UDID

by Sunit Koodli

Submission date: 20-May-2021 07:43PM (UTC+0530)
Submission ID: 1590342715
File name: Capstone_Report.pdf (1.25M)
Word count: 4249
Character count: 21693

ABSTRACT

It is a cumbersome process for the visually impaired / people with muscular dystrophy to fill out forms themselves. The objective of this project is to provide a voice-based medium for automated record entry in the UDID (Unique Disability ID) form, while simultaneously performing real time analysis with high efficiency and accuracy. The goal is to eliminate the need of a helper to fill out the form for the differently abled, giving the differently abled a sense of independence. We provide a voice assistance option on the online UDID form and once the User selects that option, the software will first send out prompts. These prompts are the questions present in the UDID form. User's data is collected in the form of voice input and it undergoes several processes within the software. The software performs speech to text translation along with intent detection (our intents are the form's field names) and validation, ensuring meaningful data is stored in a tabular format. Finally, the UDID form is filled using the records present in the table.

TABLE OF CONTENT

LIST OF TABLES AND FIGURES

CHAPTER 1: INTRODUCTION

Filling out forms independently is currently unfeasible and non-viable for the differently abled people, especially the visually impaired. The reason for this is that they are unable to pinpoint the form field's locations and can't fill out lengthy forms. In most cases, they are unable to even figure out the fields present in the form. This project mainly focuses on helping the differently abled fill out the online UDID (Unique Disability ID) form. The "UDID project" is an initiative being implemented by the Department of Empowerment of Persons with Disabilities, Government of India with the objective of building a National Database for People with disabilities and ensuring a Unique Disability Identity Card is issued to each person with disabilities. With our software, we intend on improving navigability and accessibility involved in filling out the online UDID form, by providing a voice-based medium for automated entry in the fields present in the form. This automation can be achieved by Natural Language Processing techniques. We aim to eliminate the need of a helper to fill out the form for the differently abled, giving the differently abled a sense of independence.

The differently abled, especially the visually impaired find it very difficult to access technology because using them requires visual perception. A study by WHO suggests there are nearly 40 million people in India, including 1.6 million children, who are blind / visually impaired. We intend to build a software using Natural Language Processing techniques which is completely based on interactive voice response, making it efficient to use & user friendly. Doing so will lower cognitive load taken to remember and type characters using keyboard.

This product idea emerged from the prevalent gap in the services that are currently being provided to the visually impaired, online. We will use various text to speech, speech to text and intent detection modules to send prompts (form field questions) and collect responses in a voice-based format; verify and confirm User's responses, validate the intents of the questions and the response provided; and finally achieve our goal of filling out the online UDID form.

CHAPTER 2: PROBLEM DEFINITION

It is impractical and non-viable for the differently abled, especially the visually impaired to access technology, as using them requires visual perception and prior knowledge of computers and its working. A study by WHO suggests there are nearly 40 million people in India, including 1.6 million children, who are blind / visually impaired. We have come up with this project idea to simplify the online UDID form filling process, to give the differently abled a sense of independence and inspire them to be a part of growing digital India. Our proposed system that we will develop is completely different from the current system which emphasizes more on user friendliness of normal users. Our approach is based on increasing user friendliness of all sections of people by providing an easy to use voice-based medium for automated entry in the fields present in the online UDID form.

Our software first scrapes the unfilled online UDID form's HTML script. The User enables voice assistance option on the UDID form website. Our software sends out prompts (questions present on the form) to collect data from the User for each field. If the User's voice input is not clear, they are asked to repeat their response till it is clear. Once a clear response is recorded, we identify the intent of the response and match it to the question. Our system asks the User to confirm the response that was provided. If the User doesn't confirm the same, they are asked to repeat the response till our software gets their confirmation. Once a confirmed response is recorded, it is stored. Next prompt is sent out and the same process is followed. To fill out the form, the online UDID form's HTML script that was initially scraped is modified using the records present in the table (this is done after we reach the end of each page of the online UDID form).

CHAPTER 3: LITERATURE SURVEY

3.1) Paper 1: A framework to integrate speech-based interface for blind web users on the websites of public interest

	PAPER 1 (A framework to integrate speech-based interface for blind web users on the websites of public interest)
Proposed Method	• Web server plugin (SpeechEnabler) records the form elements with their order of traversal in the form of a macro. • On trigger of the macro, response page(s) are created on the fly attaching a unique ID to each element in the order of the traversal
Shortcomings for our use case	• All the page elements except for the active ones are not 'locked', thus increasing the navigability constraints for the blind user.
Possible improvements we can implement	• We can implement the framework where all the page elements except for the active nodes are 'locked' thus allowing blind users only to traverse along the active nodes in a definitive order.

3.2) Paper 2: Noisy Speech Recognition using Wavelets and FCM

	PAPER 2 (Noisy Speech Recognition using Wavelets and FCM)
Proposed Method	This paper gives light to: • Feature extraction methods like PLP, RASTA, MFCC and some hybrid algorithms like RASTA-PLP & MRASTA-PLP. • Clustering methods like Vector Quantization (VQ) and Fuzzy C Means (FCM). • Transformation techniques like DFT, DCT, HHT, wavelets.
Shortcomings for our use case	• The accuracy of hybrid algorithms is higher but these models are prone to overfit the datasets used to train it
Possible improvements we can implement	• We can implement the hybrid algorithms and perform regularization to ensure the model doesn't overfit the train dataset.

3.3) Paper 3: Implementation of Speech to Text Conversion

	PAPER 3 **(Implementation of Speech to Text Conversion)**
Proposed Method	• This paper discusses various techniques used in each step of a speech recognition process. • Provides analysis to design an efficient system for speech recognition.
Shortcomings for our use case	• The paper focuses on speech to text conversion based on Raspberry-Pi technology, a hardware tool not all users have access to.
Possible improvements we can implement	• We can understand the various important steps undergone to improve speed and accuracy of speech-to-text conversion and do a Raspberry-Pi independent implementation.

3.4) Paper 4: Query Understanding via Intent Description Generation

	PAPER 4 (Query Understanding via Intent Description Generation)
Proposed Method	This paper gives light to: • Query understanding; which constitutes Query classification, Query clustering and Query understanding. • Query based multi-document summarisation. • Query-to-Intent-Description (Q2ID) task for query understanding using a CtrsGen model
Shortcomings for our use case	• Introducing a Q2ID task using a CtrsGen model is a cumbersome and time-consuming process requiring a large number of train datasets.
Possible improvements we can implement	• From this paper we get in depth insights of how to really fine tune the query, once our voice input gets converted to txt format. • We understand how we should focus on the correct interpretation of the text.

