

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

**AudioSangraha - An Approach Transforming Sinhala Audio into Summaries**

A Product Specification, Design and Prototype by

Mr. Aqeel Shafy

Supervised by

Mr. Ragu Sivaraman

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

**February 2024**

**ABSRTACT**

Language is a unique form of communication between human beans 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 are available for range of tasks. But still there is a prominent research area for speech recognition 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 lengthy audio contents. Even if they listen to lengthy audios, as a result they skip and try to get the information. Due to this they might get a wrong picture of information.

So, as a solution the author has proposed a system for summarizing the lengthy Sinhala audio contents. Due to this people can save their valuable time while getting the correct information through the audio easily. This system takes an audio file/ user can record an audio file to the system and get the generated summary output.

Using Whisper AI author has trained a new model for Audio to Text process. Throughout author will be correcting the grammar and spelling errors to get a high accuracy. Then the author using a model for extractive summarization and generates the summary accordingly.

**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

**TABLE OF CONTENTS**

[CHAPTER 01: INTRODUCTION 1](#_Toc158232813)

[1.1 Chapter Overview 1](#_Toc158232814)

[1.2 Problem Background / Problem Domain 1](#_Toc158232815)

[1.2.1 Natural Language Processing 1](#_Toc158232816)

[1.2.2 Low resource Sinhala language 1](#_Toc158232817)

[1.2.3 Speech to Text 2](#_Toc158232818)

[1.2.4 Text Summarization 2](#_Toc158232819)

[1.3 Problem Definition 2](#_Toc158232820)

[1.3.1. Problem Statement 3](#_Toc158232821)

[1.4 Aims and Objectives 3](#_Toc158232822)

[1.4.1 Research Aim 3](#_Toc158232823)

[1.4.2 Research Objectives 3](#_Toc158232824)

[1.6 Research Gap 6](#_Toc158232825)

[1.7 Contribution to the Body of Knowledge 7](#_Toc158232826)

[1.7.1 Contribution to the problem domain 7](#_Toc158232827)

[1.7.2 Contribution to the research domain 7](#_Toc158232828)

[1.8 Research Challenge 7](#_Toc158232829)

[1.9 Chapter Summary 7](#_Toc158232830)

[CHAPTER 02: SOFTWARE REQUIREMENT SPECIFICATION 8](#_Toc158232831)

[2.1 Chapter Overview 8](#_Toc158232832)

[2.2 Rich Picture Diagram 8](#_Toc158232833)

[2.3 Stakeholder Analysis 9](#_Toc158232834)

[2.3.1 Stakeholder Onion Model 9](#_Toc158232835)

[2.3.2 Stakeholder Viewpoints 9](#_Toc158232836)

[2.4 Selection of Requirement Elicitation Methods 10](#_Toc158232837)

[2.5 Discussion of Findings 11](#_Toc158232838)

[2.5.1 Findings from Literature Review 11](#_Toc158232839)

[2.5.2 Findings from Survey 12](#_Toc158232840)

[2.5.3 Findings from Interview 16](#_Toc158232841)

[2.5.4 Summary of Findings 17](#_Toc158232842)

[2.6 Context Diagram 18](#_Toc158232843)

[2.8 Use Case Specification 19](#_Toc158232844)

[2.9 Requirements with prioritization 20](#_Toc158232845)

[2.9.1 Functional Requirement 21](#_Toc158232846)

[2.9.2 Non-Functional Requirement 21](#_Toc158232847)

[2.10 Chapter Summary 22](#_Toc158232848)

[CHAPTER 03: DESIGN 23](#_Toc158232849)

[3.1 Chapter Overview 23](#_Toc158232850)

[3.2 Design Goals 23](#_Toc158232851)

[3.3 High level Design 24](#_Toc158232852)

[3.3.1 Architecture Diagram 24](#_Toc158232853)

[3.3.2 Discussion of tiers/ layers of the Architecture 25](#_Toc158232854)

[3.4 System Design 25](#_Toc158232855)

[3.4.1 Choice of design paradigm 25](#_Toc158232856)

[3.5 Detailed Design Diagrams 26](#_Toc158232857)

[3.5.1 Data Flow Diagram 26](#_Toc158232858)

[3.5.2 System Process Flowchart 27](#_Toc158232859)

[3.5.3 User Interface Design 28](#_Toc158232860)

[3.6 Chapter summary 28](#_Toc158232861)

[CHAPTER 04: IMPLEMENTATION 29](#_Toc158232862)

[4.1 Chapter Overview 29](#_Toc158232863)

[4.2 Technology Selection 29](#_Toc158232864)

[4.2.1 Technology Stack 29](#_Toc158232865)

[4.2.2 Dataset Selection 30](#_Toc158232866)

[4.2.3 Development Frameworks 30](#_Toc158232867)

[4.2.4 Programming Languages 31](#_Toc158232868)

[4.2.5 Libraries 31](#_Toc158232869)

[4.2.6 IDE 31](#_Toc158232870)

[4.2.7 Summary of Technology Selection 32](#_Toc158232871)

[4.3 Implementation of the Core Functionality 32](#_Toc158232872)

[4.3.1 Audio to Text model implementation 33](#_Toc158232873)

[4.4 User Interface 34](#_Toc158232874)

[4.5 Chapter Summary 34](#_Toc158232875)

[CHAPTER 05: CONCLUSION 35](#_Toc158232876)

[5.1 Chapter Overview 35](#_Toc158232877)

[5.2 Deviations 35](#_Toc158232878)

[5.2.1 Scope related deviations 35](#_Toc158232879)

[5.2.2 Schedule related deviations 35](#_Toc158232880)

[5.3 Initial Test Results 35](#_Toc158232881)

[5.4 Required Improvements 35](#_Toc158232882)

[5.5 Demo of the Prototype 36](#_Toc158232883)

[5.6 Chapter Summary 36](#_Toc158232884)

[REFERENCES 37](#_Toc158232885)

[APPENDIX 41](#_Toc158232886)

**LIST OF FIGURES**

[Figure 1: Rich Picture Diagram 8](#_Toc158225533)

[Figure 2: Stakeholder Onion Model 9](#_Toc158225534)

[Figure 3: Context Diagram 18](#_Toc158225535)

[Figure 4: Use Case Diagram 18](#_Toc158225536)

[Figure 5: High Level Design 24](#_Toc158225537)

[Figure 6: Data Flow Diagram 1 26](#_Toc158225538)

[Figure 7: Data Flow Diagram 2 26](#_Toc158225539)

[Figure 8: Flowchart 27](#_Toc158225540)

[Figure 9: Split into Training and Testing Sets 33](#_Toc158225541)

[Figure 10: Read the Audio Files 33](#_Toc158225542)

[Figure 11: Pre-trained Whisper Model 33](#_Toc158225543)

[Figure 12: Setting Training Arguments 34](#_Toc158225544)

[Figure 13: Train model 34](#_Toc158225545)

**LIST OF TABLES**

[Table 1: Research Objectives 6](#_Toc158222922)

[Table 2: Stakeholder Analysis 10](#_Toc158222923)

[Table 3: Requirement Elicitation Methods 11](#_Toc158222924)

[Table 4: Literature Review Findings 11](#_Toc158222925)

[Table 5: Survey Findings 15](#_Toc158222926)

[Table 6: Interview Findings 17](#_Toc158222927)

