CS147 - Mixed Reality Studio

Final Report

Hi-Fi Prototype

Team

Abdallah A.

Ahmed S.

Ammar A.

Alejandrina G. R.

Value Proposition

Polytone aims to convey the tone, volume, and other aspects of verbal communication through text.

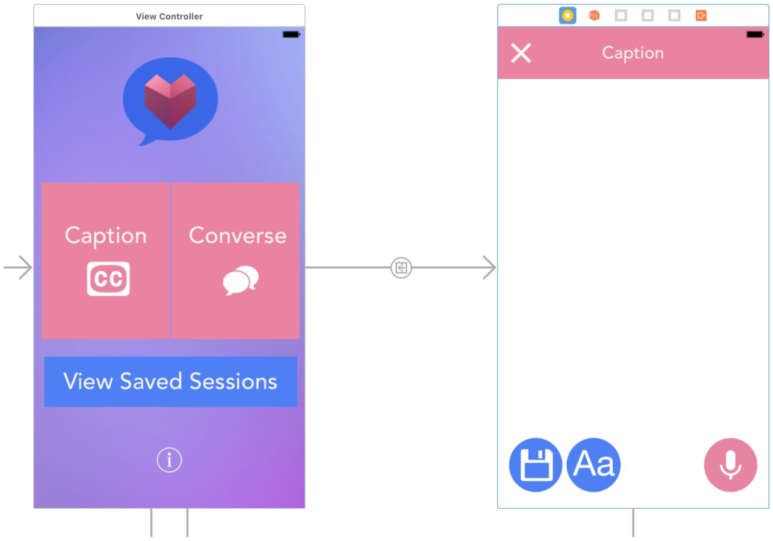
Problem and Solution Overview

People who are deaf or hard of hearing often mainly use text as their way of obtaining information. The issue is that the tone, volume, and emotions implicit in speech are not at all conveyed through text and captions. These are vital aspects of verbal communication that greatly impact the way we obtain and process information. Expressing these aspects of verbal communication through the manipulation of the size and color of text will greatly benefit those who rely on text and captions. Furthermore, it will give everyone a way to accurately express volume, tone, and emotion through text, effectively minimizing miscommunication.

Tasks & Final Interface Scenarios

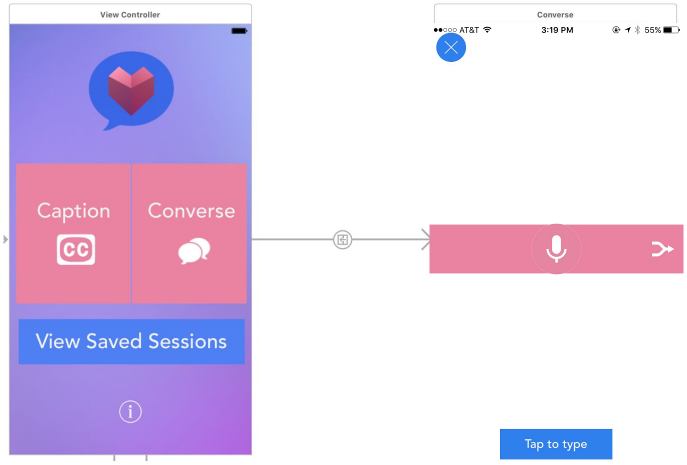
Task #1: Lecture Captioning (Simple)

As a deaf student, I want to be able to visually understand tone, style, and volume of a lecture. This task is to live caption a lecture with emotion. All what the user has to do is clicking few buttons to reach the screen that starts captioning with style. While this is typically for lectures, it could be as well used in meetings or any scenario in which a user simply needs to caption speech.

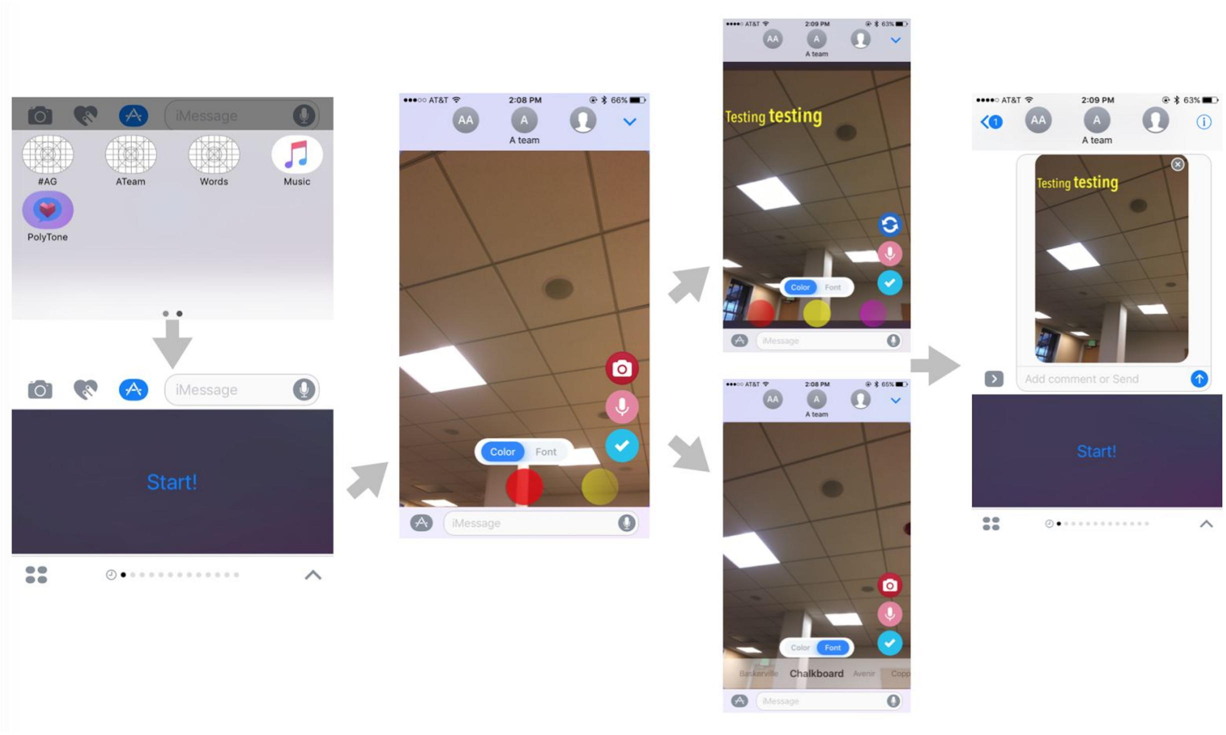


Task #2: 1-on-1 Conversation (Moderate)

