Abstract
The following manuscript introduces a revolutionary technical aid for communication between those who are sensory impaired and those who are not. This software's innovation is capable of receiving voice input and rapidly returning both the Braille and Sign Language equivalent using Amazon Web Services Cloud Technology.

Key Words: American Sign Language (ASL), Braille, Healthcare, Cloud Based Development

1. Introduction
Oral communication is the foundation of most relationships, a method for sealing business deals, and educator's primary mode of teaching. Those who are hearing impaired are highly disadvantaged in our culture which is centered on the spoken word.

The education system ordinarily delivers its lessons through auditory means, leaving deaf students separated from their peers, or requiring them to be accompanied by a paraprofessional who can effectively relay the lesson to the students using sign language. These students are forced to rely on the accuracy of the translator in order to “hear” and understand the lesson being delivered.

In the healthcare setting, communication between the deaf and the hearing is much more crucial [1]. Some hospitals do not have translators for the hearing impaired, and even fewer doctors' offices and pharmacies are equipped to converse with those without hearing. Often a hearing impaired person is left to fumble with pen and paper when seeking medical attention or picking up their prescribed medications.

Even the most successful of hearing impaired individuals struggle in the workplace. One deaf building industry worker relayed: “I think some of the frustrations I have experienced in the past have been around not passing on information…people (are) always (saying) I'll tell you later and never do… And I hate it when people (who) want to get your attention (get so) frustrated that you can't hear them shouting at you, (that) they resort to throwing things at your head” [2]. Another person had a different experience; “we were divided up into teams of six and I was the only deaf person out of around 30 hearing people. At times I would stop working and ask others (through pen and paper) what was being said in the office. I would always get in trouble for this and would be accused of not getting my work done” [2].

In this project, Words Now Spoken, the hearing impaired community is empowered. My technological medium allows opportunities for open communication between the deaf community and the rest of the world. Words Now Spoken takes the words spoken by those of the hearing community and quickly translates them to the American Sign Language (ASL) signs that are recognized by the deaf community and the 64 character Braille alphabet Braille users rely on. Words Now Spoken provides additional functionality for blind users VIA a text-to-voice function.

This software is completely portable on smartphones and laptops. This program is easy to use and is a huge step in bridging the communication gap between the deaf community and their teachers, healthcare providers, and coworkers.

2. Design Process
To appreciate the merit of Words Now Spoken we must first pay tribute to its technological predecessors. In 1952 “Bell Labs developed the first effective speech recognizer” [3]. Just nineteen years later (1971), Bell laboratories shocked the world once again with a text-to-speech generator. “The New York Times, in describing the machine's operation, ranted 'My God, it talks'” [4]. The first English-to-Braille translator was developed by IBM and the American Printing House for the Blind (APH) and dates back to 1961 [5-6]. The APH expanded on their previous development in June of 1969 when software running on the IBM 360 computer efficiently proved capable of generating both musical and mathematical Braille [7].

In 2013, PhD. Mohamed Jemni stated that “80% of deaf people in the world do not have access to education” [8]. He explained further, “the disability is not the problem, the accessibility is the problem” [8]. Dr. Jemni purposes
his lab at the University of Tunis to overcome these barriers. Professor Jemni “won the UNESCO Prize in 2008 for (his) e-learning curriculum for (the) visually impaired and (he also won) the World Summit Award (WSA) Mobile 2010 in the field of social inclusion” [8].

Words Now Spoken’s contemporaries include: SigningSavvy.com’s American Sign Language Video library [9], Microsoft’s EnableTalk [10], Google’s “augmented reality Glasses” [11], Nanopac’s Duxbury Braille Translation Software [12], and IBM’s SiSi [13].

SigningSavvy.com functions primarily as an internet dictionary for sign language. It does not allow for vocal input or the looking up of more than one word at a time. This makes SigningSavvy.com a helpful resource but not widely usable for daily communication [9].

Nanopac’s Duxbury Translation Software costs $295.00 and is not easily implemented on mobile devices [12]. It has excelled at interpreting the web and local environment into Braille, but is not feasible for many of the underprivileged.

IBM’s SiSi technology works to translate text and voice to sign language using a cartoon person who signs the words, truly an amazing feat [13]. But the problem with most of these designs is their reliance on local computing technologies.

For this reason JavaScript and HTML5 were selected making for a completely web-based design. We believe utilizing cloud computing will enable Words Now Spoken to be the solution Professor Jemni was working towards: a scalable, portable, affordable, web-based solution. Memory savings is one of hallmarks of this innovative application. Through virtualization, the substantial initial memory download is avoided, only what is necessary for runtime is loaded to the device.

The high level description of Words Now Spoken contains the following four main features:

1.  voice-to-text
2.  text-to-braille
3.  text-to-sign
4.  text-to-speech

Coding was performed using C++, HTML5, and JavaScript. Braille, Sign, SpeechClient, SpeechGenerator, and SpeechWorker.js libraries were leveraged to inform the essential design functionality.

Figure 1, shown to the upper-right, illustrates the dependencies of the text-to-braille, the text-to-sign, and the text-to-speech conversions on the voice-to-text function retrieving a JavaScript text string.