[Table 7: Summary of Findings 17](#_Toc158222928)

[Table 8: UC1 Description 19](#_Toc158222929)

[Table 9: UC2 Description 20](#_Toc158222930)

[Table 10: MoSCoW Principal 20](#_Toc158222931)

[Table 11: Functional Requirements 21](#_Toc158222932)

[Table 12: Non-functional Requirements 22](#_Toc158222933)

[Table 13: Design Goals 24](#_Toc158222934)

[Table 14: Technology Stack 30](#_Toc158222935)

[Table 15: Development Frameworks 30](#_Toc158222936)

[Table 16: Libraries used for The System Implementation 31](#_Toc158222937)

[Table 17: IDEs used for The System Implementation 32](#_Toc158222938)

[Table 18: Summary of Technology Selection 32](#_Toc158222939)

[Table 19: Required Improvements 36](#_Toc158222940)

**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 as 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 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 the 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 the 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 the 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 the technological improvement, listening audios like broadcasting of news on radio channels, speeches, lecture audios and audio messages etc are time consuming. As some of them are not intrusive to 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 Aims and Objectives

### 1.4.1 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 with the extractive summarization. And finally, the user will be able to get a summarized version of audio unless listening to continuous audio.

### 1.4.2 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 summarizers for low resource languages. | LO1, LO3, LO6 | RQ3 |
| 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 audio summarizer  RO6 - To identify the specific challenges when it comes to Sinhala summarizer  RO7 - To identify what are the datasets available for the research | L01, L04, LO5, LO8 | RQ2, RQ1 |
| Requirement Elicitation | R08 - To gather the requirements and feedbacks by the industrial experts  R09 - To collect the user reviews from the existing systems to improve   RO10 - To identify the requirements related to build the system | LO1, LO3, LO5, LO6 | RQ2, RQ3 |
| Design | RO11 - To create a user-friendly interface   RO12 - To create the required data flows, diagrams to design the architecture | LO2, LO5, LO7 | RQ2 |
| Implementation | RO13 - To develop the model to convert the audio 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, RQ3 |
| Testing and Evaluation | RO17 - To provide a high performance and accuracy in the system  RO18 - To create the test plan related to the system  RO19 - To perform the testing of unit testing, functional testing and usability testing | LO1, LO5, LO7, | RQ2, RQ3 |
| Documentation | RO20 - To organize a structured document related to the research | LO8 | RQ2, RQ3 |

Table 1: Research Objectives

## 1.6 Research Gap

Most of the research has contributed to the automatic summarization system, and Speak recognition system for the English language. And there is a noticeable research gap when it comes to the low resource languages. When compared to the existing work the author has elaborated to implement a system for the target audience of Sinhala language.

* Summarizing the low resource language of Sinhala audio files with extractive summarization (Warnasooriya et al., 2020)
* Correcting the Sinhala language grammar, spelling of speech when it converts in to audio to text (Weerasinghe et al., 2020)
* Improving the accuracy of the poor audio quality data (Weerasinghe et al., 2020)

## 1.7 Contribution to the Body of Knowledge

### 1.7.1 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 researches, 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 lengthy audios, as it will be summarized.

### 1.7.2 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 addressing the accuracy of the spelling and grammatical issues in Sinhala Language (Weerasinghe et al., 2020) as it is known as a low resourced language, because there aren't enough data resources available on the internet (Deshpande & Jahirabadkar, 2021).

## 1.8 Research Challenge

In this research there are three main tasks. They are generating the audio to text, correcting the Sinhala spelling and the grammar of the text of the audio when it converts into text, summarizing the text with 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 data, techniques and tools to improve the accuracy of the summarizer.

## 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.

# 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

2.3 Problem Domain2.3.1. Speech Recognition

Speech is the most suitable and the most natural way of communication between humans. Over the decades the computers have been trained for the tasks which the humans can do. Speech recognition also one of the tasks 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 the 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 the 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 Compute Power

When developing a speech recognition model, a significant challenge is the use of the computer 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 compute 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 the 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 realtime 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 of dataset for training will enhance the performance. Also, they state that for low resource languages this approach might be beneficial.

### 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.

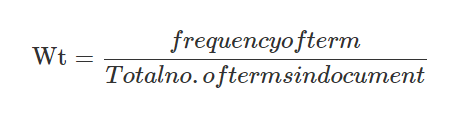
### 2.4.2 Text Summarization

#### 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.



A black text on a white background

Description automatically generated

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, 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 of capturing 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).

#### 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 stopwords 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.

**A black text on a white background

Description automatically generated**

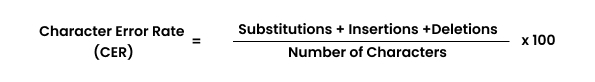
**Sentence Error Rate (SER)**

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.

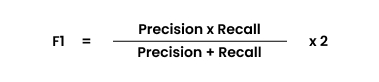
This has been useful for identifying the accuracy of the recognized characters in the low resource languages and languages which have a complex vocabulary. This also uses a calculator to find out the CER (Violeta and Toda, 2023).



### 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.



## 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 has been discussed what evaluation metrics have been used.

# CHAPTER 02: METHODOLOGY

## 2.1. 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.

## 2.2 Research Methodology

|  |  |
| --- | --- |
| 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 prioritising methods and approaches. |
| Approach | Author has selected, Deductive research approach here.  As 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. |

Table 3: Research methodology

## 2.3 Development methodology

### 2.3.1 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.

### 2.3.2 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, analyse and develop the system successfully.

### 2.3.3 Programming Paradigm

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

### 2.3.4 Evaluation (Testing) 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.

### 2.3.5 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.

## 2.4 Project Management Methodology

Here as the project management methodology, the author has chosen the Agile Prince 2. As it is flexible and it has the ability to have a proper plan to manage and complete the project.

### 2.4.1 Project scope

The scope of the project is to implement a system to summarise the lengthy audios related to the use of the target audience of Sinhala language. Therefore, it uses the specialised extractive summarization to summarise the audio input and language correction to increase the accuracy. As the project scope the system uses the data correction, model fine-tuning etc to get a quality of summary.

#### 2.4.1.a In Scope

1. Implementing a web-based application that takes the real time speech or uploading a mp3 file as an input and summarise and provides the output
2. Collecting a high quality of dataset for the summarising approach
3. Creation of a high accuracy of the grammar and spelling mistakes of Sinhala language when converting audio to text
4. Increasing and improving the accuracy of poor audio quality data when it converts in to text

