TalkSee

Project Plan

Team 6

Mark Higa
Chris Bauschka
Ashley Braun
Aaron Petry

Table of Contents

Purpose

TalkSee is a Windows Phone application that utilizes the Microsoft Research Hawaii SDK and a wireless internet connection. The application is designed to aid people who have hearing conditions or may simply be in an environment where it is difficult to hear the phone speaker. As users communicate through a voice or video call over an internet connection, the app will provide speech-to-text captions for spoken words, providing a visual means and supplement for communication. Users will also be able to manually type and send text communications during the call.

Document Purpose

The Purpose of this document is to further explain the functionality of each subsystem of the project and the associated risks and challenges. These subsystems are divided into four sections: Client-Client, Client-Server, Client-STT Cloud, and User Interface. Each of the tasks for each of the sections is sorted into one of three different build stages, as well as a cost estimate for how long it is projected to take to complete each task. This document is structured into the following sections: Purpose and Background Information, Team Organization, Subsystems, Planned Releases and Tasks, Risks and Challenges, Cost Estimates, and Gantt Chart.

Background Information

The application was inspired after observing a friend sign to his deaf mother through FaceTime on an iPad to iPhone connection. Seeing them communicate directly without the help of an added human interpreter was amazing, but there were still many limitations. The first and most immediate was that while his mother could also see me, my inability to sign still kept me from communicating with her. But if an app could quickly translate speech into text as captions, she could read what I said even though she could not hear me. Additionally, signing usually requires users to be in a stationary environment typically requiring both hands to sign. An effective speech-to-text caption application would enable them to travel with phone in hand while seeing the other user sign and being able to speak back.

We quickly realized that this concept could be improved and expanded to help an even greater number of people, particularly the elderly who may have difficulty hearing. This would allow them to communicate without special phones or hearing aids (if they don't feel like wearing one just to take a call). Even with proper hearing aids, it could provide a visual confirmation of the verbal communication. Additionally, users with adequate hearing who are in noisy environments might use this as a supplemental communication aid. And by having the captions available in a transcript form, users could review the conversation if they missed what the other user said.

There is a service called CapTel that both ATT and Sprint offer that adds captions to voice calls. However, this solution involves a call center with human translators who actively listens to the call and transcribe it to text. A new application for iOS was recently introduced called ClearCaptions, but we do not know how well it performs. We are currently unaware of any applications that provide automated speech-to-text captions for voice calls on the Windows Phone platform or speech-to-text captions for video-calls on any mobile platform. We believe that with the cross-platform nature of the new Windows environment, and the plethora of smartphone, tablet, and computer devices soon to be released by Microsoft, our application will provide an excellent solution for a very wide market.

We are exploring multiple voice and video-call platforms with public SDKs and APIs to find an ideal solution that we will be capable of implementing. We are also prepared to develop custom solutions with possibly limited functionality if necessary. We will use the Microsoft Hawaii SDK to provide the speech-to-text service.

In the past weeks, the team has gathered prerequisite information and drawn up the required design documents that specify how the implementation will proceed. With those documents completed, it will be the responsibility of each team member to implement their assigned sections. The team will use the Windows Phones provided by Dr. Korb to prototype and test the application builds. In the coming weeks, the team will work on three iterations of the application, with each build adding more functionality.

Team Organization

The team structure will reflect the subsystems defined below. Mark Higa will concentrate on the Client-STT Cloud interaction and functionality. Chris Bauschka will focus on the communication between the Client and the Directory Server (Client-Server). Aaron Petry will work on the communication involved between two clients (Client-Client). Ashley Braun will be responsible for the user interface. We will use frequent and regular scrums to organize our weekly work and goals. Mark is the project leader, so each team member will report to him for final approval and code review. The team will have at least four meetings a week at which we will discuss the progress of each team member and review recently added features and code.

Subsystems

Client-Client

This subsystem encompasses all communication between the multiple clients hosted on the directory server. The client will have the ability to start a call, end a call, accept or reject an incoming call, and send or receive a chat message. While in a call, the client will also be able to send and receive audio and video from the client currently sharing the call. Also, once the Dialer has received strings of text from the STT cloud, it will automatically send that text to the other client.

