CECS 491A

Requirement Definition

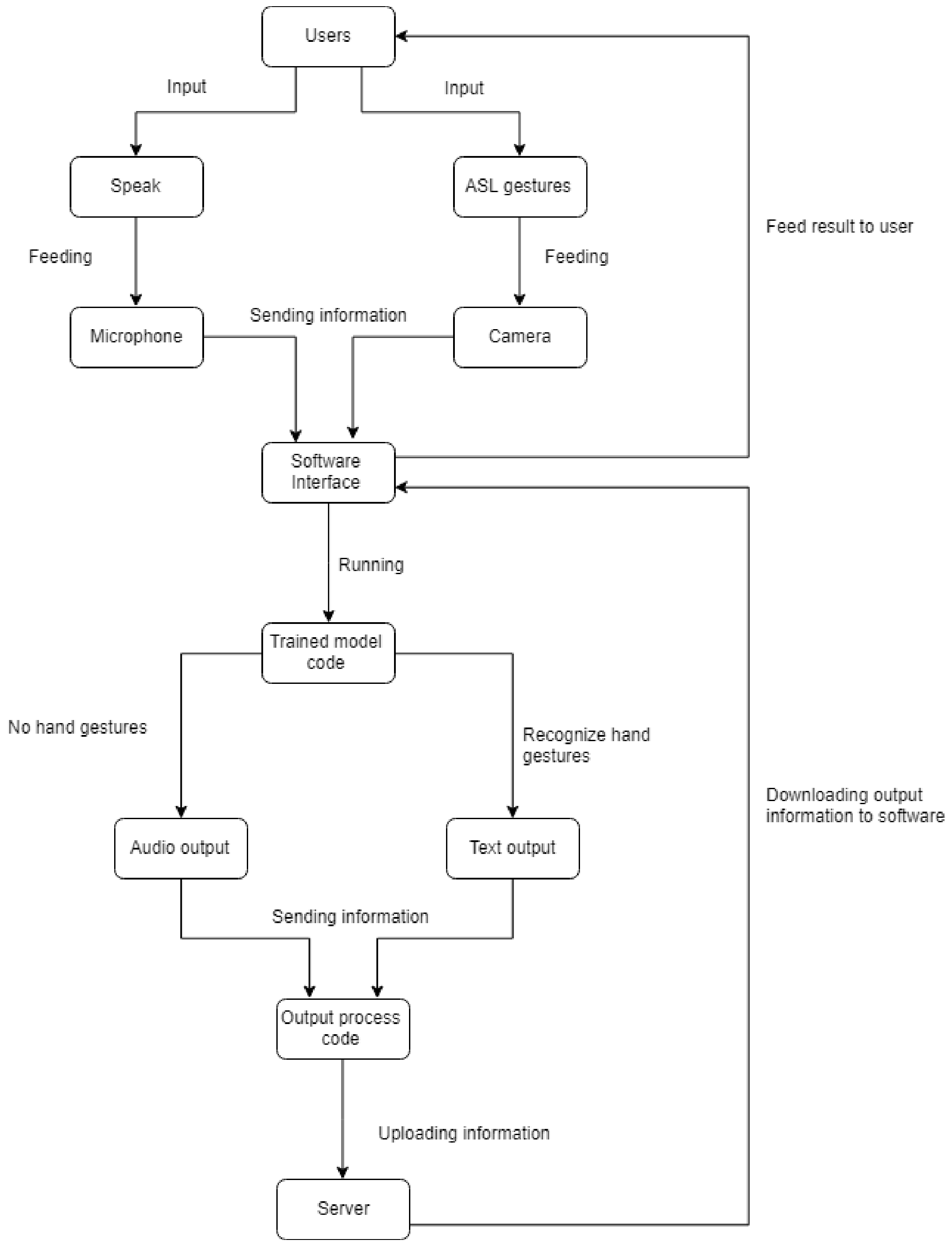
Khanh Nguyen (Leader)
Angel Franco
Christopher Imantaka
Ryan Valdriz