#### 2.4.1.b 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.

#### 2.4.1.c Diagram Depicting the Prototype Feature

A diagram of a speech and speech process

Description automatically generated with medium confidence

Figure 1: Diagram depicting the prototype for audio summarizing

### 2.4.2 Schedule

#### 2.4.2.a Gantt Chart

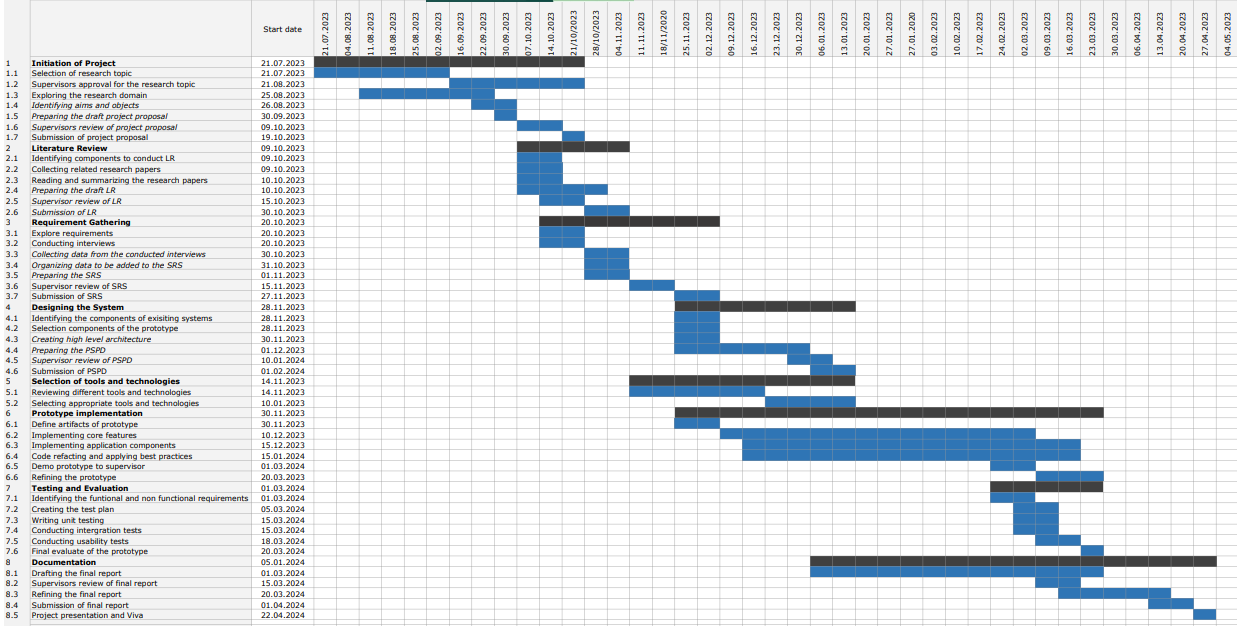


Figure 2: Gantt chart

#### 2.4.2.b Deliverables and Date

|  |  |
| --- | --- |
| **Deliverable** | **Date** |
| Draft version of Project Proposal | 1st September 2023 |
| Finalised 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

### 2.4.3 Resource Requirements

#### 2.4.3.a 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. |

Table 5: Hardware requirements

#### 2.4.3.b 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 |
| TensorFlow | This is a framework for the deep learning models of the project |
| Zotero | This is used to manage the citation and references related to the project |

Table 6: Software requirements

#### 2.4.3.c Data Requirements

Datasets related to the project will be used, which are available on Kaggle and HuggingFace

#### 2.4.3.d Skill Requirements

* A basic understand on OOP
* Knowledge on developing a web-based applications
* An understanding on NLP techniques
* An understanding on model training and dataset creation

### 2.4.4 Risk 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 the 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 |

