Introduction

Team

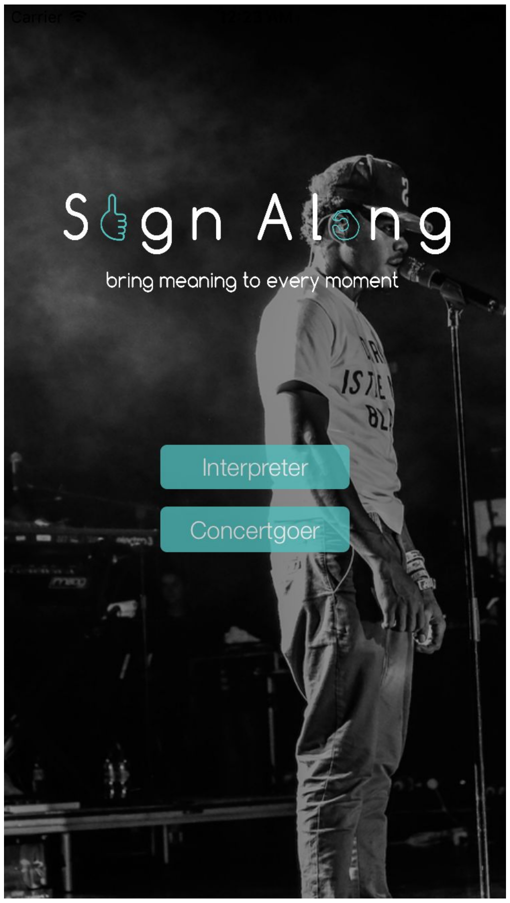
Varis N.	Design
Helen F.	Product
Christian W.	Testing
Minna X.	Dev

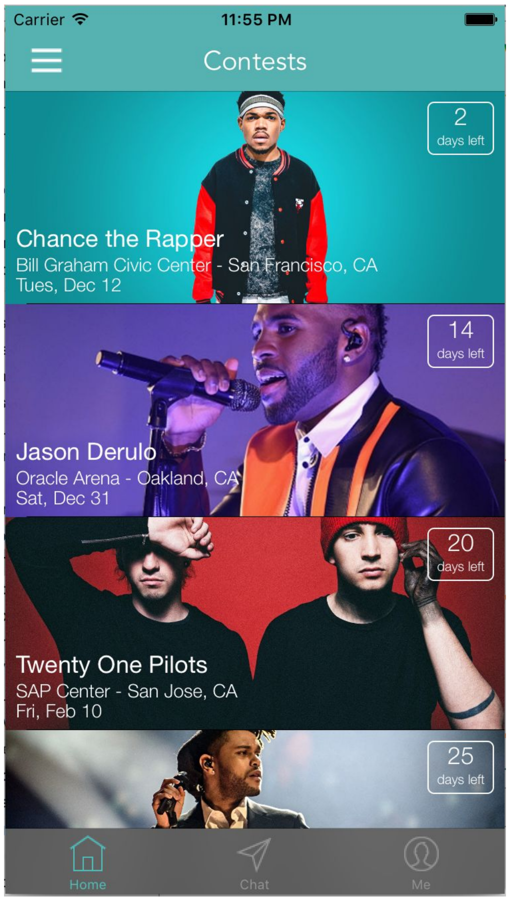
Value Proposition

Bring meaning to every moment.

Problem and Solution Overview

The deaf community is not the intended audience for most live events, making it difficult for them to enjoy and share live experiences with their loved ones. Exacerbating this issue is the shortage of interpreters; there is little incentive to become a certified ASL interpreter because of the high technical barrier to entry. Our product, Sign Along, addresses both these issues by giving interpreters the opportunity to engage in passion projects (interpreting for music concerts) that do not require certification, and by giving deaf users the agency to vote for their favorite interpreter to come onstage and sign for their favorite artists.





Chance the Rapper
Bill Graham Civic Center - San Francisco, CA
Tues, Dec 12

Jason Derulo
Oracle Arena - Oakland, CA
Sat, Dec 31

Twenty One Pilots
SAP Center - San Jose, CA
Fri, Feb 10

Tasks and Final Interface Scenarios

1. **Simple task**: Find upcoming concerts in your area to submit a video for. Reason: Interpreters want to know what upcoming concerts are coming to the area so that they can combine their love for music with their ASL fluency and potentially perform with their favorite artists/bands.

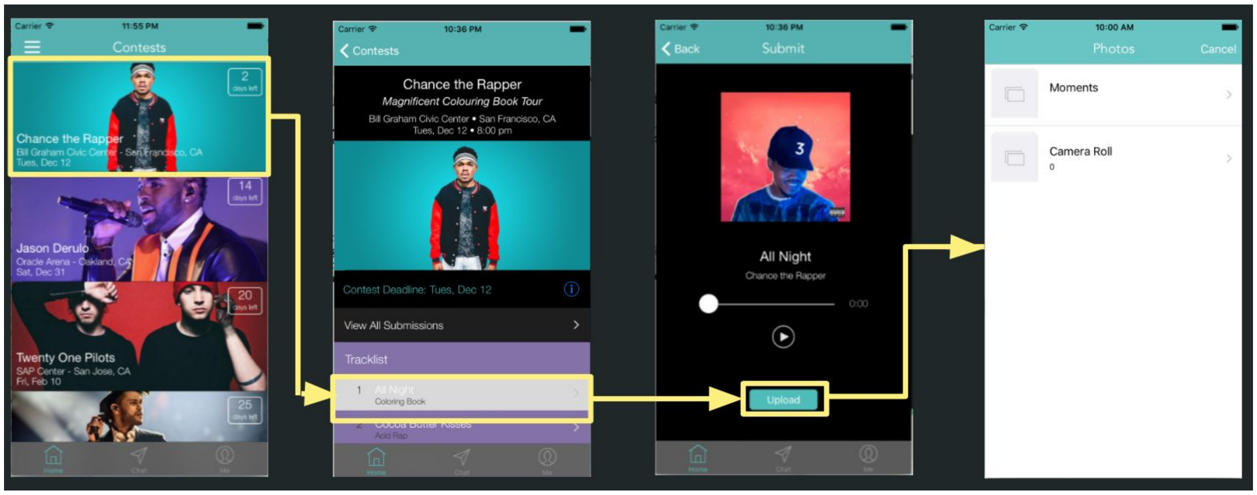


Figure 1: simple task, interpreters see a list of upcoming contests, tap for more details, tap a track, upload recording.

2. **Medium task**: Find and vote for your favorite interpreter out of the pool of submissions.
   Reason: Concert-goers want the best interpreters for the concerts they attend to maximize their concert experience.

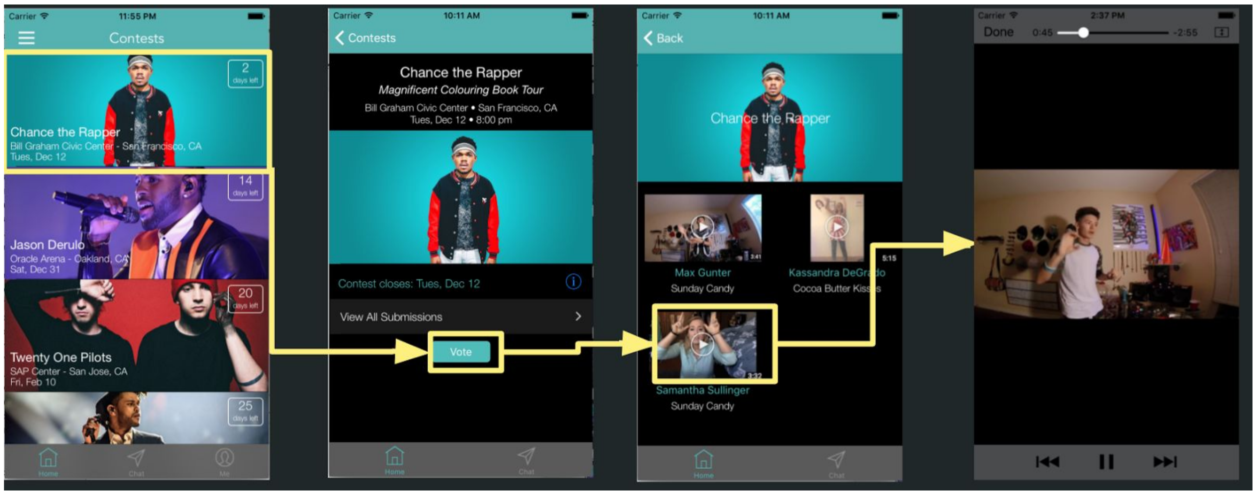


Figure 2: medium task, concertgoers view a pool of submissions for a given contest.

