Pradyumna Mukunda

Fall 2019 – Sprint 2 Individual Report

CS 8803 Mobile Application and Services
Georgia Tech, Atlanta, Georgia, USA

Team Name:
Beat Harmony
Project Name:
Beat Harmony
Team Members:

- Ankit Verma (averma46@gatech.edu)
- Rishma Mendhekar rmendhekar3@gatech.edu
- Christian Graham cgraham47@gatech.edu
- Justin Higgins jhiggins@gatech.edu
- Pradyumna Mukunda pmukunda3@gatech.edu

1 Project Overview

Following section gives description of Problem the team is trying to solve, Domain research and user interview results, approaches considered and details of chosen approach

1.1 Problem:

Novelty seeking music-heads cannot find the fresh new music they need because:

- *Music-heads are stuck in echo chambers as a result of the naive recommendation algorithms of Youtube and Spotify.*
- *Music-heads are not connected to curators who share in their unique tastes.*

1.2 Domain Research, User Interviews

Customer discovery interview was conducted by Ankit Verma who is the Team lead for this project as part of First Interview assessment [1]. Following insights could be derived out of the work:

- Major platforms used for finding music are – Spotify, Soundcloud, Reddit, Youtube, Friends or Word-of-mouth, Social media (Facebook, Twitter)
- Participants felt the current algorithmic solution suggest something based on their previous choice and can be repetitive. Music-heads worry that they may miss out on other gems as it could be in the bottom of list.
- Participants want music discovery platforms to suggest new music based on the choices of people with similar taste or on their own established tastes
- The majority of participants value ease and convenience for music discovery. Participants feel like they have to put in too much effort to find new music
- Person who seeks novelty in music and digs for obscure music is more likely to be dissatisfied with current solutions than someone who is fine with listening to what they already are familiar with.
- Many interviewees seemed to imply that getting new music recommendations from their friends was their most reliable method of getting to hear new music that they really like.

- There is no one size fits all when it comes to music discovery as each person has different preferences based on demographics

There is a significant trade-off between convenience and high-quality discovery, and yet users opt for convenience in every case. It seems like the solutions that the interviewees use all currently strike a balance between speed, convenience, and finding music that is “just good enough for now”. What is missing for most individuals is a style of discovery that closely resembles word-of-mouth recommendations from friends, but users are not willing to sacrifice convenience to get there. There seems to not yet be a solution that can find a balance between convenience and quality organic, word-of-mouth, style discovery, and our team wants to explore a potential viable product in this area.

1.3 Solution Approach – Sprint1

Each team member came up with three separate approaches to solve the problem. All suggested approaches were rated based on following criteria

- Ease of Implementation
- Customer Convenience
- Taking recommendations from Friends
- Having connectivity to platforms

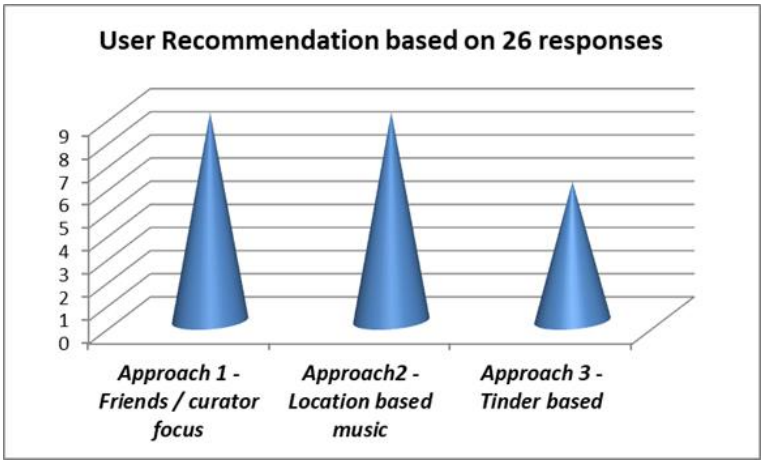
Three approaches were selected and presented as part of Sprint1 based on the above criteria**. Approach 1 below is very close to the approach I suggested by me as part of individual effort**. Mood based feature in my approach was not accepted by team as it involved predicting user mood. My idea was to predict user mood based on time of the day, user location setting, user current activity such as driving / workout / dining, etc. As it can be a hard task, team did not include this in the selected approach.

Approach presented in Sprint1 are as below:

1. Social-Media platform with each user’s ‘trusted’ curators as the focus
    a) user type A acts as a music ‘seeker’ and sees recommendations from the people they trust most
    b) user type B acts as a music curator and posts recommendations for others
2. Location based music curation
    a) Uses user’s geographic data to recommend music.
    b) Users may submit song recommendations for current location and have other users vote.
3. Tinder for music recommendations

1.4 Feedback from Sprint1 presentation and Additional interviews

This section is contributed by me. Post the Sprint1 presentation, we received feedback and also many students in the class gave their preference for the approach. Graph above summarizes the ratings received for each of the three approaches.



As can be seen from the chart above, 9 participants preferred Approach 1 which has Friends / Curator focus and Approach 2 (Location based music) and 6 participants preferred Approach 3 which was Tinder like functionality for music. In addition to the above, separate limited interviews were held to get user preference for each of the approach.

(This section is contributed by me) Following is the summary of questions / Feedback received as part of Sprint1 Peer review feedback and our reply**.**

User Feedback / Questions	Solution
Think about users who dont necessarily want to discover a new genre of music but rather just stay within their niche of music they already know	- Approach includes following known friends who have similar music taste - Also has Tinder like functionality to match users with similar tast
- Team should clarify how their solutions will differentiate amongst competitors that already have music recommendation algorithms." - I still don't see how this problem is not already solved by existing problems. If the focus is creating a social media platform based on music--this already exists and you can add friends and 'trusted' users there already.	Current solutions are based on - Recommendations based on listening history / search history / playlists containing songs you like - Reviews by unknown/trustless members of the community Our approach is based on - Word to mouth recommendations by trusted friends - Suggestions by Music curators who are connoiseurs in the field
An app that curates music without playing it is not	Added integration to Spotify

Based on the responses, team decided to go ahead with developing full blown prototype based on Approach 1 with few changes

Social-Media platform with each user's 'trusted' curators as the focus

- Tinder like function to match and connect people with similar music tast
- Integrate Spotify so that user can play music