Table 7: Risk and mitigation plan

## 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.

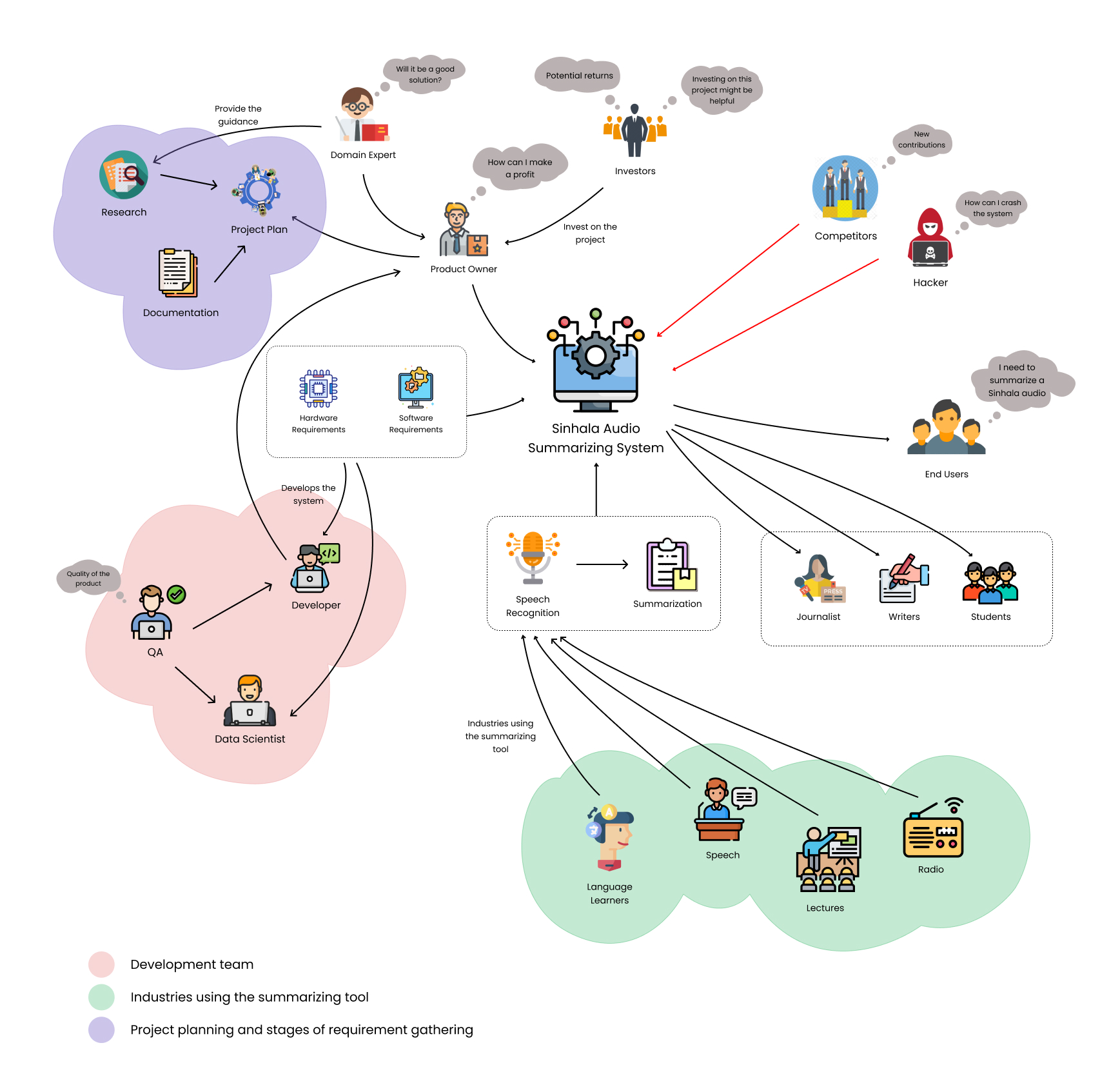


Figure 1: Rich Picture Diagram

## 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.

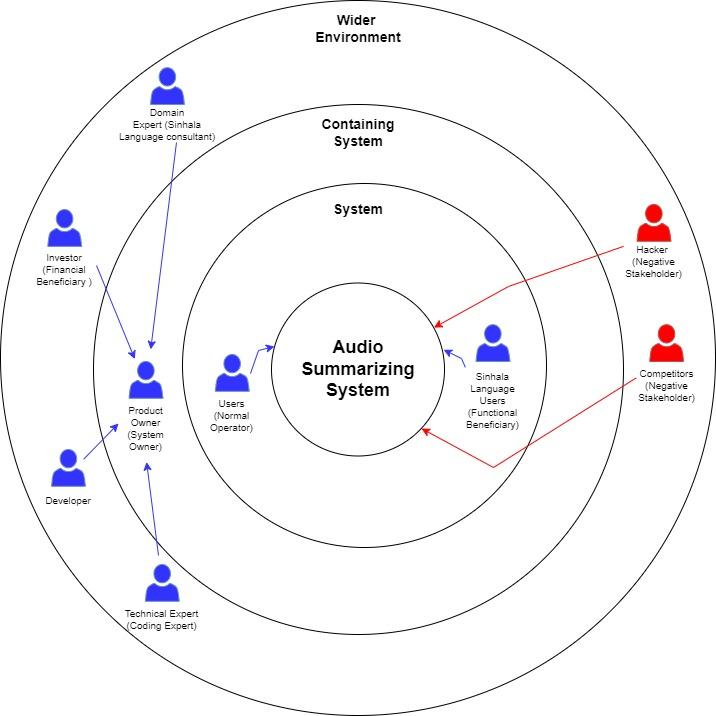


Figure 2: Stakeholder Onion Model

### 4.3.2 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 |

Table 2: Stakeholder Analysis

## 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.

|  |
| --- |
| **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. |

Table 3: Requirement Elicitation Methods

## 4.5 Discussion of Findings

### 4.5.1 Findings from Literature Review

|  |  |
| --- | --- |
| **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) |

Table 4: Literature Review Findings

### 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 the survey can be found.

|  |  |
| --- | --- |
| **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. |
| **Findings** | 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. |
| **Question** | **When it comes to a lengthy audio, how much will you complete listening?** |
| **Aim** | From the above question it identifies how much they will complete listening lengthy audio contents. |
| **Findings** | 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. |
| **Question** | **What are the challenges you face while listening to long audio contents?** |
| **Aim** | This to find out what are the challenges they encountered while 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. |

Table 5: Survey Findings

### 4.5.3 Findings from Interview

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

|  |  |  |
| --- | --- | --- |
| **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. |

Table 6: Interview Findings

## 4.6 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 |  | ✓ | ✓ |

Table 7: Summary of Findings

## 4.7 Context Diagram

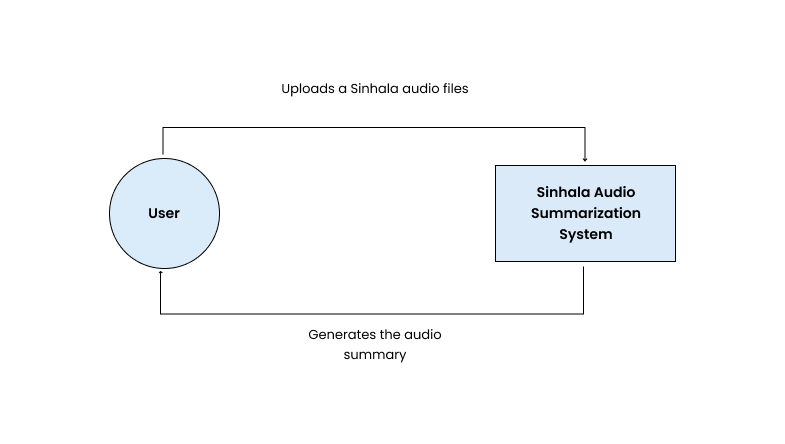
The context diagram provides the system boundaries and the interaction between the users. In the below diagram it shows the user has to upload an audio file or record an audio file to the system. And the system will generate the summary of the audio to the user.   


Figure 3: 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.

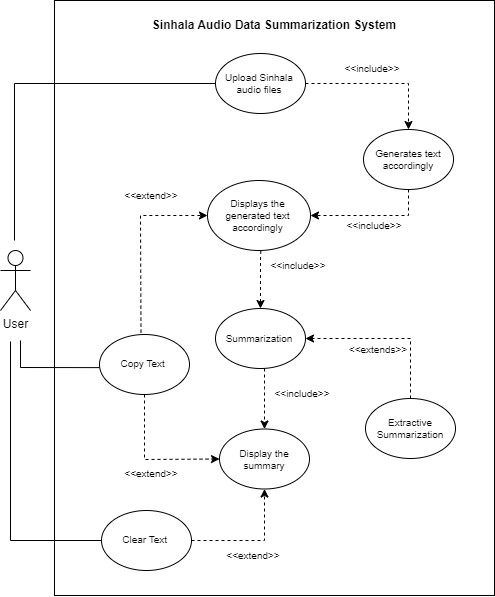


Figure 4: Use Case Diagram

## 4.8 Use Case Specification

|  |  |
| --- | --- |
| Use Case Name | Upload Sinhala audio files |
| Use case ID | UC1 |
| Description | The user needs to upload 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. |
| Alternative flow | None |
| Exceptional flow | File format errors will be displayed |

Table 8: UC1 Description

|  |  |
| --- | --- |
| 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 | Error while summarizing, or will generates the same results as the converted text input |

Table 9: UC2 Description

## 4.9 Requirements with prioritization

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

|  |  |
| --- | --- |
| 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 the scope, which will not be implemented on the system. |

Table 10: MoSCoW Principal

### 4.9.1 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 |

Table 11: Functional Requirements

### 4.9.2 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 |

Table 12: Non-functional Requirements

## 4.10 Chapter Summary

This chapter discussed about 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 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

|  |  |
| --- | --- |
| **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. |

