Comp 4768 Project Report

You Don't Say

Christopher Healey & Joshua Rodgers

Table of Contents:

Introduction / Concept

The concept for our project was to create an interface that would allow users to connect to one another via Bluetooth using the MultipeerConnectivity framework and send files to one another that would display over a chat format to the user on their screen. The main concept of the app was going to be that we would allow users to transfer any kind of data besides the regular chat format of text. This brought about the name "You Don't Say" as that's exactly what wouldn't be transferred about the interface. Along with the name, it also bred our first idea of what to transfer along our interface, photos but with the add-on of memes. This would allow the user to select a photo from their photo library or take a photo from their camera and edit the photo with a top line of text and a bottom line of text, which would then be embedded into the photo itself and sent to any connected peers. Any image or content that was sent to another user would then appear in that users chat feed, recieved information on the left hand side and sending information on the right hand side. Before a received image is loaded a border will be displayed in its place to indicate that an image has been or is being received. Clicking any content in the chat history would then bring the user to a separate page specific to that information type, photos that we implemented would bring the user to a page where they could view the photo in a larger format. If the user or peer likes the photo they can save it directly to their phone, but they should do this before they close the app as when the app closes the chat history for that user is cleared.

As from our initial concept of sending and receiving information through MultipeerConnectivity, we designed the initial app around the functionality of our concept,

before we started to move onwards to the user feel of our design which we haven't been able to finalize as of our project submission.

User Interface Interaction

Upon opening the app the user will be greeted with a short welcome message describing how the app works. The Browse and Disconnect buttons allow you to connect and disconnect from other users, this is implemented using the MultipeerConnectivity framework which allows users to choose from devices using the app and prompt them to connect with one another. The Send photo button brings you to the photo picker view, which will allow the user to choose their photo to send across the interface to the users they connected to using the browse or users that they have accepted connection to using the prompts on the screen. These buttons can be seen in Figure 1.

