INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

AudioSangraha - An Approach Transforming Sinhala Audio into Summaries

A dissertation by

Mr. Aqeel Shafy

Supervised by

Mr. Ragu Sivaraman

Submitted in partial fulfilment of the requirements for the BSc (Hons) Computer Science degree at the University of Westminster.

April 2024

Declaration

I hereby declare that this thesis and the related resources for it are entirely my own work. Also, I declare that it hasn't been submitted or shared over any other platforms or institution. And the resources taken from external materials have been cited within the research.

Name - Mohamed Aqeel Shafy

Registration Number – w1832563 | 20200705

10/05/2024

Signature **Date**

ABSRTACT

Language is a unique form of communication between human beings with their environment. In that the most natural way of communicating with others is through the voice. For this nowadays there are many speech technologies available for a range of tasks. But still there is a prominent research area on speech recognition and summarization tasks on low resource language. Sinhala language is also one of the low resource languages, as there aren't enough resources available on the internet.

Nowadays, people are not intrusive of listening to continuous audio contents. Even if they listen to continuous audios, as a result they skip and try to get the information. Due to this they might get the wrong picture of information. So, as a solution the author has proposed a system for summarizing the continuous of Sinhala audio contents. Due to this people can save their valuable time while getting the correct information through the audio easily.

This system takes continuous audio files as input and generates the summary output.

For the proposed system the author has trained a model using transfer learning approaches and fine tune the pre trained Whisper AI model for the Sinhala. Also, with the test set it obtained a CER of 0.3. Then the generated continuous of audio files combined as a paragraph and sent to the summarization model which contains on summarizing through the sentence scoring on word frequency approach. And then the summarization output will be generated.

Keywords - Natural Language Processing, Speech Recognition, Extractive Summarization, Audio Summarization

Subject Descriptors:

Computing methodologies → Artificial Intelligence → Natural Language Processing → Speech Recognition

Computing methodologies → Artificial Intelligence → Natural Language Processing → Text Summarization

ACKNOWLEDGEMENT

I would like to express my gratitude to everyone who helped me to successfully complete this project within the final year of time period. First of all, I would like to thank my supervisor Mr. Ragu Sivaraman for guiding me throughout this project and giving feedback and guidance for the work I have done. And without losing hope he gave me the confidence to complete the project successfully. Also, I would like to thank Dr. Ruvan Weerasinghe, Mr. Buddi Gamage, Mrs. Farhath and Mr. Aadhil for giving necessary feedback on the work I done and for giving information to improve the system. And also, I would like to thank the module leader Mr. Guhanathan Poravi for conducting the lectures on the final year project and lectures providing materials and guidance to complete the project successfully. Then I would thank my parents and friends for their supportiveness and motivation especially on challenging times during this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATIONS

NLP	Natural Language Processing
STT	Speech to Text
LRL	Low Resource Language
BERT	Bidirectional Encoder Representations from Transformers
E2E	End to End
GPU	Graphics Processing Unit
OS	Operational System
NLTK	Natural Language Toolkit
TDNN	Time Delay Neural Network
SSADM	Structured System Analysis and Design Method
OOAD	Object Oriented Analysis and Design
UI	User Interface
LSTM	Long Short-Term Memory
ASR	Active Speech Recognition
WER	Word Error Rate
DNN	Deep Neural Network
HMM	Hidden Markov Model
GUI	Graphical User Interface

CHAPTER 01: INTRODUCTION

1.1 Chapter Overview

The Sinhala language holds an important position as a language. Mostly Sinhala language is used in Sri Lanka, and it is the prominent language. but it is also used within the Sinhala speaking communities across the globe. There is a rich cultural and historical significance in the language. But it is considered a low resource language as there is a lack of dataset available on the internet. Speech and text are the most valuable things to communicate with others. So, in this research the author has suggested a solution for the Sinhala language users on an audio summarization system. Moreover, the author will be describing the problem background, research gap, contribution to the project (both problem domain and the research domain), aim of the project, challenges, motivation and comparing the existing systems with the limitations in detail.

1.2 Problem Background

1.2.1 Natural Language Processing

Natural Language Processing is a language that interacts with the computer and human language (Millstein, 2020). And it focuses on how computers are programmed to analyze large amounts of data. Around us there are billions of text data generated. Social media, WhatsApp, Facebook, Instagram etc and other blogs, news channels, google platforms etc are beneficial from NLP. And it can be used for text understanding, speech recognition, analytics tasks, such as classifying documents and analyzing sentiment text, as well as more advanced tasks, such as answering questions, translating documents and summarizing reports (Gruetzemacher, 2022). And these benefits are that the computer works like humans do.

1.2.2 Low resource Sinhala language

Language is the unique form of communication between humans with their environment. There are more than 7000 languages used in the world. Sinhala is one of them. The Sinhala language belongs to the Indo-Aryan language category in the small island of Sri Lanka. And there are two types of grammar in the language, one is written, and the other is spoken. And there are 25 types of structures in the language (de Silva, 2019). The median age of the population in Sri

Lanka is 34.5 and there are more than 21 billion people living in Sri Lanka. They use Sinhala as their primary language in the country. And 52.6% of people use the internet (Kemp, 2022). But still the Sinhala language is considered as a low resource language, because there aren't enough resources available on the internet (Deshpande & Jahirabadkar, 2021).

1.2.3 Speech to Text

The most natural and friendly communication through human beings is the voice to interact with others (Khandare et al., 2019). Nowadays speech technologies are commonly available for a range of tasks. Speech to text is the process to convert speech to written text. These advanced technologies empower the machines to effectively respond to a human voice. The use of human voice with the machines proves that it's faster than the traditional keyboard input (Das & Prasad, 2015). And it will be an advantage for those who are frustrated using machines using the keyboard. Even these days speech recognition has a prominent research area for the low resource language (Weerasinghe et al., 2020). Unfortunately, the Sinhala speech to text has been carried out, and fewer have been successful due to a lack of economic interest (Kasthuri Arachchige & Weerasinghe, 2023).

1.2.4 Text Summarization

Text summarization has the ability to generate long text documents into shorter and accurate summaries (Singh, 2020). Single and multi-document are the two types of summarization input types. And there are two types of output it generates. Abstractive summarization and extractive summarization are the two types (Singh, 2020). Based on the input of the document the extractive summarization collects the important sentence and forms a summary. While abstractive summarization forms its own sentences to generate the summary like the human do. Generic, domain specific and query-based types are based on the purpose of the text summarization. Due to the large number of internet users in recent days, the text summarization race is high (Prudhvi et al., 2020). But it's challenging when it comes to a low resource language like Sinhala, as there is a lack of resources available (Deshpande & Jahirabadkar, 2021).

1.3 Problem Definition

Speech recognition has the ability to convert human voice audio to machine readable format. Audio is considered as the most effective mode of communication between human-being (Khandare et al., 2019). So, it is the easier way to get or transfer information between others.

While audio is converted to text there might be grammatical and spelling mistakes. As the Sinhala letters, spellings and the pronunciation of the words are complex. So, more mistakes may occur when the speech is recognized. And when its summarized text too may have been mistaken.

Nowadays with technological improvement, listening to audios like broadcasting of news on radio channels, speeches, lecture audios and audio messages are time consuming. As some of them are not interested in listening to continuous audios.

1.3.1. Problem Statement

The Sinhala language users are unable to summarize the Sinhala continuous audio files like broadcasting news on radio, speeches, interviews, audio books, lecture audios and audio messages due to a lack of resources available.

1.4 Research Motivation

Nowadays people are busy with their lifestyles. When it comes to communication and listening, people don't spend much time on listening to continuous audios because they are time consuming and not intrusive. As a result, they skip the audios here and there to grab the information soon. Due to this some important information might be skipped and get the wrong picture of information. As a solution the author tries to implement a system to summarize the audio data. Here the author has an interest in solving the problem for the Sri Lanka people. As a result, in this research the author tries to solve the problem of Sinhala language utilizing people's valuable time, listening to continuous audios by summarizing them.

1.5 Research Gap

Most of the research has contributed to the automatic summarization system and speech recognition system for the English language. Also, there are a few researchers conducted on Sinhala too. But there is a noticeable research gap when it comes to low resource languages like Sinhala. When compared to the existing work the author has elaborated to implement a system for the target audience of Sinhala language. There were gaps mentioned in previous work correcting the Sinhala language grammar and spelling of speech when it converts into audio to text (Weerasinghe et al., 2020), Improving the accuracy of the poor audio quality data (Weerasinghe et al., 2020). After considering those exiting work and methodologies there is noticeable research dedicated to summarizing the audio files. So, in this research the author will be addressing on summarizing the low resource language of Sinhala audio files with an extractive summarization approach (Warnasooriya et al., 2020). Throughout this it will be benefited to the Sinhala language users, also it will help the community narrow the research gap between the language resources.

1.6 Contribution to the Body of Knowledge

1.6.1 Contribution to the problem domain

Natural Language Process has the ability to give high impact for low resourced Sinhala language audio summarization when it comes to the contribution to the problem domain. These days NLP has been a highly contributed area between the research, when it comes to the low resourced languages (Gruetzemacher, 2022). And Sinhala audio data summarization system will be a great deal between the Sinhala language users and Sinhala-communities utilizing people's valuable time listening to continuous audios, as it will be summarized.

1.6.2 Contribution to the research domain

After considering the existing works the author concluded by implementing a system to summarize the Sinhala audio data with extractive summarization. As contributions to the research domain the author will be using transfer learning approaches and fine tune the pre-trained Whisper model for the Sinhala language on ASR (Pratama and Amrullah, 2023). And this model has the flexibility of the accent, ability on handling background noise (González, 2022), This will be a solid contribution for the research domain.

1.7 Research Challenge

In this research there are two main tasks. They are generating the speech to text and summarizing the text with use of extractive summarization. It is challenging when it comes to low resource languages that have a lack of resources available on the internet. So, there might be some challenges when it comes to finding datasets, techniques, algorithms and tools to improve the accuracy of the summarizer. And also, in the speech to text process there might be challenges of handling background noise, the speaker's accent, recognizing punctuation marks. Also, when it comes to text summarizer there might be challenges in finding a quality dataset for training the ASR system.

1.8 Research Questions

RQ1. How to overcome the audios which have a poor audio quality in the low resource Sinhala language?

RQ2. What are the techniques that have been used for the speech recognition tasks?

RQ3. How may the extractive summarization optimize for the low resourced language?

RQ4. What are the models and techniques being used in existing works on summarization?

1.9 Research Aim

The aim of the research is to design, develop and evaluate a summarization system for the low resource of Sinhala language audio data using natural language processing.

As further elaboration on the aim the author will create a system for summarizing the continuous audios. The audio files are taken as input and produce the text output as a paragraph, and the using the extractive summarization will be produce the summarized version of it, and finally, the user will be able to get a summarized version of audio unless listening to continuous of audios, which will fulfill the research gap of this project.

1.4.2 Research Objectives

Table 1: Research Objectives

Research Objectives	**Description**	**Learning Outcomes**	**Research Questions**
Research Problem	RO1 - To explore the challenges in Low resourced languages. RO2 - To identify the specific user needs when it comes to summarize RO3 - To identify the research gap for	LO1, LO3, LO6	RQ2, RQ3, RQ4

	summarizers for low resource languages.		
Literature Review	RO4 - To review the existing approaches and limitations on low resource languages RO5 - To identify the techniques, methodologies, algorithms and models related to the speech recognition and text summarization RO6 - To identify the specific challenges when it comes to Sinhala language RO7 - To identify what are the datasets available for the research	L01, L04, LO5, LO8	RQ2, RQ3, RQ4
Requirement Elicitation	R08 - To gather the requirements and feedback from the technical and domain experts **RO9** - To collect the user reviews from the existing systems to improve	LO1, LO3, LO5, LO6	RQ2, RQ4

	RO10 - To identify the requirements related to build the system.		
Design	**RO11** - To create a user-friendly interface for the system **RO12** - To create the required data flows, diagrams to design the architecture	LO2, LO5, LO7	RQ2, RQ4
Implementation	**RO13** - To implement the model on converting the speech to text **RO14** - To develop the text summarization model. **RO15** - To manage the data storage systems. **RO16** - To develop the User Interface.	LO2, LO4, LO5, LO7, LO8	RQ1, RQ2, RQ4
Testing and Evaluation	**RO17** - To provide high performance and accuracy in the system.	LO1, LO5, LO7,	RQ2, RQ3, RQ4

	RO18 - To create the test plan related to the system **RO19** - To perform the testing of unit testing, functional testing and usability testing.		

1.9 Chapter Summary

As a summary of this chapter the author has discussed the problem domain, challenges, research aim and limitations of the existing works. And finally, after considering the existing works the author has come up with the research gap and the contribution for the project. At the end of this the chapter it has stated the research aim, questions and the objectives of the research. Also, the in scope and out scope of the project have been attached in **APPENDIX-B**.

CHAPTER 02: LITERATURE REVIEW

2.1 Chapter Overview

In this chapter the author will be discussing the previous researcher's work related to NLP on speech recognition and text summarization. First of it will discuss the concept map of the system, and then it will discuss the problem in the related domain. Also, what are the technologies and algorithms they have been used to implement the system, what are the contributions and limitations of the system and what are the improvements that can be done to enhance the system will be stated in this chapter. At the end it will discuss the evaluation metrics used in the existing systems.

2.2 Concept Map / Graph

Throughout the concept map it clearly represents the project idea, limitations of existing work, technologies used, what are the exiting works and evaluation metrics. The concept map is attached in the **APPENDIX-A** section.

2.3 Problem Domain

2.3.1. Speech Recognition

Speech is the most suitable and the most natural way of communication between humans. Over the decades computers have been trained for the tasks which humans can do. Speech recognition is also one of the tasks that can be known (Khandare et al., 2019). It is also known as Automatic Speech Recognition or Speech to Text system which basically recognizes the speech of the spoken language and converts it to text. Here the machines are trained to understand human language and communicate with them. Therefore, there are many speech recognition applications and virtual assistant systems built in domains such as telecommunication, healthcare and consumer electronics for specific languages like Google assistant, Siri, Alexa etc. For this type of speech recognition application, the researchers have used NLP techniques which will be important to analyze the meaning of the sentences semantically, to detect phonemes, words and sentences in the input of audios machine learning algorithms were used. And for getting better recognition, signal processing techniques are also used. It improves the quality of audio using preprocessing of data. But when developing speech recognition there are major challenges encountered by the researchers. Some of the challenges are stated below.

2.3.1.1 Recognition of Noisy Background

Background noise is one of the most frequently encountered challenges in speech recognition to get a better output. The noises like natural environment sounds, traffic, music, machinery noises, and conversation of other people alongside makes it hard for the speech recognition system to understand the speech patterns and leads to decrease the performance of the system.

2.3.1.2 Recognition of Speech Type

Another one challenge encountered was the accent of the speakers, speech patterns and the style of the speaking. When it comes to these speech types, it can be changed based on their situation. And also depending on the social environment the people talk differently based on their situation. For example, it can be said the speech type of a person is totally different from how they talk in a casual place and formal place. Sometimes they speak faster to communicate, and they change their style of speaking. This makes the machines complex in understanding and recognizing speech.

2.3.1.3 Recognition Based on the Vocabulary of the Language

In speech recognition the vocabulary of a language makes it hard to understand for the machines, as there are different words with different sentence patterns, including special words and slang. And also, there are various types of grammatical sentence structures and rules for the languages.

2.3.1.4 Recognition of Punctuation Marks

Punctuation marks recognition is also another challenge during the speech recognition. Like commas, question marks and exclamation marks are more important when it comes to a sentence which conveys the meaning of the spoken language.

2.3.1.5 Computational Power

When developing a speech recognition model, a significant challenge is the use of the computational power. For a better speech recognition system, it needs a larger volume of audio data. To handle this, the model requires a lot of computational resources. And gathering more data makes the system complex, and also needs even more computational power.

2.3.2 Sinhala Language

Sinhala is the national and official language of Sri Lanka. And this language is used within 87% of the population in Sri Lanka around 16.6 million of the total population (Languages of

Sri Lanka, 2023). Also, this language is not only a tool for communication but also has a historical view and linguistic importance. When it comes to the grammar of the language, Sinhala has a complex grammatical structure. There are two types of grammar, written Sinhala and spoken Sinhala. Also, in the language 25 types of sentence structures are there (de Silva 2021).

The researchers have separated the languages into two categories, high resource language and low resource language. Compared to the high resources like English language, the Sinhala language will fall into the low resource category as it has contributed very little in the domain of NLP (Manamperi et al., 2018) (Deshpande & Jathirabadkar, 2021). And also, there are above 12 million internet users in Sri Lanka. But still there aren't enough resources available on the internet as it is used by a target audience. So, implementing NLP related projects for the Sinhala language will be a large benefit and there will be a large gap for the contributors that can make contributions to the domain.

2.3.3 Why ASR System for a Sinhala Language

It's challenging when it's compared to a language like Sinhala. It's challenging because the language has a rich cultural sound system, there are a large number of vowels in the language, having a complex vocabulary which sounds similar words, complexity of the grammar and there is a smaller number of datasets available on the internet. Also, there are datasets available for ASR Sinhala in OpenSLR and Kaggle, which can be obtained, but these datasets don't show accurate results, as they are not quality checked. There are also some commercial ASR systems implemented for the Sinhala language. But due to the less accuracy and errors with published dataset on the internet the research on this particular part should be carried out to improve the accuracy (Nadungodage, 2020). As stated above, developing an ASR system for Sinhala language will be highly beneficial for the Sinhala language users for both individuals and organizations.

2.3.4 Summarization Approach

Next step in audio summarization is summarizing the generated text from the audio. The summarization is, which collects the important information from a lengthy content and gives it shorter in depth. Within the modern era people show a lack of listening or reading lengthy contents. They try to find simple ways to understand the information as quickly as possible. For this as a solution the researchers have implemented systems to summarize contents. This saves the users valuable time, summarizing the content instead of reading or listening for lengthy contents (Babar et al., 2013). There are three types of summarizing approaches used to generate the summary. Extractive, Abstractive Summarization and Hybrid summarization are the types. Extractive summarization selects the most important sentences from the paragraph generated as the summary, while abstractive summarization involves its own sentences and generates new sentences and provides the summary. Hybrid summarization involves combining the extractive and abstractive summarization methods (Sharma and Sharma, 2022). Also, the text summarization is not a modern area of research as there are various types of models and tools implemented for high resource languages like English (Deshpande and Jahirabadkar, 2021). When it comes to low resource languages like Sinhala language, there are very little research studies conducted on this domain. Due to the limitation of data like accuracy of the summary output, spelling, grammatical errors and length of the character this particular study also can be carried out (Jayawardane, 2021). This also will be a large benefit for the users of Sinhala language.

2.4 Existing Work

2.4.1 Speech Recognition

2.4.1.1 Speech Recognition Over Other Languages

The field of speech recognition has grown over the decades. For this particular part there are various types of methodologies, algorithms and technologies used for different languages to enhance accuracy and efficiency. Using a domain specific dataset and CDN a speech recognition method was introduced by (Dong et al., 2023). This research was conducted to handle unfamiliar words and language rules. They have used migration learning with pre-trained model parameters. For training purposes, they have collected domain related audio data and they have used n-gram technology to improve the model predictions. With a comparison they have mentioned that the transformer-based models give a good computation compared to the traditional RNN models.

(Wang et al., 2019) has proposed a system for speech recognition on end-to-end models. Here it has been stated of the deep learning algorithms to solve the ASR problems. They have discussed categorizing CTC-based, RNN-transformation and attention-based models in the e2e model. And they have stated that GMM-HMM models perform better compared to the DNN-HMM models. One of the limitations in the e2e model is limited understanding of the context and improving the prediction based on that. And the e2e models don't give a good performance with a background of noisy speech. Another challenge they mentioned is HMM and the e2e models require data alignments, while the HMM uses forced alignments and the e2e model uses soft alignments. Also, it needs a large amount of speech data to achieve good accuracy on e2e models for speech recognition.

Using open-source Sphinx 4 frameworks (Nasib et al., 2018) has presented an approach to convert the speech to text in real-time for Bengali Language. Using Audacity software, they have recorded the speech data from 10 speakers and prepared a dataset which has a reduction of background noise. Also, they normalized the audio, split it accurately and merged the audio data with the precise word mapping for training. They have mentioned that the model has provided good accuracy. But with the limited dataset it is challenging to get a higher accuracy on recognizing the words from new speakers, and also need an improvement on recognizing continuous speech recognition and handling natural speech patterns.

For Hindi language (Upadhyaya et al., 2019) has proposed a speech recognition system using deep learning techniques. Here they have compared 1000 phonetically balanced sentences, which were recorded by 100 speakers. And for extracting features from audio, they have used MFCC. Throughout the deep learning methods, the researchers have mentioned that the CD-DNN-HMM model has good performance over the traditional HMM-GMM models which shows good improvements in the speech recognition tasks. And here also using a large and quality dataset for training will enhance the performance. Also, they state that for low resource languages this approach might be beneficial.

Another research on speech recognition conducted by (Jain et al., 2023) for adaptation child speech recognition using Whisper model. Here, they have utilized child speech datasets namely MyST, PF-STAR, CMU KIDS and an adult speech dataset for the training and testing the model. As same as mentioned above, here also they have use necessary preprocessing techniques before training. And also, the model uses finetuning on the child speech dataset to improve the performance on child speech recognition. They have compared the effectiveness of Whisper model with self-supervised wav2vec models. Child speech recognition is more challenging than the adult speech, such as pronunciation and pitching are more different. They have stated even though with these challenges even that Whisper model performs well than the wav2vec model. And also, to train the model it requires a larger dataset to perform well. This model can be used on fine tuning for the other low resource language datasets, for an efficient ASR system for low resource languages (Pratama and Amrullah, 2024).

2.4.1.2 Speech Recognition Based on Sinhala Language

In Sinhala language also there are some studies conducted on speech recognition. Using deep neural architectures (Karunathilaka et al., 2020) has proposed a system for Sinhala speech recognition. The author has used a dataset from UCSC, LTRL which contains a 25h of speech data involving 70 speakers with 50 females and 20 males. And also, the author has used this dataset and explored different architectures like pre-trained GMM-HMM, DNN, TDNN and combined TDNN+LSTM models. And found that the TDNN performs better showing a lowest word error rate compared to the other models. The author has mentioned limitations of the available dataset is challenging for a low resource language to gain a better accuracy and also with more vocabulary creating a dataset will benefit to gain a high accuracy.

Another one proposed system was conducted by (Gamage et al., 2021) through using the e2e LF-MMI Model. Here it has explored e2e DNN architectures and LF-MMI models compared to other traditional speech recognition models. And using the e2e LF-MMI model they have developed an e2e ASR system for Sinhala language. For the model training purposes they have used a 40h of training data which has 113 native speakers. For pronunciation they used lexicon to map words and created a corpus using active learning methods to generate n-gram language models. Also, they use the Kaldi toolkit for training purposes. Compared to SGMM+MMI, DNN and a combination of SGMM+DNN models, the e2e LF-MMI model shows greater performance. Also, they have mentioned that to achieve a high accuracy large amount of data is needed. And using transfer learning approaches and fine-tuning parameters can make a great deal for a low resource language like Sinhala.

Using interactive voice response of a telecommunication (Manamperi et al., 2018) has developed a speech recognition system for Sinhala language. The goal of the research is to find the Sinhala songs and the digits by speech recognition. Author has gathered a speech dataset consisting of more than 2h, with 45 male and 40 female speakers. Author has mentioned that HMM performed well for the training and it is compiled with 10 digits and 50 songs for a phonetic dictionary and 3-gram was used for predicting the word sequences. Adding more data for the dictionary with more vocabulary, reducing the background noises will make the system performance better.

(Dinushika et al., 2019) has implemented a system for Sinhala speech recognition system on a speech command classification. The researchers have used the MFCC method for extracting the frequency in speech signals, GMM-HMM is used for acoustic modeling and for predicting words N-gram model was used. Here based on a banking domain they have created a new

Sinhala speech corpus having more than 4h of audio data and using MFCC for feature extraction. The researchers have used the GMM-HMM combination model for the training purposes. Also, they have stated that combinations of GMM-HMM models perform well to get a lower word error rate. As the limitations have been highlighted, this system performs only within the banking domain, and it requires more datasets and new modeling approaches for other domains in Sinhala.

2.4.2 Text Summarization

2.4.2.1 Extractive Text Summarization

(Jing et al., 2021) One of the researchers uses a multi-GCN method for an extractive summarization system. Multi-GCN is designed to capture the relationship among the sentences and words. Also, the model consists of semantic and syntactic relationships within the words. This model consists of word block, sentence block and sentence selector for embedding. And it generates the most representative sentences as the summary. And the multi-GCN model has performed better in the CNN/DailyMail dataset. But processing multiple relationship types and graphs requires a larger dataset and it also increases the computational power. Also, the author has stated as future works as it shows a greater performance this model can be extended within the other languages and domains.