Table 13: Design Goals

## 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.

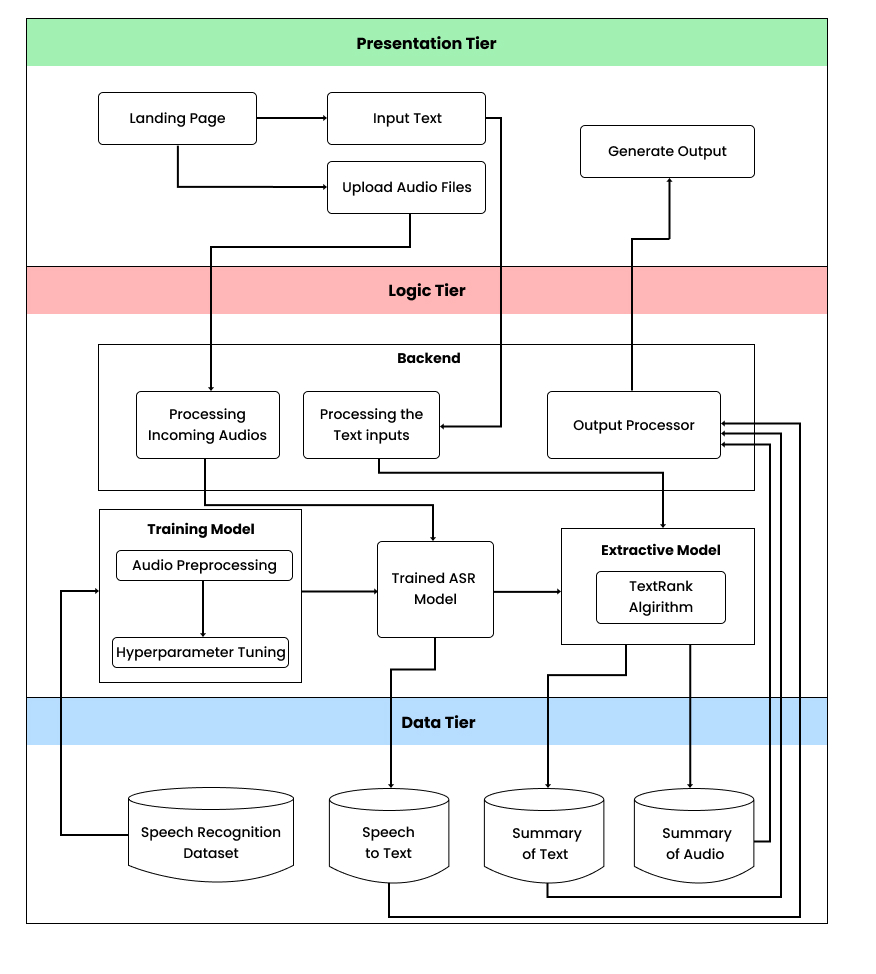


Figure 5: High Level Design

### 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 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.

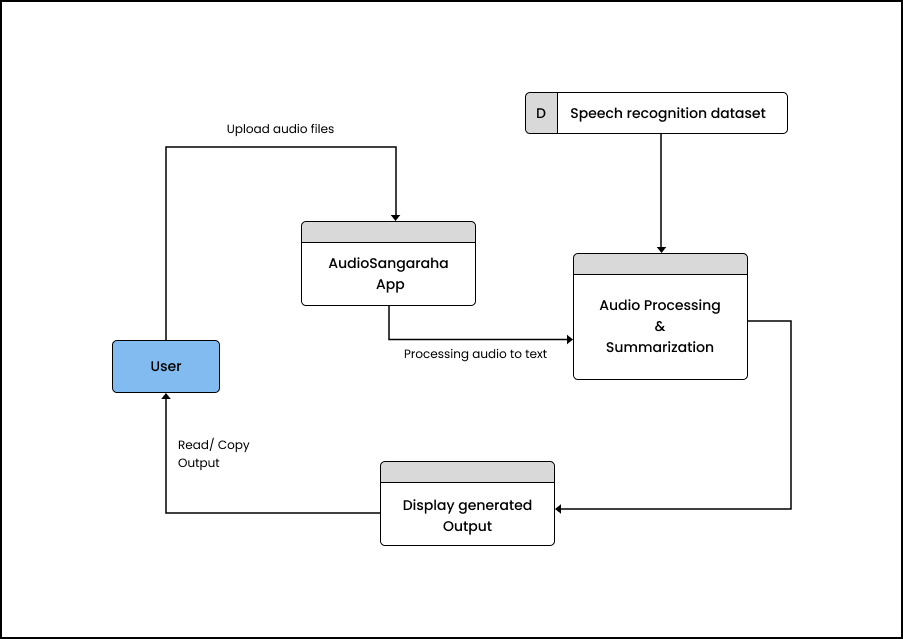


Figure 6: Data Flow Diagram 1

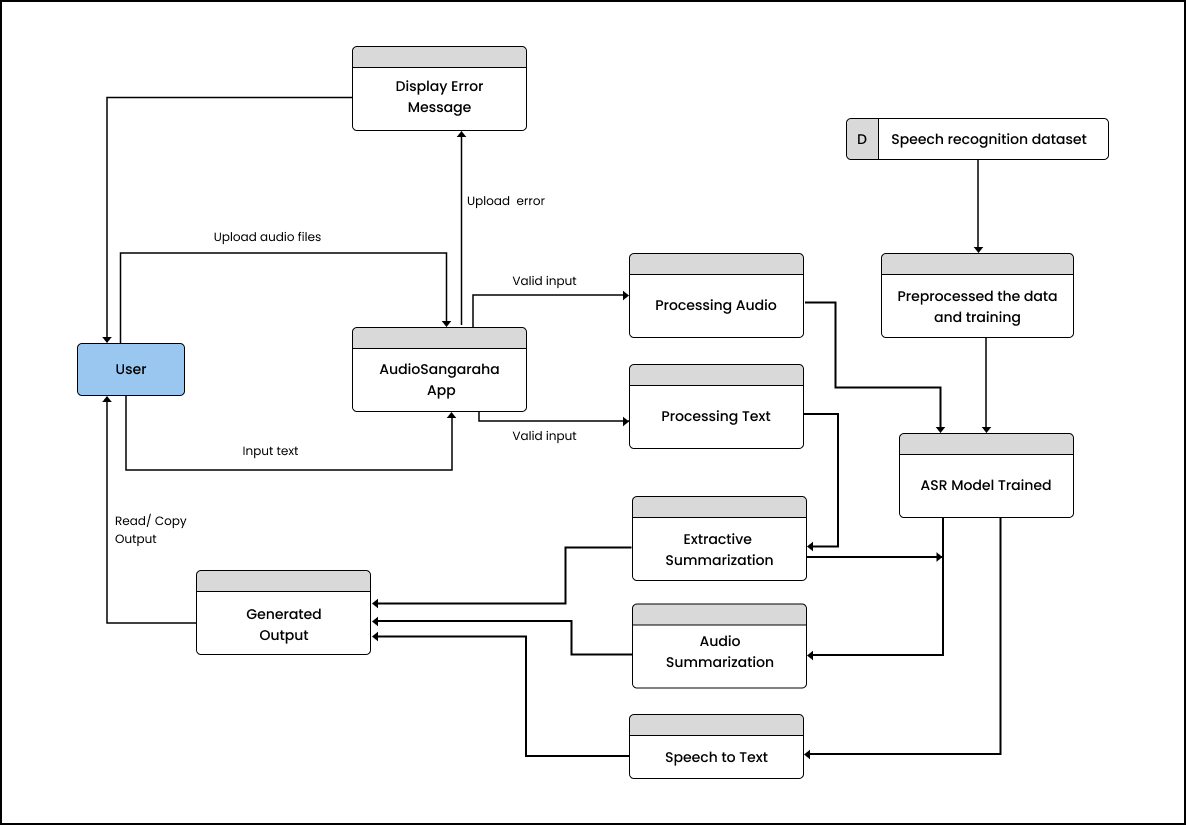


Figure 7: Data Flow Diagram 2

### 6.5.2 System Process Flowchart

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

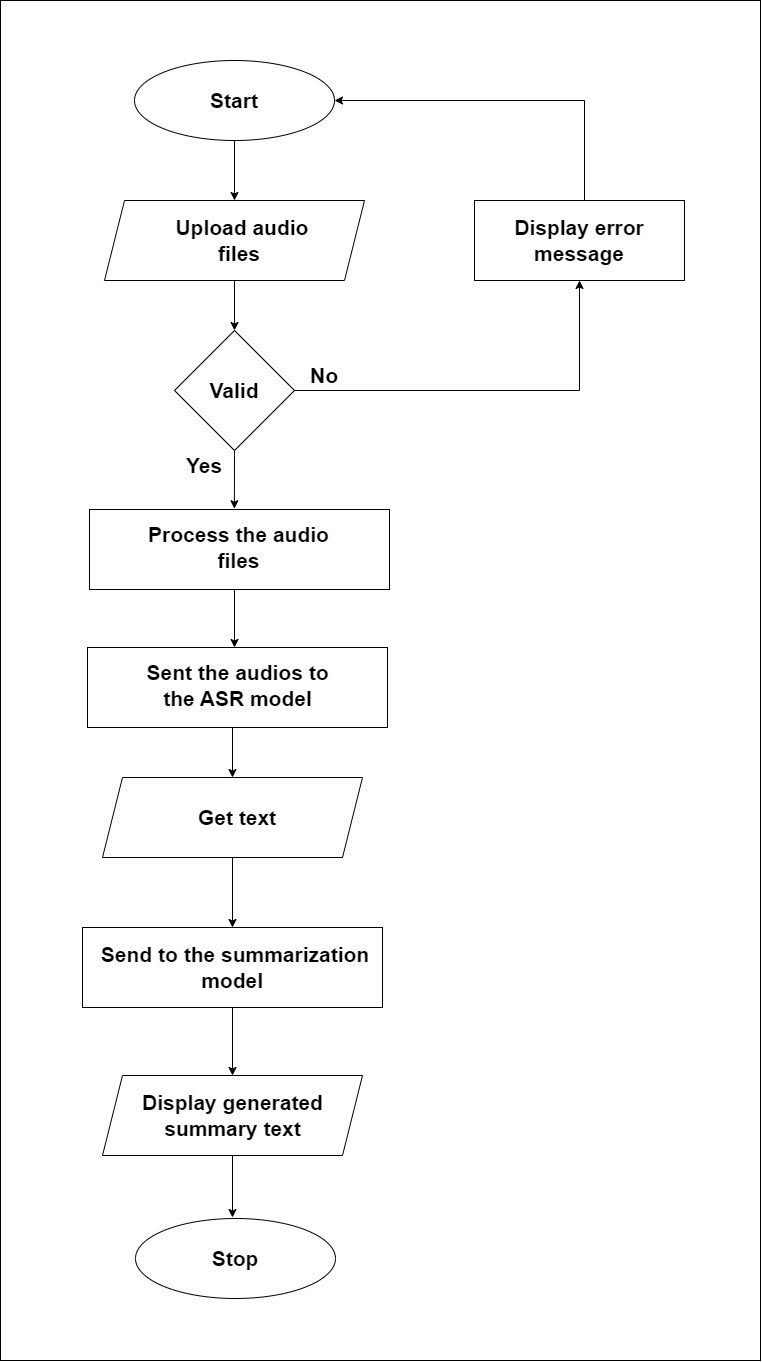


Figure 8: Flowchart

### 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

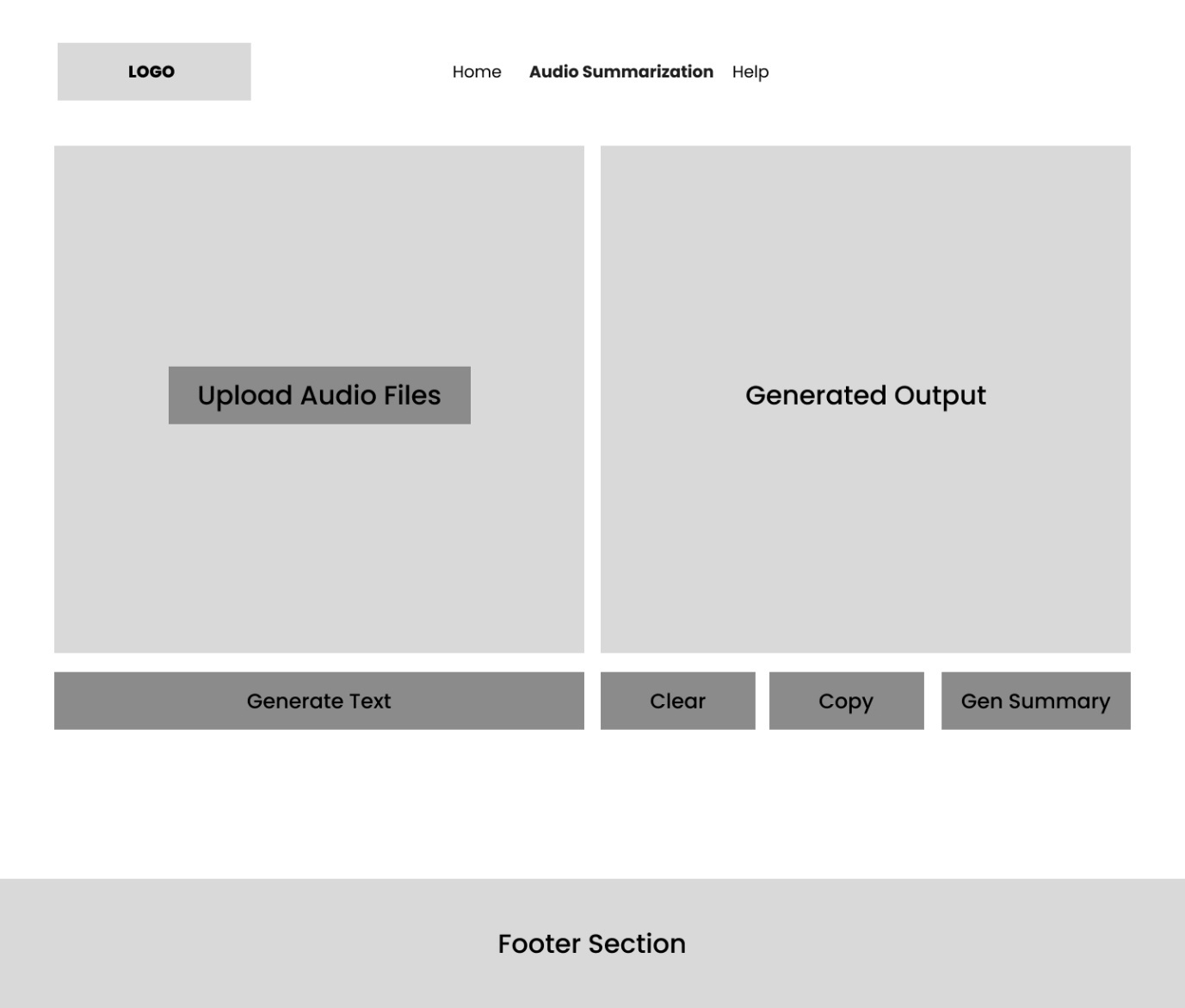


Figure 9: Wireframe of the system

## 6.6 Chapter summary

In this chapter it discussed about 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 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

|  |
| --- |
| **Presentation Tier** |
|  |
| **Logic Tier** |
|  |
| **Data Tier** |
|  |

Table 14: Technology Stack

### 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.

|  |  |
| --- | --- |
| **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. |

Table 15: Development Frameworks

### 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

|  |  |
| --- | --- |
| **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 |