3.  Implementation

Google’s Speech API is embedded in the chrome browser allowing the programmer a technological advantage in durable and accurate voice-to-text translation by utilizing Google’s voice analytics on the server-side. This line of code achieves the voice-to-text conversion:

```html
<input x-webkit-speech id="micro_phone"/>[14]
```

Creating a text box with an id identical to the input field (i.e. micro_phone) stores that input into its corresponding text box (as seen in the results and demonstration section). Next, the text box input is converted to a JavaScript (JS) variable using the command:

```javascript
vardspeechInput=document.getElementById('micro_phone');
```

The sequential diagrams for the other functions are given from the client-side perspective; in each case, the user presses the properly labeled button to initiate their desired action (Figures 2-4).
At this stage, steps one through five in Figure 1 have been completed.

To accomplish the text-to-braille, text-to-sign, and text-to-speech operations the JavaScript input variable was passed to the following sub-functions (illustrated in Figures 5-7):

\[
\text{Braille (speechInput.value); \ Braille Conversion}
\]
\[
\text{Sign (speechInput.value); \ Sign Conversion}
\]
\[
\text{Speak (speechInput.value); \ Speak Back function}
\]

### 3.1 Library Construction

**Speak.js**

The Speak.js library was transcribed from a C++ speech synthesizer called eSpeak using the Emscripten “transcompiler.” It is important to note that C++ is only functional locally, whereas the JavaScript Speak.js library is web accessible.

There are three separate functions that contribute in the Speak.js library:

1. Speech Client
2. Speech Worker
3. Speech Generator
The speech client calls the speech worker who interacts with the speech generator and subsequently returns a WAV file to an html audio element (<div id="audio"></div>). From that audio element the WAV file is played (as seen in Appendix Figure 8) [15].

The Speak.js library can function in a mode without the speech worker, yet compensation must be made by setting the objects option setting to true WAV generation code in the speakGenerator.js must be called directly [15].

---

**Fig. 5 : Text-to-Braille Function Call**

1. **Text-to-Braille**
2. Start
3. User Presses
4. `Braille (speechinput.value);`
5. Braille Displayed
6. End

**Fig. 6 : Text-to-Sign Function Call**

1. **Text-to-Sign**
2. Start
3. User Presses
4. `Sign (speechinput.value);`
5. Sign Language is Displayed
6. End

**Fig. 6 : Text-to-Voice Function Call**

1. **Text-to-Voice**
2. Start
3. User Presses
4. `Speak (speechinput.value);`
5. Voice is Generated
6. End
Braille.js

The Text-to-Braille function paired with the Braille.js library operates via an if/else loop that cycles through the entire alphanumeric spectrum (A-Z, a-z, 0-9) and punctuation (.,;:). When the selected character matches the coding compare statement, then that character's corresponding Braille picture is outputted to the screen. The braillepix folder on Amazon Web Services yielded these images to the Text-to-Braille function. An overview of the Braille.js loop construction is contained in Figure 9 (Appendix).

3.2 Coopting Braille.js into Sign.js

The Braille.js library was altered to create the Sign.js library. This process required changing the file return reference from the braillepix folder to the signalphgif folder in addition to storing a library of ASL “sign” pictures in the new reference folder.

Sign.js folder reference coding is as follows:

```
return ["<IMG HEIGHT=60 WIDTH=89 vspace=5 hspace=1 SRC="signalphgif/", bPix, ".gif", ALT=","
, bAlt, ",","">"].join(""");
```

American Sign Language images corresponding to each letter were loaded to the S3 bucket folder with names, which directly corresponded to the Braille images used [16]. This maneuver allowed the loop coding structure to remain homogenous. Moreover, this design structure enabled identical character names and compare statements, but different folder returns.

Implementation issues were minor throughout the design process, which we attribute to the storage virtualization that has made the 'cloud' so helpful for developers. We were able to upload Braille and Sign photographs to the braillepix folder and Sign folder in the AWS S3 bucket, effectively emulating our local development environment seamlessly in the 'cloud.'

4. Results & Demonstration

1. User enters application.
2. Predefined meSpeech function greets the user saying, “Welcome, please press the microphone at the top of the screen, and then speak!”
3. User clicks microphone icon()
4. User responds to the “Speak now” request displayed on the screen and speaks into the computer's built-in microphone

6. Text is received and stored in the “microphone” textbox below.

At this juncture the user has a variety of options:

6a. Continue inputting data vocally by pressing the microphone icon()
6b. Initiate Text-to-Braille conversion by pressing the Text-to-Braille bottom.
6c. Initiate Text-to-Sign Language translation by pressing the Text-to-Sign button also located on the left side of the screen.
6d. Have received words spoken back by pressing the Text-to-Voice bottom.

5. Conclusion

Words Now Spoken takes spoken words and translates them to written words, Braille characters, and American Sign Language. This program is simple to operate, only requiring the Google Chrome browser, Java Script Comaptibility, and a microphone connection. The greatest hurdle in this project was finding a voice-to-text converter.
Future implementations may allow local Braille and Sign output downloads, web independent voice-to-text translation, and text security scanning tools running simultaneously.