1. Dialer (Client)

1.1. Initialization

- When the TalkSee client is launched, it creates its own "dialer" which allows for calling functionality.

1.2. Starting a Call

- After selecting another client on the Directory Server, the client will have the ability to start a call which will set the status of both clients to busy.
- When calling another client, it is requesting permission to open communication (sending messages, video, and audio).

1.3. Ending a Call

- When a client elects to end a call, both clients are no longer set as "busy".
- Communication is no longer accepted between both clients.

1.4. Accepting/Rejecting a Call

- When a client accepts a call, communication is open and both clients may freely send video, audio, and text.
- When a client rejects a call, communication request in denied and both clients are returned to the available status.

1.5. Sending/Receiving Chat Messages

- The client will have the ability to send chat messages of text inputted by the user.
- The client will also have the ability to read any chat messages sent by the other client in the call.

1.6. Sending/Receiving Audio

- When in an active call, the two clients will be able to communicate through audio which is recorded by both of the clients.

1.7. Sending/Receiving Video

- Like audio, if the clients have the ability for video communication, they will be able to record video through the dialer as well as send it through the dialer to the other client.

1.8. Sending Text Translation Strings

- When the Dialer receives a Text Translation string from the STT Cloud, it should automatically send that text to the other Dialer.

Client-Server

This subsystem will encompass all communication between the Client and the Directory Server. The Client and Directory Server communicate over WiFi by sending text strings back and forth. The Directory Server keeps track of the status of all connected Clients. If a Client has not updated its timestamp recently, it will be removed from the directory listing. The Directory Server only sends updated directory listings to the Clients. The Client sends several messages to the Directory Server. The Client can tell the Directory Server that is registering or deregistering, busy or not busy, or update its timestamp.

1. **TalkSee (Client)**

1.1. Registering

- When the TalkSee Client is launched, the Client attempts to register itself with the Directory Server.
- The Client sends its identifying information and the Directory Server gives each client a unique ID.
- The Directory Server pushes the updated directory listing to all connected Clients.

1.2. Deregistering

- The TalkSee Client when closed sends a message to the Directory Server telling it to deregister the Client. If a Client's timestamp expires the Client is deregistered.

1.3. Updating Status

- The TalkSee Client sends its status to the Directory Server. If in a call, the Client (and the calling Client) are marked as busy and are unable to be called by other Client until they are no longer busy.

1.4. Updating the Timestamp

- The Directory Server regularly checks the timestamps of each Client. If the difference between the current time and the timestamp is too large, the Client is removed from the directory listing. Each Client regularly updates this timestamp by sending a message to the server.

1.5. Connection Error

- If the Client is unable to connect to the server it displays an error to the user.

2. **Directory Server**

1.1. Adding a User

- The Directory Server gives each Client a unique ID and stores its identifying information in its directory listing.

2.2. Removing a User

- When a Client's timestamp expires or it send the deregister command it and all its identifying information is removed from the Directory Server.

2.3. Pushing out the directory to Clients

- The Directory Server runs through its list of Clients and sends the directory list to each one. This includes information about the status of each Client.

2.4. Checking the timestamps

- The Directory Server regularly checks the timestamps of each Client. If a Client has not checked in an updated it timestamp within a certain amount of time, the Client is removed from the directory listing.

Client-STT Cloud

This subsystem encompasses all communication between the Client and the Speech-To-Text Cloud. The Client device utilizes the Speech Recognition Library provided by the Project Hawaii SDK, which implements an interface that enables communication with the Hawaii Rendezvous service (STT Cloud). The Client continually records audio clips whenever a call is active. Audio clips are delivered to the STT Cloud for translation in a single call, and the Client then receives the translated text.