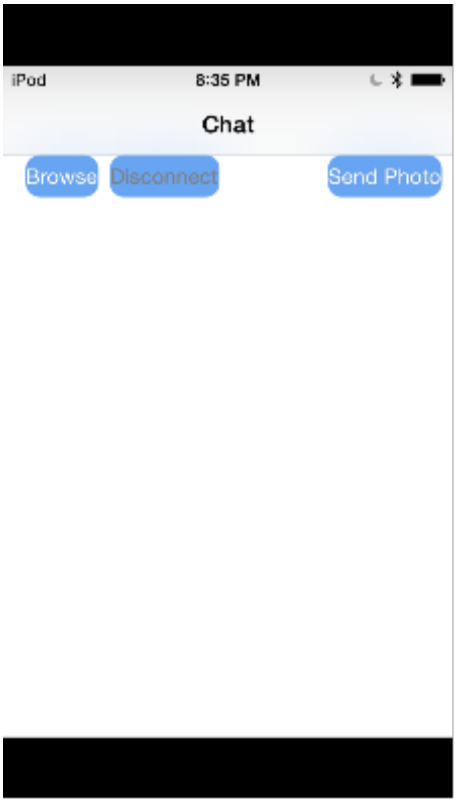

Figure 1

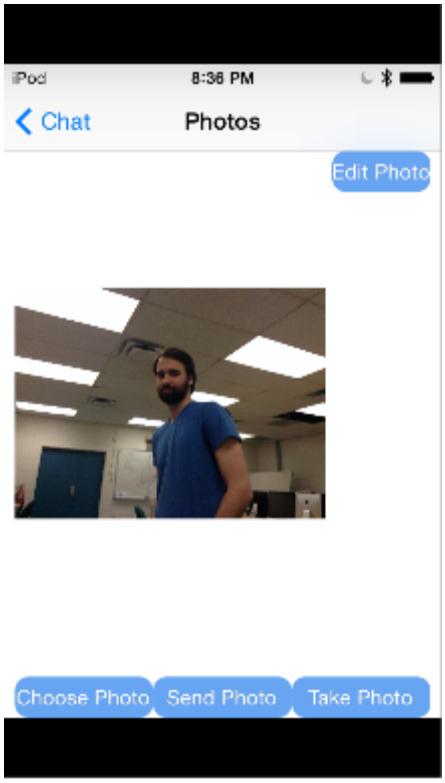

Figure 2

The photo picker view allows the user to choose a photo from their phones photo library or take one themselves that the app will use. This picture is then displayed on the screen. From here the user may also continue forward to edit the photo using the edit button provided. As shown in Figure 2.

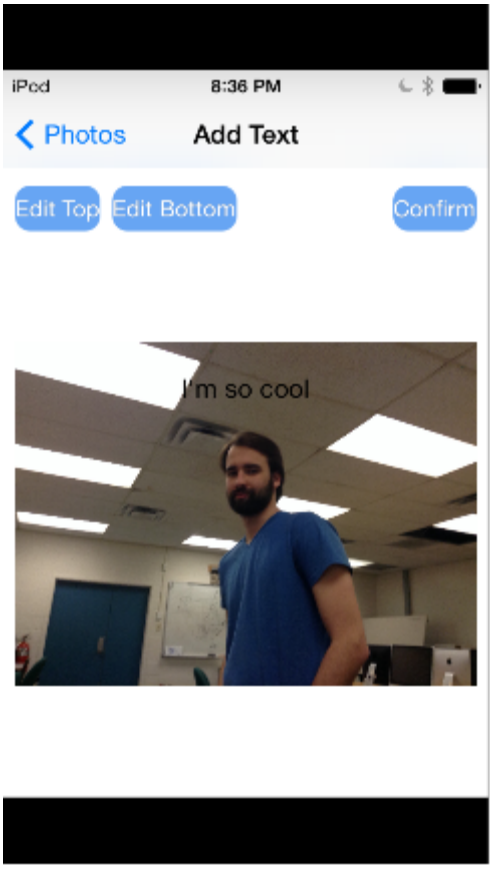

Figure 3

In the Edit view, the user will be able to edit the photos top and bottom labels with their respective buttons that will then display the text on the labels of the image, the user then has a chance to confirm if they want to add the text to the photo. This is displayed in Figure 3.

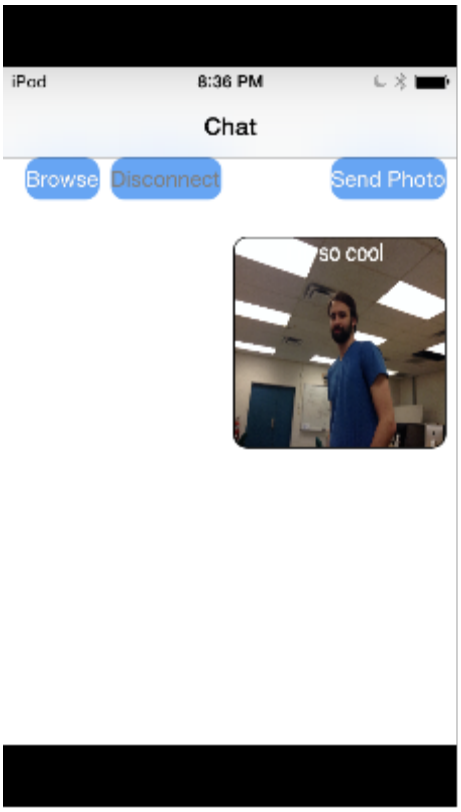

Figure 4

When the user is ready to send the photo they can work their way back the 2 view controllers, confirm on the edit view will bring the user to the photo picker view controller again, and then tapping the send photo button will then send the photo to any connected peers, the photo will then show on the users Chat interface on send side(the right side). See Figure 4.