3.5) Paper 5: Enabling Blind People to Independently Write on Printed Forms

Summary

- This paper throws light on the current scenario/methods by which the blind population is filling out forms, be it printed or online.
- Till now, it was almost impossible for the blind to fill out forms independently. They are unable to locate the position of the form fields, and in most cases, they can't comprehend what fields (e.g. name) are present in the form.
- This paper talks about a Write-it-Yourself guide (WiYG), which guides blind users to the different form fields, allowing them to fill out these fields independently without seeking anyone else's assistance.

A WiYG uses:

1. A pocket-sized custom 3D printed smartphone attachment. This comprises of two parts:
   - A base which serves as a phone stand to keep the phone upright
   - A reflector which attaches to the top of the phone, redirecting the focus of the phone's camera to the document placed in front of the phone.
2. Well-established computer vision algorithms, for e.g. OCR, using cameras embedded in smartphones. This helps in guiding the user to the different form fields via dynamically generated audio instructions.

A user study with 13 blind participants showed that with WiYG, users could fill out the form fields at the right locations with an accuracy as high as 89.5%

3.6) Paper 6: Voice based Email System Application for Blind

Summary

The paper is broken down into the proposed system and the methods of implementation:

1. **Speech to Text (STT):** Here, the audio input (either an audio file or streaming of someone speaking) will get converted to txt format and will be saved in .txt file.
2. **Text to Speech (TTS):** Here, the text format of the emails is converted to synthesized speech.
3. **Interactive voice response (IVR):** Using this, a user can interact with an email host system using a system keyboard. By listening to the IVR dialogue, users can service their own queries. The IVR system will usually respond with a pre-recorded audio voice to further inform the user how to proceed.
4. **Mel Frequency Cepstral Coefficients (MFCC):** This feature is widely used in automatic speech & speaker recognition.
5. **Screen Reader:** These software programs allow blind / visually impaired users to read text displayed on the computer screen. This is accomplished with a braille display or speech synthesizer. A screen reader is the interface between the user, the computer's OS, and its applications.

CHAPTER 4: PROJECT REQUIREMENT SPECIFICATION

4.1 Product Perspective

This product idea emerged from the prevalent gap in the services that are currently being provided to the visually impaired, online. Often, they need assistance from a third party, who may even exploit the person filling the form, by filling wrong information. This product aims to remove dependency on the third party. With this, the person can fill the online UDID form online to acquire services/benefits.

4.2 Product Features

First the voice assistant option is enabled, the options are read out, the person gives a response by speaking out the information to be filled. The audio input is converted to txt format and is filled in that field of the form. To validate the given input , the system reads out the data and asks if it is valid. If the person confirms then , the validated data is stored.

4.3 Operating Environment

It should be able to run on any OS the user is using. we will implement it on windows. Language used will be python version 3.

4.4 General Constraints, Assumptions and Dependencies

Constraints would involve:

- Different dialects and accents may prove to be difficult to recognize in speech to text translations.
- Form using google captcha may create problems for non-English speaking visually impaired people since it vocalizes some random speech in English and asks the user to type down what was said for security verification.

- Legal constraints would involve acquiring permission to integrate our application with several existing forms made by both private, institutional and government bodies.

Assumptions would involve:

- The visually impaired user is able to speak in definite terms.
- The user has the required system to be able to use our software , a laptop/desktop with stable connection.
- The user has the required hardware , i.e., braille keyboard/refreshable braille in order to find the “voice assisted filling” option.

Dependencies would involve:

- Clarity of User’s microphone to ensure clear voice inputs are recorded.
- System’s memory: Minimum of 2Gb RAM.
- System’s Processing Power: Minimum requirement of Intel i3 processor.

4.5 Risks

Some risks which might occur in the project are

- Misinterpretation of voice input which could lead to wrong data being fed.
- Misuse of confidential data given by the user.

CHAPTER 5: SYSTEM REQUIREMENT SPECIFICATION

5.1 Functional Requirements

The functional requirements for this project are:

5.1.1) Validity tests on inputs

- We re-confirm all field entries which are provided as voice inputs by the user to ensure minimal error in filling the form (e.g.: repeating name of the User letter by letter to confirm it is speltright).
- We have a mechanism which repeats questions to the User if our software has misunderstood certain field entries.
- We ensure that the input provided by the user adheres to the prerequisite validation of the form(e.g.: 10 digits for the User's phone number).

5.1.2) Sequence of operations

Below we have laid out the workflow for the sequence of operations, from a User's point of view:

- User accesses the form.
- User clicks on voice assistance option in the form.
- User gives microphone and other permissions.
- The software sends out prompts, i.e., the questions present in the UDID form, to collect and store User's data.
- Our software performs speech to text translation along with intent detection (our intents are the form's field names) and validation, ensuring meaningful data is stored in a tabular format.
- Finally, our software fills out the form using the records present in the table and allows the User to submit the form once completed.

5.1.3) Error handling and recovery

- If unclear, the software prompts the User to re-state the answer by asking the question again.
- The software re-confirms every answer by reading out the User's answers. If the user finds any mistake, he can correct it before confirming.

5.1.4) Relationship of outputs to inputs

User-software stimulus on certain triggers given by the User:

- As an input, the User selects voice option on the form as the preferred medium of filling the form.
- The respective output for the same is: A Speech based interface is presented to the User where the system prompts out the form's fields as questions.
- As an input, the user provides his voice inputs.
- The respective output for the same is: the system automates the form filling process and fills in the voice in the relevant fields of the form.

5.1.5) Hardware Requirements

The application performs intensive computations to generate the required outputs. As a result, the host systems must have high CPU speeds (~2.5GHz) and high RAM capacity (16GB and above).