Using Lexical chain and BERT (Deshpande and Jahirabadkar, 2021) has explored an automatic extractive summarization. In Lexical chaining it uses WordNet for identifying the cohesive chain within the words and analyzing the relationship between the words. The BERT model is also used for understanding the language context. Also here involves tokenization, embedding and attention mechanism. Comparing the BERT and Lexical chaining they have analyzed that the BERT shows a greater performance on extractive summarization for the low resource languages. Also, as the limitations they have stated that BERT requires a high computational resource and also for training it needs a larger dataset.

(Madhuri and Ganesh Kumar, 2019) presents a statistical method of an extractive summarization using sentence ranking. Also, the summarized version of output is given as an audio which helps the visually impaired people. Here also they used tokenization for input text and tokenized and removed the stop words and tagged. Then weights are given for every tag, and the maximum weight and frequency is calculated. They used the below equation for the weight calculation.

$$\mathrm{Wt} = \frac{frequencyofterm}{Totalno.oftermsindocument}$$

Figure 1: Wt Calculation

$$\mathrm{Wtf} = \frac{frequencyofaterm}{maximumfrequencyoftheterm}$$

Figure 2: Wtf Calculation

So, which shows with high ranks are used for the summarized version and given an audio output. Also, it can be said that the quality of the summary depends on the extract of the key sentences.

To summarize large documents (Zaware et al., 2021) has used a combination of TF-IDF and text algorithms and proposed a system. Before training the model using tokenization and preprocessing methods for removing unnecessary characters and normalization to prepare the data. Then using TF-IDF it calculates the unique words and frequency of the words to create a matrix. So, the system creates a graph based on using cosine similarity and using the Textrank algorithm, it generates a score for the sentence ranking and provides the summary output based on the ranked top sentences. Also, it's mentioned that the combination of TF-IDF-Textrank algorithm has performed better than the TF-IDF algorithm. Also, they have stated further it can be improved by rouge score.

2.4.2.1 Text Summarization on Sinhala Language

For summarizing Sinhala educational content (Rathnayake et al., 2023) has proposed a system which has the ability to summarize Sinhala textbooks using abstractive summaries.

For the dataset the researchers have distributed a questionnaire through some of the main schools and collect data which is related to grade 6 terms. And they have used GPT-3 models for the summarization. As the limitations they have stated, with the lack of dataset and the complexity of the language it's hard to generate an accurate summary of Sinhala. Also the algorithms can be improved for gaining a better result.

Compared to high resource languages like English and France, there are few studies conducted on Sinhala text summarization. To address this gap (Jayawardane, 2022) has proposed a

Sinhala text summarization to overcome the problem of summarizing Sinhala government gazettes. For the summarizing purposes both the abstractive and extractive summarization methods have been used. Using linguistic and statistical features it identifies the most important sentences and produces the summary. Here the sentences have been tokenized and removed the special characters of it. Then it has identified the relevant keywords and assigned weights for scoring the sentences. Using it the words which have a low scoring rate are removed and provide the summary. Also using 450 actual Sinhala gazettes the author has been used to evaluate it compared with the author created summaries and summaries which machines generate. This research is only specific to the Sinhala gazettes which does not apply to other related documents.

2.5 Technological Review

2.5.1 Data Preparation

For speech recognition models the main requirement is the quality of the data. Through that only it gives the performance and the applicability to the system. Also, for speech recognition model training it requires a larger dataset, also the dataset effects on the quality like accent of the speakers, speakers count, background noises, the speed of the voice, vocabulary etc affect this. But there are larger datasets and high-quality datasets created by researchers on high resource languages like English (Panayotov et al., 2015) separated for training, testing and evaluation purposes. But when it comes to low resource languages it is challenging to find a high quality of audio data (Besacier et al., 2014), also as mentioned above the datasets affect the quality.

2.5.2 Data Preprocessing

Once the data is collected, before training the model the data should be cleaned and preprocessed. Throughout this it improves the quality and efficiency of training the model and shows a better performance in the system. Removing the Background noise is one of the preprocessing steps, which helps to increase the recognition of the words more accurately without any confusion of external sounds (Besacier et al., 2014). Also removing the duplicate recording of the audio enhances the diversity of the training (Alharbi et al., 2021). Normalization is another step of preprocessing the audio data. It adjusts audio data which consist of having the lower volume of audio and higher level of audio into a standard range (Alharbi et al., 2021). Also cleaning the speech transcription is also a main preprocess step.

Reviewing the labeled audio and transcription data manually will help to improve the quality. As some of the transcriptions include unwanted characters like punctuation marks (Glackin et al., 2019). Also, when it comes to other languages, some of the data includes English words too. Some of the transcription errors can occur and may be challenged when the model trains.

2.5.3 Algorithm Selection

2.5.3.1 Speech Recognition Techniques

Hidden Markov Model

HMM is one of the mainly used algorithms in speech recognition. This is used for modeling the sequential data, to analyze the context within a timeframe. Also, this helps to model the relationships between the words or the phonemes. Also, this has the ability of recognizing 80% of speech signals (Jendoubi et al., 2013). HMM provides benefits on speech recognition like it can be customized into (phonemes, words and phrases) various levels of detail and can be incorporated for grammar and pronunciation. But as the limitation it needs larger amounts of data to train and take a lot of computational power, hard to understand on the similar sounding words (Yu and Deng, 2015).

Deep Neural Networks

DNN also has produced greater performance on speech recognition tasks. Over the traditional approaches this has provided lower word error rates and good accuracy (Hinton et al., 2012). DNN has the ability to recognize speech in different accents, speaking patterns and in other environmental conditions (Yu and Deng, 2015). Also, there are certain limitations like it requires a larger dataset for the training to achieve a high performance in a low resource languages and other related domains (Amodei et al., 2016).

Hybrid approaches - Time-Delay Neural Network (TDNN) and Long Short-Term Memory (LSTM)

These hybrid approaches are also considered by researchers for speech recognition tasks. LSTM can handle long-range dependencies in sequential data and TDNN has the ability to handle sequential data in speech signals. (Markovnikov et al., 2018) has used this technique for the Russian language and it has produced better accuracy.

Gaussian Mixture Models (GMM)

GMM is also often used in speech recognition tasks. This has the simplicity and effectiveness in featuring vectors in speech. Also, it assumes all the data points and generates a mixture of a finite number of unknown parameters which can be robust to various in speech. But the model couldn't figure out the relationship between speech frames in understanding spoken languages, also it takes a high power of computation (Kenny, 2006).

Recurrent Neural Networks (RNN)

RNN is also widely used in speech recognition tasks. This model is specially designed for handling sequential data. This has the ability to capture the context in speech, processing the input sequences of any lengths and maintaining hidden states. Also, the researchers have used RNN combinations of LSTM and GRU for speech recognition tasks. And it can capture long-term dependencies in speech sequences (Graves et al., 2013). But there are some limitations like it is time-consuming as it may require a large dataset and computational power. And it also has difficulty in training as it needs advanced techniques and hyperparameter tuning while training (Sherstinsky, 2020).

Whisper Model

The Whisper model is newly introduced by OpenAI for transcribing the speech to text for English. It has been trained on 680,000 hours of larger dataset (Introducing Whisper, #). Also, it has the ability to robustness on the background noise and accent. And it is an open-source model which can be used for future research. And there is a limitation which the model can only transcribe for the 30 seconds of audios. So lengthy that audio can be split and chunk for 30 seconds.

2.5.3.2 Techniques on text summarization

Algorithm Approaches

The researchers have used techniques on Frequency-based methods for extractive summarization approaches. This method's advantage is its simplicity. This method can be implemented quickly to summarize the contents. The commonly used frequency-based algorithm is TF-IDF algorithm. On high frequency words it removes the stop words and generates the summary (Allahyari et al., 2017). Also, there is another widely used algorithm called graph-based method for summarization. This includes both the extractive and abstractive summarization. TextRank is one of the widely used graph-based algorithms. This doesn't need labeled data for training purposes. Also, the advantage of it is for new domains it can be used without a dataset. This produces a graph of similar meaning sentences and uses the PageRank mechanism to find the most solid sentences, and finally the top ranked sentences are generated as the summary (Mihalcea and Tarau, 2004). LexRank is also an unsupervised graph-based approach which produces a graph using the TF-IDF cosine similarity between the sentences. In that the nodes appear the sentences and edges appear the weighted similarity between sentences. And using the PageRank mechanism the higher ranked sentences are provided as the summary (Erkan and Radev, 2004).

Other Approaches

Deep learning and machine learning approaches also have been used widely for text summarization tasks by researchers for learning the complex representation of the sentences in order to achieve a greater performance in the summarization, both in extractive and abstractive summarization. The researchers have used deep learning methods for identifying sentence patterns learned from the training data (Denil et al., 2014), identifying the more flexible and important sentences etc. Also including semantic and syntactic features, these ML and deep learning approaches have the ability to be incorporated in a wide range of tasks. But these deep learning models require larger datasets to achieve a high performance which is specially for the low resource languages (Liu and Lapata, 2019).

2.6 Evaluation

2.6.1 Evaluation on Speech recognition

For the accuracy, usability, the effectiveness and the overall performance of the speech recognition systems depends on the evaluation. There are several metrics used to evaluate the speech recognition models.

Word Error Rate (WER)

WER mostly used metrics on evaluation purposes for speech recognition systems (SmartAction, 2021). This is used to calculate the error rate of the words in speech using substitution, insertions and deletions of the word.

$$\text{Word Error Rate (WER)} = \frac{\text{Substitutions + Insertions + Deletions}}{\text{Number of Words Spoken}}$$

Figure 3: WER Calculation

Sentence Error Rate (SER)

SER is another evaluation metric to find the error rate of the sentences that are not recognized correctly.

Character Error Rate (CER)

CER is also an evaluation metric like WER, but this handles the error rate of the characters. This has been useful for identifying the accuracy of the recognized characters in the low resource languages and languages which have a complex vocabulary. This also uses a calculator to find out the CER (Violeta and Toda, 2023).

$$\text{Character Error Rate (CER)} = \frac{\text{Substitutions + Insertions + Deletions}}{\text{Number of Characters}} \times 100$$

Figure 4: CER Calculation

2.6.1 Evaluation on Text Summarization

Recall-Oriented Understudy for Gisting Evaluation (ROUGH) is a commonly used evaluation metric for the automatic summarization. It focuses on calculating the recall content. This calculates the word sequences, word pairs between the generated summary (Lin, 2004).

Also, there is another text summarization evaluation metric called F1 score. This uses a calculation on precision and recall. Precision involves the true positive results divided by all the positive results, while the recall involves the true positive results divided by the number of all samples which are identified as positive.

$$\text{F1} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \times 2$$

Figure 5: F1 Scor Calculation

2.6 Chapter Summary

This chapter includes the concept map for this system, has been discussed about the problem domain in depth and what the existing researchers have done in the speech recognition and text summarization domain, what the limitations and advantages they have encountered, what the algorithms and models and technologies are used to implement the systems. At the end it discussed what evaluation metrics have been used.

CHAPTER 03: METHODOLOGY

3.1. Chapter Overview

In this chapter, the author has named in detail which methodology type is used for this project, the required tools, techniques, scope, skills and about the deliverable dates are discussed which is needed to carry out this project. And at the end the author has mentioned the risk and mitigation plan while conducting this research project.

3.2 Research Methodology

Table 2: Research Methodolgy

Philosophy	Author has selected research Pragmatism here. The research is based on audio and text, the author has chosen Pragmatism as the suitable approach for the research. philosophy, as it is used for the qualitative and quantitative research data when prioritizing methods and approaches.
Approach	The author has selected, Deductive research approach here. The author will be used for testing the existing solutions as qualitative and quantitative research data will be used.
Strategy	Author has chosen Questionnaires (Survey), and Interviews to collect feedback from the users. Additionally, the brainstorming also will be used.
Methodology Choice	Author has chosen the mixed method. As the research uses qualitative and quantitative data.
Time horizon	Author has chosen the Cross-sectional frame. As the data is collected at one time.
Data Collection and Analysis	Author has chosen Interviews, surveys to collect data to the project.

3.3 Development Methodology

3.3.1 Requirement Elicitation Methodology

As the feedback purpose the author will be gathering information from surveys and conducting some interviews. Moreover, the author will identify what are the required tools and technologies needed for the project by the existing work and feedback gathered from surveys and interviews.

3.3.2 Design Methodology

Here as the design methodology, the author has chosen the SSADM compared to other design methodologies. As it has the ability to structure the design, analyze and develop the system successfully.

3.3.3 Programming Paradigm

The author has chosen the prototyping model as the programming paradigm. As it should be designed, implemented and tested to get quality and a successful output.

3.3.4 Evaluation Methodology

Compared to prototyping testing, model testing and benchmarking the author has chosen prototyping testing as the evaluation methodology. As it has the ability to test the body types separately.

3.3.5 Solution Methodology

The author will have a proper plan to gather the required technologies and tools, design the UI prototypes, develop, test, evaluate, deploy and documentation to complete the project at the given time.

3.4 Project Management Methodology

Here as the project management methodology, the author has chosen the Agile Prince 2. This was chosen because this has the ability to help the author complete this project within the time period and produce a good quality system at the end. And also, without any rush it has the ability to have a proper plan to manage the project works and complete the project within the time.

3.4.1 Schedule

3.4.2.1 Gantt Chart

The research project Gantt chart is attached in the **APPENDIX-C.**

3.4.2.2 Deliverables and Date

Table 3: Deliverables and Dates

Deliverable	**Date**
Draft version of Project Proposal	1st September 2023
Finalized Project Proposal	5th October 2023
Literature Review	31st October 2023
SRS (Software Requirement Specification)	27th November 2023
Proof of Concept	21st December 2023
PSPD (Project Specification Design and Prototype)	29th January 2024
Minimum Viable Product	7th March 2024
Thesis (Final Project Report)	4th April 2024

Table 4: Deliverable dates

3.5 Resource Requirements

3.5.1 Hardware Requirements

Table 4: Hardware Requirements

Requirement	**Justification**
Core i7 10th generation.	To provide a good system performance for the project

16GB RAM	Has the ability to manage the datasets related to the project and speed up the system without any lag
Graphics card	To train the models related to the project
Storage Space more than 50GB	To store the applications related to the project datasets, files, documents etc.

3.5.2 Software Requirements

Table 5: Software Requirements

Requirement	**Justification**
OS (Windows 10 upper/ Linux)	To handle the heavy software and hardware in the system. Windows 11 with 64 bit is used for this project.
Google Collab	This is a cloud-based platform, and it helps to test and train models for the project.
Google Docs/ MS Word	This is used to documentation the report related to the project.
Python	This is used for the backend purposes of the project.
GitHub	This is used to store the code, images and docs related to the project.
Figma	This is used to design the Wireframes and UI prototype for the project.
Draw.io	Used to create the diagrams required for the project.
Google Drive	This is to save the project related documents and code.

Zotero	This is used to manage the citation and references related to the project.
Python	This is used to develop the backend of the system and text summarization model

3.5.3 Skill Requirements

- An understanding of finetuning models.
- Knowledge of developing web-based applications.
- An understanding on NLP techniques
- An understanding on model training and dataset creation

3.4.4 Data Requirements

- Dataset for training and testing the Speech recognition model.
- Text summarization

3.6 Risks and Mitigation

Table 6: Risks and Mitigation

Risk	Probability of Occurrence	Magnitude of the loss	Mitigation Plan
Knowledge on the techniques and algorithms that will be used in the project.	5	3	Following the necessary research papers and other resources
Project delay	2	3	Will manage the project with the deliverable dates.

The complexity, as there is a lack of resources available on Sinhala language	2	3	Will handle it with the domain experts and other existing work
The system issues	5	4	Author will be using an alternative system. And will be using online platforms for documentation purposes and GitHub to store the updated code related to the project at time.

2.5 Chapter Summary

As the summary of this chapter the author has discussed the methodology type, resources and about deliverable dates that will be used to complete this project successfully. In order the author has mentioned the risks during the project and how to overcome them.

CHAPTER 04: SOFTWARE REQUIREMENT SPECIFICATION

4.1 Chapter Overview

In this chapter it provides a rich picture diagram and an onion model identifying the stakeholders of the system. And the author will be exploring the requirement elicitation including literature review, surveys and interviews. Moreover, it will discuss the use case diagram, functional and nonfunctional requirements of the system.

4.2 Rich Picture Diagram

The given rich picture diagram below provides a helicopter view of the wider environment of the system. And it clearly states the stakeholders interacting with the system and others. It also highlights the negative and positive aspects of the system.

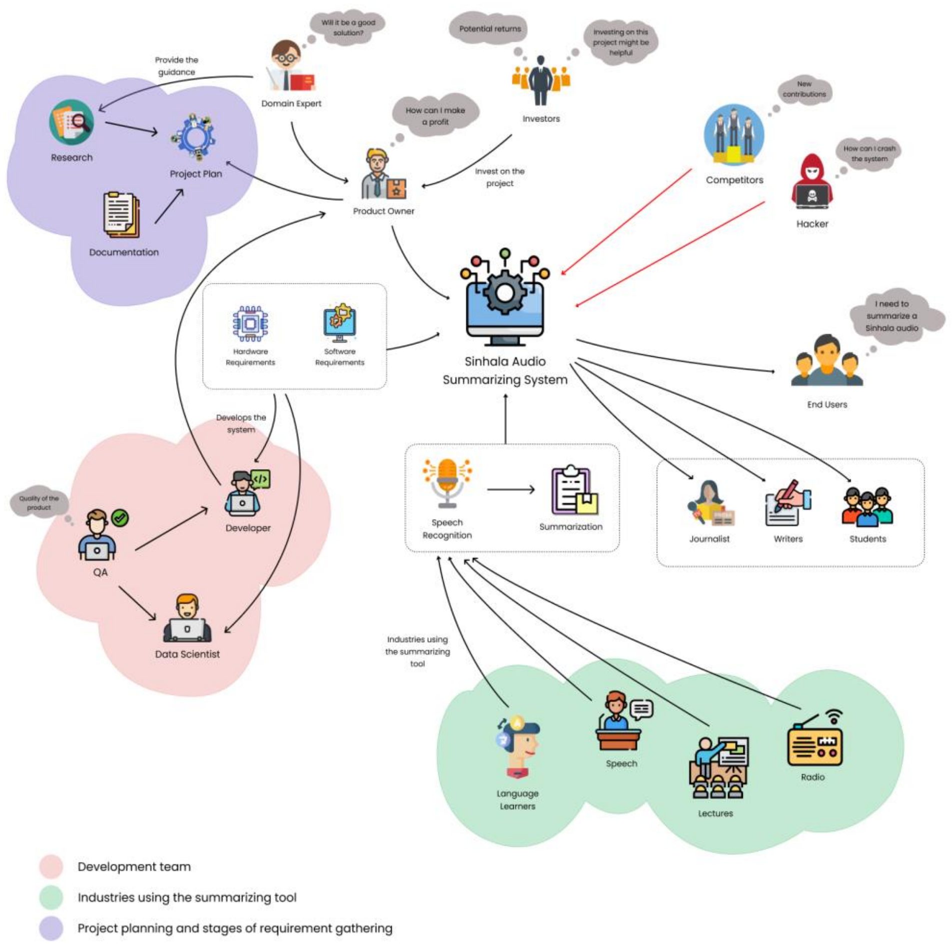


Figure 6: Rich Picture Diagram

4.3 Stakeholder Analysis

4.3.1 Stakeholder Onion Model

The stakeholder onion model below provides each stakeholder in the system which is in different environments. This helps the author to identify the stakeholders with positive and negative structure and an organizing part of the project.

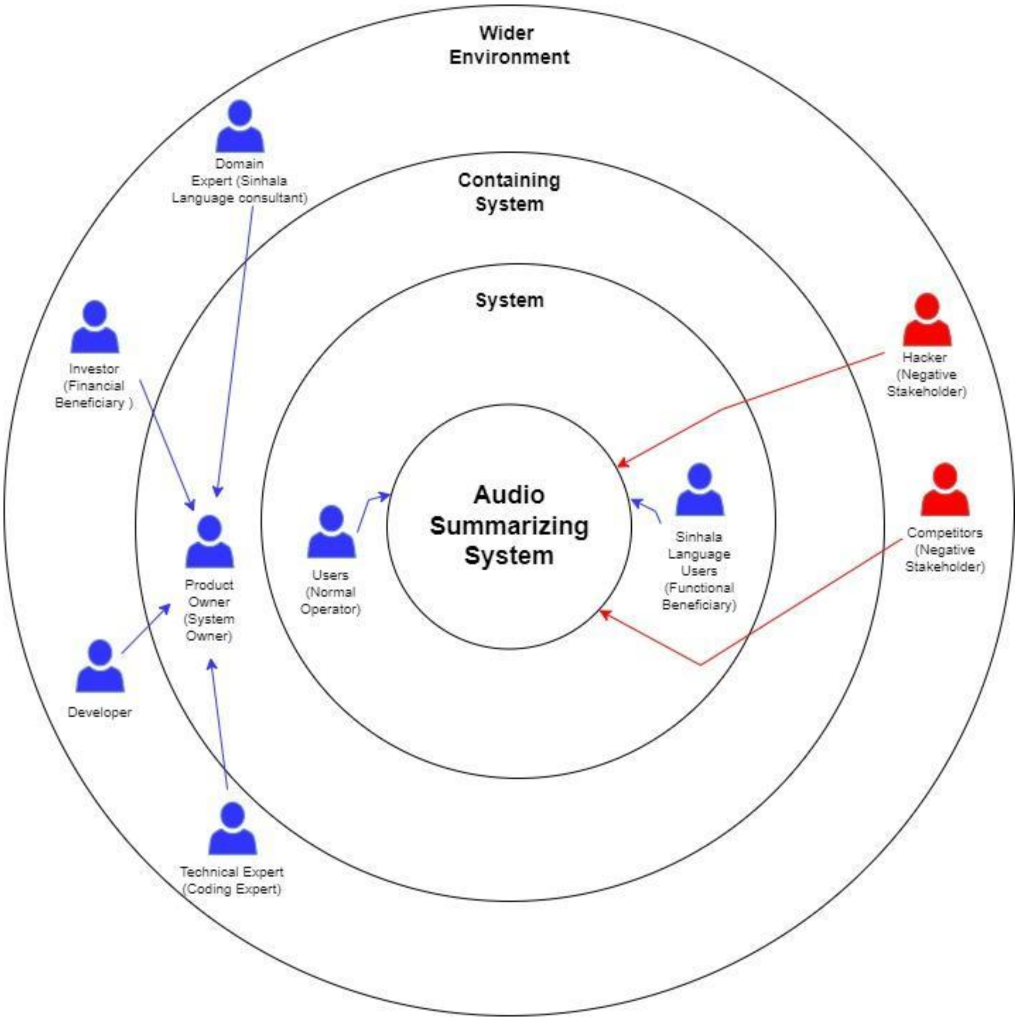


Figure 7: Stakeholder Onion Model

4.3.2 Stakeholder Viewpoints

Table 7: Stakeholder Viewpoints

Stakeholder	Stakeholder type	Description

Users	Normal operator	Who will be using the system to summarize Sinhala audio files
Sinhala language users	Functional beneficiary	They are the ones who will be benefited from the system
Product owner	System owner	The product owner is who will be handling the system
Technical expert	Consultant/ Coding expert	Who will be providing/guiding on the coding requirements
Developer	Operational Maintainer	Develops the system using the gathered requirements
Investor	Financial Beneficiary	Who will be financially investing on the project and improve the system to get
Domain expert	Consultant	Will guide on the project with necessary requirements
Competitors	Negative Stakeholder	Will be implementing similar systems
Hackers	Negative Stakeholder	Tries to crash the system

4.4 Selection of Requirement Elicitation Methods

Requirement elicitation is the process to gather requirements from stakeholders what are the expectations. There are several methods to carry out to gather the requirements. Here the author has selected the literature review, survey and interviews to gather the requirements.

Table 8: Selection of REM

Literature review
LR was selected, as it has the ability to identify the research gap of the existing work and make a contribution to the field. Using the gathered requirements (techniques used), it helps the author to improve the system with a better result.

Survey
Distributing surveys or questionnaires will help the author to understand the user's needs, experience of the existing systems and what should be improved. This will be a suitable method to gather requirements for a larger number of populations.
Interviews
Interviews will help to gather requirements in detail. The author focuses on having interviews with the domain and technical experts. This will help the author in gathering the requirements to fulfill the system on the technical and domain wise, identifying and clarifying the specific needs to the project.