Table Of Content

Introduction

This program will translate **American Sign Language(ASL)** into English text. The purpose of this is to help improve communications between those who communicate through **ASL** and those who speak English. Without a translator, someone who communicates through ASL can only communicate with an English speaker through writing/typing what they want to say. Our program will get rid of the need for a translator and make conversations between the two categories of users almost seamless by eliminating the lag that comes from writing/typing texts to communicate. One of the best things about this program is that it will not cost too much to implement. It doesn’t require much space for storage nor does it require custom made hardware. With the use of machine learning, we will be able to compress the data we will need to recognize hand gestures to within 2gb. The only things required for this program will simply be a webcam, mic, and a computer. Most laptops today come with both a webcam and a mic already built in. There are also monitors that come with both built in as well. Even if you do not own a webcam or mic, they are easily obtainable through the internet for relatively cheap prices. This means that it will be easy for consumers to be able to use our product.

The System Model



System Evolution

Although this software was created with the purpose of translating ASL into English text, in the future this software can be used for translating the sign languages of other countries. Eventually this product will be able to translate sign language into speech for English speakers to hear, as well as speech into text for ASL speakers to understand. Another feature that could be later integrated into the project is to transform speech to braille, but this will require extra hardware to implement. Further improvement could be having slang included in the program. Users will be able to update these slang words as they please as long as the hand sign for it has not been used already. An upgrade in windows operating system should not affect the software, however, it is going to affect the system if the compiler is upgraded to a different version. The availability of the software could become better by extending it to mobile devices and all other operating systems.

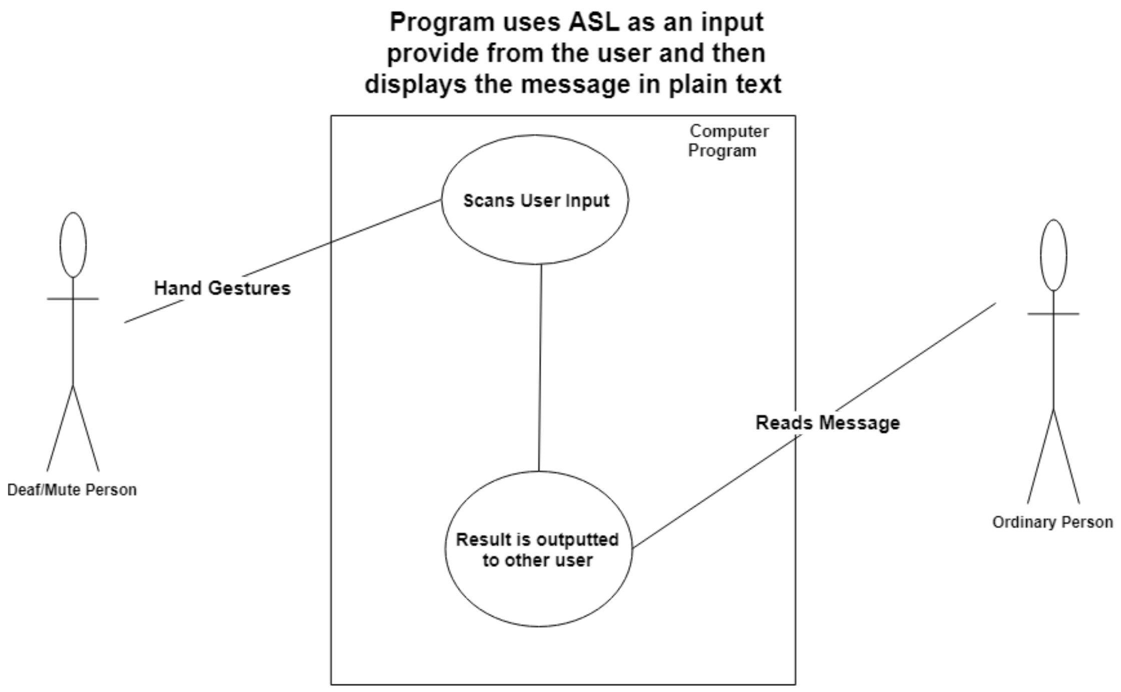
Functional Requirements

List of Requirements

1) The program should allow the user to enter text to display a message
2) The program should allow the user to edit a text message at any time
3) The program should interpret, process, and translate the user's American Sign Language hand gesture in real time
4) The program should translate ASL and display text message in real time
5) The program should translate ASL and announce message in real time
6) The program should interpret the message a user said vocally and translate it to text message in real time

