



University of Washington iSchool

1

Encaption: Real-Time Captioning Design Specification

Team HeisenBARK: User-Sensitive Inclusive Design

Kevin Le

Ryan Rinaldi

Brett Yamada

Ya-Chu Chiu



Table of Contents.....

Introduction | 3

About Our Expert User | 4

Designing as a Phone App | 5

Accounts and Signing Up/Logging In to Encaption | 6

Minimalist Homepage | 8

Captioning Dialogue | 10

Homepage/Captioning Rationale | 13

The Menu Bar | 14

Library | 15

Settings | 18



Introduction

The project we have designed is an application called Encaption that is meant to transcribe voice to text in real-time, with a variety of users in mind. One such user Laura, who is a half-deaf and half-blind, has been working with us and testing our design throughout the ten week process. She is unique in that her disability is essentially “invisible” - you would probably never guess that she had these issues unless she told you about them, as she requires little to no assistive technology. As a result, many of our design choices have revolved around her needs, specifically regarding accessibility.

Additionally, we have kept a design approach in mind in developing our project: user sensitive inclusive design, which is most concisely put as “designing for as many people as realistically possible”. We have decided to interpret this in such a way that requires us to work and listen particularly close with our expert user, while carefully making corresponding decisions knowing that a more general target population is sensitive to such changes.

There are two main ways that our design functions:

1. *A user can caption dialogue for strictly themselves on their own device.*

We see this scenario as being mainly applicable to Laura. From our first meeting with her, she gave us the example of her chatting with a couple of friends at a loud restaurant but having trouble hearing the conversation. In this case, she would most likely use the application to transcribe her friends’ audio to text so that she may read the dialogue while still being able to converse, nullifying her difficulty in hearing them.

2. *Two users may connect their devices, thus captioning for both parties.*

This obviously would require two instances of our application running, one for each user. We see this feature as being geared toward a more average user population. Say for example two users would like to converse with each other but for whatever reason do not speak the same language (maybe this is a business meeting between two international clients). They may connect their devices and caption the dialogue respectively translated to their own language.

Also available are a plethora of standard features such as the ability to save, send, and review files, etc. that we thought a typical user would find useful. The individual implementation details and rationales with each can be found starting on page five.



About Our Expert User

As facilitators of user sensitive inclusive design, one of our main goals has been to prioritize the needs of the disabled just as much as the needs of the nondisabled. As a result, it has been imperative to our design philosophy that we work particularly close with our expert user, Laura. Laura has been providing us with valuable knowledge about how she navigates her devices, as she is both blind and deaf on her left side. There were a few main requests she had which have been the focus of our design rationale:

1. The design needs to be effortlessly intuitive – Laura does not own a smart phone, so we want the design to be understood from the first experience, regardless of the user's background in technology.
2. Achieving tasks should be quick – In order for the application to be practical, the amount of time it takes for the user to accomplish a desired objective needs to be minimal or else the user is wasting effort or missing information, especially in the case of captioning.
3. Using the device should not draw much attention – Laura made it clear that aesthetics are important to her; she's reluctant to use technology that gives away her disabilities, so we want the design to be as subtle as possible.

Ultimately, all of our choices have come down to these three issues and will be referenced throughout the remainder of this document. We have broken down the design into the major components we have chosen to include and will delve into each, along with their related design choices, and specifically rationalize how they resolve at least one problem in the above list. Let's get started.

Designing as a Phone App

Reiterating, our design approach required us to design for as many people as reasonably possible. Seeing as how many people now own smart phones, both disabled and nondisabled alike, we initially thought the obvious choice was to design the project as a mobile application for a phone as opposed to its own standalone device – it was consistent with our design approach. However, the only problem was that Laura does not own a smart phone, as previously mentioned. For a while we debated whether or not the project should run on its own hardware, and it finally boiled down to this:

We could either design a unique device that would work extremely well for our expert user but maybe not others, or we could design an app on a conventional medium that would work extremely well for others but not maybe not our expert user.

Remembering from the assigned readings that the tester does not always directly benefit from the final design, we bit the bullet and opted for the latter option. Upon hearing this decision, she was understanding and still very supportive when we presented our case.

Despite her not owning a smartphone, we were not worried that this characteristic of hers would negatively impact our design, as she has proven that she is still very much acute with technology (she even has experience as an HCI designer herself). That being said, we still designed with Laura in mind, taking her feedback into close consideration and making decisions as if she *did* own a smart phone. After all, she was our best representative of our intended disabled population, and we did so in order to accommodate her specific needs as well as the needs of other disabled people.

Accounts and Signing Up/Logging In to Encaption

This is the screen that the user will see the very first time they use this app (figure 1). They will also see this screen if they had manually logged out of a previous session; this will make more sense later when we get to “logging out”. There are four main components of this login screen:

- Sign up
- Email and password entries
- Forgot password
- Sign in

Sign Up

If this is the user’s first time using Encaption, he/she will need to select “sign up” to register for an account. On this page (figure 2), there will be four entry fields: name, email, password, and retype password. The user will type their name in “Name” and the email they want to use when they login in “Email”. They will then type in the password they want to use when they login in “Password” retype said password in “Re-type Password” to ensure that they have their desired password. The “Create Account” button will then be pressed, creating the account and taking them to the homepage.

