Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

Designing digital technologies to support language diversity preservation in Africa

Author: Antria Panayiotou (2735005)

1st supervisor:	Dr. Anna Bon
daily supervisor:	Mr. Francis Saa-Dittoh
2nd reader:	Prof. Dr. Hans Akkermans

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

August 2, 2023

"Unity is strength... when there is teamwork and collaboration, wonderful things can be achieved." , by Mattie Stepanek

Abstract

Context. In a globalised world, the preservation of indigenous languages and cultures holds significant importance. Numerous indigenous languages, such as the Dagbani language spoken in the northern region of Ghana, confront the peril of becoming extinct as a result of inadequate educational provisions and diminished levels of literacy. The process of digitising indigenous languages has significant potential in terms of preservation and revitalization.

Goal. The primary objective of this study is to create an innovative methodology for gathering data in the Dagbani language, which is an integral component of a broader research endeavour. The principal aim of this initiative is to enhance the agency of the Dagbani community, preserve their linguistic heritage, and address the disparity in digital access. The use of crowdsourced data will be employed in the training of a machine learning algorithm, facilitating the development of voice-to-text transcription and voice-based agricultural assistance.

Method. The development of a data collection platform has been achieved through a collaborative co-design process involving native speakers of the Dagbani language. The methodology takes into account the intricate nuances of language, cultural background, and aspirations of the Dagbani community. The compiled recordings will function as valuable assets for the purpose of training the machine learning model.

Results. The research project made significant strides in preserving linguistic diversity in environments with limited resources. The successful implementation of the Dagbani data collection platform resulted in the systematic collection of valuable corpus in native languages. In addition, the project advanced methods for enhancing audio quality, and insights gained through collaboration with the community and experts facilitated a better understanding of the difficulties and technical requirements of language preservation.

Conclusions. This research project highlights the significance of adopting a collaborative approach to protect linguistic diversity, especially in resource-constrained settings. The project's core focus lies in actively engaging indigenous speakers in the language preservation journey through an iterative methodology and strong partnerships with local stakeholders and communities. The knowledge gained from this endeavor serves as a valuable resource for future initiatives and enhances our understanding of the intricate cultural and technical aspects involved in language preservation. By prioritizing collaboration and community involvement, we pave the way for more effective and sustainable language revitalization efforts.

Acknowledgements

This individual thesis was prepared in partial fulfillment of the requirements for a Master's degree in Computer Science at the University of Amsterdam (UvA) and Vrije Universiteit Amsterdam (VU). However, the successful completion of this thesis would not have been possible without the invaluable assistance and support of several individuals.

First and foremost, I extend my sincere gratitude to my supervisor, Dr. Anna Bon, for her trust, her guidance, and the opportunity to work on this topic. Her continuous feedback and advice throughout the entire course were instrumental in shaping the outcome of this thesis. I am also grateful to Dr. Bon for facilitating interviews with key individuals, such as Mr. Francis Saa-Dittoh (the principal investigator of the project) and Mr. Baart, whose insights greatly enriched the content of this thesis. Furthermore, I would like to acknowledge the valuable feedback provided by Prof. Dr. Hans Akkermans.

I would like to express my gratitude to the individuals who dedicated their time to test and utilize the platforms developed in this research, offering valuable feedback that significantly contributed to the platform's improvement. Their participation and input are deeply appreciated.

In addition, I extend my heartfelt thanks to my family for their unwavering love, support, and encouragement throughout my years of study and beyond. Their presence has been a constant source of inspiration and motivation.

Lastly, I am immensely grateful to my close friends, who have shown patience, unwavering support and have been by my side every step of the way. Their friendship and encouragement have been invaluable on this journey.

Contents

List of Figures

LIST OF FIGURES

List of Tables

List of Acronyms

AI Artificial intelligence

API Application Programming Interface

APK Android Package Kit

ASR Automatic Speech Recognition

BSON Binary Javascript Object Notation

CTC Connectionist Temporal Classification

CLI Command-Line Interface

CMN Cepstral Mean Normalisation

CNN Convolutional Neural Network

DOM Document Object Model

DSP Digital Signal Processing

ERD Entity Relationship Diagram

FK Foreign Key

fMLLR Feature Space Maximum Likelihood Linear Regression

GAN Generative Adversarial Networks

GUI Graphical user interface

HIDP Human Integration Design Processes

HMM Hidden Markov Model

ICT Information and Communication Technology

ID identification

IDF Interaction Design Foundation

ISTFT Inverse Short-Time Fourier Transform

JS JavaScript

JSON JavaScript Object Notation

LSTM Long Short-Term Memory

MFCC Mel-Frequency Cepstral Coefficients

ML Machine learning

NASA National Aeronautics and Space Administration

NIST National Institute of Standards and Technology

NLP Natural Language Processing

NMT Neural Machine Translation

NPM Node Package Manager

OOV out-of-vocabulary

PK Primary Key

RNN-T Recurrent Neural Network Transducer

RQ Research Question

RIR Room Impulse Response

RNN Recurrent Neural Networks

RQ Research Question

SQL Structured Query Language

SSPL Server Side Public License

SNR Signal-to-Noise Ratio

STFT Short-Time Fourier Transform

TDNN Time-Delay Neural Networks

TTS text-to-speech

UI User Interface

VM Virtual Machine

W4RA Web alliance for Regreening in Africa

WAV Waveform Audio File Format

2NF 2nd Normal Form

3gp Third Generation for mobile Platform

1

Why preserving indigenous languages

In an ever-globalizing world marked by rapid technological advancements, the imperative to safeguard indigenous languages and cultures has grown exponentially. The languages in question, which are frequently spoken in areas characterised by limited access to educational resources and low levels of literacy, are confronted with the imminent risk of extinction. Consequently, the disappearance of these languages would result in the loss of invaluable knowledge, cultural legacy, and collective identity that have been accumulated over the course of centuries. Nevertheless, there is a growing recognition of the profound importance that indigenous languages possess, and the urgent imperative to allocate resources towards their preservation for the benefit of future generations.

Our study envisions a global scenario in which the recognition and appreciation of every individual's voice, narrative, and linguistic diversity are prioritised. With the aforementioned vision as the central focus, we have undertaken a transformative endeavour to develop a novel approach for gathering data in lesser-known indigenous languages. Our mission is to empower indigenous communities and safeguard their linguistic heritage through the implementation of collaborative co-design, rigorous testing, and iterative improvements.

The enduring impact of colonisation is evident in various countries located in the Global South, as the dominant use of European languages as a common means of communication has resulted in the marginalisation of indigenous languages [1]. However, it is our strong conviction that digital technologies have the potential to serve as powerful instruments for rejuvenation and fostering connectivity. Through the process of digitization, it becomes possible to overcome cultural barriers, enhance the representation of marginalised groups, and prevent the neglect of any language.

1. WHY PRESERVING INDIGENOUS LANGUAGES

The primary goal of our case study focuses on the Dagbani language, which is predominantly spoken by a dynamic community residing in the northern region of Ghana. By actively participating in discussions and interactions with individuals who are fluent in the Dagbani language, we have collectively devised a targeted crowdsourcing way for gathering data that effectively captures the intricate linguistic subtleties, cultural background, and aspirations of the Dagbani community. This process serves as evidence of the tenacity and resolve exhibited by individuals who are unwilling to let their language become forgotten.

This research is motivated by a profound recognition of the significance and diversity that every language brings to the intricate fabric of human civilization. Through the act of sharing our experiences and knowledge, our aim is to stimulate additional research and initiatives that advocate for the preservation and promotion of linguistic diversity and cultural heritage. In unison, let us acknowledge our shared obligation to guarantee the preservation of every language and to amplify and commemorate the perspectives of all communities, regardless of their size or marginalisation.

The objective of this paper is to enhance the influence of our research by disseminating our discoveries and extracting general principles that can be implemented in the preservation of minor indigenous languages globally. Our research endeavours aim to stimulate an extensive discussion regarding the significance of linguistic diversity, cultural preservation, and digital inclusion. The overarching objective is to stimulate a worldwide initiative that advocates for the preservation of linguistic and cultural rights, thereby cultivating an environment where diversity flourishes and all perspectives are duly acknowledged.

This thesis encompasses multiple chapters that explore the process of designing and executing a platform for gathering data in lesser-known indigenous languages. In Chapter 1, the research questions that underpin the study are presented, while Chapter 2 offers a comprehensive examination of the project's requirements and specifications. Chapters 3 and 4 are dedicated in a deeper examination of the design process, which includes the analysis of user profiles, system requirements, and crucial services. Chapter 5 delves into the execution of the platform, shows the features exhibited by the mobile and web applications. Finally, Chapter 6 provides an analysis of the applications, carefully analysing their efficacy and user-friendliness. In aggregate, these chapters provide a thorough analysis of the project's progression from its inception to its implementation and assessment, shedding light on the difficulties and prospects associated with the conservation of minor indigenous languages via digital methodologies.

1.1 Challenges in Indigenous Language Preservation

The endeavour to conserve autochthonous languages, such as Dagbani, is beset with a multitude of obstacles that arise from the distinctive circumstances under which these languages subsist. The aforementioned challenges stem from a confluence of factors, including but not limited to resource scarcity, the specificities of the local environment, and the prevailing socio-economic circumstances in the areas where these languages are spoken. The present subchapter examines the hindrances encountered in the conservation of indigenous languages, elucidating the unique characteristics of these difficulties and investigating potential approaches to surmount them.

1. **Constraints of the local context in Ghana:** The project may face particular limitations associated with the local context in Ghana. The aforementioned limitations may comprise of customary beliefs, communal frameworks, and restricted provisions that are exclusive to the locality. Comprehending and effectively managing these limitations is imperative for the prosperous creation and implementation of the application. Close collaboration with members of the Dagbani-speaking community in Ghana is imperative to ensure that the app is tailored to their unique needs and expectations, while also demonstrating respect for their cultural practises and incorporating their valuable insights.

2. **Problems with infrastructure and hardware in Ghana:** Ghana may encounter obstacles related to infrastructure and hardware deficiencies, particularly in remote or rural regions. This may entail inadequate internet connectivity that is not dependable, restricted availability of smartphones or computers, and obsolete or incompatible hardware. In order to surmount these obstacles, it is imperative to optimise the application to function without an internet connection or with limited bandwidth connectivity.

3. **Low literacy in Dagbani and other languages in Ghana:** The issue of low literacy in Dagbani and other languages in Ghana is a matter of concern. The issue of low literacy levels in Dagbani, as well as other languages utilised in Ghana, poses a considerable obstacle. The app's written components may pose a challenge to users with limited reading and writing proficiency, potentially hindering their ability to effectively participate. Dagbani is predominantly an oral language, and a considerable proportion of individuals within the Dagbani-speaking populace may exhibit limited proficiency in the language's written form. This presents a formidable

obstacle in constructing the complete system exclusively in Dagbani, as it would curtail the prospective user demographic. In order to promote greater inclusivity and engagement among users, it is imperative to integrate multilingual capabilities or offer assistance for multiple languages, such as English, to accommodate individuals with diverse levels of literacy in Dagbani.

4. **Changing requirements:** The phenomenon of changing requirements in Ghana may result in a disparity between the expectations and needs of users and the comprehension of the local context and Dagbani language by developers, thereby creating a gap between the two parties. Facilitating the bridging of this gap necessitates the proactive engagement of end-users, proficient speakers, and regional linguistic specialists throughout the developmental phase. Undertaking user research, collecting feedback, and establishing transparent communication channels with the Dagbani-speaking community in Ghana are essential steps towards ensuring that the app effectively meets their specific needs. Moreover, the utilisation of Agile methodology in the system's implementation facilitates efficient navigation and adaptation to changing requirements. Furthermore, consistent communication and cooperation with stakeholders and users in Ghana facilitate the adaptation of development initiatives to address changing requirements.

5. **Working at a distance in Ghana:** The execution and establishment of the project in Ghana may necessitate the cooperation of stakeholders, developers, and users who are situated in diverse geographical locations within the country. The practise of remote work may present certain difficulties pertaining to effective communication, coordination, and collaboration. The utilisation of online collaboration tools, implementation of frequent virtual meetings, and upholding transparent communication channels are imperative in surmounting these obstacles.

1.2 Research Questions

In today's interconnected world, preserving linguistic diversity in low-resource environments is a pressing challenge. Indigenous languages, which are repositories of irreplaceable cultural heritage and traditional knowledge, are threatened by globalisation and dominant language influences. As a result of this critical situation, researchers and communities are investigating how technology can be utilised as a potent instrument for preserving and revitalising endangered languages. This study is guided by the following research question.

RQ1: How can technology be leveraged to preserve linguistic diversity in low-resource environments? This comprehensive investigation explores the multifaceted role that technology can play in empowering communities to protect and promote their native languages.

Beyond mere communication, linguistic diversity is intrinsically linked to cultural identity and social cohesion within communities. To safeguard intangible cultural heritage and ensure the continuity of distinct knowledge systems, the preservation of indigenous languages becomes a crucial endeavour. However, environments with limited resources frequently face obstacles such as limited access to technological infrastructure and funding. In order to effectively address the challenges faced by these communities, it is necessary to investigate innovative and long-term methods of language preservation.

Through an examination of the intersection between technology and linguistic diversity, this study seeks to identify locally implementable solutions. This research has the potential to inform policymakers, linguists, and technology developers, inspiring them to prioritise the needs of low-resource communities and provide effective tools for the preservation of their languages.

To comprehensively address the primary research question, this study will focus on three subquestions, each of which will delve into crucial aspects of leveraging technology to preserve linguistic diversity in low-resource environments.

RQ1.1: How can data be collected for the preservation of indigenous languages with limited resources and knowledge? To respond to this subquestion, a mixed-methods strategy will be implemented. To gain an in-depth understanding of the local community's linguistic practises and crouwsourced preferences, ethnographic research will be conducted. In addition, techniques for crowdsourcing will be investigated to engage community members in the data collection process, ensuring their active participation and ownership of the linguistic data.

RQ1.2: What are the essential considerations for designing a language preservation system that caters to the needs and values of the local community?

To comprehensively address this subquestion, an iterative and participatory design strategy inspired by agile methodology will be utilised. Throughout the entire design process, there will be close collaboration with the local community, language experts, and technology developers.

Extensive qualitative research, including interviews and observations of comparable projects, will comprise the initial phase. This will ensure that the language preservation system is tailored to the community's specific needs and values.

As the project advances, iterative design cycles will be utilised to collect continuous feedback from the community's end-users. This feedback will be meticulously analysed and incorporated into subsequent design iterations to facilitate the co-creation of the language preservation system. By involving the community in decision-making processes, the project intends to foster a sense of ownership and empowerment among community members, thereby promoting the long-term viability of the technology beyond the research phase.

The agile design methodology will facilitate the identification of potential obstacles and limitations in real time, allowing for timely adjustments and enhancements. By continuously iterating the design based on user feedback, the language preservation system will adapt to the needs and values of the community over time.

RQ1.3: What are the constraints, obstacles, and technical requirements for the preservation of indigenous languages?

To respond to this subquestion, a multifaceted strategy will be implemented. In the beginning of the project, researchers will conduct qualitative interviews and focus group discussions with community members, language specialists, and technology developers to identify potential challenges and technical requirements.

In close collaboration with the community, iterative user testing and feedback collection will be conducted as the implementation of language preservation systems advances. Through application in the real world, the project team will gain invaluable insight into any unanticipated limitations and obstacles that may arise during the deployment of the system.

In addition, during the development and integration of language preservation technologies, the team will continuously analyse technical aspects and requirements to guarantee seamless functionality in environments with limited resources. The project will investigate solutions for issues such as language and audio cleaning, data storage optimisation, and overcoming potential bandwidth limitations.

The project aims to gain a comprehensive understanding of the limitations, challenges, and technical requirements involved in preserving indigenous languages through a participatory and dynamic research process. This knowledge will inform the refinement and optimisation of the language preservation system, making its implementation in low-resource

environments feasible and sustainable. Continuous community feedback and iterative improvements will foster a contextually relevant, culturally sensitive, and technologically effective co-created solution.

The current research question exemplifies the urgency and significance of preserving linguistic diversity in environments with limited resources. As we delve deeper into the role of technology in language preservation, we hope to contribute to a more inclusive and linguistically diverse global landscape in which indigenous languages can flourish and continue to enrich humanity's cultural tapestry. Exploration of the research subquestions will shed light on pragmatic approaches to preserving languages, empowering communities, and fostering linguistic heritage's sustainable future. This study aims to have a lasting impact on language preservation practises in low-resource environments through the use of collaborative efforts and innovative methodologies.

In the pursuit of understanding and contributing to the preservation of indigenous languages, this master's thesis has been an exceptional journey, largely made possible through the invaluable opportunity presented by the TIBaLLi project. Throughout this research endeavor, the exploration of existing literature has paved the way for a groundbreaking approach to language development and audio cleaning. As a testament to the significance of this work, one of the notable contributions includes the publication of a scientific paper in the esteemed "8th Edition ICT4SD International ICT Summit & Awards" conference, published by Springer[1]. The published paper served as a platform to introduce and address the challenges faced by indigenous languages, further reinforcing the importance of language revitalization efforts.

An important fact to highlight is that the code for this project has been effectively implemented and showcased at a distinguished conference in Uganda. Additionally, to foster collaboration and open access, the codebase has been made available as an open-source repository on GitHub[2]. Interested parties are encouraged to get in touch via the provided contact email for further inquiries and collaborations.

In conclusion, the above has provided an overview of our mission to preserve the Dagbani language and has outlined the research questions guiding our efforts. The next chapter, "Background of this research," will delve into the TIBaLLi Research Framework, project requirements, and the importance of our chosen use case. This foundation will shape the subsequent design, development, and future explorations of our language preservation platform.

[1] https://w4ra.org/wp-content/uploads/2023/05/Antria_et_al.pdf
[2] https://github.com/AntriaPan/TIBaLLi-project-voice-services

2

Background of this research

This chapter presents a comprehensive overview of the project, encompassing the requirements, specifications, significance of implementing the use case, and the challenges faced in the preservation of indigenous languages. In adddition explores the precise details and factors that influenced the development and execution of the project. The initial focus of our analysis centres on the requirements and specifications that informed the design and functionality of the system. Finally, this entails an analysis of the distinct requirements and ambitions of the specific demographic being studied.

2.1 Meeting TIBaLLi Research Framework

This master's research is conducted within the framework of an existing Internet Society-led research project and in collaboration with a dedicated team[1]. "Tiballi" is a Dabgani word that means "our language". This comprehensive project includes this thesis as one of its essential deliverables. The overarching objective of the research project is to address the critical problem of language preservation and promote internet accessibility in low-resource environments, with a particular emphasis on communities in the Global South.

The primary objective of the project is to empower indigenous languages by making Internet-based information more accessible and relevant to the target communities. The "Tiballi Project" investigated the feasibility of reconstructing advanced Artificial intelligence (AI) methods, such as Machine learning (ML) and Natural Language Processing (NLP), to enable the sharing of local weather data and global climate information in people's native languages, thereby transcending the current boundaries of the internet.

[1] https://tiballi.net

Several components and packages have been developed within the project to realise this vision. These include techniques for creating language corpora, data augmentation to address the challenges of small training sets, and automatic word discovery and cross-validation experiments.

In this context, the master's thesis contributes significantly to the project's goals. Through the thesis, a larger language corpus centred on climate-related information is compiled. This thesis aims to contribute to the project's mission of preserving indigenous languages, fostering inclusivity, and expanding access to internet-based information for underserved communities in the Global South through its inclusion in the larger research project.