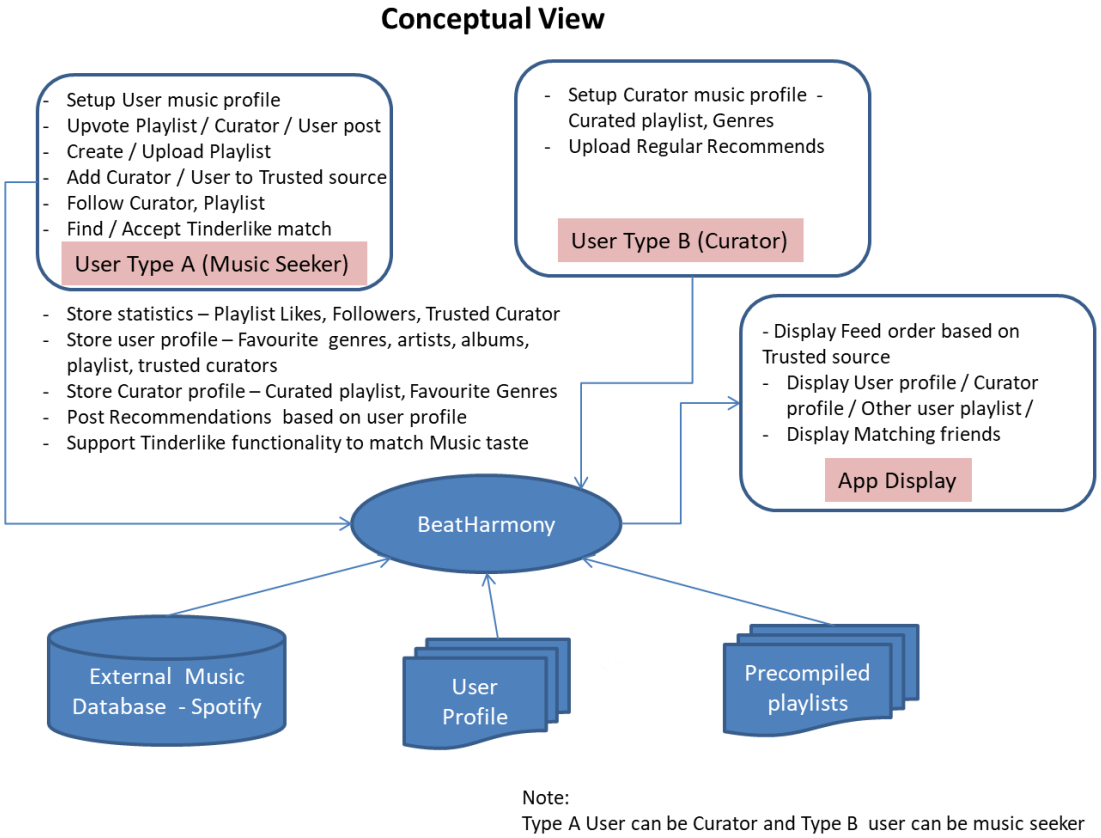
1.5 Selected Approach Details (Team decision with my participation)

Selected Approach is a feed-focused social media style app based on sharing playlists. A user can act as a seeker or curator at any time. Users can mark other users as 'trusted' music sources and increase another user's trust rating by upvoting their posts.

Features

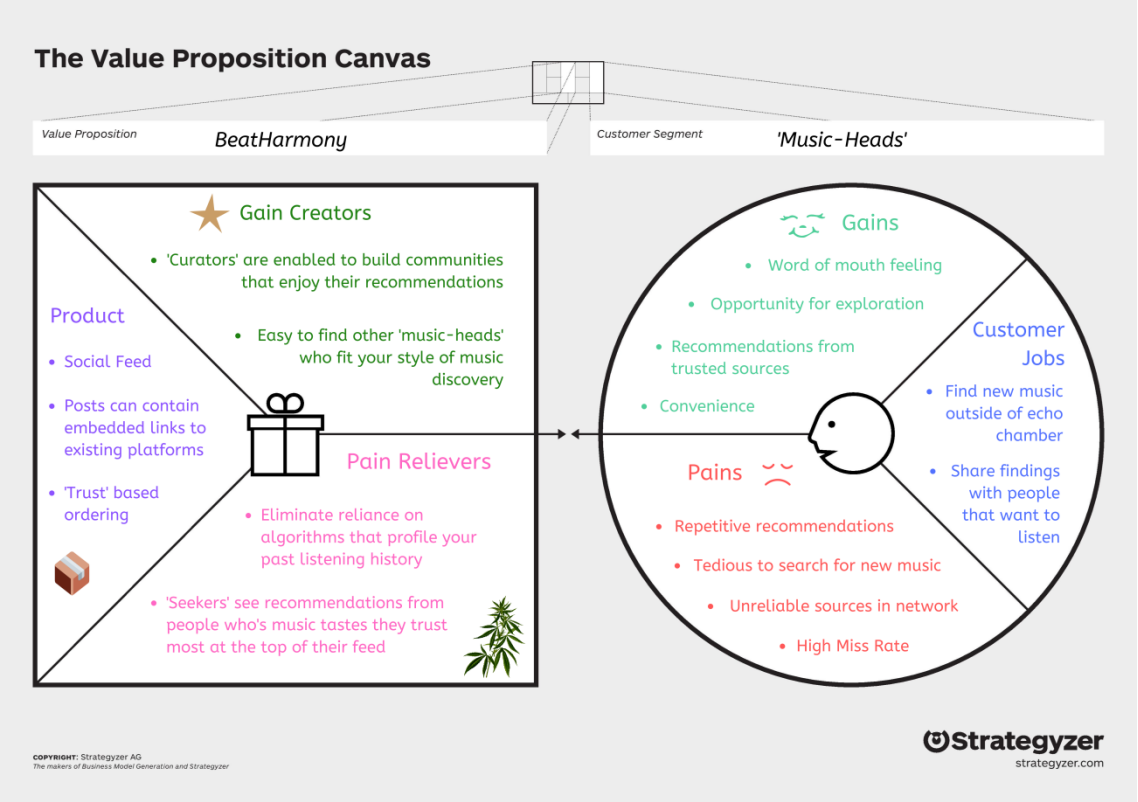
- Social feeds with scroll
- Trusted source recommends higher in the feed order
- Trusted source identification method
    - Explicit follow of curator
    - marked trusted source
    - number of posts listened or liked by user
- Spotify Integration
- Setup User / Curator music profile
- Up-vote Playlist / Curator / User post
- Tinder for matching users with same music taste

Figure below shows the conceptual view of the selected approach. (**Conceptual view is created by me).** It identifies the activities of Music seeker and Curator, Functions of the app and displays. It will have integration with Spotify database to play music. Local stores will include user profile and precompiled list storage.