The Words Now Spoken application provides a streamlined alternative to paper and pen communication. One pharmacy representative I spoke with noted that her pen and paper conversations with her deaf clients are often “shorthand and abrupt” [17]. She explained that important prescription drug details are commonly inadequately communicated due to the language barrier between the deaf patient and the hearing pharmacy technicians [17]. She proceeded to tell me about many details that are frequently omitted due to time pressures and to just for the sake of keeping things simple with deaf patients.

In the health care environment, this cursory and inadequate level of communication can lead to untreated conditions or even misdiagnosis because all symptoms might not be effectively stated during the patient consultation. Also, patients who have questions about their medication might not ask their doctor because of the burden of effectively stating their concerns or questions [17]. Likewise, a health care provider might exclude details about a diagnosis or drug to the patient they are treating because of the time and effort it would require to write it all out. “If deaf people cannot communicate with their doctor, how can they receive support when they have a mental health problem such as depression?” [17]

Using Words Now Spoken, the deaf will be able to effectively communicate with their doctors, pharmacists, and coworkers. They can sit in a classroom with thirty other “hearing” students and not miss a word. We believe Words Now Spoken is a revolutionary technology that utilizes the Cloud to empower the underserved [18].

Looking forward to the future, Words Now Spoken is just a starting point, with much room for further research. It would be beneficial to develop the program to incorporate the use of a camera. This feature would then be able to bridge the communication gap between the deaf and the blind. The deaf person could choose to sign into the program and the program would speak to the blind person. In like manner, the blind person could speak into the program, and the voice input would be signed to the deaf person!

6. Appendix

![Fig. 8: Speak.JS Process](image)

![Fig. 9: Braille.js Construction](image)
References

[1] Dimension 5 Patient-Related Factors., (Adult Meducation) [online] 2006, http://www.adultmeducation.com/PatientRelatedFactors.html (Accessed: 22 April 2013)

[2] What Deaf people and hard-of-hearing say about work, Question and Answer, Deaf Resource, [online] 2007, http://www.deafresource.com/09_7.html (Accessed 22 April 2013)

[3] "The History of Voice Recognition," (Time Toast), [online] 2010, http://www.timetoast.com/timelines/the-history-of-voice-recognition (Accessed 22 April 2013)

[4] Text-To-Speech Synthesis, Alcatel Lucent, [online] 2013, (Accessed 22 April 2013)

[5] J. Sullivan, "Early History of Braille Translators and Embossers," (Duxbury Systems), [online] 2013, http://www.duxburysystems.com/bthist.asp (Accessed 22 April 2013)

[6] Braille translation for the IBM 704, World Cat, [online] http://www.worldcat.org/title/braille-translation-system-for-the-ibm-704/oclc/693568975 (Accessed 22 April 2013)

[7] A. Schack, "Computer-Translation: Grade 2 Braille from Print," (Duxbury Systems), [online] 1969 http://www.duxburysystems.com/downloads/library/history/schack_1969.pdf (Accessed 22 April 2013)

[8] Automatic translation of text to sign, Deaf News Today, [online] 2013, http://deafnewstoday.blogspot.com/2013/02/automatic-translation-of-text-to-sign.html (Accessed 22 April 2013)

[9] Your Sign Language Resource, Signing Savvy, [online] 2013, http://www.signingsavvy.com/(Accessed 22 April 2013)

[10] Enable Talk, [online] 2012, http://enabletalk.com/ (Accessed 22 April 2013)

[11] J. Brodkin, “Google Glasses patent hints at speech-to-text display for deaf users”, (arstechnica) [online] 2012, http://arstechnica.com/information-technology/2012/05/google-glasses-patent-hints-at-speech-to-text-display-for-deaf-users/ (Accessed 22 April 2013)

[12] Ordering Information and Domestic Price List, Duxbury Systems, [online] 2012, http://www.duxburysystems.com/prices.asp (Accessed 22 April 2013)

[13] S. Tomasco, “IBM Voice-to-Sign Language Translation Tech", (youtube) [online] 2007, http://www.youtube.com/user/RFIDdotcom?feature=watch (Accessed 22 April 2013)

[14] F. Schulz, "SPEECH TO TEXT LIBRARY FOR PROCESSING", (get flourish), [online] 2011, http://stt.getflourish.com/(Accessed 10 March 2013)

[15] A. Zakai, Speak.js, [online] 2011, https://github.com/kripken/speak.js/blob/master/AUTHORS (Accessed 22 April 2013)

[16] J. Strickland, "How Sign Language Works," (how stuff works), [online] 2013 http://people.howstuffworks.com/sign-language2.htm (Accessed 19 April 2013)

[17] A. Kremer, (Senior Technician), Walgreens Pharmacy, Primary Resource, (Accessed 19 April 2013)

[18] Sign Health, “Deaf people and suicide”, (sign health) [online] 2006 http://www.signhealth.org.uk/documents/suicide_leaflet_v8.pdf (Accessed 22 April 2013)