4.5 Discussion of Findings

4.5.1 Findings from Literature Review

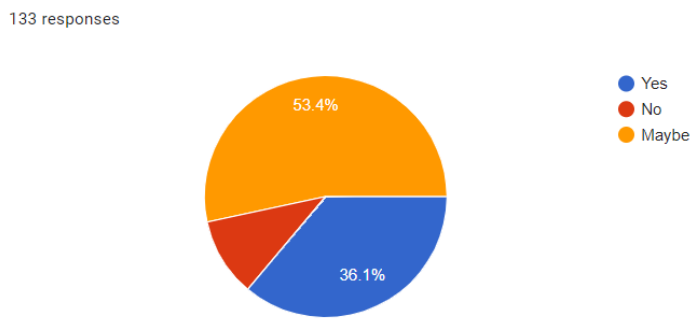
Table 9: LR Findings

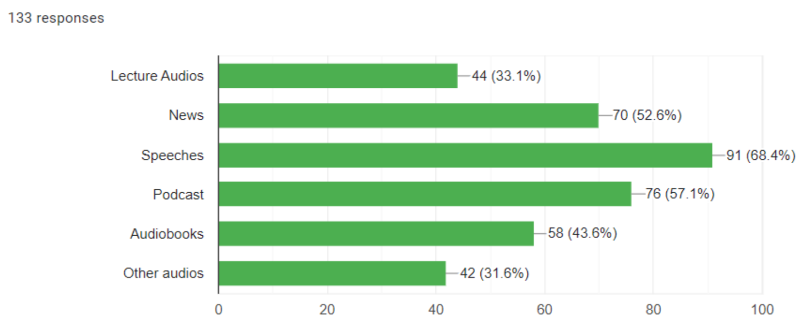
Findings	**Citation**
The HMM performs a far better accuracy than the other traditional approaches. And to get a high level of accuracy the dataset should be with more vocabulary.	(Weerasinghe et al., 2020)
For a better summarization result semantic features can be used.	(Shah et al., 2019)
Compared to TDNN+LSTM and DNNs, TDNN+LSTM shows a lower WER. But still in speech recognition tasks TDNNs perform much better.	(Karunathilaka, 2020)
The ASR system provides a lower accuracy in sentence recognition compared to IVR.	(Dinushika et al., 2020)
Figuring out the relationship between the words will give an accurate summary.	(Jing et al., 2021)

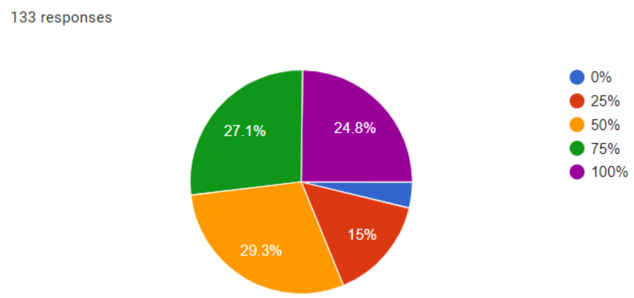
4.5.2 Findings from Survey

A questionnaire was shared publicly, as it is hard to get the response by the target audience for the particular application. The author was able to collect 133+ responses. The responses and the aim of the questions are stated below. In the **APPENDIX-D** screenshot of the survey can be found.

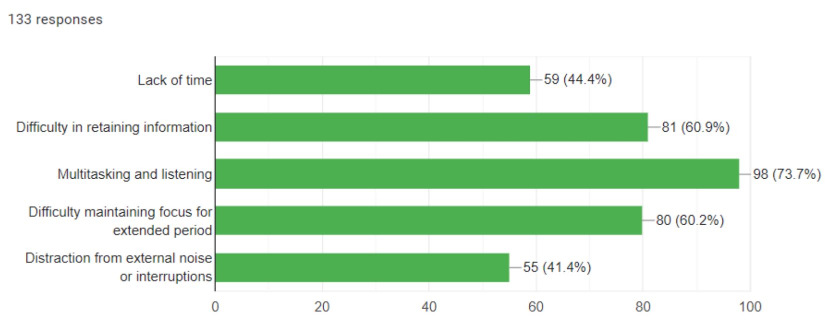
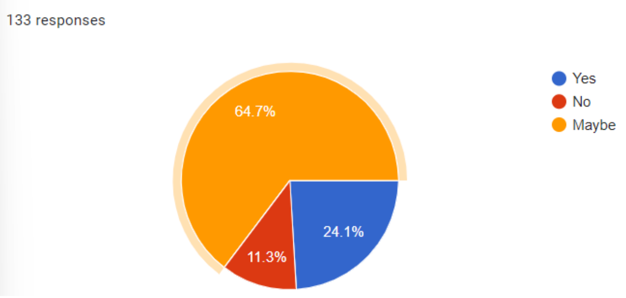
Table 10: Survey Findings

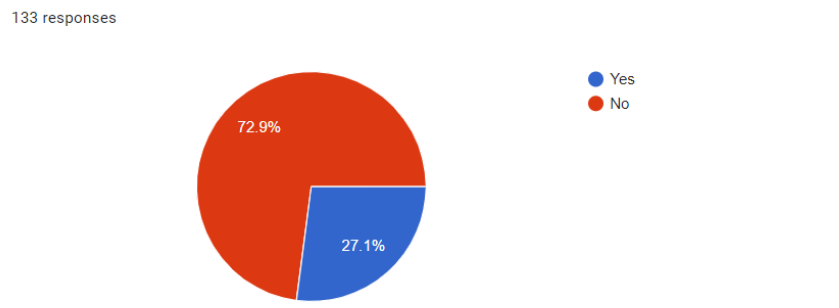
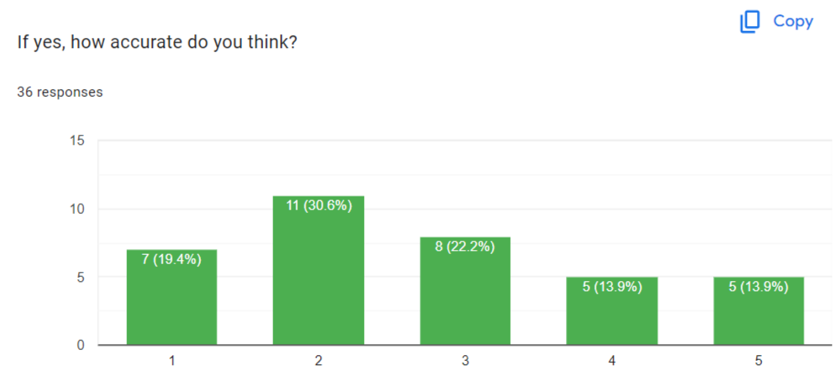
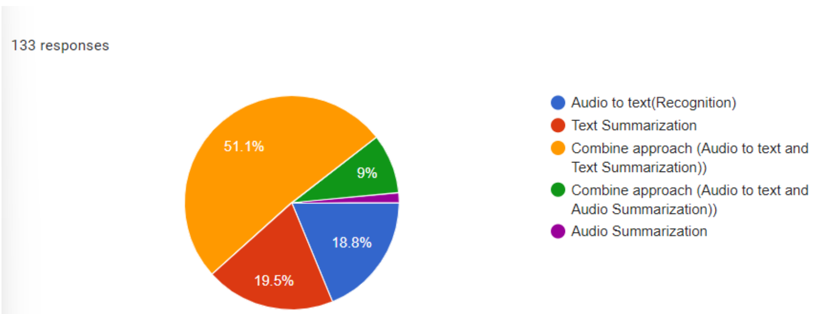
Question	**Are you a person who listens to Sinhala audio content?**
Aim	To find out if the participant is a person who listens to Sinhala audio contents.
Findings	 With the above result, it can be said there is a majority of users who listen to Sinhala audio contents. So there is the potential of users who are the target ordinance for this system.
Question	**What type of Sinhala audio do you listen to?**
Aim	To find out what the Sinhala audio content they used to listen to.

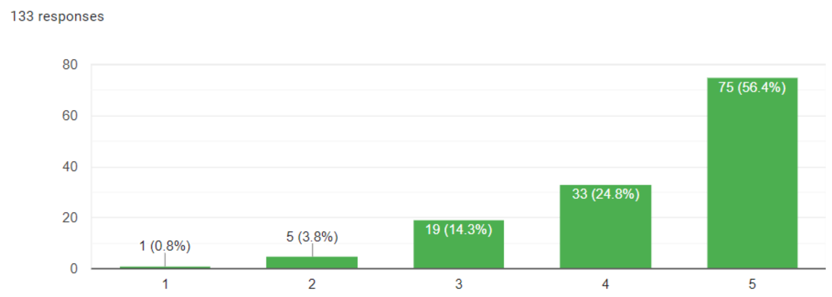
<table>
<tr><td>Findings</td><td>


It states that there is a huge amount saying that they listen to Sinhala speeches, next podcast and news. So, there is a noticeable number of users who listen to this type of lengthy audio, which means that from the proposed system they could be benefited.</td></tr>
<tr><td>Question</td><td>When it comes to a lengthy audio, how much will you complete listening?</td></tr>
<tr><td>Aim</td><td>From the above question it identifies how much they will complete listening lengthy audio contents.</td></tr>
<tr><td>Findings</td><td>


It states that a large number of users don’t 100% complete listening to the audios. Because of this they might miss the useful information in the audio.</td></tr>
<tr><td>Question</td><td>What are the challenges you face while listening to long audio contents?</td></tr>
<tr><td>Aim</td><td>This to find out what are the challenges they encountered while</td></tr>
</table>

	listening to lengthy audio files
Findings	 Most of the participants say that multitasking and listening to an audio file is the hardest, next difficult to retain information, difficult to maintain focus. So, this application will focus on these particular challenges.
Question	**Would you like to get a summarized text version of your lengthy audio?**
Aim	To find out the importance of implementing this system
Findings	 There is a considerable number of users who are saying they need a summarized version of audio files. This means this system would benefit a large number of users.
Question	**Have you use any platforms to summarize a Sinhala lengthy audio file**
Aim	This is to find out if the user has used any existing systems, if yes how was the experience

Findings	  In the above pie chart, it clearly states that there is a large number of users saying that they haven't used any other existing applications, and there is a considerable amount saying who used the existing applications also does not provide an accurate result.
Question	**What are the features you would want in this type of application?**
Aim	This is to find out what are the futures there are expecting through this system
Findings	 There is a huge amount saying that they need a combined approach (Audio recognition and Text summarization). So, it

	clearly states implementing a combined approach will be benefited through a larger amount.
Question	**How useful will this application be for you?**
Aim	To find out the users who will be benefited through this application
Findings	 There is a larger number of users that might be benefited through this application.

4.5.3 Findings from Interview

The interviews were conducted within the domain related and technical experts.

Table 11: Interview Findings

Codes	**Theme**	**Conclusion**
'Existing datasets or Audio to Text' 'Model implementation' 'User-friendly UI'	Dataset Collection and Speech recognition model	The experts mentioned that to look for publicly available Sinhala ASR datasets. So, through that dataset they said look out of the speakers, the accent and the recording conditions. And they mentioned that implementing a model for Sinhala speech recognition will be an

		advantage. When it comes to the UI, they mentioned making it simple, so the user can easily summarize the audio based on their input.
‘Existing Sinhala applications does not have summarization based on audio’	Research gap and scope of the project	There isn't a summarization system for audio for Sinhala language. So, they mentioned the research gap is valid and will be good to address. Throughout the audio recognition correcting the grammar and the spelling of the sentence will be highly recommended.
‘Background noise of an audio’	Background noise removal	There are approaches like denoising techniques, spectral subtraction, Wiener filtering they mentioned for filtering out the background noises.
‘Summarizing techniques’	Text summarizer	They mentioned that there are two ways to summarize a text. Extractive and abstractive. When it comes to this system, they recommended having an extractive summarization approach for the summarization purpose.

4.6 Summary of Findings

Table 12: Summary of Findings

Id	Findings	Literature Review	Survey	Interview
1	Expresses a need of Sinhala audio summarization system	✓	✓	✓
2	Implementing a model for audio recognition	✓		✓
3	The relationship within the words will give an accurate summary			✓
4	Generate summary with correct grammar and spellings	✓		✓
5	Identify the suitable dataset		✓	✓
6	Use pretrained models to get high accuracy	✓		✓
7	User friendly and simple interface for the system		✓	✓

4.7 Context Diagram

The context diagram provides the system boundaries and the interaction between the users. In the below diagram it shows the user has to upload an audio file or record an audio file to the system. And the system will generate the summary of the audio to the user.

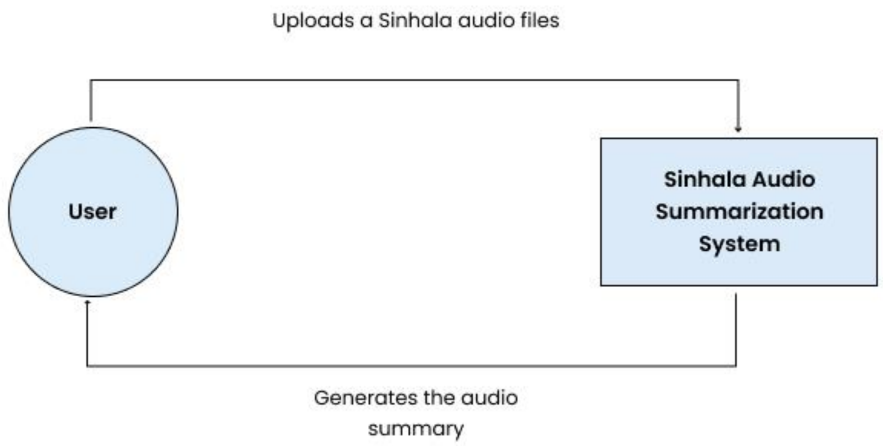


Figure 8: Context Diagram

4.8 Use Case Diagram

The below use case diagram describes the functionalities of the system, including the actors and other related components.

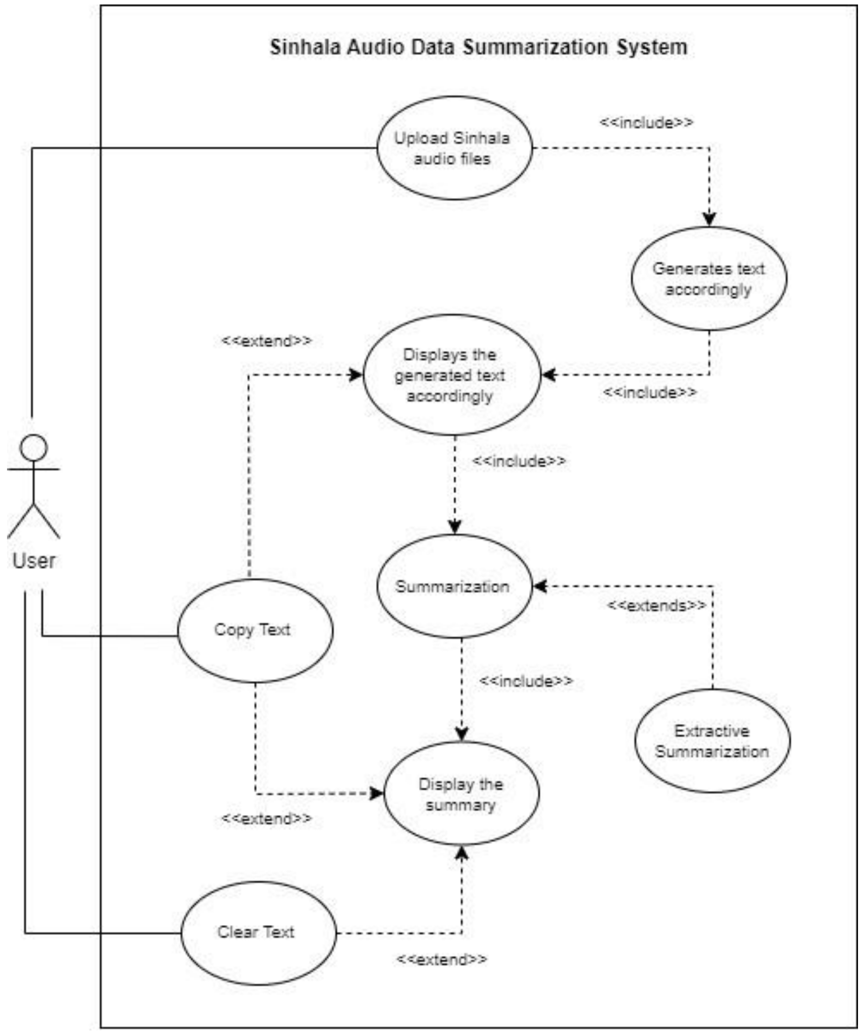


Figure 9: Use Case Diagram

4.8 Use Case Specification

Here the main use case specifications are mentioned other have been attached in **APPENDIX-H**.

Table 13: Use Case Specification (1)

Use Case Name	Upload Sinhala audio files
Use case ID	UC1
Description	The user needs to upload multiple Sinhala audio files
Priority	High

Actors	User
Pre-conditions	None
Post-condition	User is able to see the generated text, by given audio accordingly as a paragraph
Extended use case	Recognize the audio files
Included use case	None
Main flow	1. User uploads multiple Sinhala audio files 2. The system recognizes the audio files and converts the audio into text. 3. And combines as paragraph.
Alternative flow	None
Exceptional flow	If rather than audio files or one audio files is uploaded alerts will be popup or displayed.

Table 14: Use Case Specifications (2)

Use Case Name	Summarization
Use case ID	UC2
Description	The user needs to click the summary to get the summarized version of the audio files generated text
Priority	High
Actors	User
Pre-conditions	None

Post-condition	User has to get a summary of the uploaded audio files
Extended use case	Extractive summarization
Included use case	None
Main flow	1. User clicks the summary button. 2. System generated the extractive summarization of the uploaded audio files
Alternative flow	None
Exceptional flow	If there aren't more than two sentences in the generated text will generates the same results as the converted text input.

4.9 Requirements with Prioritization

The MoSCoW principle is used to manage priorities of the requirements in the project effectively.

Table 15: MoSCoW Principle

Must have(M)	The feature requirement which are mandatory to be implement the system
Should have(S)	Requirements or features which are important, but not necessary for the prototype.
Could have(C)	These are nice to have. Can be considered as future works to the system.
Will not have(W)	The functionalities are out of scope, which will not be implemented on the system.

4.9.1 Functional Requirement

Table 16: Functional Requirement

FR ID	Functional Requirement	Priority Level	Use case
FR1	The system should be able upload multiple audio files to the system	M	Upload multiple audio files to the system
FR2	The system must not support other than audio file	M	Upload audio files to the system
FR3	The system should generate the Sinhala text from the audio accordingly	M	Converts into text
FR4	User should be able to copy the generated text	S	Copy the text
FR5	User should be able to reset the generated text	S	Clear the text
FR6	User should be able to summarize the generated text	M	Summarization
FR7	User should be able to copy the summary	S	Copy the text
FR8	User should be able to reset the summary	S	Clear the text
FR9	The user should be able to upload other language audios	W	Upload Sinhala audio files
FR10	The user should be able to upload videos/ files	W	Display an error message
FR11	The system generates summary of other languages	W	Display an error message
FR12	The system stores the input audio files or the generated result	C	N/A

4.9.2 Non-Functional Requirement

Table 17: Non-Functional Requirement

NFR ID	Requirements	Non-Functional Requirement	Priority Level
NFR1	Performance	The system should be able to upload multiple audio inputs. And without taking much it should be generating the text accordingly	S
NFR2	Usability	The system should be user-friendly, understand the system functionalities and should be easy to operate to the user	M
NFR3	Security	The system should be protecting the user data while preventing unauthorized access	M
NFR4	Maintainability	The system related code should follow coding standards and should be well structured for future use	S
NFR5	Scalability	The system should run smoothly without crashing while the system is used by multiple users	C
NFR6	Quality	The ASR system should generate the user a quality output and when it summarized also it should produce a quality result	S

4.10 Chapter Summary

This chapter discussed the Rich picture diagram, identified stakeholder for the system, the onion model. And it has been discussed what are the findings from the literature review, Survey and conducted interviews. At the end it has stated the context diagram, use case diagrams, functional and non-functional requirements of the system.

CHAPTER 05: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Chapter Overview

During this project the author considered social, legal, ethical and professional issues will be discussed in this chapter.

Table 18: SLEP Issues

Social	**Legal**
• The gathered data from survey doesn't collect any personal information's from the participants. • Also, the gathered data from the survey was not published or stored. • The permission granted participated interviews names are added, others were maintained as anonymous.	• The dataset used for this project was publicly available to the contributors. • The used pre-trained models, languages, tools, algorithms and frameworks in the project was open source.
Ethical	**Professional**
• The research papers gathered for this project from conferences and publications are well cited. • The project documentation is free from the plagiarism and false information.	• The used software's during the project was open source. • The limitations of the project are mentioned to the evaluators within the feedback session and stated in the report.

5.3 Chapter Summary

In this chapter it has been discussed about what author has considered on SLEP issues.

CHAPTER 06: DESIGN

6.1 Chapter Overview

This chapter discusses the designs and architectures related to the system. There is the system architecture design, component diagrams, data flow diagrams and user interface designs and flow charts.

6.2 Design Goals

Table 19: Design Goals

Design goal	Description
Performance	As the system takes multiple audio inputs, the system should run smoothly without any failure and a delay while the system should provide a high-quality and efficient summarized output.
Usability	The UI of the system should be more simple, clean, straight forward and allow the users to easily navigate through the system functionalities to upload audio files and get the summarized output.
Scalability	The scalability of the system should be capable of performing with less time to recognize the audio file and generate the text. And the system should be able to upload multiple audios and generate the summaries.
Reusability	This project-related codes and other relevant components should be able to be reused for another project.
Correctness	The system generates the text from recognizing the audio first and generates the text accordingly. And the multiple audio outputs should be combined. Else the grabbed information will be misled.

6.3 High level Design

6.3.1 Architecture Diagram

The following high-level diagram consists of three tier architecture which has the presentation tier, logic tier and data tier.

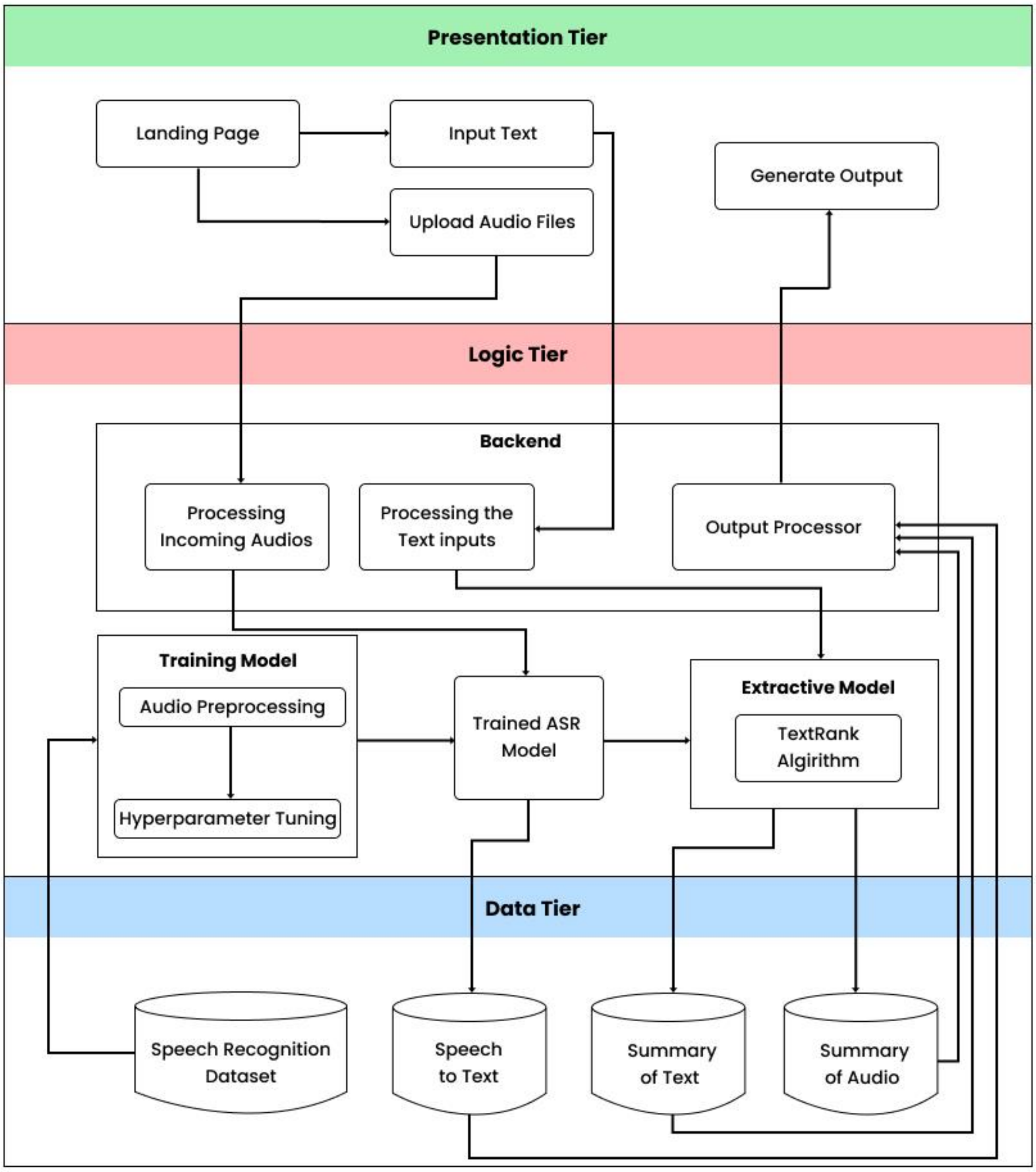


Figure 10: High Level Architecture

6.3.2 Discussion of tiers/ layers of the Architecture

Data Tier

- Speech recognition dataset **-** This will be used to convert the input audio files into text format.
- Speech to text – This will generate the text from the input audio files to the user.
- Summary of text – This will generate the summarized version of text to the user.
- Summary of audio – This will generate the summarized version of audio files to the user.