5.1.6) Software Requirements

- As the application is meant to be OS independent, it is capable of running on any major operating system (Windows/Mac/Linux).
- The host system must be able to run Python version 3.6 and above.
- The application is developed using standard python libraries which the host system must be able to run. These are:
    - NumPy >= 1.11.1
    - TensorFlow >= 1.3
    - Librosa

- TQDM
- Pandas >= 0.19.2
- Sugartensor = 1.0.0.2

5.2) Non-Functional Requirements

5.2.1) Performance Requirement

- The system must have a good response time of around three to five seconds to process User input and fill out the form.
- Precision and accuracy of the system when dealing with data should not be compromised.
- System should have good CPU & RAM capacity for quicker computation.
- The system should respond to multiple users concurrently.

5.2.2) Safety Requirements

- There will be no malicious use of User's data.
- This is accomplished by not providing User info to any 3rd party.
- Data provided by User will be deleted from the temporary database once the form is submitted, i.e., it won't be stored permanently anywhere.

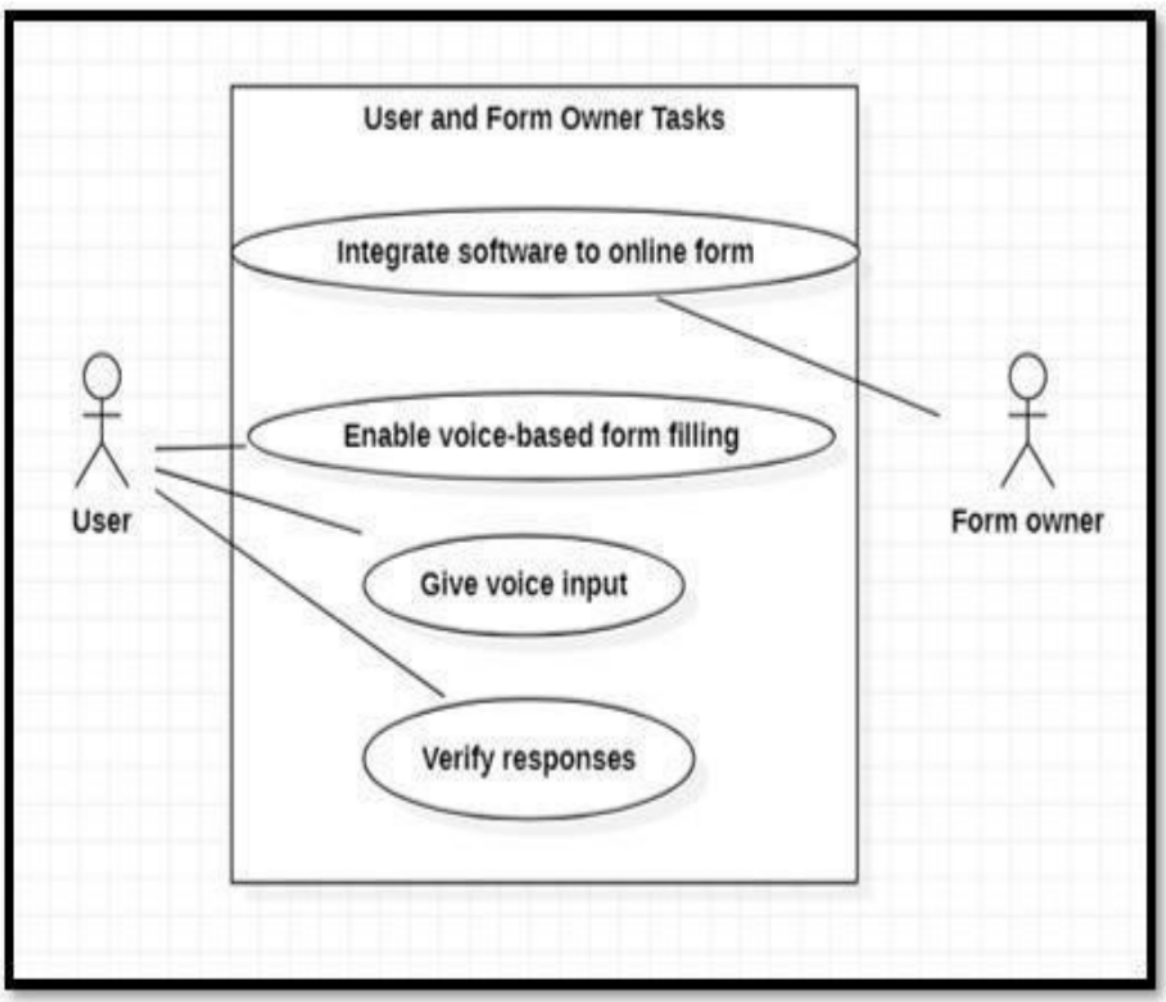
5.2.3) Security Requirements

- Necessary permissions will be taken from the respective entities and government bodies who possess legal entitlement over the form.
- Necessary permissions will be taken to carry this out, and to securely use customer data.