Use Cases

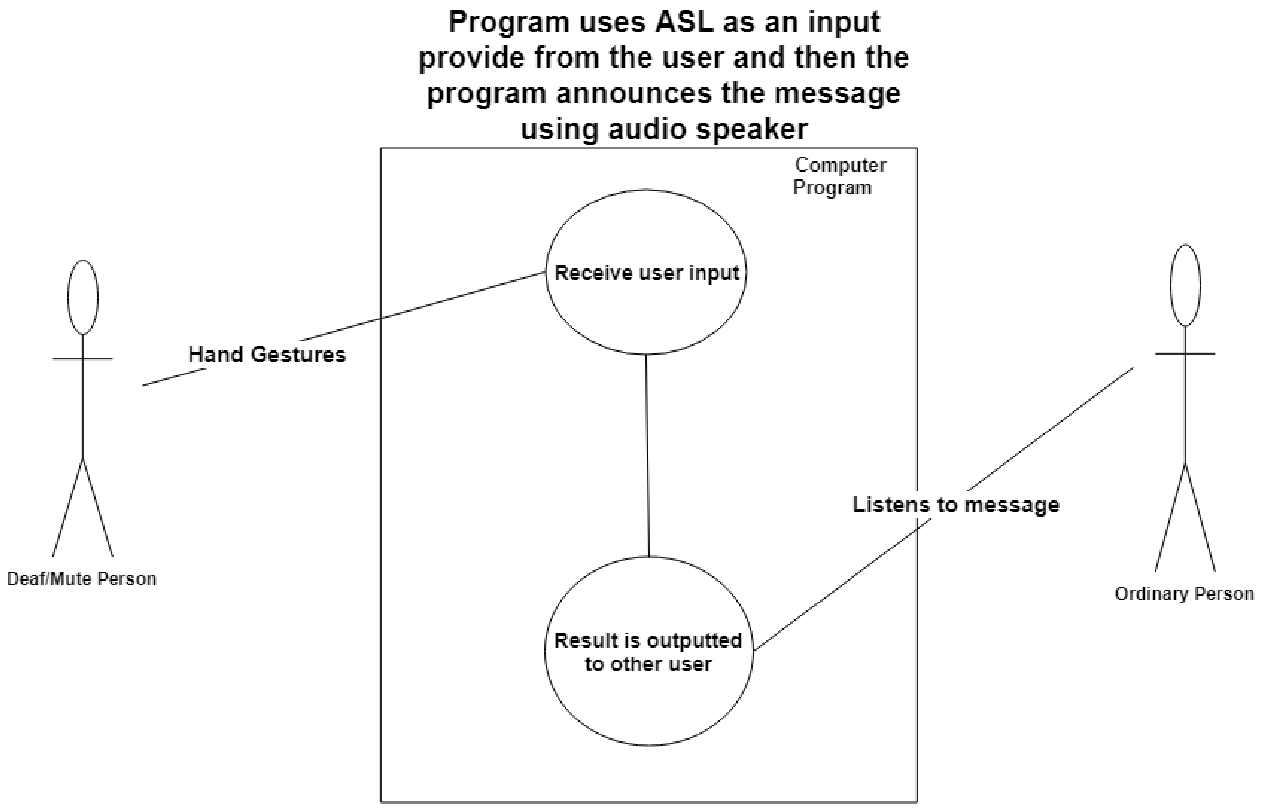
Use Case 1


Program uses ASL as an input provide from the user and then displays the message in plain text
Computer Program
Scans User Input
Hand Gestures
Deaf/Mute Person
Reads Message
Result is outputted to other user
Ordinary Person

Use Case No.	1
Scenario	Program uses ASL as input from the user and then displays the message in plain text
Characteristic Information	Goal: The program will interpret ASL and then display a message in text, in real time, so ordinary people can read and understand the interpretation of ASL. The purpose of this goal is so deaf or mute people can communicate with people who do not understand **ASL**. If a deaf or mute person has trouble communicating with an English speaker, they can use our program to translate their method of communication, which in this case is ASL, to display messages in plain text. Scope: This user case only applies to two users at a time. These two users need one and only one computer with a webcam. ASL and english are the only languages that will be involved in this project. Our program can only run on a computer with a windows OS installed.Our program does not present a user interface menu, only a display text box to display and edit. Our program is simply a translator tool that is automatically enabled once the application is running. Level: Primary Preconditions: In order for text to be displayed a user has to show American Sign Language gestures to the computer webcam so that the program can interpret it. Success End Condition: A successful end condition is when the ASL is interpreted accurately and the corresponding message is displayed simultaneously. Failed End Condition: If the program cannot interpret the user ASL input then it will display nothing. This scenario is possible if the user does not know ASL or hand gestures are too inaccurate for the program to interpret. The program is also said to fail if the message is translated but the translation is inaccurate, in