To have a 1-on-1 conversation, then using split screen is the most convenient way rather than having one screen like Lecture Captioning. We chose this task to have a 1-on-1 conversation with someone using emotion live captioning. This task was a more moderate task for it involved more steps for successful communication with another individual. Conversation can be done with keyboard as well.

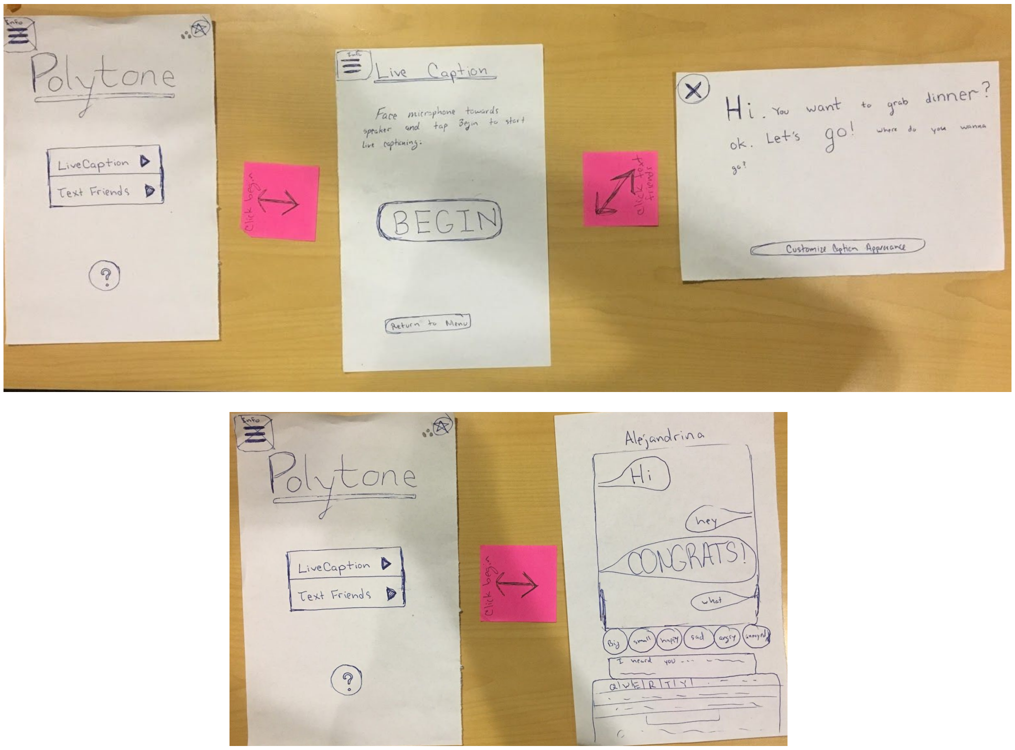


Task #3: Message a friend (iMessage Extension) (Complex)
To communicate with a friend while fully displaying emotion only through text. Although the objective here is simple, it is a very complex task for it involves the added complications of the variety of aspects of verbal communication and the attempt to convert them to a written format. iMessage extension allows users to stylize the text that they want to send to their friends. The idea is capturing a moment and giving the text with their voice, and within the extension they add the style. And when they send it to friends it gets sent as an image to so that the other person can see more than just text.

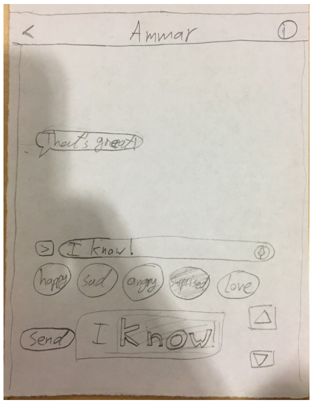


Design Evolution

The initial low fi main screen included only live captioning and an option to open messages extension from the app. The design also included a “hamburger” menu icon, for a menu we were not even sure about. Info button was located at the bottom which was removed on medium fi prototype. Initial low fi prototype design wasn’t well organized as main tasks weren’t finalized yet.

