**Audio Situational Awareness Requirements Document**

# **1. Introduction**

## **1.1 Overview**

Imagine what it would be like to watch television without sound or to watch an inaudible movie! Now try to imagine a lively area filled with the sounds of laughter, roaring music and the chirping of the birds. Yet, you are not able to hear anything! Yes, this is the scene and the anguish that a deaf and Hard of Hearing (DHH) person has to face every day in their life. It is indeed tough to imagine life devoid of sound. Silence does have a way of making even the busiest of the scenes seem still. Thus in the pursuit to bring about some ease in their day to day lives and to bridge the gap between the ones that can hear and the ones that cannot, we want to build the Audio Situational Awareness (ASA) device that would provide a DHH the independence to communicate and participate freely in social gatherings.

**1.2 Scope of the Product**

The *Audio Situational Awareness (ASA)* involves working with a hardware platform i.e the ReSpeaker which captures the speech, and extends it to work with the Google Translation APIs that provides the speech to text translation feature. The speech is then displayed in the form of text in an android based application. The ASA also provides features like text translation, transcript persistence in the form of a file and transcript sharing via other messaging tools. The ASA application would initially support only android devices. We would be working on extending the support from Android to multiple platforms as a future prospect of the product. The application is built on android studio using java/python as the programming language and HTML, javascript will be used in front end.

## **1.3 Business Case for the Product**

Being a Deaf and Hard of Hearing person, it can be an uphill struggle as dealing with hearing loss is piled on top of the usual trials and tribulations in daily life. At every phase, they deal with problems in communicating with friends, in the classroom and even in a social gathering. Vu LLC is an organization that is working to build communication tools to help the Deaf and Hard of Hearing persons live and communicate like everyone else. In collaboration with Vu LLC, we, the students of the University at Buffalo are intending to build a product that will give freedom to the DHH. It will enable them to actively engage in and follow group discussions in the community, classroom, and workplace. The device will monitor the surrounding audio environment for speech sources up to a range of 30 feet in a classroom environment, locate the direction of the source relative to the device and convert the speech to text using an Android Application. The user can also download the transcript of the speech to text translation for future use.

# **2. General Description**

# **2.1 Product Perspective**

The *Audio Situational Awareness (ASA)* device is intended to help the Deaf and Hard of Hearing (DHH) to participate freely in group discussions such as in the classroom, workplace and at other social gatherings. The device enables the user to know when someone is speaking, the content of the discussion and the location of the speaker. The ASA consists of a hardware device that listens for a speech source, records the speech and displays it to the user’s personal device in the form of text. The product also allows the user to choose a language of their preference to view the transcript in. The Audio Situational Awareness Project is being developed by the students of the University at Buffalo in collaboration with Vu LLC. The inspiration behind developing ASA is not only to help the DHH but also to help users in need of speech translations in a multilingual setting.

## **2.2 Product Functions**

The ASA is a collaborative of ReSpeaker and an android application.

ReSpeaker consists of the following functionalities:

* **Listen Mode**:
	+ In this mode, the ASA device (ReSpeaker) monitors the surrounding environment for speech sources and aims at detecting speech upto a range of 30 feet in a classroom environment.
	+ The ASA will capture the speech based on the location of the speaker and how the speaker moves relative to the device.
	+ For each speech source, the ASA will convert it to text and start processing the speech when the device is powered up.
* **Record Mode**:
	+ The ASA will start recording as soon as a successful connection is established with the android application.
	+ The ASA will record an active speech for a fixed period of time.
	+ If a fixed period of silence is detected, it will stop recording and go back to the listening mode.

An android application will enable the user:

* To connect to the ReSpeaker.
* Get the speech data via bluetooth.
* Display it to the user in the form of a text message on the smartphone display.
* To save, download and share the transcript file.

## **2.3 User Characteristics**

The ASA device is intended mostly for the Deaf and Hard of Hearing (DHH) because it will enable them to capture the voices from the surroundings and translate them to text messages. Besides them, the device can also be used by others as a translator to capture voice in any language and translate them to their desired language in the form of a transcript. The end user is expected to have prior knowledge about using a smartphone and basic idea about how to create an user account on an online application.