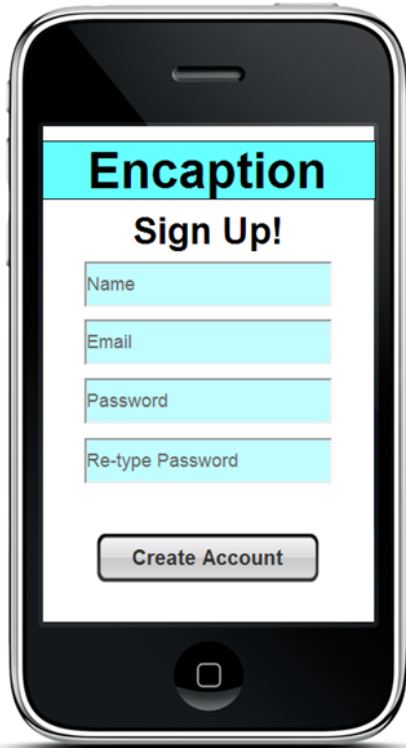


Figure 2: Sign up page

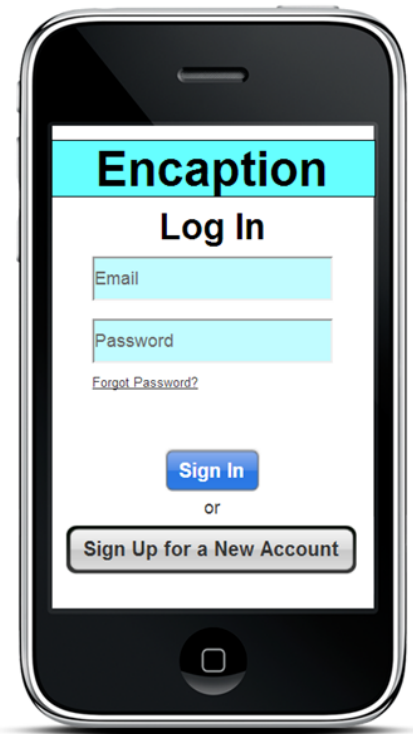


Figure 1: This is the main login screen, which will be seen upon first use of the app or a previous manual logout

Email and Password Entries

Granted the user already has an account, there will be two entry fields on the main login screen: one for “Email” and one for “Password”. The user will enter in the email and password used when signing up for the account. If either entry is empty or if the given email and password do not match an existing account, there will be a pop-up notifying the user, “Incorrect Login” in red (figure 3).

Forgot Password

If the user forgets their password, they can select “Forgot Password?”. This will take them to a new screen (figure 4). The

user will be prompted with "Please enter the email address associated with your account". There will be an input box under that for the user to enter the corresponding email address. The "Send Recovery Email" button will then be selected to take them to a new screen (figure 5). The user will be notified saying, "A password recovery email has been sent to the specified email address" *if* the given email matches an existing account. The password will then be reset to a random string, an email will be sent to that address with the corresponding reset password, and the "Return to Login" button can be hit to be taken back to the original login screen. If the given email address *does not* match any existing account, a "No such email address exists in our records" error will appear similar to figure 3.

Sign In

Given that the entered email and password fields match an existing account, the user will be taken to the homepage and signed in with the corresponding account.



Figure 3: Invalid login notification



Figure 4: Forgot password screen

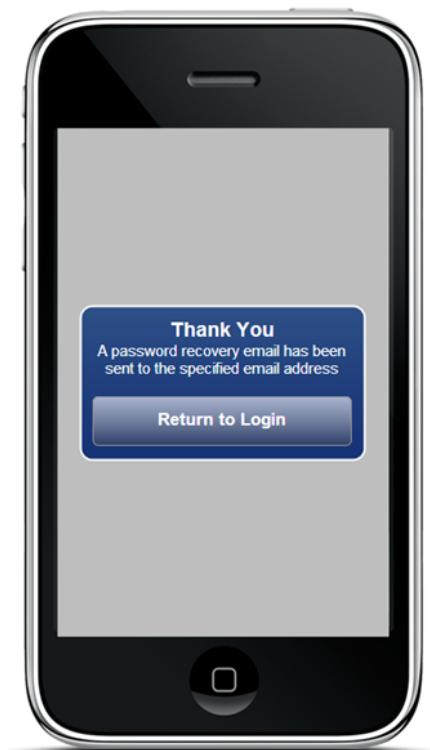


Figure 5: Confirmation notification of a sent recovery email

Minimalist Homepage

- “Caption for me only” or “Caption for me and a friend”
- Translation

Our home page is where the navigation to start recording and captioning a dialogue begins so that the user can begin as quickly as possible. This is the first screen the user sees when the app is opened if they already signed up (figure 6), provided that he/she did not previously log out manually (and would therefore have to log in from the login screen first). Because our expert user seeks simplicity and quickness, we have only the above three features available from the home screen, which are detailed below.