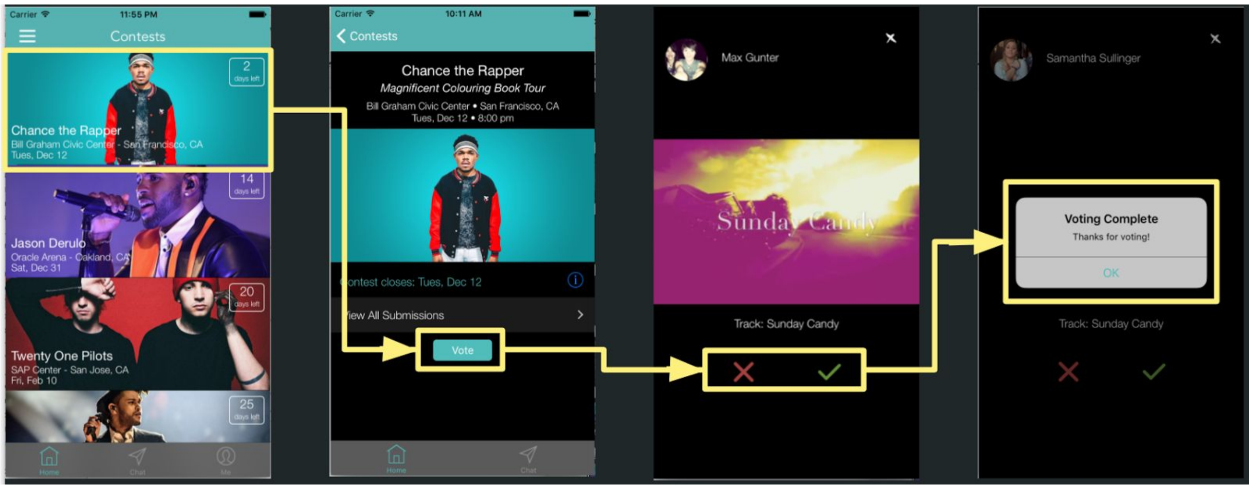
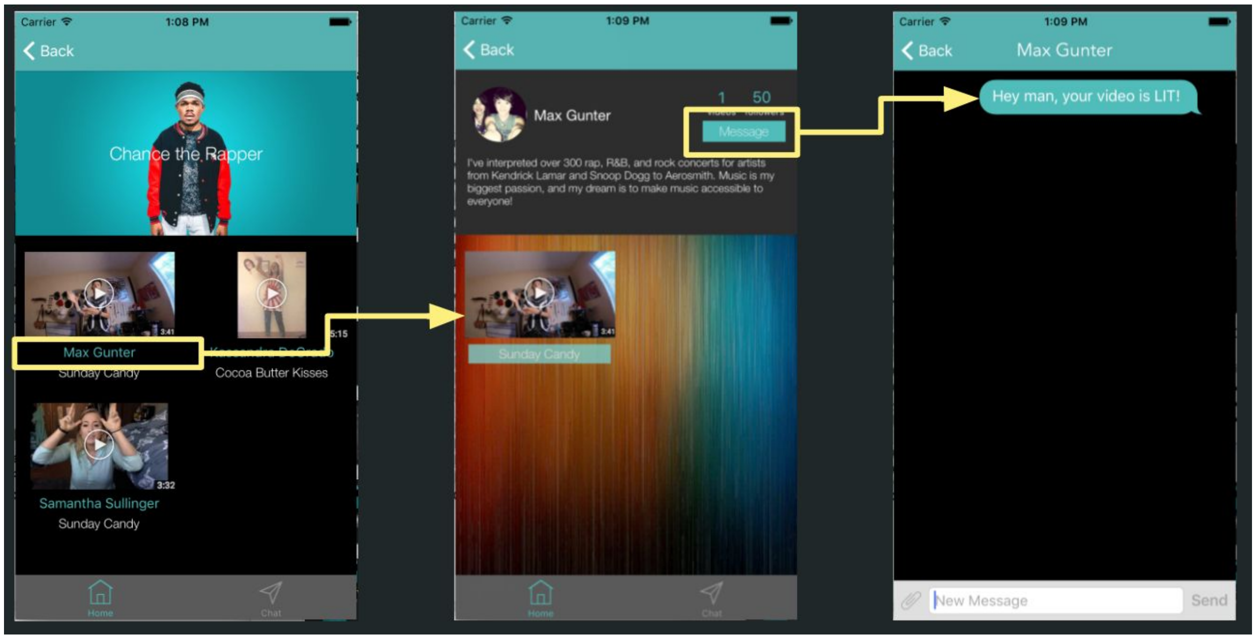


Figure 3: medium task, concertgoers vote whether or not they like a series of submissions.

3. **Complex task**: Engage and establish relationships with your favorite interpreters. Reason: Concert-goers want to build relationships with interpreters that they've had good concert experiences with, so they can continue to coordinate the concerts that they both attend.



I've interpreted over 300 rap, R&B, and rock concerts for artists from Kendrick Lamar and Snoop Dogg to Aerosmith. Music is my biggest passion, and my dream is to make music accessible to everyone!

Figure 4: complex task, concertgoers message an interpreter they think is awesome!

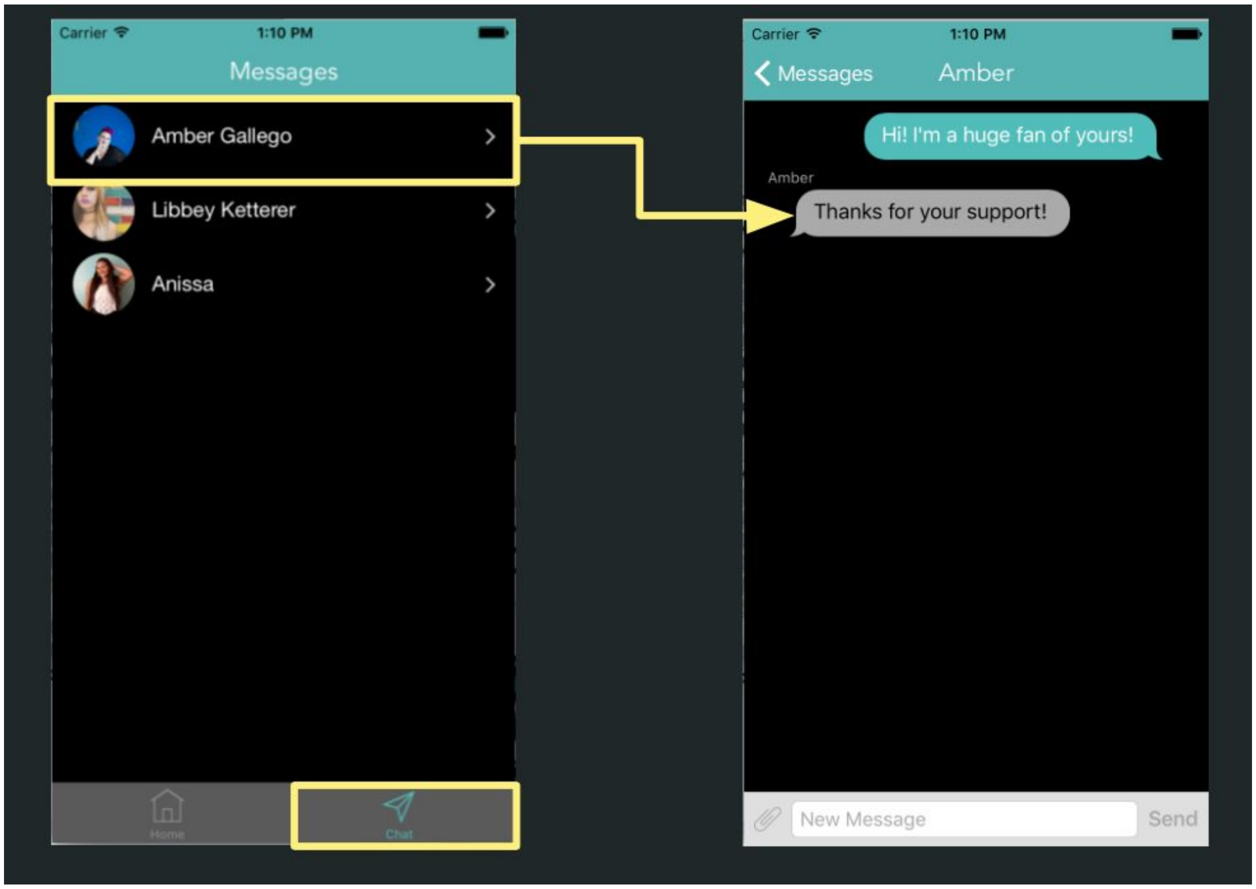


Figure 5: complex task, users can see all of their chats and continue chatting with their favorite interpreters.

