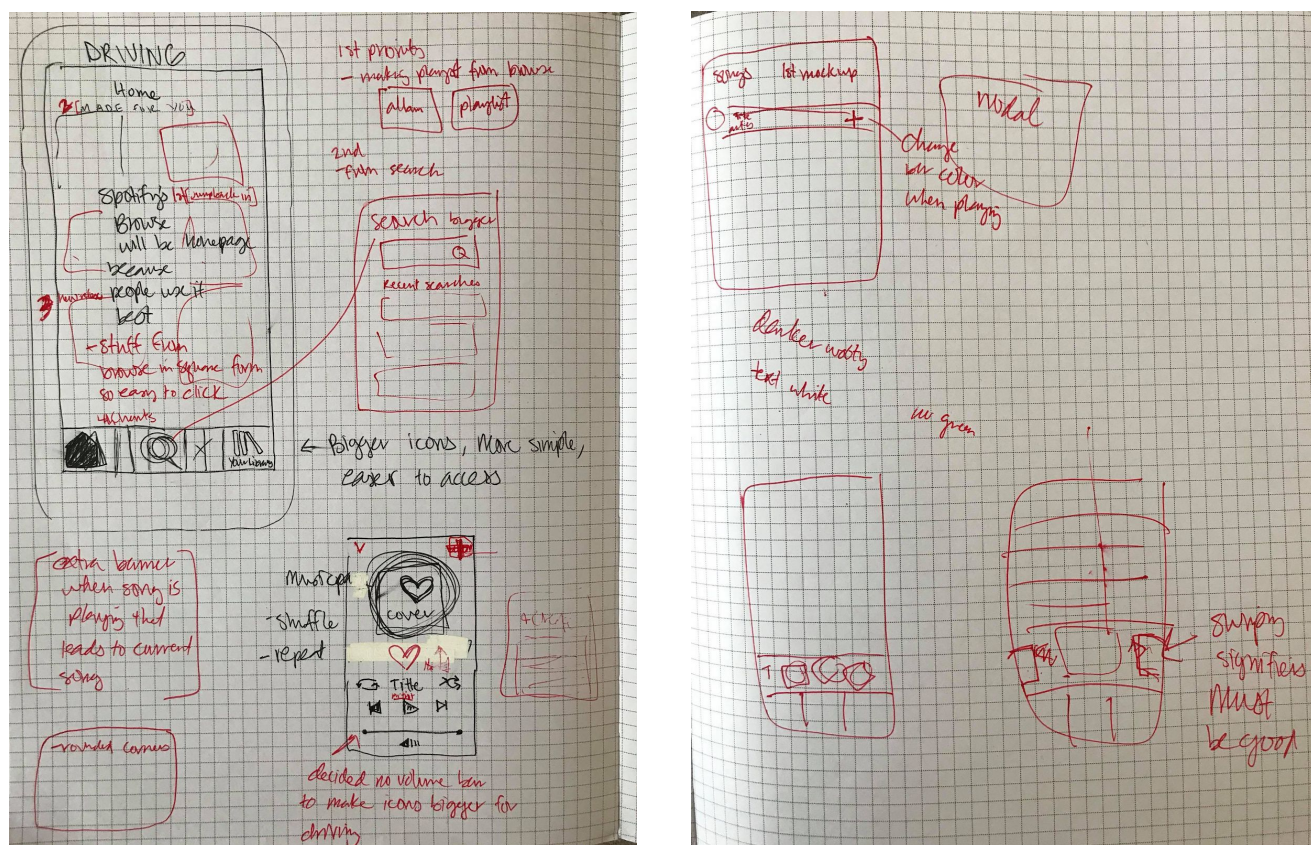


Design Decisions

We wanted to redesign an app that would maintain, or even improve, the usability of a popular music app. Seven out of nine of our interviewees noted that they used Spotify as their primary music app. Thus, we looked to reevaluate the app by taking into account the responses of our interviewees and directing that towards a target audience: young adults.

The harsh reality for young adults is that at one point in time, we have taken our eyes off the road in order to browse through our playlists on a cell phone screen. This inevitable scenario inspired us to reevaluate the usability of Spotify. This reevaluation would gear towards improving driver usability/visibility, but not at the cost of regular usability.



Sketches of our initial ideas of an app catered towards drivers. Inspired by Spotify.

Below are some of the most difficult design decisions we made while making note of our interviews.

Decision 1: Swiper No Swiping?