“Caption for me only” or “Connect to a friend”

These options allow the user to select either to caption the dialogue using only one device for themselves or caption on two devices for two users respectively. If the user selects the “Caption for me only” option, the device is to be held by the

one user, but the constraint is that it must be within reasonable distance such that the device’s built-in microphone can detect the desired dialogue to be captioned.

If the user selects the “Connect to a friend” option, then the device will search for other devices in the area using Bluetooth (figure 7). The user will select the other desired speaker’s device (we’ll call this second user the target), and then a confirmation message will be sent to the target’s device. After the target confirms connectivity, the two devices will be connected. The “Connect to a friend” button is now shown as “Connected to [target’s name associated with their account]” to show who the user is connected with.

**Limitation:* Note that device connections are limited to two devices only, meaning that the user’s phone can only be connected with one other target. We have set this limitation because conversation involved in more than two people can greatly increase the complexity of the design. Due to the time constraint, we will not be able to complete conversations

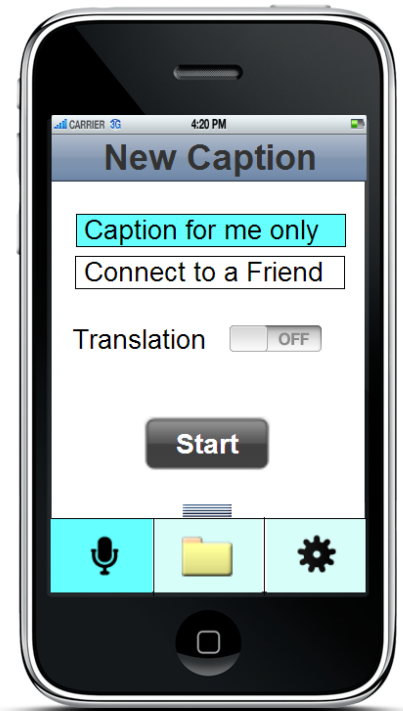


Figure 6: Captioning homepage

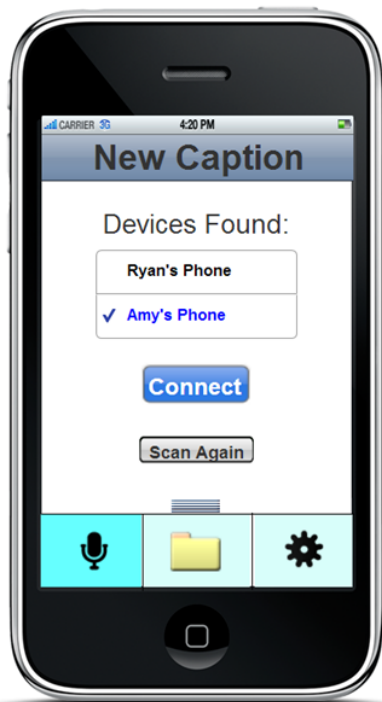


Figure 7: Searching for devices upon selecting “Connect to a friend”

involving more than two people, as this scenario is considered to be out of scope.

Translation

The user has the choice to select “on” or “off” for the translation option; “off” is the default setting. If off, then the user can proceed with captioning regardless of whether they are connected to another device or not by pressing “start”. If they select “on”, then two drop-down menus appear, one for “language #1” and one for “language #2” (figure 8). The option for language #1 will default to the user’s language chosen from the settings but can be changed to something else. Language #2 represents the language to be *translated to* language #1 and must also be selected by the user. We decided to keep these drop down menus hidden when the translation is off because we did not want to make information visible unless it is relevant.

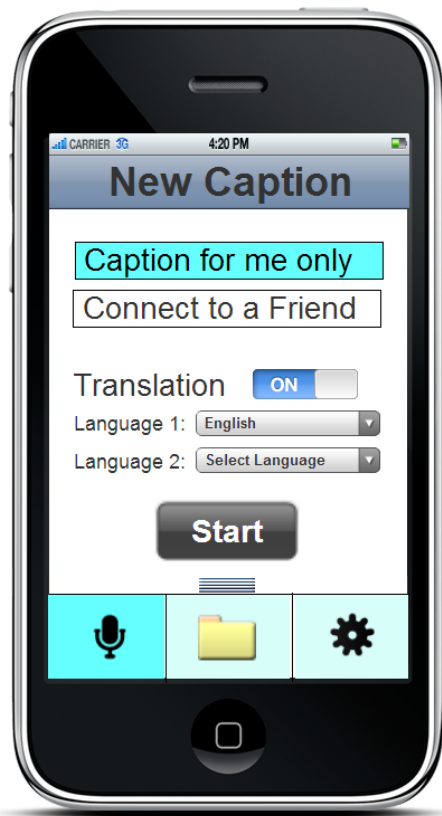


Figure 8: Options available once turning translation to “on”

Captioning Dialogue

During a Caption

Once the user selects “Start” from the homepage, he/she will be taken to a new screen where the captioning will begin. This page will start out blank. When the device hears vocal audio through its microphone, it will caption (and if selected, translate) the dialogue, and it will appear on the screen, while also recording the audio itself. It will begin at the top, and each line of transcribed text will appear below the last. When the entire screen is filled, the most recent captions will appear at the bottom, pushing older lines upwards and off screen. The user can manually scroll up and down by using the dragging gesture on the screen. There will be a scrollbar on the right of the screen to show the user where they are on the entire conversation. If two users are partaking in a connect caption, then the captioning screen on both devices will be identical. Each device will have its own different color background to differentiate the speakers (the user can change these colors in settings). There will be a tab at the bottom, so the user can swipe up on it to get to the menu bar, which is explained later, to end the conversation or go to the library or settings.

