ELEC/COEN 390

Software Design Application - Sign Sound See
Scrum Report

Submitted to :

Dr. Kharma

October 29th, 2017

Team Name: The Best

By:
Cameron Hall - 40004827
Yazan Odeh - 26556191
Yousef Abo El Foul - 27199120
Nicholas Gattuso - 40007087

Concordia University

ABSTRACT

This report discusses the ELEC/COEN 390 course project of designing a glove that translates sign language via an Android application. The report discusses our current User Stories in our Product Backlog and the tasks in our Sprints. It discusses the goals we have for Sprint 1 is to create a simple version of the application that can output the letters in either a text or voice page and to have a glove with flex sensors that can read the hand gestures relating to the American Sign Language. In Sprint 2 we switched the Arduino to the BLUNO so that we may be able to use the app to communicate with the hardware and fetch the letters from the glove readings. In addition, we plan on to create the functionality to create words, create a tutorial section and calibration section. We are currently on Sprint 3 with the plan to switch our Arduino to the BLUNO so that we may be able to use the app to communicate with the hardware and fetch the letters from the glove readings. In addition, we plan on to create the functionality to create words, create a tutorial section and calibration section.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

American Sign Language is a natural language and is the main sign language of the Deaf and impaired communities in the United States and most of Canada. English words are often borrowed through fingerspelling, which is the representation of the letters of a writing system using only the hands. This alphabet has often been used in deaf education, mainly that of young children. Figure 1 illustrates what is known as the American Manual Alphabet and illustrates the symbols that we will be following in the configuration of our project.

The purpose of this project is to design a glove that will be able to translate fingerspelling via an Android application with the goal of being used as a tool for deaf education and for the general public to understand sign language from a user.

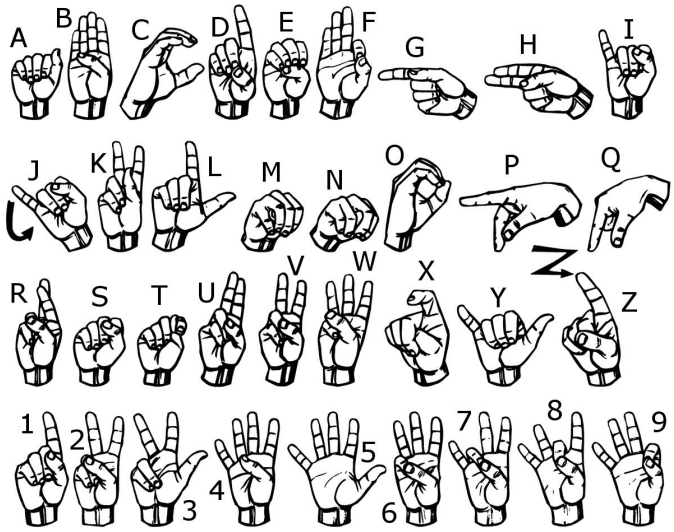

Figure 1: American Manual Alphabet [1]

1.1 DOCUMENT REPRESENTATION

This report discusses the design on our project, our current Product Backlog, including the User Stories, and the tasks for our Sprints. Furthermore, it explains the goal for our upcoming Sprint and what we expect to have completed at the end of the next two weeks. The following two sections explain the requirements of our project that should be covered in the Product Backlog and in the Sprints.

1.1.1 Glove Requirements

Our requirements for the glove consist of having it recognize gestures and hand movements. The glove should also be able calibrate itself for each user. The glove should have the ability to recognize the bending of each finger in relation to the other fingers in order to then associate the series of bent fingers to a standard American Sign Language letter. Finally, the glove should be able to transmit this information to the app.

1.1.2 Application requirements

The application should prompt the user to undergo a calibration setup for the glove. The app will store the calibration information for each individual user. The app will receive and store the letters transmitted by the glove. From there, the application should output the letters by displaying them as text and voicing them as sound. The app will include a tutorial section, which will act as a reference to teach the user the American Sign Language alphabet. Finally, the app will have a settings section that will enable the user to calibrate the glove in the case where it was originally miscalibrated. The settings will also enable the user to choose an output preference (voice or text).