2.2 Requirements and specifications

The operationalization employed in the Tiballi project deviates from variable-based hypothesis testing. Instead, it focuses on the implementation of meticulously designed field experiments, conducted in conjunction with local communities, partners, and other stakeholders involved in rural development initiatives in Africa. The field experiments are broken down and operationalized into Work Packages, as shown in Figure 2.1.

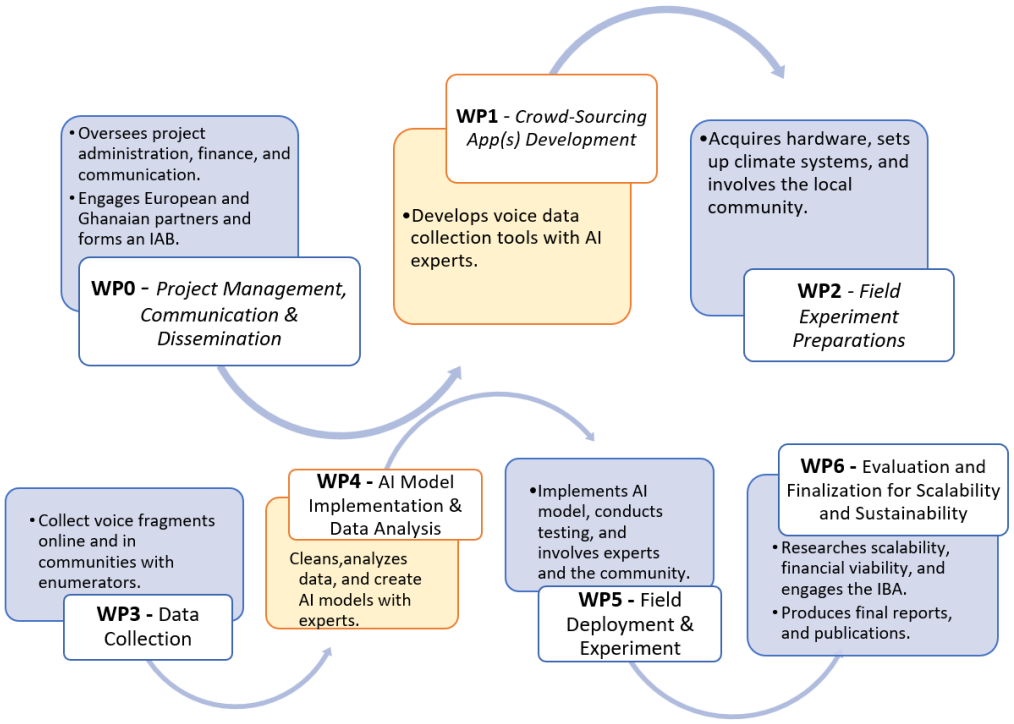


Figure 2.1: Work Packages of Tiballi project.

This document specifically focuses on the areas highlighted in yellow in Figure 2.1, which are part of the Tiballi project's Work Package 1 and Work Package 4. Work Package 1 involves the design and implementation of the web and mobile applications, while Work Package 4 entails the cleaning and enhancement of the collected data. These highlighted areas represent the key aspects of the project that are explored and analyzed in detail throughout the thesis.

The use case diagram, shown in Figure 2.2, illustrates the interaction between the Farmer, the Voice-based Information and Communication Technology (ICT) System, the External Application Programming Interface (API), and the phone call. The system is represented as a rectangle labeled "System," which encompasses the three primary use cases: "Access Weather Information," "Get Crop Planting Guidance," and "Input Recent Rainfall Data."

The interaction between the Farmer and the Voice-based ICT System proceeds as follows:

"Access Weather Information": The Farmer makes a phone call to the Voice-based ICT System and follows the prompts to select the option for accessing weather information. The Farmer presses the appropriate numbers to make the selection. The Voice-based ICT System then interacts with the External API to fetch real-time weather data. The weather information is delivered to the Farmer via the phone call.

"Get Crop Planting Guidance": The Farmer contacts the Voice-based ICT System via phone call and selects the option for crop planting guidance by pressing the appropriate numbers. The Voice-based ICT System provides the necessary guidance based on the selection, and the information is communicated to the Farmer through the phone call.

"Input Recent Rainfall Data": The Farmer dials the phone number of the Voice-based ICT System and follows the prompts to select the option for inputting recent rainfall data. The Farmer presses the appropriate numbers to make the selection. The Voice-based ICT System prompts the Farmer to provide the data using predefined categories. The Farmer inputs the rainfall data via the phone call, and the Voice-based ICT System acknowledges and stores the data for further analysis.

Additionally, the Voice-based ICT System interacts with the External API to fetch global climate information when required by the use cases.

Real-Time Rainfall Updates:

Example Question: Will it rain in the next couple of hours, today? The system can provide real-time updates on rainfall, helping farmers plan their activities accordingly.

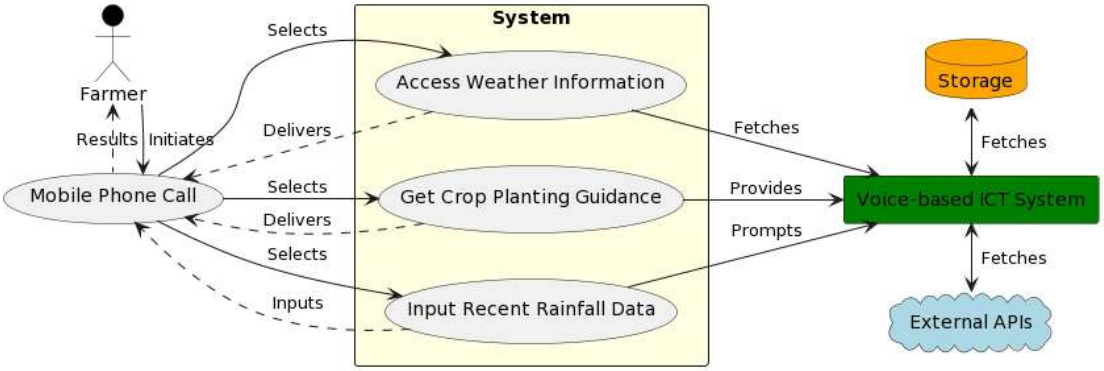


Figure 2.2: Use case diagram for Tiballi's project.

Planting Recommendations:

Example Question: What can I plant based on the previous rainfall? The system can suggest suitable crops for planting based on historical rainfall data, optimizing agricultural choices.

Optimal Planting Timing:

Example Question: What is the best time to plant to achieve maximum yield? The system can provide insights into the optimal planting time for different crops, maximizing harvest outcomes.

Climate Information Accuracy:

Example Question: How reliable are current predictions about rainfall? The system can address the farmers' concerns by providing information about the accuracy of climate predictions, fostering informed decision-making.

Animal Health and Care:

Example Question: How can I ensure the health and well-being of guinea fowl, chicken, and goats? The system can offer guidance on animal health, prevention, and treatment practices, aiding farmers in caring for their livestock.

2.2.1 Importance of the implementation of the use case

The implementation of the Tiballi system bears considerable significance for the agricultural sector in the northern region of Ghana. The objective of this system is to mitigate the difficulties encountered by local farmers in obtaining precise and prompt meteorological information during the pivotal rainy season, by means of cooperative initiatives and

acknowledgement of their requirements. The Tiballi project aims to equip farmers with crucial information for efficient rain-fed agriculture practises by merging regional weather data with online global climate information. Let us now examine the three pivotal aspects that underscore the importance of this implementation.

1. **Local farmers expressed the need for better access to meteorological data during the rainy season:** The requirement for improved accessibility to meteorological information during the wet season was articulated by nearby agricultural producers. Rain-fed agriculture is the predominant agricultural practise in northern Ghana, with crop yields and farming activities being significantly influenced by the availability and distribution of rainfall. The agricultural community has raised apprehensions and emphasised the necessity for enhanced availability of precise and punctual meteorological information, particularly in the wet season. Accessing this data would facilitate the farmers in efficiently strategizing their agricultural operations, including ascertaining the most suitable time for sowing, regulating irrigation, and executing relevant pest and disease management practises.

2. **Current weather data is unavailable, hindering rain-fed agriculture practices:** The absence of up-to-date meteorological information is impeding the implementation of rain-dependent farming techniques. The absence of dependable meteorological information poses a significant obstacle for indigenous agriculturalists in the northern region of Ghana. The existing meteorological information systems frequently exhibit inadequacies in furnishing precise and current weather predictions that are customised to the particular locality. The insufficiency of dependable meteorological information impedes the farmers' capacity to make knowledgeable determinations concerning their farming methodologies. The lack of prompt access to climatic data such as rainfall patterns, temperature fluctuations, and other related factors poses a challenge to farmers in effectively strategizing their farming activities, which may lead to decreased productivity and potential crop damage.

3. **Collaborative meetings with farmers confirmed the lack of useful weather information:** In order to understand the needs and challenges faced by local farmers, collaborative meetings were conducted with the farming community in northern Ghana. The meetings enabled farmers to voice their concerns and share their practical knowledge about the platform. The aforementioned interactions revealed a prevalent challenge encountered by farmers, namely the insufficiency of pertinent

meteorological data. The individual conveyed their discontent regarding the present condition of meteorological data accessibility and underscored its adverse influence on their agricultural methodologies. The joint gatherings provided confirmation of the necessity to tackle this crucial deficiency and devise a remedy that furnishes dependable, site-specific meteorological data to enable farmers and amplify their farming efficiency.

The Tiballi project endeavours to narrow the disparity in the accessibility of meteorological data, furnish local farmers with precise weather information, and bolster their rain-fed agriculture practises in the northern region of Ghana by focusing on these three key areas.

Having explored the essential background of the research, including the TIBaLLi Research Framework, project requirements, and the significance of the chosen use case, the next chapter, titled "Design", delves into the creation of user profiles and personas. It provides an in-depth analysis of user scenarios and functionality overview, along with outlining the system requirements to ensure a comprehensive understanding of the foundation for the language preservation platform.

3

User-Centric Design and System Requirements

This chapter provides a thorough analysis of the design elements and factors that influenced the project, incorporating a stakeholder analysis through user profiles and personas. It encompasses multiple subsections, including system requirements, methodology, and sequential stages involved in the design process. Additionally, the chapter explores essential services and data management aspects, introducing "Dagbani Speak," a platform dedicated to fostering community engagement. The primary objective of this chapter is to offer valuable insights into the design decisions and principles that contributed to the development of a user-friendly and aesthetically pleasing platform.

3.1 User Profiles - Personas

In this subsection, we conduct a stakeholder analysis through comprehensive personas representing the primary users of the Dagbani Speak platform, including community members, developers, and administrators. By analyzing their backgrounds, objectives, obstacles, and incentives, we gain valuable insights into their distinct viewpoints and requirements. Understanding these personas is crucial in developing a user-centric platform that effectively addresses the needs and aspirations of its users. The objective is to provide insight into the diverse stakeholders comprising the Dagbani Speak ecosystem and underscore their crucial contributions to propelling the platform's achievements.

1. A resident of a rural community in Northern Ghana, with limited access to resources and technology infrastructure.

Persona's Name: Abena
Demographics: Female, 28 years old , Farmer
Background:
- Lives in a rural community in Northern Ghana
- Typically from low-income background
- Resides in remote areas with limited connectivity and access to modern amenities

Digital Literacy: Basic or limited digital literacy skills, with little exposure to advanced technology.
Goals: Contribute valuable weather data for agricultural planning

Figure 3.1: Photo of persona Abena (Farmer)

Challenges:

1. Unreliable internet connectivity
2. Limited access to smartphones or computers
3. Lack of familiarity with digital platforms

Motivation: Improve farming practices and access relevant information for sustainable agriculture.

---

2. A member of an urban community in Northern Ghana with moderate digital literacy skills and access to resources.

Persona's Name: Kwame
Demographics: Male, 42 years old, Teacher
Background:
- Urban dweller in Northern Ghana
- Varied socio-economic background
- Resides in urban areas with improved connectivity and access to technology

Digital Literacy: Moderate to advanced digital literacy skills, familiar with smartphones, computers, and internet usage.
Goals: Promote community engagement, share local knowledge

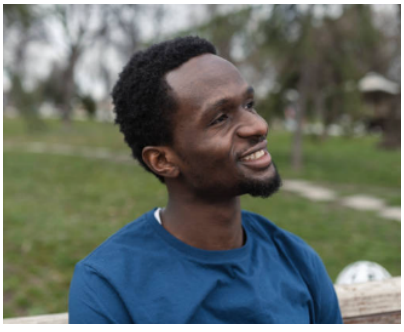


Figure 3.2: Photo of persona Kwame (Teacher)

Challenges:

1. Data affordability
2. Language barriers
3. Limited access to reliable internet connections

Motivation: Interested in engaging with digital platforms for personal and professional growth, accessing information, and connecting with a wider community.

---

3. The owner and developer of the Dagbani Speak platform, responsible for designing, implementing, and analyzing the system.

Persona's Name: Daniel
Demographics: Male, 35 years old, Software Developer
Background: - Urban dweller in Accra, Ghana
- Varied socio-economic background
- Resides in urban areas with improved connectivity and access to technology
Digital Literacy: Moderate to advanced digital literacy skills, familiar with smartphones, computers, and internet usage.
Goals: (1) Enhance the functionality and usability of the Dagbani Speak platform (2) Improve data analysis and insights for decision-making

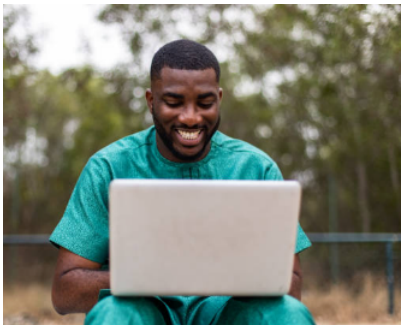


Figure 3.3: Photo of persona Daniel (Software Developer)

Challenges:

1. Balancing technical complexities with user-friendly design
2. Ensuring that all user needs are implemented and compatibility across different devices and platforms

Motivation: (1) Creating an innovative and impactful crowdsourcing platform
(2) Empowering communities through technology and data-driven solutions

---

3.1.1 User Scenarios and Functionality Overview

The utilisation case diagram for our crowdsourcing application is illustrated in Figure 3.4, showcasing the diverse interactions and functionalities accessible to distinct user personas. This diagram showcases two pivotal scenarios: Scenario 1, which portrays the viewpoint of a typical Dagbani user, personified by the persona Adena, and Scenario 2, which centres on the perspective of a developer, embodied by the persona Daniel. The use case diagram offers a graphical depiction of the potential actions and functionalities that are available to each category of users. It effectively demonstrates how our application addresses the distinct requirements and responsibilities of various stakeholders.

Use case 1: User perspective (Dagbani speaker - Persona: Adena or Kwame)

User persona: A Dagbani speaker who wants to contribute to the community by recording their voice and helping to create a knowledge base for a machine learning program.

Scenario 1: To utilise the crowdsourcing app, user must first download it on their smartphones. Subsequently, they should navigate to the recording page, where a list of Dagbani language words to be recorded is displayed. Upon selecting a word, the user should click on the record button and say the word into their smartphone's microphone. After recording, the user should listen to the playback to ensure its clarity and audibility. Finally, the user must click on the "Submit" button to save the recording on their mobile device. For uploading, the user should proceed to the second screen and select the recording file they wish to upload. The upload process should be initiated, and the user should receive a pop-up message indicating the successful or unsuccessful upload of the recording file. This process can be repeated as many times as necessary, depending on the user's availability and interest.

Use case 2: Analyst perspective (Developer - Persona: Daniel)

User persona: The analyst of the crowdsourcing app who wants to monitor the contributions and performance of the app.

Scenario 2: To access the cloud service, on Firebase, the analyst must log in using their email and password. Once authenticated, they should navigate to the "Storage"

option from the menu, where they will find an overview of the uploaded recordings. The analyst can also execute javascript scripts to obtain additional information such as the total number of recordings uploaded by users, uploaded dates, and contributor names. Furthermore, the analyst can filter and view the most popular words recorded in the Dagbani language. Additionally, they can export the data to create reports and gain insights for stakeholders. Based on the data obtained from the dashboard, the analyst can take appropriate actions to enhance the application's performance and user experience.

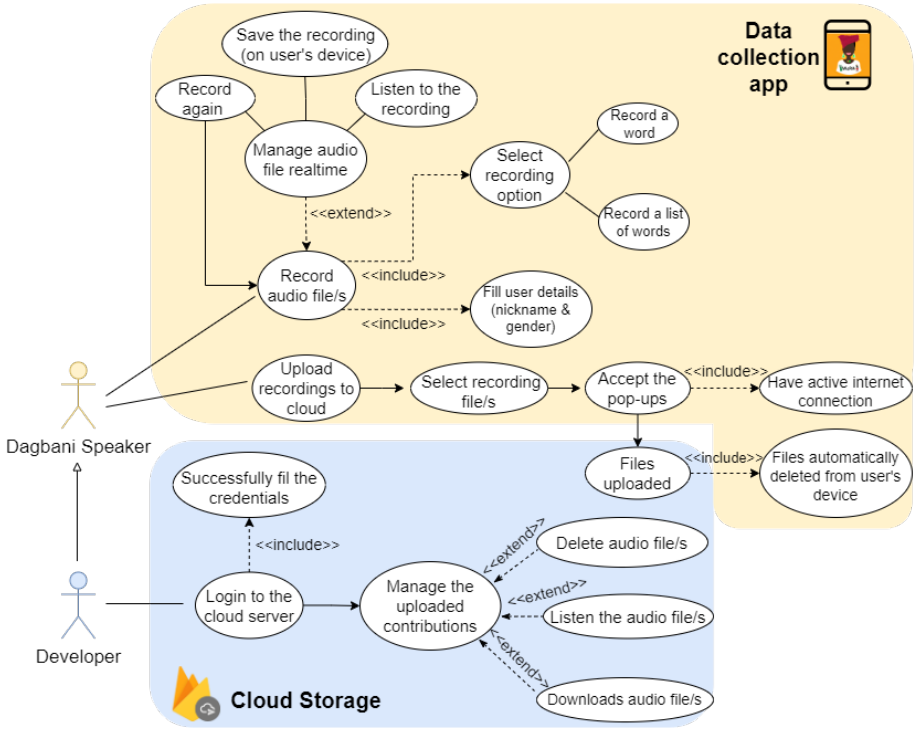


Figure 3.4: Use case scenarios for the personas.

3.2 System Requirements

The development of a crowdsourcing application for Dagbani in Ghana is deemed essential to tackle the challenges arising from its unique use case. The present chapter delineates the functional and non-functional prerequisites that serve as the underpinnings for the app's design and operation, guaranteeing inclusivity, efficacy, and cultural appropriateness. The app's functional requirements delineate the fundamental characteristics and

capabilities that empower users to document and make contributions to establish a repository of knowledge for the Dagbani language. The subsequent enumeration will elucidate the significance of each functional requirement and its implementation in the application to guarantee a resilient and user-centric encounter.

1. **Offline Capability:** The provision of offline functionality is of utmost importance for users residing in regions with restricted internet connectivity. The feature guarantees uninterrupted usage of the application and enables audio recording in the absence of an active internet connection. This characteristic fosters inclusiveness and enables individuals situated in distant areas to make contributions to the repository of knowledge.
   Inclusion in the system: The concept of inclusion within a system. Incorporate local storage functionality into the application to enable end-users to capture and store audio recordings on their device. Implement a queuing mechanism that enables the automatic transfer of stored audio recordings upon the establishment of an internet connection.