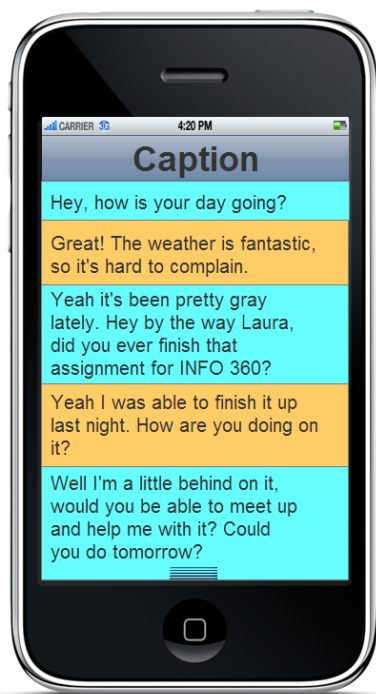


Figure 9: Captioning screen – each colored text represents the two different speakers' dialogue

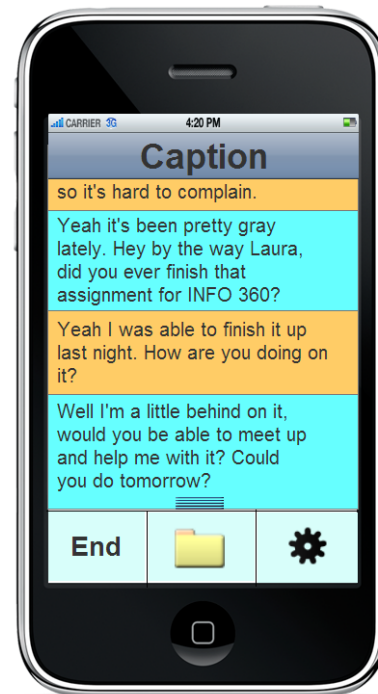


Figure 10: Captioning screen with menu bar at bottom lifted up by the user



Ending a Caption

If the user decides to end the conversation, he/she will hit the end button at the menu bar, where the button to go to the homepage would normally appear. If auto-save is off, a message will pop up (figure 11), asking the user if they are sure they want to end the caption and is granted three choices: “Yes, and save this caption”, “Yes, don’t save this caption”, and “No, keep captioning”. Here is what should happen, respective to each option:

Chosen Option:	Result:
Yes, and save this caption	The caption ends. The user is taken to the library to manually designate a save location (see next page).
Yes, don’t save this caption	The caption ends, and the user is taken back to the homepage. The file is not saved anywhere.
No, keep captioning	The user is taken back to the captioning screen, which has still been captioning in the background, and continues as if the user never hit “end”.

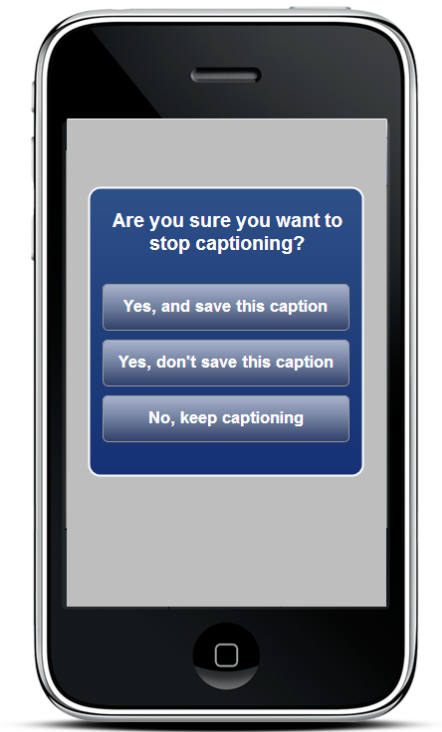


Figure 11: Pop up message that appears when selecting “end” from the menu bar

Saving Caption

If the user selects “Yes, save this caption” with having auto-save off, the user will be brought to the library, where he/she will name the caption and decide the format to be saved (text, audio, or both). It will be explained later what should happen if auto-save is on. The user will also choose which folder to save the file in. If the folder already exists, he/she will select the folder and select the save button located at the bottom of the page. If the user wishes to save the conversation in a new folder, he/she can use the add folder feature to create and name a new folder, and then save the conversation to the created folder. If the user changes their mind and decides not to save the conversation, the cancel button can be selected. The conversation will not be saved and the user will be navigated to the homepage.