Note:
Type A User can be Curator and Type B user can be music seeker

Figure below shows the Value proposition Canvas (Team design)



Value seen for customer are

- Word of mouth feeling
- Opportunity to explore new music
- Recommendation from trusted sources
- Convenience

Business value

- Music curators are enabled to build community that enjoy their recommendations
- Easy to find music heads who fits your style of music

2 Use cases

List of Use cases covered in the screen flows are shown below**. I contributed partially to identify the use cases**

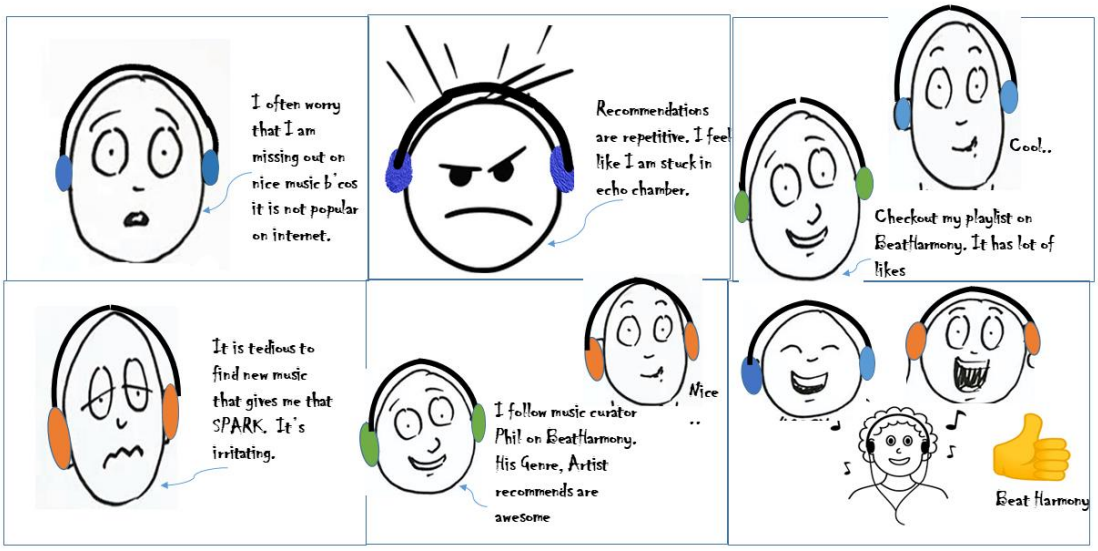
Music Seeker use cases

- Setup User music profile
- Up-vote Playlist / Curator / User post
- Create / Upload Playlist
- Add Curator / User to Trusted source
- Follow Curator, Playlist
- Find / Accept Tinderlike match

Music Curator use cases

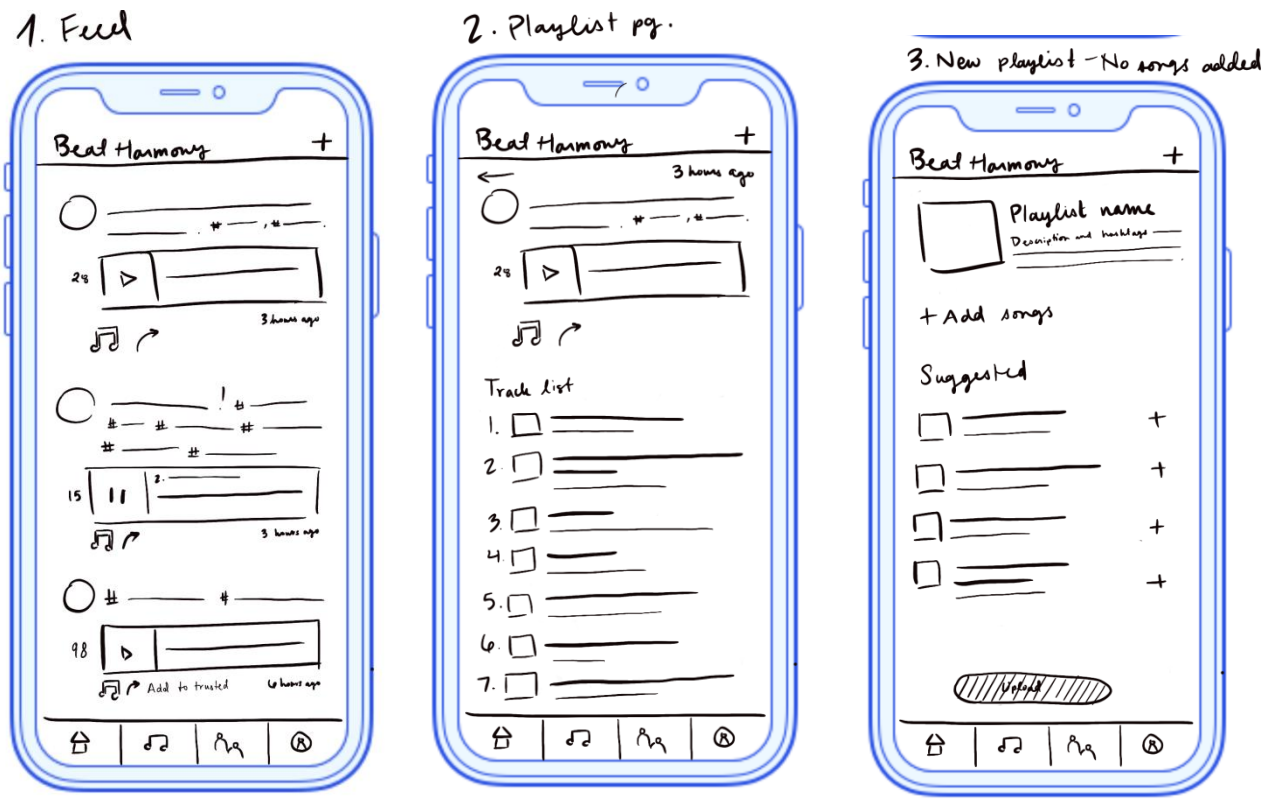
- Setup Curator music profile - Curated playlist, Genres
- Upload Regular Recommends