CHAPTER 6: DESIGN DESCRIPTION

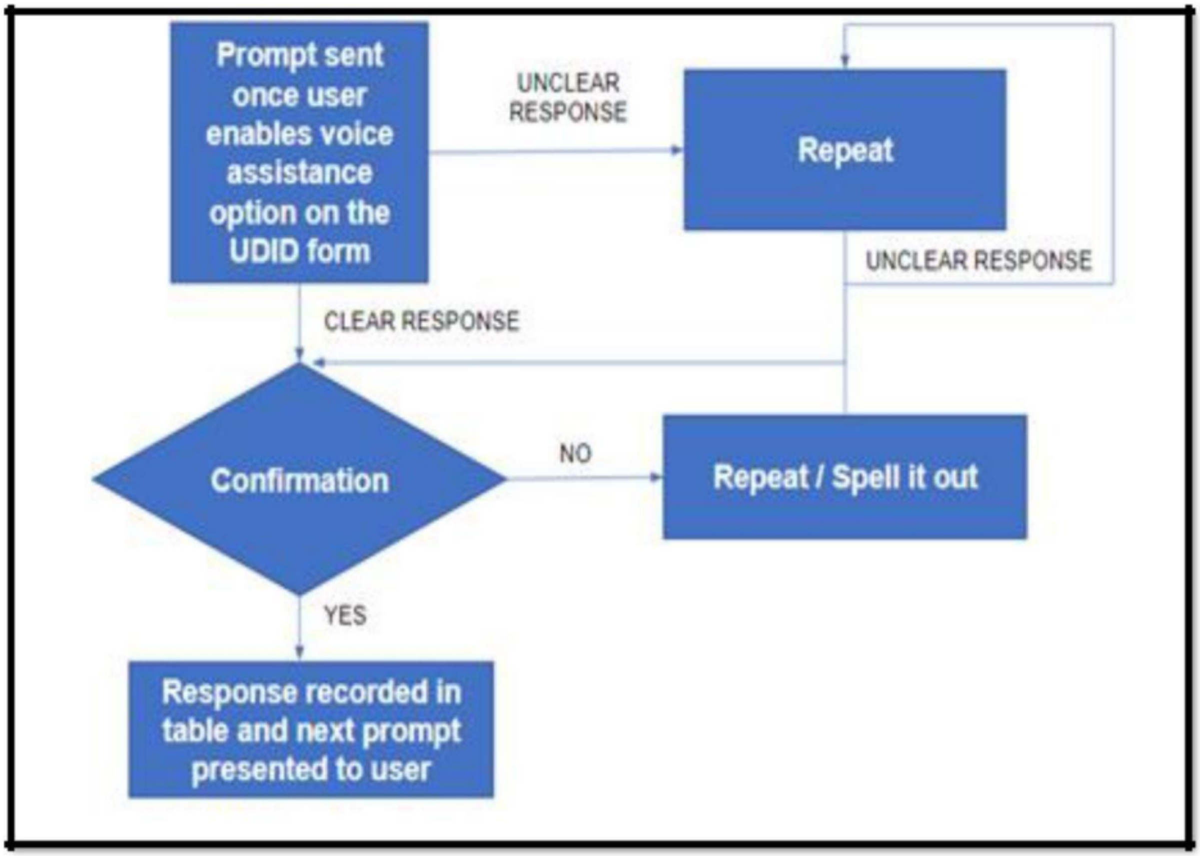
6.1) Use Case Diagram

This diagram graphically depicts different types of users and their interactions with our system.



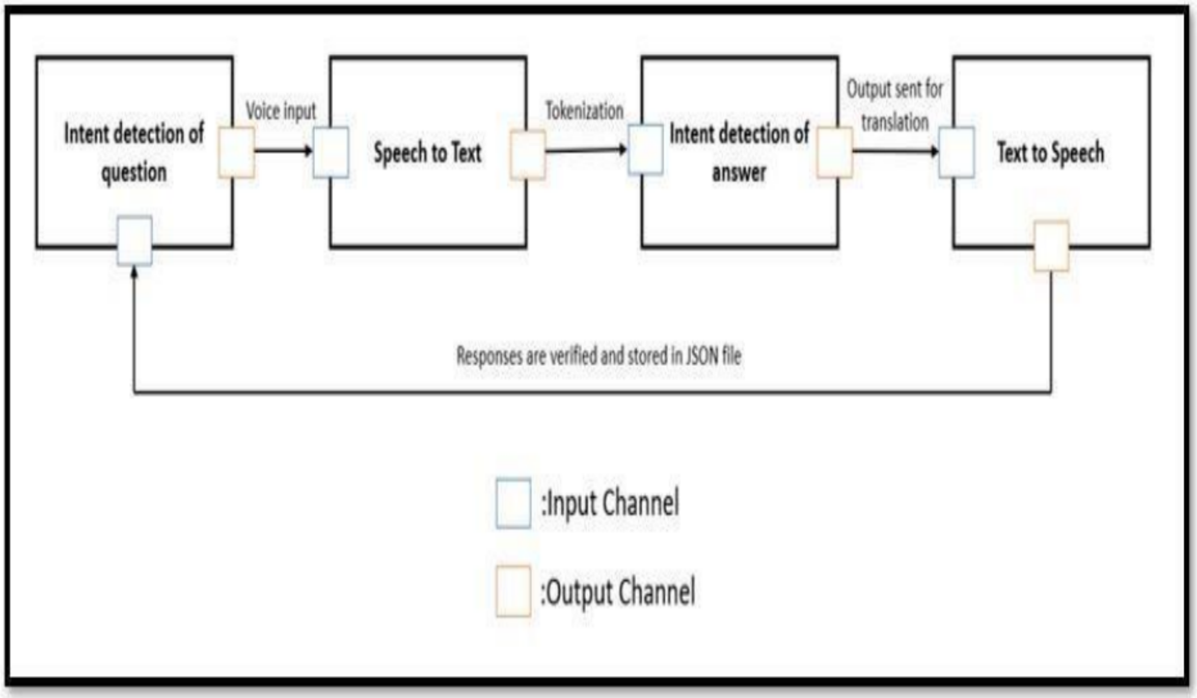
6.2) Flow Diagram

This diagram shows the step-by-step procedure for a User to answer a given prompt for it to be recorded