1. **Dialer (Client)**

1.1 Initialization

- When the TalkSee Client is launched, the Dialer authenticates itself with the STT Cloud using a Hawaii Application ID
- It also queries the STT Cloud for a list of available grammars and automatically selects Dictation

1.2 Recording Audio Speech Clips

- Audio clips are continually recorded whenever a call is active. The duration of audio clips will be set to minimize delays and provide the best possible accuracy

1.3 Sending Audio Speech Clips and Receiving Translated Text

- As soon as an audio clip is finished recording, it is delivered to the STT Cloud for translation. The Client then receives the translated text from the STT Cloud
- After receiving the text translation, the audio clip is deleted from storage

1.4 Creating Captions

- As each translated text segment is received, it will be compared with the previously displayed segment. Adjustments may be necessary to account for the break between audio recordings and improve overall accuracy

2. STTCloud

1.1 The STTCloud is provided by the Microsoft Research Group Hawaii Server

User Interface

This subsystem encompasses all graphical user interface elements. The user interface includes the home page, help and settings pages, call notifications, and in-call screens. It also comprises the ability to interact with the system through a combination of touch gestures and hardware controls.

1. TalkSee (Client)

1.1. Initialization

- When the application is launched, a loading screen will appear while everything else is initialized

1.2. Display Connection Error

- If something goes wrong with the initialization, a connection error will display

1.3. Display Directory

- Once the application successfully launches, a directory screen will appear with all of the contacts connected to the network

  1.3.1. These names will be navigable by letter blocks

  1.3.2. Names will be actionable buttons

1.4. Display Call Screen

- When a call is connected successfully, a call screen will display

  1.3.1. A video of the other client will appear initially

  1.3.2. A chat box and the most recent messages will display below that

1.5. Display Request Screen

- When a user is clicked, a call request screen will appear

1.6. Display Reject Screen

- When there is an incoming call, a screen will appear with the options to accept or reject

1.7. Display Chat Screen

- While in a call the user can toggle to the chat screen to see message history

1.8. Display Settings

- A settings screen will display when requested

1.9. Display Help

- A help screen will display when requested

Planned Releases and Tasks

Build 1 (11/02/12):

General Dialer Functionality and Chat Messaging:

1. **Call Functionality**

   1.1. Add ability to start a call

   1.2. Add ability to reject/accept a call

   1.3. Add ability to end a call

2. **Request for Communication**

   2.1. Add request for communication from targeted client whenever a client attempts to make a call.

   2.2. Add an acceptance of communication

   2.3. Add a denial of communication

3. **Sending Chat Message**

   3.1. If communication is open, send a string of inputted text taken from TalkSee.

   - Text is inputted by user

4. **Receiving Chat Message**

   4.1. If a chat message has been sent by the other client, accept it and display it on the TalkSee UI

Communicating with the Directory Server:

1. **Set up the Client to send messages to the Directory Server**

   1.1. Create a TCP socket on one of the Client's ports to communicate with the Directory Server

   1.2. Collect the pertinent identification data

   1.3. Conduct basic error checking on the Client's name

2. **Set up the Directory Server to accept messages from the Client and display its current directory list**

   2.1. Create a TCP socket on the Directory Server

   2.2. Record the information sent from the Clients in the directory listing

General Speech-To-Text Functionality:

1. **Set-Up STT Cloud Prerequisites**
    1.1. Add required Project Hawaii SDK assemblies
    1.2. Reference the Hawaii Speech namespace in code
    1.3. Set up the Hawaii Application ID

2. **Initialize Client to STT Cloud communication**
    2.1. Configure Dialer to authenticate with STT Cloud using Hawaii Application ID and select Dictation Grammar when TalkSee Client is launched

3. **Record Audio Speech Clip**
    3.1. Add function to record audio clip using a button press to begin recording and button press to end recording
    - Recording is stored locally on device

4. **Send Audio Speech Clip**
    4.1. Add function to send audio clip to STT Cloud using a button press

5. **Receive a Text Translation String**
    5.1. Configure Dialer to receive a text translation from STT Cloud and store the String
    - Automatically delete audio clip after receiving text translation to preserve memory

General User Interface Display

1. **Display Connection Error**
    1.1. The application should be able to display a connection error (helpful for early debugging)

2. **Display Directory**
    2.1. The directory should display names, and these names should be actionable

3. **Display Request Screen**
    3.1. The application should be able to display a request screen, and the buttons should be actionable

4. **Display Reject Screen**
    4.1. The application should be able to display a reject screen, and the buttons should be actionable

5. **Display Settings**

5.1. The application should be able to display a settings screen, and these settings should be editable

6. Display Help

6.1. The application should be able to display the help screen

Build 2 (11/16/12):