2.1 Storyboard (**This storyboard is created by me**)



2.2 Mockup Screens (Team design)

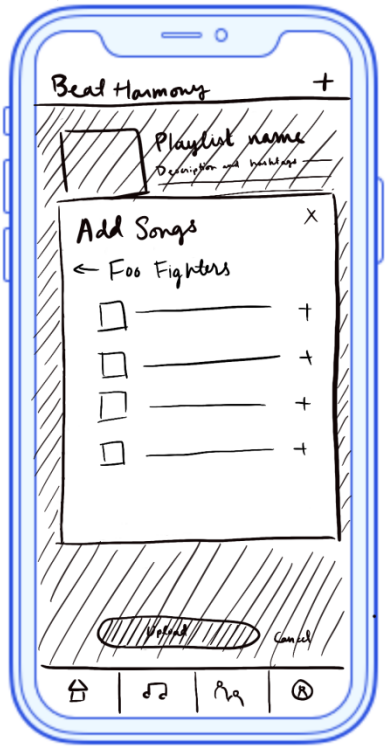
Below are the preliminary mockup screen designs (Team design)



4. Add Songs

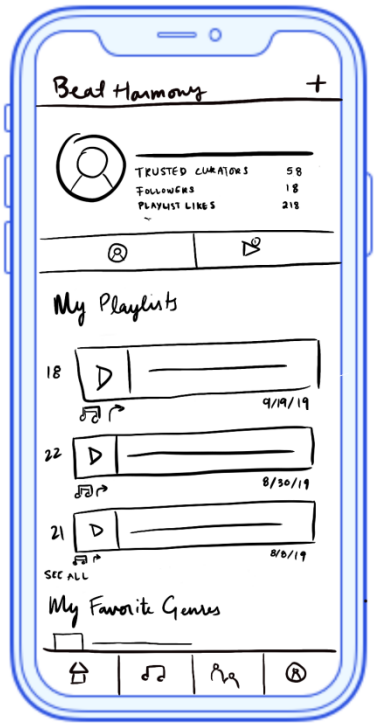
5. Search for Songs

6. View artist

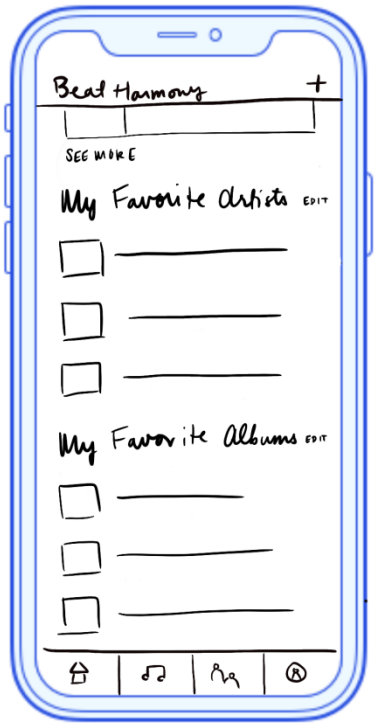


7. Populated Playlist

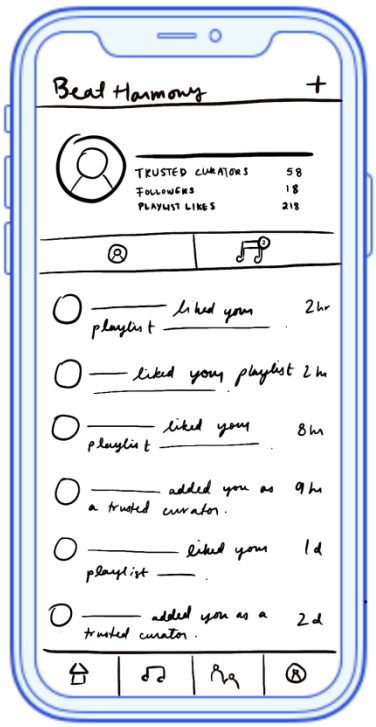
8. Own Profile - Myself



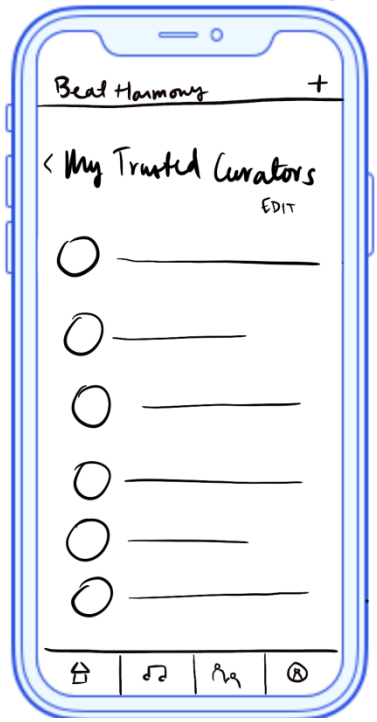
8a. Own Profile - Myself



