A Project Report on

COMMUNICATIONS SYSTEM FOR HEARING-IMPAIRED

Submitted in partial fulfilment for the award of the degree of

B.Tech Computer Science and Engineering
By

Henil Jayesh Thakor (18BCE0934)

Mrinmay Date (18BIS0147)

Harshith Chukka (18BCE0941)

Under the guidance of
Dr. Sweta Bhattacharya
Assistant Professor (Senior)
SCHOOL OF COMPUTER SCIENCE & ENGINEERING
November, 2020

INDEX

ABSTRACT

Since the beginning of the human kind, people with hearing and speaking disabilities faces a lot of problems in communicating with other normal people who don't know sign languages. This language gap becomes their weakness and starts dominating and suppressing their talents and capabilities. Hence a system is required to fill this communication gap to overcome their disabilities and highlight their skills and talents.
This project aims to create one such system which fills these communication gap by converting speaker's speech audio signals to Sign Language gestures understood by the hearing-impaired user.

ACKNOWLEDGEMENT

We take immense pleasure in thanking **Dr. G. Viswanathan**, our beloved Chancellor, VIT University and respected Dean, **Dr. R. Saravanan**, for having permitted us to carry out the project.

We express gratitude to our guide, **Dr. Sweta Bhattacharya**, for guidance and suggestions that helped us to complete the project on time. Words are inadequate to express our gratitude to the faculty and staff members who encouraged and supported us during the project. Finally, we would like to thank our ever-loving parents for their blessings and our friends for their timely help and support.

INTRODUCTION

1.1 Motivation

Communicating with people with hearing disabilities is not easy and can be very problematic if one doesn't know sign languages. Hence, the project aims to create a platform where an individual can communicate with a hearing-impaired person effortlessly.

1.2 Aim of the proposed Work

The purpose of the project is to develop a communication system for hearing impaired, which helping hearing impaired people to communicate with others. We can also implement this idea into various application such as video calling where in a person can just speak as usual and the app shall process his/her audio input and display the words in a sign language on the other end where the hearing impaired is situated.

1.3 Objective of the proposed work

The main objective of the project is to develop a communication system for hearing impaired, helping hearing impaired people to communicate with others by converting speech audio signal to corresponding American Sign Language gesture in real-time.

1.4 Report Organization

This report briefs the various key elements of the project *"Communication System for Hearing-Impaired".* The report firstly describes the design requirements, followed by proposed methodology and project implementation and finally derives the conclusion and the future scope of the project.

PROPOSED SYSTEM REQUIREMENTS ANALYSIS AND DESIGN

2.1 Introduction

The following section describes all the proposed system requirement analysis and design for the project which includes requirement analysis, functional requirements, non-functional requirements, SRS document, the work breakdown structure followed by pert and Gantt charts.

2.2 Requirement Analysis

2.2.1 Stakeholder Identification

i. Customers: The hearing-impaired customers using the website to communicate with other people.

ii. Owners and software developers: The owners and software developers of the website.

iii. Cloud service providers: The cloud service providers who are providing the hardware support for hosting, running and maintenance of the website.

2.2.2 Functional Requirements

i. Purpose: To convert audio message to ASL.

ii. Input: Speech audio signal in English.

iii. Processing: The input message is processed in order to convert speech into text and text to ASL.

iv. Output: Input message is displayed in ASL gestures.

2.2.3 Non-Functional Requirements

i. Safety Requirements:

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

ii. Security Requirements:

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

iii. Software Quality Attributes:

- Availability: The website should be available and should be in working condition all the time.
- Correctness: The translator should convert the audio input into correct American Sign Language (ASL).
- Reusability: The user should be able reuse the product as many times as he/she desires.
- Maintainability: The administrator should maintain correct and up to date translations.

2.2.4 System Requirements

i. H/W Requirements (details about Application-Specific Hardware):

- A networking device connected to internet which should be equipped with a microphone
- Dedicated Web server for hosting website.

ii. S/W Requirements (details about Application-Specific Software)

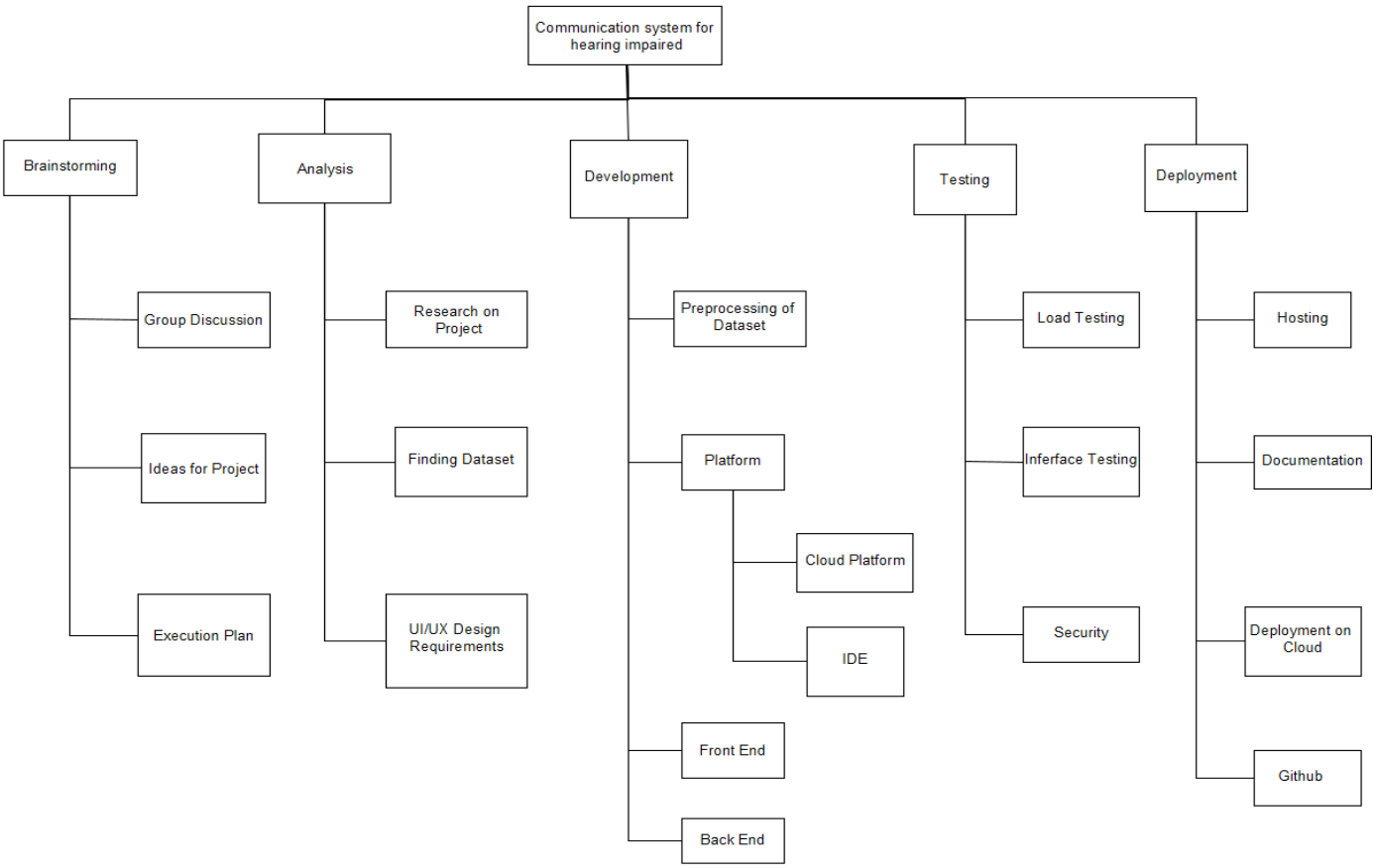
- Dedicated Web server for hosting website.
- Web Browser which supports JavaScript for running application.