Design Evolution

View Upcoming Concerts

In our earliest prototypes, we had interpreters (one-time only) set their preferences based on genre and artist before viewing a list of relevant upcoming concerts in the area. Based on usability testing on our low-fi prototype, users told us that they would rather list of concerts in their area right away. We decided to remove the preferences setting and instead directly take interpreters to the list of upcoming concerts. This made the process simpler and more streamlined. The final screen design/function changed based on HE data, so that will be addressed in the next section.

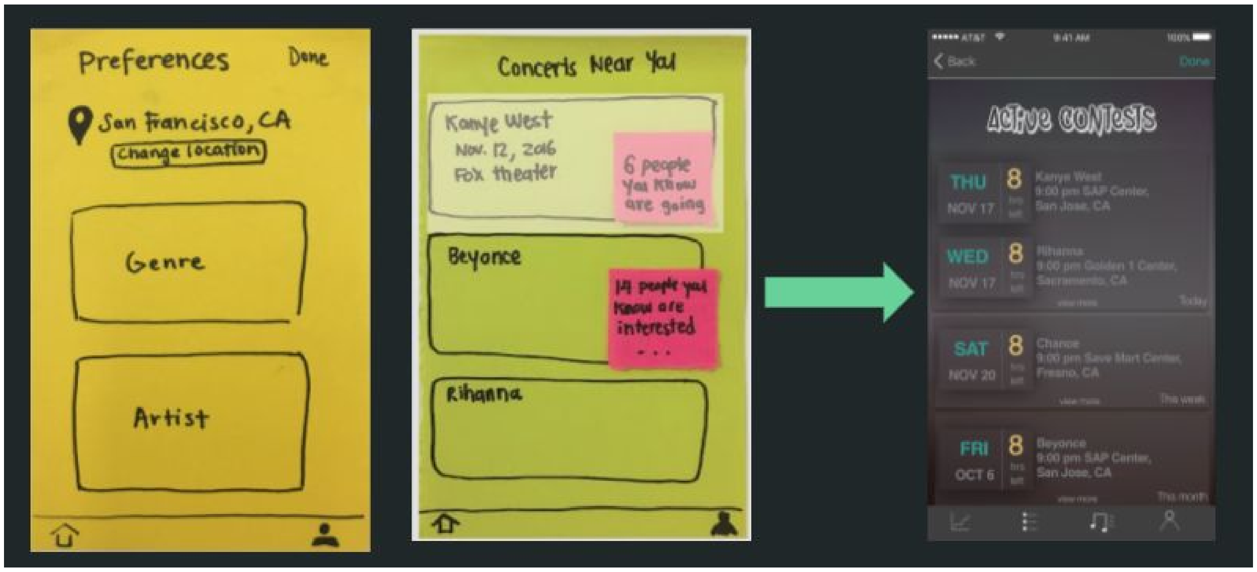


Figure 6: Evolution of upcoming concerts page

Uploading Audition Submission

From usability testing on our low-fi prototype, users expressed interest in being able to practice before uploading their submission (if they were recording their submission within the app). To address this, we included the option to play the track (to hear the song), see the lyrics as the track is played, and the record and save buttons (so that they can record and save multiple drafts of their interpretation). Changes were made in the high-fi based on HEs which will be discussed in the next section.

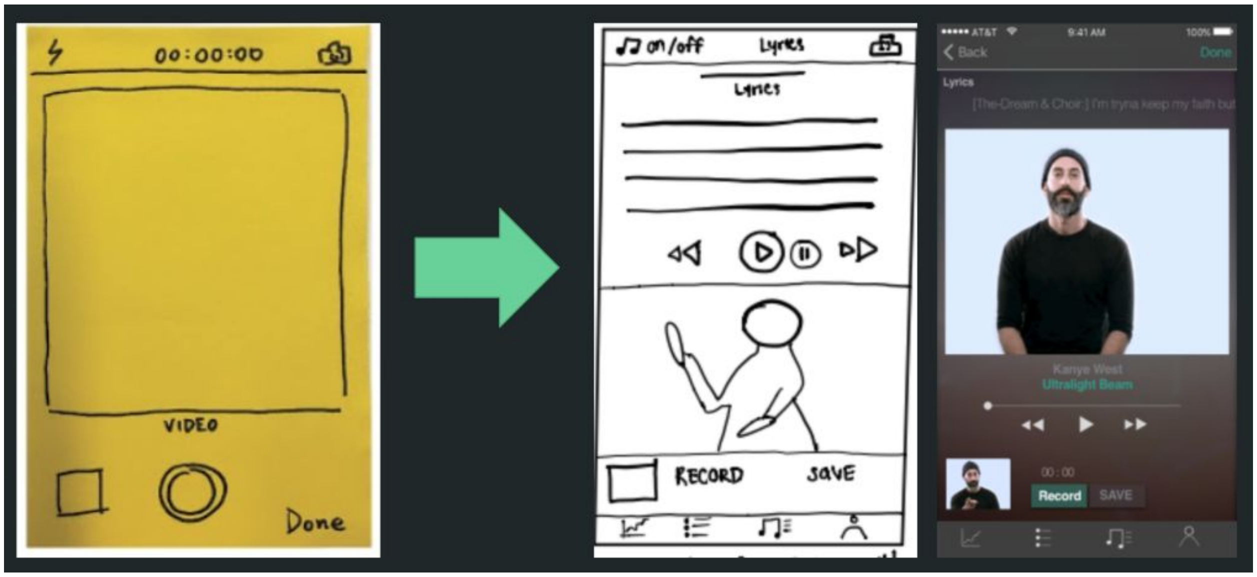


Figure 7: Evolution of video submission process

Voting for Interpreters

Usability testing on our low-fi prototype,revealed that users were confused by the voting mechanisms. The “yay” and “nay” tapping buttons were confused with swiping motions. Swiping up and down to leave interpreters feedback on their submissions was unintuitive. In the medium-fi prototype, we eliminated the up/down swiping features and replaced the yay/nay buttons with a heart for “like” and a x for “dislike”. To leave a comment, users would tap on the comment bubble. Changes were made to voting after receiving group HE feedback and will be addressed in the next section.

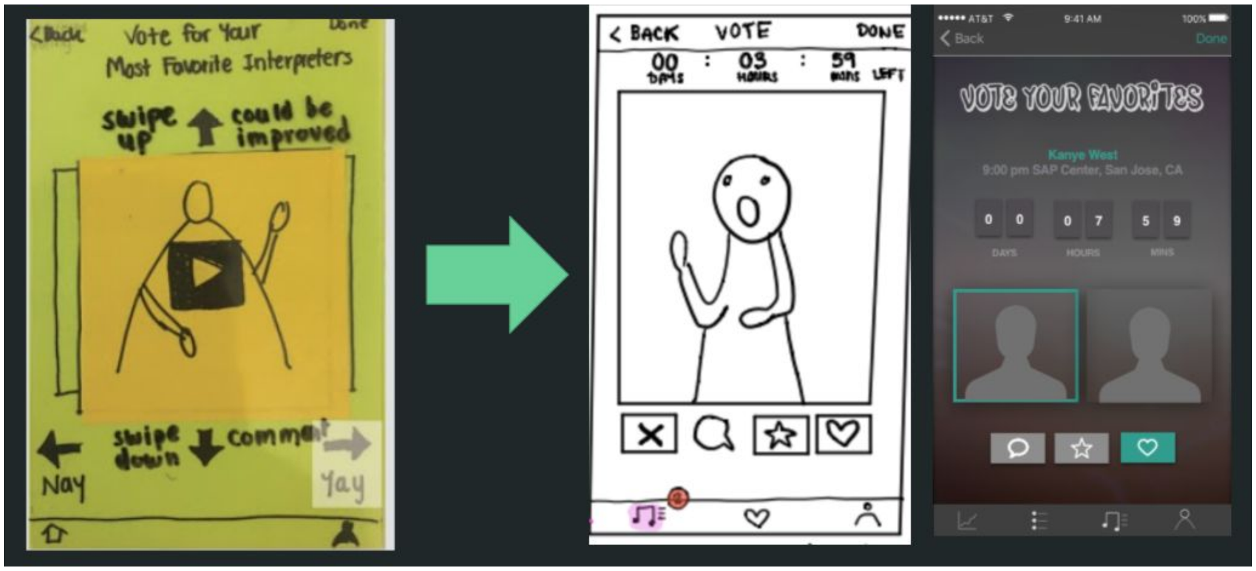


Figure 8: Evolution of the voting process.