2. **Resource-Friendly Design:** The development of a resource-friendly design is crucial in ensuring optimal performance on low-end devices that are frequently utilised in Ghana. This involves the creation of a lightweight application that consumes minimal system resources. The optimisation of the app guarantees a seamless user experience for individuals with restricted device capabilities, thereby mitigating any potential lag or performance-related concerns.
   Inclusion in the system: To enhance the performance of the application, it is recommended to optimise the codebase, eliminate superfluous animations or intricate graphics, and minimise the memory footprint of the application. It is recommended to prioritise efficiency and take into account device-specific optimisations in order to improve the performance of the application on low-end devices.

3. **Multilingual Support:** The provision of multilingual support, including the local language of Dagbani alongside English, fosters greater inclusivity and participation among diverse linguistic communities in Ghana. The application enables users to interact with it and access information more efficiently in their chosen language, thereby fostering cultural diversity and augmenting the user experience.
   Inclusion in the system: Incorporate a dropdown menu within the application interface that offers users the option to select their preferred language for pronunciation, with two distinct language choices available.

4. **Audio Recording and Submission:** The provision of a user-friendly interface for audio recording in Dagbani is of utmost importance to facilitate the ease of contribution of language data by users. The technology allows for precise transcription of oral language and expressions, which can be utilised to improve the training of language models.
   Inclusion in the system: Develop a user-friendly recording interface that facilitates the audio capturing process by providing intuitive guidance to users. Incorporate functionalities such as a timer for recording, visual indicators for optimal recording levels, and the capability to review and modify recordings prior to submission.

5. **Data Storage and Management:** The development of a resilient system that ensures the safekeeping and organisation of audio data is of utmost importance to maintain data integrity, facilitate accessibility, and enable scalability. The process guarantees that the gathered audio recordings are systematically arranged, securely stored, and readily accessible for subsequent analysis and training of models.
   Inclusion in the system: Employing Firebase Cloud Storage or analogous services is recommended for the secure storage of audio data obtained from users. It is imperative to establish appropriate organisational and backup protocols to ensure that data is stored in a structured format that facilitates efficient retrieval. It is advisable to implement access controls and encryption methodologies in order to safeguard user data.

6. **Language Support and Localization:** The provision of language support and localization in Dagbani is of utmost importance to adequately address the language requirements of Dagbani-speaking individuals. The incorporation of this feature improves the comprehension, involvement, and acceptance of the application, thereby increasing its accessibility and user-friendliness.
   Inclusion in the system: The task at hand involves the development of interfaces, instructions, and error messages that are tailored to the Dagbani language and culture. It is imperative to establish a precise correlation between the content and user-facing components of the application, while also ensuring that they are culturally suitable and aligned with the language and context of Dagbani-speaking users.

Apart from the functional requirements, the non-functional requirements concentrate on facets that augment the holistic user experience and guarantee the efficacy and durability of the application. The subsequent enumeration delves into the rationale behind the

importance of each non-functional requirement and expounds on its implementation in the application to promote cultural sensitivity, collaboration, user support, performance, usability, and reliability.

1. **Cultural Sensitivity:** The significance of incorporating cultural sensitivity in the app development process cannot be overstated, as it is imperative to demonstrate reverence and safeguard the cultural conventions, sensibilities, and customs of the Dagbani-speaking populace in Ghana. The implementation of this measure guarantees that the application conforms to the users' principles and promotes a feeling of inclusiveness and confidence within the user community.
   Inclusion in the system: Undertake comprehensive research and engage in consultations with experts and community representatives in the area to gain a deeper understanding of cultural subtleties and sensitivities. Integrate suitable visual representations, symbolic elements, and linguistic expressions into the application that accurately depict and honour the cultural milieu of the Dagbani-speaking populace.

2. **User Support and Training:** The provision of extensive user documentation, tutorials, and support channels is imperative in aiding users with technical difficulties and guaranteeing their proficient utilisation of the application. The application endows its users with the essential knowledge and resources to effectively participate in the platform and provide their significant linguistic data.
   Inclusion in the system: Produce documentation and instructional materials that are easily understandable by users, elucidating the features of the application, providing guidance on audio recording techniques, and outlining optimal approaches. Establishing convenient support channels, such as frequently asked questions (FAQs), in-app messaging, or specialised customer support, can effectively handle user inquiries and offer prompt assistance.

3. **Performance and Scalability:** The significance of performance and scalability in computing systems. It is imperative to guarantee optimal performance of the application across diverse usage scenarios and to accommodate escalating user traffic and data volume to ensure a seamless and effective user experience. The application's ability to manage an expanding user population and changing requirements while maintaining optimal performance is facilitated by this feature.
   Inclusion in the system: Perform comprehensive performance testing to detect and enhance any bottlenecks or performance concerns. To manage growing user traffic

and data storage needs, it is recommended to incorporate effective coding practises, caching techniques, and expandable infrastructure. It is recommended to monitor and analyse the performance of applications in order to preemptively address any potential degradation in performance.

4. **Usability and User Experience:** The significance of usability and user experience in the field of design and technology cannot be overstated. The creation of an intuitive and user-friendly interface is essential to cater to users with varying degrees of technical expertise and linguistic abilities. The optimisation of user engagement, satisfaction, and adoption of the application is instrumental in promoting a smooth and uninterrupted user experience.
   Inclusion in the system: Performing user research and usability testing is crucial in comprehending user inclinations, actions, and difficulties. The application should integrate principles of user-centered design, employ intuitive navigation, and provide clear instructions to enhance its usability. It is recommended to frequently collect user feedback and engage in iterative design processes to consistently enhance the usability and overall user experience of the interface.

5. **Reliability and Availability:** The maintenance of user trust and consistent usage is contingent upon the assurance of app reliability and availability. Ensuring a continuous and uninterrupted user experience can be achieved through the implementation of efficient error handling and recovery mechanisms, as well as the minimization of downtime.
   Inclusion in the system: It is advisable to incorporate a resilient error handling and exception management system to effectively manage unforeseen circumstances in a graceful manner. It is recommended to consistently monitor the application's accessibility and efficiency, and establish contingency plans for data backup and disaster recovery. It is recommended to furnish users with precise and instructive error messages to assist them in resolving any problems that may arise.

6. **Collaboration with Local Organizations:** Engaging in partnerships with indigenous language conservation and cultural institutions in Ghana offers significant opportunities for community participation and assistance. The establishment of partnerships can potentially augment the credibility, reach, and efficacy of the application by utilising the ability and resources of said organisations.
   Inclusion in the system: It is recommended to actively involve local organisations in

the process of app development and deployment by seeking their input, guidance, and participation. Engage in collaborative efforts, such as content generation, linguistic validation, outreach to the community, and user involvement, to ensure that the application is in line with the requirements of the local populace and receives support from the community.

3.3 Technologies and Toolkits Employed in the Project

The section provides an insight into the technical components of the project. The document encompasses the software development tools utilised and emphasises the programming languages and technologies employed. This part of the paper offers significant insights into the technical underpinnings that facilitated the effective execution of the project.

3.3.1 Development Software

1. NPM

Figure 3.5: NPM logo.

Node Package Manager (NPM) is the name of the JavaScript programming language's plugin management[1]. It is primarily supported by developers who work with open-source software, who frequently distribute and utilize open-source applications or libraries that add functionality to the project they are working on. It consists of three distinct components: a web page, a Command-Line Interface (CLI) that communicates with the remote index, and a data file. When utilizing this library, a package.json file containing the names and versions of all installed packages will be generated. As a result, they can be upgraded automatically without any additional effort from the developer. Using a single command to install all required files and libraries for a program makes it incredibly easy to utilize this file.

[1] https://www.npmjs.com

Figure 3.6: Visual studio code logo.

2.Visual Studio Code

Microsoft developed Visual Studio Code, a free code editor that supports a wide variety of programming languages for both the graphical user interface and the background programming language. It is a development tool for web and cloud applications that is compatible with Windows, Linux, and macOS. It can be utilized to write code in numerous programming languages, such as Python, JavaScript, PHP, C, etc. It enables debugging, Git and Github auditing, simple code refactoring, and intelligent code completion for more efficient and straightforward programming. In addition to supporting Web application development and providing a class designer, it also provides a database schema designer. There is the option to add specific plug-ins, which can increase the developer's overall productivity. In the complete toolbox, which contains CBM boxes, radio buttons, buttons, text boxes, etc., we can find a variety of tools to create the system's user interface much more quickly. In addition, this tool enables the use of a number of system-user-familiar features that are comparable to those of the operating system (in this case, Windows).

3.3.2 Programming Languages and Technologies

1. HTML

Figure 3.7: HTML logo.

The syntax used for hypertext HTML, which stands for Hyper Text Markup Language, is the primary markup language for many websites. Typically, its structure is defined by fundamental structural elements in the form of tags. Web browsers are responsible for reading HTML files, establishing the structure of each page, and displaying the content in place of the tags. It allows the embedding of images, videos, and other media that can be used to display interactive forms on a webpage. By incorporating the technologies described below (such as CSS and JavaScript), it is possible to make a page dynamic, as opposed to a page whose format is based solely on HTML.

Figure 3.8: CSS logo.

2. CSS

CSS, or Cascading Style Sheets, is a technology that allows you to control how each page appears to site visitors. This is accomplished by formatting the various HTML elements, saving designers and developers time and effort by reducing the amount of work required due to the reusability of formatting in multiple locations. HTML 3.2 introduced color elements and additional HTML tags that produce effects comparable to CSS. Each subpage of a page should use the same font, color scheme, and text-to-content ratio, making CSS particularly useful. As demonstrated above, the separate CSS file that has existed since HTML 4.0 can alter the formatting of the entire web application at once.

3. JavaScript

Figure 3.9: HTML logo.

The original purpose of JavaScript (JS) was to manage client-side scripts, enable asynchronous data exchange, and transform static (HTML-only) content into context-sensitive dynamic content. It is a scripting and programming language that heavily relies on prototypes (prototype-based). Its syntax is clearly influenced by C, but its names and naming conventions are extremely similar to Java's. Contrary to popular belief, which is influenced by the similarity of their names, the semantic relationship between the two languages is nonexistent. Similar to the programming languages Self and Scheme, it supports object-oriented, prescriptive, and functional programming styles. PDF documents, site-specific browsers, small desktop applications (desktop widgets), and more recent VM and development frameworks, such as Node.js, are compatible with JavaScript.

Figure 3.10: Material UI logo.

4. Material UI

Material UI (User Interface) is a set of design rules that Google presented at a conference on June 25, 2014. It is a robust React JS component library that provides a beautiful application with editable premade and custom components. In general, it is an open-source library containing numerous automated components that offer presentable color and page structure formatting. Material UI is utilized by hundreds of thousands of developers, and well-known companies such as Netflix, Amazon, Unity, Spotify, etc., base their applications on its design.

5. Node JS

Figure 3.11: Node JS logo.

Node JS is a software development platform primarily for servers, so it is primarily a JavaScript-based back-end. Node JS aims to make it easier for developers to create scalable web applications; it employs the asynchronous I/O communication model, which is still a challenge in the majority of modern network application development environments. It was initially released by Ryan Lienhart Dahl in May 2009, is licensed under the MIT Software License, and is written in C++ and JavaScript.

As shown in the image below, Figure 3.12, Node JS is one of the most popular and widely used technologies among a variety of developers due to its numerous and significant advantages. As it is based on JavaScript, it is very simple to learn, and with basic program-

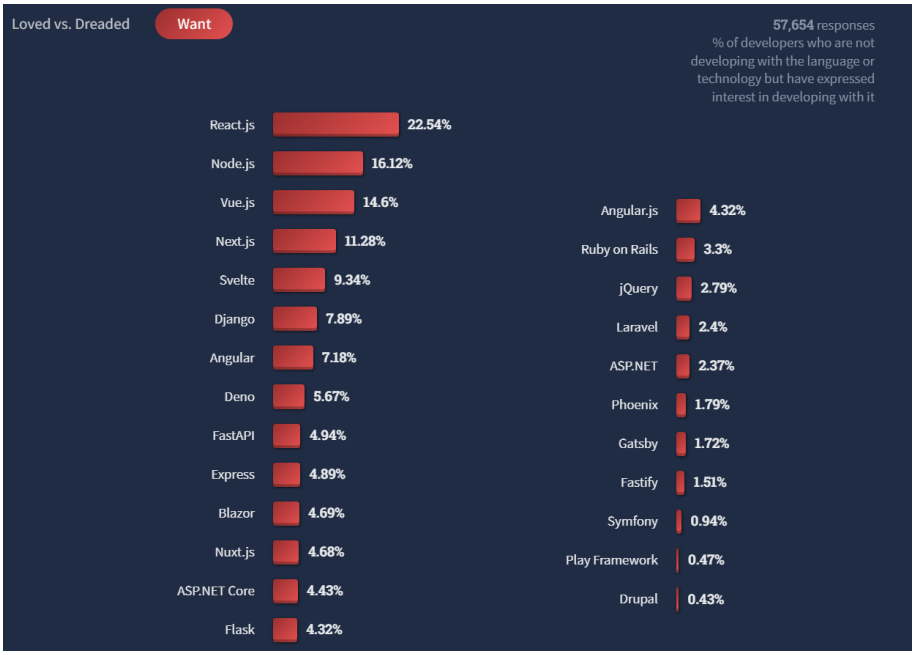


Figure 3.12: Developer Survey 2022 of Stack Overflow with 57,654 responses

ming knowledge, it simplifies things considerably. In addition, the fact that it is open source encourages support and contributions from various developers in order to enhance it and create additional reusable components and resources (there are more than 650,000 free code packages). It is also essential to note that the application's high speed and performance are guaranteed, particularly for real-time applications such as our platform.

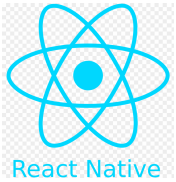


Figure 3.13: React Native logo.

6. React Native

It is an open-source JavaScript-based framework created by Facebook to fulfill the increasing demand for its mobile services. It is used to build applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows, and UWP by letting developers to leverage the React framework in conjunction with native platform features. Additionally, Oculus uses it to construct virtual reality apps. React Native is a hybrid mobile application framework that enables the development of mobile apps from a single codebase. This JavaScript framework enables the development of mobile applications that render natively across several platforms, including iOS and Android. a single framework allows the development of mobile applications for both iOS and Android.

React Native's operational concepts are nearly identical to those of React, with the exception that React Native does not change the Document Object Model (DOM) via the Virtual DOM. It operates as a background process directly on the end device and connects with the native platform across an asynchronous and batched bridge using serialized data. React components encapsulate native code and communicate with native APIs using JavaScript and the declarative User Interface (UI) paradigm of React.

7. Expo CLI

Figure 3.14: Expo CLI logo.

The expo package includes npx expo, a tiny and potent CLI utility designed to keep you moving quickly throughout application development. The prebuild mechanism provided by Expo CLI builds the project's native code. Expo CLI is built on top of React Native and is the quickest method to start up a React Native zip project. Simply create the project and begin coding. Expo is separated into two sections. Expo CLI is a command line interface used by developers to construct, execute, publish, etc. applications. On the other side, the Expo Client App is an Android and iOS mobile application that enables you to visually execute the program on a physical device. Expo CLI makes it simple to start up the React project. Expo incorporates its own collection of fundamental libraries for a typical project, including push alerts, asset managers, and so on.

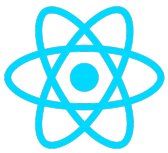


Figure 3.15: React JS logo.

8. React JS

React JS is an open-source front-end JavaScript library used primarily for UI and UI component development. It was introduced by Jordan Walke in May 2013 and is written in JavaScript and distributed under the MIT Software License. React's main goal is to develop a page or mobile app and manage its state and rendering in the DOM. There is a possibility that the addition of additional libraries and routing will increase the application's size and complexity. React simplifies the creation of interactive user interfaces because it can easily and efficiently update the appropriate elements when program data changes. In addition, as a declarative language, it enables the user to find errors with greater ease and makes the code more predictable. In addition, it is component-based, and since the component logic is based on JavaScript, it is possible to very easily move and reuse encapsulated elements in and out of the DOM.

10. Expo Go

Expo Go is a mobile application developed by Expo that enables you to launch and test React Native applications without the need to build or install them separately on physical devices. Expo is a free and open-source platform that streamlines the process of developing React Native cross-platform mobile applications.

Figure 3.16: Expo Go logo.

Expo Go enables developers to rapidly preview and interact with React Native applications on iOS and Android devices. It offers a practical method for testing app functionality, design, and performance in real-time, allowing developers to iterate and improve more efficiently. Expo Go also provides access to device features such as the camera, accelerometer, and location, allowing you to test the incorporation of your app with these capabilities during development.

Figure 3.17: Firebase logo.

9. Firebase

Google offers a web-based interface called the Firebase Console for managing Firebase projects. Powerful platform Firebase provides a range of tools and services for creating, growing, and managing web and mobile apps. Developers may configure and manage a variety of Firebase project components with the Firebase Console, including authentication, real-time databases, cloud storage, hosting, and more.

Developers can set up authentication options including email/password, social logins (such as Google, Facebook, etc.), and third-party providers through the Firebase Console. Additionally, they may set up cloud functions to handle server-side logic, establish and maintain the database layout, and setup numerous analytics and performance monitoring capabilities.

11. Python

Figure 3.18: Python logo.

Python is an interpreted, high-level programming language that is known for its simplicity and intelligibility. It was developed by Guido van Rossum and published for the first time in 1991. Due to its adaptability and extensive library and framework support, Python has acquired tremendous popularity among developers.

Python prioritizes code readability, which makes it simpler to write and maintain. It supports procedural, object-oriented, and functional programming styles, among others. Python's extensive standard library provides ready-to-use modules for file management, network communication, and data manipulation, among other duties.

Python is used extensively in a variety of fields, including web development, data analysis, scientific computation, machine learning, and artificial intelligence. Its ecosystem consists of well-known frameworks such as Django and Flask for web development, NumPy and pandas for data manipulation, TensorFlow and PyTorch for machine learning, and many others.

The "Design" chapter concludes with an exhaustive exploration of user profiles, personas, scenarios, and system requirements. Moving forward, the "Methodology" chapter unveils the intricate system architecture, followed by a step-by-step account of the design process, encompassing ideation, sketching, prototyping, and user testing. Additionally, the chapter introduces "Dagbani Speak," a community engagement platform characterized by thoughtful design elements, shaping its unique brand identity.

4

Methodology: Building the Language Preservation Systems

This section will provide a thorough analysis of the design process that led to the system's final version. The project utilises a methodology that integrates the waterfall, agile, and user-centered design approaches. This methodology has been specifically tailored for the development of a targeted crowdsourcing application that aims to safeguard and advance small indigenous languages, with a particular emphasis on Dagbani.

The project was initiated using a waterfall approach, which involved conducting comprehensive research to develop a profound comprehension of the difficulties encountered in the conservation of small indigenous languages. The process encompassed an extensive examination of scholarly literature, seeking guidance from esteemed linguistics specialists, and actively involving individuals within the Dagbani-speaking community who possess a high level of proficiency in the language. The knowledge acquired from this study established the basis for delineating the application's prerequisites and goals.

During the transition to the agile phase, the development process adopted iterative and incremental practises in order to facilitate ongoing enhancements and adapt to changing requirements. Agile methodologies, specifically Scrum, were utilised to effectively oversee the development tasks, incorporating frequent sprints and feedback cycles. The utilisation of an iterative approach enabled the seamless integration of novel features, resolution of concerns, and adjustment to evolving user needs.