	this case the user has to manually edit the text with its keyboard. Primary Actor: The primary actors is a deaf or mute person who understand ASL and can accurately perform ASL hand gesture to give an input to the program. Trigger: In order for a message to be displayed, input must be shown in ASL by a person’s hand gestures. As soon as ASL hand gestures are shown to the webcam our software will begin to translate and display the message in real time
Main Success Scenario	Step 1: The user shows hand gestures that represent words or letters in ASL. Step 2: The camera is scanning the hand gestures. Step 3: In real time the program interprets the input and processes it to figure out the meaning of the hand gestures. Step 4: The program shows the message in plain text. Step 5: The user can edit the text manually using the keyboard or can leave the text the way.
Extension	Extensions to our program include the use of our program on mobile devices, interpreting different languages aside from ASL, displaying text in other languages aside from english, and being able to run on different operating systems.
Sub-Variations	● The user ASL input is inaccurate because they did not make a valid ASL hand gesture, in this case the program will not translate anything and a blank screen will be displayed. ● If the user shows correct ASL hand gestures but the translation is inaccurate due to program error, the user has to manually edit the message using the computer keyboard. ● Instead of outputting text the output will be in audio format.
Schedule	April 19th, 2019

Use Case 2


Program uses ASL as an input provide from the user and then the program announces the message using audio speaker
Computer Program
Receive user input
Hand Gestures
Deaf/Mute Person
Listens to message
Ordinary Person
Result is outputted to other user

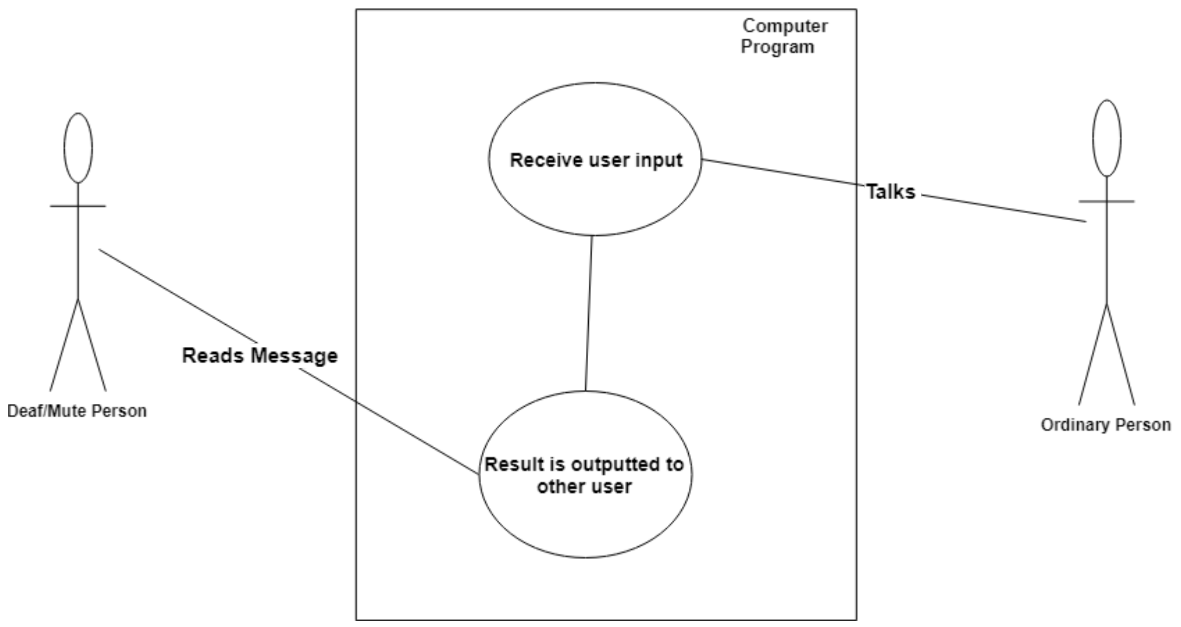
Use Case No.	2
Scenario	Program uses ASL as an input provide from the users then announces the message using audio speakers
Characteristic Information	Goal: The program will interpret ASL and then announce the translation with audio, in real time, so ordinary people can hear and understand the interpretation of ASL. The purpose of this goal is so deaf or mute people can communicate with people who do not understand **ASL**. If a deaf or mute person cannot speak or has trouble communicating they can use our program to translate their method of communication, which in this case is ASL, to announce the translation using the computer speakers. Scope:

	This user case only applies to two users at a time. These two users need one and only one computer with a webcam and speaker. ASL and english are the only languages that will be involved in this project. Our program can only run on a computer with a windows OS. Our program does not present a user interface menu, only a display text box to display and edit. Our program is simply a translator tool that is automatically enabled once the application is running. Level: Primary Preconditions: In order for our program to announce the translation a user has to show American Sign Language gestures to the computer webcam so the program can start to interpret. Success End Condition: A successful end condition is when the ASL is interpreted accurately and the corresponding message is announced simultaneously. Failed End Condition: If the program cannot interpret the user ASL input then it will announce nothing. This scenario is possible if the user does not know ASL or hand gestures are too inaccurate for the program to interpret.The program is also said to fail if the translation is inaccurate, in this case the user has to manually edit the message, which is in text, with its keyboard. Primary Actor: The primary actors is a deaf or mute person who understand ASL and can accurately perform ASL hand gesture to give an input to the program. Trigger: In order for a message to be announced an input must be shown in ASL by a person's hand gestures. As soon as ASL

	hand gestures are shown to the camera our software will begin to translate and announce in real time.
Main Success Scenario	Step 1: The user shows hand gestures that represent words or letters in ASL. Step 2: The camera is scanning the hand gestures. Step 3: In real time the program interprets the input and process it to figure out the meaning of the hand gestures. Step 4: The program displays text and announces the message. Step 5: The user can edit the message manually using the keyboard or can leave the message the way.
Extension	Extensions to our program include the use of our program on mobile devices, interpreting different languages aside from ASL, displaying text in other languages aside from english, being able to run on different operating systems.
Sub-Variations	● The user shows inaccurate ASL, in this case the program will not translate or announce anything ● If the user shows correct ASL hand gestures but the translation is wrong, the user has to manually edit the message using the computer keyboard.
Schedule	May 1st, 2019