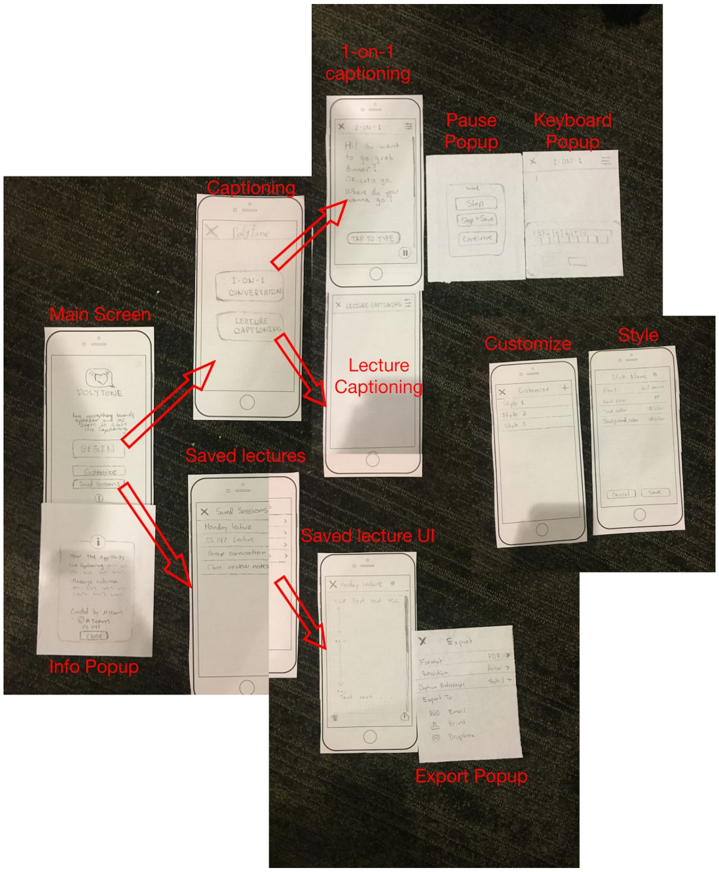

Info
Polytone
LiveCaption
Text Friends
?
click begin
Live Caption
Face microphone towards speaker and tap Begin to start live captioning.
BEGIN
Return to Menu
Hi. You want to grab dinner?
ok. Let's go! where do you wanna go?
Customize Option Appearance
Alejandrina
Hi
hey
CONGRATS!
what
Big small happy sad angry annoyed
I want you ...

Initial low fi design + flow



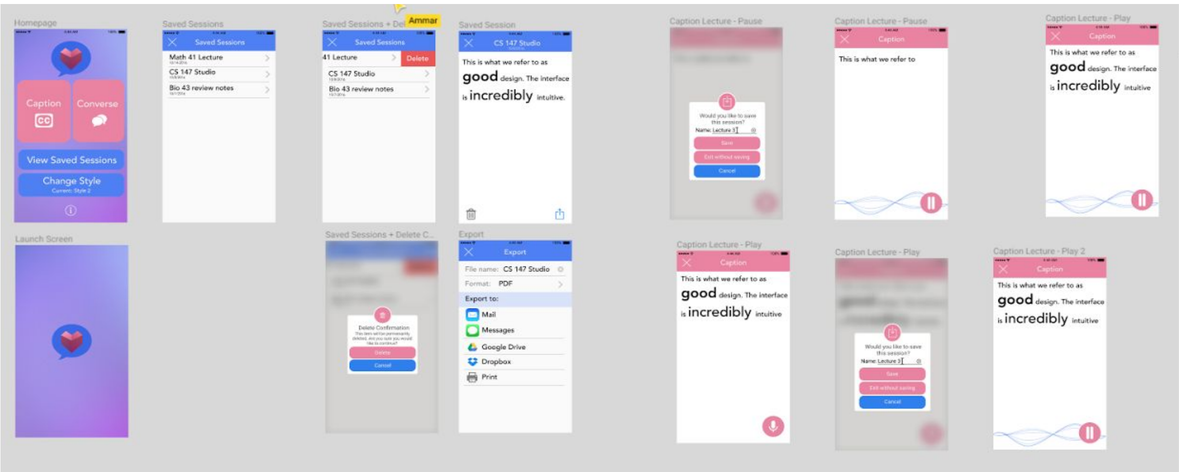
The revised low fi prototype design added a 1-on-1 conversation option which was included on the main screen. It also added the “Change Style” option and “Saved Sessions” option. The revised lo fi design had a more logical hierarchy and clean design compared to the initial low fi design.

The main app was improved on the revised low fi prototype but the iMessage extension was still lacking thought and design (image on left). We were still not sure about how the iMessage extension was going to work and we were not able to design it well without the specifics of how the user might actually use it. This lack of specifics on iMessage extension were carried over to the medium fi prototype design and there wasn't much improvement.



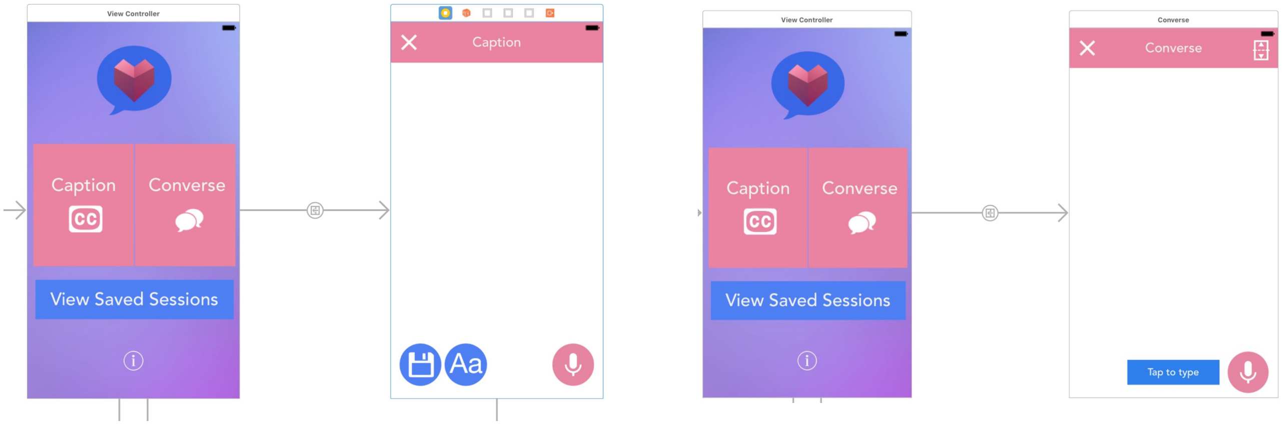
The revised low fi prototype (image on the left) had a better logical flow and arrangement. The revised low fi design was more detailed than the initial one. There were some key points that helped us transition from the revised low fi prototype to the medium prototype after we had users test the low fi prototype. First, users believed there should be an evident and more accessible pause and stop button when live captioning a lecture. This was a clear flaw on our low fi prototype. From fixing that flaw we were able to identify other similar flaws on the medium fi for simple actions of record/start/ stop/save/exit.