Table 16: Libraries used for The System Implementation

### 7.2.6 IDE

|  |  |
| --- | --- |
| 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. |

Table 17: IDEs used for The System Implementation

### 7.2.7 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 |

Table 18: Summary of Technology Selection

## 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

### A screen shot of a computer program Description automatically generated

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

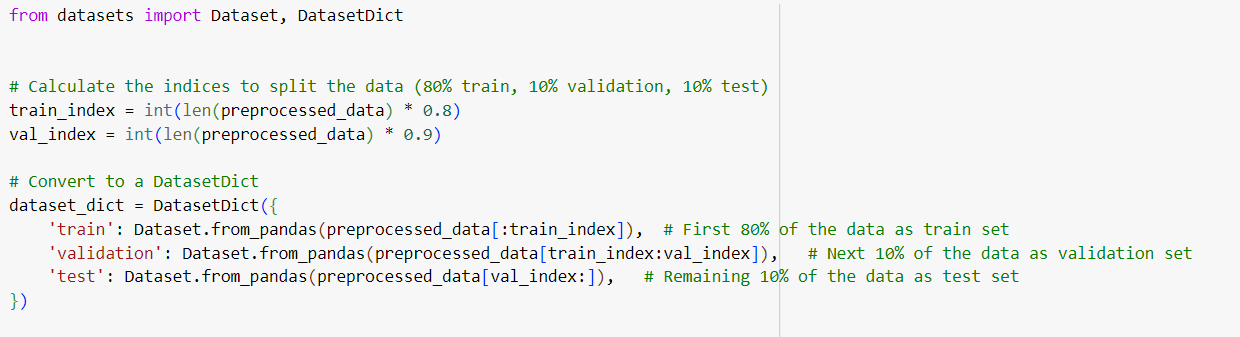


Figure 10: Split into Training, Testing and Validation Sets

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.

A white background with colorful lines

Description automatically generated

Figure 11: Map the Audio Files

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.

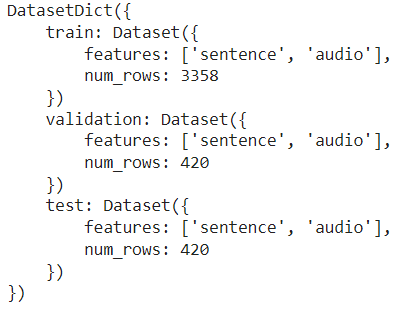


Figure 12: Mapped Output

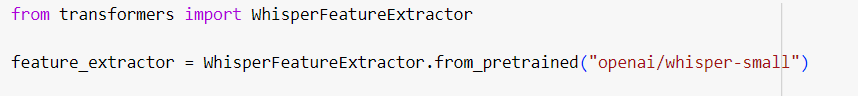




Figure 13: Loading Whisper Model

Here a pre-trained whisper model will be loaded using the ‘transformers’ library. 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



Figure 14: Setting Training Arguments

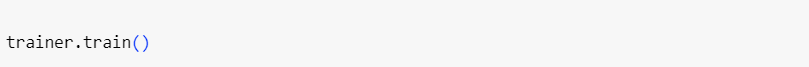


Figure 15: Train model

### 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.

A screen shot of a computer program

Description automatically generated

Figure 16: Stop-word Removal

A screenshot of a computer program

Description automatically generated

Figure 17: Generate frequency and Scoring sentence

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.

## A computer screen shot of a code Description automatically generated

Figure 18: Generate Summary

## 7.4 User Interface

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



Figure 19: Home Page UI

## 

Figure 20: Audio Summarization UI

## 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.

A screenshot of a computer program

Description automatically generated

Figure 21: MER Testing

### 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.

A screen shot of a computer code

Description automatically generated

Figure 22: 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 reason for getting a high WER score.

## 

Figure 23: 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 to copy and reset | Should be able to copy the generated summary and reset | Able to copy the summary text and reset | Pass |

## 8.6 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.

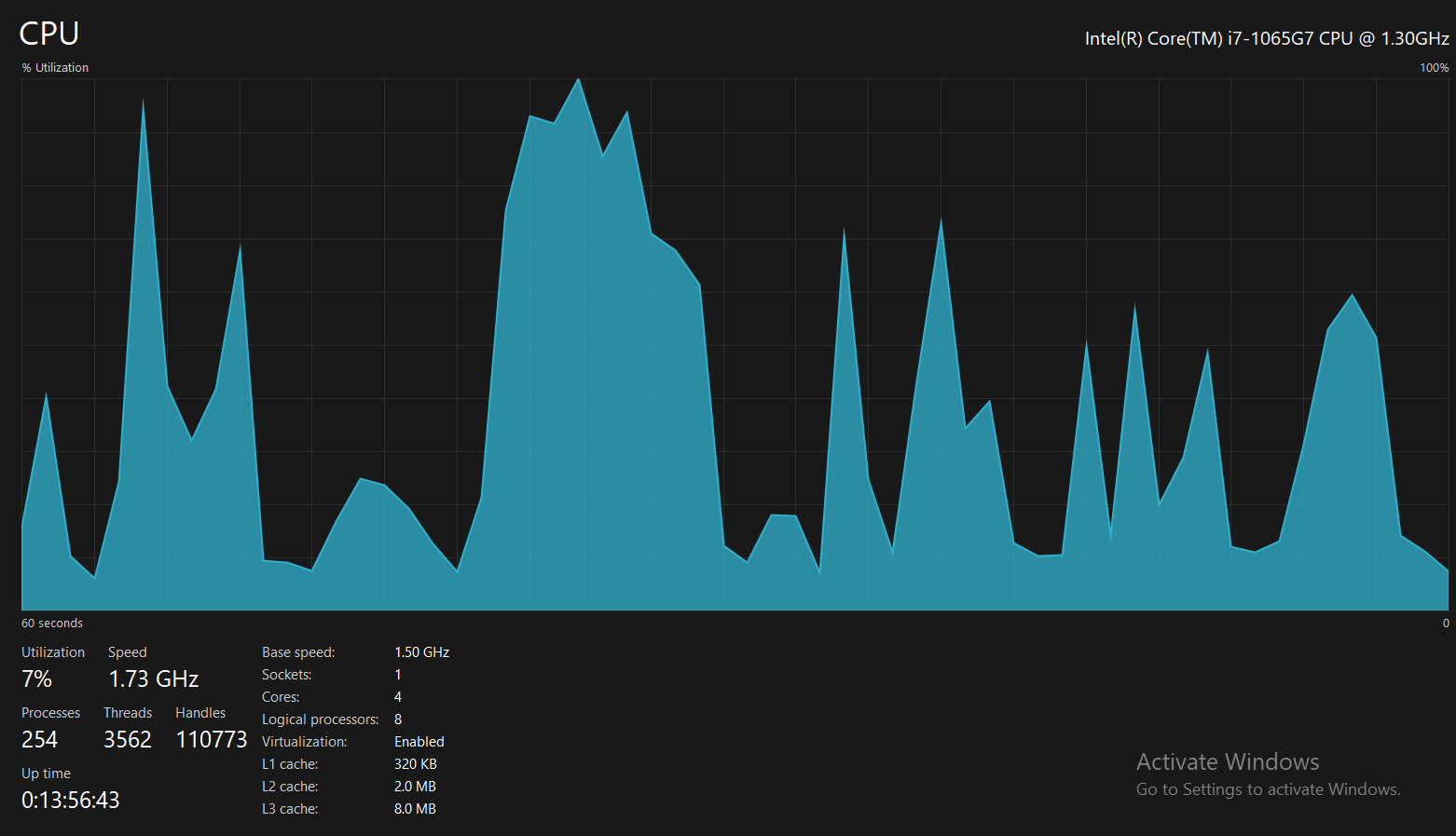


Figure 24: CPU Performance

A screenshot of a computer

Description automatically generated

Figure 25: 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.