When a user receives a photo they will be prompted saying who the photo was received from and display the photo on the received side of the chat view(the left hand side)(As shown in Figure 5) . Before this image is completely loaded the chat view will show a border to indicate that a picture has/is being received.

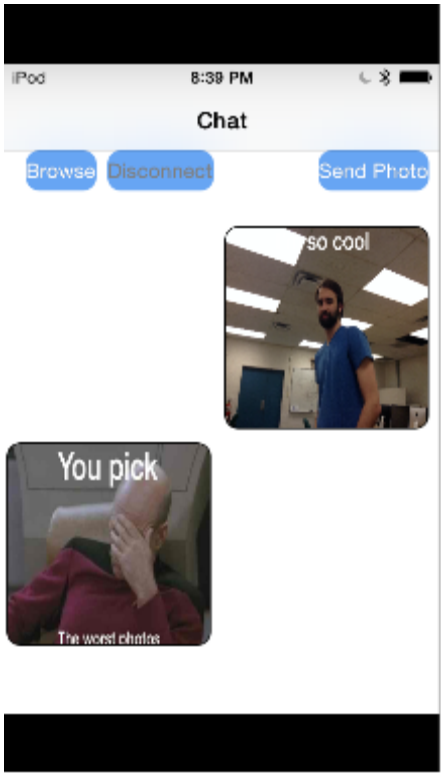

Figure 5

Figure 6

The user can select any photo from the chat view and it will be displayed in the view image view at which time the user can choose to save the image directly to their phone in their photo library. If the user fails to do so and closes the app that image will no longer be available to them as their chat history will be cleared.

Implementation

This app utilizes the Bluetooth networking capability via the MultipeerConnectivity Framework to connect each device with its peers. There are currently 4 views of this app, the chat view, the photo picker view, the edit photo view, and the save photo view and when the application is completed there will be more. The app uses a navigation controller interface to switch smoothly between these pages. We had initially decided to implement a tab bar interface however we ran into some difficulty calling functions in other tabs and found it much easier to use the navigation controller interface. "You Don't Say" also uses the UIImagePickerController class to allow the user to easily select a photo from their photo library and it also invokes the UIImageWriteToSavedPhotosAlbum method to save any image directly to their phone.

The Future

In the future we intend to allow the sending of many different files instead of just images. This may include, text files, music files, business cards, etc. Business cards may be sent as their own file type, or we could create an edit business card view as we did for adding text to the pictures and have a business card template for the user to fill out. We intend to be able to send all of these types of files, however as described before we will not be implementing a chat function to keep to the name "You Don't Say". For each of these different file types we will also have to implement a view page to allow the user to properly view their file, for example an audio view which only contains an audio player or a text view which allows the

user to read any text files. Also we could make the app more user friendly by displaying a progress bar that shows that a message is being sent or received instead of simply showing the border of the image that is to be displayed. Another idea that we had for this app was to save the entire conversation. Instead of clearing the chat history whenever the phones disconnect never to see the chat again, it would instead allow the users to disconnect and when they reconnected their previously sent messages would still be visible to them in the chat view. Our overall goal is to make a user friendly app that allows the user to share all types of files with their friends.

Conclusion

In conclusion, You Don't Say is an iOS app which allows users to share photos in a chat format (received images on the left hand side and sent images on the right), add text to those photos, and save any image that has been sent to them directly to their phone. Alerts are used to help guide the user through the process as well as letting them know they have accomplished what they wanted to do such as an image has been sent or received. The border of an image is displayed while the image is still loading to indicate that an image is being received or is still loading. It uses the Bluetooth networking capability via the Multipeer Connectivity Framework as well as a navigation controller interface. In the future we hope to make it as user friendly as possible as well as saving the chat history so that when users close the app and reopen it their previous chat is still visible. Also we would like to be able to send all types of files to other users without the ability to chat through text("You Don't Say"...anything) accompanied with the proper view for each type of file.