While most found the interface to be very clean, clear, and somewhat easy to navigate, we had an unnecessary screen after users tapped the "Begin" button on the main screen which lead to two different captioning options. It made it hard for users to navigate. For the medium fi prototype we improved the design by removing redundant and unnecessary elements on the screens flow and hierarchy. Most of the design on the revised low fi prototype was carried over to the medium-fi prototype.

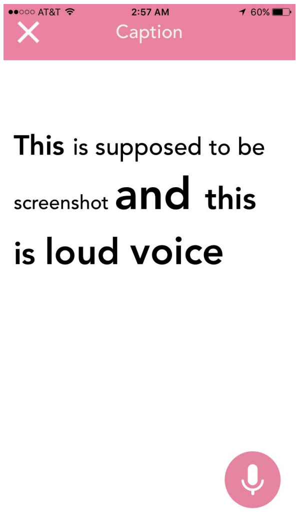


Medium-fi prototype

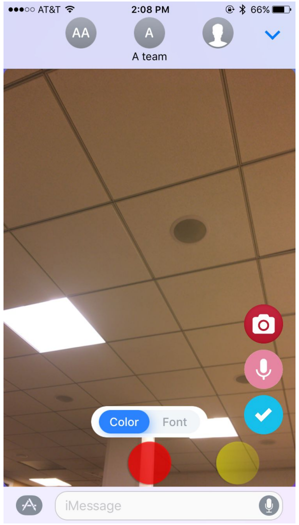
For the medium fi prototype design It wasn’t clear what changing the style in the application did on the main screen. For the high fi prototype design we removed the “Change Style” button on the main screen to within a recording session to make it clear that it changes the font for that particular session.



High fi prototype design



This is supposed to be screenshot and this is loud voice



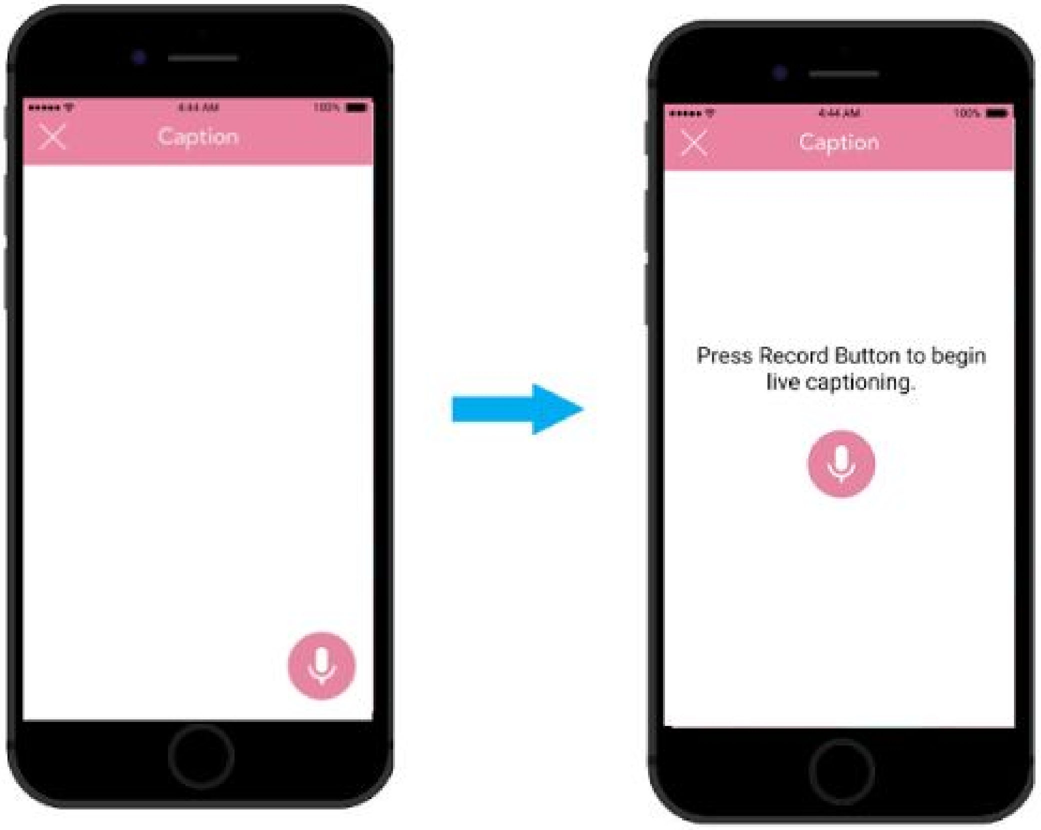
The biggest design improvement from the medium fi prototype to the high fi prototype was the iMessage extension design. Even in the medium fi design we were not sure about the final design until we actually started implementing it and improved while testing it.

Major Usability Problems

Severity 3 Problems

1. [H2-1 Visibility of System Status] [H2-3 User control and freedom] [H2-4 Consistency and Standards] [Severity 3] [Found by A, B, C]. When a user clicks on the "Caption" button, the interface directly transitions a user to the voice recognition portion of the app. This does not seem intuitive; a user would likely expect to click on a "mic" button before being voice recorded. If the user clicked this by a mistake then they will then immediately click the back button but have to go through an extended dialogue to back out. Instead, the app should not start recording until the user specifies. Ask user to click a "mic" button to stop a recording.