2.0 DESIGN (V.1)

2.1 WIREFRAME-DESIGN

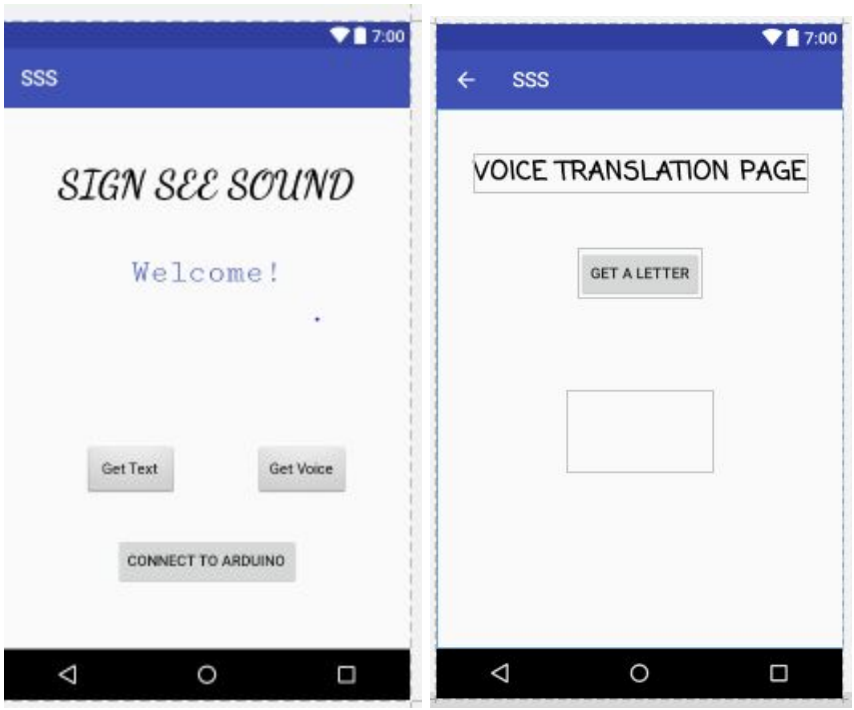

Figure 2: Screenshots of SSS Application in its Current Version

The basic interface of our application is shown in figure 2. If we click on “Get Voice”, we will be directed to the page where we receive the motion as a voice. Moreover, in the upcoming Sprints, we're going to focus more on applying the tutorials for the alphabets and the user interface.

2.2 SYSTEM ARCHITECTURE

The architecture of our product includes an Arduino which acts as a database where the user will send data to the Android application for further processing.