Recording and Sending Audio and Video:

1. Record Audio

1.1. Using the dialer, continually record set duration audio clips

2. Send Audio

2.1. Configure Dialer to automatically send each audio clip to the other client as soon as it is finished recording

3. Record Video

3.1. Like audio, continually record set duration video clips

4. Send Video

4.1. Configure Dialer to automatically send each video clip to the other client as soon as it is finished recording

5. Send Text Translation Strings

5.1. Configure Dialer to automatically send text translations when they are received

Communicating with the Directory Server:

1. Allow the Client to tell the Directory Server it is busy

1.1. The Client sends a special message to the Directory Server telling it is busy

1.2. The Directory Server will accept the message and flag the Client as busy

2. Allow the Client to tell the Directory Server it wants to deregister

2.1. The Client sends a special message to the Directory Server telling it that it wants to deregister

2.2. The Directory Server removes that Client from its directory listing

3. Add the functionality to allow the Directory Server send out its directory listing to all connected Clients

3.1 The Directory Server sends out its directory list and the identifying information to each connected Client

Continuous Speech-To-Text Functionality:

1. **Record Audio Speech Clips Continuously**

    1.1. Configure Dialer to continually call function to record set duration audio clips

    - Recordings are stored locally on device

2. **Automatically Send Audio Speech Clips**

    2.1. Configure Dialer to automatically send each audio clip to the STT Cloud as soon as it is finished recording

3. **Receive Multiple Text Translations Strings**

    3.1. Configure Dialer to continually receive text translations from STT Cloud and store multiple Strings

    - Once a text translation is received, delete the corresponding audio to preserve memory

Advanced User Interface Display

1. **Display Connection Error**

    1.1. A connection error should display if something goes wrong with the initialization

2. **Display Directory**

    2.1. Once the application successfully launches, a directory screen should appear with all of the contacts connected to the network

    - These names should be navigable by letter blocks
    - Names should be actionable buttons

3. **Display Call Screen**

    3.1. A call screen should display when a call is connected successfully

    - A video of the other client should appear initially
    - A chat box and the most recent messages should display below that

4. **Display Request Screen**

    4.1. When a user is clicked, a call request screen should appear

5. **Display Reject Screen**

    5.1. When there is an incoming call, a screen should appear with the options to accept or reject

6. **Display Chat Screen**

6.1. While in a call the user should be able to toggle to the chat screen to see message history

7. **Display Settings**
   7.1. A settings screen should display when requested

8. **Display Help**
   8.1. A help screen should display when requested

Build 3 (11/30/12):

Optimize Audio and Video Communication:

1. **Optimize Audio Speech Recording Settings**
   1.1. Adjust audio clip durations for optimal performance

2. **Sync Video and Audio Recordings**
   2.1. Send video and audio recordings (same set duration) at the same time to ensure that the video corresponds to the audio

3. **Sync Audio/Video and Text Translation**
   3.1. Send audio and text translation at the same time. Allow for text translation string to be received from the STT cloud.

Disgraceful Disconnect and Testing:

1. **Allow the Client to send timestamps to the server**
   1.1 Each Client will periodically send a ping to the Directory Server, indicating that the Client is still connected

3. **Have the Directory Server store the timestamps**
   2.1 The Directory Server will store the timestamp associated with each Client

4. **Configure the Directory Server to periodically check to see if the timestamps have expired**
   3.1 The Directory Server will periodically check to see if each Client's timestamp has expired. If it has, the Client is removed from the directory listing

Combine Speech-To-Text Translations:

1. **Optimize Audio Speech Recording Settings**

1.1. Adjust audio clip durations for optimal performance with the STT Cloud translation service

2. **Improve Text Translation Accuracy**
   2.1. Develop method to compare first word in current text translation and last word in last text translation to erase duplicates and replace cut-off words if necessary

User Interface Testing

1. **Initialize**
   Test 1: loading screen appears
   Test 2: screen redirects

2. **Display Connection Error**
   Test 1: graceful connection error
   Test 2: disgraceful connection error

3. **Display Directory**
   Test 1: All names appear correctly
   Test 2: List is updated periodically to reflect the correct people on the list
   Test 3: Names are easily accessed through letter tabs
   Test 4: Names redirect user