Use Case 3

User talks and then the message is display in plain text

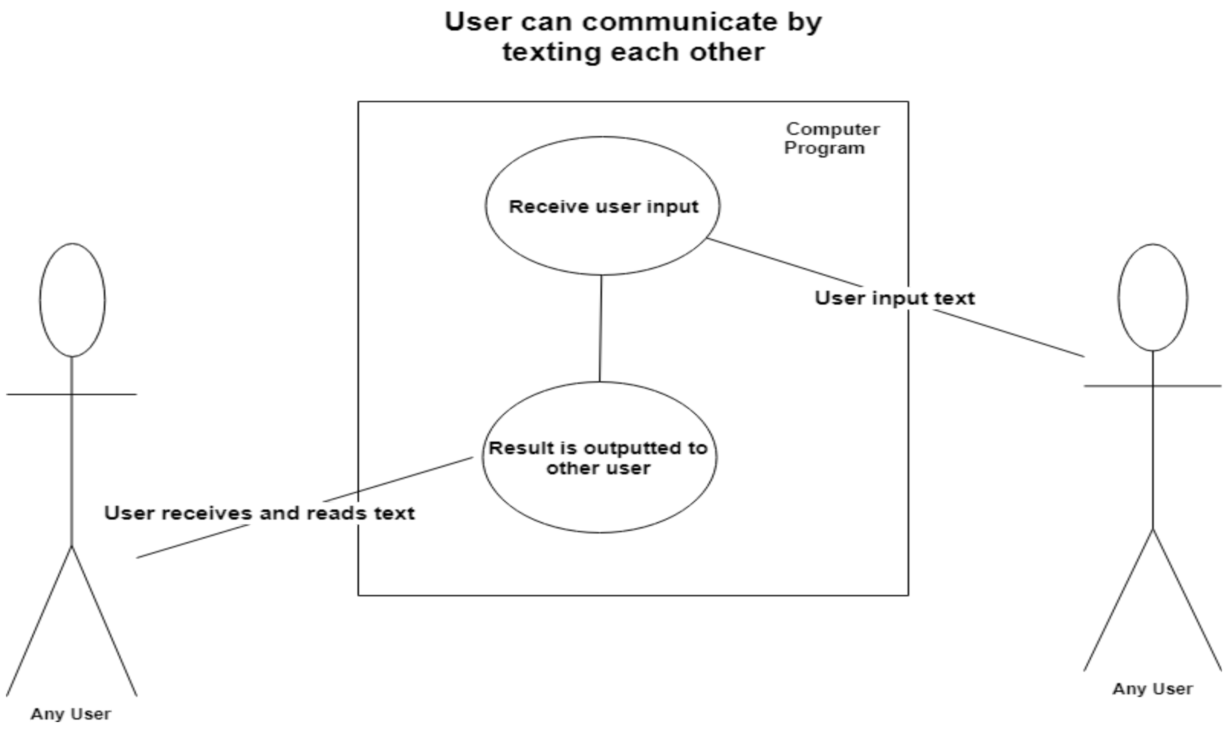


Use Case No.	3
Scenario	Users talk and then the program display the message in plain text
Characteristic Information	Goal: The program will use the computer resource such as the mic, to listen when a user speaks. The program will then display the message in plain text so that a deaf person can read the message. The purpose of this user case is to give ordinary people a way to communicate with deaf/mute people. Scope: This user case only applies to two users at a time. These two users need one and only one computer with a mic. ASL and english are the only languages that will be involved in this project. Our program can only run on a computer with a windows OS. Our program does not present a user interface menu, only a display text box to display and edit. Our

	program is simply a translator tool that is automatically enabled once the application is running. Level: Primary Preconditions: In order for our program to display a message in plain text a user must speak to the mic of a computer in english. Success End Condition: A successful end condition is when the program is able to listen to what the user is saying and simultaneously display what he or she is saying. Failed End Condition: If the program cannot understand what the user is saying then the program will not display a message. This scenario is possible if the user is not a fluent english speaker. The program is said to fail is if the program in accurately interprets what the user says. Primary Actor: The primary actors is a fluent english speaker. Trigger: In order for a message to be display in plain text, a user must speak to the computer.
Main Success Scenario	Step 1: The user talks to the computer in english. Step 2: The mic hears this and inputs it to the program. Step 3: In real time the program interprets the input and process it to figure out what was said. Step 4: The program displays text. Step 5: The user can edit the message manually using the keyboard or can leave the message the way.

Extension	Extensions to our program include the use of our program on mobile devices, understanding other languages besides english, displaying text in other languages aside from english, being able to run on different operating systems.
Sub-Variations	● If the message is wrongly interpreted by the program then the user can manually edit the message using the computer keyboard ● If the person speaks in another language aside from english or is not a fluent english speaker then the program will not recognize the input and not display any message
Schedule	March 15th, 2019