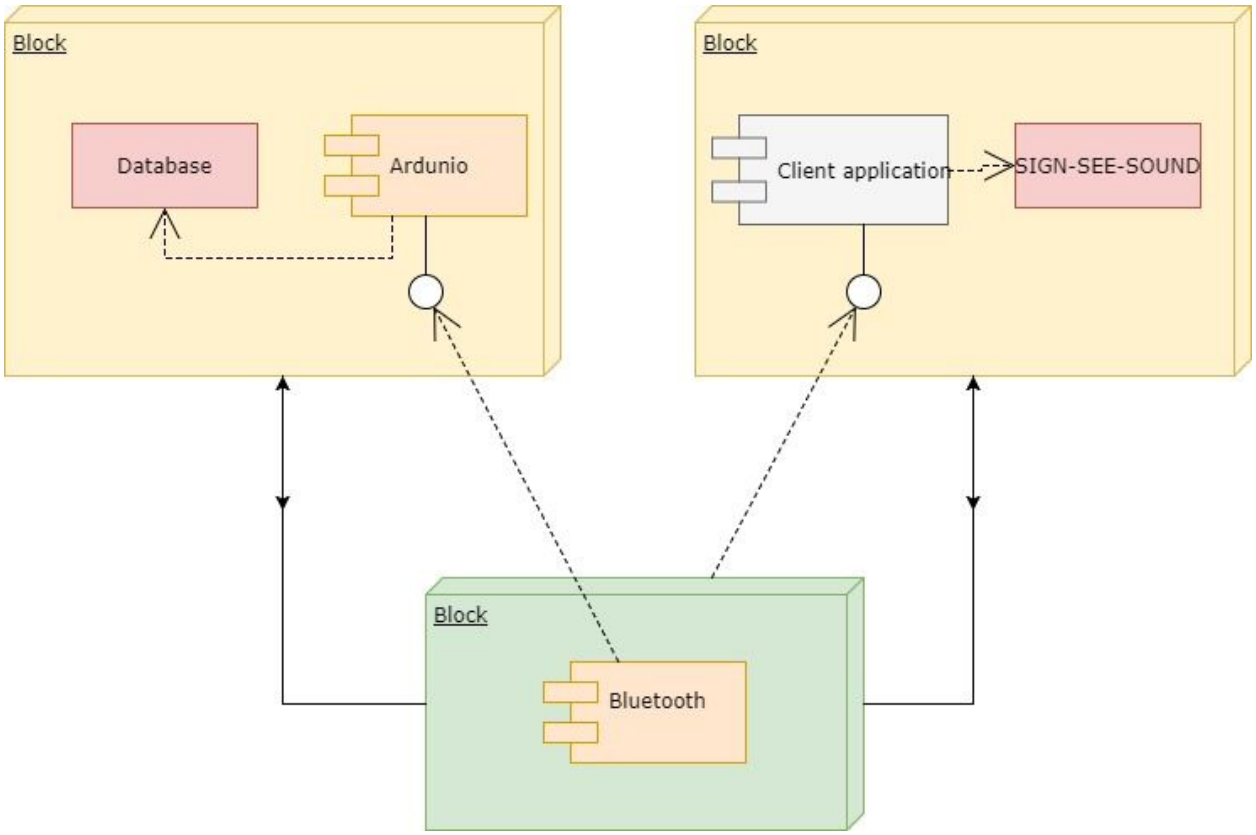

Figure 3: System Architecture Diagram

2.3 HARDWARE ARCHITECTURE

For this project, we have the five flex sensors connected to an Arduino. We have an accelerometer to measure the motion on the x-axis and the y-axis. In the future, we plan on implementing the z-axis of rotation in order to get Sign Language Motions as well. For the sake

of simplicity, this project will be focusing more on using two axes instead of three axes. We have a BLUNO Arduino in order to connect the application and sensors wirelessly through a bluetooth module contained in the BLUNO.

2.4 SOFTWARE ARCHITECTURE

For the software part, the programming is done in both the Arduino and the application. For the case of the Arduino, we define a calibration for the sensors and the accelerometer in order to have a set of user defined constraints. For the letters, we get the values of the sensors and the motion of the accelerometer and output the associated letter on the application. The Android application is where data is received from the Arduino via bluetooth to either display the letter as text or as a sound.