Logic Tier

- Dataset preprocessing - Before training the model the dataset should be preprocessed. For the audio summarizing, the audio preprocessing is used.
- Model training – The preprocessed data fed to the model, for speech recognition it learns to make predictions and gives effective outcomes.
- Processing audio files - This is where the audio is converted into text. One by one audio will be fed to the model and generate the text combining as a paragraph.
- Processing text input - This is where the text is summarized.
- Extractive Model – Using Text Rank algorithm it generates the summaries according to sentence score.
- Output Processor – This will be used to get the audio to text, text summarization or the audio summarization output and send back to user.

Presentation Tier

- Landing page - This provides a user friendly and understandable user interface for the user to navigate through the system functionalities.
- Upload audio files - The system allows the user to upload audio files to the system.
- Input text - This displays the generated summary version of the provided text by the user.
- Generate summary - This displays the generated summary version of the provided audio files or the text input by the user.

6.4 System Design

6.4.1 Choice of design paradigm

After a clear understanding of the design paradigm, SSADM was chosen by the author over OOAD. SSADM is more suitable for this project, as it is systematic, perfectly structured and easy for prototyping. There are several factors for rejecting OOAD. One of those is that an object-oriented approach doesn't benefit this project, as it is based on a data science component. SSADM has the ability to improve the accuracy, efficiency and documentation of information systems.

6.5 Detailed Design Diagrams

6.5.1 Data Flow Diagram

The level 01 DFD provides a basic understanding of the system. And the level 02 DFD provides a more detailed version of how the system function elaborates.

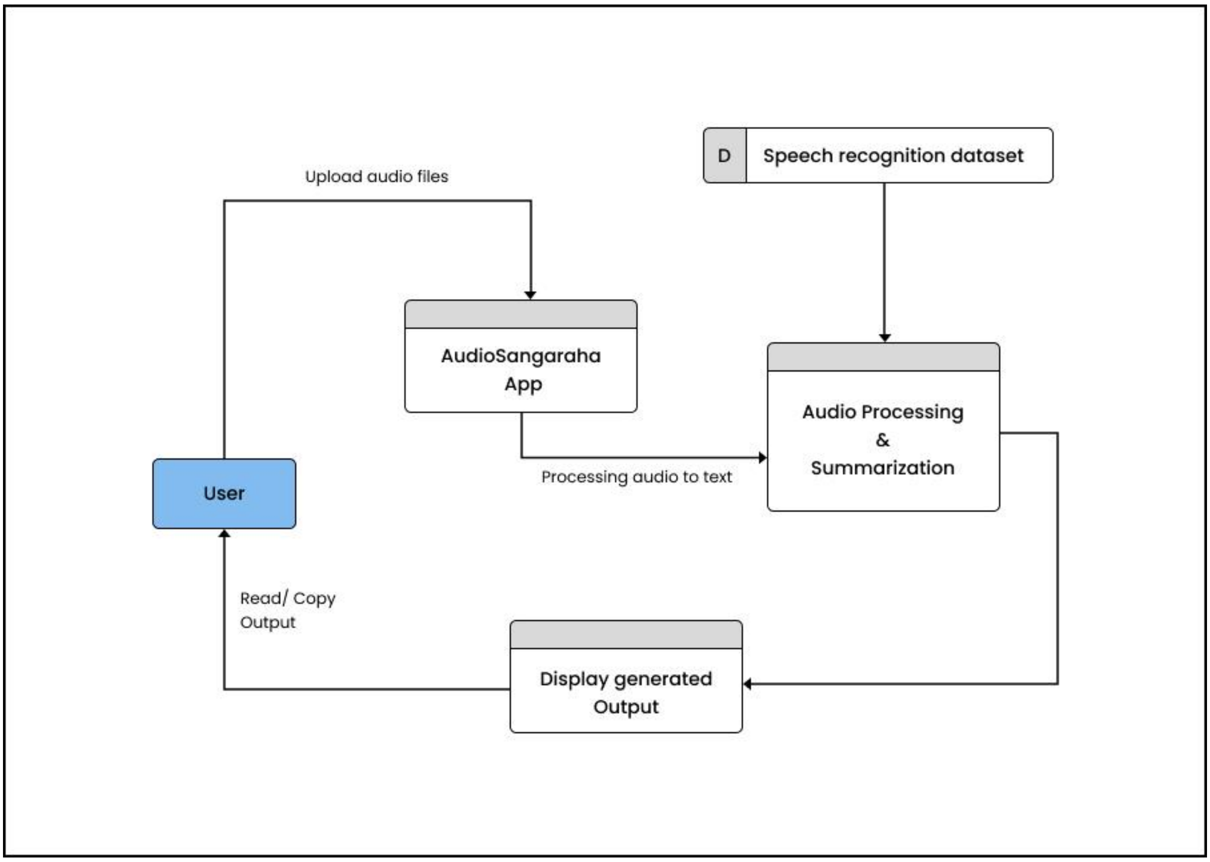


Figure 11: Data Flow Diagram (1)

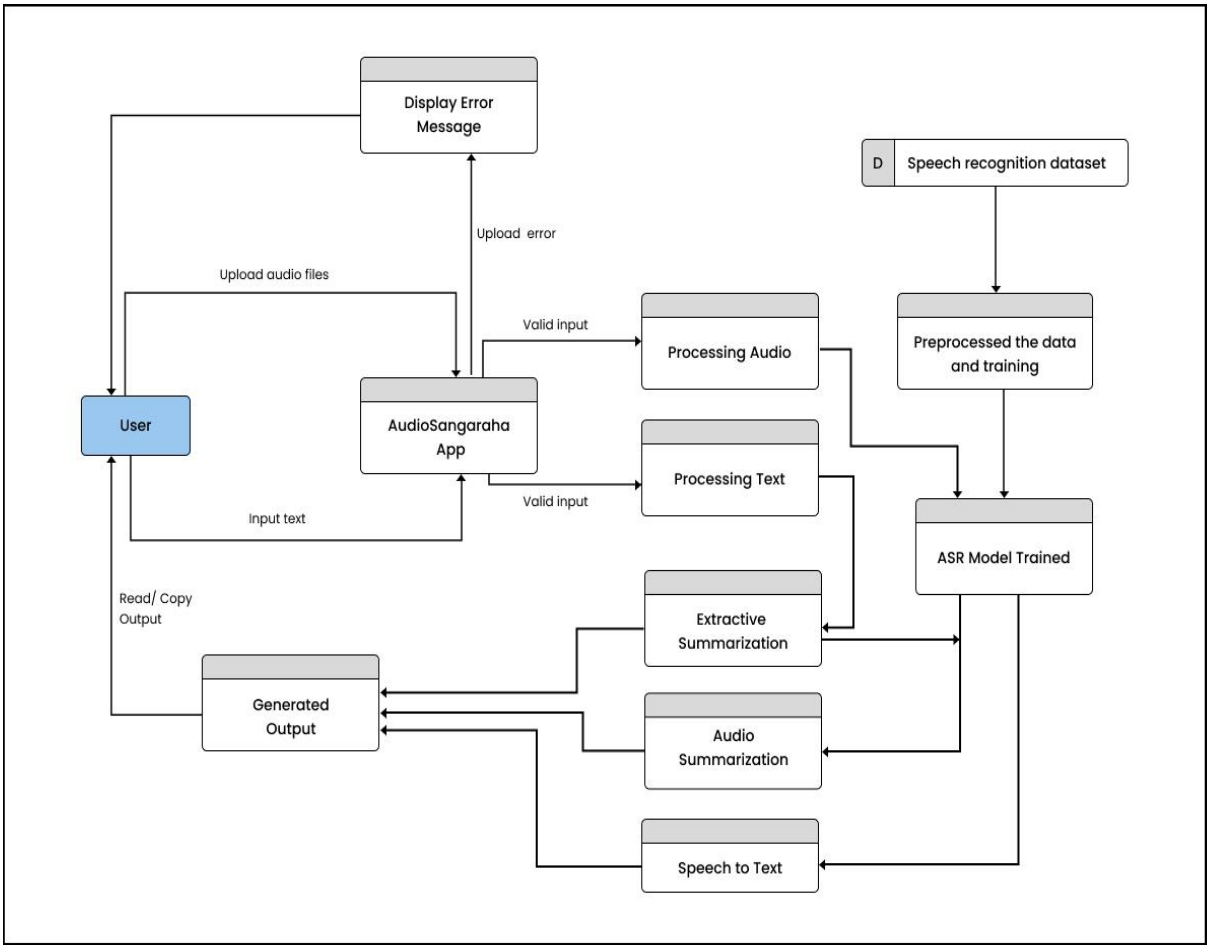


Figure 12: Data Flow Diagram (2)

6.5.2 System Process Flowchart

The following flow chart describes the key steps involved in the audio data summarization system.

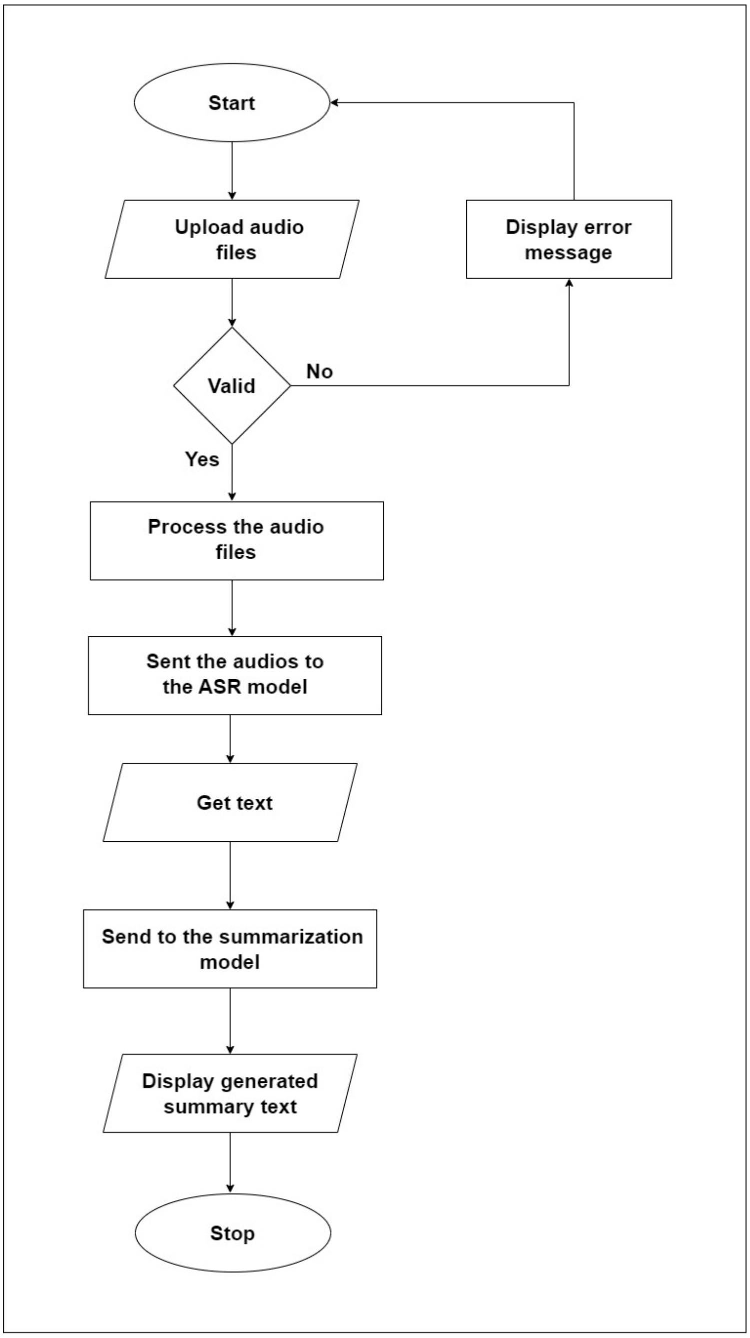


Figure 13: Flow chart

6.5.3 User Interface Design

The design of the UI is more important, as it helps the users to easily navigate through the system and understand the functionalities easily. So, for the proposed system the prototype is a web-based application. Also, the system should be responsive for mobile users. The following provides the Wireframe for the proposed audio summarization system. It has a simple and user-friendly interface. Other related UI low-fidelity and the high-fidelity designs of the system are attached in the **APPENDIX-E** and **APPENDIX-F**.

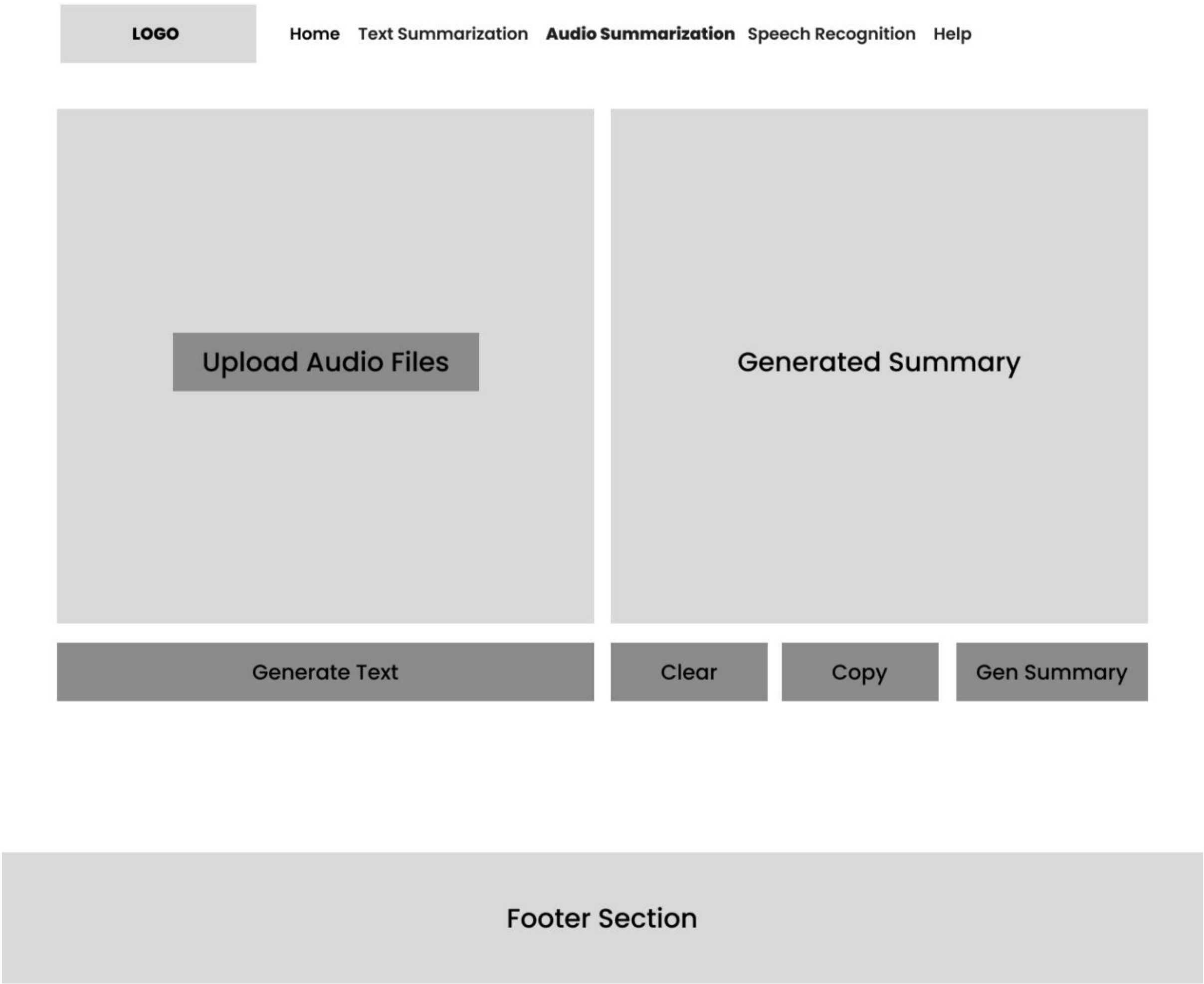


Figure 14: Wireframe of the UI

6.6 Chapter summary

In this chapter it discussed the design goals of this system. And for the design methodology author has selected the SSAD. Moreover, the high-level designs, data flow diagrams and the flowchart for the system are discussed. At the end of the chapter the UI for proposed system is also included.

CHAPTER 07: IMPLEMENTATION

7.1 Chapter Overview

In this chapter it is going to discuss the implementation of the system, like what technology is going to be used, the dataset for the model training, the frameworks, libraries and IDEs with a clear justification. At the end the implementation of the core functionality is discussed.

7.2 Technology Selection

7.2.1 Technology Stack

The following presents what are the technologies used to build the system, presenting in the presentation tier, logic tier and in the data tier.

Table 20: Technology Stack

Presentation Tier
Logic Tier
Data Tier

7.2.2 Dataset Selection

The dataset for this project is the main requirement. A dataset with high quality audio files which gives a clear speech, minimizing the background noises, and containing more data will give a high accuracy for this type of project. So, the first point of a dataset is needed for the audio to text process. The author was able to find an ASR dataset from Kaggle which contains more than 50000+ data. So, the author has chosen the dataset taken from Kaggle and created a subset on 5000 data. Also, the author used the Audacity software to record audios and created a custom dataset which contains 500 data and combine with that dataset. Here the dataset is taken to a .csv file which contains two columns namely the labeled sentences and the related file path. For the text summarization a dataset wasn't required as it uses only an algorithm.

7.2.3 Development Frameworks

For the development of audio summarization there are various frameworks available. The below frameworks are used for this project as this is a web application.

Table 21: Development Frameworks

Framework	**Justification**
Flask	For the backend deployment of this project Flask framework is used. For python it will be a great choice as it is a lightweight and flexible framework.
Bootstrap	Developing a responsive and visually appealing web application bootstrap will be a great framework. This will make the author build the application.

7.2.4 Programming Languages

Python is specifically suited for data science related projects. And the Python language is easy to learn, use, understand and it has the capability of handling multiple libraries and frameworks. For the existing systems like for text summarization and audio recognition, the researchers have used Python for the implementation. So, the author has chosen Python as the programming language.

7.2.5 Libraries

Table 22: Libraries Used

Library	**Justification**
Librosa	This will be used for audio analysis tasks
NLTK	NLTK is widely used library for NLP tasks, this provides for tokenization, stemming, tagging and text preprocessing tasks
Tensorflow	This library is used for tasks like audio processing and text summarization process
Pytorch	This will be used for summarization tasks
Transformers	This is used for training ASR, and it provides access for the pre-trained models
Pandas	Pandas is mainly used to manipulate data and analysis structured data
NumPy	This is used for working with the arrays

7.2.6 IDE

Table 23: IDE’s Used

Google Colab Pro	Google Colab Pro version performs well for the project

	related model training and testing. As in the Pro version it provides computer units and high ram for the training the model. And it allows to run Python codes and easily imports the libraries related to the project
VS Code	This is a valuable IDE for the project implementation, the frontend and the backend.

7.2.7 Summary of Technology Selection

Table 24: Summary of Technology Selection

Component	**Tools**
Programming Languages	Python
Frameworks	Flask, Bootstrap
Libraries	Pytorch, NLTK, Tensorflow, Librosa, Transformers, Pandas
IDE	Google Colab Pro, VS Code
Version Control	Github, Huggingface

7.3 Implementation of the Core Functionality

This system involves several key steps to implement the Sinhala audio summarization system. As the first step the user needs to upload audio files to the system. After it has been processed it should generate the text according to that. For the specific task the author has used the ASR dataset as mentioned above. The dataset was preprocessed and using a transfer learning approach the dataset was fine-tuned with a pre trained whisper AI model. After that that author create a model for the summarization purpose which is an extractive summarization approach. Using word frequency and sentence scoring algorithm it selects the most important sentence and generates the summary output. Once the audios are fed to the ASR model it generates the sentences combining as a paragraph. Then it is passed to the summarization model and generates the summary.

7.3.1 Audio Preprocessing

```
import pandas as pd
import re

# Define a function to remove punctuation marks
def remove_punctuation(text):
    return re.sub(r'|["?!,.]', '', text)

# Assuming data_df is your DataFrame

# Filter rows where the 'sentence' column does not contain English letters
filtered_data_df = data_df[~data_df['sentence'].str.contains('[a-zA-Z]')]

# Remove punctuation marks from the 'sentence' column
filtered_data_df['sentence'] = filtered_data_df['sentence'].apply(remove_punctuation)

# Remove rows with NaN values
filtered_data_df = filtered_data_df.dropna()

# Remove duplicate rows
filtered_data_df = filtered_data_df.drop_duplicates()

# Check the count of rows in filtered DataFrame
row_count = filtered_data_df.shape[0]
print("Number of rows in filtered DataFrame:", row_count)
```

Figure 15: Preprocessing the ASR Dataset

Before training the ASR model as the first part the dataset is cleaned. The punctuation marks are removed, the duplicated rows are removed, English words and sentences are removed, and also the null values are also removed from the dataset.

7.3.2 Spilt the Dataset to Train and Test

```
from datasets import Dataset, DatasetDict


# Calculate the indices to split the data (80% train, 10% validation, 10% test)
train_index = int(len(preprocessed_data) * 0.8)
val_index = int(len(preprocessed_data) * 0.9)

# Convert to a DatasetDict
dataset_dict = DatasetDict({
    'train': Dataset.from_pandas(preprocessed_data[:train_index]),  # First 80% of the data as train set
    'validation': Dataset.from_pandas(preprocessed_data[train_index:val_index]),  # Next 10% of the data as validation set
    'test': Dataset.from_pandas(preprocessed_data[val_index:]),  # Remaining 10% of the data as test set
})
```

Figure 16: Splitting the Dataset

Using the necessary libraries the audio data is split into training, testing and validation sets. The first 80% of data will be for training and 10% for testing and remaining 10% data will be split for validation sets. This helps to organize the sets of data for training, testing and validation.

```python
import os
import librosa
from datasets import DatasetDict, Dataset

def read_audio(audio_path):
    audio_path = os.path.join(flac_path, audio_path)
    array, sr = librosa.load(audio_path, sr=16000)
    return array, sr

# Map the read_audio function to the 'audio' column in the dataset
dataset_dict = dataset_dict.map(lambda x: {'audio': {'path': x['audio'], 'array': read_audio(x['audio'])[0], 'sampling_rate': read_audio(x['audio'])[1]}, 'sentence': x['sentence']})

# Create the DatasetDict
data_dict = DatasetDict({'train': dataset_dict['train'], 'validation': dataset_dict['validation'], 'test': dataset_dict['test']})
```

Figure 17: Creating Dataset Dictionary

Here the ‘librosa’ library helps to read audio files from the provided function. And it maps the function to each row in the dataset. For each row it adds an audio column which contains the audio path, array and the sampling rate. Below image states it creates a new dataset dictionary which includes updated dataset contain training, testing and validation.

```
DatasetDict({
    train: Dataset({
        features: ['sentence', 'audio'],
        num_rows: 3358
    })
    validation: Dataset({
        features: ['sentence', 'audio'],
        num_rows: 420
    })
    test: Dataset({
        features: ['sentence', 'audio'],
        num_rows: 420
    })
})
```

Figure 18: Dataset Dictionary

```python
from transformers import WhisperFeatureExtractor

feature_extractor = WhisperFeatureExtractor.from_pretrained("openai/whisper-small")
```

```python
from transformers import WhisperProcessor

processor = WhisperProcessor.from_pretrained("openai/whisper-small", language="Sinhala", task="transcribe")
```

Figure 19: Extracting the Whisper model

Here a pre-trained whisper model will be loaded using the ‘transformers’ library from the Hugginface. A tokenizer and a whisper processor are initialized to the whisper model.

