Talking Memories: Android Mobile Accessible Camera Application for Blind and Low-Vision Users

Abdelelah Salama
Computer Science and Engineering
engobada@cs.washington.edu

Chung Han
Computer Science and Engineering
han@cs.washington.edu

Nikhil Karkarey
Computer Science and Engineering
nikhilk@uw.edu

ABSTRACT

Many people in the blind community and those having low vision enjoy taking and sharing pictures with their friends and families. However, currently there is no Android application that is accessible by blind users, which allows them to take pictures and identify pictures taken earlier. To address this problem we have developed an Android application which allows blind and low vision users to take photos with ease, tag those photos with voice labels, browse through the pictures and even share them with others. Our application has a blind and low-vision user friendly UI. In the following report, we will explain our app development process and functionality details.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies, Voice I/O.

K.4.2 [Computers and society]: Social issues – assistive technologies for persons with disabilities.

General Terms: Design, Human Factors.

Keywords: Accessibility, blindness, mobile devices, touch screens, speech output, voiceover

1. INTRODUCTION

According to the Centers for Disease Control and Prevention, more than 3.4 million (3%) Americans aged 40 years and older are either legally blind or are visually impaired [1], and that vision loss is among the top ten disabilities affecting Americans [2]. Although there are specialized devices available for the visually impaired to use, they are often very expensive or not readily available. In this case, mobile phones and tablets are the perfect platform for developing applications which can be accessible by visually impaired users. Mobile phone and tablets are readily available, relatively cheap and can serve a multitude of functions depending on the blind user's requirements. Currently Apple's iOS and Google's Android are the two major mobile platforms available. Although iOS has built in accessibility features like voiceover, there are no camera applications specially built for blind and low-vision users. Android is worse, with poor accessibility features, no voiceover and very few blind accessible applications.

With this in mind, we developed Talking Memories: an accessible Android application which allows low-vision and blind users to take, browse and share photos.

1.1 Intended Users

While researching for the project, we found that many blind people and those with low vision enjoy taking pictures [3]. They use their limited vision and other senses like hearing and touch to frame images in their cameras. The target users for Talking Memories are visually impaired persons who would like to take and share pictures with others. This application may also be used by sighted people who wish to share pictures tagged with voice, with their friends and family members that may be visually impaired.

1.2 Project Purpose

Currently non-sighted persons take photos either using a traditional camera or smartphones. Some traditional cameras may have clear tactile features like buttons which make it easy to feel your way around the camera, and operate its features. Increasingly, many cameras have their navigation features displayed on the screen. This reduces their accessibility by non-sighted people.

The same problem is found in newer smartphones, many of which have no hardware buttons, and all operations for the built in camera app are carried out by touching the screen. There is no voice guidance available which might allow a blind person to navigate through the application, and perform functions like browsing, deleting or sharing images.

Another problem faced by non-sighted users is that once a picture has been taken, there is no easy way to identify the pictures at a later time; for example when browsing and selecting photos. This also makes sharing photos difficult because the blind user cannot communicate the emotions involved at the moment that the picture was taken.

The purpose of our capstone project was to create an accessible Android application which would address the issues identified above, and provide blind and low-vision users an accessible way to take, browse and share photos. We wanted to create an application that would provide the following features:

1) Blind and low vision friendly user interface.
2) Allow users to record voice labels for their photos for easy identification.
3) Allow users to attach emotions and context to pictures they have taken.
4) Allow users to share their photos along with the voice labels, thereby sharing the emotion involved at the time the photo was taken.

1.3 Use Cases

This section describes how a blind or low-vision user may use the application to take pictures, browse through their saved photos, and delete or share photos with others.

1.3.1 Taking a photo and recording a voice tag.

Mary is on vacation with her friends at the beach. She wants to take a photo of her friends, so she starts the Talking Memories app on her Android smartphone. She has not used the app before so she listens to the voice instructions played on the Home Screen. Following the instructions, she moves her finger over the touch screen, and listens to the button names as her finger passes over each button. She stops her finger over the 'Capture' button, and then double clicks it. The Camera Screen opens up giving Mary instructions on how to take a photo. She get distracted for a second and misses the instructions, so she touches the screen and the instructions are played back to her. She then points the camera at her friends and double taps the screen to capture a photo.