2.2.5 Software Requirement Specification document

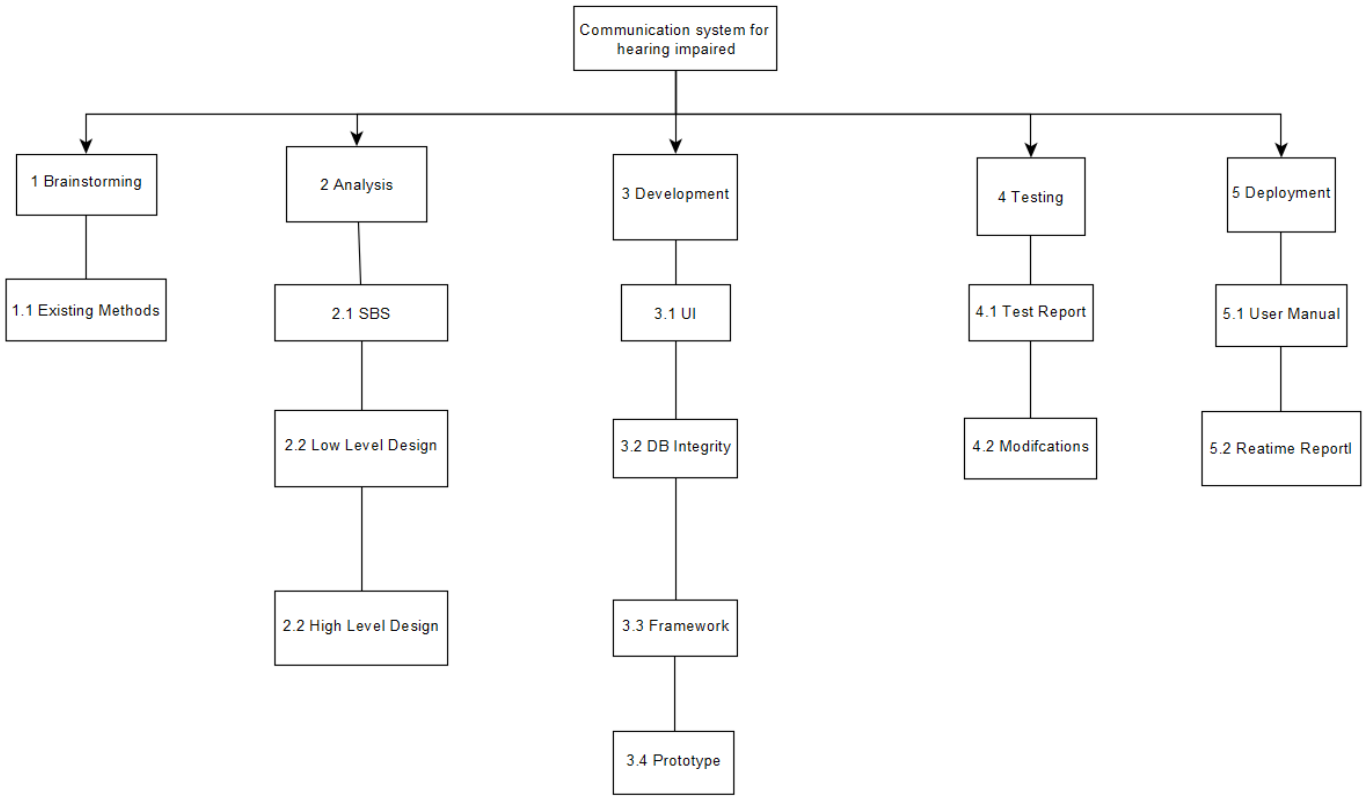
The Software Requirement Specification (SRS) document for the project can be viewed using the following link:
https://drive.google.com/file/d/1LEaccATMMiAxVOdUSYfF3baug3h3O0ZA/view?usp=sharing

2.2.6 Work breakdown Structure

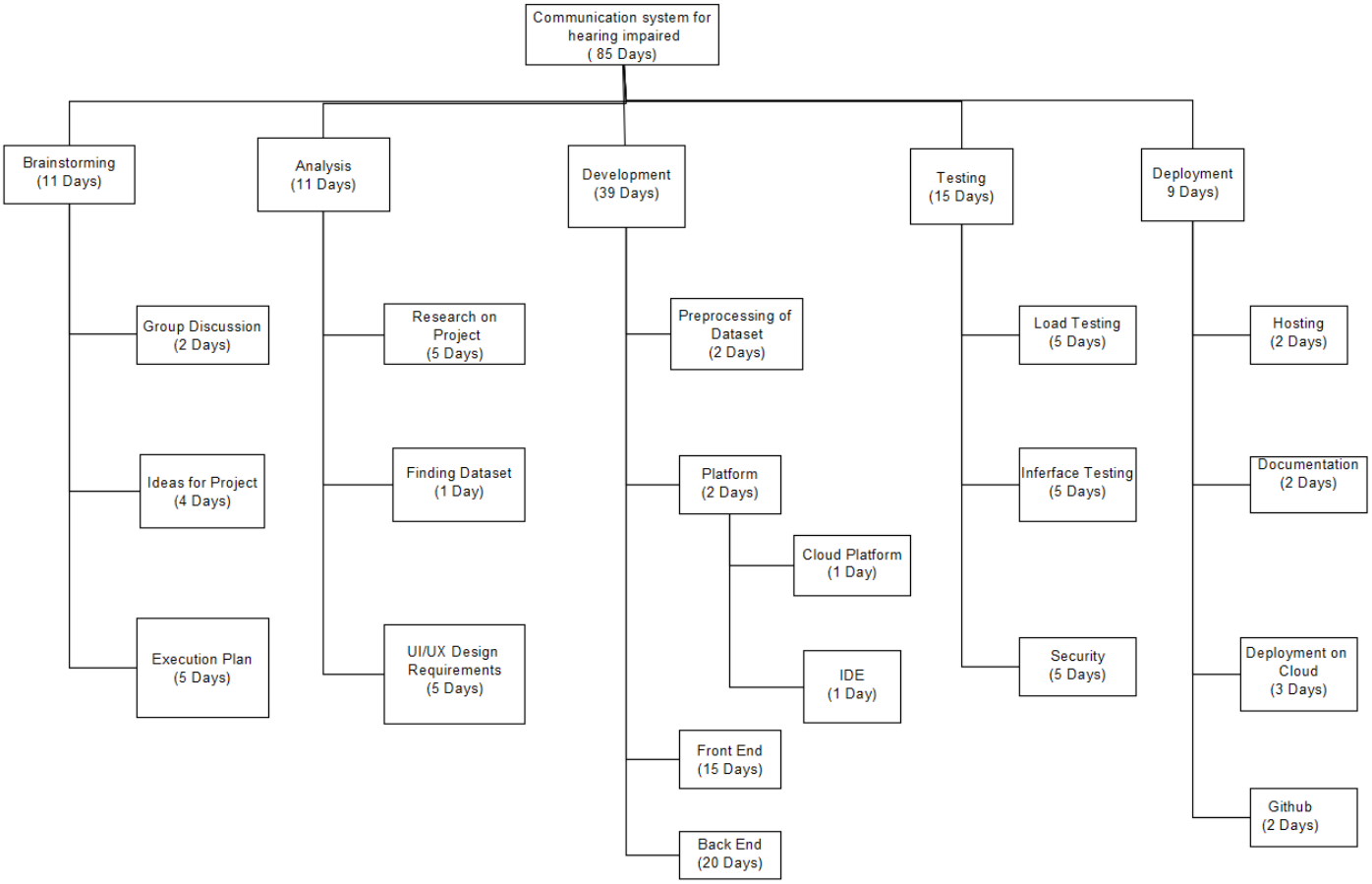
i. Verb Based WBS:



ii. Noun Based WBS:



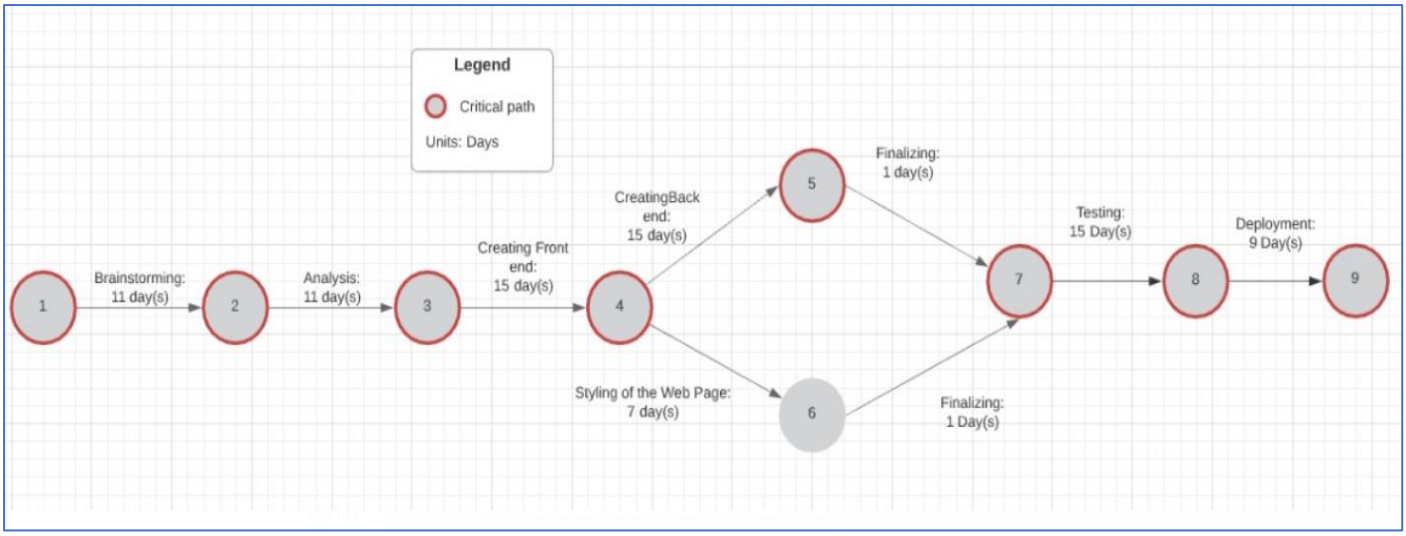
iii. Time Based WBS:



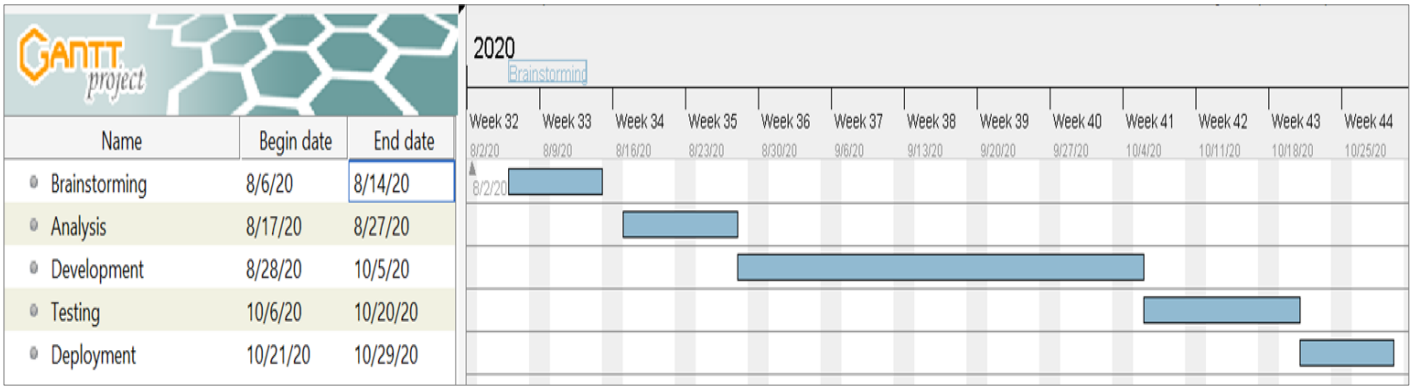
2.2.7 Pert Chart

Task Table:

Task	Prerequisite Task(s)	Day(s) to Perform
1	None	0
2	1	11
3	2	11
4	3	15
5	4	15
6	4	7
7	5,6	2
8	7	15
9	8	9



2.2.8 Gantt Chart:



DESIGN OF THE PROPOSED SYSTEM

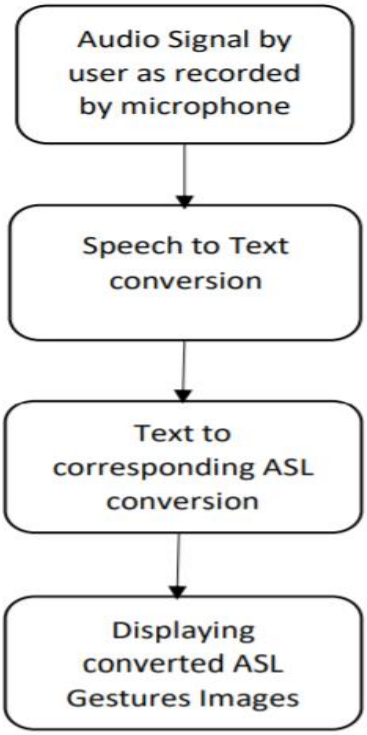
3.1 Introduction

The following section contains the

3.2 High Level Design:

3.2.1 Architecture Design:

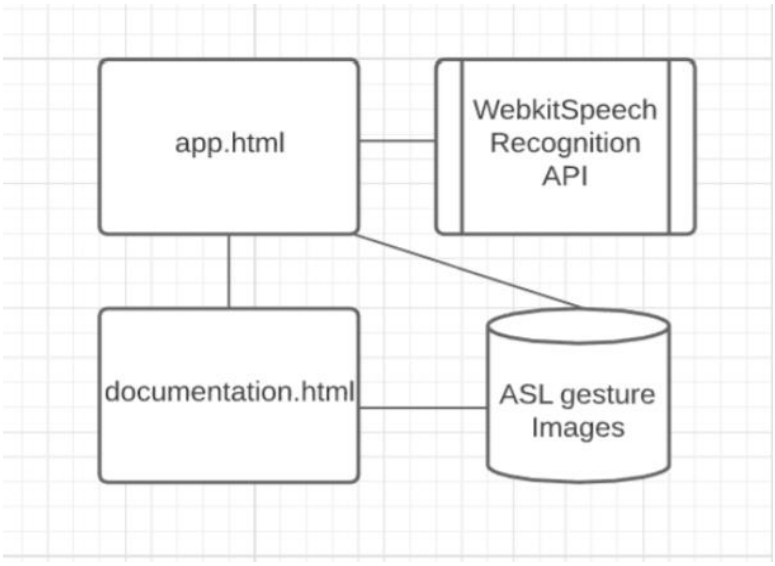
The architecture design of the project is:



It includes conversion recording of speech signal followed by speech recognition and then text to corresponding ASL gesture conversion.