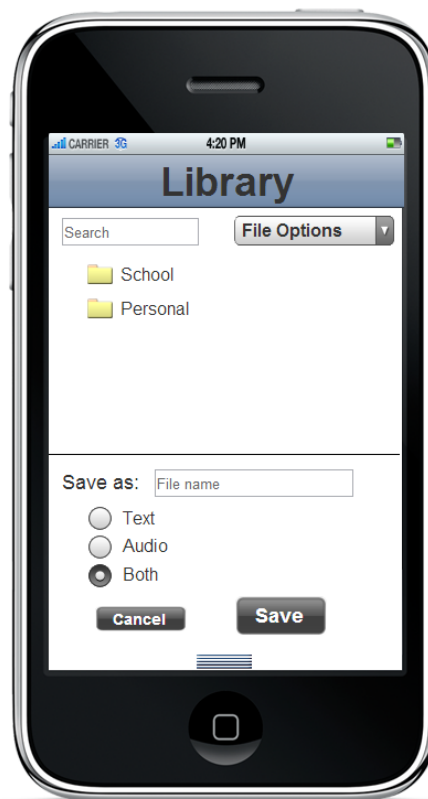


Figure 12: Saving the file upon selecting “Yes, save this caption”



Homepage/Captioning Rationale

The homepage and captioning screen are two aspects of our design that are very closely related. Thus, we felt the best way to present their rationales was together:

Virtually every smartphone comes with Bluetooth technology, allowing two devices to connect directly on their own network. We decided to take advantage of this existing technology in our design in addressing of issue #3: using the device should not draw attention. The initial idea was that the captioning would be forced to take place on a single device, meaning that the user would have to pass their phone to another person to speak in to and would then pass it back to the user so that he/she could read the caption. We later decided this method to be rather tacky, awkward, and it just looked weird from an outsider's perspective, which conflicts with this issue. So instead of running on a single phone, we had the idea that phones can connect via Bluetooth. This way each person is comfortably speaking into his or her own device without the conversation being interrupted. The conversation is able to play out organically how one normally would, and as a result does not look bizarre when using.

Applying our design approach, we decided to include the translation feature in hopes that a general audience would find it useful. People from around the world are starting to travel to other countries more often, but they may not necessarily know different languages. If the user wants to engage in a conversation with a different-language speaker, our application allows one to do so without completely breaking the dynamic of a natural conversation. By having the translation feature, we hope to have diversity in our users from all across the world.

The Menu Bar

The menu bar is displayed at the bottom of every page (figure 13). The menu bar can navigate the user to the following three pages:

- Captioning homepage
- Library
- Settings

We use a microphone icon for the captioning homepage, a folder icon for the library, and a gear icon for the settings. Whichever the user is currently on, the corresponding icon is highlighted to give the user a sense of orientation of where they are in the app. The menu bar is always present but can be hidden by dragging it downward off screen. When hidden (figure 14), the tab still remains visible and can be dragged back up to show the menu bar again.



Figure 13: The menu bar displaying (from left to right) the captioning homepage, library, and settings



Figure 14: The menu bar when hidden, still showing the tab to let the user know of its presence

Rationale: Menu Bar

The menu bar is very small to begin with in order to keep the interface clean. Given that, we wanted to use icons as opposed to text to designate the menus because our expert user has trouble reading small text. We were initially unsure if it would be obvious what these icons represented, but our expert user was able to quickly understand what each icon represents without help, so we decided to keep them.

We decided to make the menu bar always present in order to quickly navigate the app and achieve tasks, relieving issue #2. As proved with Laura's testing, making it hide-able keeps the interface clean while still preserving intuitiveness, issue #1.

Library

- Reviewing a file
- Adding folders
- Deleting existing files/folders
- Sharing files
- Searching for files via the search bar

The library has reference to files that have been saved. Files are not stored locally on the device; rather they are stored in a cloud service. The concept is similar to accessing emails on a smartphone in that the desired files are pulled from a server. When the user wants to view, delete, or share a file, the application will access the stored file in the cloud and perform tasks accordingly. Because of this, an Internet connection is necessary if the user wants to perform any task related to the library. All of the above tasks are designed to be performed on the main library page (figure 15) to eliminate the level of complexity and amount of time in performing a certain task (issue #2), as suggested by Laura.

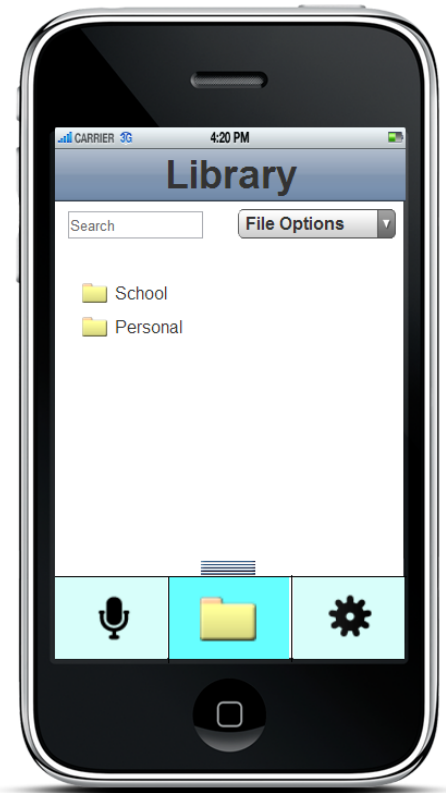


Figure 15: The main library page; a search bar at the top left and a drop down menu for file options at the top right