Fix and rationale: It was not immediately clear to the users that moving to the caption screen would automatically start the recording and transcription. Thus, we added an intermediate screen before the empty caption screen to prompt the user to press a button before beginning the recording session.

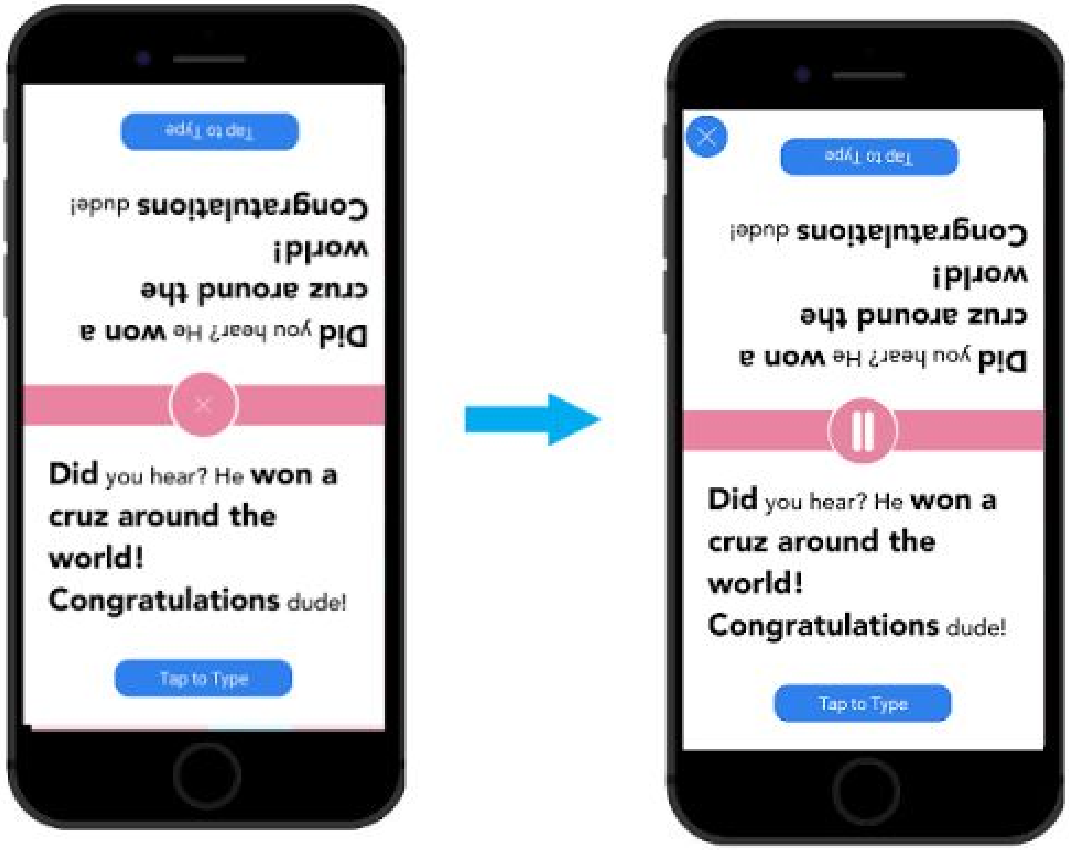


Press Record Button to begin live captioning.

2. [H2-5 Error Prevention] [Severity 3] [Found by B]. In the "Converse" flow, the x button in the middle of the screen is in a location that would suggest central functionality. A user might accidentally click on this button when looking at the other person's message, or might not

understand that this button is actually an exit button due to its location on the screen. Moving this exit button to a corner would deprioritize it in the flow.

Fix and rationale: As suggested, we removed the exit button from the middle of the screen so as not to mislead users. We replaced it with a record/pause button and moved the exit button to elsewhere on the screen.