3.2.2 Architecture Diagram:

The architecture diagram of the project is:

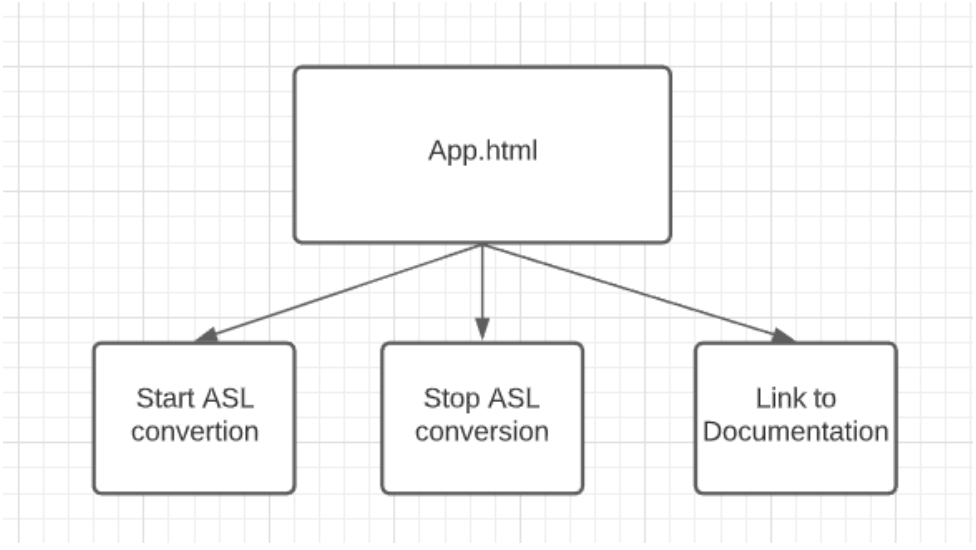


The app.html firstly records the speech audio signal of the speaker and converts it to text using WebkitSpeechRecognition API. The converted text is then converted to corresponding ASL gesture images by referring the database. The project also includes a documentation page containing the user manual and ASL guide linked to the main app.html page.

3.2.3 UI Design:

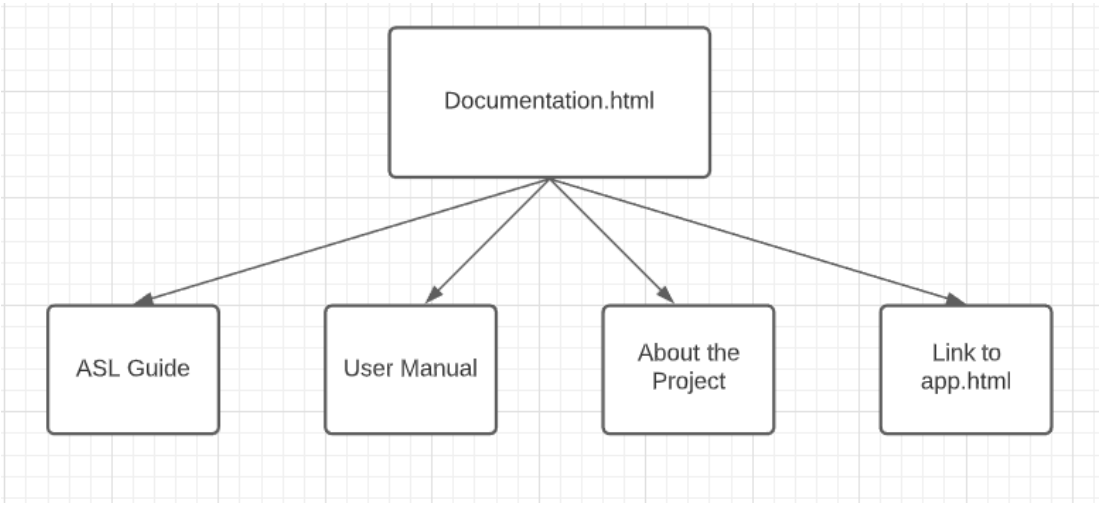
i. app.html:

It is the main conversion interface used by the hearing-impaired user which displays the gesture images of the input speech.



ii. documentation.html:

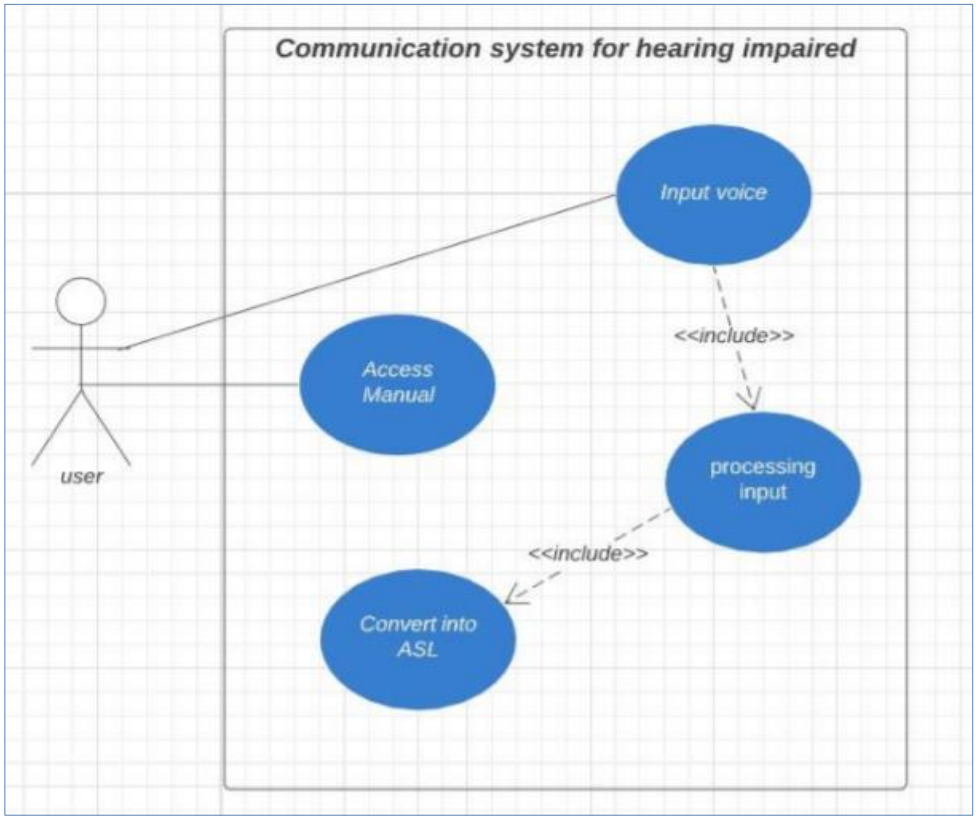
Displays the user manual for the web application and documentation of ASL gestures along with its meanings.