4.5 USE CASES AND SEQUENCE DIAGRAMS

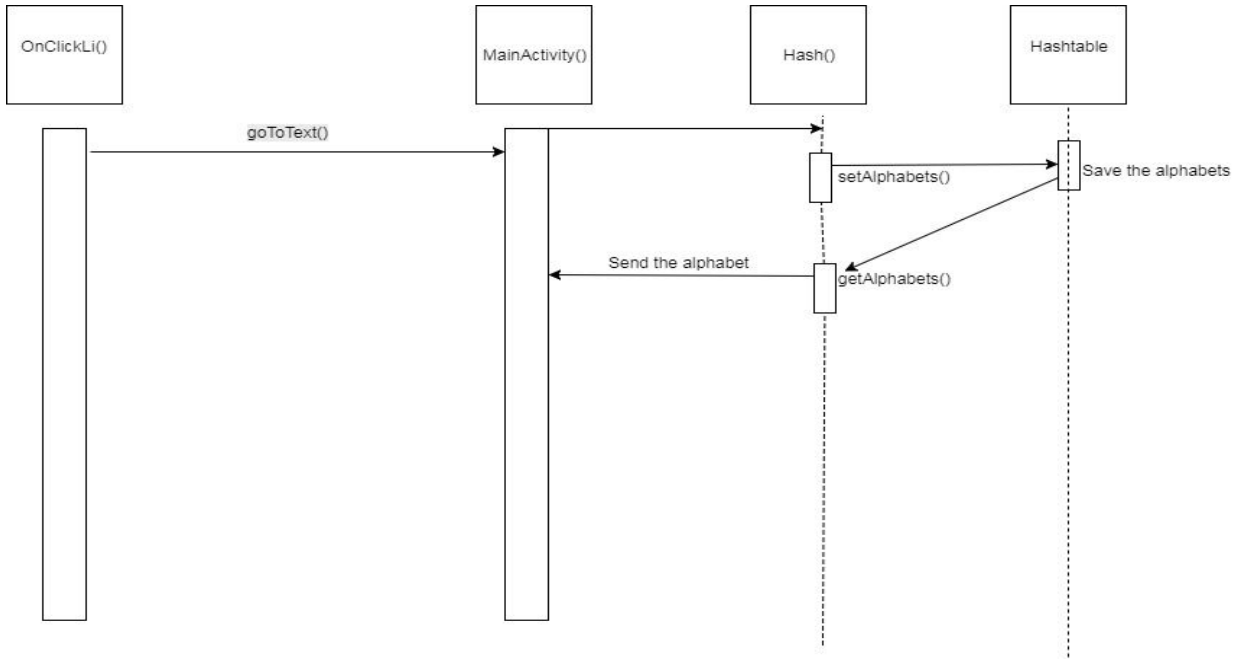

Figure 4: Sequence Diagram

Our system has a hash table that contains the letters of the alphabet. Once a letter is called using the glove, we'll get the alphabet from the hash table and send it to the MainActivity, which is basically the User-Interface(UI).

3.0 SCRUM #0

In this section, we will be discussing our previous Product Backlog and the tasks in our Sprint 1. It is divided in two sections, respectively covering the two topics.

3.1 PRODUCT BACKLOG

Our Product Backlog contains User Stories that we've obtainED and some functionality for our application that we would like to work on. Table 1 is an excerpt from our Product Backlog document.

Table 1: Product Backlog during Sprint 1 [2]

Story ID	Story Title	*Card*	Story Points(1-10)
UI-1	User Buttons	*As a user, I want to have two buttons so that when clicked one will get you to the sound translation section and the other will get you to the textable translation section of the sign language*	3
UI-2	Text variation	*As a user, I want to have the ability to see the text so that I can understand what the sign that i just did mean*	5
UI-3	Sound variation	*As a user, I want to have the ability to hear the sign done so that I can have the accessibility of learning while listening*	7
H-1	Glove Functionality	*As a user, I want to be able to use a glove that can read sign language gestures*	7

AA-1	Arduino to Android	*As a developer, I want to have an immediate connection between Arduino and Android so that i can get information from the Arduino to be placed in Android*	9
I-1	Internal functionality	*As a user, I want to be able to go back from a specific page setup to the homepage of the application so that i can switch variation(sound or text)*	3
D-1	Database	*As a user, I want to have the letters saved in the database so that we can fetch for the letter called from the database*	9
UI-3	User Learning Functionality	*As a user, I want to have the letters shown so that when clicked on one will be directed to a video explaining how to do the letter clicked*	6
UI-4	User Format Control	*As a user, I want the action bar to contain a setting that switches between voice and text so that it will be an easy access*	5

In the table, we have six different type of Story IDs; “UI” relates to the user interface features, “H” relates to the hardware, “AA” represents the connection between the Arduino and the Android Application, “I” relates to the internal functionality of the App, and “D” represents a database story. Every user story currently in the Product Backlog (except UI-3) will be covered in our first Sprint. The following sections explain the result behind the choices for our first Sprint.