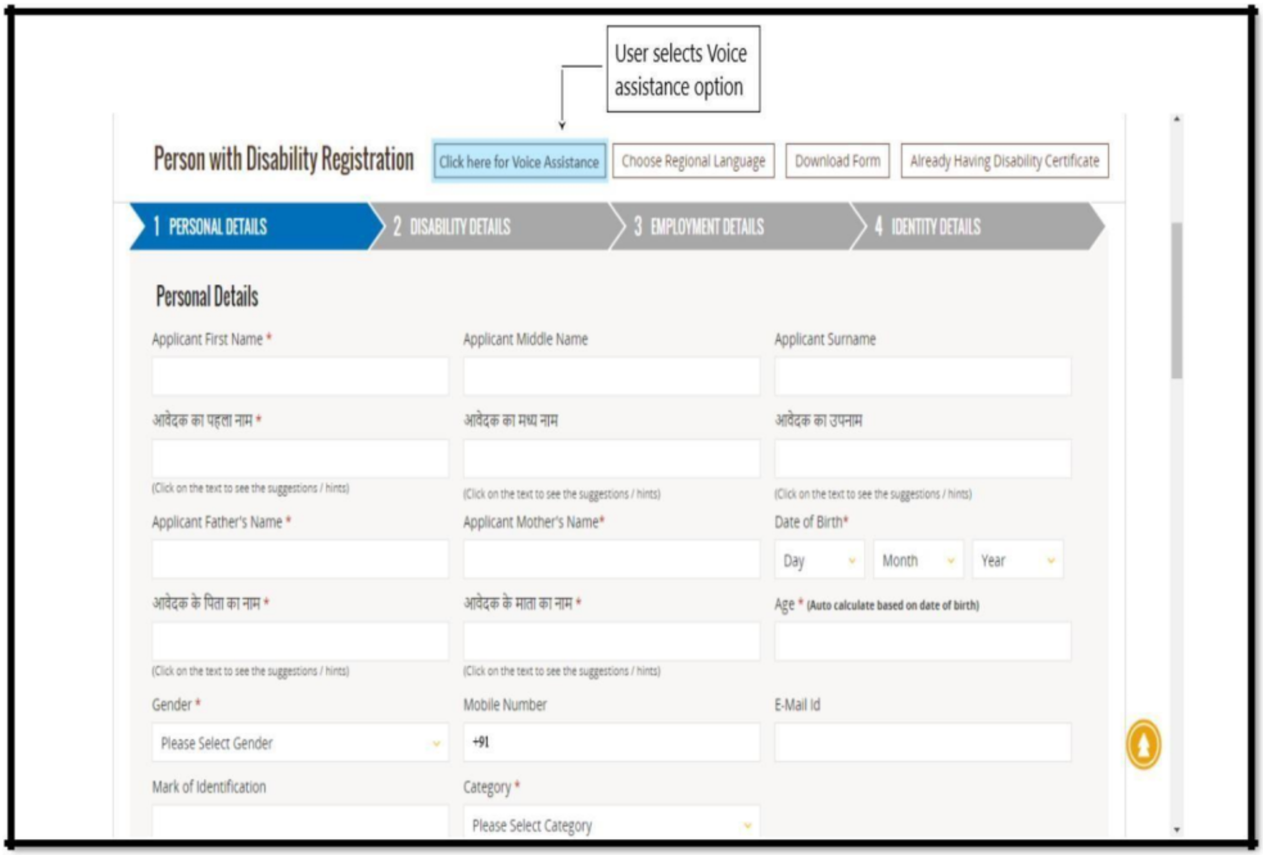
6.3) Process and Data Flow within the software

These are the internal processes that will take place in our software once the system receives a prompt from the User