7.3.3 Setup the Training arguments and Train the ASR Model

Then the training arguments will be set and passed to the Seq2Seq model trainer, it uses the Hugging Face transformers library. These include the training process, how it learns from the data, performance and save the checkpoints. And after the arguments are set, the model will be trained. And after the model has been trained it has been pushed to the Hugging Face

```
from transformers import Seq2SeqTrainingArguments

#training arguments definition
training_args = Seq2SeqTrainingArguments(
    output_dir="./Whisper-Sinhala_Audio_to_Text",
    per_device_train_batch_size=8,
    gradient_accumulation_steps=2,
    learning_rate=1e-5,
    warmup_steps=500,
    gradient_checkpointing=True,
    fp16=False,  # Set to False to disable mixed precision
    evaluation_strategy="steps",
    per_device_eval_batch_size=8,
    predict_with_generate=True,
    generation_max_length=225,
    save_steps=1000,
    eval_steps=1000,
    logging_steps=10,
    num_train_epochs=50,
    report_to=["tensorboard"],
    load_best_model_at_end=True,
    metric_for_best_model="wer",
    greater_is_better=False,
    push_to_hub=True,
)
```

Figure 20: Setting the Training Arguments

```
trainer.train()
```

Figure 21: Training the model

7.3.4 Text Summarization Model

Here the necessary libraries are imported. And then the imported stop word text file is used for stop-word removal.

```python
import nltk
nltk.download('punkt')

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
from langdetect import detect


#  extractive approach
a=[]
with open('stopWords.txt', 'r',encoding="utf-16") as f:
    a+=f.readlines()
f.close()
for i in range(0,len(a)):
    a[i]=a[i].rstrip('\n')
stopWords = a
```

Figure 22: Loading the Stop Words text file

```python
#genrate the frequency table
def _create_frequency_table(text_string) -> dict:
    words = word_tokenize(text_string)
    ps = PorterStemmer()

    freqTable = dict()
    for word in words:
        word = ps.stem(word)
        if word in stopWords:
            continue
        if word in freqTable:
            freqTable[word] += 1
        else:
            freqTable[word] = 1

    return freqTable


def _score_sentences(sentences, freqTable) -> dict:

    sentenceValue = dict()

    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        word_count_in_sentence_except_stop_words = 0
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                word_count_in_sentence_except_stop_words += 1
                if sentence in sentenceValue:
                    sentenceValue[sentence] += freqTable[wordValue]
                else:
                    sentenceValue[sentence] = freqTable[wordValue]

        if sentence in sentenceValue:
            sentenceValue[sentence] = sentenceValue[sentence] / word_count_in_sentence_except_stop_words
```

Figure 23: Scoring the Sentences

Then the author uses to generate frequency for the words, and then using word frequency it scores the sentences and find the average score of the sentence. And it will look out of the sentences with scores greater and produce as the summary.

```python
def _find_average_score(sentenceValue) -> int:

    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]

    average = (sumValues / len(sentenceValue))
    return average


def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ''

    for sentence in sentences:
        if sentence in sentenceValue and sentenceValue[sentence] >= (threshold):
            summary += " " + sentence
            sentence_count += 1

    return summary
```

Figure 24: Finding for the Average Score

7.4 User Interface

For the UI implementation the author has used HTML, CSS, JavaScript and the Bootstrap framework. Below presents the UI of Home page and Audio Summarization. Other pages in the system are placed in the **APPENDIX-F**.

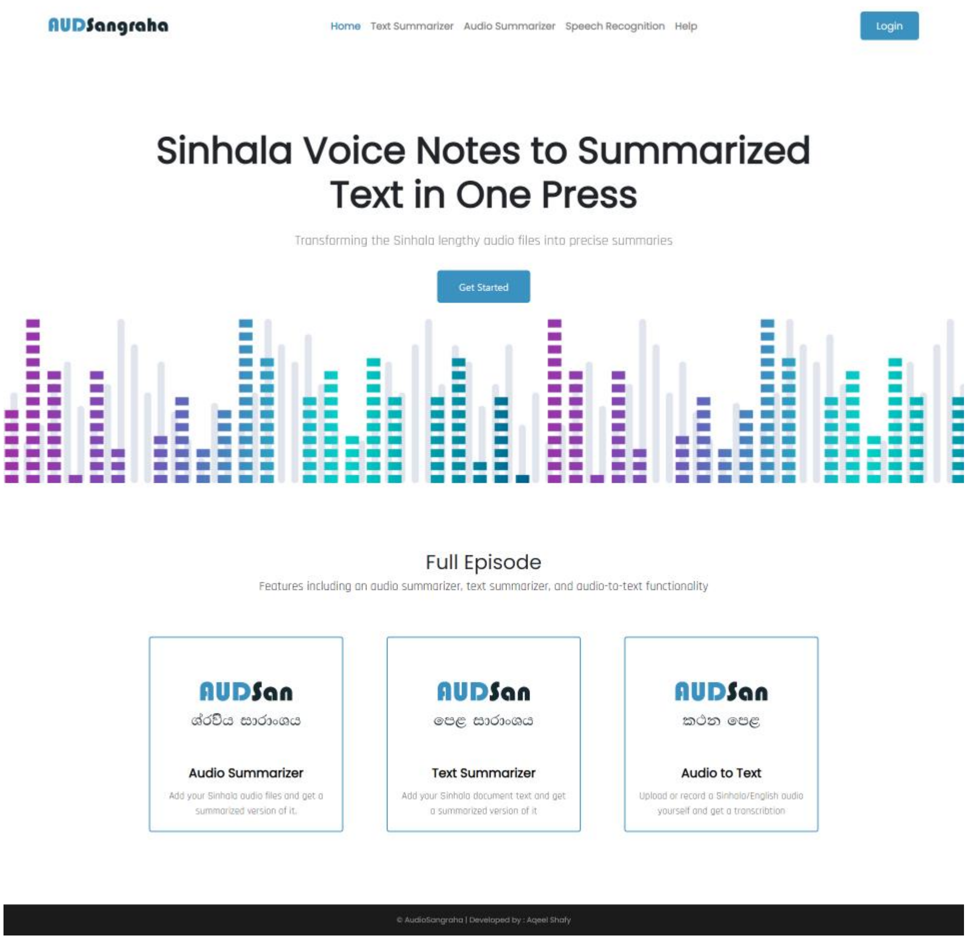


Figure 25: UI of the Home Page

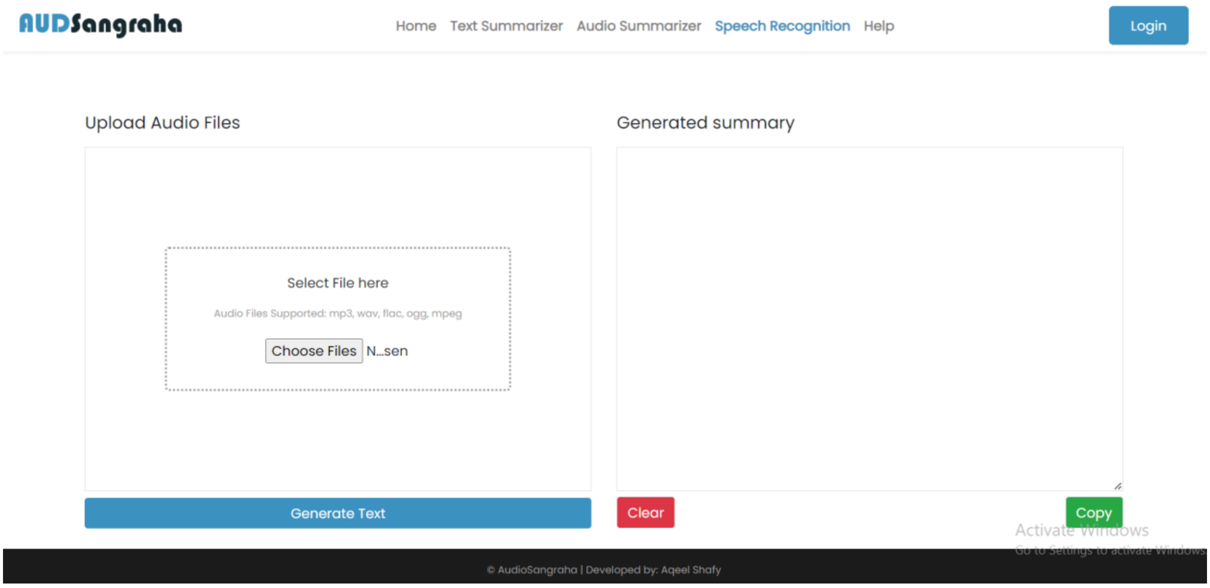


Figure 26: UI of the Audio Summarization Page

7.5 Chapter Summary

This chapter clearly explains what languages, technologies and tools are used for the implementation. Moreover, it has been discussed the implementation core functionality in detail with the necessary code snippets.

CHAPTER 8: TESTING

8.1 Chapter Overview

In this chapter it will discuss the testing methods used for the system. It will discuss the testing criteria, functional and non-functional testing, testing of the model, integration of the module and what was the limitation of testing faced by the author.

8.2 Objectives and Goals of Testing

The objective and the main purpose of testing is to verify that the developed system functionalities work as expected, without any errors. To achieve these priorities, the objectives of testing are stated above.

- To verify the implementation of the system works fine without any errors.
- Verify the ASR model in AudioSangaraha system operates as expected and has gone through the testing process.
- Also to verify the models produce the results as expected.
- To verify that the system has fulfill functional requirement which is the "Must have" and "Should have" in the MoSCoW technique.
- Also to ensure that the system fulfills the non-functional requirement.
- To state the potential area of improvements in the system

8.3 Testing Criteria

For the testing criteria the author uses to access the system in two methods. The two methods are stated above.

- Functional Testing – In this method it uses to test the functional requirements to ensure that all the features are determine well.
- Structural Testing – In this method it uses to test the non-functional requirements of the system. Also checks the system compliance with the performance of function requirements.

8.4 ASR Model Testing

For an ASR model there are several metrics to test the model as mentioned it literature review. Here the author will be using MER, WER and CER for the Whisper speech recognition model testing.

8.4.1 Match Error Rate (MER)

The testing metrics of MER for the model show 0.4, which means 0.6 of words are recognized correctly. This value is high as the author has trained the model with a limited dataset.

```
import jiwer

def evaluate_wer(dataset, model, processor):
    total_substitutions = 0
    total_deletions = 0
    total_insertions = 0
    total_words = 0

    for i in range(len(dataset)):
        # load the audio and target sentence
        audio, target_sentence, sampling_rate = dataset[i]['audio']['array'], dataset[i]['sentence'], dataset[i]["audio"]['sampling_rate']

        # convert audio to input features
        input_features = processor(audio, sampling_rate=sampling_rate, return_tensors="pt").input_features

        # generate token
        predicted_ids = model.generate(input_features)

        # decode token ids to text
        predicted_sentence = processor.batch_decode(predicted_ids, skip_special_tokens=True)

        # calculate substitution, deletion, and insertion errors
        measures = jiwer.compute_measures(target_sentence, predicted_sentence)

        total_substitutions += measures['substitutions']
        total_deletions += measures['deletions']
        total_insertions += measures['insertions']
        total_words += len(target_sentence.split())

    # calculate match rate error
    match_rate_error = 1 - (total_substitutions + total_deletions + total_insertions) / total_words
    return match_rate_error


match_rate_error = evaluate_wer(dataset, model, processor)
print("Match rate error:", match_rate_error)
```

```
Match rate error: 0.49917763157894735
```

Figure 27: MER Tesing

8.4.2 Character Error Rate (CER)

CER measures the percentage of incorrectly recognized characters. For the testing metrics of CER for the model shows 0.3, which means 0.7 of characters are recognized correctly.

```python
import jiwer

def evaluate_cer(dataset, model, processor):
    total_chars = 0
    total_errors = 0
    for i in range(len(dataset)):
        # load the audio and target sentence
        audio, target_sentence, sampling_rate = dataset[i]['audio']['array'], dataset[i]['sentence'], dataset[i]["audio"]['sampling_rate']

        # convert audio to input features
        input_features = processor(audio, sampling_rate=sampling_rate, return_tensors="pt").input_features

        # generate token
        predicted_ids = model.generate(input_features)

        # decode token ids to text
        predicted_sentence = processor.batch_decode(predicted_ids, skip_special_tokens=True)

        # calculate CER
        cer = jiwer.cer(target_sentence.lower(), predicted_sentence[0].lower())
        total_chars += len(target_sentence)
        total_errors += cer

    # calculate average CER
    cer = total_errors / total_chars
    cer_percentage = (total_errors / total_chars)*100
    print("CER Percent",cer_percentage)
    return cer

cer = evaluate_cer(dataset, model, processor)
print("CER:", cer)
```

```
CER Percent 0.32408714093726426
CER: 0.0032408714093726428
```

Figure 28: CER Testing

8.4.3 Word Error Rate (WER)

WER measures the percentage of incorrectly recognized words. For the testing data, testing metrics of WER for the model shows 0.7. Which means it shows a high rate of WER score. One reason for this is the trained dataset, it was a small amount of data. And the other reason is the Whisper model used for speech recognition only has the ability to recognize the first 30 seconds of audio. So, these were the reasons for getting a high WER score.

```python
# define a function for evaluating WER
import jiwer

def evaluate_wer(dataset, model, processor):
    hypothesis = []
    references = []
    for i in range(len(dataset)):
        # load the audio and target sentence
        audio, target_sentence, sampling_rate = dataset[i]['audio']['array'], dataset[i]['sentence'], dataset[i]["audio"]['sampling_rate']

        # convert audio to input features
        input_features = processor(audio, sampling_rate=sampling_rate, return_tensors="pt").input_features

        # generate token
        predicted_ids = model.generate(input_features)

        # decode token ids to text
        predicted_sentence = processor.batch_decode(predicted_ids, skip_special_tokens=True)

        # add to hypothesis and references lists
        hypothesis.append(str(predicted_sentence))
        references.append(str(target_sentence))

    # calculate
    wer = jiwer.wer(references, hypothesis)
    return wer


wer = evaluate_wer(dataset, model, processor)
print("WER:", wer)
```

```
WER: 0.7216282894736842
```

Figure 29: WER Testing

8.5 Functional Testing

The functionalities in the system which are mentioned in CHAPTER 04 are tested and stated below in the table.

Table 25: Functional Testing

FR ID	Use case	Expected result	Actual result	Status
FR1	Upload multiple audio files to the system	The system allows to upload multiple audio files	The system allows to upload multiple audio files	Pass
FR2	Upload other than audio files to the system	The system restricts or gives a prompt saying doesn't support other than audio file	The system gives a prompt saying not defined	Pass
FR3	Generate the text from the audio accordingly	The audio files output text should be combined as paragraph	The system produces the generated text from the audio files combining as a paragraph	Pass
FR4, FR5	Generated text should be able to copy and reset	Should be able to copy the text and reset	Able to copy the text and reset	Pass
FR6	Generated text should be able to summarize	System should be able to summarize the text	System summarizes the text	Pass
FR7, FR8	Generated summary should be able	Should be able to copy the generated	Able to copy the summary text and reset	Pass

	to copy and reset	summary and reset		

8.6 Module Integration Testing

Table 26: Module Integration Testing

Module	Input	Expected Result	Actual Result	Status
Input of audio files	Upload audio files	Upload multiple audio files in the audio input section	Can upload multiple audio files	Pass
Input of audio files	Verify the audio files	Popup a message saying if other than the audio files selected/ Or any other audio formats are added	Message popups saying not defined	Pass
Input Text	Enter a Sinhala paragraph/ Copy and paste a Sinhala paragraph	Paste Sinhala paragraph in the text area	Able to paste Sinhala paragraph in the text area	Pass
Input Text	Verify the sentence size	Popup a message saying add more than one sentence	Message popups asking for more sentences	Pass
Generate audio summary	Once the audio generated text displays, should be able form the summary	Click the summary button and should be to get the summary	Able to produce the summary accordingly	Pass

8.7 Non-Functional Testing

8.7.1 Performance Testing

The performance testing for the web application is crucial on producing the user experience. This system shows that with a minimum resource this web application can be run on a local environment. The performance on a CPU screenshot is placed below.

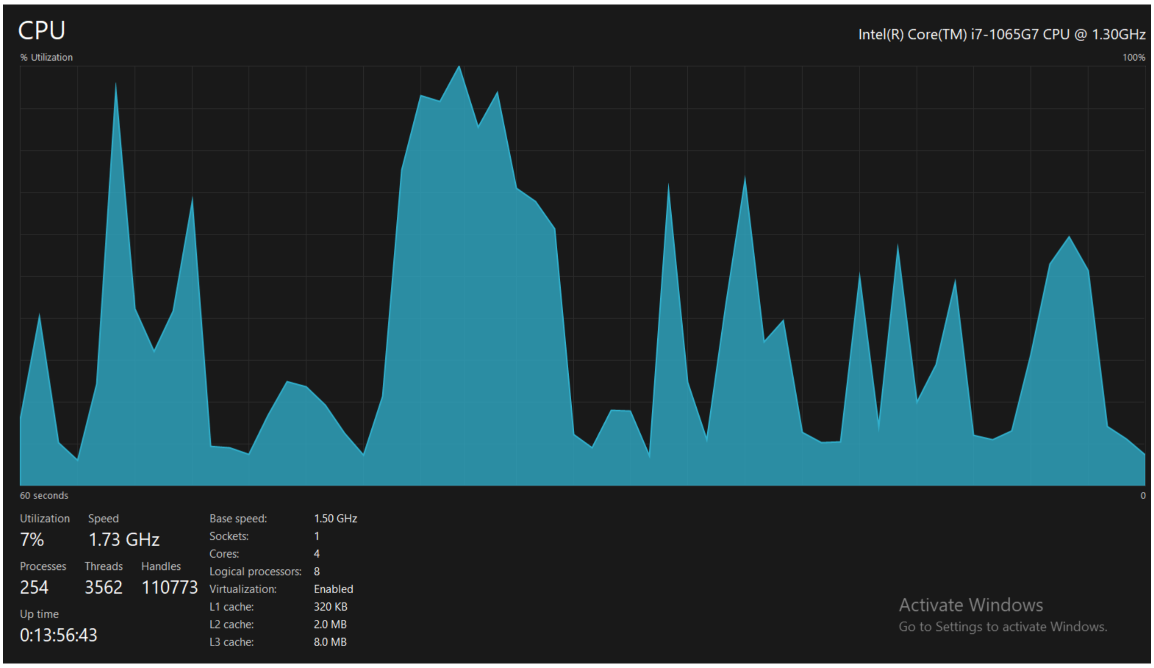


Figure 30: CPU Performance

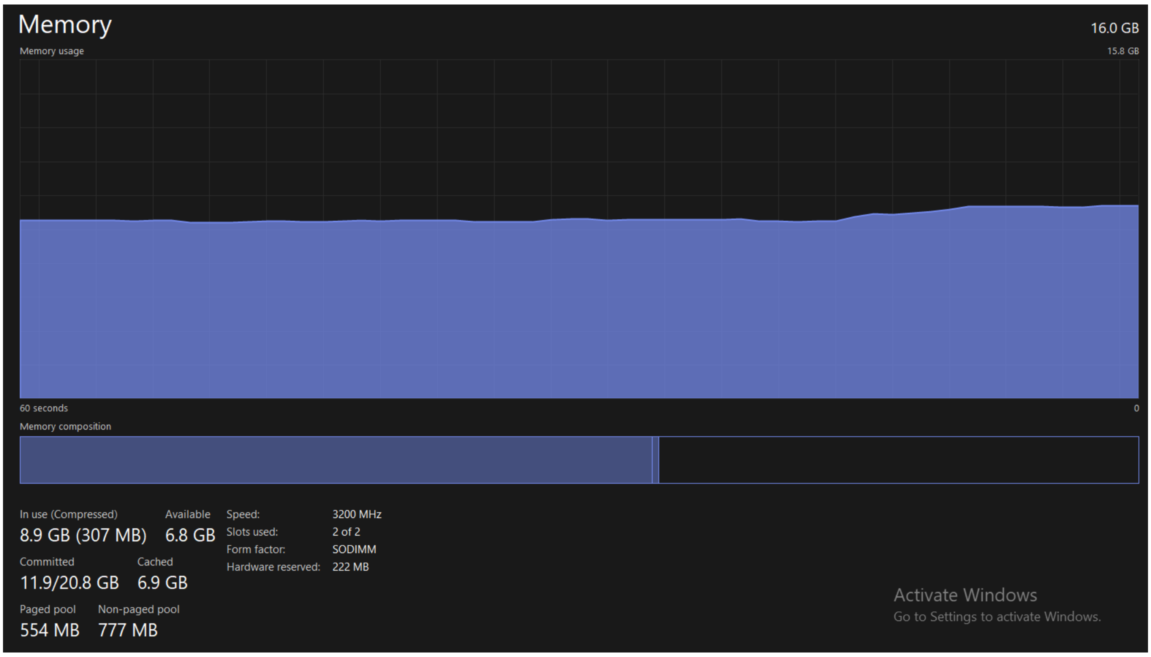


Figure 31: Memory Performance

8.7.2 Usability Testing

The author has thought of the usability and developed this web application. This has a simple UI which helps the user to navigate through the pages, and functionalities in the application. This was tested within the end users.

8.7.3 Security Testing

The security testing for a system involves in application to protect data. In this system it doesn't store or collect any of user information's and data, or any harmful contents. Also, it doesn't include any other third-party activities involved.

8.7.4 Maintainability Testing

The implementation code of the system is available in the author GitHub repository. To check the maintainable testing, the author used a tool to address the code quality (Codefactor.io). The below image shows that the system code has an A+ code quality.

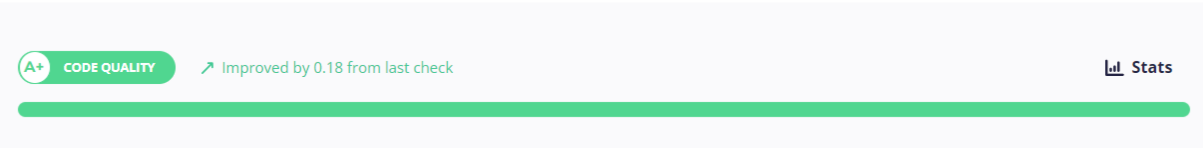


Figure 32: Code Quality

8.8 Limitations of the Testing Process

The author has faced various limitations during the testing process due to the limitation of resources in Sinhala language. One of the major limitations of the testing is that the Whisper model which is used for the speech recognition has a limitation of predicting the correct text within a time limit (It is able recognize the first 30 seconds of audio). Also, with the limitation of computational power and resources a limited data set was used to train the model. Because this the model WER was high and was difficult to get a lower WER. For the text summarizing a testing wasn't conducted as the author was unable and there wasn't a publicly available dataset for the extractive summarization.

8.9 Chapter Summary

First of all, in this chapter it has stated the goals of the objectives in testing and then it has been discussed about the testing metrics used to test the speech recognition model, the status of the functional and non-functional testing in the system. Also, at the end of the chapter it has been discussed what are the limitations of the testing the author faced.

CHAPTER 09: EVALUATION

9.1 Chapter Overview

This chapter will be discussing the evaluation of the proposed implemented system. Here it will be discussing the self-evaluation, evaluations from the domain and technical experts. Also, at the end it will be discussing the evaluation of the functional and non-functional requirements.

9.2 Evaluation Methodology and Approach

For the quality evaluation of the project the author has conducted both the qualitative and quantitative approaches. While in the previous chapters the author has stated the quantitative approach for the project. And in this chapter the author has conducted interviews with the experts to evaluate the qualitative approach for the project utilizing thematic analysis.

9.3 Evaluation Criteria

The below table states the thematic analysis on what are the criteria are followed to evaluate the qualitative approach.

Table 27: Evaluation Criteria

Criteria	**Evaluation**
The challenges in the domain and the gap	To find out the importance and challenges faced by the domain and figure out the research gap
Contribution to the research and problem	To figure out the research and technical contributions on the domain of speech recognition, text summarization and audio summarization. And how it has impacted on Sinhala language users.
The literature review on the research	To understand the problem domain and exiting works on the speech recognition and text summarization domain and what are the technologies, algorithms are used by the research and to figure out the research gap.

