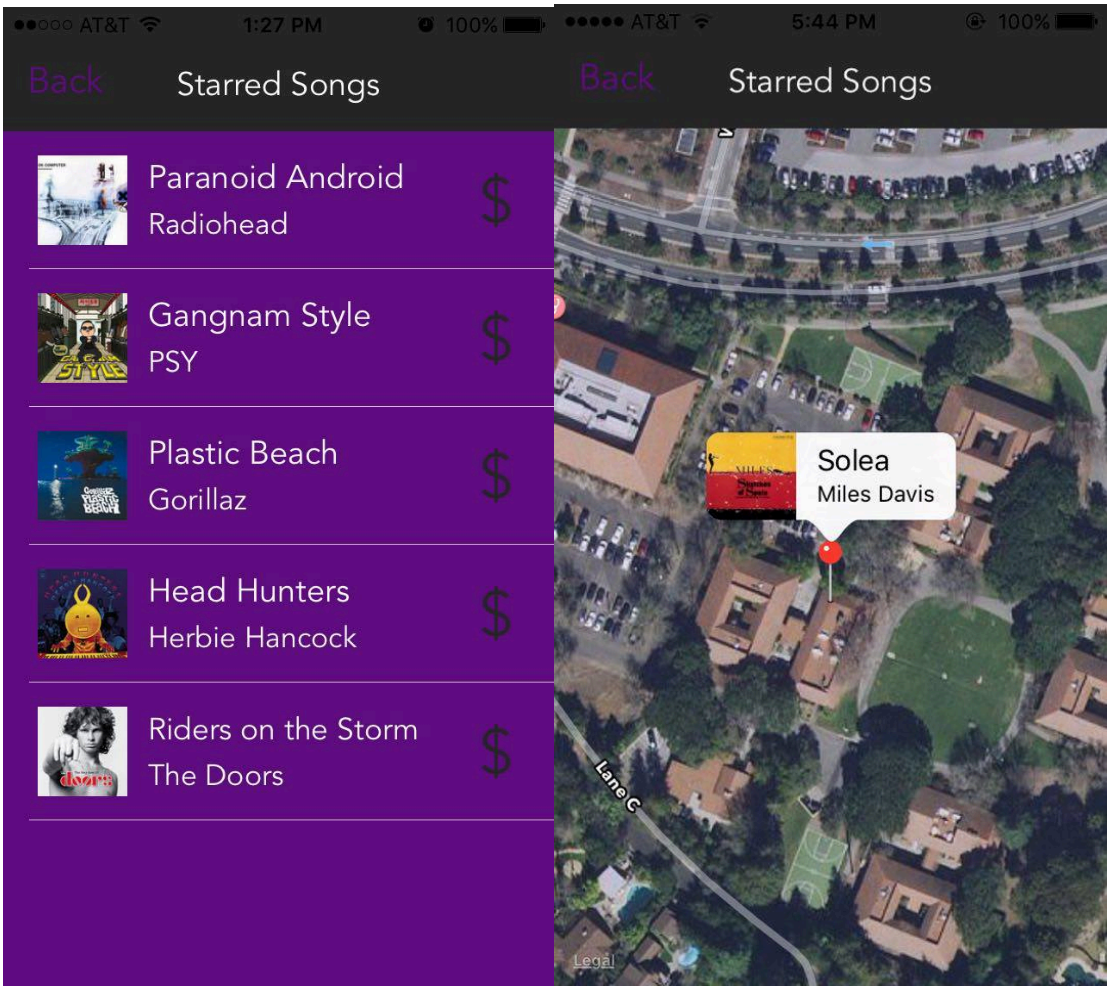
Faithfully
Journey
2

Add Song

QUEUE SCREEN- Made arrows bigger and added numbers. Also functionality-wise, votes now do stuff

Q: locally synchronized social music.

[H2-5: Error Prevention][Severity 3][Found by C]
The upvote and downvote buttons are very small and close to one another. Fingers are a lot less precise than a mouse. Users will likely click on the wrong arrow sometimes. Make them bigger and next to one another.
[H2-1: Visibility of System Status][Severity 4][Found by B, C]
Once I've upvoted/downvoted a song there is no indication of what I did/of how close the song is to being moved. Users might think that their votes don't matter. Add a vote counter to show users how close songs are to one another.