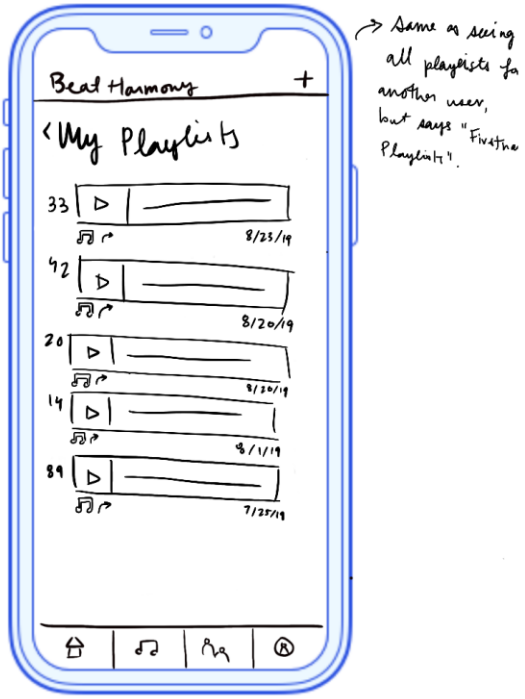
9. Own Profile - Notifications



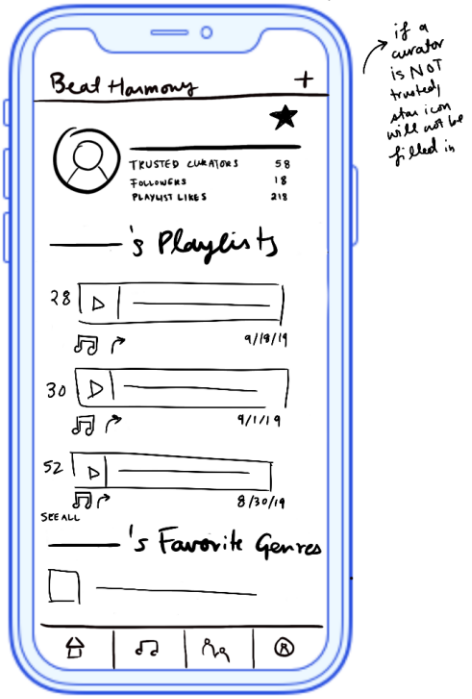
10. Own Profile - Trusted Curat



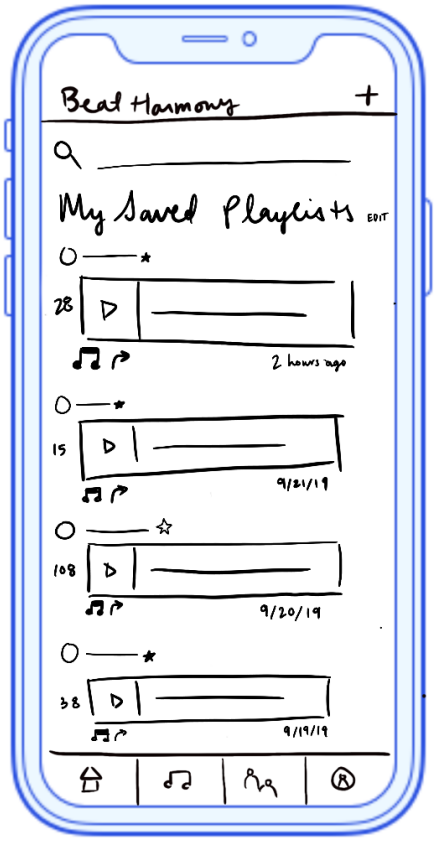
11. My/others Playlists



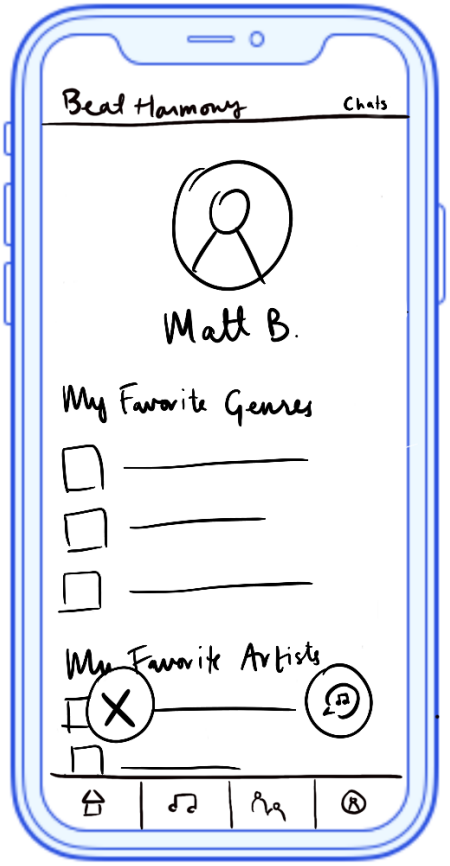
12. Trusted Curator's Profile



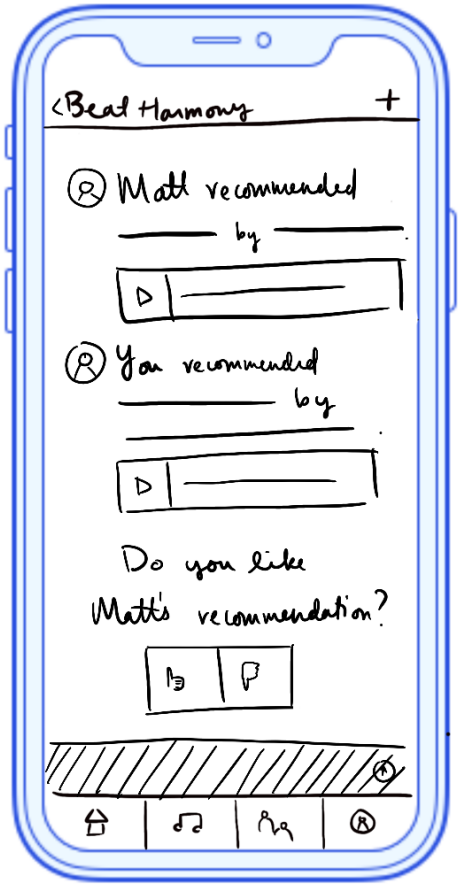
13. My Saved Playlists



14. Match w/ users

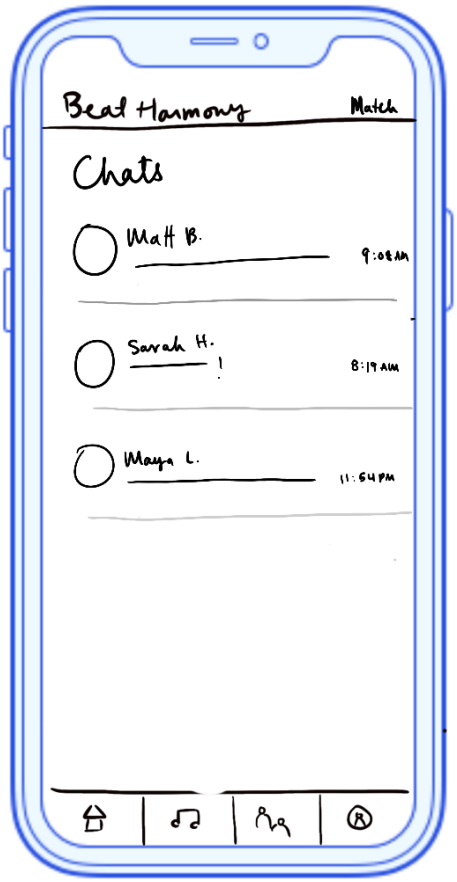


15. Send recs.



will be activated if both like recs.

