

Team Name: Explorify

Members: Christopher Chang (cyc2136), Gracie Gilbert (gyg2104), Ahmad Maruf (aim2122)

Development Document

Overall Process:

At first, we came up with the idea of searching for a song, and then giving users a option of 3 possible ways to remix it. We liked the idea, and even designed the page using out MyBalsimiq, but feedback from TAs that it would be too difficult as well as a closer look at the API deterred us from actually implementing this. The next time that we met, we decided instead to make an application that would allow users to get interesting information about tracks as well as read articles/blogs about the track's artists as well. We first designed the three separate pages that we needed using MyBalsamiq after talking about what we wanted the user to be able to do. We decided on 3 pages because we thought it would be important to have the search results displayed on a separate page from the actual search, and for the info page to be a separate page from everything else. It was decided that each team member would be mostly responsible for one of the pages in terms of HTML/CSS - Christopher did the first page, Gracie the second, and Ahmad the third. Then, we worked together on the Javascript and on connecting the pages. Then, we split up work for the writings that needed to be turned in. Gracie and Christopher wrote the Use Cases/Personas, Gracie and Ahmad worked on the Development Document, Christopher worked on the User Manual, For the video, there was a collaborative effort.

Target Users:

Our target group of users would be people more interested in technical details of specific tracks, or interested in finding out lots of information about the track's artist - people like musicians, music reviewers, music enthusiasts. This app is not for casual listeners.

Our use cases are as follows:

Persona 1: Chris Johnson is a college graduate who recently just lost his job. His girlfriend just broke up with him so he's been hitting up the bar scene to meet more people. He loves music but doesn't know too much about it. He figures that it would be a good conversation starter in any situation. The problem is he doesn't know where he can go to for a central hub for all things music.

Scenario 1: Chris is at the Nomad Bar downtown and he starts talking to a girl. He brings up his favorite music/bands in the conversation (as planned) and she seems to be interested. He remembers everything he read through our app and starts talking more confidently about his passion for music. She is impressed by his vast knowledge and things seem to be clicking. He then remembers reading the news section in our app and remembers that his favorite band is having a concert in Brooklyn this weekend. He asks her to go with him and she agrees! Because of our app, Chris was able to share his passion for music and land himself a first date!

Persona 2: Tom Smith is a junior in college. He loves rock and electronic music, and is the guitarist of a band. One of his favorite activities is covering and writing popular rock songs. He often draws inspiration from existing songs by using similar keys, tempo, chord progression, etc, when he is writing his own songs. However, he doesn't have the best ear, and struggles with figuring out information such as the correct key of the songs on his own.

Scenario 2: Tom Smith's band has a show in a month. They want to cover a popular rock song, and also want to premiere one new original song. Tom already knows what song he wants to cover, but he is struggling to figure out specific information about how the song is played, and isn't able to tell just by listening. Luckily, Tom has heard about an app that will provide him with the data he needs to know about the song. Tom opens, and types in the name of the song his band wants to cover into a search bar. The correct song is the first result, a fact he discovers after clicking on the 30 second preview of the song. He clicks the info button for the song, and is taken to the song information page. Information about the original song is displayed, including the song's key/tempo/danceability/etc. He is also able to listen to the song by clicking on the play widget. He takes note of all of the song's information, and is able to successfully figure out a good arrangement of the song. Then, he clicks on the Home button, and is returned back to the search bar. Tom knows that he has to write an original song for his band. He has recently been very inspired by a song he has been hearing on the radio, and wants to write a similar song. In order to imitate the feel of this song, he wants to write a song in the same key, and the same tempo, and in the same time signature. He searches this song, finds the correct song by listening to the song audios returned in the search results, and then clicks on the information button. From there, he learns all about the song's specific information, and gets an idea of how he wants to write his own song. He also notices some articles about the artist on the information page, and finds one about the artist's songwriting process that he finds inspiring.

Persona 3: Jill Williams is a recent college graduate who works as a high school teacher. She prefers to listen to classical music and jazz, but she feels that she only is able to understand pieces on a cursory level. She wants to start learning more about both music theory and of music history.

Scenario 3: Jill is listening to a classical music playlist, and hears a piece that she loves. She wants to know more about this song, especially because it is in a very complex time signature and strange key that she is unable to figure out with her limited amount of musical knowledge. She decides that she wants to learn about this song, so she enters the app. Next, she types in the song name, and listens to the 30 sec preview of 4 songs in the search results until she finds the correct song. Then she clicks on the info button. On the info page, she learns all about the technical theory behind the song, but is also happy to see that there are artist biographies to help her to learn more about the history of the composer, as well as some really interesting articles that help her to better understand both the artist and classical music as a whole.

Design Decisions:

We decided to go for a minimalist design that would be easy to use and intuitive, but that also would not look like Spotify. In order to achieve autonomy from Spotify, we used a color scheme that doesn't include Spotify's colors. and return results in a table that contains different information and formatting than a regular Spotify search. We decided to make three separate pages.

The first page is a search page, which allows users to type in the name of a song to search for. We added brief instructions for clarity. This page redirects to the second page when the search button is clicked.