The user-centered design approach played a crucial role in the methodology, ensuring that the application effectively addressed the needs and expectations of its target users(2). The utilisation of user research and usability testing proved to be instrumental in the

acquisition of valuable insights and the substantiation of design decisions. The team obtained feedback on the usability, language support, and overall user experience of the app through interviews and usability sessions conducted with individuals who possess proficiency in Dagbani.

In order to strengthen the user-centered approach, the principles of participatory design were adopted, which entailed the active engagement of community members who possess expertise in Dagbani throughout the entirety of the development process(3). The incorporation of their contributions, recommendations, and cultural perspectives was incorporated into the app's design and functionality, thereby guaranteeing its genuineness and applicability to the intended user demographic.

4.0.1 System Architecture Overview

The architectural design of the crowdsourcing application for Resourcing Small Indigenous Languages in the Field, as shown in Figure 4.1, encompasses a range of interconnected components that collaborate harmoniously to deliver the intended functionality. The primary constituents of the architectural framework encompass the React Native Client, Server/Controller, Phone/PC, and Cloud Services.

1. **React Native Client:**
   The React Native Client refers to the mobile application and web app that has been created utilising the React Native framework and React framework respectively. The user interface functions as the medium by which users engage with the application. The applications allows users to capture audio recordings of themselves pronouncing words in the Dagbani language. The software offers a user-friendly Graphical user interface (GUI) that facilitates seamless navigation across multiple screens and enables users to effortlessly execute diverse operations.

2. **Server/Controller:**
   The Server/Controller component serves as the central control point within the system. The system manages the logical operations and executes the API requests that are received from the React Native Client. The Server/Controller assumes the responsibility of validating and processing user actions, encompassing tasks such as recording word submissions, managing user authentication, and coordinating data synchronisation.

3. **Phone/PC:**

   The Phone/PC component symbolises the electronic device utilised by the user, encompassing both mobile phones and personal computers. The platform on which the React Native Client app is installed is provided by it. The Phone/PC engages in interaction with the Cloud Services component in order to facilitate the synchronisation of data.

4. **Cloud Services:**

   The Cloud Services component is of utmost importance in the architecture of your application. Cloud-based services, such as Firebase or comparable platforms, are employed to store and oversee user data and facilitate the synchronisation of data. The Cloud Services component offers a range of functionalities such as data storage, authentication, real-time data synchronisation, and updates. This feature enables users to conveniently retrieve their recorded content and other data associated with the application from various devices, while also ensuring the uniformity and coherence of data across different platforms.

The interactions among these components can be characterised as follows:

- The React Native Client establishes communication with the Server/Controller component by means of API requests. The Server/Controller receives these requests in order to execute various actions, including but not limited to submitting recorded words, authenticating users, and retrieving data. The requests are processed by the Server/Controller, which verifies the inputs and executes the required operations.

- The Server/Controller is responsible for generating API responses in response to API requests. The aforementioned responses encompass pertinent data, error messages, or success notifications that are exhibited to the user within the React Native Client application.

- The Phone/PC component facilitates the exchange of data with the Cloud Services in order to achieve data synchronisation. When a user utilises the React Native Client application to capture a word, the data pertaining to the recorded word is transmitted to the Cloud Services for the purpose of storage. The Cloud Services subsequently enable instantaneous data synchronisation, guaranteeing that the recorded information is readily available across multiple devices.

- The Cloud Services module facilitates the process of data synchronisation between the Phone/PC and the Server/Controller. This feature guarantees that modifications made on one device, such as the addition of a new word to the vocabulary, are promptly synchronised across multiple devices. Additionally, it is responsible for overseeing updates and resolving conflicts, if they arise, in order to uphold data consistency across various platforms.

In general, this architectural design facilitates the ability of app users to capture and store words in the Dagbani language through the utilisation of their mobile devices or personal computers. The React Native Client facilitates a cohesive user interface, while the Server/Controller manages the computational and logical aspects of user interactions. The integration of Phone/PC and Cloud Services components facilitates the synchronisation and storage of data, thereby ensuring a seamless and uniform user experience across various devices.

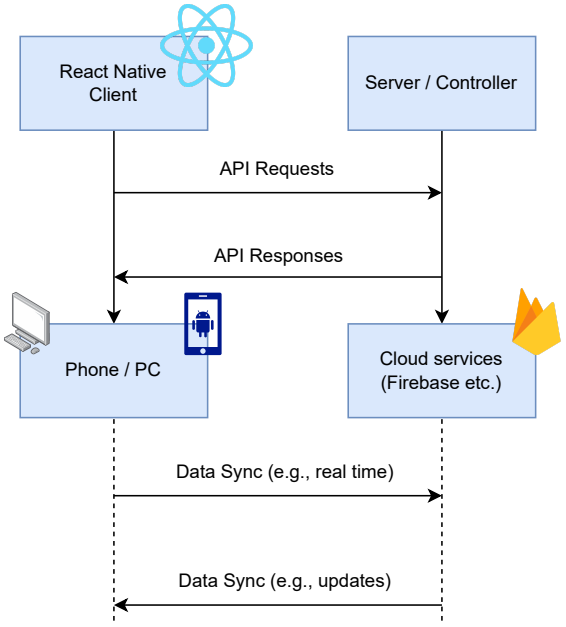


Figure 4.1: Architecture Diagram: Interconnecting Components within the System

4.1 Steps in the Process of Designing

The designing phase consisted of creating multiple iterations of the system based on prototypes, with each version incorporating feedback and insights gained from the previous iteration. This allowed the team to evaluate various design solutions and determine the

optimal approach. The testing phase involved gathering user feedback on the system's functionality and usability. This feedback was used to refine the design further before arriving at the final solution.

Throughout the design process, it was crucial that the system meet all requirements and specifications outlined in Chapter 2.2. Figure 4.2 depicts the design thinking process, and sections 4.1.1 to 4.1.7 describe each of its phases in detail. The final solution was designed to be low-resource, user-friendly, and effective in environments with limited internet connectivity.

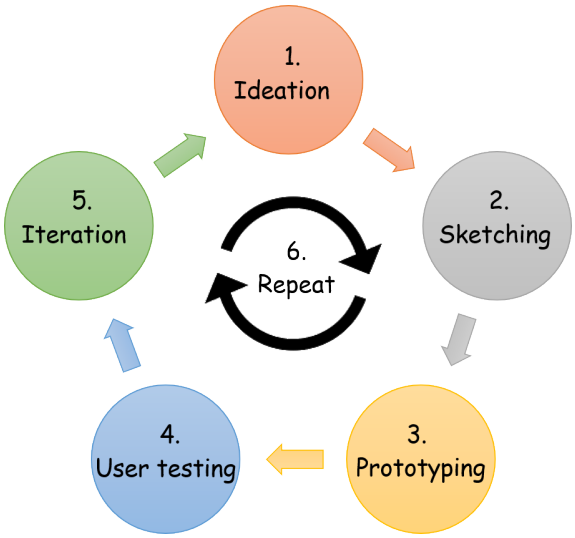


Figure 4.2: Five iterative steps for final solution design process

4.1.1 First step: Ideation

The project's ideation phase began with a series of rigorous brainstorming sessions focused on generating novel concepts for the application's layout, design, and functionality. The main goal of the team was to create a user-friendly interface that would facilitate comprehension and utilisation for all users. Significant importance was given to the integration of offline functionalities within the application, alongside the development of an effective mechanism for uploading recorded data to the server upon the availability of an internet connection. This subsection presents a thorough overview of the project's advancement, showcasing two activity diagrams that depict the sequential procedures associated with completing user details and commencing the recording process. These diagrams are highly valuable visual tools for understanding the sequence of actions and decision points within the application.

The activity diagrams were constructed with meticulous attention to detail in order to serve as visual aids and effectively outline and organise the desired features of both the mobile app and the web app. The diagrams effectively illustrate the sequential progression of actions and decision points within the application by organising the processes and states in a structured manner. The provided materials function as a comprehensive instructional tool, demonstrating the manner in which users will navigate through multiple stages and engage with diverse elements within the application. The utilisation of activity diagrams has been crucial in delineating the trajectory of user interactions and guaranteeing a cohesive and efficient encounter on both mobile and web interfaces. The commencement of the recording process is initiated by the start symbol. The process is initiated by users accessing the primary "home page" that includes the recording feature. In order to guarantee precision in the documentation of language, individuals are obligated to carry out a series of consecutive actions. These procedures consist of three subordinate categories. Initially, it is necessary for users to furnish their chosen moniker and gender, thereby enabling the retention of their contribution particulars. In addition, the user is required to make a selection between two available alternatives, namely "Word" or "List," when opting for the recording feature. Finally, the users are requested to complete the "category list" pertaining to the words. When opting for the "Word" alternative, an extra measure entails the selection of the particular term to be recorded.

Simultaneously, the user performs the actions of filling in the nickname and gender, selecting the recording option, and choosing the category list, which are represented by parallel bars. In the event that any of the compulsory fields are not completed accurately, an error notification will be exhibited. In contrast, in the event that all requisite components have been fulfilled, the procedure proceeds seamlessly without any instances of failed pop-up notifications.

The ensuing procedure bifurcates into two discrete subcategories, depending on the user's choice of the recording alternative. Irrespective of the selected alternative, the initial word materializes on the user's display.

In the event that the user opts for the "List" recording feature and encounters an unfamiliar term, they may elect to utilize the "Skip" function and advance to the subsequent word within the designated category list. As an alternative, individuals have the option to capture the presented term and subsequently playback the recorded audio file for the purpose of confirmation. In the event of user dissatisfaction with the recording, it is possible to re-record the chosen word until a satisfactory outcome is achieved. After the user

submits the word and saves the recording on their device, a subsequent word is generated from the chosen list if there are additional words available.

On the other hand, in case the user opts for the "Word" recording alternative, it is inferred that they possess prior knowledge of the designated term, given that they have made a prior selection. Users may proceed to capture the word and subsequently listen to the recorded file, repeating this process iteratively until they are content with the recording. Ultimately, the users submit the ultimate recording of the term, and the document is stored on their respective device.

In either scenario, a message denoting successful completion of the recording process is exhibited on the display. After accomplishing all the required tasks, a conclusive measure, such as "Documenting the term," can be executed. The procedure is completed, resulting in the intended objective being attained.

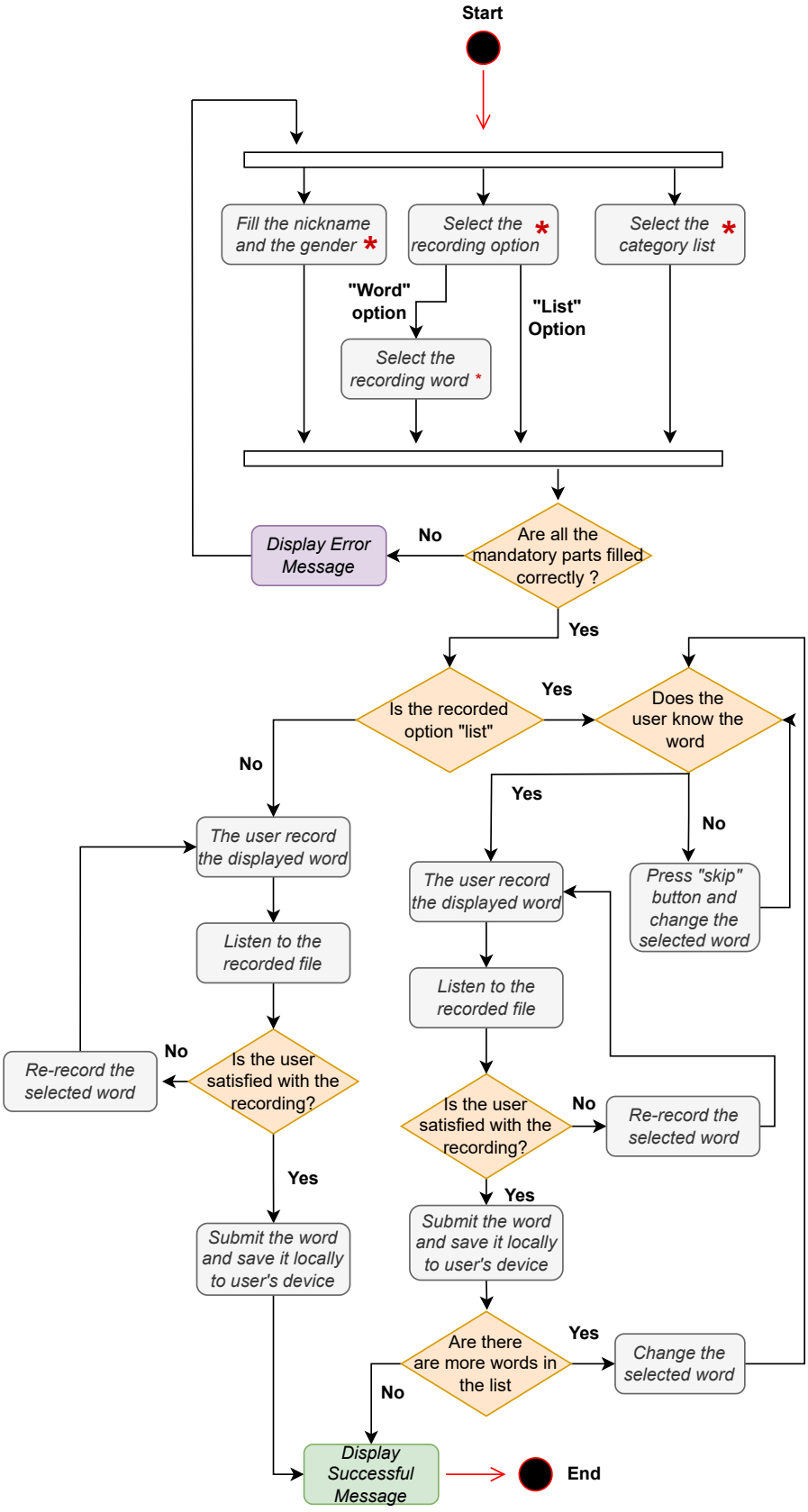


Figure 4.3: Activity diagram of the recording process

The initiation of the uploading process is signaled by the "start/initial node". The process is initiated by users who access the primary "uploading page" that encompasses the uploading functionality. At the outset, individuals are prompted to designate the recording files they intend to transfer to the crowdsourced cloud storage platform through the act of selecting the corresponding checkboxes. In the event that the quantity of checkboxes that have been selected is equal to zero, an error message will be exhibited to indicate that the selection made is invalid. On the other hand, in the event that the checkboxes that have been chosen satisfy the requisite standards, the process of uploading is initiated.

The initial step involves verifying whether the device utilized by the user is equipped with a WiFi connection. In the event that the prescribed condition is not satisfied, an error notification is exhibited, directing the user to retry the upload process upon the availability of a WiFi connection. In the event that the user's device is equipped with internet connectivity, the process of uploading ensues.

Upon selection of each recording, the respective file is transferred to Firebase, and a progress bar is systematically updated until it reaches 100% completion. Upon completion of the procedure, in the event that all of the recordings were successfully uploaded without any errors, a confirmation message will be displayed, indicating a successful upload. In the event that errors were encountered during the uploading process, an error message will be presented to the user. This message will enable the user to view the files that were not uploaded, but are still present in the list.

In conclusion, the process culminates in the successful attainment of the intended objective, which involves the uploading of the designated files.

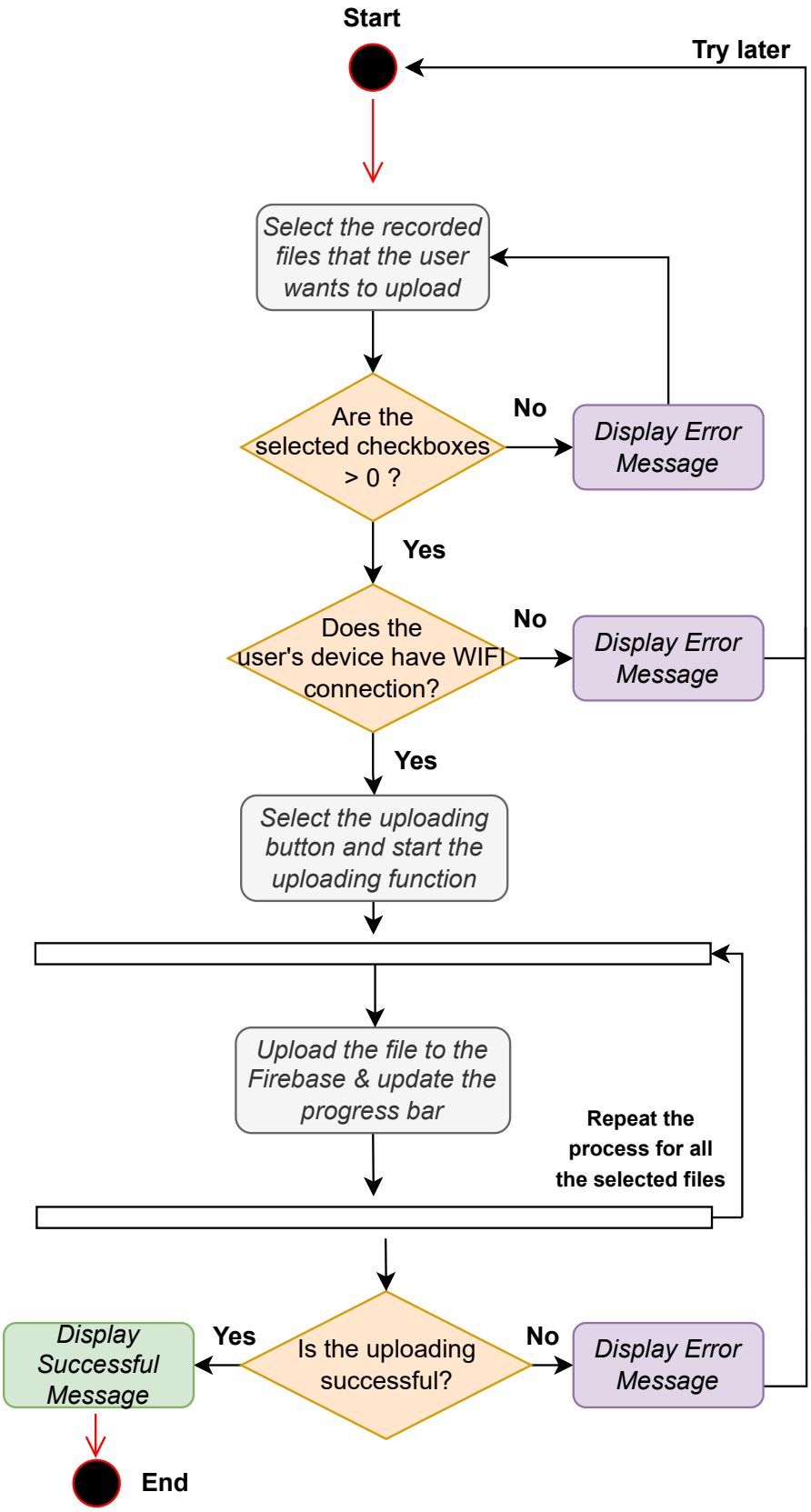


Figure 4.4: Activity diagram of the uploading process

4.1.2 Second step: Sketching

Following the successful completion of the ideation stage, the team proceeded to enter the sketching phase in order to enhance the conceptualization of the application. In this stage, the team generated initial visual depictions of the application's interface, incorporating components such as buttons, menus, and text fields. The utilisation of these sketches played a crucial role in effectively communicating the desired visual aesthetics and user experience of the application to both stakeholders and the development team.