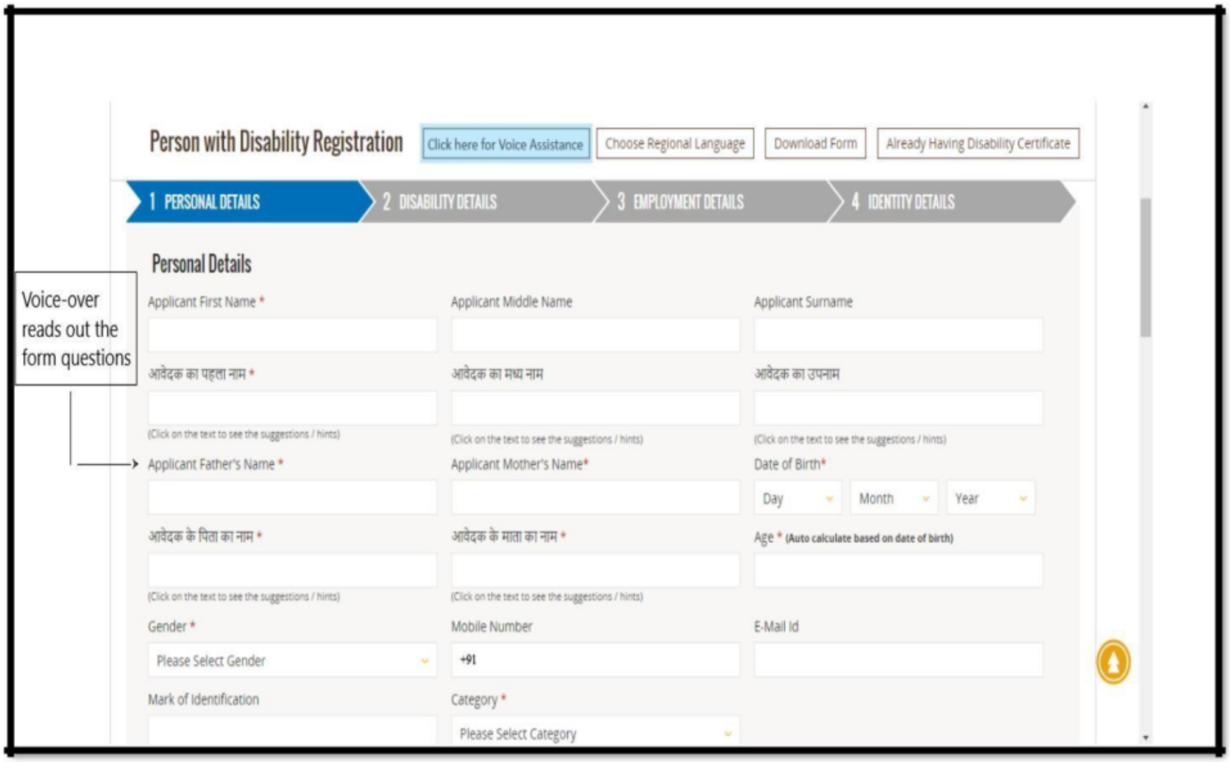


6.4) User Interface Diagrams

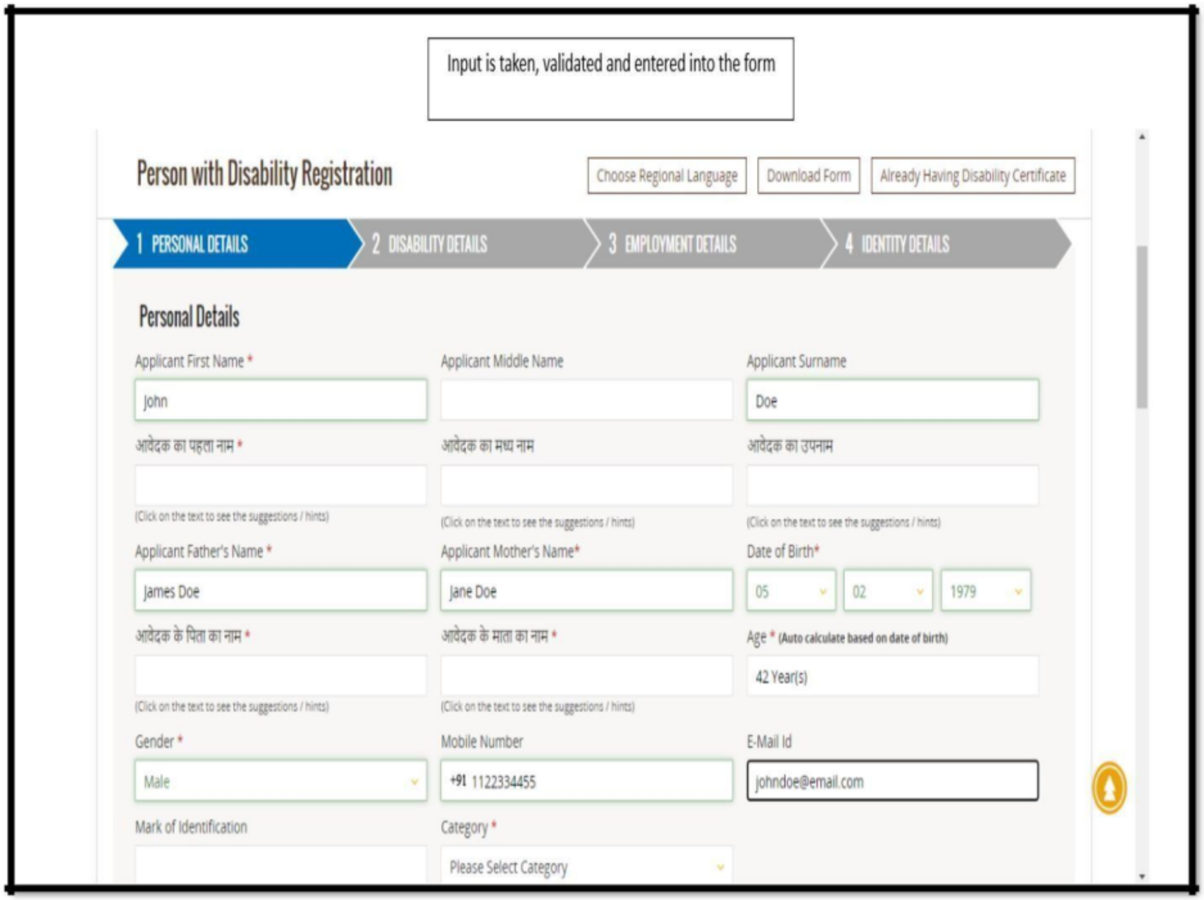
6.4.1) The User selects the voice assistance option on the online UDID form



6.4.2) Once the voice assistance option is selected, our system will read out the form questions to the User



6.4.3) This is how the form finally looks after it has been filled



CHAPTER 7: SYSTEM DESIGN

7.1) Design Goals

The goal is to enable the visually impaired to fill online forms, based on speech to text conversion with ease. The user interface will present the form, with a software application summarizing the content, purpose and what is to be filled in the form. The user clicks on the voiced based filling option, filling the form then begins sequentially, starting with the system presenting the online form in the form of questions, which will be answered by the user, the answers are then recorded and filled via speech to text conversion. Current solutions focus on how the blind can physically navigate through the form we aim to make that easier with auto navigation.

7.2 Architecture Choices

Benefits of this approach:

- Eliminates the need of a helper to fill out the form for the differently abled.
- Gives the differently abled a sense of independence.
- By providing easy voice-based form filling, we offer better navigability and accessibility.

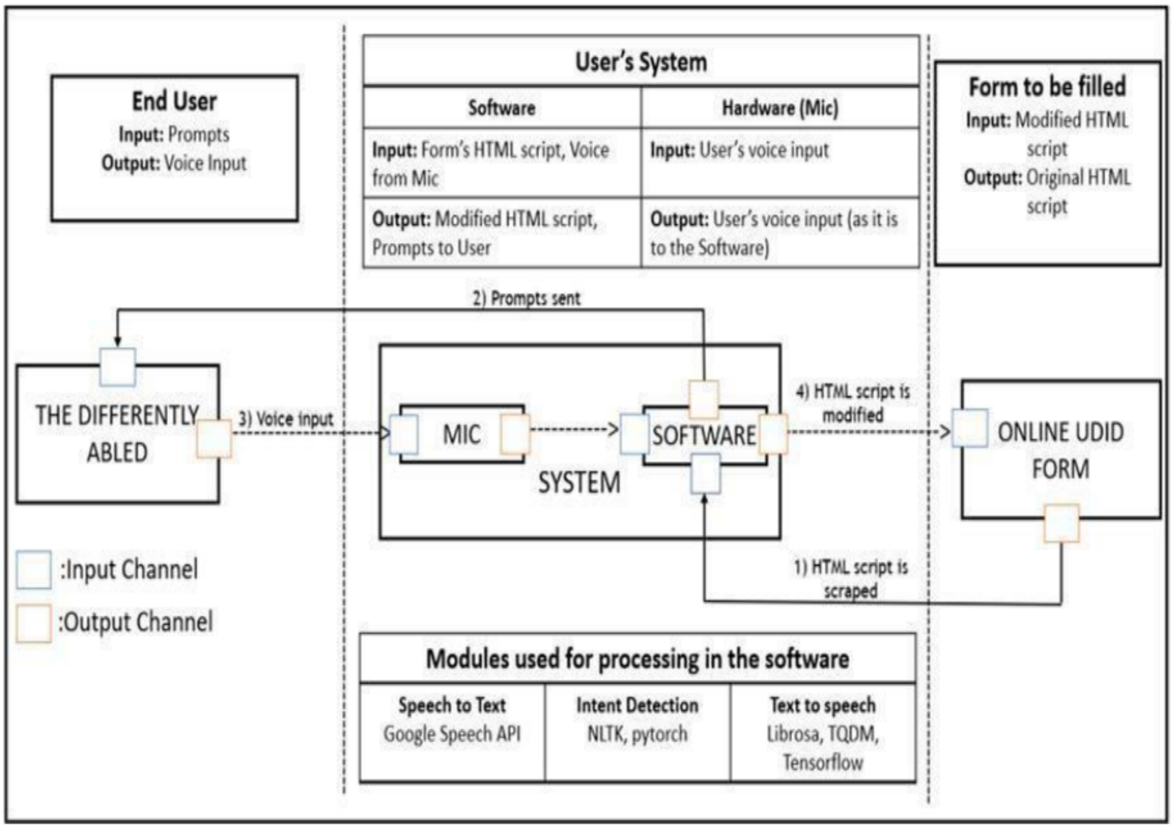
Drawbacks of this approach:

- It is a must for the user to have braille keyboard (if Visually impaired) and microphone.
- Restricted only to English as a medium of conversation.
- Misinterpretation of the input due to different accents and wide range of dialects.