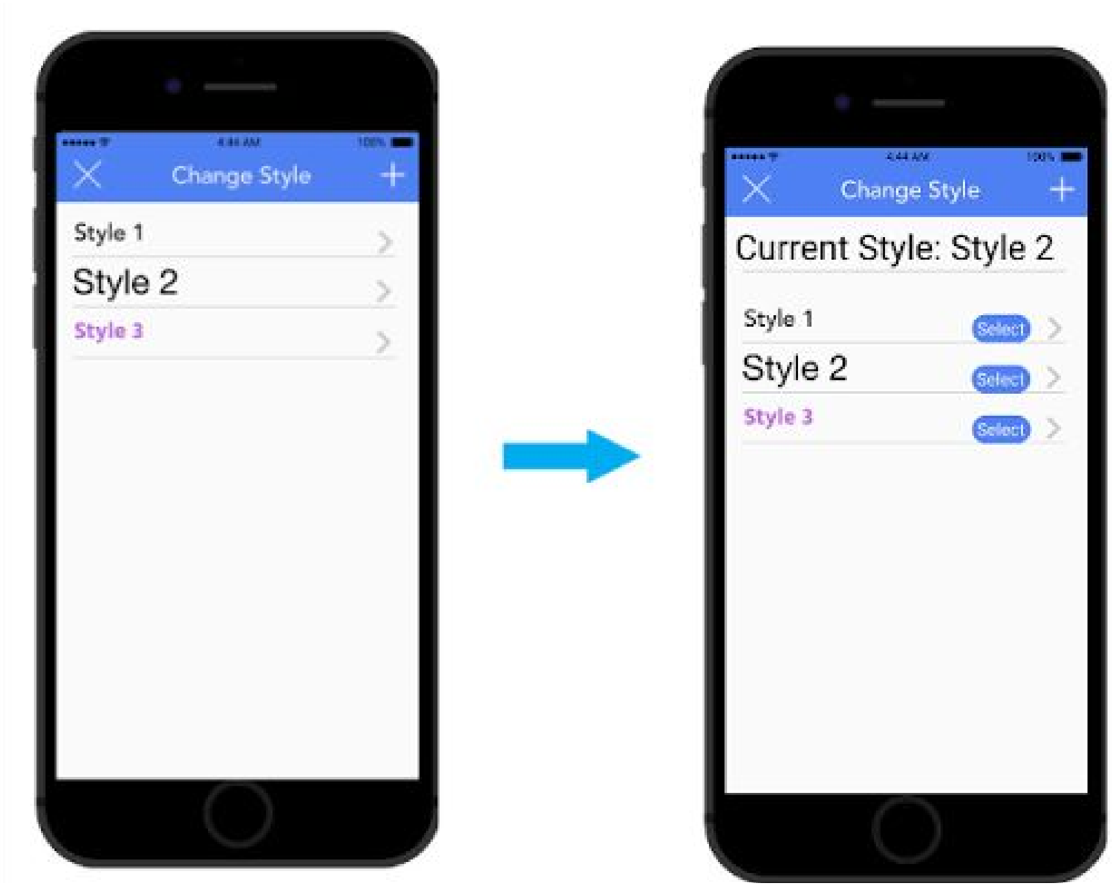


Did you hear? He won a cruz around the world! Congratulations dude!

Did you hear? He won a cruz around the world! Congratulations dude!

3. [H2-7 Flexibility and efficiency of use] [Severity 3] [Found by A]. The process for changing which style is currently selected seems complicated and confusing; I couldn't figure out how to do it. On the "Change Style" screen, the current style should be clearly marked (I see that "Style 2" is in larger font, but it's unclear whether it's large because it's selected or large because the style has a larger font size) and there should be a quick one-tap method of selecting a different style.

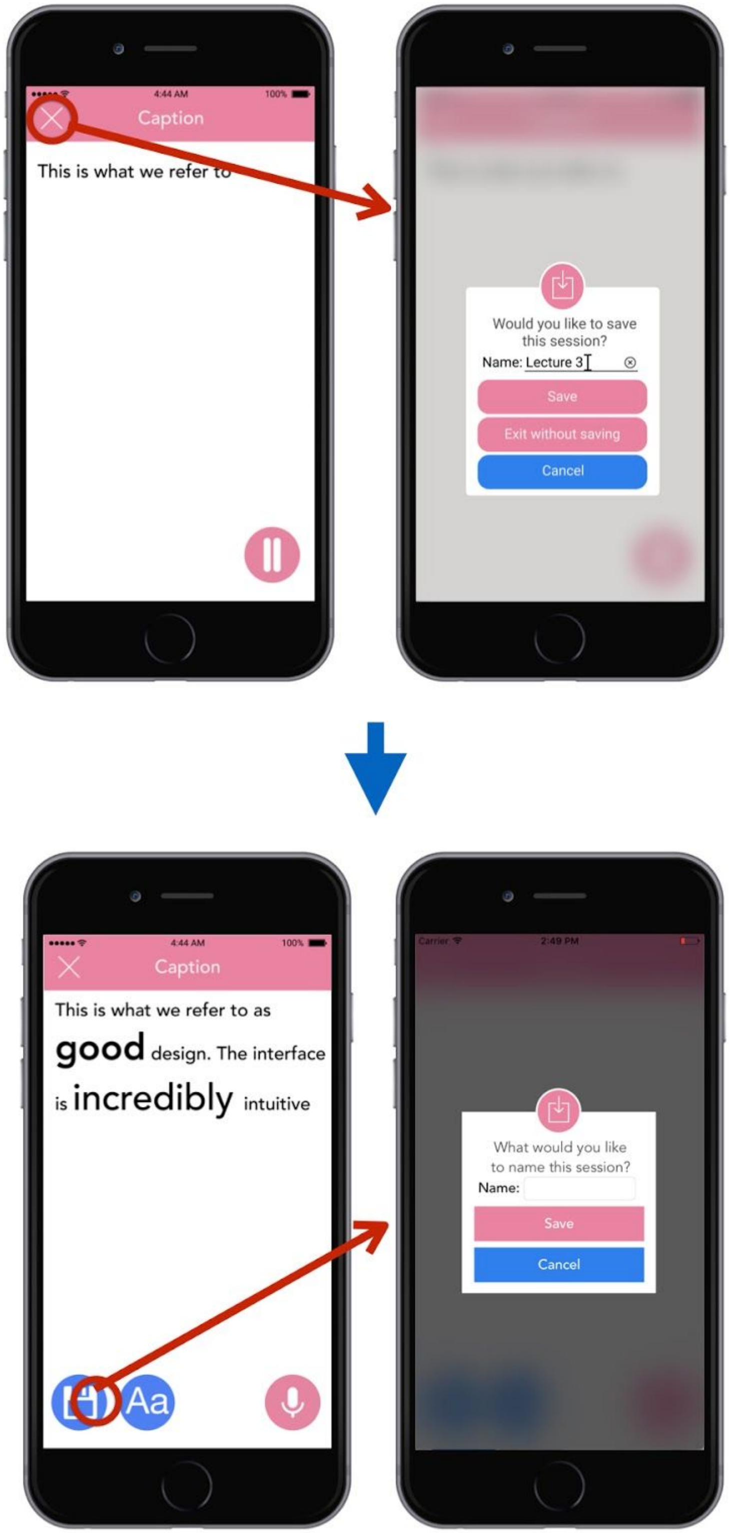
Fix and rationale: To use the style customization and selection processes, we added a bar at the top informing the user of the current selected style. Furthermore, we added quick select buttons to immediately change to another style. This made the process less tedious and more understandable (instead of having to open up the customization settings of a style before selecting it).