In conjunction with the visual components, the team incorporated explanatory annotations and informative notes within the sketches to depict the conceptualised functionality of various features. This facilitated a comprehensive comprehension among the stakeholders and development team regarding the intended functionality and user engagements of the application.

A notable inclusion during this phase involved the development of a diagram that visually depicted the manner in which the screens would transition between various stages of the application, as shown in Figure 4.5. The diagram presented offers a graphical representation of the sequential progression of screens, illustrating the manner in which users would traverse the different sections and phases of the application. The diagram functions as a valuable instrument for promoting effective communication and comprehension among stakeholders and the development team, facilitating a collective perception of the user journey of the application.

The sketches and accompanying screen transition diagram served as the basis for further design and development efforts, facilitating the team's advancement towards the implementation stage with a well-defined trajectory and consensus among all involved parties.

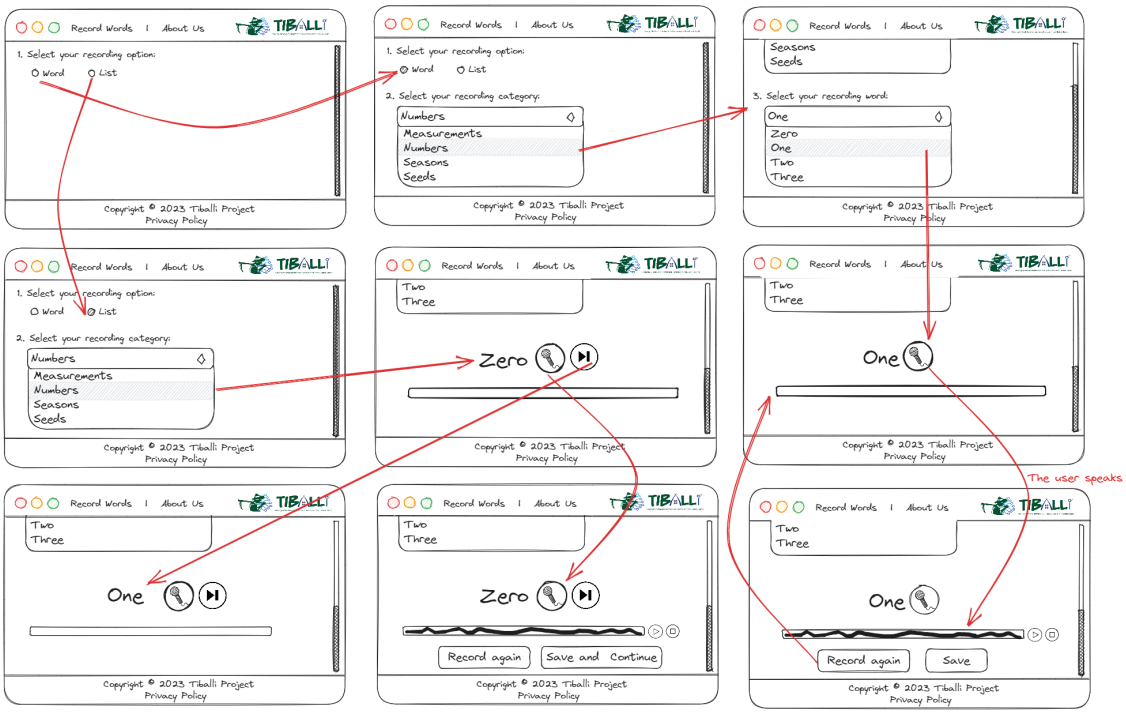


Figure 4.5: Concepts into Visual Representations: Illustrating the Journey towards User-Friendly Design

4.1.3 Third step: Prototyping

This process of prototyping and testing is a vital component of the design process and has been widely acknowledged as an efficient method for ensuring that designs meet the needs and expectations of users. According to a study by the Interaction Design Foundation (IDF)[1], prototypes play a crucial role in the design process, allowing designers to explore and test various ideas, collect feedback, and make informed decisions about the design's direction. Also, according to the National Institute of Standards and Technology (NIST), prototypes are a powerful tool for evaluating and improving designs because they allow designers to find and fix problems early in the design process before it becomes more costly to do so[2].

[1] https://www.interaction-design.org/literature/topics/prototyping
[2] https://www.nist.gov/publications/prototyping-new-products-systems-and-processes

4.1.3.1 Interactive prototype

During the prototyping phase, designers utilize tools like proto.io[1] to create a functional model of their design. Proto.io is a digital prototyping tool that enables the creation of interactive prototypes with a variety of features. These functionalities are intended to bring the design to life and enable users to test the design as if it were a real product. With the ability to record audio, upload files to the cloud, and display warning pop-ups, the prototyping process becomes more immersive and realistic, allowing for a greater understanding of how the design will function in real-world scenarios.

In addition, the interactivity of proto.io prototypes enables designers to test the functionality of their designs and identify potential obstacles. By testing the design in a controlled environment, designers can make informed decisions about the design's direction and make any necessary modifications to enhance its functionality and usability. The process of prototyping and testing also enables designers to collect feedback from stakeholders, enabling them to make well-informed decisions about the design based on the stakeholders' needs and expectations.

In addition to the functionalities offered by proto.io, the tool also features an intuitive interface, making it accessible to designers of all skill levels. This makes it easier for designers to implement their ideas and test their designs without requiring extensive technical knowledge or programming skills. As a result, proto.io has become a popular option for designers seeking to create prototypes for their designs, and its ability to streamline the prototyping process and improve the quality of designs has received widespread recognition.

4.1.3.2 Video prototype

In addition to the proto.io prototype, a video of the same prototype was also created. The video is recorded and edited to highlight the design's functions and features. A pre-recorded video, unlike a real-time prototype, is a static representation of the design and does not permit user interaction. A pre-recorded video prototype, however, has limitations in comparison to a real-time prototype. Therefore, it is used in conjunction with other prototyping techniques to gain a thorough understanding of the design and its potential impact on users.

A prerecorded prototyping video and a real-time prototype are two distinct presentation methods for a design concept. Key distinctions between the two include:

[1] https://proto.io/

- **Interactivity**: A real-time prototype enables users to interact with the design, while a prerecorded video does not. This interaction allows designers to test the functionality and usability of their designs, as well as receive immediate feedback on how users feel about the designs. With a prerecorded video, however, only visible feedback is available.

- **User testing**: Real-time prototypes enable designers to perform user testing, which is a critical aspect of the design process. This enables designers to collect data and insights regarding the design and make informed decisions regarding the project's direction. This level of testing and data collection is impossible with a pre-recorded video.

- **Flexibility**: Real-time prototypes are more flexible than pre-recorded videos, as they can be updated and refined in response to user feedback. Any changes to the design must be reflected in a new video if the video has already been recorded.

The opportunity to present the design at a workshop in Ghana in February 2023 was a major factor in the decision to make both prototypes. This workshop provided the team with an opportunity to present their ideas to a larger audience and solicit feedback from stakeholders. Critical to the design process, feedback enables creators to comprehend the strengths and weaknesses of their designs and make well-informed decisions regarding the future of the design. By presenting a model of the design's functionality, usability, and user experience, workshop attendees could gain a better understanding of the concept and provide constructive feedback on various aspects of the design, such as functionality and usability.

In addition, the designers could gain insight into the needs and expectations of a larger audience by soliciting feedback from a diverse group of stakeholders. This information could be used to refine the design and ensure that it meets the needs of users in a variety of situations and contexts.

In conclusion, the prototyping process is a crucial step in the iterative design cycle, allowing designers to test the functionality and usability of their designs by bringing their ideas to life. In that process, real-time prototypes and pre-recorded videos were included since both have their benefits and limitations. By presenting a working model of the design, workshop participants could gain a better understanding of the concept and provide feedback that could be used to refine the design and ensure it meets the needs and expectations of users. The prototyping process, in conjunction with the workshop, played a vital

role in the design process by allowing designers to gain valuable insights and feedback and ultimately refine their design to better meet the needs and expectations of users.

Below are the links for the two different prototype styles:

- Video prototype: `https://youtu.be/U6P6FmC2_4s`

- Proto.io prototype: https://antriapanayiotou.proto.io/player/

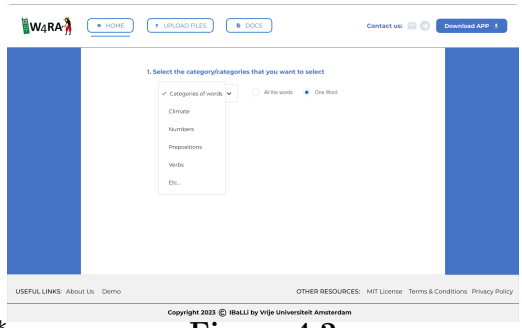


Figure 4.2
Selecting category

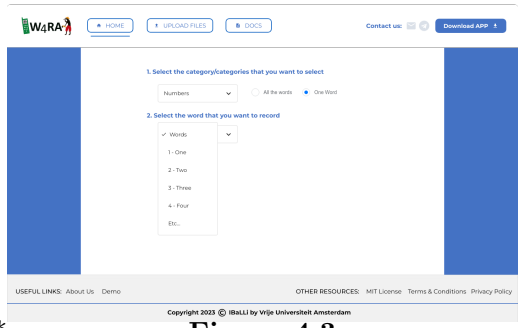


Figure 4.3
Selecting word

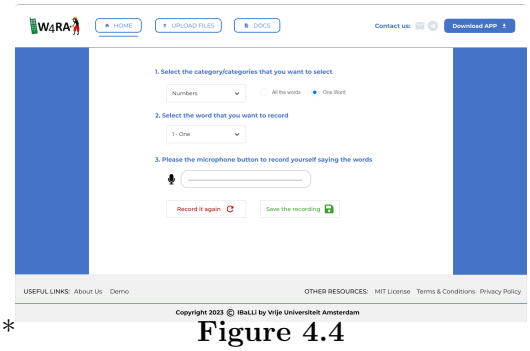


Figure 4.4
Before recording one word

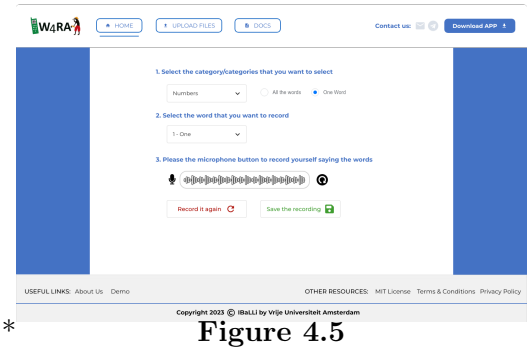


Figure 4.5
After recording one word

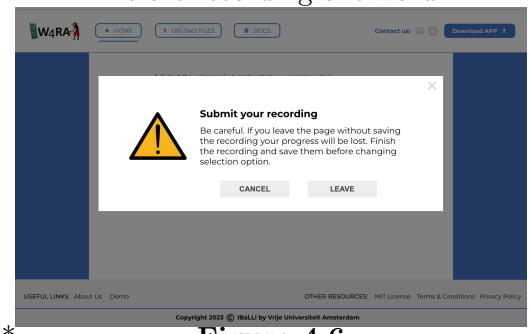


Figure 4.6
Warning message - submit empty file

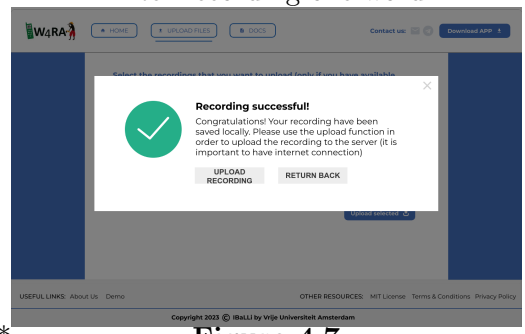


Figure 4.7
Successful message for recording

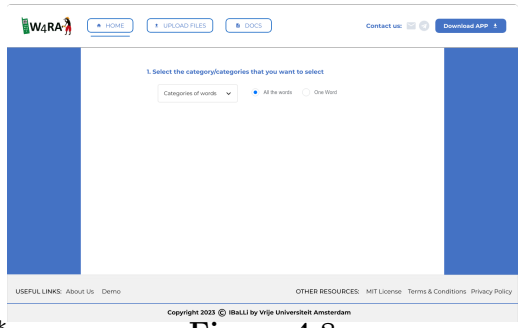


Figure 4.8

Default screen for many words record

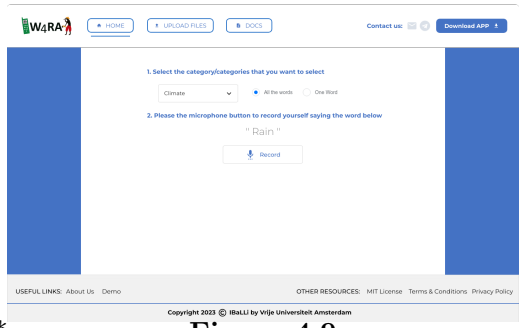


Figure 4.9

First screen after selecting category of many words option

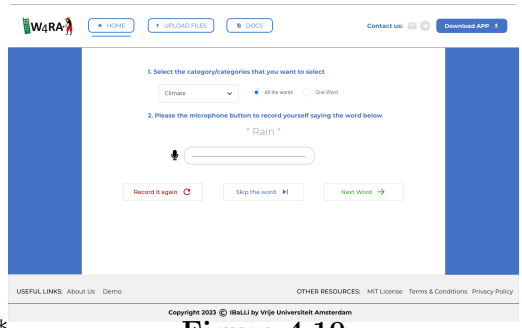


Figure 4.10

Showing the first word of the category

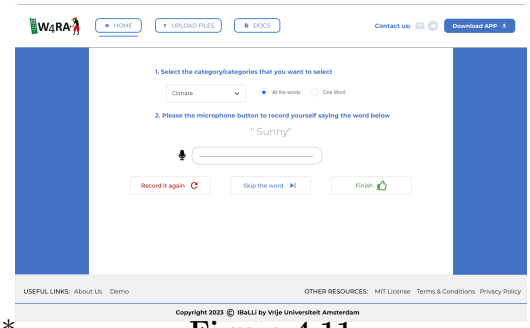


Figure 4.11

Showing the last word of the category

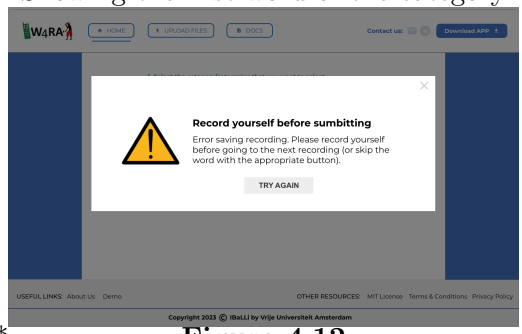


Figure 4.12

Warning message of submitting an empty file

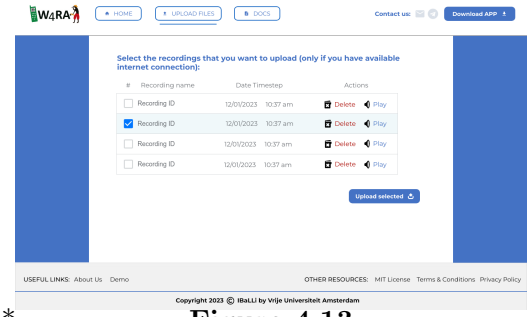


Figure 4.13

Default screen of local recordings

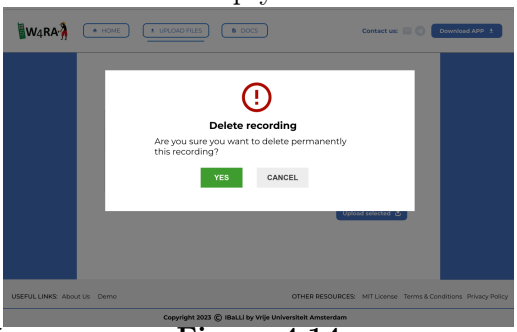


Figure 4.14

Confirmatory message for removing a recording

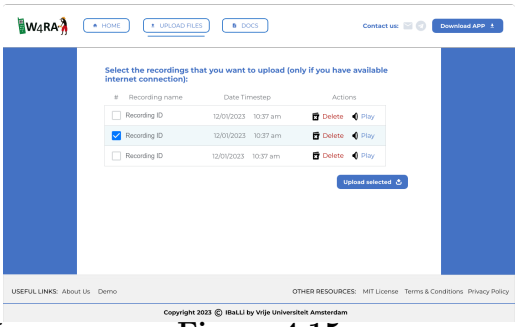


Figure 4.15

Default screen of local recordings after deletions

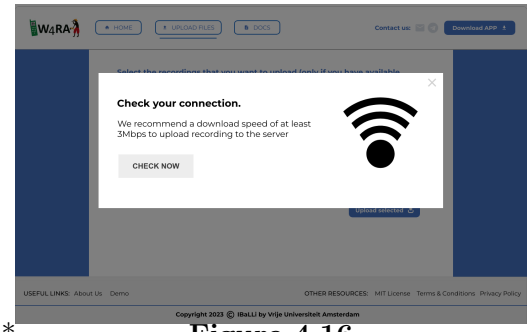


Figure 4.16
Starting screen of checking the connection

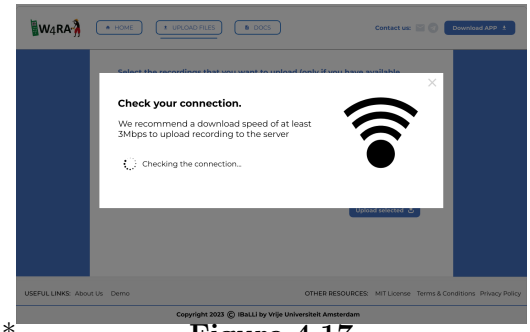


Figure 4.17
Screen while checking the connection

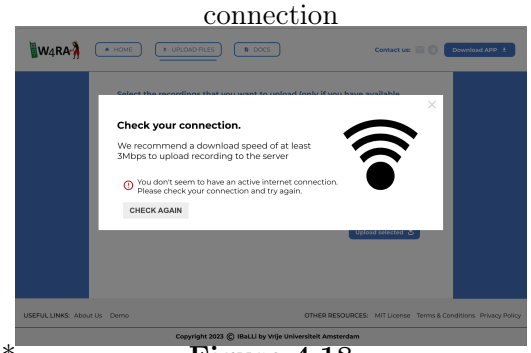


Figure 4.18
Screen while having problem with the connection

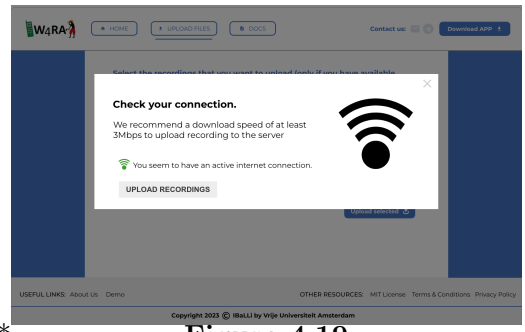


Figure 4.19
Screen without problem with the connection

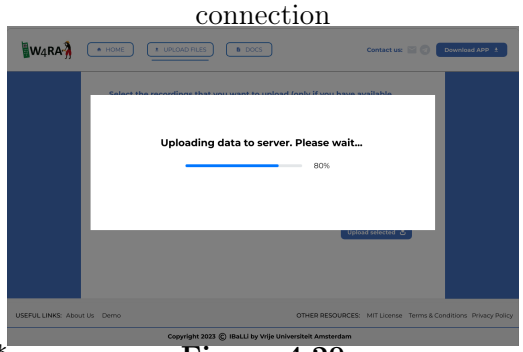


Figure 4.20
Loading bar screen for uploading data

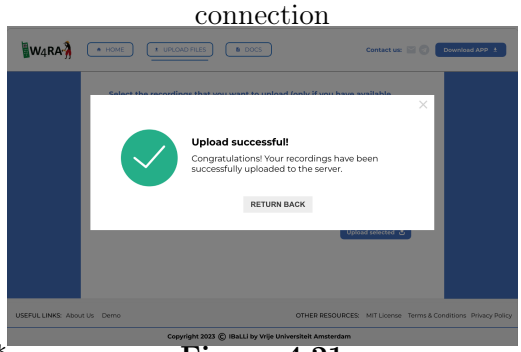


Figure 4.21
Successful message for uploading data

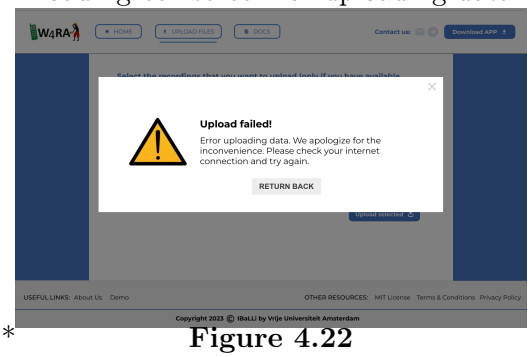


Figure 4.22
Warning message for not uploading data

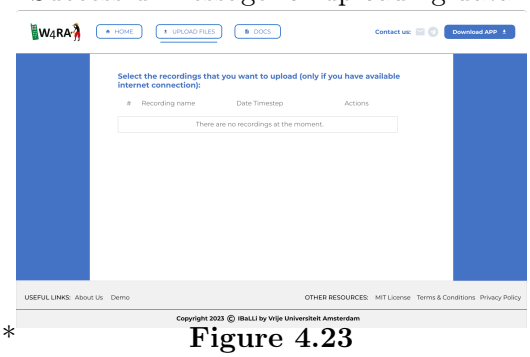
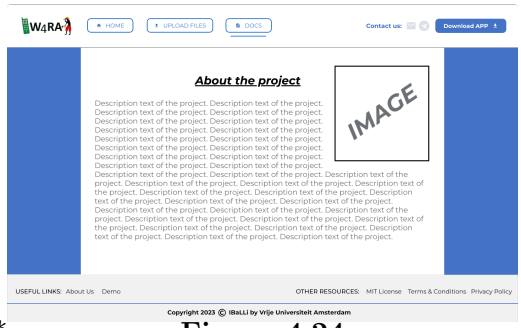