Paranoid Android
Radiohead
$

Gangnam Style
PSY
$

Plastic Beach
Gorillaz
$

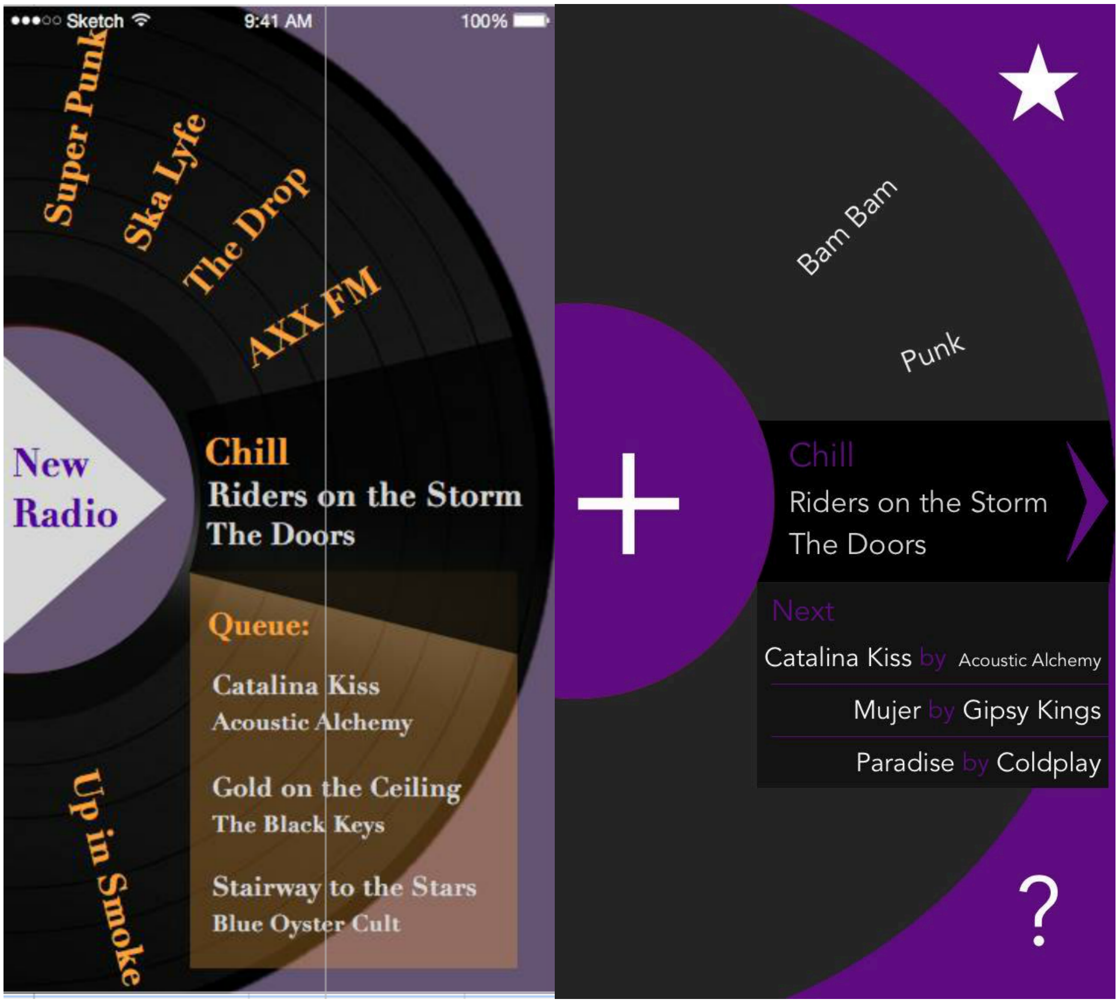
Head Hunters
Herbie Hancock
$

Riders on the Storm
The Doors
$

SAVED SONGS PAGE-Revamped the page (gave it purpose), also made the star consistent across the whole app, added map function. Original version not pictured (we couldn't find a screenshot because it was so unimportant, but it was just a tableview with song names and date when saved)
[H2-2: Match Between System & Real World][Severity 3][Found by B, C]
"Saved" vs "Starred" songs. We thought that saving the songs would actually download them which legally cannot be the case. We recommend changing the vocabulary to Starred, or removing this feature completely.
[H2-2: Match Between System & Real World][Severity 4][Found by B, C]

Q: locally synchronized social music.

It is not clear what I can do from the saved song page other than go back. The songs do not appear actionable. Make the songs looks more like cells that can be played or looked up. Also, consider if there's really a reason to have a saved songs list at all.