Design and implementation of the system	To evaluate the system by the model implementation, used algorithms, frameworks, and other design approaches
Quantitative evaluation on the research	To analyze the quantitative result on proposed system
System UI/UX	To figure out the user friendliness and user satisfaction throughout the functionalities in the system
Future Works	To find out the limitation within the system and how can be improved in the future

9.4 Self-Evaluation

As the above stated evaluation criteria, below in the table author has stated the self-evaluation on it.

Table 28: Self Evaluation

Criteria	**Self-evaluation**
Choice of the research	After going through the exiting works on the domain the author found a proper research gap on Audio summarization for the low resources Sinhala language. Research on speech recognition and text summarization has been widely conducted on high resource languages. But there are few studies conducted on Speech recognition and text summarization in the Sinhala language.
Contribution to the research	The author has used a transfer learning approach for a pretrained Whisper model for the Sinhala speech recognition and fine tune it accordingly. This will be a solid technical contribution to the research.
Implementation of the system	The author has used the necessary steps for the model implementation, and techniques for the development of the system

Evaluation of the model	In the previous chapter the author has used quantitative evaluation approach for the model and here it has used the qualitative evaluation approach
UI/UX of the system	The system has produced a user-friendly interface which is easy to understand to the user on the functionalities of the system
Limitations and the Future Work	As the ASR model is only able to generate text for the above of 30 seconds of audio, and it doesn't detect punctuation marks also, so in future it can be improved to handle lengthy audio files and make it able to detect punctuation marks too.

9.5 Selection of Evaluators

The author has gone through some interviews to conduct the evaluation of the system. In the below the table author has stated the count of the evaluators. The **APPENDIX-I**, the opinion of the evaluators are mentioned in.

Table 29: Count of Evaluators

Evaluators	**Count**
Domain experts	3
Technical experts	3
Normal users	5

9.6 Evaluation Result

9.6.1 Opinion of Domain Experts

Table 30: Domain Experts Opinion

Theme	**Opinion**

Choice of the research	They mentioned that this is a good research choice for addressing audio summarization for a low resource language like Sinhala. And this will be a benefit for the Sinhala language users.
Contribution to the research	They mentioned that the recently introduced Whisper AI is a good selection for the speech recognition model. So, using transfer learning approaches and fine tune will be a greater contribution for the research.
Implementation of the system	They mention that output result generated from the ASR model is okay with the trained dataset size. The ASR model can be improved by training by quality and a larger dataset.
UI/UX of the system	They mentioned that the proposed system is user-friendly as it is easy to operate

9.6.2 Opinion of Technical Experts

Table 31: Technical Experts Opinion

Theme	**Opinion**
Contribution to the research	Nowadays high resource languages like English use Whisper model for the speech related task has a high accuracy on it. So, using transfer learning for a low resource Sinhala creating a model will be a solid contribution.
Implementation of the system and Evaluation of the model	For the training it needs a large dataset to get accurate results. With the limitation of the computational power and resources it shows a decent performance with that small dataset. It would be grater if the model was trained with a quality and larger dataset. Also, with the limitation on the model taking 30 seconds of audio inputs the used approach was appreciated.

UI/UX of the system	They mentioned that the system UI is very clean and easy to understand for the user.
Limitations and the Future Work	It’s hard to compare the WER with the other speech recognition models as in this research it uses only a small dataset for training with limited resources. As the future works, they mentioned is training with larger and quality checked dataset will show less WER and good performance in speech recognition model. And they stated for summarization purposes in future abstractive approach or hybrid approached can be used, so it will generate as human summary.

9.6.2 Opinion of Focus Group

Table 32: Focus Group Opinion

Theme	**Opinion**
Implementation of the system	It was great to have a system for Sinhala language. Also, the system can be improved by uploading lengthy audios.
UI/UX of the system	The system functionalities are easy to understand and has very clean structure
Limitations and the Future Work	The spelling errors can be improved when it converts to text from the audio. Also, it can be improved by uploading lengthy audio files to generate summary.

9.7 Limitation of Evaluation

When it comes to low resource language like Sinhala it is hard compare the results with English languages, as there are limited resources for Sinhala language. And also, in this research it uses a small dataset for training it is hard to evaluate with other speech recognition systems.

9.8 Evaluation on Functional Requirements

Table 33: Evaluation of Functional Requirements

FR ID	Functional Requirement	Priority Level	Status
FR1	The system should be able upload multiple audio files to the system	M	Implemented
FR2	The system must not support other than audio file	M	Implemented
FR3	The system should generate the Sinhala text from the audio accordingly	M	Implemented
FR4	User should be able to copy the generated text	S	Implemented
FR5	User should be able to reset the generated text	S	Implemented
FR6	User should be able to summarize the generated text	M	Implemented
FR7	User should be able to copy the summary	S	Implemented
FR8	User should be able to reset the summary	S	Implemented
FR9	The user should be able to upload other language audios	W	Not implemented
FR10	The user should be able to upload videos/ files	W	Not implemented
FR11	The system generates summary of other languages	W	Not implemented

FR12	The system stores the input audio files or the generated result	C	Not implemented

9.9 Evaluation on Non-Functional Requirements

Table 34: Evaluation of Non-Functional Requirements

NFR ID	Requirements	Non-Functional Requirement	Priority Level	Status
NFR1	Performance	The system should be able to upload multiple audio inputs. And without taking much it should be generating the text accordingly	S	Implemented
NFR2	Usability	The system should be user-friendly, understand the system functionalities and should be easy to operate to the user	M	Implemented
NFR3	Security	The system should be protecting the user data while preventing unauthorized access	S	Implemented
NFR4	Maintainability	The system related code should follow coding standards and should be well structured for future use	S	Implemented
NFR5	Scalability	The system should run smoothly without crashing while the system is used by multiple users	C	Partially Implemented

NFR6	Quality	The ASR system should generate the user a quality output and when it summarized also it should produce a quality result	S	Implemented

9.10 Chapter Summary

This chapter has been discussed about the evaluation methodology and approaches used, and the evaluation criteria. Then the author itself had a self-evaluation on the prototype. And then author categorized the domain experts, technical experts and a focus group on evaluation of the system and what was their opinion are stated clearly. And at the end of the chapter, it has been discussed about evaluation of the functional and non-functional requirements.

CHAPTER 10: CONCLUSION

10.1 Chapter Overview

In this chapter, it will be discussing the achievement of aims, utilized throughout the course contents how has it been benefited to this project, what are exiting skills and throughout this project what are the gained skills. Also, it has discussed what the challenges faced during this project, the limitations of the project and what will be the future enhancement of the project.

10.2 Achievements of Research Aims and Objectives

The aim of the research is to design, develop and evaluate a summarization system for the low resource of Sinhala language audio data using natural language processing.

The author was able to successfully complete this project by designing, developing and evaluating the audio summarization system in the Sinhala language. Also, the author has built a model using pretrained whisper for the speech recognition task. And also, for the summarization the author has used the frequency based extractive summarization approach was used to generate the summary of the audio generated text.

10.3 Utilizing of Knowledge from the Course

The knowledge from the course gained is stated below in the table with the justification for how it helps to achieve to complete this project.

Table 35: Utilized Kowledge of the Course

Module	**Justification**
Software Development 1	This module was produced to understand the basic concepts of Python language. This helps the author when implementing the backend of the system and also helps while implementing the summarization model.

Web Design and Development and Advanced Client-Side Development	These modules help to understand the basic of UX principles. And the knowledge gain from this module is on HTML, CSS, JavaScript, and it helps when developing the frontend of this system.
Software Development Group Project	This module helps a lot on how to conduct research. Also, with the gain of this module it helped to complete the project within the time period, how to maintain the documentation and implementation, design and testing for the project.
Client-Server Architecture	This helps to gain the knowledge of connecting the frontend and backend on how the client and server is connected.
Applied Artificial Intelligence	From this module the author gains a knowledge of what are the concepts of training a model.
Usability Testing and Evaluation	This module gave an understanding of collecting responses from surveys and analyzing them and how the usability is measured.

10.4 Use of Existing Skills

Stated below existing knowledge skills helped the author on developing the project.

UI/UX Designing – The author had a good understanding on UI/UX design as the author was a UI/UX designer during his internship period. And the author has the knowledge of UI/UX principles and also, he gains knowledge through self-leaning too.

Frontend Development – From the start of the degree the author was interested on developing web pages with HTML, CSS and JavaScript. So, it helped the author to build the front end of the system.

Backend Development – The author has an understanding on Python Flask server as he worked on the previous SDGP module.

10.5 Use of New Skills

These were the new skills gained by the author on developing this project.

NLP – The author was new to NLP domain, so before starting the project the author has gone through some online YouTube and LinkedIn tutorials to get an understanding on Natural Language Processing. Also, during this project the author gain a lot of knowledge on NLP reading research papers.

Speech Recognition – During this project it helped the author to gain the knowledge and skills on speech recognition domain.

Text Summarization – Also throughout this project it helped the author to gain the knowledge and skills on text summarization domain.

10.6 Achievement of Learning Outcomes

Table 36: Achievements of Learning Outcomes

Description	**Learning Outcome**
After a clear research, the author has find out the necessary methods, techniques and tools to sort out the problem. And also used the proper testing metrics to test it.	LO1, LO4, LO5
The author has scheduled his work plan and accordingly to complete the project on time.	LO2
Author has gathered the area of improvements within the functional and non-functional requirements.	LO3

The author gathered data and development of the project has involved the SLEP rule.	LO6
Regularly the author has gotten feedback from the supervisor on his decisions.	LO7
The author has organized and well maintained the documentation.	LO8

10.7 Problems and Challenges Faced

Table 37: Problem and Challenges Faced

Problem and Challenge faced	**Description**
ASR dataset for Sinhala	The author was unable to find quality checked dataset for the speech recognition task. Also, there were two datasets publicly available in OpenSLR and Kaggle. But these datasets weren't quality checked, so the author created a subset from this dataset and created a custom dataset and combine together.
Limitation of computational power	For a better transcription output in ASR model, it needs a larger and quality checked dataset. Also, for training the model with a larger data set it needs a high range of computational power. So, the author had to use the Google Colab Pro version for training purposes. Also, after spending more than $40, the author was able to train the model successfully. But within that 5000 data it was unable to get a high accurate of output.

Audio Summarizer	As the author created an ASR model using Whisper, there was a limitation which generates the text only within 30 seconds of audios. And when it comes to low resource languages it is hard to predict the punctuation marks like full stops. And the author uses sentence scoring using the word frequency for the summarization purpose. So, when the audio is generated to text it is compulsory to have the full stop at the end of the sentence. So as a domain experts feedback author uses a method which the system takes multiple audios as input (But in an audio file only one sentence should be included). And fed to the ASR model one by one and combines as paragraph (full stops will be added at the end in an audio generated text). And then it generates as a summary.
Testing of the model	For lower WER an ASR model should be trained on a larger dataset. So as mentioned above with the limitations of the computational power, dataset quality and the size of dataset trained on model is hard to get a lower WER. And when it comes to extractive summarization in Sinhala it was unable to find a dataset for testing the model, and within the time period it was hard to create a dataset too.

10.8 Deviations

First of all, the author was planning to summarize lengthy Sinhala audio files. The Whisper model generates only 30 seconds of audio as mentioned above. But when it comes to lengthy audios it can be split into 30 seconds of chunks. But ASR models with low resource languages are hard to predict the punctuation marks (full stops). Also, the summary is done using sentence scoring using the word frequency, it is compulsory to have full stop at the end of a sentence. So, with the feedback of a domain expert author had to change the scope to handle multiple audio inputs (Which contains only a sentence in an audio). And then it will be adding full stops at the end of an audio generated text. And the generated audio will be combining a paragraph accordingly and will generate a summary output.

10.9 Limitation of the Research

While conducting this research, the author has to face various limitations.

- The dataset used for training the ASR model wasn't sufficient.
- The ASR model training needs a high amount of data, with the time period, computational power and with limited resources it was unable to get a quality result. Also, for getting compute power the author has spent a lot of money.
- As use of small dataset it was unable compare the WER result with the exiting systems.
- As there was no publicly available dataset for extractive summarization, the author was unable test the result.

10.10 Future Enhancement

Over the limitations the following Future works can be untaken by future research.

- The model can be trained on creating a larger Sinhala and quality checked dataset to get a better result avoiding a higher WER.
- The system can be improved by taking lengthy Sinhala audio inputs to produce summary.
- For summarization purpose abstractive or hybrid approaches can be used to get human summaries.
- The model can be improved to produce a high accuracy of Sinhala spelling.
- Also, this model can be improved for the videos as well.

10.11 Achievements of the Contribution to Body of Knowledge

The author was able to successfully complete the research gap stated on audio summarization for Sinhala language. Also, the author was able to build a model for speech recognition using pre trained whisper. This makes a solid contribution to this research to the ASR Sinhala domain. And using algorithmic approach for extractive summarization also have been implemented for the summarization purposes. Also, these models are pushed to Hugginface which can be used for future research purposes.

10.12 Concluding Remarks

The author was able to complete this project within the limitation of time and resources available. In this chapter it discusses the achievements of the research aim, objectives of the research, use of the existing skills and what new skills were gained. Also, what was the challenges faced while conducting this research, what are the limitations and what can be improved in the future are discussed clearly.

REFERENCES

A, V. and Jose, D. (2019). Speech to text conversion and summarization for effective understanding and documentation. *International Journal of Electrical and Computer Engineering (IJECE)*, 9, 3642. Available from https://doi.org/10.11591/ijece.v9i5.pp3642-3648.

Alharbi, Sadeen et al. (2021). Automatic Speech Recognition: Systematic Literature Review. *IEEE Access*, PP, 1–1. Available from https://doi.org/10.1109/ACCESS.2021.3112535.

Allahyari, M. et al. (2017). Text Summarization Techniques: A Brief Survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8, 397–405. Available from https://doi.org/10.14569/IJACSA.2017.081052.

Amodei, D. et al. (2016). Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin.

Babar, S., Tech-Cse, M., and Rit. (2013). Text Summarization:An Overview.

Besacier, L. et al. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56, 85–100. Available from https://doi.org/10.1016/j.specom.2013.07.008.

Braun, S. and Gamper, H. (2021). Effect of noise suppression losses on speech distortion and ASR performance. Available from https://doi.org/10.48550/arXiv.2111.11606 [Accessed 1 April 2024].

Das, P. and Prasad, V. (2015) VOICE RECOGNITION SYSTEM: SPEECH-TO-TEXT. Available at: https://www.researchgate.net/publication/304651244_VOICE_RECOGNITION_SYSTEM_SPEECH-TO-TEXT [Accessed: 01 September 2023].

Dhananjaya, V. et al. (2022) Bertifying Sinhala -- a comprehensive analysis of pre-trained language models for Sinhala Text Classification, arXiv.org. Available at: https://arxiv.org/abs/2208.07864 [Accessed: 12 September 2023].

de Silva, N. (2019) Survey on Publicly Available Sinhala NaturalLanguage Processing Tools and Research. Available at: https://www.researchgate.net/publication/333649787_Survey_on_Publicly_Available_Sinhala_Natural_Language_Processing_Tools_and_Research [Accessed: 05 September 2023].

Denil, M. et al. (2014). Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network. Available from https://doi.org/10.48550/arXiv.1406.3830 [Accessed 1 April 2024].

Deshpande, P. and Jahirabadkar, S. (2021). Study of Low Resource Language Document Extractive Summarization using Lexical chain and Bidirectional Encoder Representations from Transformers (BERT). *2021 International Conference on Computational Performance Evaluation (ComPE)*. December 2021. 457–461. Available from https://doi.org/10.1109/ComPE53109.2021.9751919 [Accessed 30 March 2024].

Digital 2022: Sri Lanka. (2022). *DataReportal – Global Digital Insights*. Available from https://datareportal.com/reports/digital-2022-sri-lanka [Accessed 28 March 2024].

Dinushika, T. et al. (2019). Speech Command Classification System for Sinhala Language based on Automatic Speech Recognition. *2019 International Conference on Asian Language Processing (IALP)*. November 2019. Shanghai, Singapore: IEEE, 205–210. Available from https://doi.org/10.1109/IALP48816.2019.9037648 [Accessed 30 March 2024].

Dong, Z. et al. (2023). A Speech Recognition Method Based on Domain-Specific Datasets and Confidence Decision Networks. *Sensors*, 23 (13), 6036. Available from https://doi.org/10.3390/s23136036.

Erkan, G. and Radev, D.R. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22, 457–479. Available from https://doi.org/10.1613/jair.1523.

Gamage, B. et al. (2020a). Usage of Combinational Acoustic Models (DNN-HMM and SGMM) and Identifying the Impact of Language Models in Sinhala Speech Recognition. *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. November 2020. 17–22. Available from https://doi.org/10.1109/ICTer51097.2020.9325439 [Accessed 29 March 2024].

Gamage, B. et al. (2020b). *Usage of Combinational Acoustic Models (DNN-HMM and SGMM) and Identifying the Impact of Language Models in Sinhala Speech Recognition*. Available from https://doi.org/10.1109/ICTer51097.2020.9325439.

Gamage, B. et al. (2021). Improve Sinhala Speech Recognition Through e2e LF-MMI Model. In: Bandyopadhyay, S. Devi, S.L. and Bhattacharyya, P. (eds.). *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*. December 2021. National Institute of Technology Silchar, Silchar, India: NLP Association of India (NLPAI), 213–219. Available from https://aclanthology.org/2021.icon-main.26 [Accessed 29 March 2024].

Glackin, C. et al. (2019). *Smart Transcription*. Available from https://doi.org/10.1145/3335082.3335114.

González, S.S. (2022). Whisper's OpenAI: The AI whisperer model. *Narrativa*. Available from https://www.narrativa.com/whispers-openai-the-ai-whisperer-model/ [Accessed 8 April 2024].

Graves, A., Mohamed, A. and Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. Available from https://doi.org/10.48550/arXiv.1303.5778 [Accessed 1 April 2024].

Gruetzemacher, R. (2022) The power of Natural Language Processing, Harvard Business Review. Available at: https://hbr.org/2022/04/the-power-of-natural-language-processing [Accessed: 31 August 2023].

Hinton, G. et al. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29 (6), 82–97. Available from https://doi.org/10.1109/MSP.2012.2205597.

Introducing Whisper. (no date). Available from https://openai.com/research/whisper [Accessed 7 April 2024].

Jain, R. et al. (2023). *Adaptation of Whisper models to child speech recognition*.

Jendoubi, S., Yaghlane, B.B. and Martin, A. (2013). Belief Hidden Markov Model for speech recognition. *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*. April 2013. 1–6. Available from https://doi.org/10.1109/ICMSAO.2013.6552563 [Accessed 1 April 2024].

Jing, B. et al. (2021). Multiplex Graph Neural Network for Extractive Text Summarization. In: Moens, M.-F. Huang, X. Specia, L. et al. (eds.). *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. November 2021. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 133–139. Available from https://doi.org/10.18653/v1/2021.emnlp-main.11 [Accessed 31 March 2024].

Karunathilaka, H. et al. (2020). Low-resource Sinhala Speech Recognition using Deep Learning. *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. 4 November 2020. Colombo, Sri Lanka: IEEE, 196–201. Available from https://doi.org/10.1109/ICTer51097.2020.9325468 [Accessed 29 March 2024].

Kasthuri Arachchige, T. and Weerasinghe, R. (2023) Tacosi: A Sinhala text to speech system with Neural Networks | IEEE ..., TacoSi: A Sinhala Text to Speech System with Neural Networks. Available at: https://ieeexplore.ieee.org/abstract/document/10145749 [Accessed: 05 September 2023].

Kenny, P. (2006). Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms.

Languages of Sri Lanka. (2023). *Wikipedia*. Available from https://en.wikipedia.org/w/index.php?title=Languages_of_Sri_Lanka&oldid=1192799326 [Accessed 28 March 2024].

Lewis, Mike, et al. "BART: Denoising Sequence-To-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, https://doi.org/10.18653/v1/2020.acl-main.703.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*. July 2004. Barcelona, Spain: Association for Computational Linguistics, 74–81. Available from https://aclanthology.org/W04-1013 [Accessed 1 April 2024].

Liu, Y. and Lapata, M. (2019). Text Summarization with Pretrained Encoders. Available from https://doi.org/10.48550/arXiv.1908.08345 [Accessed 1 April 2024].

Madhuri, J.N. and Ganesh Kumar, R. (2019). Extractive Text Summarization Using Sentence Ranking. *2019 International Conference on Data Science and Communication (IconDSC)*. March 2019. 1–3. Available from https://doi.org/10.1109/IconDSC.2019.8817040 [Accessed 11 February 2024].

Manamperi, W. et al. (2018). Sinhala Speech Recognition for Interactive Voice Response Systems Accessed Through Mobile Phones. *2018 Moratuwa Engineering Research Conference (MERCon)*. May 2018. Moratuwa: IEEE, 241–246. Available from https://doi.org/10.1109/MERCon.2018.8421888 [Accessed 30 March 2024].

Markovnikov, N. et al. (2018). *Deep Neural Networks in Russian Speech Recognition*. Available from https://doi.org/10.1007/978-3-319-71746-3_5.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing Order into Text. In: Lin, D. and Wu, D. (eds.). *Proceedings of the 2004 Conference on Empirical Methods in Natural*

Language Processing. July 2004. Barcelona, Spain: Association for Computational Linguistics, 404–411. Available from https://aclanthology.org/W04-3252 [Accessed 1 April 2024].

Millstein, F. (2020) Natural language processing with python: natural language processing using NLTK, https://scholar.google.com/. Available at: https://books.google.lk/books?hl=en&lr=&id=vXzvDwAAQBAJ&oi=fnd&pg=PA4&dq=Frank+Millstein.+(2020).+Natural+Language+Processing+Using+NLTK.+Frank+Millstein.&ots=02SOrlVUaE&sig=i1bsvq75ZnWN9HC2lhcD0F3dIc&redir_esc=y#v=onepage&q=Frank%20Millstein.%20(2020).%20Natural%20Language%20Processing%20Using%20NLTK.%20Frank%20Millstein.&f=false [Accessed: 31 August 2023].

Mohd, M., Jan, R. and Shah, M. (2020). Text document summarization using word embedding. *Expert Systems with Applications*, 143, 112958. Available from https://doi.org/10.1016/j.eswa.2019.112958.

Nadungodage, T. et al. (2018). (1) (PDF) Sinhala G2P Conversion for Speech Processing. Available from https://www.researchgate.net/publication/328072699_Sinhala_G2P_Conversion_for_Speech_Processing [Accessed 29 March 2024].

Nasib, A. et al. (2018). *A Real Time Speech to Text Conversion Technique for Bengali Language*. Available from https://doi.org/10.1109/IC4ME2.2018.8465680.

Panayotov, V. et al. (2015). Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. April 2015. 5206–5210. Available from https://doi.org/10.1109/ICASSP.2015.7178964 [Accessed 31 March 2024].

Phair, David, and Kerryn Warren. “Saunders’ Research Onion: Explained Simply.” Grad Coach, Jan. 2021, gradcoach.com/saunders-research-onion/. Accessed 28 Sept. 2023.

Prudhvi, K. et al. (2020) Text summarization using Natural Language Processing, SpringerLink. Available at: https://link.springer.com/chapter/10.1007/978-981-15- 5400-1_54 [Accessed: 04 September 2023].

Pratama, R. and Amrullah, A. (2023). ANALYSIS OF WHISPER AUTOMATIC SPEECH RECOGNITION PERFORMANCE ON LOW RESOURCE LANGUAGE. *Jurnal Pilar Nusa Mandiri*, 20, 1–8. Available from https://doi.org/10.33480/pilar.v20i1.4633.

Rathnayake, B.R.M.S.R.B., Manathunga, K. and Kasthurirathna, D. (2023a). 'Talking Books' : A Sinhala Abstractive Text Summarization Approach for Sinhala Textbooks. *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. April 2023. 1–6. Available from https://doi.org/10.1109/I2CT57861.2023.10126205 [Accessed 30 March 2024].

Rathnayake, B.R.M.S.R.B., Manathunga, K. and Kasthurirathna, D. (2023b). 'Talking Books' : A Sinhala Abstractive Text Summarization Approach for Sinhala Textbooks. *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. 7 April 2023. Lonavla, India: IEEE, 1–6. Available from https://doi.org/10.1109/I2CT57861.2023.10126205 [Accessed 31 March 2024].

Shah, M., Jan , R. and Mohd, M. (2019) Text document summarization using word embedding, Expert Systems with Applications. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0957417419306761?via%3Dihub [Accessed: 14 January 2024].

Sharma, G. and Sharma, D. (2022). Automatic Text Summarization Methods: A Comprehensive Review. *SN Computer Science*, 4 (1), 33. Available from https://doi.org/10.1007/s42979-022-01446-w.

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306. Available from https://doi.org/10.1016/j.physd.2019.132306.