3.2 SPRINT #1

Our goal for Sprint 1 is to create the communication setup between Arduino and the Android application. Moreover, we will focus on creating a simple version of the User Interface (UI). We will create buttons that will take us to different activities, such that we have an activity that associated with receiving the translation in text format and on the voice translation for the American Sign Language. In addition to creating the buttons, we will setup and populate the database. The database will contain all the information of the letters that are programmed using the Arduino so that we may be able to find the proper letter received from the Arduino's signal. On the side of the hardware, we are sticking with the formats that are associated with the code for the sake of having the accurate functionality for our glove. To do so, we are putting the circuit together and assuring that the programming is giving us proper readings.

Using our goals for Sprint 1, we populated Table 2 with the task breakdown relating to the stories in the Product Backlog.

Table 2: Sprint 1 Backlog [2]

Story ID	Task ID	Task Title	Task Description	Ideal Hours
UI-1	UI-1.1	Create 'Receiving Text' page	Create a page where the received signal from the hardware will output a the respective alphabet text	0.2
	UI-1.2	Create 'Receiving Voice' page	Create a page where the received signal from the hardware will output the respective alphabet through a voice	0.2
	UI-1.3	Create Button to go to Text Page	Create a button in the main page called "Get Reading in Text"	0.2
	UI-1.4	Create Button to go to Voice Page	Create a button in the main page called "Get Reading in Voice"	0.2
	UI-1.5	Link the buttons to their respective page	Connect the buttons to their respective pages. When the buttons are pressed, the program should go to the respective page	0.2

UI-2	UI-2.1	Fetch the received signal from the Arduino in the app database	Look up in the database for the value that's associated to the received value from the Arduino	1
	UI-1.2	Output the return value from the database	Output the value returned from the database in an EditText	0.5
UI-3	UI-3.1	Fetch the received signal from the Arduino in the app database	Look up in the database for the value that's associated to the received value from the Arduino	1
	UI-3.2	Output the return value from the database	Output the value returned from the database as a Voice output	1
H-1	H-1.1	Glove Creation	Attach the sensors on a glove so that the user can be able to perform sign language	1
	H-1.2	Arduino Program	Download the Arduino program to the Bluetooth Arduino (BLUNO)	0.5
	H-1.3	Connect Glove with Arduino	Attach the sensors to the accelerometer and the accelerometer to the Arduino in order to get the two components to function	1
	H-1.4	Test the Glove Functionality	Test the program to make sure that the reading of the sensors is giving a value	2
AA-1	AA-1.1	Connection between Android and Arduino	Import the Arduino library in the Android application. Use the necessary functions to link the application and the Arduino together through a bluetooth signal	3
I-1	I-1.1	Back Button	Add the Up Navigation in the application so that the users can go back to home page	0.5
D-1	D-1.1	Database Creation	Create a database with one field being the values that will be received from the Arduino and another field being the associated alphabet text	2
	D-1.2	Database Fetching	Create a function that will fetch the alphabet letter from the database according to a received input from the Arduino	3

4.0 SCRUM #1

In this section, we will be discussing our quick thoughts about how Sprint 1 went, our current Product Backlog and the tasks in our Sprint 2. It is divided in three sections, respectively covering the three topics.

4.1 SPRINT 1 REVIEW

During the last two weeks, we have been able to create a working application that can route to different pages where we can get letters in a TextView, get letters through voice output, and where we can view an image with the American Sign Language. The hardware has been put together; the sensors and the accelerometer have been attached to the Arduino and have returned the expected readings as formulated by the user. Additionally, we managed to attach the sensors to a glove. During the Sprint, we agreed that setting up the bluetooth is a challenging and long task. For that reason, we ended up focusing on setting up the app and Arduino with their bluetooth functionality and leave the linking between the two for Sprint 2.

4.2 PRODUCT BACKLOG

The content in this section is very similar to what was discussed in section 2.1, our previous version of the Product Backlog. For that reason, we will be discussing only the recent changes to our Product Backlog. Table 3 is an excerpt from our Product Backlog document.