## **2.4 General Constraints**

The ASA in general has the following constraints:

* The ASA is currently paired with the ReSpeaker Core v2.0. So, to get the ASA to work we would require a ReSpeaker Core v2.0 to be present and working.
* The ASA application that allows the user to connect with the ReSpeaker via Bluetooth is currently planned to support only to android devices.
* The ASA may support multiple language translation based on the flexibility of the Google Translation API and other existing python libraries.

## **2.5 Assumptions and Dependencies**

**Assumptions:**

* We assume that initially, every user will speak in English and we will support only one language translation (later we can work on supporting multiple languages).
* We assume that the end-user will be comfortable using the Android App. The App will be responsive and support all device configurations.
* We assume that the end user will have a Google Account to register to the App.
* We assume that the user will connect the ReSpeaker to a power source (later to make it portable we can use an internal battery as a standalone power source).
* If there is a way to distinguish between collocated speakers, they may be assigned individual identifiers.

**Dependencies:**

* We would require GCP credits to work on the Google Translation API.
* We will check for APIs to integrate that will enable multiple language support.
* A stable power source is required for the functioning of the ReSpeaker device.
* The end user should have an Android device to install and work with the ASA application.

# **3. Specific Requirements**

## **3.1 User Requirements**

The ASA should adhere to following user requirements:

* The end user should be able to create an user account on the android application and then connect their respective android devices to the ReSpeaker.
* Everytime the user wants to connect to the ASA, he/she has to login to his account using his credentials.
* The user will have the privilege of downloading and saving the transcript for future use.
* The user will be able to forward the same to the other users through email or any other online messaging services (e.g. Whatsapp, WeChat etc).
* The user will have the privilege to view the translated message in the desired language.
* The user can also select the language in which he/she wants to operate the app.

## **3.2 System Requirements**

The System requirements are listed as below:

* The ASA will use the ReSpeaker Core vers. 2.0 as its hardware platform.
* Python/Java will be the programming language for the ASA.
* Google Translation API will be used to convert text to speech.
* The text will be displayed over a message through an Android App.
* The ASA will track active speech sources by location and show the location by lighting one of the six LEDs closer to the location of the speaker.
* Multiple speech sources collocated within 10 degrees of each other shall be considered one speaker.
* ASA will start processing speech when the ReSpeaker is powered up.

## **3.3 Interface Requirements**

The end user would have an android application, that would have the capability to connect with the ReSpeaker via Bluetooth. The user interface should have the following capabilities:

* The user would be presented with a login screen to enter into the android application.
* Once connected, the user would have the capability to control the ReSpeaker via the application (running on the android device that is connected with the ReSpeaker via Bluetooth).
* The home screen would display the transcript received from the ReSpeaker, a general ID and the direction of the speaker will also be displayed.
* A ‘Save Transcript to File’ option would be provided that would allow the user to save a speech transcript to a file.
* A ‘Delete Transcript File’ feature would be provided that would allow the user to delete any previously saved transcript file from the device memory.
* A ‘Reset’ option would be provided that would allow the user to delete all the pre-existing transcript files from the device memory.
* A ‘Share Transcript’ feature would allow the user to share transcript files from their android device via other messaging applications or email.
* The user will have the ability to select the language they want the transcript to be translated into.
* A language filter would be provided within the application to help the user change the application’s display language.

# **4. Appendices**

*If you wish to append any documents, do so here. You may wish to include some or all of the following:*

· *Personas and scenarios developed for this project*

· *Transcripts of user interviews, observations, or focus groups*

· *Copies of communications which contain user requirements*

· *Original project proposals or other historical documents*

· *Lists of similar projects or products, with notes about how they differ from yours*

· *A list of requirements which were "wish-listed" or marked unfeasible at present*

· *Original screen mockups, if they are relevant*

# **5. Glossary**

*Include a glossary of definitions, acronyms, and abbreviations that might be unfamiliar to some readers, especially technical terms that may not be understood by end-users or domain-specific terms that might not be familiar to developers.*

# **6. References**

*List references and source documents, if any, in this section.*