Figure 4.23
Default screen of local recordings - empty



* **Figure 4.24**
Default screen of "About us" page

Sketch prototypes of the web app - desktop view

4.1.4 Fourth step: User Testing

As each version of the mobile application was produced, a test APK file was distributed to other team members for comments on the application's functionality, user experience, and design. Enhancing the mobile application was greatly aided by team member input. But, because there was a great number of feedback, it was necessary to create a mechanism to sift it and address the most pressing issues first. As a result, we utilised the MoSCoW[1] method to compile the team members' feedback and to rank the concerns according to their significance and impact on the user experience.

A major factor in the success of any software development project is the calibre of team member input. To enhance the quality of the mobile application, it is essential to prioritise the input based on its impact on the user experience and its relative importance. The MoSCoW approach is one of the most efficient techniques employed for this purpose. In this chapter, we will describe how we used the MoSCoW approach to evaluate the input obtained from team members in order to enhance the mobile application.

MoSCoW is a frequent technique used in software development projects to determine which tasks must be completed first. It entails classifying needs into four groups based on their relative importance and impact on the project. The four categories consist of "must have", "should have", "could have", and "won't have". By utilising the MoSCoW technique, we are able to prioritise the team members' feedback and address the most pressing issues first.

In the beginning of this step, we will provide a summary of what the team members informed us. Then, we will discuss how we utilised the MoSCoW approach to categorise and rank the comments. The MoSCoW approach was used to analyse user comments and

[1] Moscow Explanation

make the mobile application better. During the development of the mobile app, several feedback points were received from team members regarding the user experience. Below is listed a summary of the feedback:

Francis Saa-Dittoh (UDS, Tiballi) One significant issue that was identified was the *number of pop-ups during the uploading process*, which created confusion for users and gave the impression that something was wrong. Therefore, it is suggested to limit the number of pop-ups during the uploading process to create a simpler and smoother user experience. Another issue identified was on the main page under the "Record Words" section. It was suggested to *replace the option "category" with "list"* to avoid confusion with the second drop-down menu that appears, which also contains the word "category". This minor change can improve the user experience and make it less confusing for users. Finally, it was suggested that after a user records a word under the "word" option, *the app should return to the same selected category* and word option instead of going back to the default page with no selected category. This change will make the experience easier and faster for users, especially when they want to record the same word to create a larger training dataset.

André Baart (Babafla) One suggestion was to *simplify the interface or include a 'simple' mode* that is easy to navigate for users, allowing them to complete tasks without assistance. Additionally, it was suggested that the app could include the option to display the interface in Dagbani language as part of its *multilingual capabilities.* This feature can make the app more accessible and user-friendly for native Dagbani speakers. After uploading recordings, it was suggested that the app should provide users with the ability to *view their contributions on the device.* This feature can improve user engagement and motivation by allowing them to see their progress and contributions. It was also suggested that the contributions could be grouped by the time of recording or speaker. For training purposes, it was suggested that the recordings could be categorized based on the gender, age, or background noise of the speaker. This categorization can provide users with a more diverse and comprehensive understanding of the language and improve their language learning experience. Lastly, it was suggested that the *app should load words and categories through an API.* This enhancement can streamline the process of updating and managing the app's content, making it more efficient and scalable.

Naafi Ibrahim One suggestion that was made was to include *recordings of whole sentences* in addition to single words. This change can help users understand how words are used in the context of a sentence and provide them with a more complete language learning experience. Furthermore, since the scope of the app was initially focused on recording Dagbani words, it was suggested to *include some actual Dagbani words* as part of the vocabulary. This addition can help users build a foundation of commonly used words in the language and provide a more engaging and authentic learning experience.

4.1.5 Decisions Based on MosCoW Feedback Technique

After making the MosCoW diagram, Figure 4.7, and meeting with the project team, Figure 4.6, it was decided that changes would be made based on how important they were and how long it would take to make them. The goal of each change that was suggested was to improve the user experience and make it easier to use. Feedback that did not align with these objectives was deemed unsuitable and rejected. For instance, a suggestion to retain information on user contributions was deemed undesirable since it would necessitate users to provide personal information, leading to a more complex application. As simplicity was a priority, this idea was deemed unsuitable for the project, particularly since the target users were unlikely to be familiar with such applications.

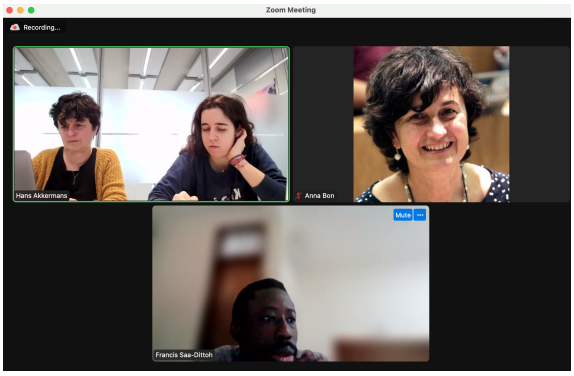


Figure 4.6: Conducting a virtual feedback session on the 1st version of our mobile app with Mrs. Anna and Mr. Francis

In the same way, the idea of having all the steps on one page was thought to be useful and faster, but it could lead to data loss and problems with the Internet connection and resources in Ghana. To avoid such issues, steps needed to be separated into

different screens. This feedback was taken into account for the website version of the project, as users accessing the site would have internet connectivity and could directly upload data to the server after recording.

Another idea was to make the app support both English and Dagbani by making it multilingual. But since most Dagbani speakers in rural areas can't read and even educated Dagbani speakers might find it hard to read Dagbani, it was thought to be hard to make prompts, buttons, and menus in Dagbani. So, it was decided that it would be best to put the English translations of words like "yes," "no," and numbers in brackets next to the recorded words, or the other way around. This method would make it easier for native speakers who can read to identify words based on how they sound.

Last but not least, we talked about small changes that would be easy to make, such as reducing the number of pop-ups during the uploading process, changing the "category" option to "list," and changing the "word" option's return state after each recording. People thought these changes were important because they would make things easier and faster for users without adding any confusion.

In conclusion, changes were made to improve the user experience and make the application easier to use by using the MosCoW feedback method and working with the project team. The team rejected feedback that did not align with these objectives and focused on making changes that would be easy to implement and add value to the project. On the other hand, the team believes that the changes that were selected will significantly improve the application's usability and enhance the user experience.

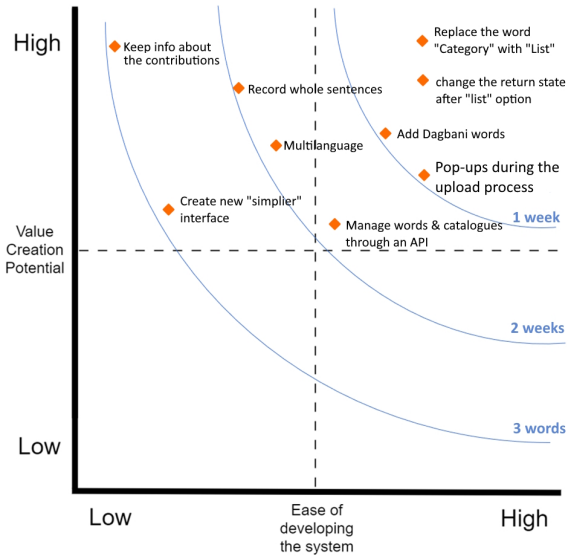


Figure 4.7: Prioritization of mobile app feedback using the MoSCoW method.

4.1.5.1 Implementation of the multi-language feature:

The incorporation of the multi-language feature into the mobile application was executed during one of the implementation phases, despite its ultimate non-utilization. The visual representation depicted in Figure 4.8 illustrates the appearance of the mobile application subsequent to the integration of the multilingual version. Upon selecting a language from the dropdown menu, the website will automatically present the content in the language that has been chosen by the user. The website's entire textual content, encompassing headings, menus, and descriptions, will be rendered in the language of the user's choice.

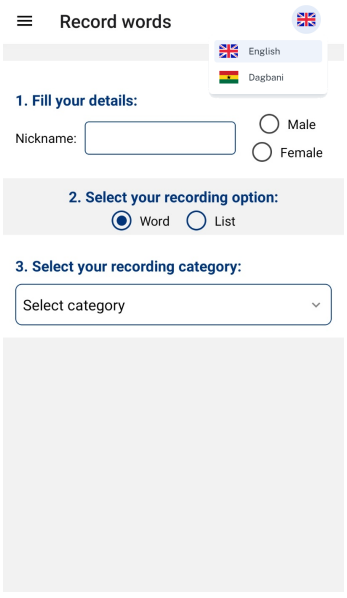


Figure 4.8: Dropdown of the multi-language implementation

4.1.6 Fifth step: Iteration

Following a series of user tests and iterative design processes, the team obtained significant insights regarding the system's design and functionality. These insights were based on the latest version of the system, which was developed after incorporating the feedback obtained from the previous iteration. A request was made to introduce novel functionality for the purpose of tracking user information, including their gender and nickname. This would facilitate the analysis and modeling of audio file recordings within the training data set. The team conducted a thorough evaluation regarding the tracking of the user's complete name versus their nickname, and ultimately opted for the latter option to ensure more accurate identification. Upon conducting a market analysis, the team arrived at the conclusion that utilizing nicknames would prove to be a more efficacious approach, particularly in the event that the application were to be expanded for implementation by a multitude of users. This phenomenon can be attributed to the high probability of encountering numerous users who share identical full names, especially in areas like northern Ghana where the aforementioned application is currently in use. By employing the practice of monitoring nicknames, the process of identifying specific users can be enhanced in terms of precision. This is due to the fact that there may be more than 10 individuals with the name "Alhassan Abdulai". The team's commitment to meeting the needs of its users is exemplified by their meticulous attention to detail and responsiveness to user feedback.

4.1.7 Sixth step: Repeat

The modifications requested following the fifth iteration, which included adding a text field and a radio group for the user's nickname and gender, were implemented in the system's sixth round of testing. Nevertheless, it was discovered during functional testing that the text box and the radio button did not maintain their values on the mobile application, which turned out to be inconvenient for users of mobile apps.

It was suggested that the mobile app should track the user's details even though the website does not require users to create accounts and cannot, therefore, retain user information because the owner of the mobile phone is typically the user. A new APK was developed to address this problem; it contained extra data saved on the user's device.

Overall, the requested changes were successfully implemented after the sixth round of testing, and the problem with the text area and radio button values not being retained was fixed in the new APK for the mobile app.

4.2 Essential Services for Managing the system

Any system's success is largely dependent on its data management and utilisation. When creating a word recording management system, it is crucial to consider the various services required to manage and store data effectively. These services consist of a database for data storage and retrieval 4.2.2, a cloud service for real-time data synchronisation and management 4.2.1, and an email account for communication and notifications 4.2.3. In this section, we will discuss the significance of these services and how they can be integrated to create a dependable and effective word recording management system.

4.2.1 Hosting Server Selection

The creation of a contemporary mobile application requires not just developing code but also hosting it on a server. Hosting an application on a server is essential for a variety of reasons. Firstly, it offers a reliable and secure environment for the program to execute, guaranteeing that it is constantly available and accessible to users. Secondly, putting an application on a server helps it to manage a high number of users and traffic, guaranteeing that it can scale as needed. Hosting an application on a server offers the infrastructure necessary for storing, managing, and serving data, which is vital for many current applications, such as dynamic crowdsourcing apps ([4], [5], [6]).

There are several free-tiered solutions for hosting dynamic React Native applications, each with its own pros and limitations. In this comparison, we will explore five prominent options: Firebase, Heroku, AWS Amplify, DigitalOcean, and Glitch.

Firebase[1] is an all-inclusive app development platform that offers a variety of services for storing, serving, and managing data, such as real-time databases, cloud storage, and authentication. Setting up and utilizing Firebase is straightforward, and it works with React Native apps flawlessly. Firebase's free tier includes 1 GB of storage by default, although more storage can be bought if necessary. Some of the downsides of utilizing Firebase in the free tier include restricted storage, bandwidth, and server resources, which may not be adequate for bigger or resource-intensive projects. Additionally, Firebase may not give the greatest performance for highly high-traffic or sophisticated projects, and additional services or technologies may be needed to satisfy specific needs.

Heroku[2] is a cloud platform that supports a range of programming languages and frameworks, including React Native. It is simple to install and operate, and it offers a free

[1] https://firebase.google.com/
[2] https://www.heroku.com/

tier with up to 550 hours of monthly server usage. Heroku includes a number of add-ons for storing and serving data, making it a versatile alternative for hosting dynamic React Native apps. Heroku's free tier offers restricted storage and server resources, which may not be adequate for bigger or resource-intensive applications. With increased traffic or usage, performance may deteriorate, and new services or tools may be required to satisfy specific requirements.

AWS Amplify[1] is a cloud platform that offers a variety of data storage, serving, and management services, including managed databases, storage, and authentication. AWS Amplify is simple to connect with React Native applications, and the free tier offers a set amount of storage and server resources. However, the free tier of AWS Amplify also has restricted storage and server capabilities and may not be suitable for bigger or resource-intensive projects. Additionally, AWS Amplify can be hard to set up and operate, especially for first-time users, and other services or tools may be needed to suit certain requirements.

DigitalOcean[2] is a platform for hosting that provides easy, adaptable, and scalable hosting options. It provides a variety of managed services, including databases, storage, and backups, and offers free hosting for static websites. However, the free option for hosting static websites on DigitalOcean has limited capacity, and dynamic applications may require more storage. In addition, DigitalOcean may require more technical skill to set up and administer than other alternatives, and extra services or tools may be required to suit the specialized needs of dynamic applications.

Glitch[3] is a cloud-based development platform with a simple and intuitive UI that is easy to set up and use. It supports a number of development languages and frameworks, including React Native, and provides a free tier with a specific amount of storage and server resources. However, the free tier of Glitch has restricted server and storage capacities, which may not be suitable for larger or more resource-intensive applications. With increased traffic or usage, performance may deteriorate, and new services or tools may be required to satisfy specific requirements.

In conclusion, when it comes to hosting a dynamic React Native application, Firebase is a complete and dependable solution that provides a variety of data storage, serving, and management capabilities. Not only must the program supply the required server resources, but it must also be able to store the recordings submitted by the app's users. To fulfill this requirement, Firebase is one of the best freeware options as evidenced by both Table 4.1

[1] https://aws.amazon.com/amplify/
[2] https://www.digitalocean.com/
[3] https://glitch.com/

and the preceding explanation. Firebase is straightforward to set up and use, and it works effortlessly with React Native applications, despite the fact that the free tier has storage and server resource constraints. Firebase is an excellent option for those seeking a flexible, scalable, and user-friendly hosting solution for their dynamic React Native application.

Functionalities:	**Firebase**	**Digital Ocean**	**AWS Amplify**	**Glitch**
Easy to set up and use	✓	✗	✗	✓
Integrates with React Native easily	✓	✗	✓	✓
Build Deploy Limitations	-	-	1,000 build minutes(per month)	-
Data Storage Limitations	1GB	1GB	5GB	200MB
Provides a number of services (storing, serving, and managing data, including managed databases, storage, and authentication)	✓	✓	✓	✗
Hosting Dynamic Content	✓	✗	✗	✓

Table 4.1: Comparing various hosting services

4.2.2 Database Selection

MongoDB is a highly flexible and scalable cross-platform NoStructured Query Language (SQL) database system that use a document-based data architecture as opposed to the traditional table-based approach utilized by relational databases [1]. This enables greater flexibility and the capacity to manage complicated and massive data sets. The usage of JavaScript Object Notation (JSON)-like documents with optional schemas is one of the defining characteristics of MongoDB. This provides greater flexibility in terms of data storage and access, as well as the ability to easily interact with other systems that utilize similar data structures. MongoDB Inc., created in 2007, is responsible for the program's development and maintenance. The open-source license it employs, the Server Side Public License (SSPL), is considered by some to be non-free; yet, it is designed to maintain MongoDB free and open-source while safeguarding the company's and contributors' rights. MongoDB is widely utilized by businesses of all sizes and in a variety of industries. It has a

[1] https://www.mongodb.com/

big and active community of users, developers, and contributors who offer an abundance of help and resources to anyone who want to use or work with the application. It is renowned for its efficiency, scalability, and dependability. It enables enterprises to organize, store, and query vast quantities of data with ease. In addition, it offers various benefits, including scalability to handle a growing user base or an increase in data, flexibility, and speedy performance. It employs a memory-mapped storage engine, allowing it to function with data sets bigger than available memory, making it a great choice for high-performance applications. It provides a range of indexing options, including full-text search, geographic, and hashed indexes, to improve the performance of searches. When horizontal scaling is required, it also allows sharding, which permits the dissemination of data across multiple servers. Lastly, MongoDB stores data in a JSON-like format known as Binary Javascript Object Notation (BSON), making it easy to work with and understand data and a suitable fit if the application employs JSON. All of these characteristics make MongoDB an attractive alternative for a vast array of applications, from basic web applications to large-scale, high-performance systems. Below is some information about the version used in the platform:

- Version: 6.0.1
- Documentation: `https://www.mongodb.com/docs/`

The following ERD, Figure 4.2.2, illustrates the relationships between the tables within our database, as well as their attributes, data types, and properties. This diagram has been constructed in accordance with principles of database design, and has been designed to meet the functional requirements of users, optimize the storage and retrieval of data, and minimize the unnecessary use of system memory. The association graph has been implemented in its second common form (2nd Normal Form (2NF)) to prevent information redundancy.