The voice Tag Recording Screen opens up and following the instructions, Mary double taps the screen and says, "On the beach in Hawaii, with Susan and George. The sand feels so warm under my feet." She double taps the screen again to stop the recording. The app confirms that her voice label is now saved with the photo she took, and opens up the Camera Screen again for her to take another picture. Mary decides to take another photo, this time of the sea. She follows the same procedure as described, but this time she does not want to record a voice label for the image. So she uses a two finger swiping action at the Tag Recoding Screen to go back to the Camera Screen. She now wants to quit the application, so she again uses a two finger swipe action to go back to the Home Screen, and the same action again to quit the application.

1.3.2 Browsing through photos

In the evening, Mary wants to browse through the photos she took during the day. Mary starts the Talking Memories app, and as before navigates the Home Screen. She double taps on the 'Browse' button, which open up the Browse Screen which displays the latest picture she has taken. Following the voice instructions, she double taps the screen to play back the voice label she had recorded for the photo. As she listens to the voice label, she knows which photo it is and she remembers the warm sand under her feet when she took the photo. She uses a one finger swipe action from left to right to go to the previous photo, and a swipe from right to left to view the next photo. With each photo transition, she hears a 'click' sound. She double taps the screen to listen to her voice labels. When there are no more previous or later pictures to show, the app voice feedback informs Mary. Mary decides that she want to view all images once again. So she starts a slideshow, by using a one finger bottom to top swipe gesture. The app displays all her images and speaks out their associated voice labels one-by-one. If there is no attached label, the app informs Mary that the image being displayed has no voice tag. Mary stops the slide show by touching the screen once. She goes back to the Home Screen by using a two finger swipe action on the Browse Screen, and quits the app by using the same action on the Home Screen.

1.3.3 Deleting a photo

Mary wants to delete all images that she took on vacation, for which she did not record a voice tag. She starts the app, and navigates to the Browse Screen. She flips through her photos as described in the previous use case until she finds a photo which has no voice tag attached. She uses a one finger long click on the screen, which opens up the Delete or Share Screen. As always the voice feedback informs Mary of the new opened screen and gives her instructions. Mary uses her finger to navigate the screen and finds the 'Delete' button. She double taps the button to delete the photo. The app then asks Mary to confirm that she wants to delete the photo or cancel the deletion by using a two finger swipe action to return to the Browse Screen. Mary decides to confirm deletion, and double clicks the screen again, to delete the photo and any associated voice tag, if any. The app gives Mary a voice confirmation of the same, and the opens up the Browse Screen. Mary proceeds browsing through her photos as described earlier to find other photos with no voice tag. After she has completed deleting all non-tagged photos, she quits the application as described in the previous use case.

1.3.4 Sharing a photo

Mary has returned home from her vacation and wants to share a photo that she took with her mother. She navigates to the Browse Photo Screen, selects the photo she wants to share, and then navigates to the Delete or Share Screen as described in the previous use case. She now double clicks on the 'Share' button, which opens up the keyboard screen. She follows the voice instructions and enters her mother's email address using the voice feedback accessible keyboard. She checks that the correct email address is entered by pressing the 'read characters' key on the keyboard. Once satisfied, she presses the 'done' button. The app the sends the selected photo and its associated voice label as email attachments to Mary's mother. While the email is being sent, the app repeatedly gives Mary feedback of the app's status. The app informs Mary that the email is successfully sent, and then returns to the Browse Screen. Mary having shared the photo, quits the app as described earlier.

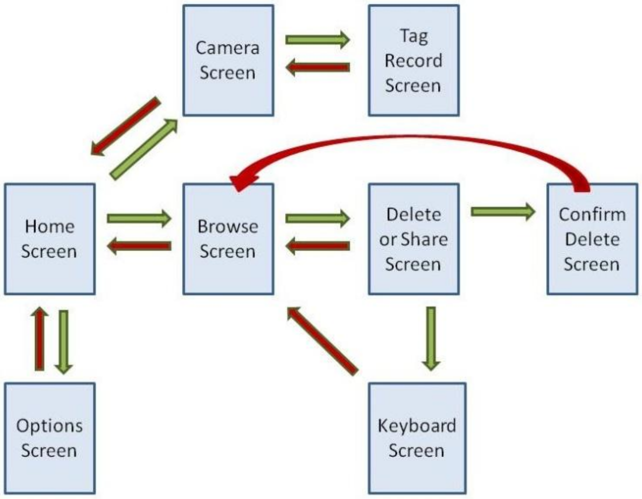

Figure 1. Application screen transition diagram.

2. RELATED WORK

2.1 Projects

For our application, we required an accessible keyboard interface for users to enter email addresses of recipients. Michael Q. Lam, Levi Lindsey and Chris Raastad have implemented such a soft keyboard for text entry that is accessible to blind users. This keyboard implementation was a part of their UW CSE 481H Accessibility Capstone project for the quarter of Winter 2011 called 'Phone Wand'. [4] We used their open source code for incorporating the accessible keyboard in our application.