7.3) Architecture Diagram

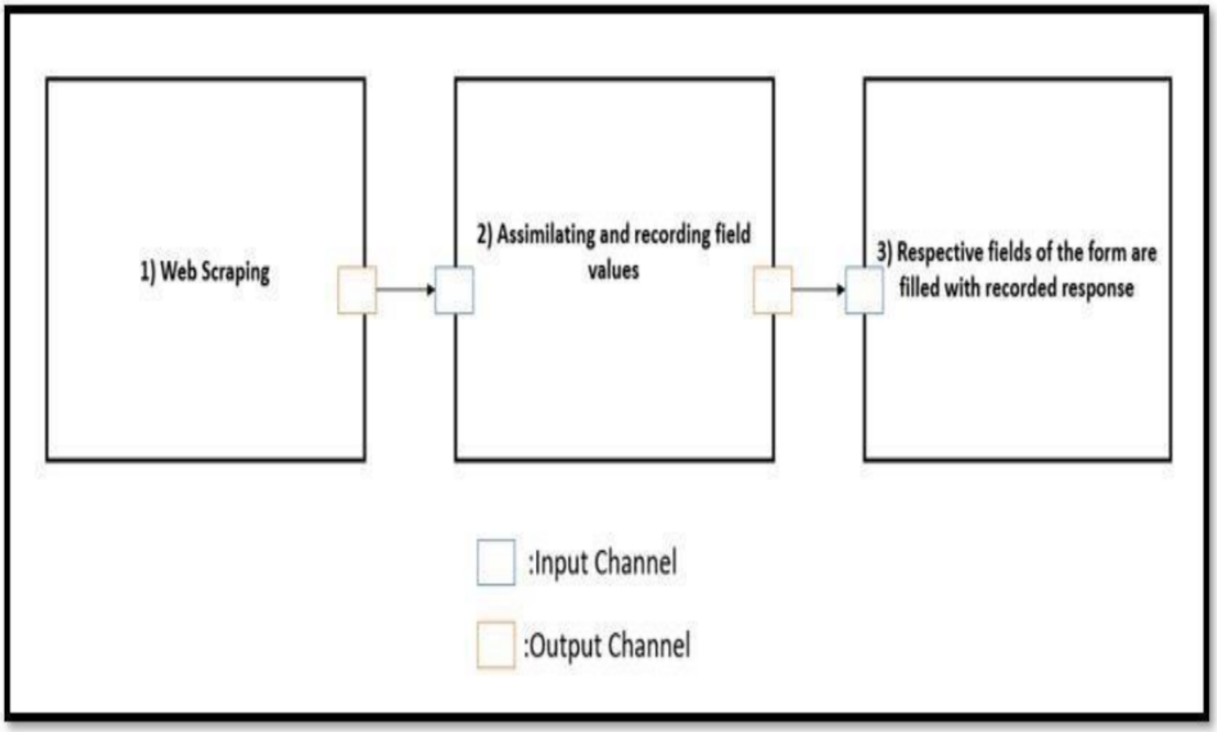
High-level design view of the system after identifying the logical user groups and various components.

The diagram illustrates the collaboration and interaction between the major components.



7.4) Software Overview

This high-level design diagram gives an overview of the major software tasks that will be implemented and their interaction.



CHAPTER 8: PSEUDOCODE

The following is the pseudocode of our implementation:

```
preparation:
   train:
      text to speech
      intent recognition

webscrape (html script) //identifies fields to be filled store questions in json

for each question:
   intent_detection_of_question ()
   voice_input
   if(no input):
      repeat prompt (2)

   speech_to_text () <google API>
   tokenization () <nltk>
   intent detection()
   if (verify):
      store corresponding answer in json
      goto next question
   else :
      repeat (1)
      spell it out

modify_script_with_answers () //fills each question field with respective answers from json file
```

CHAPTER 9: CONCLUSION OF CAPSTONE PROJECT PHASE-1

In Capstone Project Phase-1, we did an extensive research on existing papers related to how the visually impaired navigate through online forms; how they use the computer system in general; and what additional hardware and software components are required to make their navigational process easier. This gave us an understanding of the current situation regarding our problem statement and if it was valid or not. After this we did an in depth research of the functionalities needed to implement voice based form filling, such as, speech to text conversion, query intent detection, etc., to give us a clearer image of how to go about implementing this application. This research and study have helped us decide on the workflow of this project keeping in mind the challenges.

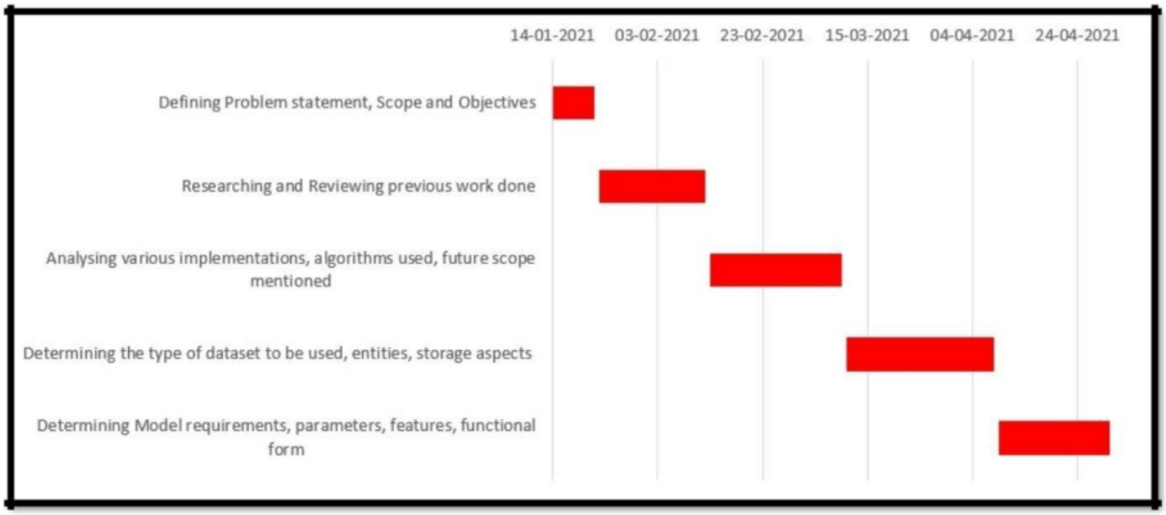
To build our software, we will be using:

- Language: Python 3
- Platform: Anaconda/Spyder
- Operating environment: Windows
- Datasets: LJ Speech Dataset, ATIS

The modules we plan on using to implement our software are:

- Speech to text: Google Speech API
- Intent detection: NLTK, pytorch
- Text to speech: Librosa, TQDM, Tensorflow

Capstone Project Phase - 1 progress:



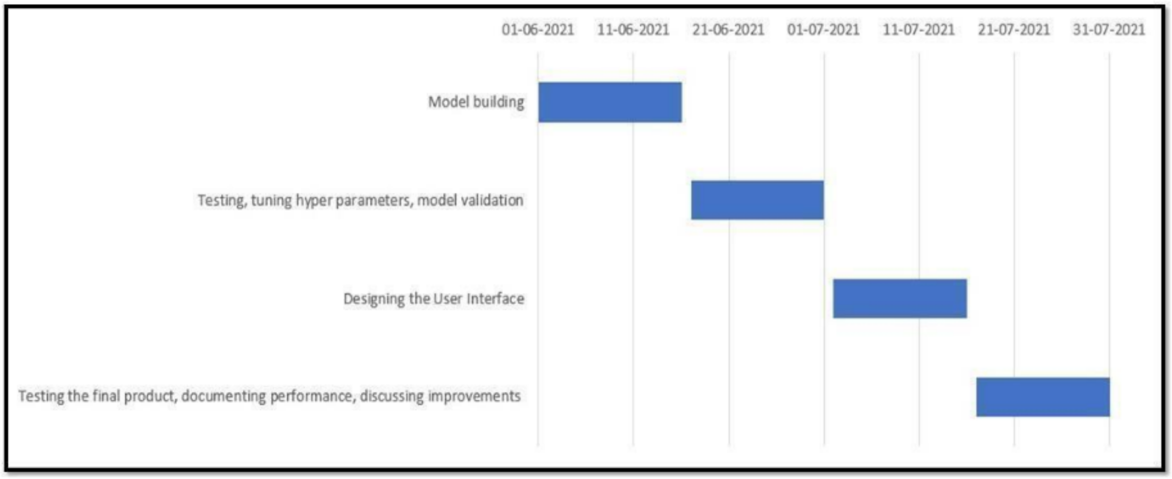
CHAPTER 10 : PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2

For capstone project phase - 2, our plan of action is to begin the process of implementing the system design we have finalized in Capstone Project Phase-1. We intend on pre-processing the datasets required for the following models:

- Speech-to-Text
- Intent Detection
- Text-to-Speech

These models will be required to build the desired software. The models specified above will be trained and tested on the pre-processed datasets. The different parameters involved will be tested and tuned until the best possible result is achieved, with respect to performance and accuracy. Once the models are validated, we will build the final software by incorporating the mentioned models in the appropriate order of workflow, as specified in our design diagrams. The developed software will be tested under all possible scenarios and the performance metrics will be documented.

Capstone Project Phase - 2 plan:



REFERENCE / BIBLIOGRAPHY

[1] Verma, Prabhat, Raghuraj Singh, and Avinash Kumar Singh. "A framework to integratespeech based interface for blind web users on the websites of public interest." Human-Centric Computing and Information Sciences 3.1 (2013): 1-18.

https://hcis-journal.springeropen.com/track/pdf/10.1186/2192-1962-3-21.pdf

[2] H. Y. Vani and M. A. Anusuya. 2016. Noisy Speech Recognition using Wavelets and FCM. In Proceedings of the 4th International Conference on Information and Network Security (ICINS '16). Association for Computing Machinery, New York, NY, USA, 83–88. https://doi.org/10.1145/3026724.3026737

[3] Khilari, Prachi, and V. P. Bhope. "Implementation of speech to text conversion." International Journal of Innovative Research in Science, Engineering and Technology 4.7 (2015): 6441-6450. http://www.ijirset.com/upload/2015/july/167_IMPLEMENTATION.pdf

[4] Zhang, Ruqing, et al. "Query Understanding via Intent Description Generation." Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020. https://arxiv.org/abs/2008.10889

[5] Shirin Feiz, Syed Masum Billah, Vikas Ashok, Roy Shilkrot, and IV Ramakrishnan. 2019. Towards Enabling Blind People to Independently Write on Printed Forms. In <i>Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems</i> (<i>CHI '19</i>). Association for Computing Machinery, New York, NY, USA, Paper 300, 1–12. DOI:https://doi.org/10.1145/3290605.3300530 https://dl.acm.org/doi/10.1145/3290605.3300530

[6] Ingle, Pranjal, Harshada Kanade, and Arti Lanke. "Voice based e-mail System for Blinds." International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) 3.1 (2016): 25-30.

https://www.arcjournals.org/pdfs/ijrscse/v3-i1/5.pdf

Appendix A: Definitions, Acronyms and Abbreviations

- API - Application programming interface
- OS - Operating software
- NLTK - Natural Language Toolkit
- TQDM - a Python library allowing you to output a smart progress bar by wrapping around any iterable.

UDID

ORIGINALITY REPORT

10%	6%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	chi2019.acm.org Internet Source	2%
2	Shirin Feiz, Syed Masum Billah, Vikas Ashok, Roy Shilkrot, IV Ramakrishnan. "Towards Enabling Blind People to Independently Write on Printed Forms", Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019 Publication	1%
3	Submitted to PES University Student Paper	1%
4	"Voice based e-mail System for Blinds", International Journal of Research Studies in Computer Science and Engineering, 2016 Publication	1%
5	Submitted to Southern Polytechnic State University Student Paper	1%
6	gcsecomputingmrf.wordpress.com Internet Source	1%

7	trap.ncirl.ie Internet Source	1 %
8	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	1 %
9	timesofindia.indiatimes.com Internet Source	1 %
10	www.tribuneindia.com Internet Source	<1 %
11	Submitted to University of Ruhuna Matara Student Paper	<1 %
12	www.irjet.net Internet Source	<1 %
13	github.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		