16. All Chats



3 Learning Prototype (Team design)

- Learning Prototype Definition
    - A tool in the design and customer discovery process focused on testing ideas, values, features, costs, etc.
    - Influence the MVP, **Product Design**, and execution plan through validated learning.
    - Learning Prototype Data
        - Collect & Analyze
        - Develop metrics
        - Develop content and secret sauce
- Goals of this learning prototype
    - Collect customer data referencing our overall product design.
    - Collect customer data for a future interactive learning prototype.
    - Narrow in on a UI design.
    - Narrow in on a color palette or aesthetic style.
- Learning Prototype
    - For this learning prototype we presented volunteers with a series of questions referencing specific color palettes and UI designs.
    - Learning prototypes can reference a variety of different aspects regarding your product. For example they can reference the look and feel, interactivity, responsiveness, and/or your products feasibility. For this learning prototype we focussed on honing in our look and feel.
    - The UI Designs (w/ associated names), Color Palettes (w/ associated names), and learning prototype results are listed below.

- Learning Prototype Questions:
    - Q1: Disregard what the UI is referring to. Which of these **UI designs** do you feel better suits a social media platform?
    - Q2: Disregard what the UI is referring to. Which of these **UI designs** do you feel better suits a music discovery platform?
    - Q3: Which of these **color palettes** do you think better suits a music discovery platform?
    - Q4: Disregard what the UI is referring to. Please rank this **UI design** on a scale from 1-10 based on aesthetic appeal. With 10 being very appealing and 1 being completely unappealing.
    - Q5: Please rank this **color palette** on a scale from 1-10 based on aesthetic appeal. With 10 being very appealing and 1 being completely unappealing.

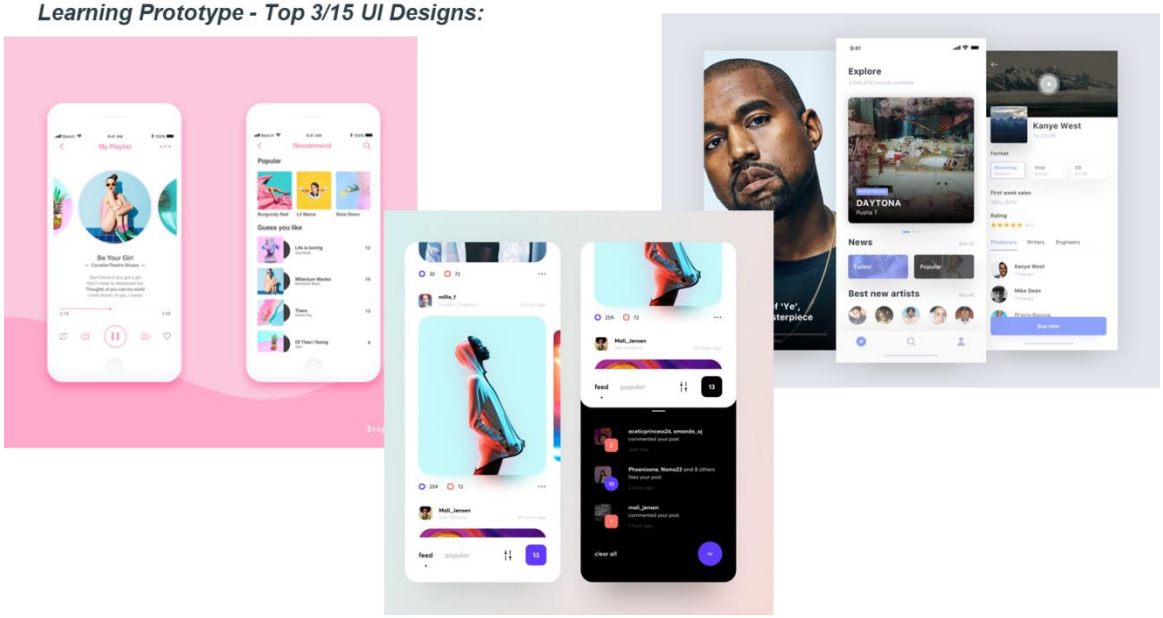
Number of people surveyed: 28. Results of the user response is shown below..

UI Design Label	Q1 Rank (__/15): UI for	Q2 Rank (__/15): UI for	Q4 Rank (__/15): aesthetic
UI Design #1	12	10	4
UI Design #2	6	7	7
UI Design #3	9	9	13
UI Design #4	10	3	1
UI Design #5	14	11	14
UI Design #6	15	15	3
UI Design #7	1	4	2
UI Design #8	7	2	8
UI Design #9	3	12	12
UI Design #10	8	13	15
UI Design #11	2	1	11
UI Design #12	11	5	10
UI Design #13	5	14	9
UI Design #14	13	6	6
UI Design #15	4	8	5

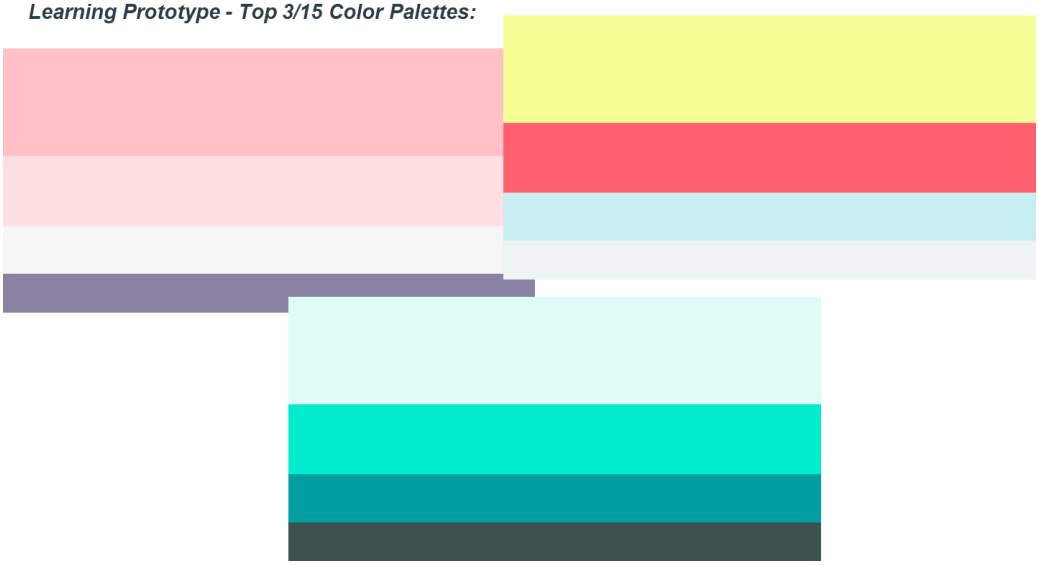
Color Palette Label	Q3 Rank (__/15): color	Q5 Rank (__/15): aesthetic
Color Palette #1	10	4
Color Palette #2	3	7
Color Palette #3	9	8
Color Palette #4	6	1
Color Palette #5	1	3
Color Palette #6	8	11
Color Palette #7	2	2
Color Palette #8	13	15
Color Palette #9	11	9
Color Palette #10	5	10
Color Palette #11	12	14
Color Palette #12	15	12
Color Palette #13	14	13
Color Palette #14	4	5
Color Palette #15	7	6