4. **Display Call Screen**
   Test 1: call works well even if one or both parties does not have video
   Test 2: messages appear correctly

5. **Display Request Screen**
   Test 1: Request screen goes away if other party disgracefully disconnects
   Test 2: Buttons correctly begin and end calls

6. **Display Reject Screen**
   Test 1: Reject screen handles disgraceful disconnects
   Test 2: Buttons begin and end calls correctly

7. **Display Settings**
   Test 1: Settings do not save in case of disconnect
   Test 2: Correct information is displayed initially

8. Display Chat Screen

Test 1: Messages show up where they need to

Test 2: Screen should scroll if there are too many messages to fit on one screen

Risks and Challenges

Client-STT Cloud

1. **Hawaii Rendezvous Speech-To-Text Service**

   1.1. Issue: STT Service may not function appropriately with unacceptable delays and/or levels of translation accuracy.
   - Solution 1: Consult Windows developer resources and reliable forums to learn how others have resolved or managed this issue
   - Solution 2: Seek alternative speech-to-text services such as Skype or the in-built service in the new Windows Phone 8 mobile operating system

2. **Hawaii Rendezvous Speech-To-Text Service**

   2.1. Issue: The service may not be able to handle continuous, sequential translation requests.
   - Solution: Incorporate a time buffer between translation request (may add significant delays from time user speaks to time text translation is received)

3. **Speech Recordings**

   3.1. Issue: Words may be cut off during the beginning or end of an audio recording
   - Solution 1: Attempt to slightly overlap the recordings and adjust the word comparison method (substantially more complicated)
   - Solution 2: Record longer audio clips with visual warnings for when recording clip will end so the user can avoid speaking then
     a. This will add significant delays between the time a word is spoken to the time the text translation is received because of the longer recording duration.
     b. This will also restrain a user's natural speech flow

Client-Server:

1. **Losing Connection**

   1.1. Issue: The Client may lose its connection to the Directory Server, which would result in a Client in the directory listing that cannot be reached
   - Solution: Keep a timestamp for each Client that is periodically updated by the Client and checked by the Directory Server. If the timestamp has expired, remove the Client from the directory listing

2. **Firewalls**

   2.1. Issue: The Client's and the Directory Server's communication is blocked by some sort of firewall.

   - Solution: Ensure that the network has exceptions in its firewall policy for both the Directory Server and the Client

3. **Multiple Clients with the same name**

   3.1. Issue: Multiple Clients with an identical name may try to register with the server

   - Solution: The Directory Server gives each client that connects a unique ID so that it can identify them. On the Client side, Clients with the same name will be listed.

4. **A Client's name is displayed on its own listing as callable.**

   4.1. Issue: A Client can see its own name in the list of callable phones

   - Solution: When the Client displays the names of available Clients to call, don't display itself.

Client–Client:

2. **Syncing of Audio/Video with Text Translation**

   1.1 Issue: The Audio/Video is not synced with the text translation strings.

   - When calling another client, it is requesting permission to open communication (sending messages, video, and audio).

3. **Memory Usage**

   2.1 Issue: The Dialer is using too much memory because of the locally stored audio/video recordings.

   - Solution: Delete the recordings as soon as they are sent to the other client.

4. **Calling a Busy Client**

   3.1 Issue: The Dialer tries to call another client who is currently in another call (their status is busy).

   - Solution: Create error handling to inform the client of their target's status.

User Interface

1. **Intuitive Design**

   1.1. Issue: The user interface on the Client needs to be intuitive.

   - Solution: Keep the number of buttons to a minimum and follow the style of similar applications.

2. Clean Style

2.2. Issue: The user interface needs to have a clean and efficient style.

- Solution: Use the default Windows Phone style guides.

