Songscope Needs Analysis

0. Your Team Members

Tim Rolshud, Yusuf Sallam, Krish Arora, Matt Lerman

1. Primary Client

Mr. Mikhail Rolshud, Senior Software Engineer at Millennium Management LLC, in New York City. Mr. Rolshud **would not like ownership of the Songscope** project nor would he like **any of the responsibilities that come with being an owner for a social media site of this sort**. He has simply agreed to offer his years of experience as a software developer and project manager to help us develop a project with maintainable, professional level software, with longevity and maintainability being top priorities. In addition to guiding us from a software standpoint, Mr. Rolshud has agreed to act as a beta tester for our project, using the application in its early stages and providing us with suggestions as to how we can improve the project in the future from a user standpoint.

2. Other identified stakeholders

- Mrs. Natalia Pevneva, Mr. Vadim Koretskey, and Mrs. Margarita Koretskaya. These are also senior software engineers in the greater NYC area who have years of experience in the field and have agreed to help us with our software development standpoint (building an application with longevity and code maintainability in mind), and they will act as stakeholders who will also be providing thorough feedback as users of the application
- The music listening community: Specifically college students and high school students we know will mainly be offering us feedback from a user standpoint, with some students with CS knowledge potentially providing feedback from a software standpoint
  - This is important because these students will be an accurate representation of potential users for this platform in the future, and their feedback will resemble the thoughts and ideas of many other users of this platform, which will allow us to build a solid application that users will enjoy using

3. Key Documents

There will be no documents necessary for our clients and stakeholders to give us feedback on this project. We will be responsible for taking notes and writing down the ideas that our clients and stakeholders will be giving us ourselves, in our own documents.

Working with a public repository, our clients will have the ability to open issues on Github, as well as provide code reviews on pull requests before we merge them to give us feedback from a software standpoint. This will make the software development process a lot easier as Github has engineered these tools specifically for this purpose.

From a user standpoint, users who encounter issues with our application may open issues on Github with a description of the user issue they are encountering, after which we could submit a PR to address that issue. Other user feedback can be discussed over messages or emails.

4. Interview Notes:

a. Rationale for stakeholders not interviewed

Of our college students and high school students, we did not need to interview every single person who we reached out to for help as there isn't much of a working product to discuss, test, and comment on from a user perspective. Mrs. Natalia Pevenva and Mrs. Margarita Koretskaya were unable to make it to any meetings we've scheduled so far due to scheduling issues and them being unavailable, but we will definitely meet with them for feedback in the future.

b. Interview Notes:

i. Group 1: Mr. Mikhail Rolshud and Mr. Vadim Koretskey

- The meeting was about 30 minutes long, where we introduced ourselves and our project needs, and explained the scope of the project and the actual real-world problem that we were trying to solve
- Mr. Koretsky made a suggestion for us to use Github's software development centered features and not just its integration with git in order to enhance our development experience, as well as allow us to have an organized form for receiving feedback and resolving issues from clients, stakeholders, and other potential open source contributors
    - We should take advantage of Github's issue, pull request, and code review features in order to facilitate discussion and feedback from a software engineering standpoint. For instance, if one of us wants to make a change, we would branch/fork and then make a PR.
    - If a client wants to comment on a block of code on a PR, they could do so with a code review
    - If a client wants to see a general change, or one of us (developers) wants to add a change for us or someone else to tackle, we could do so with a Github Issues
    - The gist of it is that all the tools are there and have been developed over the years, and we don't have to reinvent the wheel and can just use them to make it easy for ourselves, the clients, and the stakeholders
- Mr. Rolshud raised concern for an inevitable issue that we would have, and that all social media platforms similar to ours have when they first start out: a lack of users, and therefore a lack of content for more users
    - This situation makes it very hard to get the ball rolling as people will only want to actively use the app when there is lots of lots of content and other users on there already using it, but to get them we need a lot of content and users in the first place, and it's a closed loop
    - The solution Mr. Rolshud suggested would be to explore ways in which AI can be used to generate content, including users, their profile descriptions, and some comments that they make for different songs. That way, users won't be looking at blank review pages and a site with 0 users. They will see a vibrant site with lots of content and users on it already. That way, it is a lot easier to get the ball rolling and get more users to actually use the application. Additionally, by portraying the site as one with many active users, newer and true users will be more motivated to jump in and participate in the app
- Tech stack and implementation details regarding the software aspect of the project were discussed with the clients as well, and Bun was suggested as a new alternative to NodeJS, with the clients agreeing. They agreed that it was similar to having a Bundler and Runtime all in one, as well as it being reasonably reliable and stable at this point. If the need to switch runtime ever comes up, the switch can easily be made to NodeJS and a builder like Webpack can used instead

ii. Group 2: Daniel Keselman (College) and Nicolas Kichukov (High School)