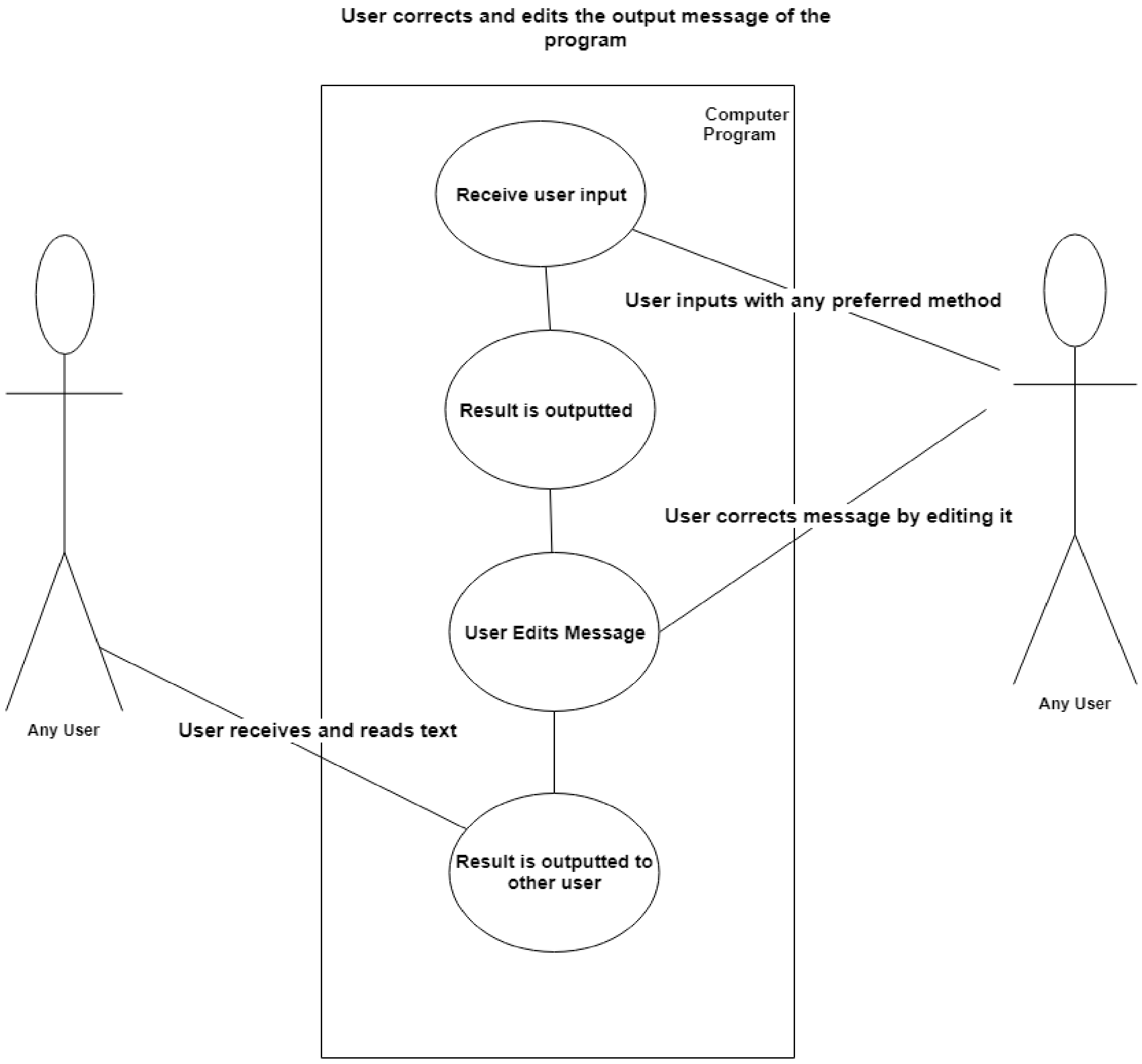
Use Case 4


User can communicate by texting each other
Computer Program
Receive user input
User input text
Result is outputted to other user
User receives and reads text
Any User
Any User

Use Case No.	4
Scenario	Users can communicate with each other by texting
Characteristic Information	Goal: Give both all users an easy and familiar way to communicating with each other. Scope: This user case only applies to two users at a time. These two users need one and only one computer. Our program can only run on a computer with a windows OS. Our program does not present a user interface menu, only a display text box to display and edit. Level: Primary Preconditions:

	The user must type anything in the messenger box for communication to happen between users. Success End Condition: A successful end condition is when the message typed and the other user sees it. Failed End Condition: If the message is not written or cannot be seen then the program failed Primary Actor: The primary actors is any single user who can type. Trigger: In order for a message to be typed the user must utilize the keyboard
Main Success Scenario	Step 1: The user types anything Step 2: The program receives the text as an input Step 3: In real time the program displays the message in text
Extension	Extensions to our program include the use of our program on mobile devices, and be able to run on different operating systems.
Sub-Variations	None
Schedule	January 15th, 2019

Use Case 5


User corrects and edits the output message of the program
Computer Program
Receive user input
Result is outputted
User Edits Message
Result is outputted to other user
User inputs with any preferred method
User corrects message by editing it
User receives and reads text
Any User
Any User

Use Case No.	5
Scenario	Users correct and edits the output message of the program
Characteristic Information	Goal: Give a way for any type of user to correct and edit their message if the program inaccurately translated it.