In Figure 4.2.2, various types of relationships between different entities are depicted, along with their meanings. As an example, the diagram shows that the "categories" table and the "words" table have a "one-to-many" relationship, where each category has multiple words, but each word is associated with only one category. It is important to note that in this relationship, it is possible for a category to have no associated words, but if a word exists, it must be associated with a category.

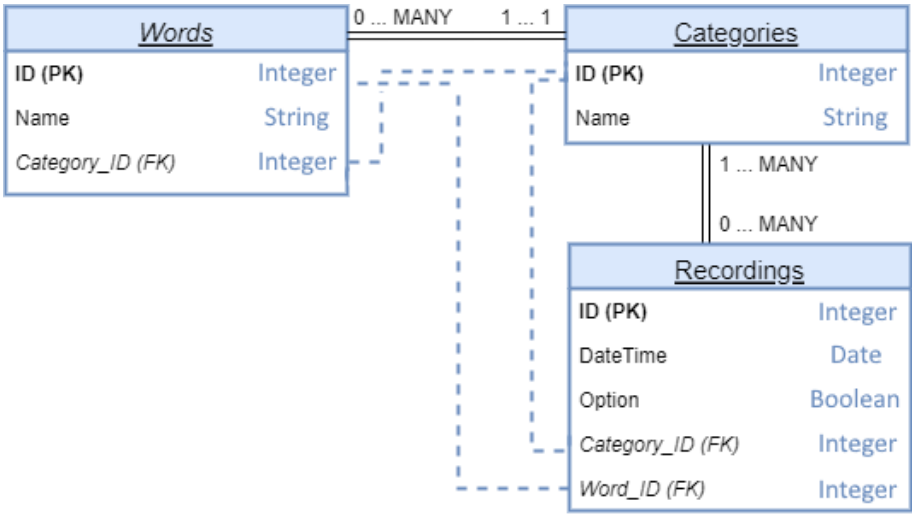


Figure 4.9: MongoDB schema ERD

In addition, the ERD displays the fields of each table, as well as their data types and other features, such as primary keys and foreign keys. The tables of the Database, as well as a description of how they are utilized on our platform and the main key that converts their data into unique ones, are listed below:

- **Table "Words"**
  Primary Key (PK): identification (ID)(increasing unique number)
  All recorded words by users through the crowdsourcing app are saved in this table. The corresponding verbal form for each word is also stored. This applies to all different types of words used in the project. Additionally, each word is associated with a specific category (Foreign Key (FK)), and the word generally falls under that category.

- **Table "Categories"**
  PK: ID(increasing unique number)
  The table in question stores all categories that can be recorded by users through the crowdsourcing application, including but not limited to weather, numbers, and verbs. The correspondence between each category's verbal representation and its corresponding noun is also stored in the table. Each category has a distinct meaning and contains a set of associated words (FK).

- **Table "Recordings"**
  PK: ID(increasing unique number)

The aim of the crowdsourcing app is to collect as many recordings of words as possible. To achieve this, the app utilizes a table called "Recordings" to store information regarding the folders containing the voice files. Specifically, the table associates an ID (PK) with the name of the file, and a category ID (FK) with the name of the folder file in the cloud, as well as the word (FK) which the recorded file resides. Additionally, the date and time of the recording, as well as whether it was part of the "one word" or "all words" recording option, are also recorded in order to distinguish between different recordings.

4.2.3 Email Service

Google's Gmail[1] is a free email service that enables users to send and receive emails, manage their contacts, and utilise additional features such as Google Drive and Google Docs. Gmail is one of the world's most popular email services. It has a straightforward, user-friendly interface and numerous security features to protect user data. Gmail accounts are important not only for personal use but also for business and development purposes, such as the "Dagbani speak" digital service for app development, cloud storage management, and online marketing. Gmail is an essential tool for providing users with a dependable and effective email service that facilitates communication and collaboration.

In digital services, the email service is essential for a number of functions and connections, such as the "Contact Us" button and the app's connection to the Firebase cloud service. The "Contact Us" function allows users to send support or feedback requests directly to the app's developers. For this feature to function, your Gmail account must be capable of sending and receiving emails. The user's Gmail account acts as an intermediary between them and the app developers, making it simple for them to communicate and receive assistance. In addition, a Gmail account is utilised to link the digital services to the Firebase cloud service.

In addition to enabling the "Contact Us" feature and connecting the app to the Firebase cloud service, a Gmail account is useful for managing the app's user base and sending notifications. By linking a Gmail account to the app, developers can take advantage of Gmail's built-in features for managing user emails and sending automated notifications, making it easier to stay in touch with users and provide support when necessary.

Setting up a Gmail account is required for a digital service that uses Firebase cloud services to function properly. It can also facilitate communication with users and data

[1] https://www.google.com/account/about/

management.

4.3 Streamlining Data Retrieval for Word Recording Management

In spite of the fact that databases can be advantageous in certain system configurations, there are situations in which they should be avoided. In the case of the 'Dagbani Speak' system, the use of a database may not be advantageous after weighing the advantages and disadvantages. The decision not to use a database in the mobile app designed to store word recordings can be justified by the fact that essential data can be obtained from Firebase cloud storage using a JavaScript script.

The category ID, word ID of the recording, and upload time can be extracted from the Firebase URL and metadata, respectively. Consequently, the use of a separate database may be deemed unnecessary and may result in increased costs, maintenance, and complexity.

In addition, since the app is designed to store recordings of words, using a database may not provide many advantages for acquiring or managing data.

The Firebase cloud storage is a dependable and efficient way to store the required amount of data and support the app's crowdsourcing features.

In conclusion, the decision to not use a database in the context of the mobile app for saving word recordings can be justified by the ability to extract required data directly from Firebase cloud storage using a JavaScript script. Given how the application operates, a separate database may not provide many benefits and can make the application more difficult to use and maintain.

4.4 Designing Dagbani Speak: A Platform for Community Engagement

4.4.1 Introducing Dagbani Speak - A Platform for Community Engagement on Mobile and Web

The software applications designed for our crowdsourcing platform, encompassing both mobile and web-based interfaces, are jointly referred to as "Dagbani Speak." The nomenclature inherently conveys substantial significance and mirrors the fundamental objective and intended demographic of our platform.

The term "Dagbani" denotes the linguistic system utilised by the Dagbamba ethnic group, predominantly concentrated in the Northern Region of Ghana. The inclusion of the language's name in the title of our platform serves as a tribute to the indigenous culture and underscores the significance of linguistic inclusiveness in fostering community involvement endeavours.

The term "Speak" serves as a fundamental aspect of our platform, as it aims to facilitate the ability of individuals to express their personal experiences, observations, and expertise through the means of crowdsourced contributions. Dagbani Speak endeavours to enhance the voices of the local community, encourage cooperation, and facilitate significant discourse by cultivating an environment that encourages active participation and the sharing of insights.

The nomenclature "Dagbani Speak" is not only indicative of the linguistic and cultural milieu of the Northern Ghanaian locality but also encapsulates the objective of our platform to enable community-based data gathering and dissemination of knowledge. By utilising both the mobile and web applications, individuals have the ability to express their opinions, provide significant data, and actively participate in endeavours aimed at resolving regional issues and prospects.

4.4.2 Design and Visual Elements: Creating an Iconic Brand Identity

The establishment of brand identity and communication of purpose are crucially facilitated by the visual representation of the crowdsourcing platform, Dagbani Speak. A novel .ico file has been specifically designed for the purpose of serving as the thumbnail for the mobile app and web app in this project, as shown in Figure 4.10. The said file has been tailored to encapsulate the essence of the target audience and context of our platform.

The recently created .ico file prominently showcases a female figure with a dark brown complexion, representing the heterogeneous populace of Northern Ghana, where the intended use of Dagbani Speak is focused. The deliberate selection of colour is intended to mirror the demographic of the area and foster a feeling of inclusiveness, thereby demonstrating our dedication to representing and interacting with the community we cater to. The woman is adorned with a traditional accessory commonly worn by Ghanaians, known as "Duka," which is a red African head wrap. This serves as a representation of Ghana's cultural heritage.

Figure 4.10: Iconic Brand Identity of the project

The incorporation of this particular element serves to imbue a unique character, emblematic of the affluent cultural customs and individuality of the area, concurrently establishing a visual association with the neighbouring populace. The woman's attire is imbued with local fashion and style, as evidenced by her prominent triangle earrings and necklace. These accessories serve as symbols of the lively and expressive nature of Ghanaian fashion. These elements serve to strengthen the association with the nearby community and establish an aesthetically pleasing depiction that strikes a chord with our intended audience. The woman's eyes are intentionally obscured in the Dagbani Speak platform to underscore its inclusive character, symbolising the notion that she embodies all prospective users of the platform. The objective of this methodology is to guarantee that the miniature image evokes a sense of familiarity with a diverse audience, irrespective of their individual traits or origins, thereby strengthening our dedication towards promoting inclusiveness and ease of access. The image portrays a female individual donning a white shirt that features the emblem of Web alliance for Regreening in Africa (W4RA), a multidisciplinary collective that conducts research and undertakes practical initiatives pertaining to real-life issues associated with the Digital Society. The Dagbani Speak crowdsourcing platform is a community project that is in line with its vision of leveraging digital technologies to achieve societal impact. The incorporation of the W4RA emblem serves to underscore the affiliation of our platform with this community and the cooperative endeavours aimed at harnessing technology for constructive transformation.

The image's backdrop exhibits a meticulously selected dark yellow hue, which serves as a representation of the qualities of warmth, vivacity, and dynamism that are emblem-

atic of Ghana's lively ambiance. The selection of this particular colour contributes to the complexity and aesthetic appeal of the thumbnail, while also harmonising with the overarching design components, resulting in a visually impressive depiction of the Dagbani Speak crowdsourcing platform.

The purpose of developing a new .ico file is to establish a distinctive brand identity for the Dagbani Speak crowdsourcing platform. The objective is to reflect the platform's aspirations of promoting diversity, cultural heritage, inclusivity, and societal impact through a meticulous creation process. The distinctive visual depiction functions as a potent emblem of our platform's objective and commitment to actively involving and enabling the populations we cater to in the Northern region of Ghana.

Having thoroughly described the methodology behind the language preservation platform, the "Implementation" chapter now takes center stage. Within this chapter, the intricacies of the mobile and web app functionalities unfold, encompassing installation procedures, navigation menus, and recording processes. Additionally, the chapter explores essential JavaScript files, design and usability rules, and highlights the distinctions between the mobile and web applications.

5

Implementation

This chapter provides a detailed exploration of the various functionalities of the mobile and web app. In this section, each of these features will be explored in depth, providing a comprehensive guide on how to use them to their fullest potential. In addition, this chapter is designed to help users, whether new or experienced, to learn more about the capabilities of the mobile and web app. Therefore, it is an essential resource to discover the full range of functionalities of the platform.

5.1 Functionalities of the mobile and web app

5.1.1 Steps for install the mobile app

The first step in installing the APK file on a mobile device is to download the file from this website (link). Once the download is complete, the APK file should be located in the device's file manager and clicked on to begin the installation process. If prompted, the device may require enabling installation from unknown sources(since is not available in the Play store yet).

Next, the user should follow the on-screen instructions to complete the installation, which may involve clicking "Install". During the installation process, the device will extract the necessary files and create an icon for the app on the home screen.

Once the installation is complete, the app icon called "Dagbani Speak" should be found on the home screen or in the app drawer. Clicking on the icon will launch the app and allow for its use. If any issues occur during the installation process, the user should consult the documentation or contact the app developer for assistance through the mail tiballi.project@gmail.com.

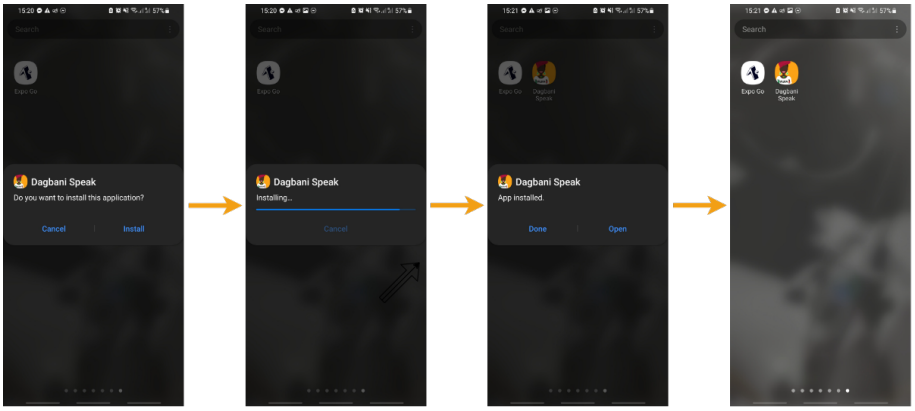


Figure 5.1: Installing steps for APK file

5.1.2 Loading/Splash screen:

The splash screen (Figure 5.2) is the initial screen that appears when a mobile application is launched. It is an introductory screen that displays the logo of the project team (W4RA) as the app loads its content. The purpose of the splash screen is to provide users with a brief, visually appealing introduction to the app while also indicating that the app is loading. The splash screen will automatically disappear once the content has loaded, and the user will be taken to the app's main screen. The design of the splash screen is intended to generate excitement and anticipation in the user as they eagerly await the app's launch.

Figure 5.2: Loading/Splash screen

5.1.3 Drawer navigation menu

The navigation drawer is a menu that slides-out out from the left side of the app's screen, giving users access to the app's features and functionalities. This feature is accessible from anywhere within the app, allowing users to navigate between app sections quickly and easily. In the provided screenshots, three distinct pages are displayed, and each screenshot displays the navigation drawer with the highlighted page. When the user clicks the menu icon, the navigation drawer slides out and displays a list of options corresponding to the app's various pages and features. The three primary pages with the most functionality are "Record Words," "Upload recordings," and "About us".

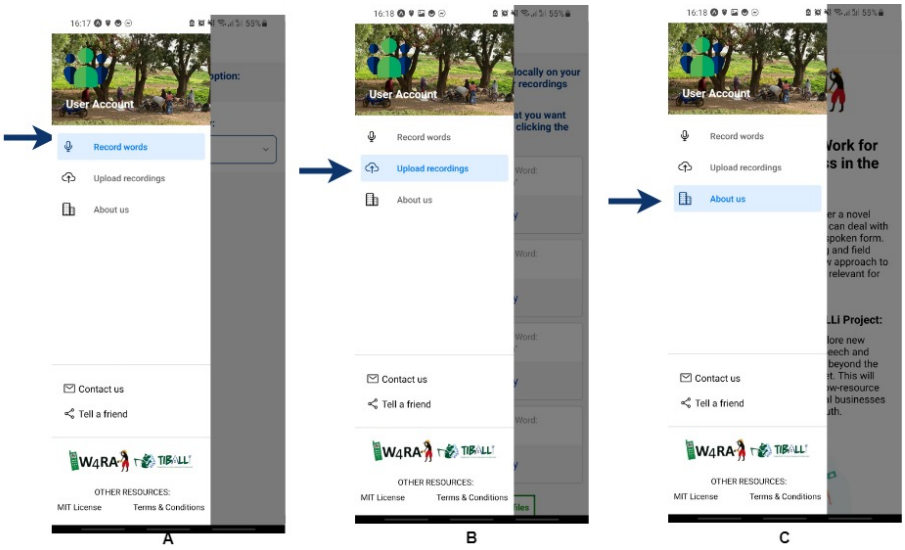


Figure 5.3: Different states of drawer navigation menu

By selecting one of the options, the user can easily navigate to the corresponding page, and the drawer will transform as depicted in the screenshots. The navigation drawer is an integral component of the app's user interface, providing an intuitive and user-friendly means of navigating between the app's various sections.

5.1.4 Secondary functionalities on drawer navigation menu

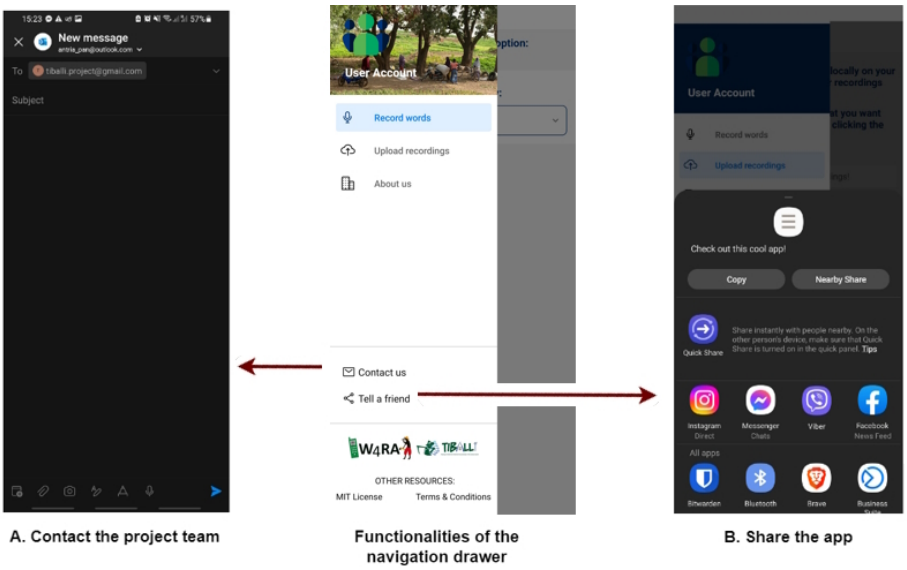


Figure 5.4: Secondary functionalities of drawer navigation menu

In addition to primary navigation options, the Navigation Drawer Menu contains secondary functionalities that enhance the user's overall experience.

5.1.4.1 "Contact Us" Functionality:

"Contact Us" is an example of a secondary feature that lets users talk to the app's developers or customer service staff. This feature is especially helpful if a user runs into a problem or wants to give feedback while using the app. The "Contact Us" feature in the Navigation Drawer Menu is a link that opens the user's mobile phone's mail app. The sender's mailbox is already filled with the project's mailbox, which is tiballi.project@gmail.com, as displayed in Figure 5.1.4A.

5.1.4.2 "Tell a Friend" Functionality:

The "Tell a Friend" feature is an extra secondary feature that lets users share the app with their friends and family. This feature is especially helpful when a user finds a useful app and wants to tell others about it. Users can click the "Tell a Friend" button in the Navigation Drawer Menu to share the app via social media or email, as displayed in Figure 5.1.4B. Along with the app link, users can also include a personalized message to make the recommendation more persuasive.