- Meeting wasn't too long, online for about 15 minutes
- Two things that these stakeholders suggested were for us to figure out how to organize comments for the app, and how they would be structured to make it easiest for users to chat amongst each other within comment sections
- After speaking with them about our target audience, which includes teens like ourselves, the issue of comment moderation was brought up
    - In every social media platform, there is an issue of spam comments and comment filtering
    - Some form of moderation will be necessary to control the comment sections
- Stakeholders requested for us to come up with UI/UX wireframes for the application for them to look at
    - Lots of emphasis was placed on making it a very intuitive interface – they agreed with the concept of having something such as TikTok or Instagram style comments with the addition of more music-specific features such as potentially embedded snippets or hyperlinks to sections of a song

5. Competitor Products / Solutions

Social Media Platforms: Twitter (mainly text based discussion), Instagram (mainly videos and pictures, and comments under them), Reddit (form style discussion), Tiktok (short form content)

The issue with general social media platforms is that although they provide the opportunity for users to discuss music in theory, the ways of actually discussing music are very disorganized, as these platforms were created for general socialization as opposed to specifically music oriented discussion. Our app will tailor specifically to music oriented discussion that will support users' needs specifically for music discussion and improve upon the lack of features and organizations music enthusiasts experience on these general social platforms.

Music Streaming Services: Spotify, Apple Music, Pandora, Prime Music, Youtube Music, etc.

These services are very good at providing their users a platform to listen to music. There are lots of features that make it very convenient for users to listen to high quality music, browse, get recommendations, etc. These services are very great and convenient, but they don't offer any solutions for their users to discuss the music that they listen to and build communities based on their favorite songs, albums, artists, etc. This is what Songscope will do that is unique and useful for its users.

A mix of both: Soundcloud

Soundcloud took a very "Instagram" style approach to music, allowing anyone to post almost anything. There is a more community focus approach in Soundcloud. However, its system of music discussion is also extremely primitive, offering just comments and timestamps. The comments fly by faster than the user can read them, especially when there are a lot of them on one song, not all artists are on Soundcloud, and the reply and discussion aspect itself is as limited as just mentioning the original commenter in a new commenter. These discussion specific features specifically are what Songscope would improve upon. Generally speaking, allowing users to spend more time discussing, rather than simply leaving note-style comments on a song would enhance the music-listening experience.

A direct competitor: Rateyourmusic.com

This site does something similar to what we are planning to do in our project, however it is littered with ads, does not have a lot of functionality for comments and discussion, has more of a long form blog focus, an awkward UI, and we could we could drastically improve upon this and create a better product for the music enthusiast community.

Overall, Songscope aims to create a friendly user experience for anyone looking to share their opinions on different music and artists. We aim to replicate some of the better parts of competitors while simultaneously solving some issues their sites have.

6. Direct Observations

As this project isn't a traditional capstone project in the sense that we are making a product for a client in industry, and are making a product for the greater music community with our clients needing feedback, we will discuss the observations that we and our clients, as members of the music community, have faced. There is currently no easy way for us to discuss the music we listen to with others, and discover the communities behind our favorite songs, albums, EPs, and artists. Songscope will act as a solution to this problem, and will provide a platform for not only us, but the greater music community to discuss the music they listen to online.

7. Analysis

There is an issue with our platform, as well as effectively all social media platforms when they first start out, and that is the lack of content and users for the applications. Users want to see a vibrant community when they use the app, and comments and users will be created using AI tools in order to paint the picture of an app with an active user base.

This app, like all social media apps, would make sense to be available on all platforms, web and mobile. However, we will keep the focus to make a MVP on the web and make it a high quality product before we branch out to work on other platforms.

Because we will be working primarily on the web for now, we will engineer the code in a way to use both server and client side rendering, server for the Songscope content (to avoid back and forth web requests) and client for the Spotify content.

The issue of comment moderation and filtering came up in discussions with clients, and in order to resolve this issue, we will utilize different message filtering systems, most likely npm packages that we can find to dictate if a message is appropriate or not to be publicly available on the site. These types of systems are used extensively throughout effectively all social media platforms and we can make it work as well.

8. Summary of Findings

Make a bullet-list of the changes you intend to make based upon what you've learned from your needs assessment.

- Switch from React/Next to LitHTML as we are looking for something more lightweight and possibly do server side rendering right now and then make routes in the future if we want web apps to get user data
  - As opposed to doing a client side rendering approach and having lots of requests and API endpoints to maintain all we have to do is
  - Further research is necessary from our end to decide whether this is necessary, but we are looking to avoid frameworks that provide features we won't use
  - Consider importing comments from other platforms
- We must find an effective comment moderation system
  - Also find a way to prevent inappropriate usernames
- In the beginning stages of the website, to avoid empty comment sections, fill some comment sections with AI alongside real comments
  - DB Schema will be updated
- We will take inspiration from Tiktok's style of a comment section since it is a friendly UI and will match the theme of our site
  - Potentially take into consideration benefits from other platforms as well
    - Reddit's easy-to-use threads and replies, as well as upvote/downvote system
    - Embedded images/gifs from Instagram
- Include timestamps on all comments
- Have a persistent view for comments so that there are no issues with comments "flying by" as a user is trying to listen and read simultaneously.