Cost Estimates

Build 1

Task Description	Optimistic	Average	Pessimistic
Setup STT Cloud Prerequisites	6	12	18
Initialize Client to STTCloud communication	4	8	12
Record Audio Speech Clip	2	4	6
Send Audio Speech Clip	2	4	6
Receive a Text Translation String	1	2	3
Call Functionality	5	10	15
Request for Communication	2	4	6
Sending Chat Message	4	8	12
Receiving Chat Message	2	4	6
Send messages from Client to the Directory Server	4	8	12
Server accepts messages from client	4	8	12
Client displays current directory list	2	4	6
Display Connection Error	4	8	12
Display Directory	2	4	6
Display Request Screen	1.5	3	4.5
Display Reject Screen	2	4	6
Display Settings	1.5	3	4.5
Display Help	1	2	3
Summary of Cost Estimate:	50	100	150

Build 2

Task Description	Optimistic	Average	Pessimistic
Record Audio	2	4	6
Send Audio	2	4	6
Record Video	4	8	12
Send Video	6	12	18
Send Text Translation Strings	9	18	27
Notify Server Client is Busy	2	4	6
Notify Server Client will deregister	2	4	6
Send listing from server to all clients	4	8	12
Record audio clips continuously	6	12	18
Automatically send audio speech clips	5	10	15
Receive multiple text translation strings	4	8	12
Display connection error	2	4	6
Display Directory	4	8	12
Display Call screen	6	12	18
Display Request Screen	2	4	6
Display Reject Screen	2	4	6
Display Chat Screen	6	12	18
Display Settings	4	8	12
Summary of Cost Estimate:	72	144	216

Build 3

Task Description	Optimistic	Average	Pessimistic
Optimize Audio Speech Recording Settings	4	8	12
Sync video and audio recordings	9	18	27
Sync audio/video and text translation	6	12	18
Allow the client to send timestamps to the server	4	8	12
Have the directory server store the timestamps	2	4	6
Server should check if timestamps expire	1	2	3
Optimize audio speech recording for STT	6	12	18
Improve text translation accuracy	4	8	12
Test Initialization	2	4	6
Test Connection Errors	2	4	6
Test Directory	2	4	6
Test Call Screen	4	8	12
Test Request Call	2	4	6
Test Reject Call	2	4	6
Test Setting Updates	2	4	6
Test Chat Screen	4	8	12
Test translation Accuracy	2	4	6
Test call quality	4	8	12
Summary of Cost Estimate:	62	124	186

Totals

Builds	Optimistic	Average	Pessimistic
Build 1	50	100	150
Build 2	72	144	216
Build 3	62	124	186
Total Cost Estimates	184	368	552

Gantt Chart

Key

▩	Client - STT Cloud
▨	Client - Server
■	Client - Client
▦	User Interface

	Build 1			Build 2		Build 3	
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Setup STT Cloud Prerequisites	▩						
Initialize Client to STTCloud communication	▩						
Record Audio Speech Clip		▩	▩				
Send Audio Speech Clip		▩	▩				
Receive a Text Translation String			▩				
Call Functionality	■	■	■				
Request for Communication	■						
Sending Chat Message		■	■				
Receiving Chat Message			■				
Send messages from Client to the Directory Server	▨						
Server accepts messages from client		▨					
Client displays current directory list			▨				
Display Connection Error	▦						
Display Directory	▦	▦					
Display Request Screen		▦					
Display Reject Screen		▦					
Display Settings			▦				
Display Help			▦				
Record Audio				■			
Send Audio				■			
Record Video					■		
Send Video					■		

Send Text Translation Strings					■		
Notify Server Client is Busy				■			
Notify Server Client will deregister				■			
Send listing from server to all clients					■		
Record audio clips continuously				■			
Automatically send audio speech clips					■		
Receive multiple text translation strings					■		
Display connection error				■			
Display Directory				■			
Display Call screen				■	■		
Display Request Screen				■			
Display Reject Screen					■		
Display Chat Screen					■		
Display Settings					■		
Optimize Audio Speech Recording Settings						■	
Sync video and audio recordings						■	■
Sync audio/video and text translation							■
Allow the client to send timestamps to the server						■	
Have the directory server store the timestamps						■	
Server should check if timestamps expire							■
Optimize audio speech recording for STT						■	
Improve text translation accuracy							■
Test Initialization						■	
Test Connection Errors						■	
Test Directory						■	
Test Call Screen						■	
Test Request Call						■	
Test Reject Call							■

Test Setting Updates							■
Test Chat Screen							■
Test translation Accuracy							■
Test call quality							■