HOME SCREEN – consider when actually coding home screen, many of these are simply prototypical problems, but were still very helpful in determining the specifics of how the home screen would behave.

[H2-6: Recognition Rather Than Recall][Severity 3][Found by C]

When I tap on the back arrow from a network page it takes me to the page when no network is selected. If I still wanted to play that network I would have to find it again. Take me back to that network being selected.

[H2-10: Help & Documentation][Severity 3][Found by B, C]

After the prompt to "Choose a Network" it is impossible to get back to that help screen. The user might need to refresh the available options. Add a ? button in the corner.

[H2-1: Visibility of System Status][Severity 3][Found by C]

It is not clear how the Networks are ordered on the home page. The user might not know how to search for a network. Use alphabetical ordering.

First two were simply prototype limitations that were fixed in the working app, and the last one isn't a real concern because the number on networks is so limited (they have to be within Bluetooth range)

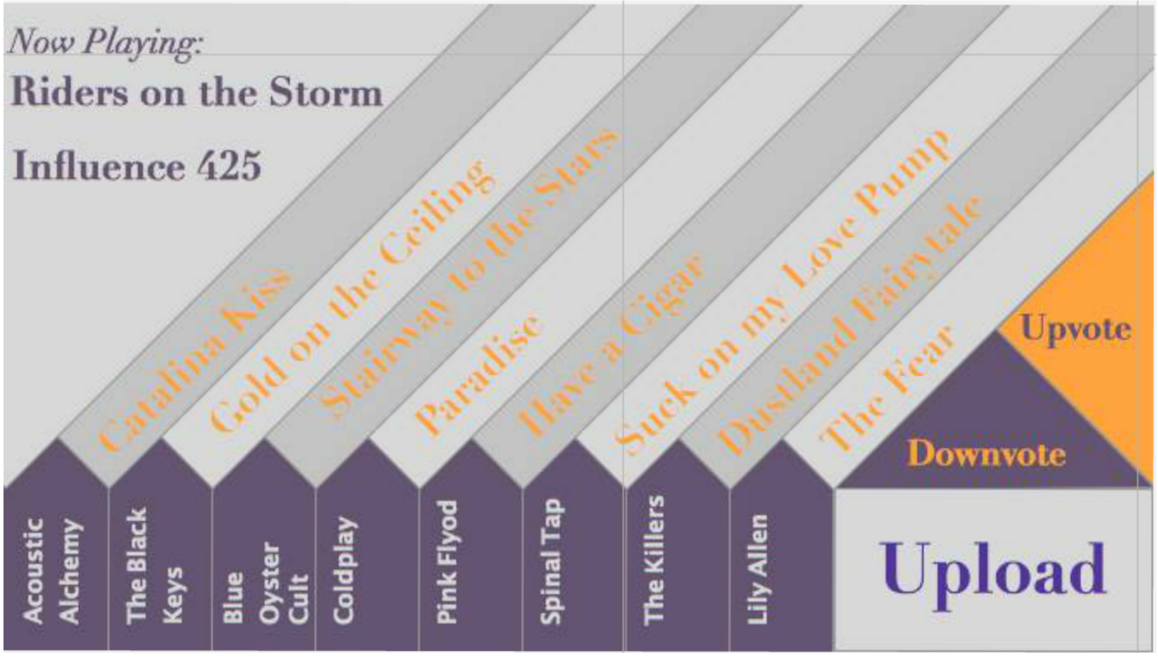
Q: locally synchronized social music.

[H2-1: Visibility of System Status][Severity 3][Found by A, C]
From the home screen there is no visual cue that music is playing, or which Network it is coming from. Users might not know that the sound is coming from the app. Add a speaker icon next to the selected network.
It seems that you cannot look at what is in other networks without stopping from listening to the currently selected network. Have a button to play a network and then allow browsing without changing the "chosen" network.
[H2-5: Error Prevention][Severity 2][Found by B, C]
It is very easy to change Networks from the home page. If a user accidentally swipes the screen to another Network they will change what they are listening to, which is not what they want to do. Users could be annoyed when they accidentally change the song. Make playing and browsing separate actions.
We had to discuss how/when to play music and settled on sound only coming out when you enter the queue screen, though we toyed with holding down to preview a network.