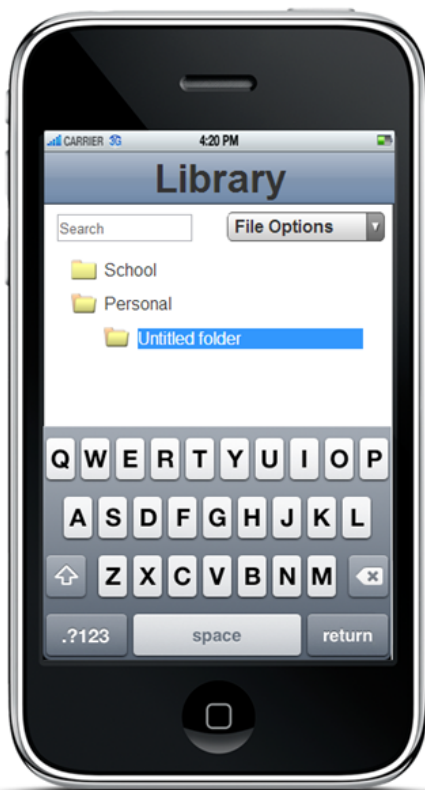


Figure 26: Naming the newly created folder

Reviewing a File

To review a file, the user simply double taps on the desired file. This results in the user being taken to a screen identical to that of figure 9 but with the respective caption, and the user can scroll through the caption at will as if he/she was captioning in real-time. Any recorded audio will also play if it was saved.

Adding Folders

We have decided to allow the user to organize the files using the conventional format of folders. This allows organization of files into a hierarchy format, much like on Windows or iOS. Through the "add folder" button, the user can create and name a folder, along with designate its location so a file can be saved into it later. The user will select "add folder" from the file options drop down menu. A folder is then created in the current location and can be named by use of the device's virtual keyboard (figure 16).

The user can also create a folder while attempting to save a file as previously mentioned on page 12.

We understand that sometimes people would like to create a folder at the time of saving, so we wanted to make all of these decisions completely up to the user. Many people prefer to organize their files differently, and we encourage them to do so in a way that is conventional for them in support of their own personal intuitiveness, issue #1.

Deleting Existing Files/Folders

Obviously with adding files and folders also comes the need to delete them. The user will checkmark the file(s) and/or folder(s) he/she wants to delete by tapping the checkbox next to them and selects the delete button from the file options (figure 17). In order to ensure the user did not accidentally delete files that they wanted to keep, a message box will appear to confirm this action before the system actually deletes them from the server.

Sharing Files

Similar to the deleting files/folders functionality, the user will checkmark the files and folders he/she wants to share to another person and selects the share button from file options. A text box then appears for the user to input the recipient's name if the user has the recipient's email address already saved in contact list, or just input the recipient's email address, and the file is sent from the email address that is linked to the user's account (another reason why we find accounts to be beneficial).

Searching for Files via the Search Bar

A search bar will be displayed at the top left of the library page. Here the user can input the desired filename. The application will quickly scan through all the folders on the server to look for the file. All the files that contain the inputted keyword in their name will be displayed on the page. If no files match, then a message will be displayed at the window to tell user that no matching file can be found.

Rationale: Library

This implementation allows the user to view and organize previously recorded/translated conversations at their own leisure. We believe the library will be most efficiently navigated if the user is given the choice in how they organize it via folders.



Figure 17: A good look at the file options drop down menu once selected. It contains the ability to create a folder, delete folders/files, and send a file.



Initially we opted to include two ways of deleting and sharing files based on different circumstances:

Scenario 1: If the user was interested in *multiple* files/folders, the user could checkmark each appropriate one and simultaneously delete them without ever leaving the main library screen (our current design).

Scenario 2: If the user was instead interested in a *single* file, they could find and select it from the library where the user would then be taken to a separate task page. There, the user could decide whether to view, share, or delete the file.

However, our expert user thought the task page from scenario 2 redundant, so we have chosen to completely abandon this idea and keep scenario 1 only. This way the procedure for viewing/deleting files is the same regardless of whether a single file or multiple files are of interest for the sake of clarity, issue #1.

The search box can also help the user look for saved captions more efficiently. In order to perform the deleting and sharing functions, the user needs to first have visual access to the desired files – they need to be physically selected.

At the beginning, the only way to do this was to manually find them one at a time by browsing through each folder. As a group of average users, we agreed that we often bury files under many levels of folders or forget the location of where a file is saved.

As you can imagine, this is problematic when potentially 5+ levels of subfolders exist and even more so if you're trying to select multiple files all in completely different directories. We ran these examples by Laura who said she was sometimes guilty of doing the same. Our solution to this is the inclusion of a search bar at the top of the library page, which will allow the user to select the file without having to dig, cutting down time and effort (issue #2).