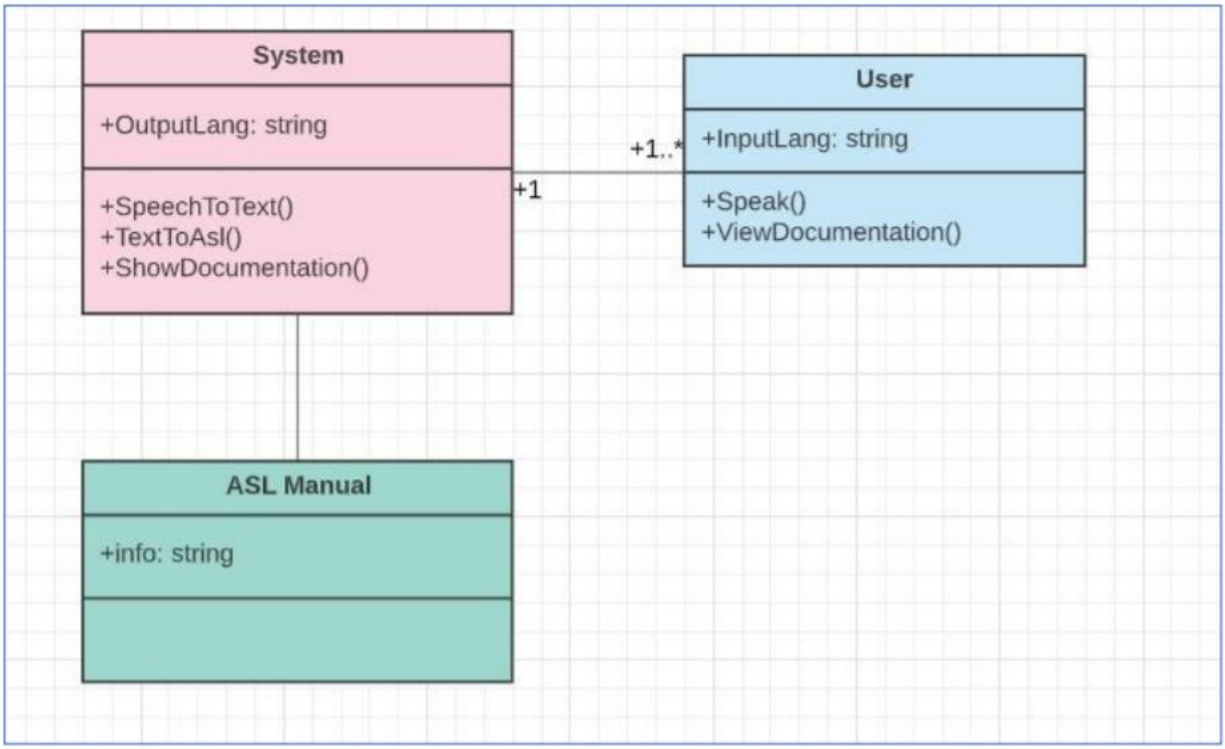
3.3 Detailed Design:

3.3.1 UML Diagrams:

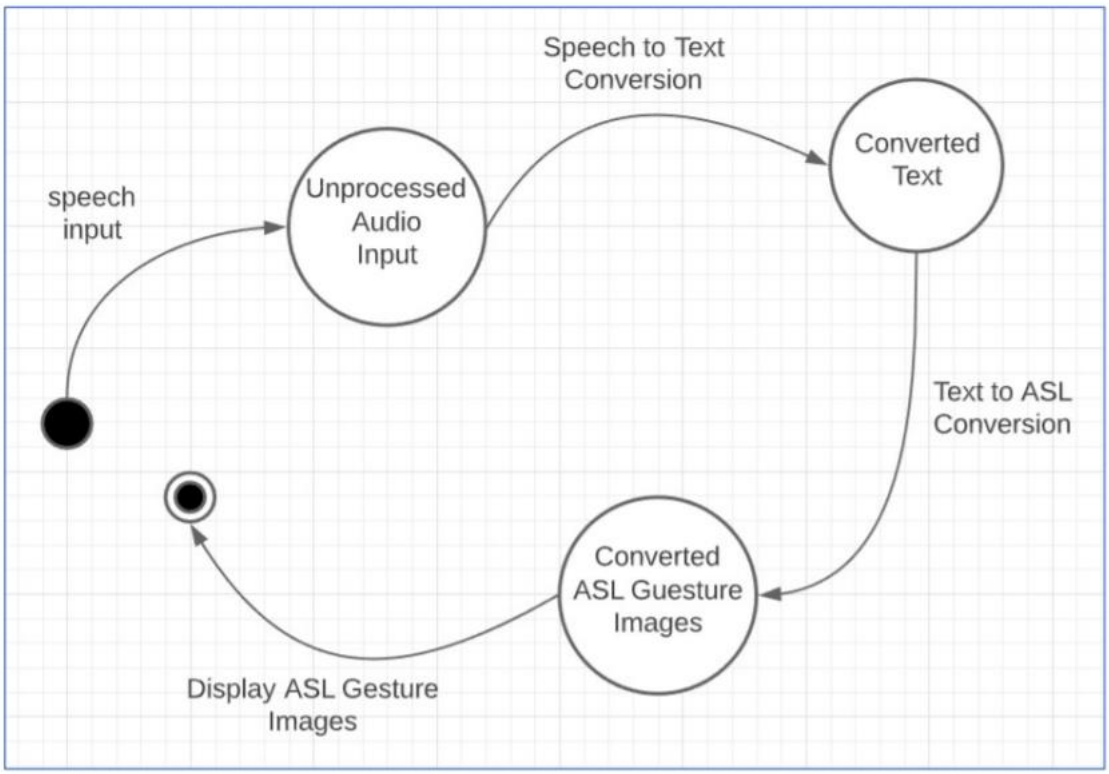
i. Use-Case Diagram:



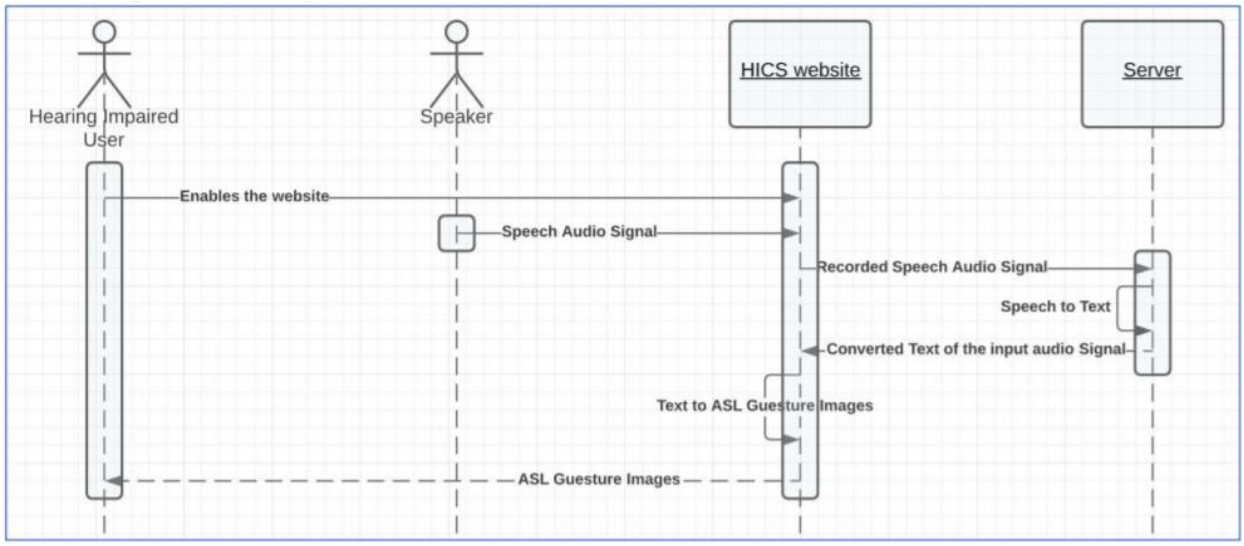
ii. Class Diagram:



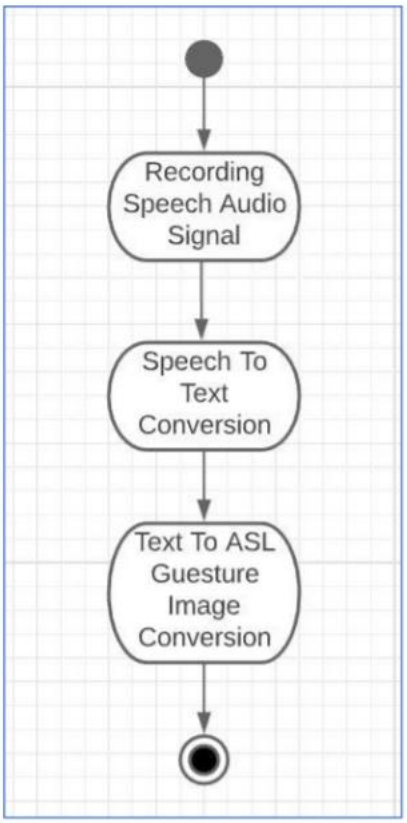
iii. State Transition Diagram:



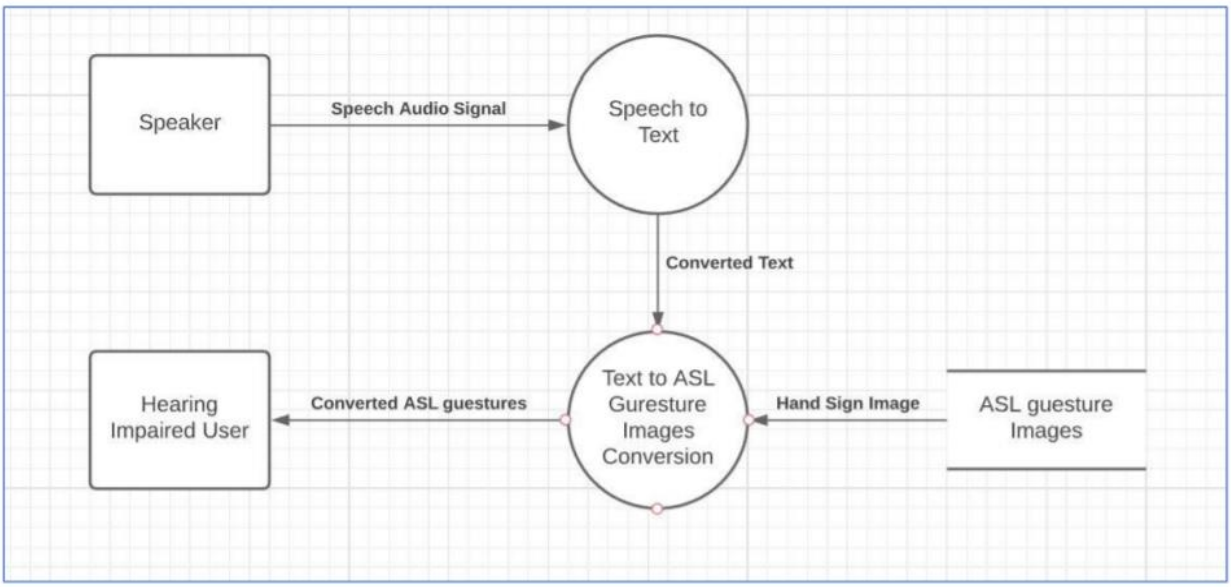
iv. Sequence Diagram:



v. Activity Diagram:



vi. Data Flow Diagram:



IMPLEMENTATION AND TESTING

4.1 Implementation Details

4.1.1 Module 1: index.html

i. Description: Takes user's speech audio signals as input and converts it into ASL gesture Images

ii. Time Complexity: $O(n^2)$

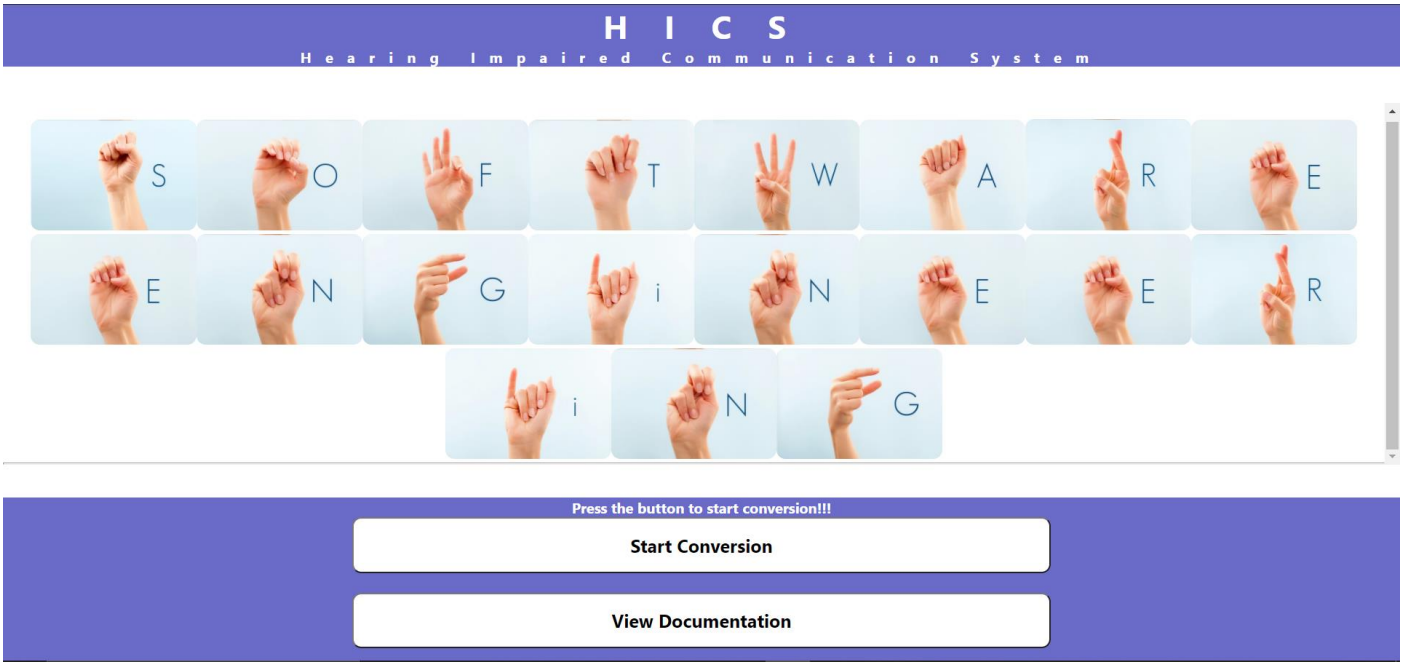
iii. Source Code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>HICS</title>
    <link rel="stylesheet" href="design.css">
  </head>
  <body>
    <center>
    <header>
```