[H2-2: Match Between System & Real World][Severity 4][Found by B, C]
The "New Radio" button is pointing at whatever Network is selected. This implies that that network is new. It is not clear that that is an actionable button unrelated to the selected network. Change this to a round button concentric with the vinyl.
After the user enters the name for their new radio station, there is no "Enter" button which allows them to proceed. This will confuse users because it looks like there is no way to register their station with the app. Include an "Enter" button below the textbox.
We made new radio a circle that reveals a triangular cancel button when tapped. Added an arrow to continue (create new network/go to existing network's queue)

[H2-4: Consistency & Standards][Severity 3][Found by C]
On the home screen: The red "New Radio" button makes me feel like the red "Saved Songs" option is also a button that would open saved songs. Users might press on saved songs instead of rotating the vinyl. Use something other than color to distinguish saved songs from other networks.
The "Saved Songs" option is only visible when the bottom network is chosen. Users might not know where to find it once they've rotated the disk. Have saved songs be fixed and somehow indicate that it is not on the record that rotates.
The "Saved Songs" option is in the list of networks but it is not a network. Users might be confused by this. Have saved songs be fixed and somehow indicate that it is not on the record that rotates.
On the first page the yellow help section says to "Pull up the saved song list" but gives no indication of what you can do from there. Users might be confused about the functionality of that list. Change this to "View Saved Songs by Rotating Up"
In a network page I can only save the current song (or does it save the whole playlist? Unclear). Users might want to save other songs. Use a "..." button for each song to access additional functionality like save. Or use a star icon. I might want to be able to save a song other than the song currently playing, so adding a star icon next to each song in the queue would allow me to do this.

Q: locally synchronized social music.



HORIZONTAL PAGE- CUT, (vacuously fixing all these heuristic problems)
[H2-6: Recognition Rather Than Recall][Severity 3][Found by B, C]
On a network page there is no indication that going horizontal will allow the user more control and options. Have a little line of text showing that rotating the phone gives more voting functionality. Also consider that some users may have their phones locked to not allow rotation.

[H2-2: Match between system and the real world][Severity 1][Found by B] On the horizontal queue screen, the songs appear in a diagonal fashion, which does not seem to be laid out in a natural and easy-to-read way. It feels a little confusing and disorienting to me. I would make this look cleaner. It feels more natural to order things top to bottom than right to left, so I would suggest scrapping the horizontal and keeping the screen vertical so it's clearer to the user when reordering songs.

[H2-4: Consistency & Standards][Severity 2][Found by C]
When horizontal the current song is in the list of songs that can be reordered and it is in the same color scheme. It looks like you could reorder it too. Users might try to drag it around. Remove that from the reorderable list.

[H2-8: Aesthetic & Minimalist Design][Severity 3][Found by A, C]
On the horizontal screen, what does "420" mean? If it is the influence score, is 420 high or low? There is no benchmark. Users will be confused by this large number. Get rid of this feature or it will get too complicated.

[H2-6: Recognition Rather Than Recall][Severity 3][Found by C]
When horizontal, after being told by the gray box that the bottom left is a notification feed, there is no labeling of what that is other than when things pop up. Users might not know what the pop ups are. Add a title, "notifications:".

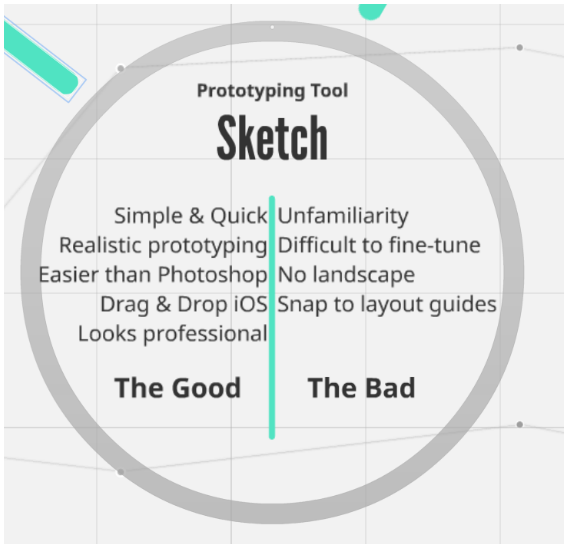
[H2-6: Recognition Rather Than Recall][Severity 4][Found by C]

Q: locally synchronized social music.

When horizontal, after being told by the gray box goes away it is not intuitive that the songs can be reordered. Users might be confused. Show a click and drag icon under the list.