We initially prototyped the song screen because users use a music app because they want to listen to a song(s). If a user listened to a song they really enjoyed, they would add that song to a playlist through the 'currently playing' screen; All users we interviewed noted that they found a song **then** added it to the playlist. Taking this into account,

Our first solution to improve driver usability and visibility was to mimic the Spotify user interface and redesign it in three different ways:

1. Increase the size of the buttons to afford easier pressing
2. Prioritize the back, pause/play, next buttons
3. Place the '+' button (add to playlist) in spot that would discourage pressing unless safe (Spotify's love/like button is a single press, while its '+' button opens a separate screen to add to playlist)

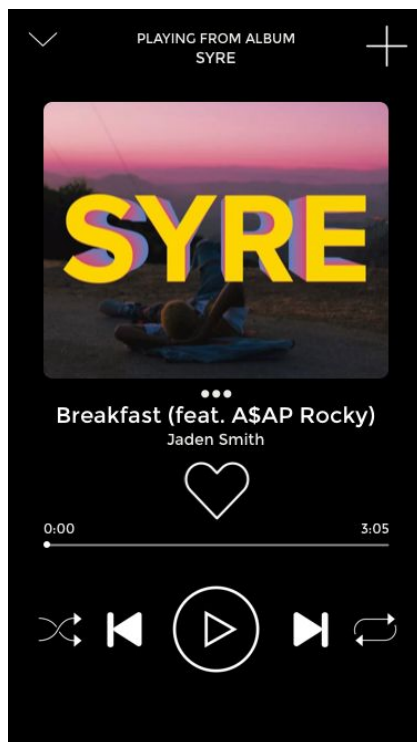


Figure 1.1
Rearranging the main buttons

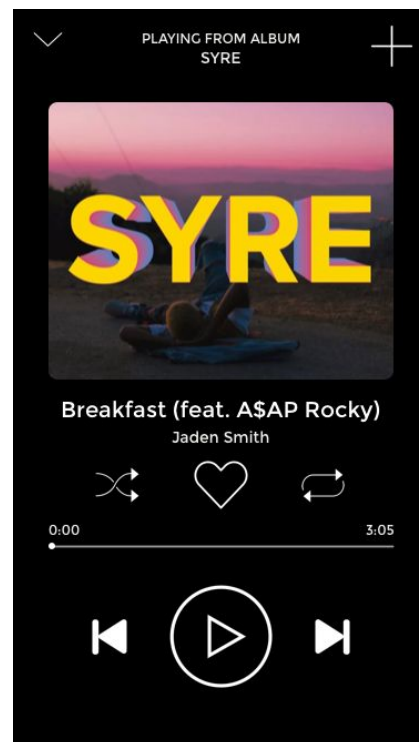


Figure 1.2
Rearranging the main buttons

Once we sketched and prototyped these interfaces, the usability issues became clear:

A user needs to press buttons.

While buttons do allow for more flexibility in terms of user interface, if a user is repeatedly distracted from the screen or isn't looking at the screen, buttons are hard to aim for. Music apps such as Spotify and Apple Music place their main functions towards the bottom of the screen, hinting at a possible standardization of the button placement. However, in a scenario where a single missed touch could quickly redirect the user's attention, we concluded that simply increasing button sizes and making the play button more prevalent wasn't enough.

Implementing Swiping

We concluded that the best way to solve this button pressing issue was to implement more swiping movements. Why?

Swiping decreases the amount of slips and increases the window of error.

Sometimes a user will know where the button is, but will just slightly miss the button they intended to press. By implementing more swiping on the song screen and throughout the entire app, users will have less trouble....

However, once we concluded the swiping was optimal, another issue arose. How would we implement these swiping functions? Topics that were discussed included:

Would swiping features clutter the user interface?

In addition to the crucial buttons that are necessary to a music application, would swiping lead to more unnecessary noise for the user? Creating visual signifiers would be cool, but wouldn't that continue to clutter the UI? The idea of swiping seemed to directly contradict with our goal of simplifying the app as much as we could. (See Figure 1.3)

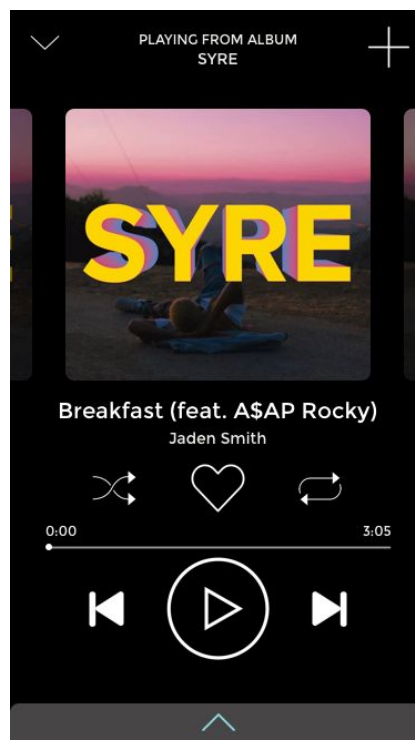


Figure 1.3

Note the signifier at the bottom to swipe

Would swiping features making the user interface too plain?