Major Usability Problems Addressed

Our medium-fi prototype contained 29 violations in total, 11 of which were level 3 or 4 violations.

Level 3 Violations and related changes

1. H2-5. Error prevention / Severity 3 / Found by: A, D
   In the contest screen, interpreter-users are given the option to select the “Done” button despite not having chosen any song/track option to sign along to. This also occurs in subsequent screens, where users can select the “Done” button without having chosen anything. This can be confusing to users, and can be alleviated by only providing “Done” buttons when the user has chosen an option.

   Solution: In the hi-fi prototype, we eliminated the “Done” button from the contest detail screens, preventing the user from progressing to a track screen if nothing was selected. Now, Interpreters can only progress to the tracklist submission page if they tap/select a track. See Figure 9 for reference.

Additionally, we removed all instances of the countdown timer for the submission deadline. The same information would be communicated by giving the deadline date and was programmatically simpler to implement. Also a help button was added so users could read more about how the contest works.

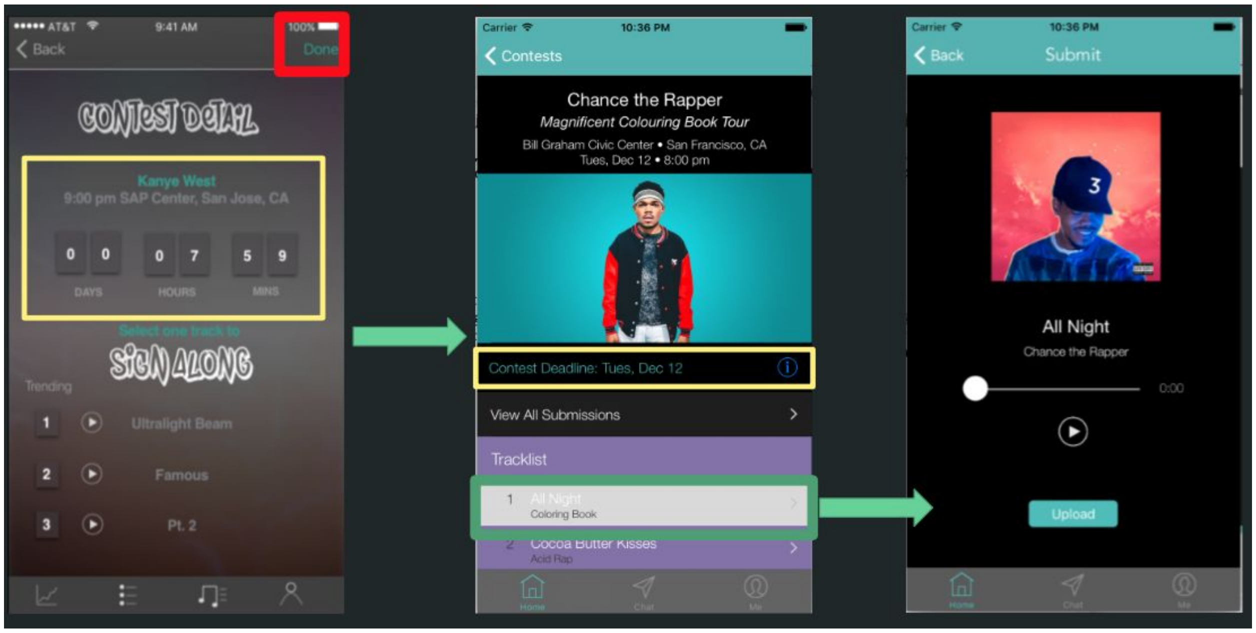


Figure 9: Violation 1 before/after..

2. 7. H2-10. Help and documentation / Severity 3 / Found by: B, C, D
   As a user, I am unclear as to the difference between the star and heart icons on this display. Intuitively, I wonder the difference between "starring" or "hearting" something-- in my mind, it would equate the same action, like "liking" something. If the starring equated bookmarking something, it should have a more bookmark-like icon. Instead, in order to eliminate this obscured status of system components, I would choose icons that carry more specific connotations than heart and star do in order to differentiate them, or I would merge them into one.

   Solution: In the hi-fi prototype, we eliminated the star and heart icons and replaced it with a green check for "like" and a red x for "dislike". The new color and symbol scheme are commonly associated with their meaning (from further testing, users found this to be more intuitive). See Figure 10 for reference.

3. H2-5. Error Prevention / Severity 3 / Found by: A, B, D
   When the user is taken to the screen where she is to vote on her favorite interpreter, it is unclear how to vote. Currently, the method is to tap on the picture of the person and then tap on the "heart" button. However, this method is confusing and prone to errors, so I would suggest a Tinder-like interface where the user can swipe left or right based on whether or not she liked the interpreter.

Solution: In the hi-fi prototype, we decided that a Tinder-like interface would be appropriate and more intuitive to use. The voting mechanism was simplified to a green check for “like” and a red x for “dislike”. In order to vote “like”, the user taps on the check mark and similarly for dislike. We decided that because the primary focus is voting, we only want to present users with voting options here. See Figure 10 for reference. We opted to have each video submission start autoplaying upon display to create a more involved and immersive voting experience.

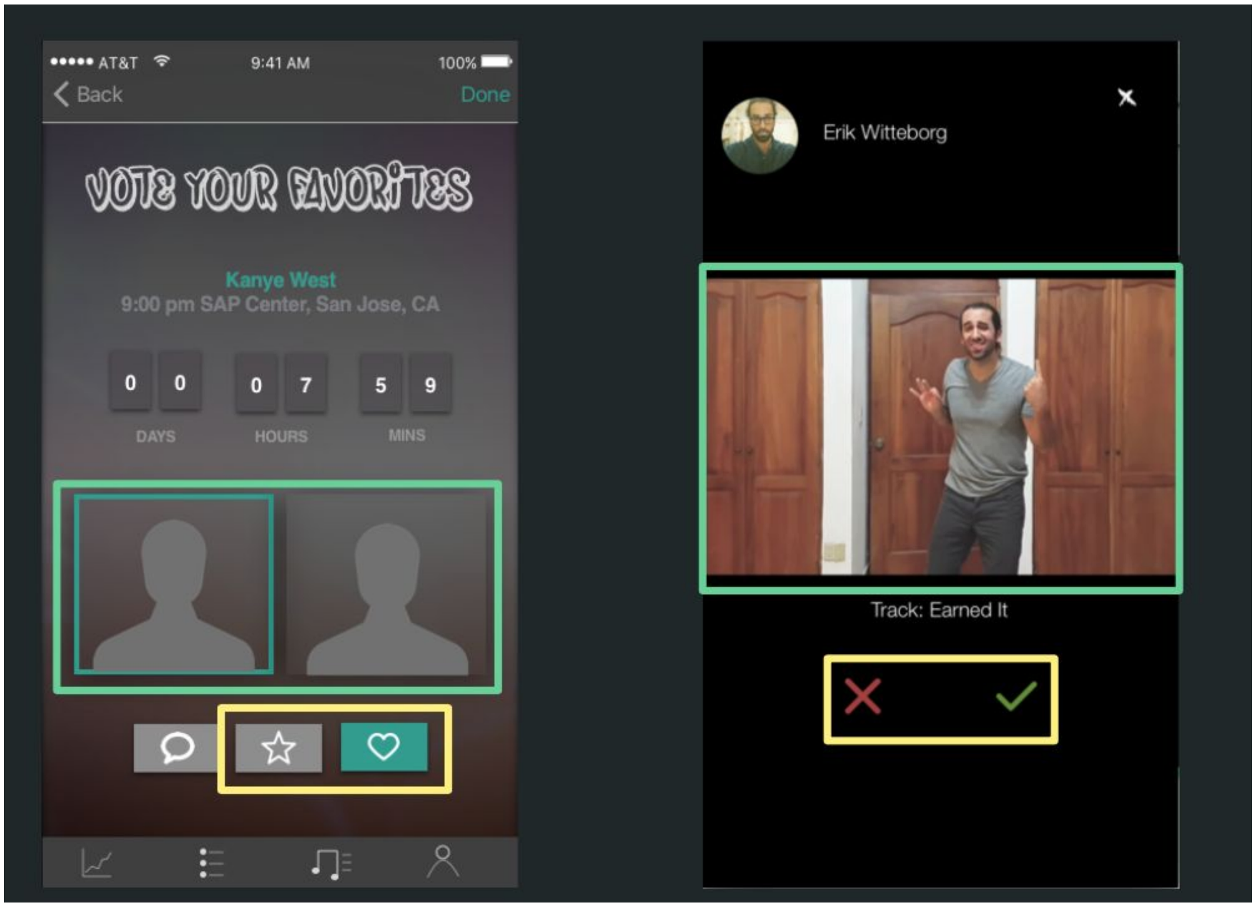


Figure 10: Violation 2 and 3 before/after..