Severity 4 Problems

1. [H2-1 Visibility of System Status] [H2-3 User Control and Freedom] [Severity 4] [Found by A, C] . On the “Caption” screen, it is unclear whether the information being recorded is being autosaved continuously, or whether the user must take some action to manually save the information. If it is the latter, then there is a problem; there is no icon visible that may allow the user to save the content. Selecting the X icon in the upper left corner prompts the user to save their content before exiting, but this interaction is non-intuitive, and the user may be afraid to select the icon, since exiting a screen without saving content usually means that the content will be lost. There should be a “Save” icon prominently displayed on this screen.

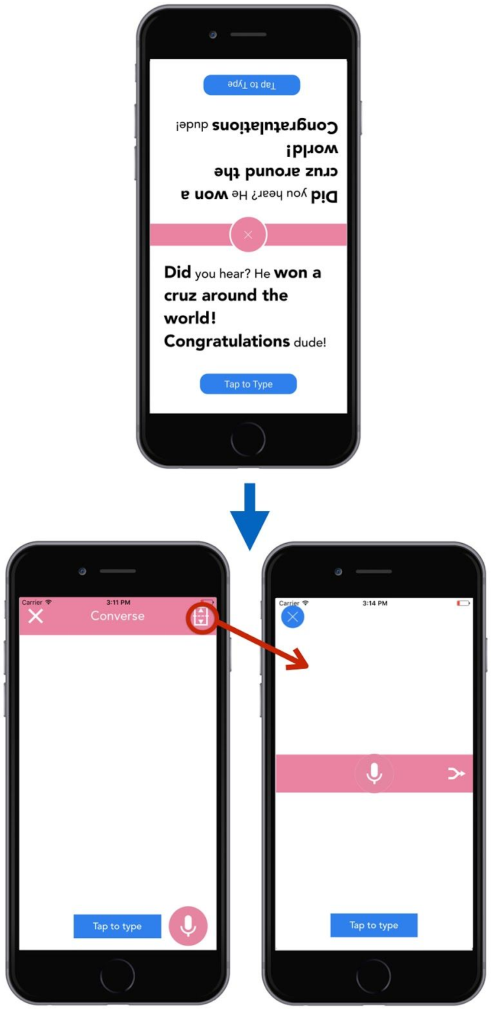
Fix and rationale: The user really had no way of knowing that their recorded content could be saved and named without first attempting to exit. As suggested, we added a prominent save button to allow users to save their content while recording. If the users do not save, they are still prompted to save upon exiting.


4:44 AM
100%
Caption
This is what we refer to
Would you like to save this session?
Name: Lecture 3
Save
Exit without saving
Cancel
4:44 AM
100%
Caption
This is what we refer to as good design. The interface is incredibly intuitive
Aa
Carrier
2:49 PM
What would you like to name this session?
Name:
Save
Cancel

Problem 1: Users were not able to save their recording until attempting to exit. A save button was added to the record screen to allow the user to save at any moment

2. [H2-7 Flexibility and efficiency of use] [H2-8 Aesthetic and minimalist design] [Severity 4] [Found by A, C]. The Converse split-screen does not seem to be a very efficient way of conducting a two-way conversation under many circumstances. The only application where this makes sense is if the phone is sitting on a flat surface and the two users are sitting across from each other. Another likely scenario is that the phone is being passed back and forth between people, in which case they would need to constantly rotate the phone 180 degrees between messages. There should be a way to switch between split-screen and one display depending on the user's current needs.

Fix and rationale: The main reason for creating the converse screen is to provide a "tap to type" option which can enable heard-of-hearing or deaf people who are also mute to type responses during a conversation for the other person to see. However, it is true that conversations don't necessarily take place face-to-face with two people sitting facing each other. We make a single-screen display for conversing the main option, with "tap to type" still available, and provide the option of toggling a split-screen display in the case that the two people having the conversation are indeed facing each other

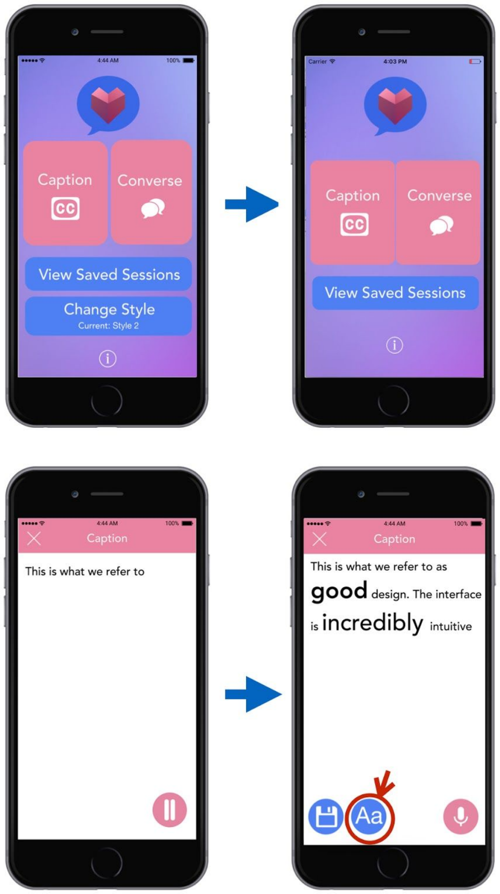


Did you hear? He won a cruz around the world! Congratulations dude!

Did you hear? He won a cruz around the world! Congratulations dude!

3. [H2-8 Aesthetic and Minimalist Design] [Severity 4] [Found by A, B]. It is not immediately clear what the “Change Style” functionality accomplishes. Does it change the general appearance of the app? The appearance of text in captions? The appearance of text in conversations? More than that, it is unclear why changing style in any of these areas would be particularly important. It might make sense to remove the “Change Style” button, or to make the functionality of changing text style more central to app purpose.