Top 3 UI designs and color palette selected from above survey is listed below

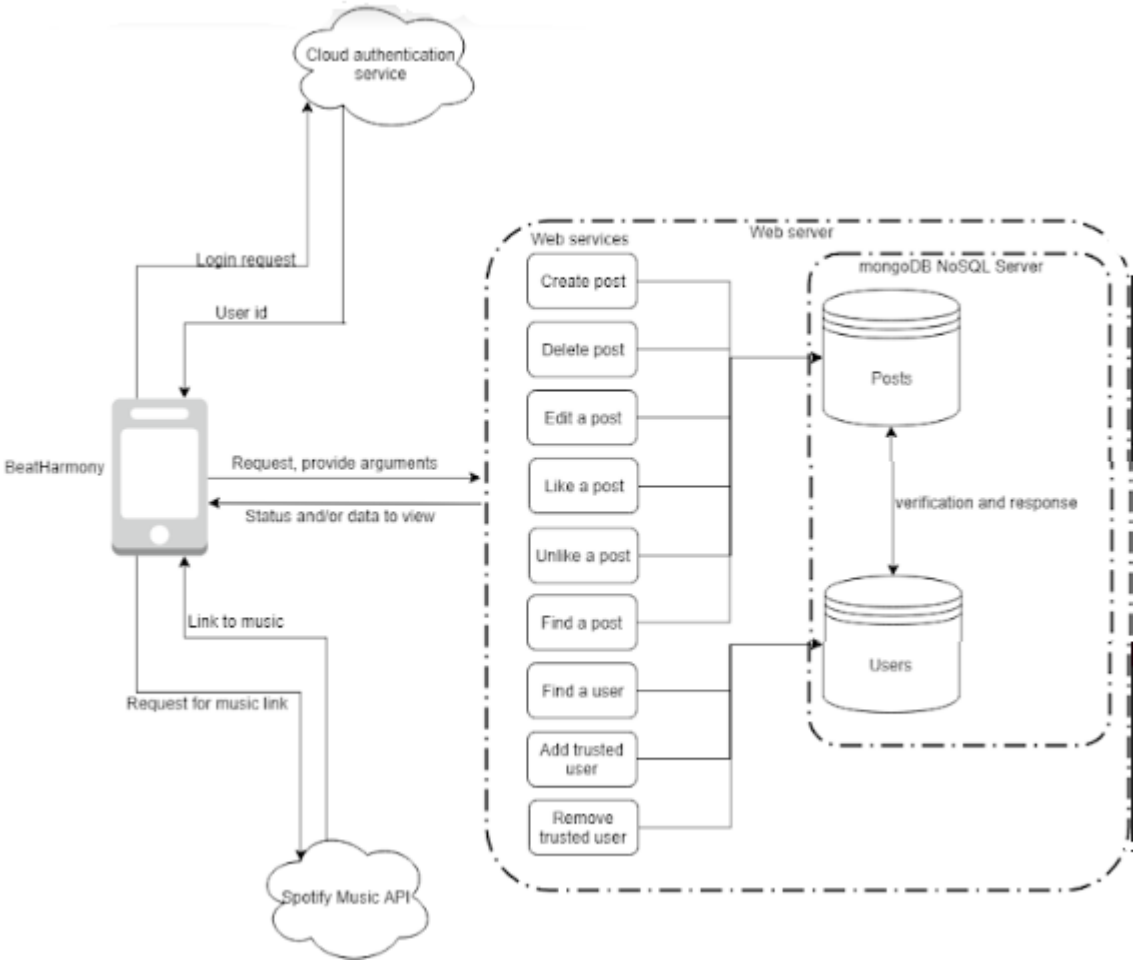
Learning Prototype - Top 3/15 UI Designs:



Learning Prototype - Top 3/15 Color Palettes:



4 Initial Project Architecture (Team design)



Above diagram shows the initial project architecture considered for the application.

On the frontend side, we will have the mobile application which fetches and posts data to the backend server. The user can view and edit posts, add trusted users etc. The app also uses the Spotify API to provide links for music. Users are verified through an external authentication service

On the backend side, we use a MongoDB server. We will be using two databases, one for Posts and another for Users. The operations on the Posts database include creating, deleting, editing, liking, unliking and searching for posts. The operations on the Users database include finding users, adding trusted users, removing trusted users etc.

The above diagram also shows the dataflows needed between front end, backend. Server and music database.

5 Data Stores

Collections we need:

- Posts
    - Post id (database-created id)
    - Posted by (key to user)
    - Post text (string)
    - Liked by (list of user keys who like the post)
- Users
    - User id (database-created id)
    - Username
    - Name (actual name in words)
    - Email
    - Trusted users
    - Interest profile (list of genres and values from 0-1, 0 being no interest, 1 being full interest)

Possible HTTP request frameworks:

GET

app/users - returns all relevant information about a user, basic searching, only accessible through sub-requests

- Sub-requests:
    - app/users/id/(user id)
    - app/users/username/(username)

app/posts/(post id) - returns post id, posted by user, post text, list of users who have liked post

- Sub-requests:
    - app/posts/(post id)/by - returns who created the post
    - app/posts/(post id)/text - returns post text
    - app/posts/(post id)/likes - returns list of users (user ids only) who have liked said post

POST

app/newpost - creates a new post

- Arguments:
    - User - user who is creating the post
    - Post text - text inside post