4. H2-10. Help and documentation / Severity 3 / Found by: B, D
   There is no documentation as to what the tabs at the bottom of the screen are used for – and they are also not used at all in the Med-fi prototype, so we are also unable to identify their function. An improvement could be to include a descriptive label at the bottom of each symbol.

   Solution: To eliminate confusion about the purpose of each icon on the bottom tab, each icon now contains a brief description attached to it. The “Home” icon takes you to the contest features, “Chat” takes you to all of the messages/chats you have, and “Me” takes you to your profile page. This brief annotation should give users a clear sense of the purpose of each icon. See Figure 11 for details.

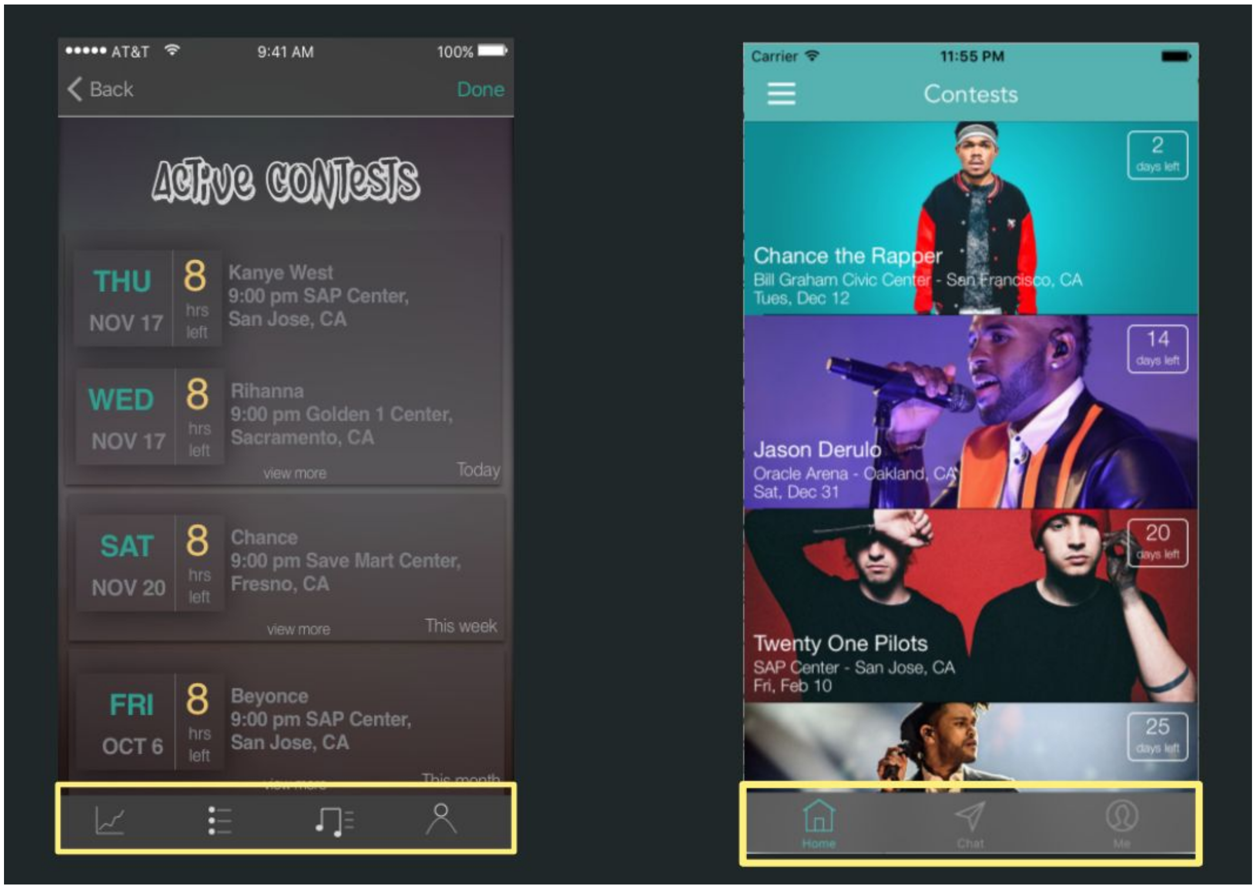
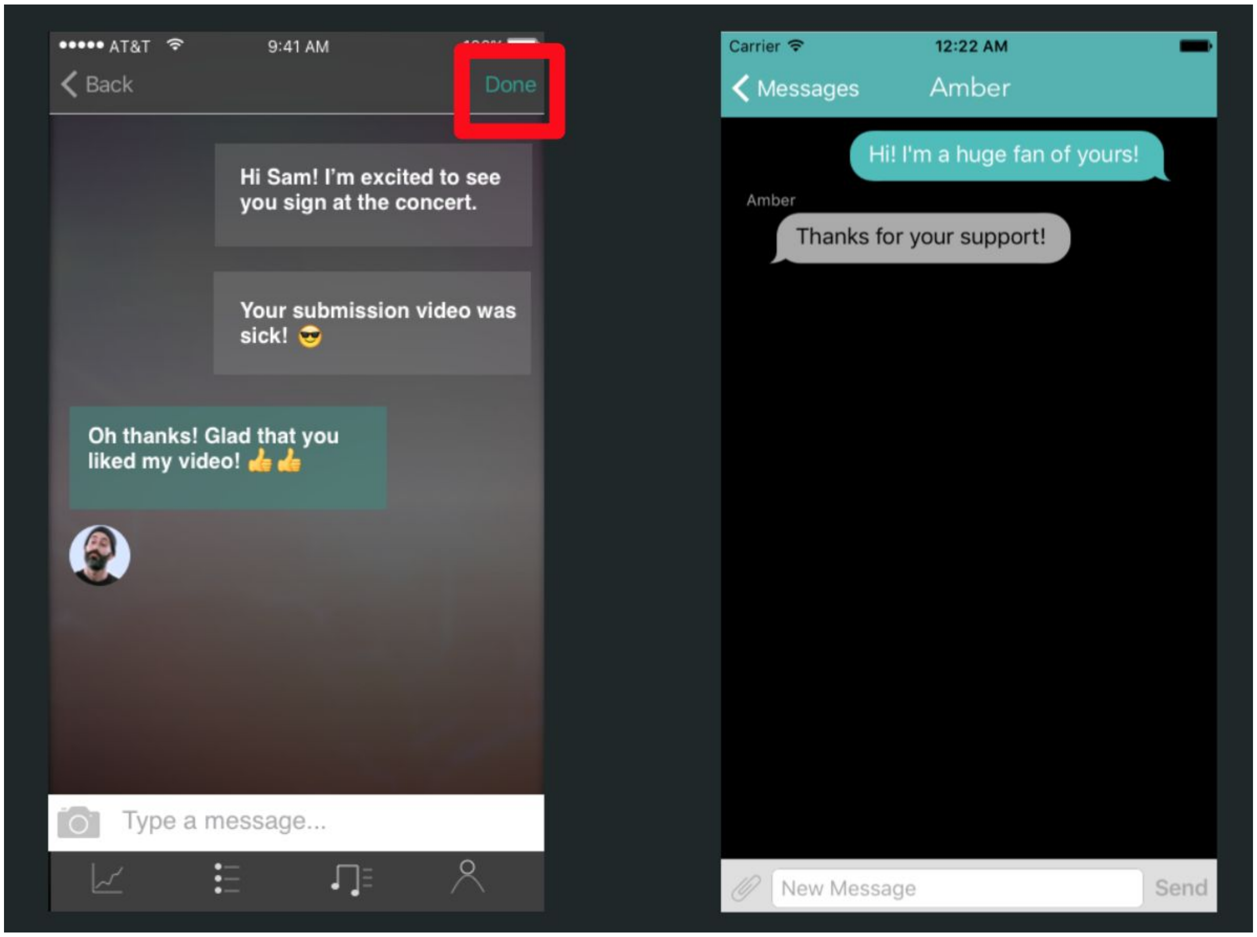

AT&T 9:41 AM 100%
Back
Done
Active Contests
THU NOV 17 8 hrs left Kanye West 9:00 pm SAP Center, San Jose, CA
WED NOV 17 8 hrs left Rihanna 9:00 pm Golden 1 Center, Sacramento, CA
view more
Today
SAT NOV 20 8 hrs left Chance 9:00 pm Save Mart Center, Fresno, CA
view more
This week
FRI OCT 6 8 hrs left Beyonce 9:00 pm SAP Center, San Jose, CA
This month
Carrier 11:55 PM
Contests
2 days left
Chance the Rapper
Bill Graham Civic Center - San Francisco, CA
Tues, Dec 12
14 days left
Jason Derulo
Oracle Arena - Oakland, CA
Sat, Dec 31
20 days left
Twenty One Pilots
SAP Center - San Jose, CA
Fri, Feb 10
25 days left
Home
Chat
Me

Figure 11: Violation 4 before/after.