Table 3: Current Changes to Product Backlog [2]

Story ID	Story Title	*Card*	Story Points(1-10)
UI-3	User learning Functionality	As a user, I want to have the letters shown so that when clicked on one will be directed to a video explaining how to do the letter clicked	6
UI-4	User Format Control	As a user, I want the action bar to contain a setting that switches between voice and text so that it will be an easy access	5
F-1	Front End	As a user, I want to have a cleaner and more attractive app	5
UI-5	Word Outputting	As a user, I want to be able to output words in addition to letters	8
AB-1	BLUNO to Android	As a user, I want to be able to connect the application with the BLUNO Arduino	9
H-2	BLUNO	As a user, I want to be able to use the BLUNO Arduino	9

Every user story currently in the Product Backlog will be covered in our second Sprint. The following sections explain the results behind the choices for our next Sprint.

4.3 SPRINT #2

Our goal for Sprint 2 is to switch our Arduino to the BLUNO so that we may be able to use the app to communicate with the hardware and fetch the letters from the glove readings.

With this change, we will be able to put the bluetooth function that we wrote during Sprint 1 into use and establish the link between the Arduino and app. In addition, we plan on to create the functionality in the app that will allow the user to create words, a tutorial section where the user can practice signing letters and, a section to calibrate the glove. The "creating a word" function will enable the user to go beyond the basic education level of Sign Language. It also will enable the user to communicate with others that don't understand Sign Language. Lastly, we plan on fixing up the Front End of the application so that it no longer follows the basic Android Project layout. Thus, making the application more appealing to the user.

Using our goals for Sprint 2 and the Product Backlog, we populated Table 4 with the task breakdown relating to the stories in the Product Backlog.

Table 4: Sprint 2 Backlog [2]

Story ID	Task ID	Task Title	Task Description	Ideal Hours
UI-3	UI-3.1	Tutorial Page	- Create a tutorial page	3
	UI-3.2	Alphabet Sign Tutorial	- Create alphabet list for tutorial	4
	UI-3.3	Calibration Page	- Create a calibration page	3
UI-4	UI-4.1	Page Routing Through the action bar	- Add action bar button to access tutorial - Link calibration page to home page - Add translating pages	4
UI-5	UI-5.1	Store Letters	- Store consecutive letters in a Queue	3
	UI-5.2	Output Word	- Fetch the letters stored in the queue - Connect the letters together in a string - Output the string	4

F-1	F-1.1	App Design	- Customize and design user-friendly layout pages	10
AB-1	AB-1.1	Connection between Android and BLUNO	- Import the BLUNO library in the Android application. - Setup necessary functions to link the app and the BLUNO	12
	AB-1.2	Remove push button	- Replace Arduino push button with app "Get Letter" button	5
	AB-1.3	Linking Button	-Create a button to link the BLUNO and the application	1
	AB-1.4	Broadcast receiver	-Verifying the states of bluetooth as messages on the Android device	5
H-2	H-2.1	BLUNO implementation	- Replace Arduino with BLUNO	5
D-2	D-2.1	Sprint 2 Documentation	- Scrum Assignment 1	10

4.4 TESTING

In this section, we will be discussing the various test cases that we will perform in order to verify the User Stories in our Sprint. The test cases are derived from the “Confirmation” section in our Product Backlog. Note, this section isn’t added in the Scrum document because it is covered in more detail in this section.

Our goal for testing is to verify the functionality and performance of each User Story. In addition, we will assure that the Story has met its requirements.

4.4.1 Test Plan 1: Tutorial Page

This test plan is to verify UI-3, the tutorial section of the application. The user should be able to go to the tutorial page and browse through the alphabet with their respective American Sign Language hand gesture. The following is the test case to verify this story.

Test Case: 1		
Pre-Condition: *Application Launched*		
Steps:	**Expected Results**	**Actual Results**
1. Click "Tutorial" in the action menu	User should be brought to the tutorial page	TBD
2. Browse through the list of letters	Should be able to see all 26 letters of the alphabet	TBD
3. Click some of the letters	Image with the letter gesture should pop up	TBD
Result:	TBD	

4.4.2 Test Plan 2: Routing Pages Through Action Bar

This test plan is to verify UI-4. UI-4 is the action menu that will allow the user to go to different sections of the application. The user should be able to press the menu, view the various sections, and be routed to the respective section when pressed. The following is the test case to verify this story.