PUT

app/likepost - adds a like to a post

- Arguments:
    - Post id - post that is being liked
    - User id - user that is liking a post

app/unlikepost - removes a like from a post

- Arguments:
    - Post id - post that is being unliked
    - User id - user that is unliking a post, must exist in the liked list of the post

app/editpost - edits text of post

- Arguments:
    - Post id - post that is being edited

    - User id - user that is editing a post, must match user who created post
    - Text - new text to display in post

app/user/(user id)/addtrusted - adds new trusted user to user's trusted list

- Arguments:
    - User id - user that is being added to list, must not be present in list

app/user/(user id)/removetrusted - removes trusted user from user's trusted list

- Arguments:
    - User id - user that is being removed from list, must be present in list

DELETE

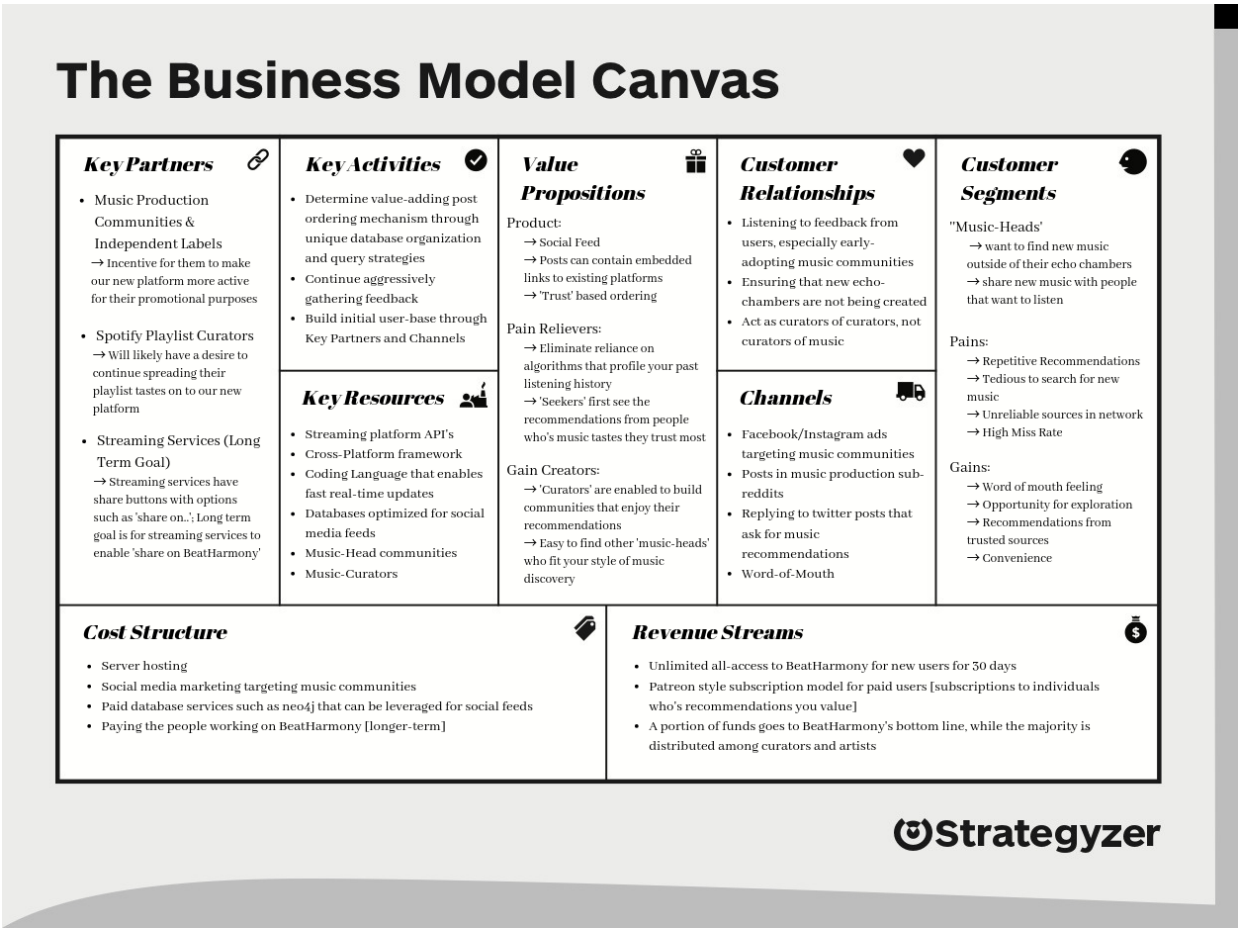
app/deletepost/(post id) - delete an existing post

- Arguments:
    - Post id - post that is being deleted
    - User id - user that is requesting deletion of a post, must match user who created post

6 High-Level Team Task Breakdown and Project Plan (Collective Team effort)

- Important decisions to be taken:
    - Decide on a hosting service for the web server
    - Decide on a platform for the web server
    - Decide on authentication
    - Decide what music API we want to hit (for now)
    - Finalize the decision on a mobile development platform
    - Decide what web server we want to use (Tomcat, Glassfish, etc.)
    - Decide if we want to use a noSQL option (mongoDB preferred)
- Big tasks for sprint 3 (number of people necessary in parentheses):
    - Set up and configure server architecture (best if one person does this)
        - Set up physical server
        - Set up web server environment
        - Set up backing database environment
    - Set up and configure dev environment for mobile (1)
    - Develop mobile UI on InVision (2-3)
        - Screens
            - Login
            - Home page with feed
            - Profile and Playlist page
        - Test UI with users
    - Set up authentication (1-2)
    - Develop initial web services (2-3)
        - Develop dummy web services to test
        - Set up database tables
        - Expand web services to interact with database
        - Interest algorithm
        - Feed generation algorithm

7 Initial Business Model Canvas



Value seen for customer are

- Word of mouth feeling
- Opportunity to explore new music
- Recommendation from trusted sources
- Convenience

Business value

- Music curators are enabled to build community that enjoy their recommendations
- Easy to find music heads who fits your style of music

References

1. CS 4261 First Interview Assessment by Ankit Verma
2. https://medium.com/the-sound-of-ai/spotifys-discover-weekly-explained-breaking-from-your-music-bubble-or-maybe-not-b506da144123
3. https://www.startups.com/community/questions/1669/where-are-the-big-opportunities-in-music-curation-and-why-are-companies-like
4. http://www.soundandmusic.org/resources/artists-toolkit/presenting-your-work/curating
5. https://observer.com/2016/12/the-business-of-music-curation-in-the-garden-of-infinite-choice/