```
      <h1>H I C S</h1>
      <p>Hearing Impaired Communication System</p>
    </header>
    <main>
      <br>
      <div id="result"></div>
      <p id="message" hidden aria-hidden="true">
        Your browser doesn't support Speech Recognition. Sorry.
      </p>
      <hr id="bottom">
    </main>
    <footer>
      <p id="buttontext">Press the button to start conversion!!!</p>
      <button id="button">Start Conversion</button>
      <br><br>
      <button id="doc" onclick="window.location.href = 'documentation.html';">
        View Documentation
      </button>
    </footer>
    </center>
    <script>
      window.addEventListener("DOMContentLoaded", () => {
        const button = document.getElementById("button");
        const result = document.getElementById("result");
        const main = document.getElementsByTagName("main")[0];
        let listening = false;
        const SpeechRecognition =
          window.SpeechRecognition || window.webkitSpeechRecognition;
        if (typeof SpeechRecognition !== "undefined") {
          const recognition = new SpeechRecognition();

          const stop = () => {
            main.classList.remove("speaking");
            recognition.stop();
            button.textContent = "Start Conversion";
            document.getElementById("buttontext").innerHTML= "Press the button to st
art conversion!!!";
          };

          const start = () => {
            main.classList.add("speaking");
            recognition.start();
            button.textContent = "Stop Conversion";
            document.getElementById("buttontext").innerHTML= "Press the button to st
op conversion!!!";
          };

          const onResult = event => {
            result.innerHTML = "";
            for (const res of event.results) {
              text = String(res[0].transcript);
              //console.log(text);
```

```
            console.log(res[0].confidence);
            text = text.toLowerCase();
            //console.log(text);
            alpha = text.split("");
            //console.log(alpha);
            var i;
            const p = document.createElement("p")
            if (res.isFinal)
            {
              p.classList.add("final");
            }
            for(i=0;i<alpha.length; i++)
            {
              //console.log(alpha[i]);
              var path = String(alpha[i]).concat(".jpg");
              var x = document.createElement("IMG");
              x.setAttribute("src", path);
              x.setAttribute("width", "12%");
              x.setAttribute("height", "8%");
              x.setAttribute("alt", String(i));
              x.setAttribute("class", "image");
              if(alpha[i]==" ")
              {
                 x = document.createElement("br");
              }
              p.appendChild(x);
            }
            result.appendChild(p);
            document.getElementById( 'bottom' ).scrollIntoView();
          }
        };
        recognition.continuous = true;
        recognition.interimResults = true;
        recognition.addEventListener("result", onResult);
        button.addEventListener("click", event => {
          listening ? stop() : start();
          listening = !listening;
        });
      } else {
        button.remove();
        const message = document.getElementById("message");
        message.removeAttribute("hidden");
        message.setAttribute("aria-hidden", "false");
      }
    });

  </script>
 </body>
</html>
```

iv. Output Screenshot:



4.1.2 Module 2: documentation.html

i. Description: Displays American Sign Language documentation.

ii. Time Complexity: O(1)

iii. Source Code:

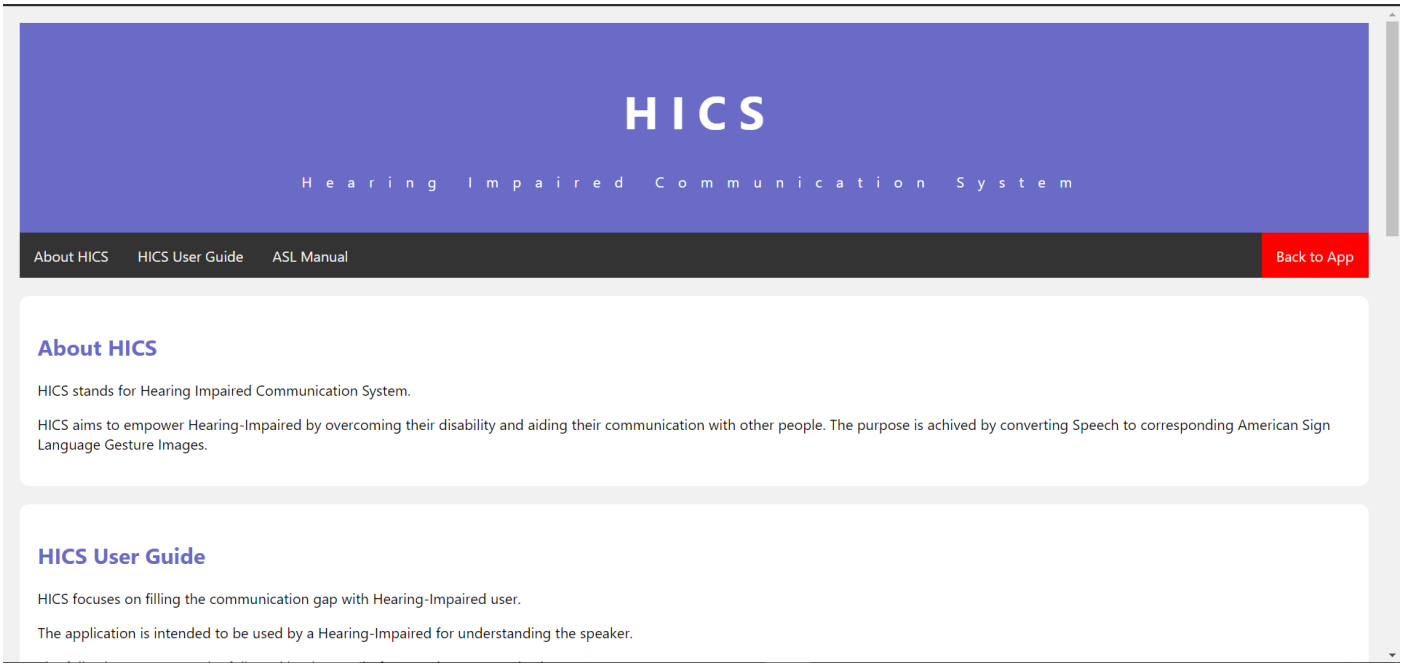
```
<!DOCTYPE html>
<html lang="en">
<head>
<style>
* {
  box-sizing: border-box;
}

body {
  font-
family: 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue'
, sans-serif;
  padding: 10px;
  background: #f1f1f1;
}
.header {
  padding: 30px;
  text-align: center;
  background: rgb(106, 106, 199);
  color: white;
}

.header h1 {
  font-size: 50px;
```

```
}

.header p{
    letter-spacing: 1vw;
}
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
.fakeimg {
  background-color: #aaa;
  width: 100%;
  padding: 20px;
}
.card {
  background-color: white;
  padding: 20px;
  margin-top: 20px;
  border-radius: 10px;
}

.card h2{
    color:rgb(106, 106, 199);
}
.row:after {
  content: "";
  display: table;
  clear: both;
}
</style>
<title>
  HICS Documentation
</title>
</head>
<body>

<div class="header">
  <h1>H I C S</h1>
  <p>Hearing Impaired Communication System</p>
</div>
```

```
<div class="topnav">
  <a href="#aboutHICS">About HICS</a>
  <a href="#HICSuserguide">HICS User Guide</a>
  <a href="#ASLguide">ASL Manual</a>
  <a href="index.html" style="float:right; background-color: red;">Back to App</a>
</div>