2.2 Research Paper

There has been research done in the UW on interaction of blind users with mobile touch screens using multi touch gestures. [5]

We would have liked to implement the touch gestures identified in this research in our application, but unfortunately we only became aware of this research after our final project demo. We intend implement the Slide Rule gestures in our app in the future.

2.3 Products

We carried out the app development and testing on the following phones:

1) Samsung Galaxy S II (Android 2.3.6)
2) HTC T-Mobile MyTouch 3G Slide (Android 2.2)
3) Motorola Droid (Android 2.2)

3. SOLUTION

Our solution to the problems identified in Section 1.2, was to create a mobile Android application that provides blind and low-vision users an accessible way to take, browse and share photos.

The first step in the application development was to implement and test the underlying functionality of the application. For this we had a very basic initial user interface.

3.1 Taking photos and recording voice labels

This application allows the users to record a descriptive voice label for each image immediately after taking the photo. This voice label adds context and emotion to the images, which later be played back to bring back memories of the moment that the picture was taken.

Since this application had to provide a custom user interface for blind users, we had to avoid using the in-built Android camera and audio recording applications. This required directly accessing the phone camera hardware to display the camera preview screen and capture pictures.

3.1.1 Data storage location

Early on, we had to decide whether to store the images and their voice tags in the phone's internal memory (within the application) or in the external memory (SD card). We decided to store the image and audio label data in the phones internal memory. The reasons for doing so are as follows: If stored internally, the data captured by the application is only accessible to the app, and cannot be modified by external applications. We did not want accidental deletion of images or their associated tags by external applications. Also, storage of data in SD card forces user to have an SD card mounted, and not all phones may have an SD card mounted.

3.1.2 Associating images and their audio labels

We also had to decide whether to store images and their associated audio labels as a single custom file, or as separate files. We decided to store them separately, so that the files are portable across applications. This would allow sharing the files via email, and the receivers would be able to open the image and play the audio file using an application of their choice.

In order to associate the images and audio files, we used the Android supported SQLite Database and a consistent file naming system. Every image file is captured by the camera is named 'tm_imageX.jpg' and the associated audio file if recorded, is named 'tm_audioX.3gp'. Here X is an integer, and every associated pair has the same X number. For example, when the first photo is taken it will be named 'tm_image1.jpg' and the associated audio file if recorded will be named 'tm_audio1.3gp'. The files are stored in the internal memory in no particular order. Once the files are stored, their absolute path is entered in a relation in a SQL database. This database can later be queried to retrieve association information for the image and audio files.

3.2 Browsing photos

While browsing photos, the user is able to play back the voice tag which was recorded when the image was taken. This feature was implemented by using the Android framework provided Cursor class. We used a Cursor over the SQLite database to get the file paths for each image and its associated audio file. Once we have the correct paths, we can display the images on the screen and play back the correct associated audio file for the image.

Every time that the user performs an action to advance an image or go back one image, the Cursor is advanced or retarded by one row in the database table, to get the path of the image to display, and its audio tag to playback. If an image has no audio tag, a pre-recorded file is played back instead which informs the user of the missing audio tag.

The application also has an additional slideshow feature which cycles through all the saved images one by one and plays their associated audio tag.

3.3 Deleting and Sharing images

3.3.1 Photo deletion

When the user selects and confirms image for deletion, the image and its associated audio tag are deleted from the phone's internal memory and their paths are removed from the database. The Cursor position is set to the next available image in the database. When the user returns to the photo browsing screen, the next in line image is displayed on the screen.

3.3.2 Sharing photos and audio labels

For sharing photos and audio tags, we considered using MMS, Bluetooth, a Web Service and email. Bluetooth was discarded because it only works for short ranges and requires users to synch their devices, which is not the easiest operation for a blind user. We found out that Android does not support MMS transmission directly through a user application, and one has to use the built in phone messaging app. This does not allow for a custom accessible UI to be built for the sharing activity. This is the reason we did not use MMS for sharing. Since we were working within the constraints of a quarter system, we did not have enough time to implement a Web Service to handle photo sharing. Therefore we decided to share images via email.

In order to send email with attachments directly from the application, we used Oracle's JavaMail API which provides a platform-independent and protocol-independent framework to build mail and messaging applications.

When a user selects a photo to share from the Browse Screen, the image and its audio tag if any are added as attachments to the email to be sent. We incorporated a blind accessible keyboard from a previous UW CS481h capstone project [6], with which the user can enter the recipient's email address.