Test Case: 2		
Pre-Condition: *Application Launched*		
Steps:	**Expected Results**	**Actual Results**
1. Click the menu button	User should be able to see all the sections (Get Letter, Get Word, Tutorial, Settings)	TBD
2. Click a page	User should be brought to that respective page	TBD
3. Repeat steps 1 and 2 until all sections have been tested		
Result:	TBD	

4.4.3 Test Plan 3: App Design

This test plan is to verify F-1, the application design. The application should be appealing and should not crash when running through user test cases. The following is the test case to verify this story.

Test Case: 3		
Pre-Condition: *Download the program on the Android*		
Steps:	**Expected Results**	**Actual Results**
1. Launch the App	The App should open	TBD
2. Go to a page and perform the function on that page (ex: get a	Successfully open a page and function returns	TBD

letter)	correctly	
3. Repeat steps 2 until all sections have been tested		
Result:	TBD	

4.4.4 Test Plan 4: Word Outputting

This test plan is to verify UI-5, the outputting of words. The user should be able to go to the "Get Word" page and create words. The following is the test case to verify this story.

Test Case: 4		
Pre-Condition: *Launch the Application*		
Steps:	**Expected Results**	**Actual Results**
1. Click "Get Word" in the menu	The user should be brought to the "Get Word" page	TBD
2. Send a letter	Letter should appear on screen	
3. Repeat step 2 for a various amount of letters	Additional letters should get added to the previous	
Result:	TBD	

4.4.5 Test Plan 5: BLUNO to Android

This test plan is to verify AB-1, BLUNO to Android connection. The user should be able to make gestures with the BLUNO connected on the glove and the Android application should receive its data. The following is the test case to verify this story.

Test Case: 5		
Pre-Condition:		
Steps:	**Expected Results**	**Actual Results**
1. Connect the BLUNO with the Android using the "Connect Device" button	Toast saying "Connected" should appear	
2. Disconnect circuit from Arduino	Arduino should be disconnected from circuit	
3. Connect circuit with BLUNO	BLUNO should be connected to the circuit and tested to assure proper connections were made in circuit.	
Result:	TBD	

4.4.6 Test Plan 6: Word Outputting

This test plan is to verify H-1, switching to the BLUNO from the Arduino. The following is the test case to verify this story.

Test Case: 6		
Pre-Condition: *Launch the Application and Connect to the BLUNO*		
Steps:	**Expected Results**	**Actual Results**
1. Send a letter using the glove	The associated letter should be outputted on the app	
2. Repeat 15x with a different letter each time		
Result:	TBD	

5.0 CONCLUSION

In conclusion, we populated our Product Backlog with User Stories relating to the requirements of our project. From there, we were able to come up with the task and goals for our next sprint. By the end of the next two weeks, we should have the BLUNO connected and the get word feature. As well, we should have worked on the tutorial and calibration sections.

6.0 APPENDIX

Android Studio: IDE to program and design Android applications.

Arduino: microcontroller with open source platform.

BLUNO: microcontroller Arduino with bluetooth module integrated within it.

Product Backlog: list of User Stories containing Stories and ideas for the project.

Sprint: a time box of two weeks during in which the scrum performs the Sprint cycle.

Sprint Backlog: list of stories broken down into task that are to be completed in the Sprint.

String: datatype that contains a sequence of characters.

TestView: A design item to show Strings on the app.

User Story:: short, simple descriptions of a feature told from the perspective of the user.

7.0 REFERENCES

[1]Bill, "Sign Language", Lifeprint.com, 2017. [Online]. Available: http://lifeprint.com/asl101/topics/wallpaper1.htm. [Accessed: 15- Oct- 2017].

[2] The Best, "Product Backlog SSS". [Online]. https://docs.google.com/spreadsheets/d/1m39TcqAVbTpy9B4Q3jKM5yM03KhjG0AcWXFu6flKy9Y/edit#gid=1701699609 [Accessed: 29- Oct- 2017]