<div class="row">
  <div class="leftcolumn">
    <div class="card" id='aboutHICS'>
      <h2>About HICS</h2>
      <p>HICS stands for Hearing Impaired Communication System.</p>
      <p>HICS aims to empower Hearing-
Impaired by overcoming their disability and aiding their communication with other pe
ople.
          The purpose is achived by converting Speech to corresponding American Sign
 Language Gesture Images.
      </p>
    </div>
    <div class="card" id='HICSuserguide'>
      <h2>HICS User Guide</h2>
      <p>HICS focuses on filling the communication gap with Hearing-
Impaired user.</p>
      <p>The application is intended to be used by a Hearing-
Impaired for understanding the speaker.</p>
      <p>The following steps are to be followed by the user before starting communic
ation: </p>
      <ul>
        <li>Step 1: Open the HICS web application.</li>
        <li>Step 2: Before communication with other user click on "Start Conversion"
 button.</li>
        <li>Step 3: Remember to allow the application access to the microphone of yo
ur device</li>
        <li>Step 4: During communcation the panel displays the ASL gestures of the l
atest speech. The earlier gestures can be view by scrolling the panel.</li>
        <li>Step 5: Click on the "Stop Communication" button to stop translation.</l
i>
      </ul>
    </div>
  </div>
  <div class="rightcolumn">
    <div class="card" id='ASLguide'>
      <h2>American Sign Language Manual</h2>
      <p>American Sign Language (ASL) is a natural language that serves as the predo
minant
        sign language of Deaf communities in the United States and most of Anglophon
e Canada.
        Besides North America, dialects of ASL and ASL-
based creoles are used in many countries around the world,
        including much of West Africa and parts of Southeast Asia</p>
```

```
      <p>The below image depicts the Americal Sign Language gestures along with it c
orresponding meaning.</p>
    <img src="asl 2.jpg" style="width:100%;">
    </div>
</div>

</body>
</html>
```

iv. Output Screenshot:



4.2 Testing:

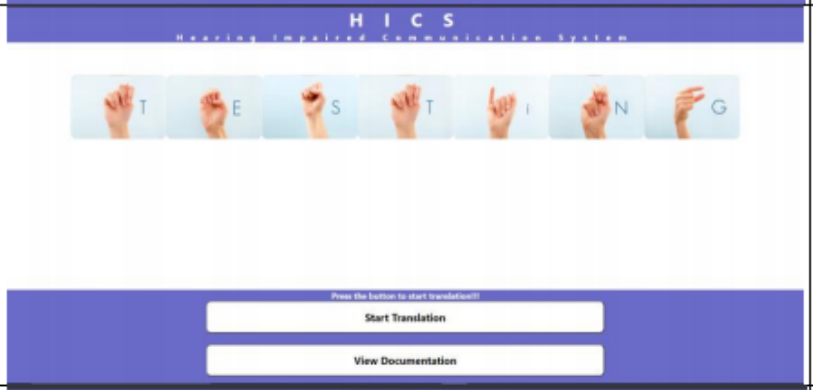
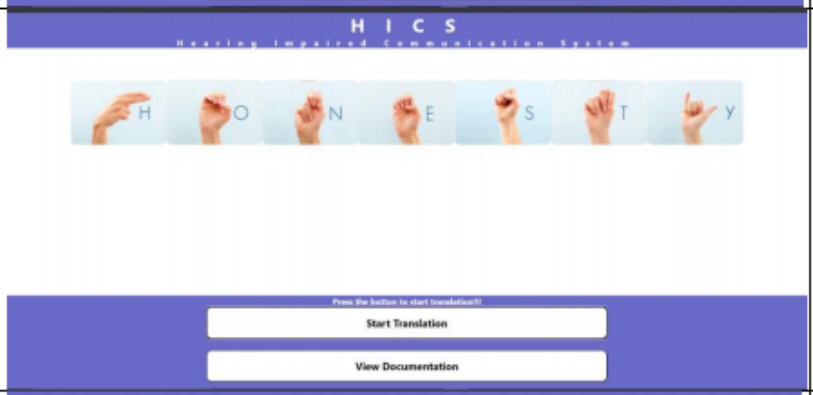
4.2.1 Type of Testing:

The software cannot be automatically tested using an automated testing tool since the speech audio signals has to be given as input and the corresponding American sign language gesture images should be displayed as output which has to verified manually by an American Sign Language expert and automated testing tool cannot verify the output ASL gesture images in the correct sequence.

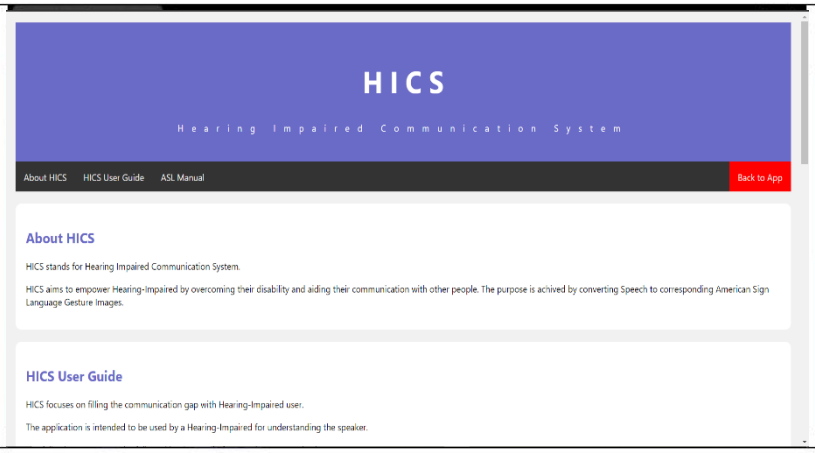
Hence, manual testing has to be performed for the software by a subject matter expert of American Sign Language.

4.2.2 **Testcases:**

i. Test Report for module1: index.html

ID	Input Speech	Expected Output	Output	Result
1	"*Speech To Text*"	"*Speech To Text*" as correct ASL gesture images		Pass
2	"*Testing*"	"*Testing*" as correct ASL gesture images		Pass
3	"*Honesty*"	"*Honesty*" as correct ASL gesture images		Pass
4.	Speaking nothing or background noise	Displaying no images.		Pass

ii. Test Report for Module2: documentation.html

ID	Input	Expected Output	Output	Result
1	none	ASL Documentation displayed		Pass

CONCLUSION, LIMITATIONS AND SCOPE FOR FUTURE WORK

5.1 Conclusion

The software was developed by following all the software engineering designed standards and using waterfall lifecycle model. The software was designed according to all the proposed UML and architecture diagrams. The project finally succeeded in achieving the desired goals by converting speech audio signals to American Sign Language Gestures with and average accuracy of approximately 91%.

5.2 Limitations

The project only deals with alphabetical American Sign Language gestures where in the software converts spoken English alphabets to corresponding gesture image.

Since the phrase level gesture languages are majorly used in the present times, people not knowing alphabetical ASL or knowing some other Sign Language might find it difficult to use to software.

5.3 Scope for future Work

The project can be further embedded in various other applications such as video calling applications, also the project could be improved by including phase level animated gestures with multiple sign language support.

REFERENCES

1. https://www.webtoolkitonline.com/
2. https://www.nidcd.nih.gov/health/american-sign-language
3. https://www.w3schools.com/
4. https://www.visual-paradigm.com/tutorials/