The user's phone number is obtained from the phone and added to the email subject line, so as to identify the sender of the email. An email with the attached photo and voice label is sent via a forwarding Gmail account.

3.4 Blind and Low Vision Friendly UI

This section describes the features of our app UI which is blind and low vision user friendly.

3.4.1 User Testing and Feedback

We tested our prototype UI twice with two blind users during the intermediate development stages, and took their feedback. We also conducted blindfolded testing with 2 sighted users who were not familiar with our app, to fine tune the UI.

Following were suggestions from the blind users that we incorporated in the app UI:

1) Require consistent gestures on all screens to perform forward going actions.
2) Require a universal gesture (escape gesture) to go back to previous screen.
3) Need a way to allow user to replay instructions on every screen
4) Require clear indication of screen transitions within app
5) Require clear indication for photo transitions while browsing, and to indicate end of gallery reached when browsing photos.

We implemented all the UI features above that were suggested by the blind testers, and we got positive feedback from the blind users when they used the app with the final UI.

3.4.2 Voiceover for Android

Android does not built in voiceover support. Therefore we implemented voiceover functionality for our app. Instead of default XML layouts for buttons, we had to create a custom canvas paint view to draw the GUI programmatically, giving more flexibility and control over the view. We then implemented double click gestures for this custom view, as well as custom listeners to detect when a finger transitioned from one button to another. In testing, we found our voiceover design to be robust and error free.

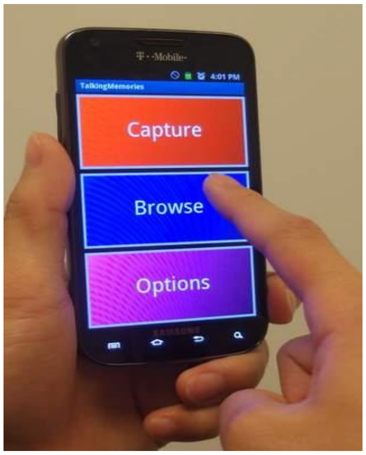

Figure 2. Using the voiceover feature. Note the bright colors and large buttons

3.4.3 Bright colors, large button

All the screens have bright bold colors with lighter text, large buttons, and touch responsive surfaces.

Since Android does not have built in voice over support, we implemented the feature for this app.

3.4.4 Screen transitions

On each screen transition, the user is notified by voice feedback, and the name of the newly displayed screen is spoken out. This assists blind users in navigation.

3.4.5 Consistent gestures:

For ease of use and to facilitate familiarization with the UI, the app features consistent gestures on every screen. A double click performs a forward going action, a single click on a screen replays instructions for that screen, a two finger swipe on each performs a cancelling action, and a long click is used to perform additional features which on a screen. All actions on completion trigger voice feedback from the app, so the user gets immediate confirmation of their action.

3.4.6 Voice instructions:

On each screen the applications provides voice instructions for that screen. The user can easily replay the instructions by ingle tapping the screen. A user can interrupt the instructions at any time by performing an action the screen. Once a user gets familiar with the app, long instructions on each page can get annoying. To address this, the user can set details of instructions that they want hear. A low vision user can turn off voice instructions if needed and use vibration feedback from the phone.

4. Future Work

We would like to continue making improvements to the functionality and usability of the app. Some work which we have identified for the future is as follows:

1) Implement a web service, which would allow users to create accounts for sharing images, instead of using Gmail.
2) Implement the Slide Rule user interface to allow more efficient app navigation.
3) Implement voice recognition to perform actions for navigating the app.
4) Develop Talking Memories app for iPhone.

5. REFERENCES

[1] Eye Diseases Prevalence Research Group. Causes and prevalence of visual impairment among adults in the United States. *Archives of Ophthalmology* 2004; 122: 477–485.

[2] Centers for Disease Control and Prevention. Prevalence of disabilities and associated health conditions among adults – United States, 1999. MMWR 2001; 50(7):120-125.

[3] Daw, Matt and Badenoch, Gina. Sensory Photograph: Photography for blind and partially sighted people. DOI= http://www.photovoice.org/html/methodology4sp/report.pdf

[4] Chris Raastad, Levi Lindsey, Michael Lam. 2011. The Phone Wand - Navigating routes using orientation and vibration feedback. UW (CSE481h) Accessibility Capstone Project.

[5] Shaun Kane, Jeffrey Bigham, Jacob Wobbrock. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility. (2008). DOI=http://doi.acm.org/10.1145/1414471.1414487