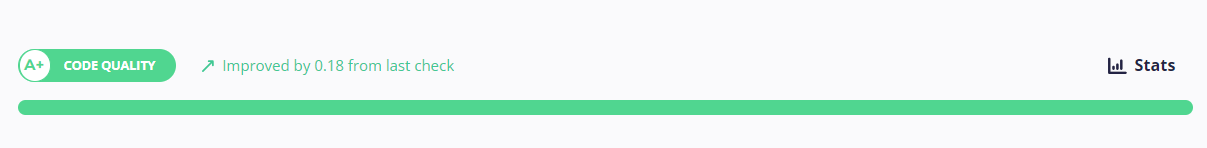


Figure 26: Code Quality Checker

## 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 on 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 on 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.

|  |  |
| --- | --- |
| **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.

|  |  |
| --- | --- |
| **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 below table author has stated the count of the evaluators.

|  |  |
| --- | --- |
| **Evaluators** | **Count** |
| Domain experts | 2 |
| Technical experts | 4 |
| Normal users | 5 |

## 9.6 Evaluation Result

### 9.6.1 Opinion of Domain Experts

|  |  |
| --- | --- |
| **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 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

|  |  |
| --- | --- |
| **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

|  |  |
| --- | --- |
| **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

|  |  |  |  |
| --- | --- | --- | --- |
| **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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 has been what is the progress level of the project, what are the future steps going to be taken. And using a small presentation the author has been demonstrating the prototype.

## 10.2 Achievements of Research Aims and Objectives

## 10.3 Utilizing of Knowledge from the Course

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

|  |  |
| --- | --- |
| **Module** | **Justification** |
| Software Development 1 | This module 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 understand 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 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 gain 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 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

|  |  |
| --- | --- |
| **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

|  |  |
| --- | --- |
| **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 Colab Pro version for training purposes. Also, after spending more than $40 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 include). 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 on various of the limitations.

* The dataset used for training the ASR model wasn’t sufficient.
* The ASR model training need 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.

## 10.11 Achievements of the Contribution to Body of Knowledge

## 10.12 Concluding Remarks

1. Chapter Overview
2. Achievements of Research Aims & Objectives (Based on chapter 1)
3. Utilization of Knowledge from the Course
4. Use of Existing Skills (What is learned from the course applied to the project)
5. Use of New Skills (What you have learned through the project – Not part of the curriculum) – Technical skills should be given preference
6. Achievement of Learning Outcomes
7. Problems and Challenges Faced – Need to mention how did you overcome the problems and challenges
8. Deviations – Any deviations from the original plan should be mentioned and justified
9. Limitations of the Research – Should be linked with the test output
10. Future Enhancements
11. Achievement of the contribution to body of knowledge
12. Concluding Remarks

## 5.2 Deviations

### 5.2.1 Scope related deviations

The author needs to build a model for the audio to text process. So, he has used a transfer learning approach for a whisper AI pretrained model and fine tune it according to the dataset gathered.

### 5.2.2 Schedule related deviations

Within the scheduled time period the author has completed the documentation related to the project and partially completed the prototype. The author is confident on completing the prototype within the given time period.

## 5.3 Initial Test Results

The dataset gathered was used to train, as there is a large amount of data the author has created a subset from the dataset. But unfortunately, the Colab notebook was getting crashed due to the GPU level. So, the author will be using the Colab pro version and training the model in the upcoming days.

## 5.4 Required Improvements

|  |  |
| --- | --- |
| **Required improvements** | **How you intend to do it** |
| Audio to Text model | Author will be using the Colab pro version for the training purposes. As the audio data is at a high level. And the author will improve grammatical and the spelling errors accordingly. |
| Text summarization | For text summarization the author will be implementing a model. |

Table 19: Required Improvements

## 5.5 Demo of the Prototype

The following link provides the code for the prototype

<https://colab.research.google.com/drive/1c010yHPWUTGNWVhHmbtU7vzSv9uqIrtg?usp=sharing>

The following YouTube link provides the video for the prototype

<https://youtu.be/3rNK2r6RJTw>

## 5.6 Chapter Summary

In this chapter it is been explained about the deviations from the scope and schedule of the project. Then it discusses about the improvement of proposed system. And at the end author gives a demo of the prototype.

## REFERENCES

Das , P. and Prasad, V. (2015) VOICE RECOGNITION SYSTEM: SPEECH-TO-TEXT. Available at: https://www.researchgate.net/publication/304651244\_VOICE\_RECOGNITION\_SYST EM\_SPEECH-TO-TEXT (Accessed: 01 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).

Deshpande, P. and Jahirabadkar, S. (2021) A survey on statistical approaches for abstractive summarization of low resource language documents, SpringerLink. Available at: https://link.springer.com/chapter/10.1007/978-981-16-4016-2\_69 (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).

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: 25 September 2023).

Dinushika, T. et al. (2020) Speech Command Classification System for Sinhala Language based on Automatic Speech Recognition. Available at: https://ieeexplore.ieee.org/document/9037648/ (Accessed: 15 January 2024).

Gamage, B. et al. (2021) Improving Sinhala Speech Recognition Through e2e LF-MMI Model, https://aclanthology.org. Available at: https://aclanthology.org/2021.iconmain.26.pdf (Accessed: 24 September 2023).

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).

Jing, B. et al. (2021) Multiplex Graph Neural Network for extractive text summarization, ACL Anthology. Available at: https://aclanthology.org/2021.emnlp-main.11/ (Accessed: 15 January 2024).

Karunathilaka, N.A.K.H.S. (2020) Low-resource Sinhala Speech Recognition using Deep Learning, Digital Library of University of Colombo School of Computing: Home. Available at: https://dl.ucsc.cmb.ac.lk/jspui/ (Accessed: 14 January 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).

Kemp, S. (2022) Digital 2022: Sri Lanka - datareportal – global digital insights, DataReportal. Available at: https://datareportal.com/reports/digital-2022-sri-lanka (Accessed: 05 September 2023).

Khandare, P. et al. (2019) AUDIO DATA SUMMARIZATION SYSTEM USING NATURAL LANGUAGE PROCESSING, 06(09). doi:https://www.irjet.net/archives/V6/i9/IRJETV6I957.pdf.

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>.

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&d q=Frank+Millstein.+(2020).+Natural+Language+Processing+Using+NLTK.+Frank+M illstein.&ots=02SOrlVUaE&sig=i1bsvq75ZnWN9HC2lhcD0F3dIc&redir\_esc=y#v=onepage&q=Frank%20Millstein.%20(20 20).%20Natural%20Language%20Processing%20Using%20NLTK.%20Frank%20Mill stein.&f=false (Accessed: 31 August 2023).

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).

Rathnayake, B.R.M.S.R.B., Manathunga, K. and Kasthurirathna, D. (2023) ‘Talking Books’ : A Sinhala Abstractive Text Summarization Approach for Sinhala Textbooks. Available at: https://ieeexplore.ieee.org/search/advanced (Accessed: 08 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.

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).

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

Warnasooriya , W.M.P.N. et al. (2020) Sinhala speech recognition system for journalists in Srilanka, SINHALA SPEECH RECOGNITION SYSTEM FOR JOURNALISTS IN SRILANKA. Available at: https://www.researchgate.net/publication/346624775\_SINHALA\_SPEECH\_RECOGN ITION\_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).

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).

## APPENDIX