5. H2-3. User control and freedom / Severity 3 / Found by: A, B, D

   In the same screen mentioned above, when the user hits the "Done" button, he/she will be in the user selection screen. This is extremely confusing, as concert going users who decide to end a conversation with an interpreter would more likely want to view the other concerts in the area, or see more details about the upcoming concert. An improvement could be to take the user back to the screen that displays the details of the concert which they voted for.

   Solution: The "Done" button was confusing given that there is also a back button. As a result, we adopted a layout that is commonly used across popular messaging apps like Facebook Messenger. By eliminating the "Done" button, the user simply has to tap the messages back button to return to the list of messages when they are done chatting or if they decided to not send any messages. See Figure 12.



Hi Sam! I'm excited to see you sign at the concert.

Your submission video was sick! 😎

Oh thanks! Glad that you liked my video! 👍👍

Hi! I'm a huge fan of yours!

Amber

Thanks for your support!

Figure 12: Violation 5 before/after.

6. H2-2.Match between system and the real world / Severity 3 / Found by: A, C
   It's a bit confusing why there is a choice to log in as an interpreter versus a concert goer (shouldn't this be permanent for each user based on their hearing impairment or knowledge of ASL?). The app also doesn't make clear whether this is a one-time selection upon registration or a choice the user makes every time he/she logs in, though this may be due to the nature and incomplete implementation of a medium-fi prototype.

   Solution: We decided to keep the interpreter vs concertgoer screen as a one-time selection upon registration because this how the app determines which features are available to the user. This HE was largely invalid because of the incomplete implementation of the medium-fi prototype. Additionally we also removed the back and done buttons to provide a cleaner interface. For purposes of demonstrating the full functionality of the app, our login screen has a button for both the interpreter and concertgoer, so that both flows can be experienced. See Figure 13.

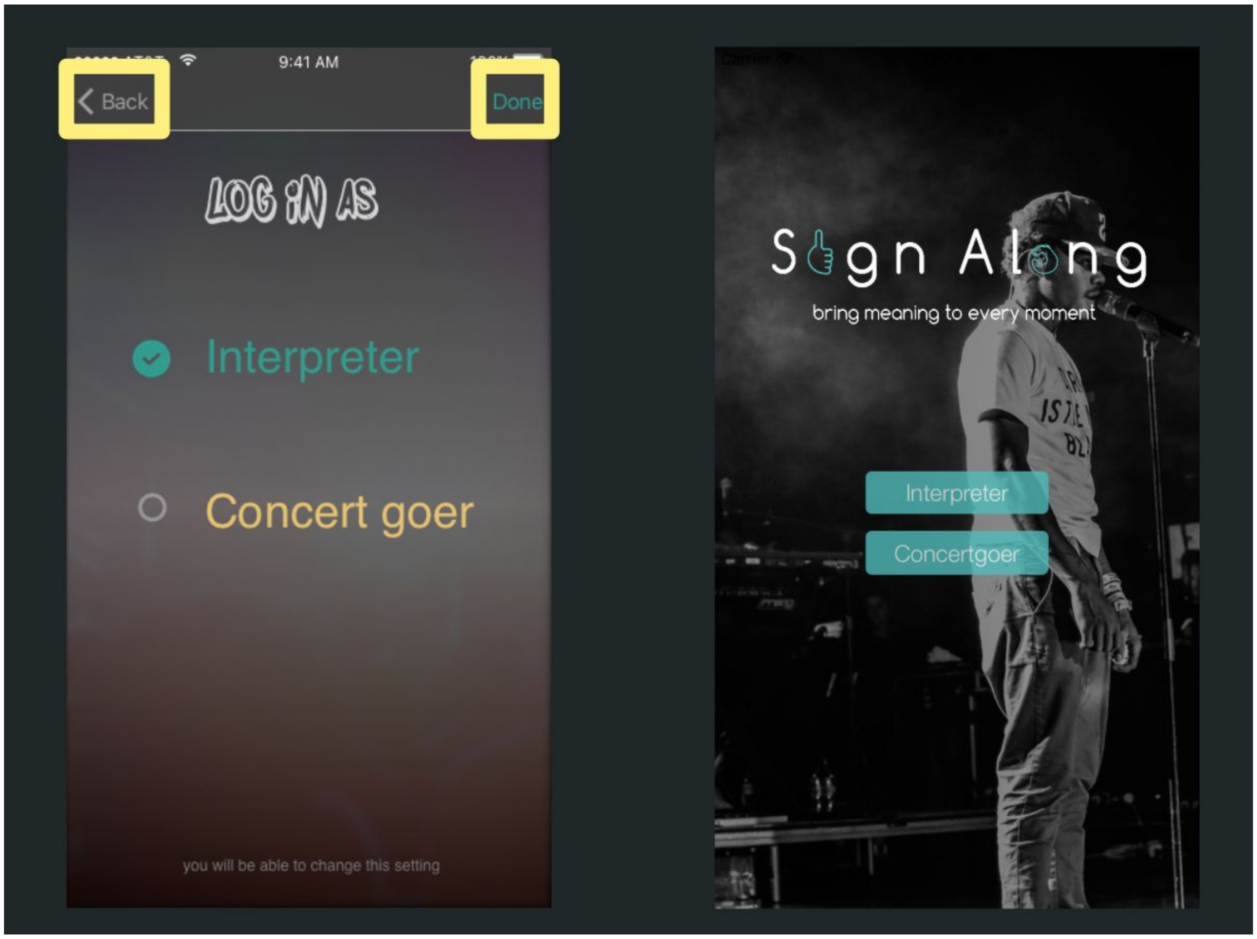


Figure 13: Violation 6 before/after.

7. H2-1. Visibility of System Status / Severity 3 / Found by: A
   When the interpreter is about to record her video, she is given the option to "create" the video or to "upload." It is not clear whether "upload" means to upload an external video from the user's camera roll. Once she clicks "create," she is then taken to a screen to record and save the video; the screen after that allows her to "submit" the video. It is not clear what the difference is here between "upload" and submit." The app should keep the user informed about the difference between the two options, with documentation if necessary.

Solution: Initially, we had a “create” and “upload” option in order to allow interpreters to both record and submit audition videos from within the app and upload videos from their phone. For those recording within the app, we then wanted to allow them to create and save multiple drafts of their submission. However, through HE we learned this is over-complicated the process of submitting videos. In fact after doing more user testing, we found that users are likely to practice off camera multiple times before filming themselves and making a submission. As a result, we decided that users would only be able to upload videos from their camera roll as a submission, simplifying the task flow significantly. See Figure 14 and 15 for details.

8. H2-1. Visibility of system status / Severity 3 / Found by: B, C, D
In the screen when the interpreter-user has just recorded a video of his/her interpretation, there is insufficient system indication as to whether the video has actually been saved – in that the “Record” button is still available for selection, and the “Save” button is the same color as before when no video had been recorded. Create a notification message to inform the user that he/she has finished recording a video. Additionally, there could be a change in color in the buttons to inform the user that the video can now be saved and the record button cannot be selected.

   Solution: Our solution to violation 7 subsequently addressed violation 8 by eliminating the practice mode screen. See Figure 14 and 15 for details.