Settings

Main Overview

First off, it should be stated that the term “settings” is an umbrella term for all sub-setting interfaces, which are further addressed below. The settings should be available at all times from the menu bar and is designated by the gear icon on the bottom far right of figure 13.

There are three sub-setting interface: general, account, and logout. Each can be accessed by a navigation bar at the top that is displayed in an inline-block format. Whichever sub-setting interface the user is currently on, the respective button on the navigation bar is solid black to serve as visual feedback while the other two remain grey (figure 18).



Figure 18: Settings navigation bar; the user is currently on general settings

General definitions of the sub-setting interfaces are described here but go into more depth on the next couple of pages. They are separated from within settings with the rationale that they keep finding the information organized, neat, and intuitive:

- General – This is where options to the appearance and behavior of the overall interface are found.
- Account – Here, the user can make specific changes regarding information linked to their account.
- Logout – A manual log out is achieved from this page.

** if you are in any of these sub-setting interfaces, you are considered to be in settings*

Given that the user has at one point navigated to the settings prior, the application should load the exact place and screen orientation that the user was on before exiting the settings menu previously. For example, if the user is under account settings with the drop down

menu for “language” activated, then decides to intermittently go to the library menu, the screen should appear exactly how it did when the user left upon return.

This is to keep a sense of fluidity for the user when navigating to other areas of the device and keeps consistency in achieving tasks, even if interrupted. If this is the first time that the user is using the application and has never been to the settings menu before, then the default screen will be the general settings interface just how it appears in figure 19, with the exception font size is medium and auto-save is off.

General Settings:

- Adjusting font size
- Customizing caption colors
- Toggling auto-save

Adjusting font size

Font size is chosen via a drop down menu with the available choices being small, medium, large, and extra-large like in figure 20. These should equate to standard font sizes 10, 12, 14, and 16 for body text respectively, and standard font size 24, 28, 32, and 40 for headings.

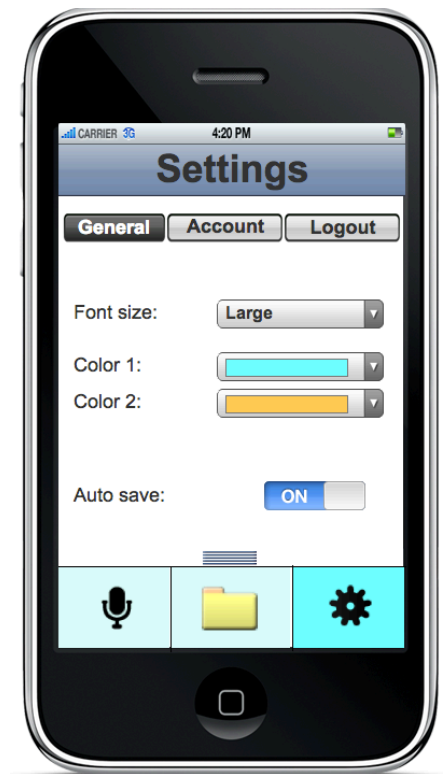


Figure 19: General settings menu



Figure 20: Changing font size via drop down menu

These values were chosen because we have found them to work best given the screen size. Anything larger and margin/spacing errors begin to appear. Upon selection, the font size should adjust immediately – no need for a confirmation window to show up.

Our expert user suggested this feature, finding it useful as a half-blind user, and we have chosen to include it specifically for her use. Because she often has time reading small text, she would like the ability to easily change it to a larger size.

Customizing caption colors

“My color” constitutes to the background color of the text for the user when captioning and likewise with “Friend’s color” but for the person the user is talking with, like in figure 9. Both should be selected by a drop down color slider like in figure 21 but running vertically instead of horizontally.

This was also suggested by our expert Laura due to her difficulty in making out certain colors.



Figure 21: This is what the color slider should like for changing caption color but instead running vertically

Toggling auto-save

This should be implemented as a simple on/off switch and should be off by default. Should the user turn it on, when a user ends a caption, the options for “Are you sure you want to stop captioning?” become a simple “yes” or “no” (refer to page 12). If yes, the user is taken to the caption homepage and the file is immediately saved in a folder called “Auto-saves”. It is created if it does not yet exist. If this feature is off, then the saving occurs as previously described on page 12.

This feature is included for the occasional “power user” in mind, efficiently being able to cut down on time, granted they have somewhat advanced experience with the application.

Account Settings:

- Changing the name associated with the account
- Changing the default language
- Changing the email address associated with the account
- Changing the user’s password

Changing the name associated with the account

Upon first encounter with the account settings menu, the “name” field will be filled in by whatever the user entered for the name field when creating the account. This name is what is used to identify the user’s device when another user is searching the area for phones to connect to and is therefore public.

This feature was chosen in concern of the user’s desired level of privacy, for example if they wanted to include/exclude their last name or use a completely ambiguous alias altogether.



Figure 22: Account settings; absent in this mockup is the inclusion of a “change password” button

This should be achieved by tapping on the input box, typing in the desired name, and selecting “save”.

Changing the default language

The default language constitutes two things: the language that all menu text appears in and the default selection for “language #1” when translation is turned on from the begin caption homepage.