[H2-1: Visibility of System Status][Severity 3][Found by C]
The notification feed tells me that a song was upvoted. Why does that matter to me as another user who didn't necessarily add or interact with that song in any way. It would seem that the entire point of the app is up and down voting, that is a lot of notifications for everyone holding their phones horizontally. Users will be annoyed by these if they cannot do anything with that information. Remove the notification feed.

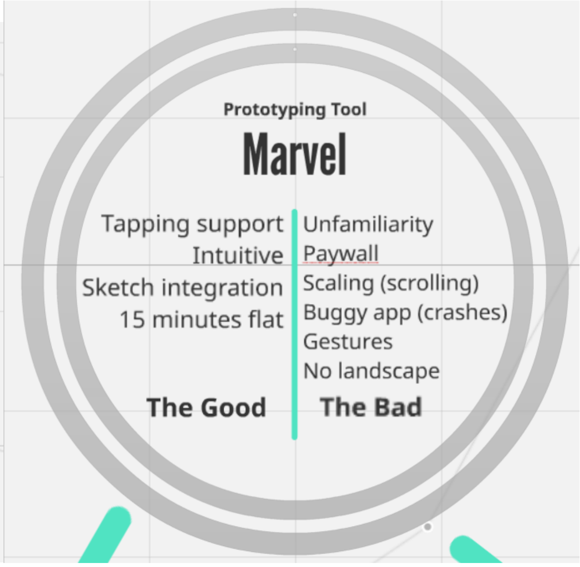
For the medium-fi prototype, we used sketch and marvel as our main tools, both of which had varying levels of success, which I've outlined below.



Prototyping Tool

Sketch

The Good	The Bad
Simple & Quick	Unfamiliarity
Realistic prototyping	Difficult to fine-tune
Easier than Photoshop	No landscape
Drag & Drop iOS	Snap to layout guides
Looks professional	



Prototyping Tool

Marvel

The Good	The Bad
Tapping support	Unfamiliarity
Intuitive	Paywall
Sketch integration	Scaling (scrolling)
15 minutes flat	Buggy app (crashes)
	Gestures
	No landscape

For this final prototype, we did everything in xcode and produced something that does everything we want it to, besides networking of any kind. Xcode is an extremely clean and easy to use program, but the documentation for Swift is currently not as robust as many other languages, including objective C, so finding examples of others doing what we were trying to get working was oftentimes impossible. We had to deal with examples in objective c (and translate them over into swift's syntax) so much that at this point, we almost feel as though we could code fairly proficiently in objective c.

We do have a working mesh network, but it exists separately right now, as integrating it with the prototype proved to be too much for us to get done during dead week. Fortunately, the prototype seems to do everything else, so the only wizard of oz technique is really in faking the existence of a network (by creating one locally that doesn't broadcast). The only notable hard coding is of the initial networks that appear when the app is launched. We managed to get not only all the core functionality, we even got many things functional that are not necessary (playback with the screen off). Of course the prototype doesn't work perfectly, as is apparent on the home screen (which glitches out sometimes) but all things considered, they work extremely well, especially considering how logistically difficult they were to get working. The most apparent disconnect with the prototypes behavior and our ideal app's behavior is upon joining a network. Normally, we would anticipate tuning in to skip to the place in the song that everyone is listening at, but right now, as the networks are entirely on the phone, it just starts at the beginning of the first song. There didn't seem to be a good reason to try and fake this.

In terms of future functionality, other than cleaning up bugs, all we really want to do is integrate the network into our app so everything actually works. There is also a very strong possibility that we will revisit some of the ideas that were cut in order to make this happen (god mode, rave). It is unclear whether they would be a part of the app or exist as separate entities, but they would definitely have a strong level of interplay between each other.

We are definitely very excited about the prospects of our app. We originally set out to make something that we wanted to use that didn't exist yet, and I think we succeeded magnificently in that goal. Not only is the baseline functionality of playing the same thing from multiple phones generally something that would be useful in an enormous number of contexts, as well as the ability of a crowd to decide jointly what it will listen to, but the interface we designed is something that we are incredibly proud of. The experimentation that we allowed for early on in terms of throwing out alternative UI designs paid off more than is quantifiable. Our join screen, originally seen as an afterthought overlaid on a map, came to become our most distinctive screen, with a tactile interaction and visual appeal that testers fawned over. It presented the most technical difficulty out of any other UI choice, but now that it works, its hard to imagine a more satisfying way to browse networks around you.