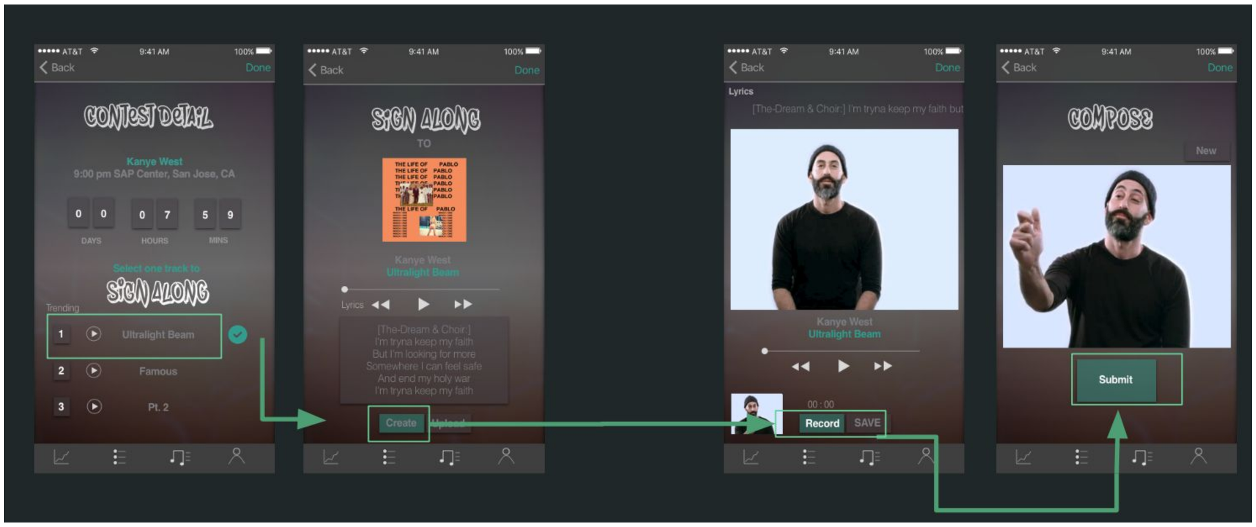


Figure 14: Violation 7 and 8 before.

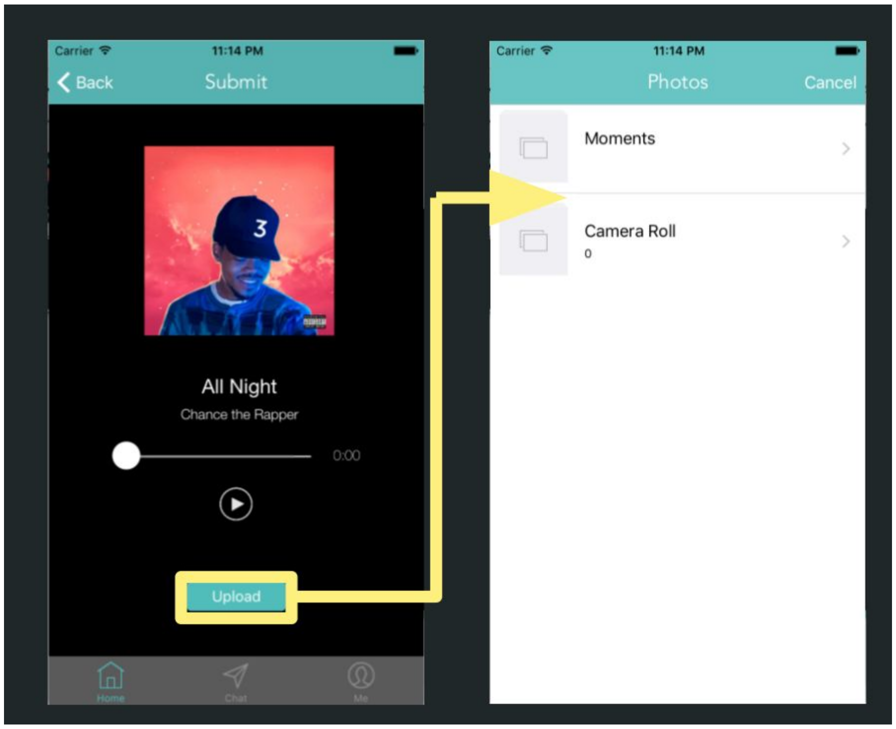


Figure 15: Violation 7 and 8 after.

9. H2-2. Match between system and real world / Severity 3 / Found by: B
   I am confused why you would need to know how many hours there were left until an event that seems so far away (i.e. Nov 20). Intuitively, I would think this timer would only be relevant for events that are really quickly approaching. If it weren't specifying how many hours until the event itself, I wouldn't know what the "hours left" corresponded to. Therefore, in order to eliminate confusion about the "hours until" upcoming events, I would eliminate this metric except on events that are less than 24 hours away.

   Solution: We opted to continue using a timer feature in order for interpreters to quickly tell when submissions are due and how they should prioritize these submissions. However, we realize it doesn't make sense for the timer to be set at the "hour" level for all events, especially those months in the future and showed those at the appropriate level of granularity. See Figure 16.

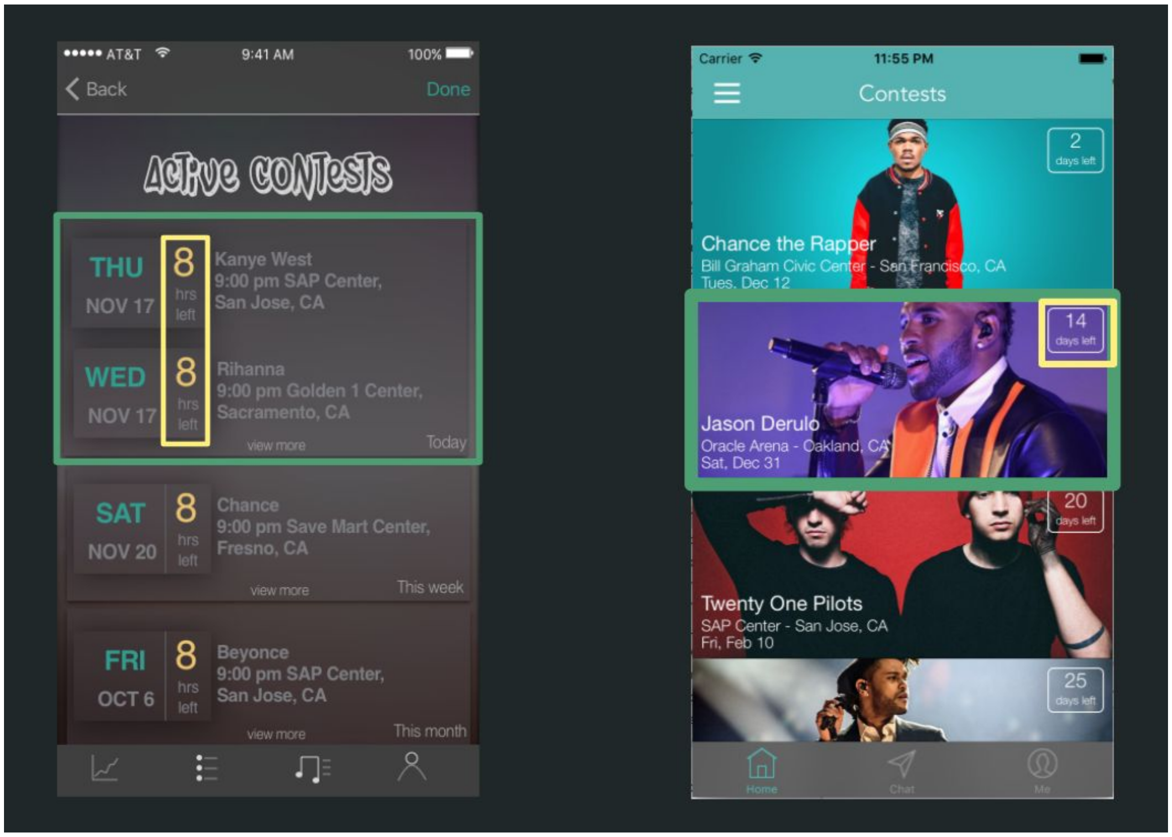


Figure 16: Violation 9 before/after.

Level 4 Violations and related changes

10. H2-10. Help and documentation / Severity 4 / Found by: C, D

    In the screen that shows all concerts close by, it is not apparent that concert-going users have to first vote for their favorite interpreters before they can purchase early bird tickets. Additionally, the medium-fi prototype is confusing because the concert is today – and presumably, early bird tickets would have been released way earlier. Better documentation can be provided to explain to the concert going-user that they have to first vote on their favorite interpreters before they can get early bird tickets.