SmartAction. (2021). Does Word Error Rate Matter? *SmartAction*. Available from https://smartaction.ai/blog/does-word-error-rate-matter/ [Accessed 1 April 2024].

Singh, A. (2020) Text summarization using NLP, Medium. Available at: https://medium.com/analytics-vidhya/text-summarization-using-nlp-3e85ad0c6349 [Accessed: 04 September 2023].

S. Yu, Philip , et al. "Understanding Pre-Trained BERT for Aspect-Based Sentiment Analysis." Aclanthology, Dec. 2020, aclanthology.org/2020.coling-main.21.pdf.

Upadhyaya, P. et al. (2019). *Continuous Hindi Speech Recognition Using Kaldi ASR based on Deep Neural Network*. Available from https://doi.org/10.13140/RG.2.2.16897.97126.

Violeta, L. and Toda, T. (2023). *An Analysis of Personalized Speech Recognition System Development for the Deaf and Hard-of-Hearing*.

Wang, D., Wang, X. and Lv, S. (2019). An Overview of End-to-End Automatic Speech Recognition. *Symmetry*, 11 (8), 1018. Available from https://doi.org/10.3390/sym11081018.

Warnasooriya, W.M. et al. (2020). *SINHALA SPEECH RECOGNITION SYSTEM FOR JOURNALISTS IN SRILANKA*. Available from https://www.researchgate.net/publication/346624775_SINHALA_SPEECH_RECOGNITION_SYSTEM_FOR_JOURNALISTS_IN_SRILANKA [Accessed: 11 September 2023].

Weerasinghe, R. et al. (2020) Low-resource sinhala speech recognition using Deep Learning | IEEE ... Available at: https://ieeexplore.ieee.org/document/9325468 [Accessed: 12 September 2023].

Yu, D. and Deng, L. (2015). *Automatic Speech Recognition: A Deep Learning Approach*. London: Springer London. Available from https://doi.org/10.1007/978-1-4471-5779-3 [Accessed 1 April 2024].

Zaware, S. et al. (2021). Text Summarization using TF-IDF and Textrank algorithm. *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. June 2021. 1399–1407. Available from https://doi.org/10.1109/ICOEI51242.2021.9453071 [Accessed 30 March 2024].

APPENDIX-A: Concept Map

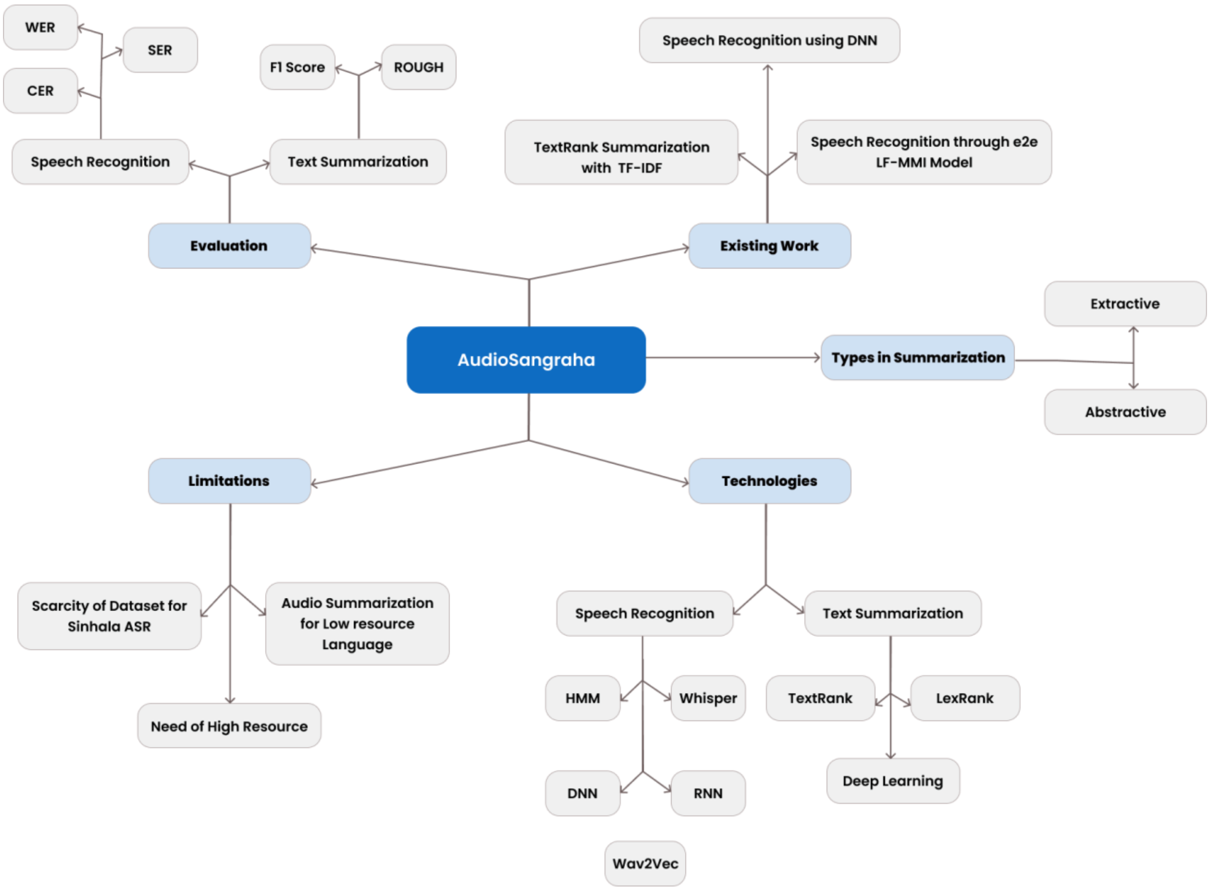


Figure 33: Concept Map

APPENDIX-B: In Scope and Out Scope of the Project

In Scope

1. Implementing a web-based application that takes multiple audio files as an input and summarizes and provides the output.
2. For the summarization purposes it will use extractive summarization approach.
3. Collecting a high quality of dataset for the ASR approach.
4. Evaluate the system through the domain and technical experts.

Out Scope

1. Speech recognition through video will not be available in the system.
2. Uploading images/documents (converts the text by image processing) will not be available.

Diagram Depicting the Prototype Feature

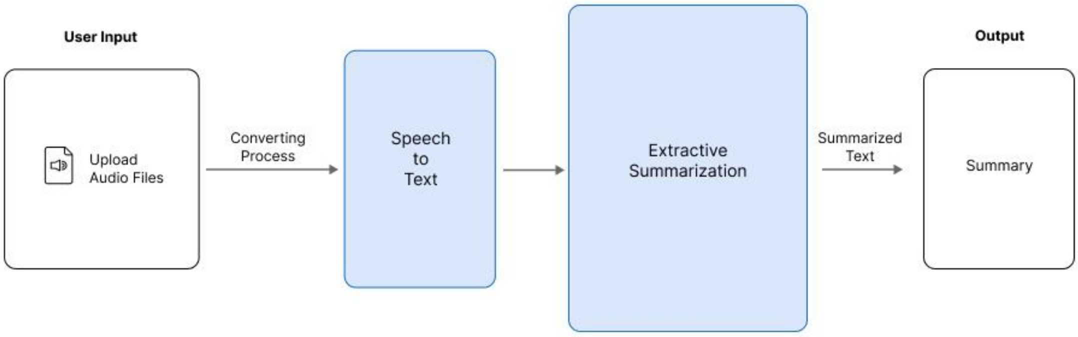


Figure 34: Diagram Depicting the Prototype

APPENDIX-C: GANTT CHART

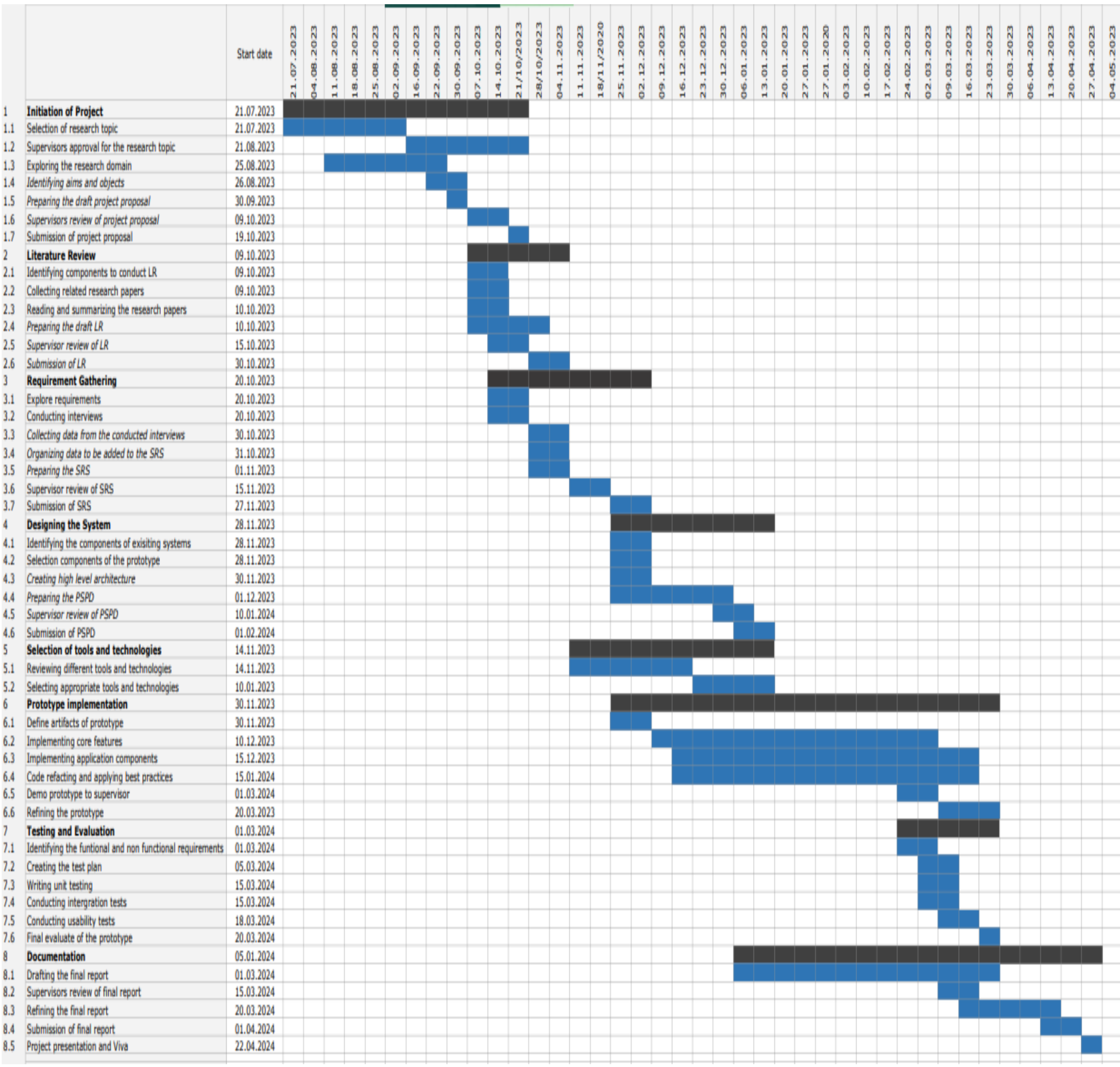
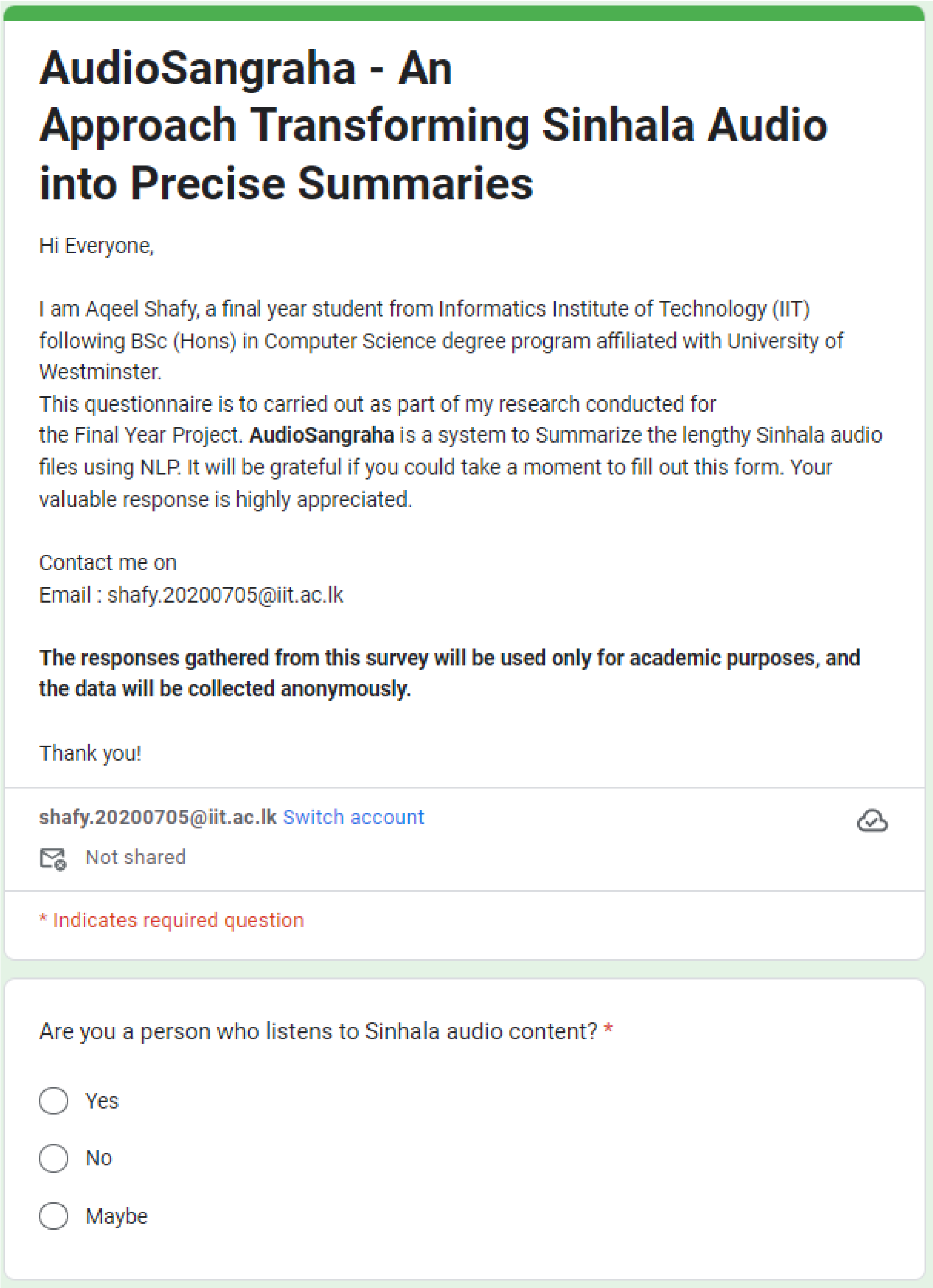


Figure 35: Ghantt Chart

APPENDIX-D: SURVEY QUESTIONS



AudioSangraha - An Approach Transforming Sinhala Audio into Precise Summaries

Hi Everyone,

I am Aqeel Shafy, a final year student from Informatics Institute of Technology (IIT) following BSc (Hons) in Computer Science degree program affiliated with University of Westminster.
This questionnaire is to carried out as part of my research conducted for the Final Year Project. **AudioSangraha** is a system to Summarize the lengthy Sinhala audio files using NLP. It will be grateful if you could take a moment to fill out this form. Your valuable response is highly appreciated.

Contact me on
Email : shafy.20200705@iit.ac.lk

The responses gathered from this survey will be used only for academic purposes, and the data will be collected anonymously.

Thank you!

shafy.20200705@iit.ac.lk Switch account

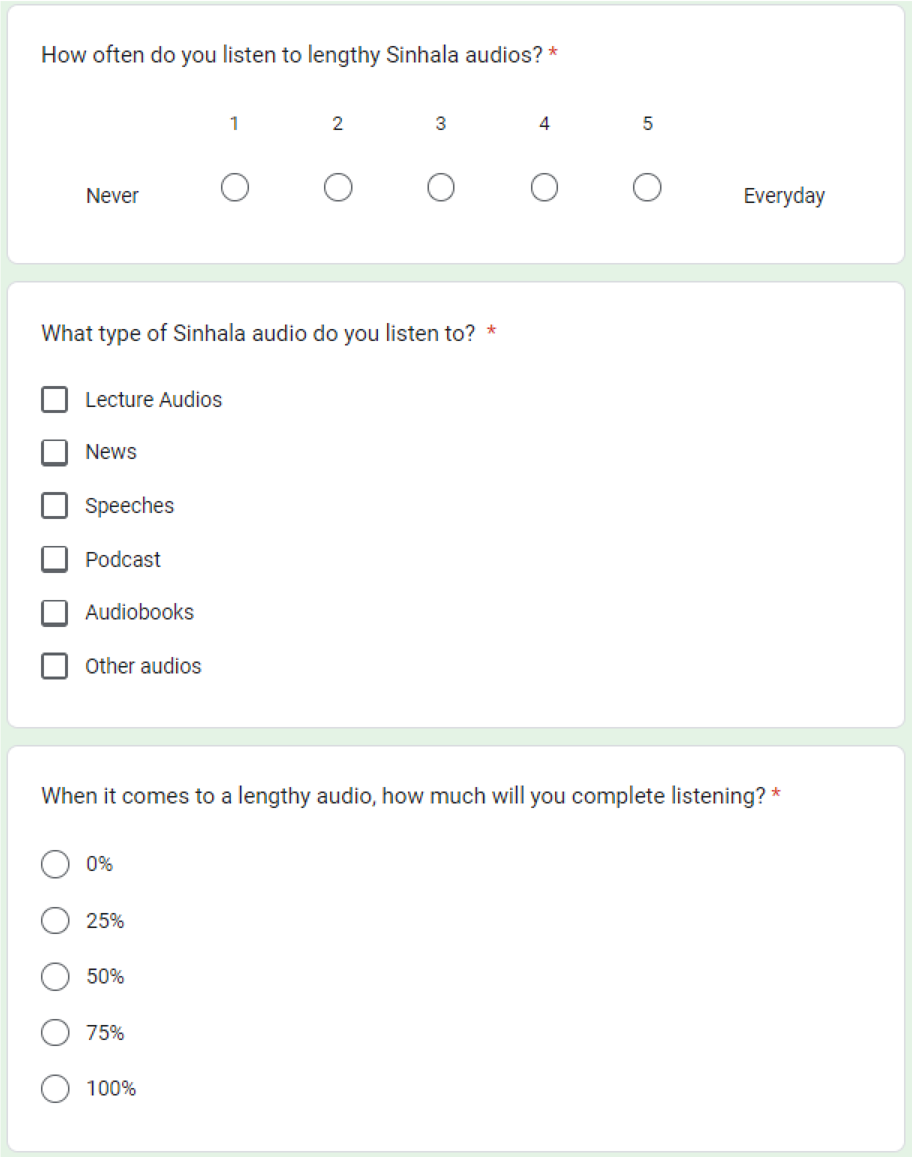
Not shared

* Indicates required question

Are you a person who listens to Sinhala audio content? *

- Yes
- No
- Maybe

Figure 36: Survey Questions (1)



How often do you listen to lengthy Sinhala audios? *

	1	2	3	4	5	
Never	○	○	○	○	○	Everyday

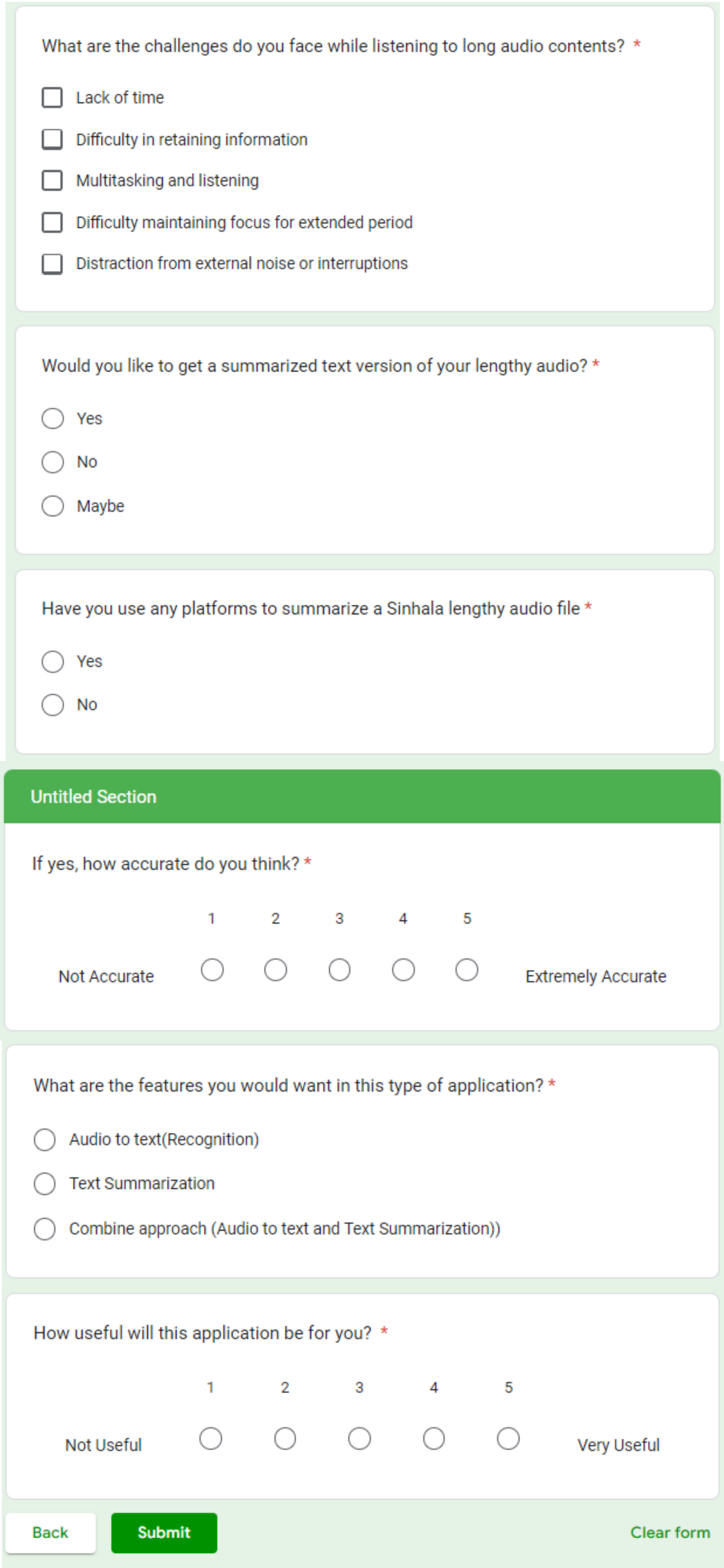
What type of Sinhala audio do you listen to? *

- ☐ Lecture Audios
- ☐ News
- ☐ Speeches
- ☐ Podcast
- ☐ Audiobooks
- ☐ Other audios

When it comes to a lengthy audio, how much will you complete listening? *

- ○ 0%
- ○ 25%
- ○ 50%
- ○ 75%
- ○ 100%

Figure 37: Survey Questions (2)


What are the challenges do you face while listening to long audio contents? *

- [ ] Lack of time
- [ ] Difficulty in retaining information
- [ ] Multitasking and listening
- [ ] Difficulty maintaining focus for extended period
- [ ] Distraction from external noise or interruptions

Would you like to get a summarized text version of your lengthy audio? *

- ○ Yes
- ○ No
- ○ Maybe

Have you use any platforms to summarize a Sinhala lengthy audio file *

- ○ Yes
- ○ No

Untitled Section

If yes, how accurate do you think? *

	1	2	3	4	5	
Not Accurate	○	○	○	○	○	Extremely Accurate

What are the features you would want in this type of application? *

- ○ Audio to text(Recognition)
- ○ Text Summarization
- ○ Combine approach (Audio to text and Text Summarization))

How useful will this application be for you? *

	1	2	3	4	5	
Not Useful	○	○	○	○	○	Very Useful

Back Submit Clear form

Figure 38: Survey Questions (3)