The second page contains a table with all the search results. The table consists of 2 columns, 1 with playable audio objects of the song (the 30 second previews we obtain by calling the regular Spotify API) and 1 with a button that, when clicked, takes the user to an information page about the track selected. We use session storage in order to remember the correct tracks when going from page to page.

The third page is the track info page for the selected tracks. This page contains all the data that the user could be interested in. To make the page more readable, we split the information into 2 separate panels, one for Track Information, and one for Artist Information. We decided that the information we wanted the users to know about were the following: Track key, mode, loudness, tempo, time signature, danceability, title, album, popularity, and artist, and Artist biography, blog, twitter handle, image, hottness, and news. The reason we decided on the track endpoints was because we thought they would be the most helpful in terms of learning music theory behind a song, which was part of what we wanted to give users access to (along with basic track info). The artist information we chose was more towards just interesting information news wise about artists because our idea was that the user who likes the track might be more interested in how the artist works when writing, what the artist is actually like, etc. We wanted the twitter there also just for easy followup on the artist later. The third page is also where we have the Play widget. Our reasoning was that on the search results page (page 2), it is only necessary to have 30 second previews because the user is just trying to figure out which song is the correct song, but on the information page, the user may want to hear the complete song (and hear it using Spotify).

In order to make the website more like pages the user has visited before, as well as easier to use, we added a Home button and Back button to all the pages but the homepage (for consistency, we added it in the same place on each page). We also tried to keep the CSS as consistent as possible, and we decided to make the tables look the same on pages 2 and 3, as well as the idea of using panels.

Here are some other ways we thought about design heuristics when making our app:

Heuristic Analysis:

1. Match between system and real world. E.g. home button, back button, everything appears in a logical order, we don't use any technical speech in any part of our match between system and real world.

2. User control and freedom: Our home and back buttons are emergency exits from unwanted states.
3. Consistency and standard: We maintained c. & s. both in our platform conventions and design.
4. Error prevention: We reduce error conditions by making it really difficult to do anything incorrectly - if the user mis-selects something, they can just go back, and other than that, the design is so simple that there really isn't anything the user can do wrong.
5. Recognition: We add the player (in page 2), so that user doesn't have to remember the song when looking through search results.
6. Aesthetic and minimalistic design: We make it as simple and intuitive as possible.

Prototyping and Testing Process:

For the prototyping process, we made multiple versions of our website on MyBalsamiq to see what we thought worked best. We decided to use the MyBalsamiq to show how each page should look and so it was all high-level design rather than getting really in depth about the mechanics. At first, we were going to have the search results appear on the same page as the search bar, but after going over the designs again, we decided that it was too cluttered. We also ended up adding endpoints that we realized could also be useful - at first, we were not going to have the Artist section, but we ended up adding it. Then, we made the website and tried to make it look exactly like the MyBalsamiq that we designed.

For testing, we only tested within our team. What we would do was that whenever a team member finished a part, they would upload it to github and everyone else would download it and test it on their own computers. They would let the original member know what they thought of the design, for example whether or not they thought it looked consistent with their own pages, and also pointed out areas that didn't work. Then the original member would make changes and the process would be repeated. This was also helpful if a member was having trouble getting something to work correctly, because then another member could easily step in.

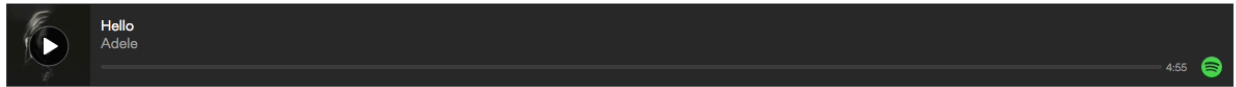
Then we each tested the final site. The way we did it was by searching for a variety of songs, and then making sure that search results came up that we correct given the search terms, and then making sure that all the information was correct.

Here are some screenshots of issues we solved (check MyBalsamiq for the prototypes):

This is a screenshot of the final working last page -

[< Back](#)[Home](#)

Explorify



Track Info	
Track Name: Hello	Album Name: Hello
Artist: Adele	Popularity: 100 (Scale from 0 to 100)
Loudness: -6.129 (Decibels)	Danceability: 0.471 (Scale from 0.0 to 1.0)

As opposed to these other times:

Here, the track info was working, but the key and mode were in the returned from API state of numbers which didn't make sense, so we realized we'd have to change it.

Track Info	
Key: 5	Mode: 0 (modality)
Tempo: 157.98 (BPM)	Time Signature: 4 (Meter)
Energy: 0.431 (Scale from 0.0 to 1.0)	Liveness: 0.085 (Scale from 0.0 to 1.0)

The Track Info was working here, but not the Artist info, and we were having issues getting the widget to load on one of the browsers

[< Back](#)

[Home](#)

Explorify

Track Info	
Track Name: Hello	Album Name: Hello
Artist: Adele	Popularity: 100 (Scale from 0 to 100)

⌵ Elements Console Sources Network Timeline Profiles Resources Audits 15 ⋮

This was what it looked like when the Artist info wasn't working during testing

Artist's Info	
Biography	Recent Blog Article
Recent Reviews	Recently in News

Software Engineering:

We used HTML, Javascript, CSS, and Bootstrap. The APIs we used were Echo Nest, Spotify API, and then we used the Spotify Play Widget.