	Scope: This user case only applies to two users at a time. The user can not edit all messages, only the current message in the display box. These two users need one and only one computer. Our program can only run on a computer with a windows OS. Our program will not have an user menu interface. Our program does not present a user interface menu, only a display text box to display and edit. Level: Subfunction Preconditions: In order for the user to edit a message, a message had to have already be displaced. Success End Condition: A successful end condition is when the user corrects and edits their message Failed End Condition: If the program erases the current message because too much time has passed or the program is restarted then the user can never correct its message which is considered a failure. Primary Actor: The primary actors is the user who has wanted to edit their message. Trigger: The user just needs to click on the messenger box and use the keyboard to edit their message.
Main Success Scenario	Step 1: The user message is displaced. Step 2: The user clicks on the messenger box. Step 3: The user adds, deletes, or modifies the text in the messenger box.

	Step 4: The program displays the corrected message
Extension	Extensions to our program include the use of our program on mobile devices, and being able to run on different operating systems.
Sub-Variations	● The user can simply just re-enter the input by typing, speaking, or ASL hand gestures.
Schedule	Febuary 15th, 2019

Non-Functional Requirements

Constraints:

1. Webcam quality
    a. Might need a good webcam to track fast hand gestures?
    It does not need a good web cam, any webcam should work
2. Light exposure
    a. Will different levels of light exposure affect the accuracy of the program
    b. This should be use in a room where there is very little sunlight exposure
3. Contrast with background
4. Old CPUs
    a. Anything old than a 5th gen cpu will cause performance drops in the application

Glossary

- American Sign Language (ASL)
    - A form of sign language developed in the US and used also in English-speaking parts of Canada.
- Machine Learning
    - Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed.

Index

Hardware

- 720p Webcam
- Laptop/ Computer
  - Minimum requirements:
    - Intel 6th
    - NVIDIA 10 series

Data Storage

- For users:
  - 300mb of storage
- For developers
  - 40gb of storage

Budget

- Man Hours

  Each member is expected to contribute **160 to 200 hours.** The reason why we require each member to contribute so many hours is because our project requires many hours of training and gathering data. An entire data set will be created be created from scratch, we are estimating our data set will be minimum 7 GB.
- Supplies
  - Green Screen: $17.
  - 720p Webcam: $ 50
- Computing Resources
  - Google Cloud Service:
    - NVIDIA Tesla P100: $1.50/hr
      - Min training time: 70 hours → $105
      - Max training time: 200 hours → $300

Development Schedule

222 days

September 30, 2018 - May 10, 2019

Milestone Chart

Activity ID	Activity Description	Period 1	Period 2	Period 3	Period 4
1.0	Component 1: User can communicate with each other by texting	**X**			
2.0	Component 2: User corrects and edits the output message of the program	**X**			
3.0	Component 3: Program uses **ASL** as an input from the user and then displays the message in plain text		**X**	**X**	
4.0	Component 4: Program uses ASL as an input provide from the user and then announces the message using audio speakers				**X**
5.0	Component 5: User talks and then the program displays the message in plain text				**X**

Gantt Chart

Activity ID	Activity Description	Calendar Days Unit	Period 1 (50 days)	Period 2 (50 days)	Period 3 (50 days)	Period 4 (50 days)
1.0	Component 1	25	■			
1.1	Design Component 1	5	■			
1.2	Implement Component 1	10	■			
1.3	Test Component 2	10	■			
2.0	Component 2	25	■			
2.1	Design Component 2	5	■			
2.2	Implement Component 2	10	■			
2.3	Test Component 2	10	■			
3.0	Component 3	100		■	■	
3.1	Design Component 3	30		■		
3.2	Implement Component 3	50		■	■	
3.3	Test Component 3	20			■	

4.0	Component 4	25				█
4.1	Design Component 4	5				█
4.2	Implement Component 4	10				█
4.3	Test Component 4	10				█
5.0	Component 5	25				█
5.1	Design Component 5	5				█
5.2	Implement Component 5	10				█
5.3	Test Component 5	10				█