APPENDIX-E: Low-Fidelity Design

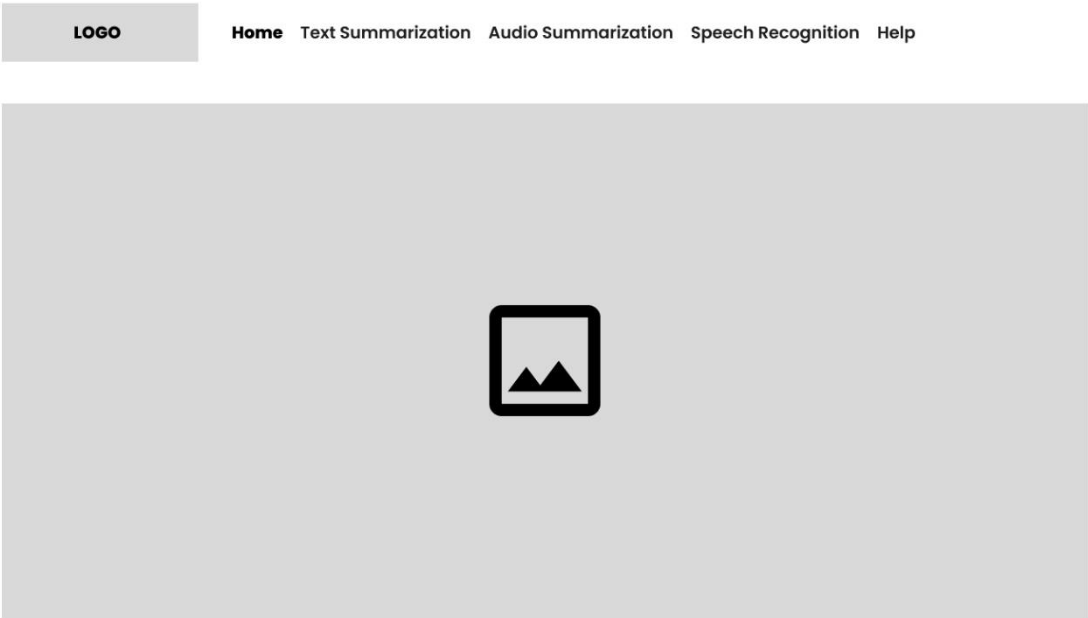


Figure 39: Wireframe of the UI (2)

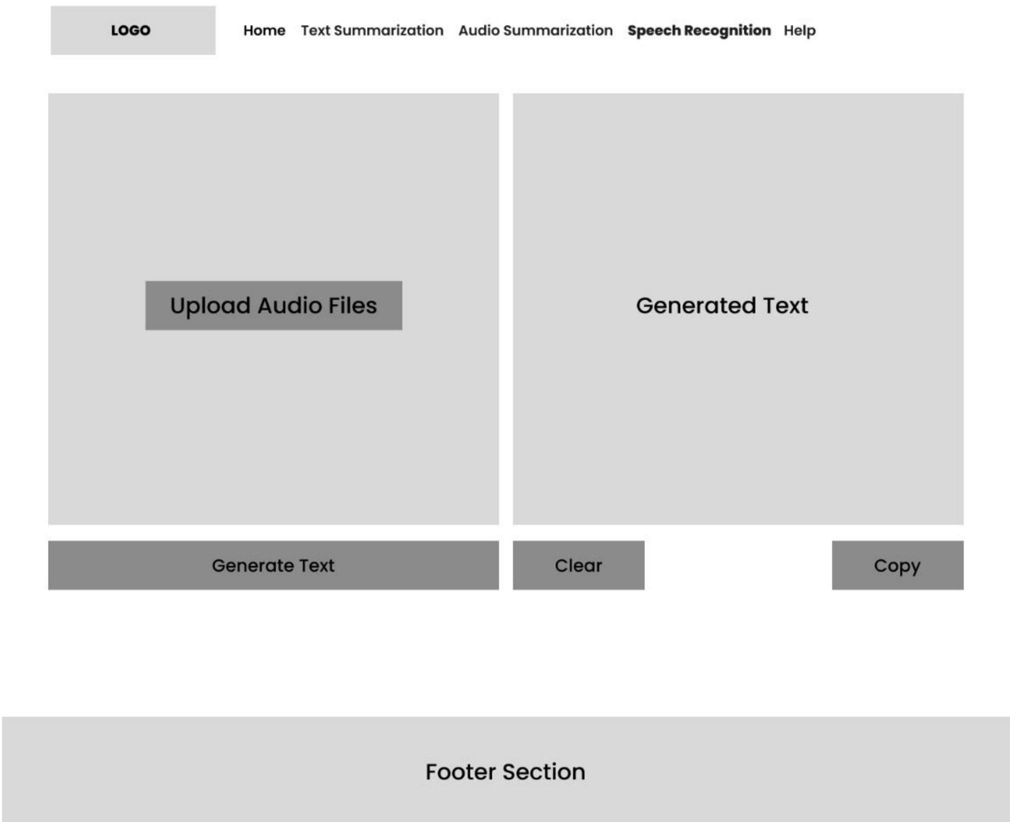


Figure 40: Wireframe of the UI (3)

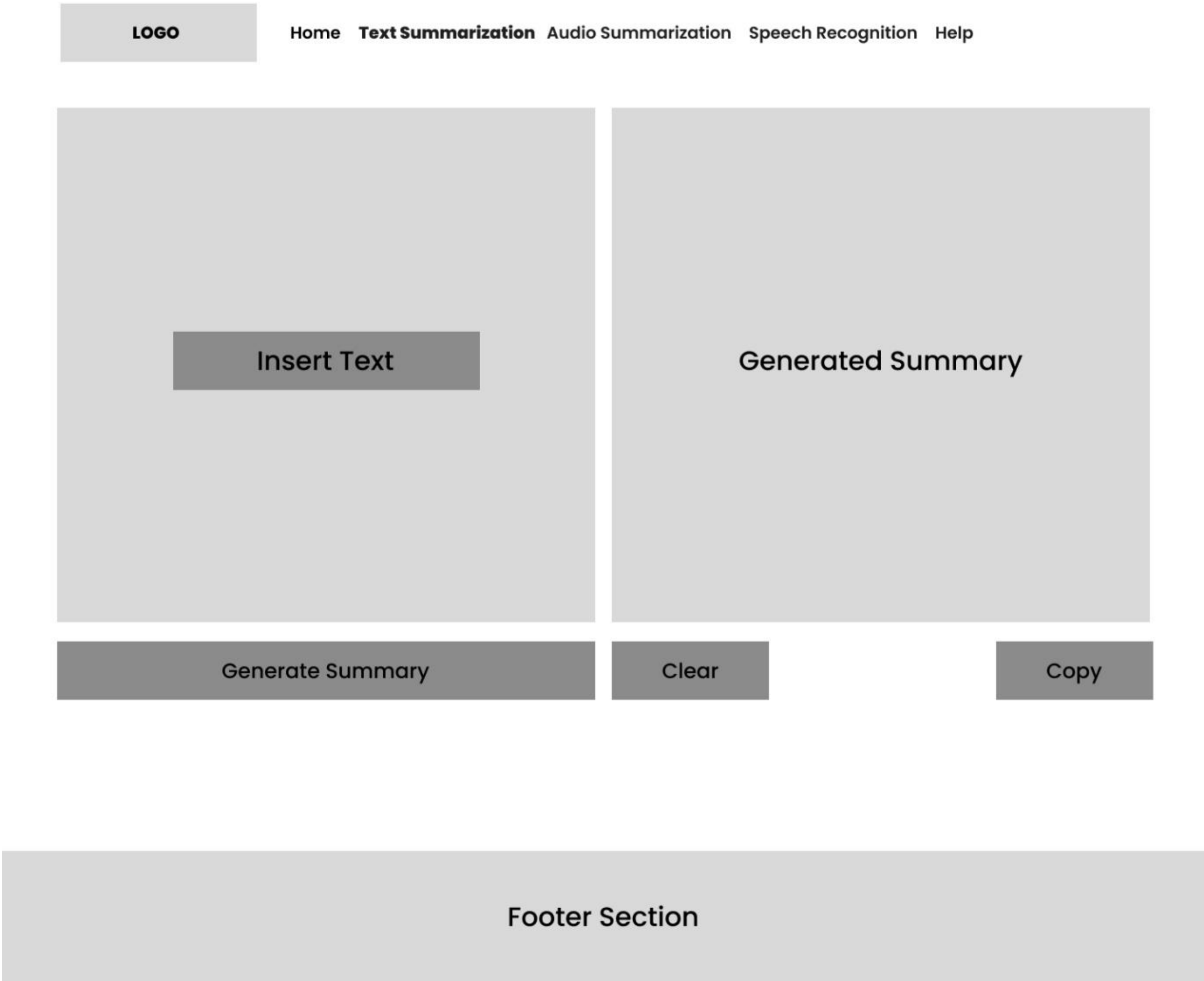


Figure 41: Wireframe of the UI (4)

APPENDIX-F: High-Fidelity Design

Figure 42: High-Fidelity of the UI (1)

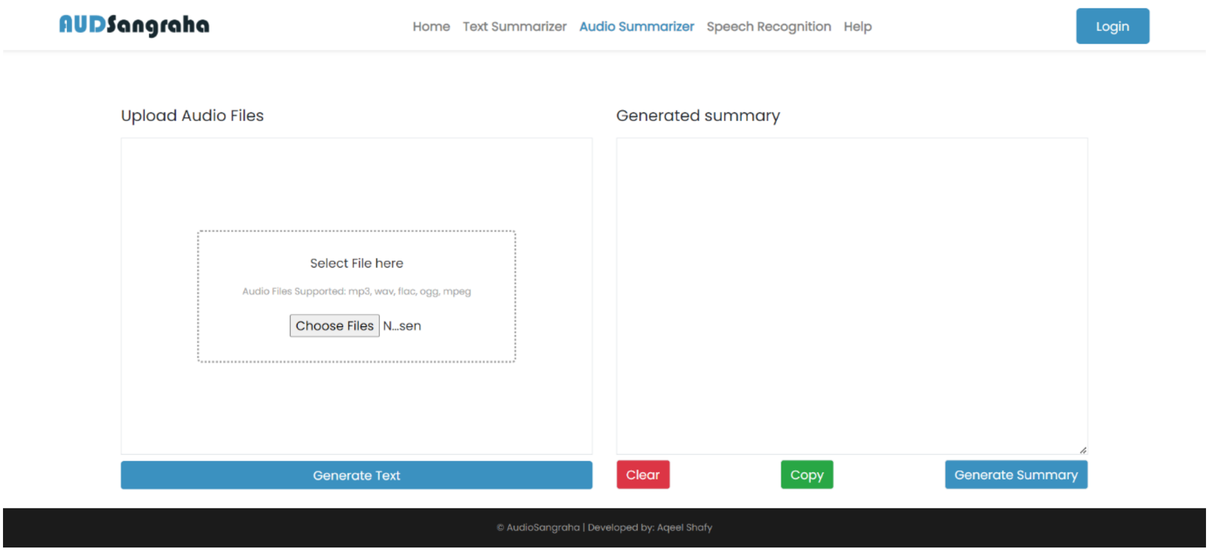


Figure 43: High-Fidelity of the UI (2)

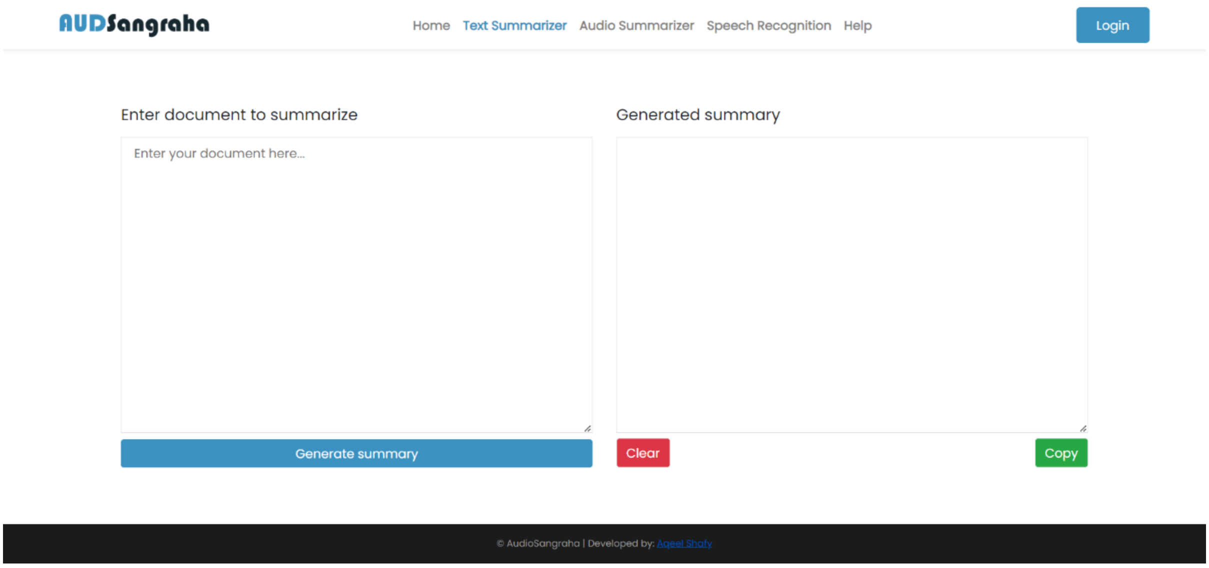


Figure 44: High-Fidelity of the UI (3)

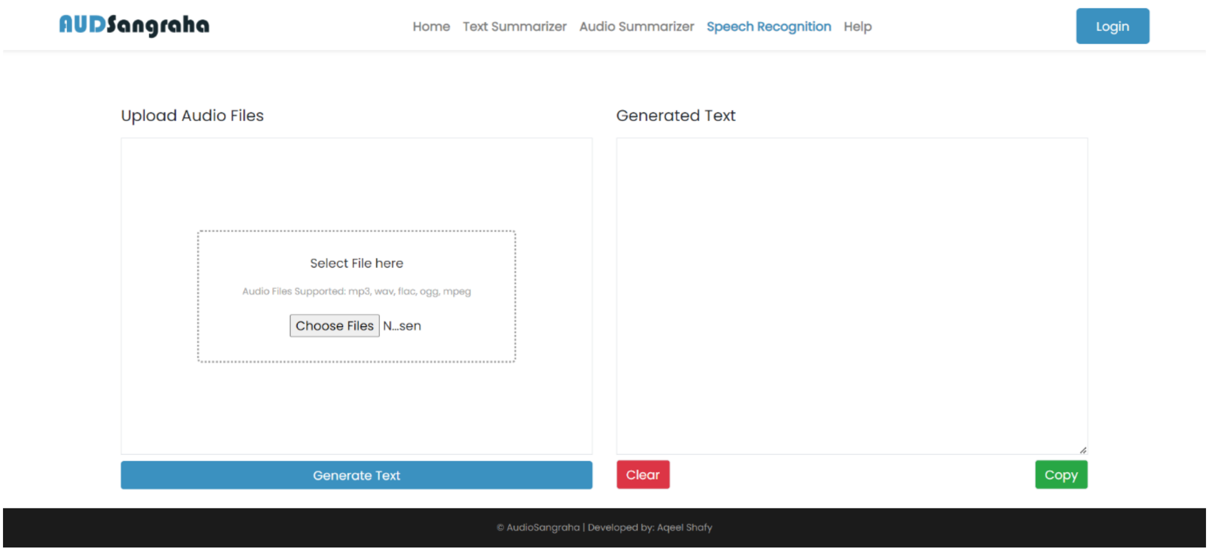


Figure 45: High-Fidelity of the UI (3)

APPENDIX-G: IMPLEMENTATION

Figure 46: Implementation of Home Page

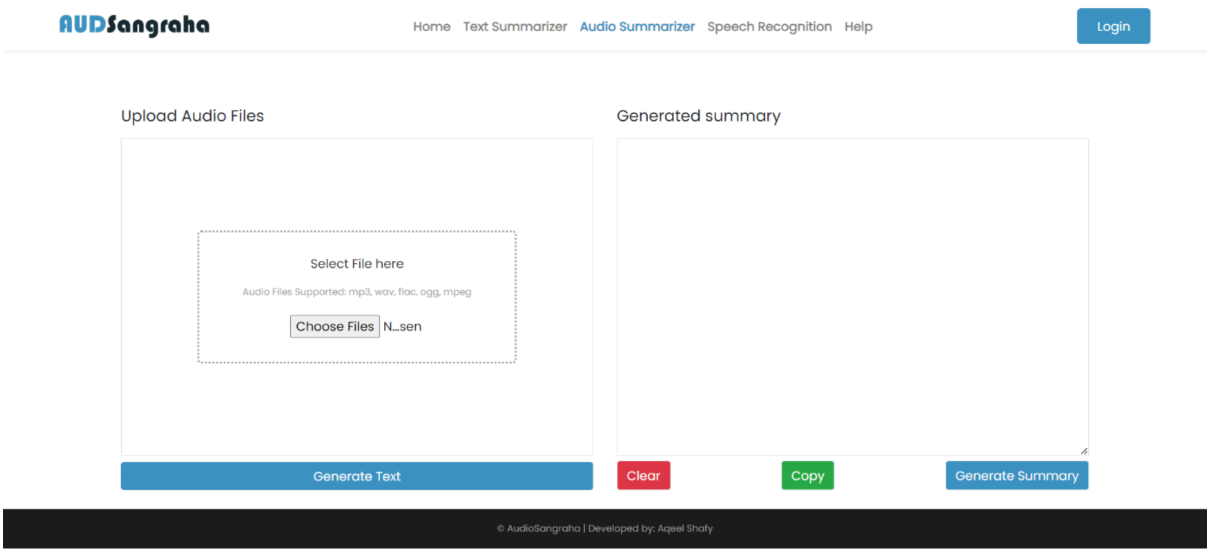


Figure 47: Implementation of Audio Summarizer Page

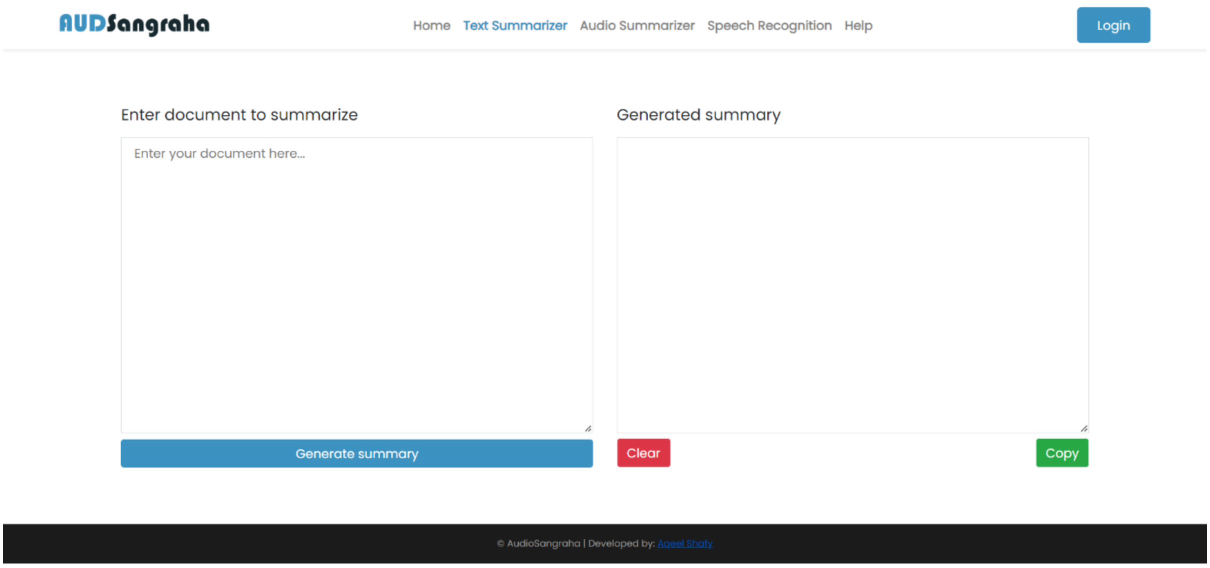


Figure 48: Implementation of Text Summarizer Page

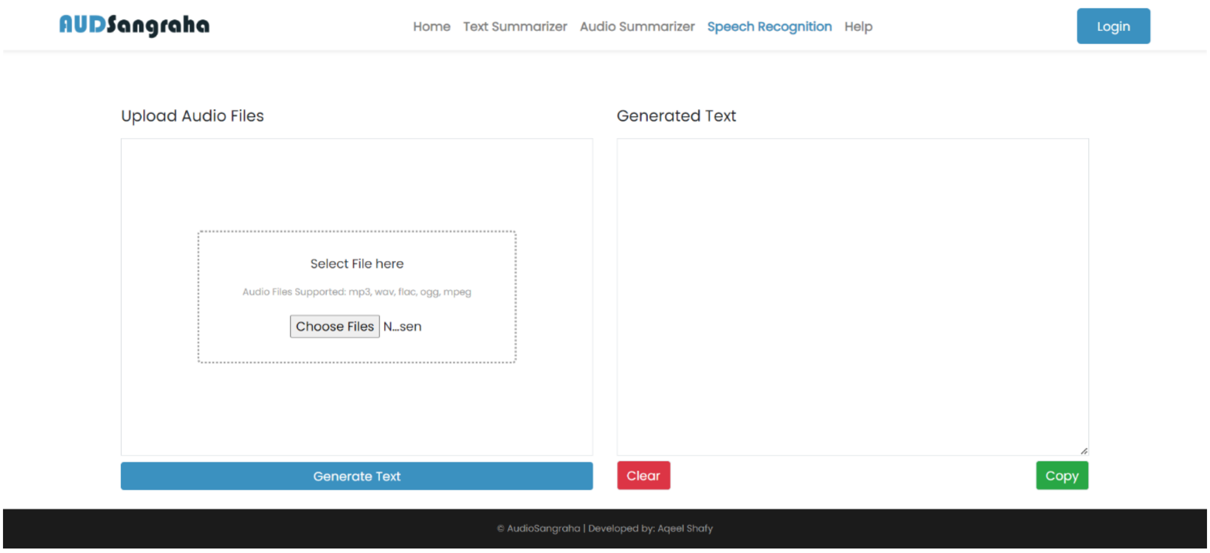


Figure 49: Implementation of Speech Recognition Page

APPENDIX-H: USE CASE

Table 38: Use Case Specification (3)

Use Case Name	Input Sinhala Text
Use case ID	UC3
Description	The user needs to add Sinhala text to input field.
Priority	High
Actors	User
Pre-conditions	Should contain more than one sentence.
Post-condition	User is able to see the generated text, by given text paragraph
Extended use case	None
Included use case	None
Main flow	1. User inputs Sinhala text for the input field. 2. System will generate the summary.
Alternative flow	None
Exceptional flow	If user input less sentence to the field, it might display same text as the summary output.

APPENDIX-I: EVALUATION

Table 39: Opinion of Evaluators

The Evaluators	**Opinion**
Dr. Ruvan Weerasinghe Lecturer at IIT	*"This point of area is valid research on summarizing the Sinhala audios. Also, it has a great contribution using transfer learning the whisper model for speech recognition on Sinhala. With the resources available the testing result is okay. This feedback was given before (and as the whisper model recognizes only 30 seconds of audio you can use a loop on taking multiple audio inputs and combine as a paragraph and then using the extractive approach you can summarize it accordingly). Also, it's great you have used the approach that I have mentioned. Also mentioned that in future worked this can be improved"*
Mr. Buddhi Gamage Lecturer at UCSC	*"As I have recently published a research paper on Deep speech toolkit for speech recognition for Sinhala. Using transfer learning from the Whisper model for speech recognition is a great choice which you can be achieved the same from it. But you used a very small training dataset for the training purpose, that what you have got a higher value of WER. But with the resources available within you it's okay. Also mentioned in the future that this can be improved with a larger and quality checked data set for training. And for the summarization you try the abstractive approach in the future."*

Mr. Aadhil Mohamed Senior Software Engineer – Anonymous workplace	*"Nowadays going through summarization systems is time saving and crucial thing within the busy life. Also, in that audio summarization a great approach for the Sinhala language user around the globe. As my knowledge a speech recognition model should be trained on a larger dataset, also it needs high computational power for the training. Within your resources available the output result provide is fine. Also, this can be improved in the coming days."*
Mr. Malik Works at Anonymous workplace	*"When I go through the application itself, I got to know what the application is, as the UI of system is very clean and very easy to understand. It's a great deal for the Sinhala language users to summarize the audios. But it would be great if it could handle a lengthy audio file into summaries in the future. But I know that it was hard to get it with your available resources."*
Mr. Franco de Silva	*"This is great application for the Sinhala community on summarizing the audios. And it is a great and useful idea for summarizing domain. And the UI of the system is very interesting as it is very easy to understand the functionality of the system. Also, I know that you had a very hard time implementing that accuracy for the speech to text model. Without losing hope you can improve the system in the future."*