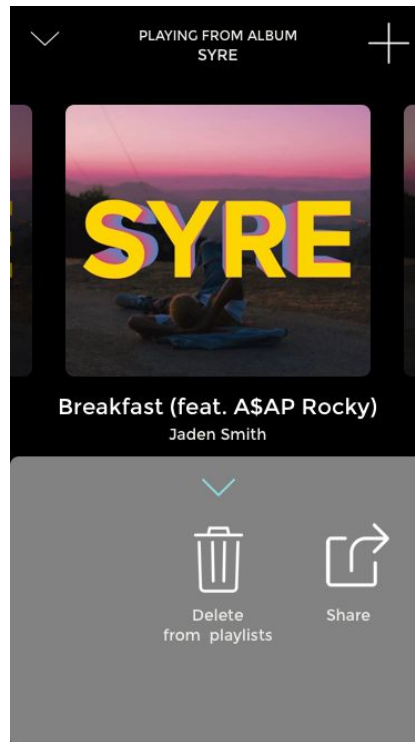


Figure 1.4

Bigger = better = uglier?

Swiping up to bring up additional features while also having big enough icons would not be aesthetically pleasing (see Figure 1.4). A visually displeasing interface turns into a poor user experience because the user simply will not want to navigate through different screens. We wanted to find a way to retain the swiping feature while we minimally sacrifice aesthetics. To do so, we looked to SoundCloud.

Despite only one of the interviewees saying that she used Soundcloud, she noted that one of her favorite functions of the SoundCloud user interface is its swipe function. The screen of a Soundcloud song affords swiping from any point on the screen to go to the next or previous song (See Figures 1.5 and 1.6). In contrast, Spotify allows you to change the song by swiping on the album cover, rather than the entire screen.



Figure 1.5

The screen of the song that is currently playing



Figure 1.6

Swiping to change the song

Taking ideas from both Spotify and SoundCloud, we decided to sacrifice the primary play/back/next buttons in order to promote the swiping movement. We concluded that if the user were to locate a song from browse, a 'Like' button would take priority, especially if the user was distracted. These can be seen in our final prototype on InVision.

Decision 2: Information Architecture: Organizing the Homepage

What looked to be a trivial issue ended up being one of the topics that sparked the most discussion.

On Spotify's homepage, the headings for playlists are organized as followed: (see Figure 2.1)

Made For You
Jump Back In
More Like [playlist name]
Recently Played

(see Figure 2.1)

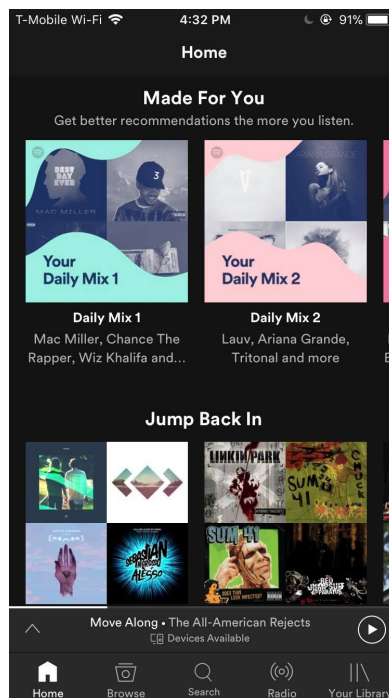


Figure 2.1

Spotify homepage

In terms of usability and aesthetics, we believed that Spotify implemented the best possible interface. Swiping left/right allows a user to further dive into a heading without endlessly scrolling down. There is a two column layout, as well as a visual signifier to swipe left. We mimicked this layout in our final prototype. What we didn't mimic is the order in which Spotify organized the headings.

Spotify's home page begins with the heading "**Made For You**", a collection of playlists that Spotify creates for the user based on the kind of music they listen to. Our initial disagreement towards this heading was its actual usefulness:

Argument A: Spotify implementing this is helpful because it allows the user to discover new music that they never knew existed. It's a shortcut to browse music that may not be popular.

Argument B: Based on the users who preferred Spotify as their main app, none reported that they used "**Made For You**" to discover new music. Most recall songs that they like or ask for recommendations from friends.

There are two reasons as to why Argument B emerged as the superior argument.

1. It's all about the user.

We initially lost sight of who we were redesigning for. We were so caught up in our personal sentiments towards the heading that we forgot to base our decisions off of user interviews.

2. The target audience is young adults.

We wanted to cater our app towards young adults. All nine of our interviewees were college students, with eight out of nine interviewees reporting that they made their own playlists. Leading us to the inference that we make our own playlists because we want to listen to our own playlists. In turn, we would want to resume listening to a playlist if it were to be interrupted.