This is to accommodate for all users, regardless of their native tongue, and is selected by a drop down menu.

Changing the email address associated with the account

If for any reason the user would like to change the email they used to create the account to a new one, this can be done from account settings. This should be done the same way as changing the name associated with the account: the user types into the text box the new desired email address and selects “save”. This should result in all files being sent to the user going to this new address as well as requiring this new email address upon logging in to the app.

We allow this so that the user has full control over their credentials and thus their security.

Changing the user’s password

This can be achieved by pressing the “change password” button at the bottom of the account settings menu. When pressed, the user is taken to a new screen (figure 23). There are three input boxes present: one prompting for the user’s current password, the new desired password, and the new desired password again. Also present is a back button that returns the user to the account settings menu and a next button that is faded out until all fields are filled out. Once they are all filled out with the corresponding information, the next button becomes fully opaque and selectable. There are three possible outcomes when hitting next:

1. The inputted current password is incorrect, in which case the device will ignore whether the new password and re-entered password fields match. This situation results in all input boxes emptying, and prompting the user “The given current password is incorrect.” The user may then try again.
2. The inputted current password is correct, but the new password and re-entered new password fields do not match. This also results in all input boxes emptying, but

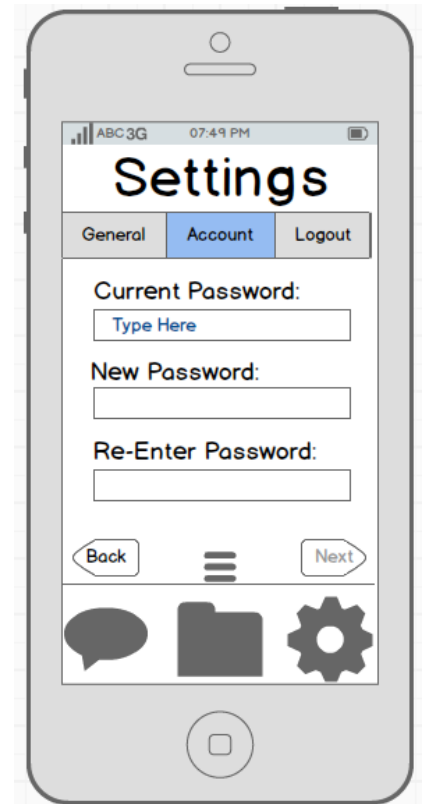


Figure 23: Change password screen; apologies that this mockup was done in a different program due to time constraints.

the user is instead prompted with "The new password and re-entered password did not match."

3. Success case: the inputted current password is correct, and both new password fields match. The user is now prompted with "Your password has successfully been changed!" (the exclamation point is very important). A return button is also present that brings the user back to the account settings menu.

Given that the user has succeeded in changing their password, this new password will be required when logging in to the app.

Logout

Logging out is achieved from within the settings menu and is done by selecting "logout" on the far right of the navigation bar. When selected, a new log out button will appear (figure 24), and the user will be prompted to press it for confirmation. If the user obliges, he/she will be signed out and taken back to the sign in screen.

Reiterating from page 6, logging out is *not* required by the user when ending a session with the application. We came to the unanimous conclusion that we want the login procedure to be as quick as possible. Thus, we have decided to use a method used by many other applications: to keep the user always logged in until they manually log out, even after closing the application or restarting the device. The idea is that the user would only have to login once and could quickly start captioning. We had to slightly sacrifice security to increase navigation speed (issue #2), but the good news is that if the user favors their security over performance, they can manually log out at the end of each session.

A minor problem we foresee with this feature is that if a user was using a friend's device, for example, and wanted to log in to their account, they would have to open the app (logged in as their friend), manually logout, then log in with their own credentials – a rather dragged out process. However, we consider this a relatively specific circumstance, and we feel the benefit of not being forced to input login information every single time the app loads outweighs these kinds of sacrifices, as we do not expect them to occur all that often.

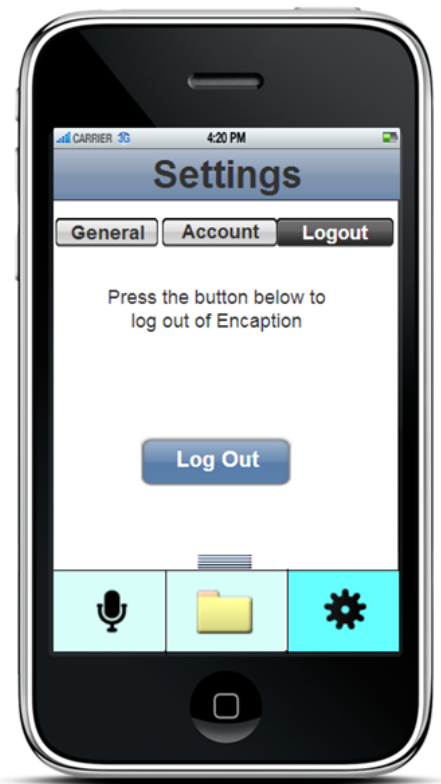


Figure 24: Logout screen