    Solution: Initially we wanted to incentivize concertgoers to vote for their favorite interpreters by letting them buy tickets in the app. This HE highlighted that this flow is not intuitive; concertgoers are primarily interested in voting for the best interpreter. Additionally, it doesn't make sense for our app to support ticket purchasing, since that is already independently handled by the event organizers through either third-party ticketing websites, or the venues themselves. As a result we decided to remove the vote and buy buttons so that concertgoers simply see the list of all upcoming concerts in the area. If they are interested in a particular concert, they can then tap on the screen to learn more details, see video submissions, and vote. See Figure 17.

11. H2-4: Consistency and Standards / Severity 4 / Found by: A, B, C

When the concert-goer sees the list of upcoming concerts, some concerts have a “vote” button next to them, and some have a “buy” button. This part of the interface is inconsistent. It is unclear why each concert doesn’t have both buttons, since the user should presumably have the option to both vote and buy for each concert. Buttons for both “vote” and “buy” should be added to each concert to make the user’s available actions more consistent.

Solution: In addressing violation 10, we have also addressed violation 11.

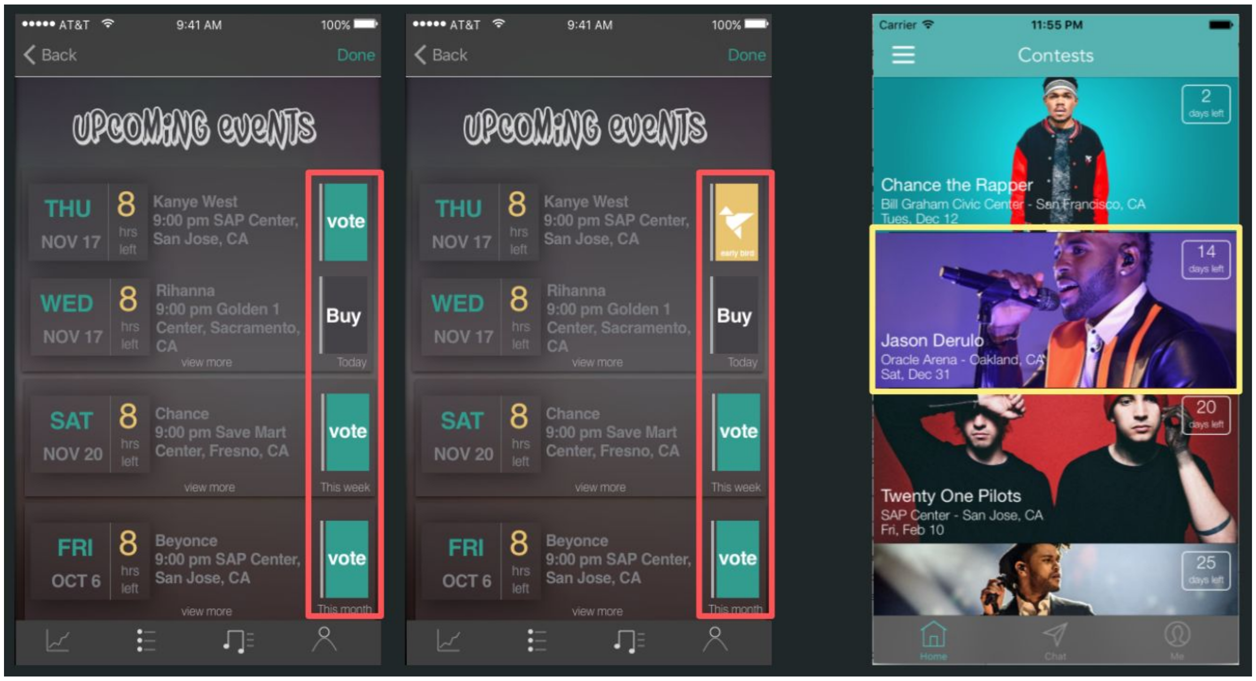

UPCOMING EVENTS
THU NOV 17 8 hrs left Kanye West 9:00 pm SAP Center, San Jose, CA vote
WED NOV 17 8 hrs left Rihanna 9:00 pm Golden 1 Center, Sacramento, CA Buy Today
SAT NOV 20 8 hrs left Chance 9:00 pm Save Mart Center, Fresno, CA vote This week
FRI OCT 6 8 hrs left Beyonce 9:00 pm SAP Center, San Jose, CA vote This month
Contests
Chance the Rapper Bill Graham Civic Center - San Francisco, CA Tues, Dec 12 2 days left
Jason Derulo Oracle Arena - Oakland, CA Sat, Dec 31 14 days left
Twenty One Pilots SAP Center - San Jose, CA Fri, Feb 10 20 days left
25 days left
Home
Chat
Me

Figure 17: Violation 10 and 11 before/after.

Prototype Implementation

Hi-Fi Prototype

Tools

The high-fidelity prototype was implemented as an iOS app in Swift (2.2) using Xcode. One of the main draws of Xcode was the Interface Builder, which allowed us to graphically lay out the screens in the Storyboard and design the navigation flow of the app through the various view controllers. Additionally, Xcode provides a simulator that allows us to test the app on specific iPhone screens, so we could incrementally test features and track how images/text would actually appear on the screen. Another benefit of using Swift is the availability of third-party libraries and frameworks, and Cocoapods, a dependency manager for Swift that really streamlines the process of incorporating frameworks into projects. For our app, we used Cocoapods and the JSQMessagesViewController module to help implement the chat/messaging feature of our app.

However, there were some drawbacks with using these tools. For instance, since Swift is fairly new language, the updates to Swift often have significant changes. Although the latest update to Swift is Swift 3, we decided to develop in Swift 2 because there are considerably more resources, tutorials, and troubleshooting available for Swift 2 online (Swift 3 was just released in September of this year). Since the changes between 2 and 3 are pretty big, and a large swath of the API was renamed in Swift 3, it was a bit challenging to sift through resources that varied between the two updates. Moreover, we experienced irksome Xcode/iOS compatibility issues, since iOS 10 requires Xcode 8, which is tailored to Swift 3.

Wizard of Oz

We used a Wizard of Oz technique for the voting feature of our app. The voting feature gives off the appearance to the user that he or she can vote for or against a submission (with the “x” and check buttons). However, when a user votes for/against a submission, we are not actually tracking their votes (the button taps have no backend effect beyond shifting to the next submission).

Hard-coded Data

Hard-coded features include the set of concerts/contests, tracklist metadata, chat messages, track mp3s, and interpreter information. Additionally, we hard-coded the data for the video submissions. Instead of having actual video audition submissions, we procured videos from Youtube of people performing ASL interpretations of songs.

Limitations

Limitations include that real-time video uploads will not be displayed with the submissions. New messages sent by a user are not saved. Votes for each contest are not tracked so contest winners are not programmatically determined.

Future Work

For future iterations, we plan to eliminate all of the hard-coded features by pulling concert, artist, and track information from a music library API (i.e. Spotify, BandsInTown, Apple Music), allowing contest winners to be determined programmatically, and saving all messages sent by users. To accomplish this, we will need to build a robust backend service for the app for a database/hosting video and audio data. There are many limitations to using Swift's CoreData feature, and unfortunately Facebook shutdown Parse, a really good BaaS provider) early this year. Instead, we will probably look to either using Firebase or AWS Mobile Hub.

Additionally, we are interested in building up an interface/platform that would allow concert event organizers to open up new contests on the app, view submissions, and directly contact/communicate with the winning interpreter. The ideal goal is to entirely streamline the process, so that a deaf concertgoer could put in a request for an interpreter for a concert, and the concert organizer could receive this request and open up a contest for that event.

Summary

Through our needfinding, we discovered that two large problems in the inclusion space were a lack of interpreters and subpar live event experiences for deaf people. When we saw a video of an interpreter signing a performance of "Rap God" by Eminem, we realized that more of these performances could solve both issues. This idea manifested in a crowdsource platform where interpreters could use their ASL fluency to sign along their favorite artists and concert-goers could vote for their favorite interpreter for an event. Layering on this, concert-goers can also communicate and vote on their favorite interpreters. While still keeping the idea the same, through our multiple prototypes, we refined our user interface and design to allow the simplest possible implementation of Sign Along. Our ultimate vision is to empower deaf people in having meaningful and memorable concert experiences by streamlining the process of matching the best interpreters to events. This would require us to coordination with artists and concert venues, but with further refinement, we can eventually develop a fully functional solution to two majors problems in the inclusion space.