Our interviewee Kelly stated that she would like more options to add songs for a long drive. Once she goes on a road trip with her friends, the car would likely make a pit stop. When the road trip starts up once again, it'd make sense for the playlist (that Kelly took so long to create) to resume playing.

Therefore, instead of prioritizing pre-made playlists, we prioritized the "Jump Back In" heading. From there, we decided to remove all headings that tailored the playlist to the specific user (based off our interviews); young adults tend to be more excited about a new album coming out than undiscovered music. When the user opened the app, they would see the following headings without scrolling down:

Jump Back In

New Releases

Genre/Mood

.

(see InVision prototype)

Decision 3: Flow

In Spotify, when creating a new playlist from the song that is currently playing, the modal that pops up is very different from the aesthetic of the app (see Figure ...). If the user is using an iPhone, the modal that pops up is a part of iOS rather than Spotify. Our team couldn't decide whether or not this interrupted the flow of the system.

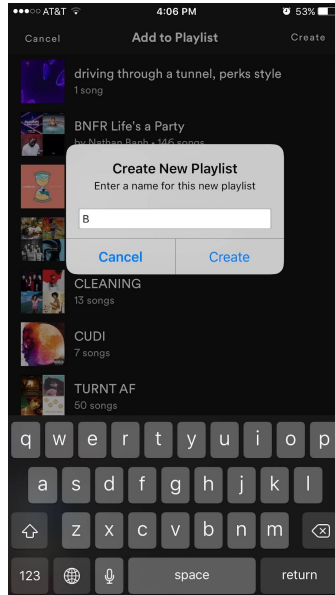


Figure 3.1

The iOS modal pops up when the user attempts to create a playlist in Spotify

There were two arguments to this issue as well.

Argument A: The Apple modal is in the center of the screen and is aesthetically different from the Spotify application. Thus, it will redirect the user's attention much quicker. Aesthetics don't matter here because the user has already reached their goal of creating a playlist.

Argument B: Aesthetics and flow do not need to be sacrificed in order to create something just as usable. As long as the modal to create a playlist popped up in the app and was distinct, there would be an appropriate place to put it.

We went back and forth on this subject, but once we referenced our user experience and workflow comparisons, we came to a conclusion.

Recall in our user experience and workflow comparisons, we compared the "share playlist" functions of Spotify and Soundcloud. For each sharing experience, we concluded that Spotify was superior because it remained in the same app throughout the entire sharing process; SoundCloud opens the default Apple modal.

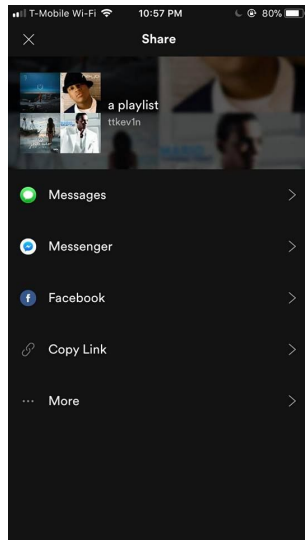


Figure 3.2

Spotify implements its own UI to share playlists

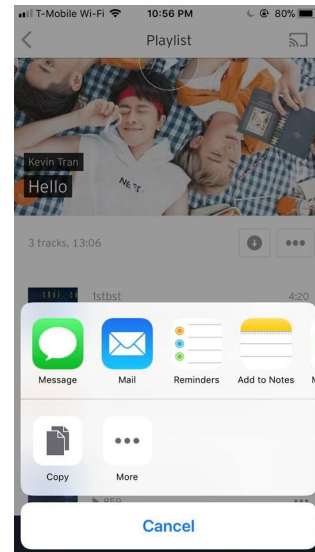


Figure 3.3

To share SoundCloud playlists, the iOS modal is

Drawing inspiration from our comparisons, we decided that we should be consistent with our design and in turn, leads to a flow within the system. We designed a modal within the app that would be minimal (see Figure 3.4).

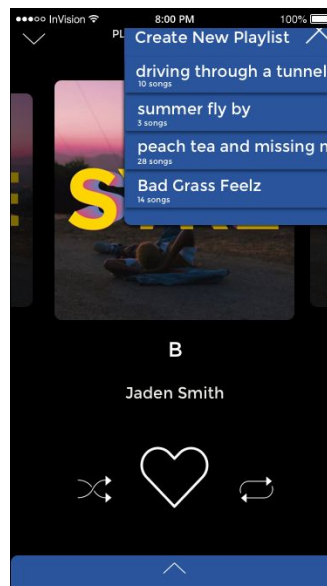


Figure 3.4

Final prototype of creating a new playlist from the currently playing song.