Fix and rationale: The “Change Style” flow had a much more central purpose in our original app design, as it was meant to allow users to configure different style options that map to a certain emotion or tone when speech is being converted to text. However, since our final design only includes mapping volume to text, and neither emotion nor tone, such a variety of styling options is no longer available, and instead the only variety available is changing the base font size since the app automatically changes font size based on volume. Since our “Change Style” option simply ended up being a selection of font rather than a design of a whole style configuration, and since it applies only to the text that is being recorded in the “Caption” flow, we decided to change “style” to “font” and removing it from the main screen to place it directly in the “Caption” screen instead, to indicate that it applies only to the text that is currently being captioned (image below)

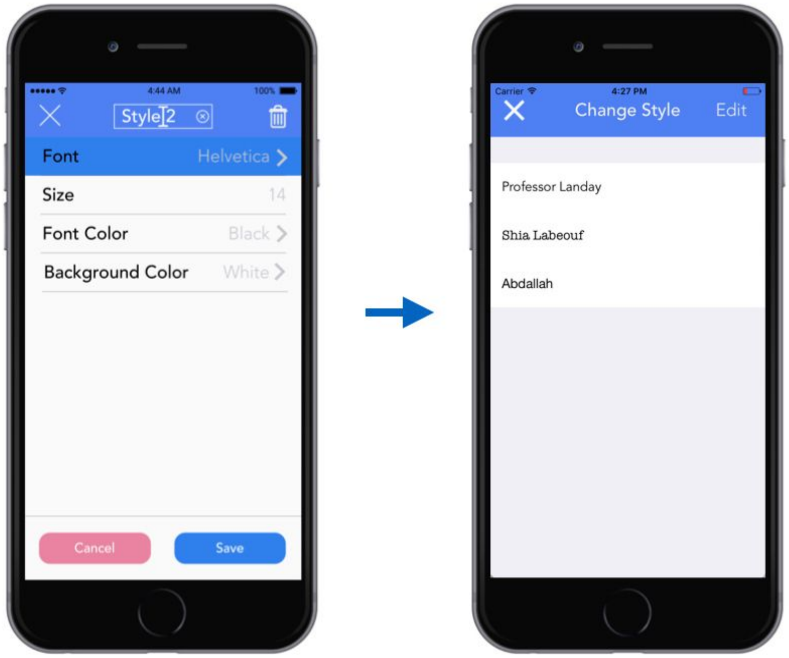


This is what we refer to

This is what we refer to as good design. The interface is incredibly intuitive

4. [H2-8 Aesthetic and Minimalist Design] [Severity 4] [Found by B]. It is unclear what the real functional difference between the “Caption” and “Converse” flows are. It seems that caption is audio-based and one sided, whereas converse is written and two sided. This distinction, though, seems semi-arbitrary--why couldn’t converse use audio rather than text?--and it’s not immediately clear why both flows are necessary. It might make sense to mesh both flows, or differentiate them more completely.
Rationale: The converse option is meant to help people who are deaf or hard-of-hearing and who are also mute or use sign language as their main method of communication to respond and engage during a conversation using text. This is opposed to the “Caption” option, which is to be used in cases where the user is only trying to receive captions and not reply back.

5. [H2-6 Recognition Rather than Recall] [Severity 4] [Found by B]. In the “Converse” flow, if a user is typing, how might that user determine size and font of text? This seems like a potentially complicated flow, but this functionality is suggested in your prototype. It might make sense to remove dynamic sizing from supported functionality in the “Converse” section.
Fix and rationale: As in the 3rd violation, this is fixed by moving the “change style” function to the recording screen itself. In addition, the “change style” now only changes the base font, and does not change the font based on every mapping of volume or tone, so users cannot change the size of the font either.



Style2

Font Helvetica
Size 14
Font Color Black
Background Color White

Change Style

Professor Landay
Shia Labeouf
Abdallah

Prototype Implementation

Tools

Our final prototype runs on iOS devices, so to implement it we had to use xcode.

Design-wise, using xcode made it fairly simple to create a UI that matches our med-fi design. In addition to this, using Swift helped us in utilizing Apple libraries and kits for speech recognition and recording instead of having to search for 3rd party libraries, which sometimes could be hard to deal with. Also, with iOS 10, Apple introduced iMessage applications, so we took advantage of this to do our third task instead of building a separate messaging service in our app.

On the other hand, we faced few roadblocks because we used Swift and xcode. For example, to resize text depending on volume, we had to code a considerable amount of code instead of having a simple code that does resizing. Also, some limitations in the libraries used led to having sizing show only when you pause a session while when the app is doing live captioning, it just shows plain text. Another annoying thing about iOS development is deploying the app to ipa format, and we had few issues when we were trying to do it.

Wizard of Oz techniques

In our final prototype we were able to implement all what we initially intended to do, so we did not need to use any Wizard of Oz techniques.

Hard-coded data

Due to time limitations, we had to hardcode small parts of data such as the different font styles in "Caption" window, and the choices that the user has for font and color in the iMessage extension.

Possible future additions

Our app in its current form changes font size depending on volume, but it does not infer any other information from voice. Although inferring things like emotions through voice might be too hard using current technology, we see this as our end goal. In addition, there are small additions that could be done in different parts of the app. For example, in iMessage, we could let users send the photo as a sticker, and reuse it multiple times. Moreover, the way we size text might not be the ideal way, but we have not done enough tests on different ways to see what works best.

Summary

We believe in a future where people have equal opportunities regardless of their disability. In addition, with technology and smartphones becoming more widespread, we think that there is great potential to help disabled people. Polytone focuses on helping people who are deaf or hard-of-hearing to obtain information about the world in a better way, and opens new doors for people to express their emotions through text. More so than ever, Polytone enables speech recognition to be more meaningful than just words.

Developed through many weeks of need-finding, testing, and prototyping, our final product is currently a high-fidelity prototype that allows users to perform the three main tasks of captioning, conversing and stylizing messages. We hope to continue our exploration of Polytone in the future and keep iterating to improve our application that will include features beyond our main tasks.