5.1.5 Alter the option in the middle of a recording

Figure 5.1.5 illustrates a warning message that appears when a user is in the process of recording one word or multiple words and wishes to alter the option from "Word" to "Category" or vice versa. Specifically, this message is displayed to inform and prevent the user from losing recordings made in this recording stream prior to completing the final step. The user is then presented with two options, "Cancel" and "Proceed." The "Cancel" option returns the user to the previous option without making any changes, Figure 5.1.5A, while the "Proceed" option changes the option, and it is possible that not all recording files will have been saved as it is showed in Figure 5.1.5B.

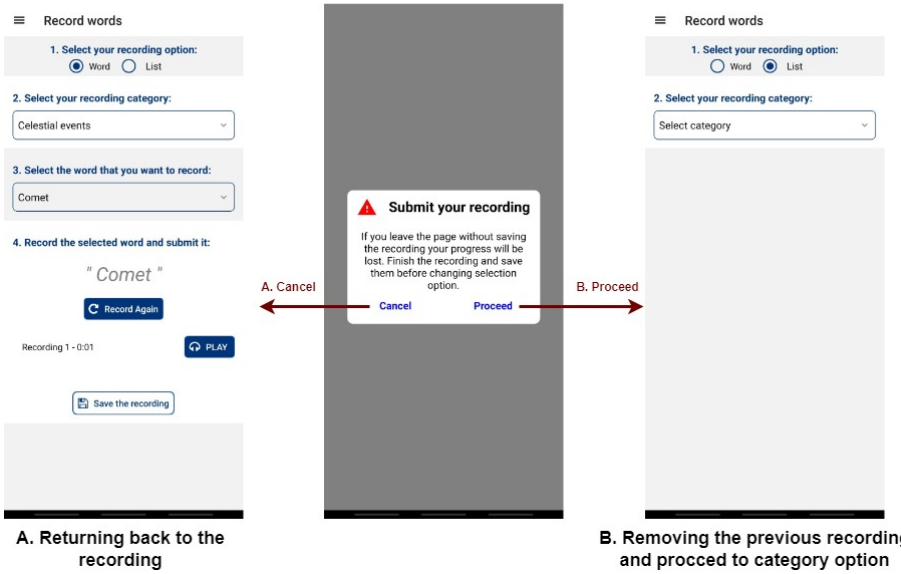


Figure 5.5: Warning message after changing recording option in the middle of the recording

5.1.6 About us page:

The third option on the navigation drawer, "About us", provides the user with access to a plethora of project-related information. This includes information about the project's action area and its future goals. As depicted in Figure 5.6, an excerpt from this subpage provides a glimpse of the wealth of information the user can access. In addition, the "About us" section is an indispensable resource for those seeking a deeper understanding of the project and its objectives. It is a valuable instrument for monitoring the project's progress and ensuring that all stakeholders are involved. Overall, the "About us" option provides a comprehensive and informative view of the project, making it an essential part of the menu for those interested in the development of the project.

Figure 5.6: View of the "About us" page

5.1.7 Record a single word process

In the event that the user desires to record a single word, as indicated by the selection of the "one word" option within the ratio group, as indicated in Figure 5.1.7.2A, they have the ability to choose the category in which the word it is contained, as displayed in Figure 5.1.7.2B. Upon selecting a category, such as "Seasons" an additional option will become available, as illustrated in Figure 5.1.7.2C. This allows the user to view the available words and choose and record the desired word. As depicted in Figure 5.1.7.2D, upon recording the word "Spring", by pressing the "Start recording" button, the options to "Record again" and "Save the recording" and "Play" will appear, as outlined in sub-sections 5.1.7.1, 5.1.9 and 5.1.7.2 respectively.

5.1.7.1 Recording again a word

After a user records a word for various reasons (wrong word, super noisy background, something unexpected happened, etc.), they might want to re-record it. So, the user can delete the previous recording and replace it with a new one before it is saved locally by clicking on the "repeat" icon on the right side of the recording bar or by choosing "Record again". The status of the recording will change by removing the already-recorded file and displaying the message that is shown in Figure 5.1.7.2E while recording again.

5.1.7.2 Play a recording

The "Play" button allows the user to listen to the recording they have made before deciding whether to save it or re-record it, as displayed in Figure 5.1.7.2G. This feature can be particularly useful for users who want to ensure that their recording is of good quality before submitting it for voice-to-text processing. If they are satisfied with the recording, they can choose to save it by selecting the "Save the recording" button. If, on the other hand, they are not satisfied with the recording, they can choose to re-record it by selecting the "Record Again" button, as displayed in the Figure 5.1.7.1.

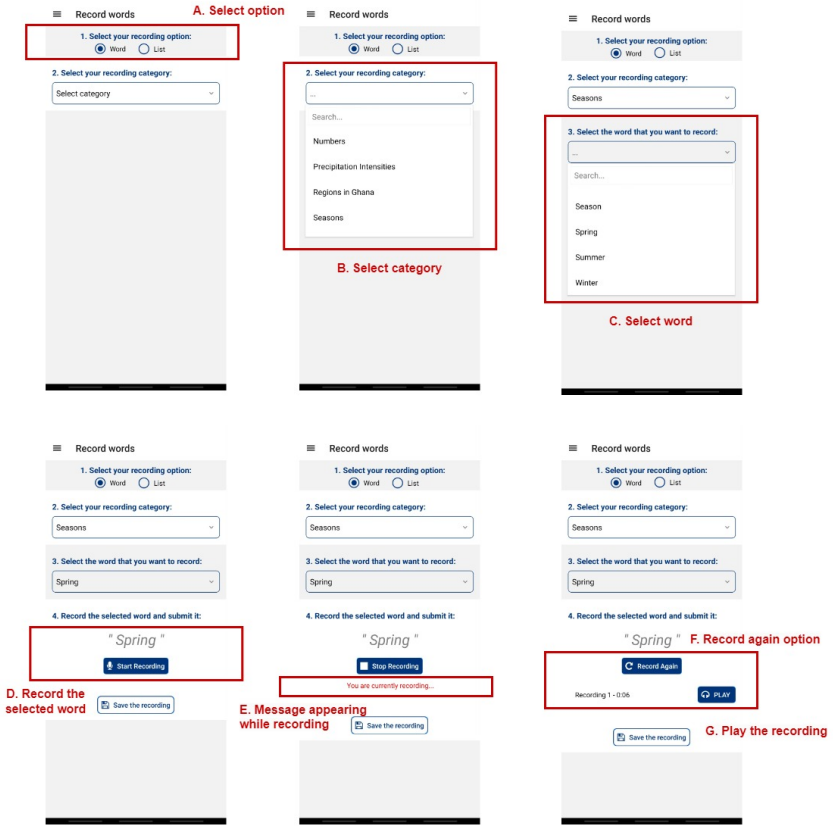


Figure 5.7: Screen for recording the one-word option

5.1.8 Record a category of words process

In the event that the user desires to record a specific category of words, for example, by selecting the "Category" option within the ratio group, they have the opportunity to do so from the home page, as depicted in Figure 5.1.8.3. As illustrated in the Figure, once the user has selected a category, such as "Colours," the first word within that category will be immediately displayed, in this case, "Blue." Upon selecting the category, additional options become available, as depicted in the Figure. These options include "Skip the word," and "Save and Continue" as outlined in subsections 5.1.8.1 and 5.1.8.2 respectively. Upon reaching the final word in the category, an additional option, "Finish and Submit" as outlined in subsection 5.1.9, will become available.

5.1.8.1 Skipping a word recording

When a user selects "Category" option a set of words is requested to be recorded. As a result, some of them may be unfamiliar to them, and they may be unable to accurately

record them. As a result, in order to eliminate the possibility of an incorrect recording, the user can select the button "Skip the word" and not record anything. Automatically, the word will change to a new one if there are more words left, as showed in Figure 5.1.8.3 in middle position.

5.1.8.2 Continue to the next word (temporal saving)

While the user is under the option "Category" a list of words must be recorded in order to reach the final step and be able to select the "Finish and Submit" button so all the words are saved locally. Therefore, after a user record a word, they have the option to click the button "Save and Continue" button which appears in Figure 5.1.8.3B, so the recording is saved temporarily and a new word appears in order to be recorded.

5.1.8.3 Empty submission

Figure 5.1.8.3D illustrates an additional potential warning message that could appear on the user's screen. Specifically, it appears when a user attempts to save a word in "category" mode without having previously recorded something (an list of empty files). The only option available to the user is to select "Okay" and start the process of recording the category again from the beginning.

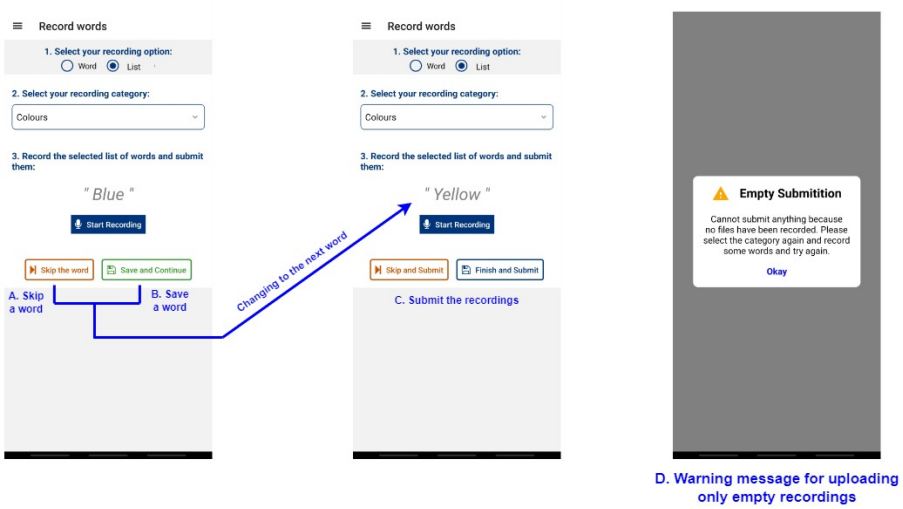


Figure 5.8: Screen for recording the category-words option

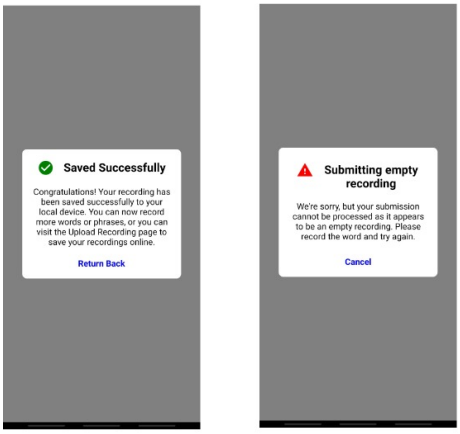
5.1.9 Save a recording

After pressing the button "Save the recording" and the situation following a normal/without any errors course, as described in subsection 5.1.9, the recording is saved on the user's local device.

5.1.9.1 Save a recording

Successful saving: After recording a word and without causing any warning or error on the screen, a pop-up will appear notifying the user that the recording was successfully done and saved locally, as shown in Figure 5.9A.

Unsuccessful saving: If the recording fails to save despite following the normal course of action as described in above subsection, the user's local device may not have sufficient storage space or there could be a technical issue with the application. It is recommended to check the available storage space and try saving the recording again. If the problem persists, please refer to the contact us subsection to for assistance, as described in subsection 5.1.4A.



A. Successful message B. Unsuccessful message

Figure 5.9: Possible messages after recording a file

5.1.10 Managing the recording

When a user attempts to upload their recording to the server, they are directed to the "Upload recordings" section of the navigation drawer. On this page, the user can view all of their recordings that have not yet been uploaded to the server, as shown in Figure 5.1.10.2A.

5.1.10.1 Available actions for each recording

All files have a unique name, the date the file was created, and two editable options for the user. Specifically, there are the options "Play" and "Delete"; with the former, the user can listen to their recording, and with the latter, they can delete a recording they do not wish to upload. Particularly, when the user selects the "Delete" option, the screen displays a warning message, as depicted in Figure 5.1.10.2A1, asking the user to confirm their decision as it is a permanent deletion. If the user selects "Delete" the recording is removed and they are returned to the list of recordings with one less line. Alternatively, if

the user selects the option "Cancel," they are returned to the same screen with no changes made.

5.1.10.2 Process of uploading the recordings

Then, if the user wishes to upload their files to the server, they select only the check boxes for the files they desire, as displayed in Figure 5.1.10.2B and click "Upload selected".

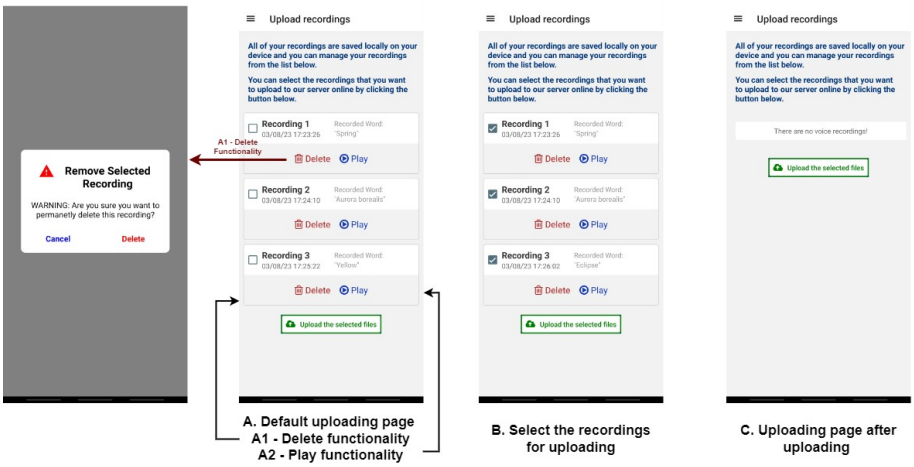


Figure 5.10: Screen for uploading recording audios to server

5.1.11 Checking the internet connection

As shown in Figure 5.1.11 the check begins to determine whether or not there is an internet connection. Depending on the status of the internet connection, the user's screen either displays Figure 5.1.11.2.1, where there appears to be no connection and the "TRY AGAIN" button is displayed, or Figure 5.1.11.2.2, where the internet connection is functioning properly. The "Upload" option begins the process of uploading the selected files to the server, assuming the connection is stable. While the loading bar in Figure 5.1.11.2.2.1 indicates the status of the upload and how close it is to completion, an informational message is displayed to the user once the status reaches 100%. Assuming the user has selected all files to upload to the server, they are returned to a screen similar to Figure 5.1.10.2C where there are no files and an appropriate message is displayed.

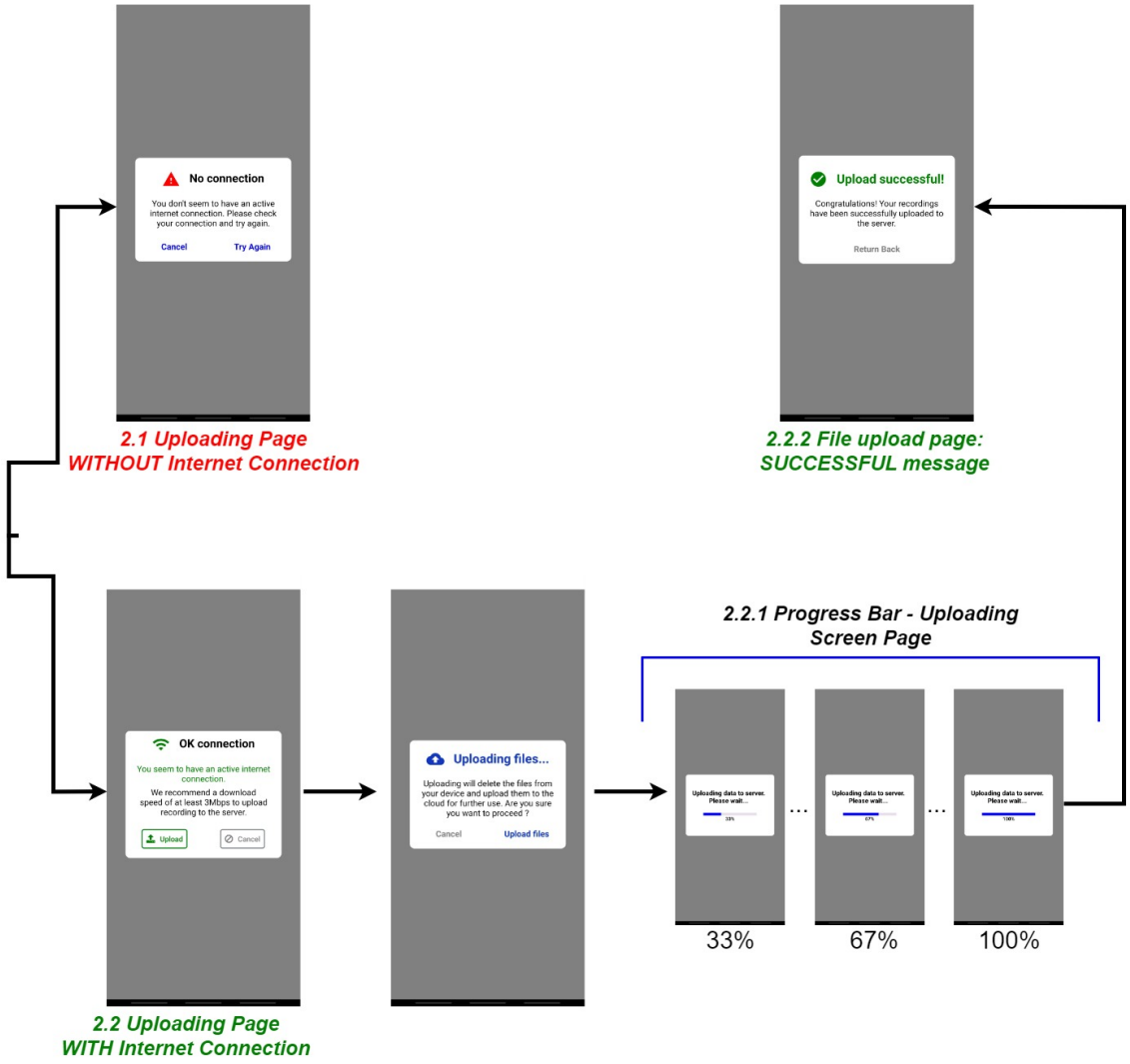


Figure 5.11: Pop-up for checking the internet connection

5.1.12 Required permissions

Two permissions are necessary for the app to function properly: microphone access and storage access.

5.1.12.1 Microphone Permission:

This permission allows the app to capture audio, which is saved locally on the device and potentially uploaded to the server if an internet connection is available. This permission is critical for the app's functionality, and without it, users will not be able to use the app

as intended.

5.1.12.2 Storage Access Permission:

This permission enables the app to store data(recordings) locally on the device and facilitates sending recordings to the server when an internet connection is available. This permission does not grant the application access to any other files or data on the device. It can only access the files it creates.

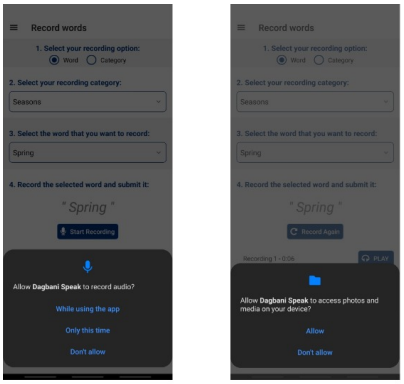


Figure 5.12: Pop-up for permissions

5.2 Differences between the mobile app and the web app

This section will examine the fundamental distinctions between the mobile application and web application created for the project. The main attention is directed towards three fundamental elements, namely accessibility, upload capability, and device adaptability. Comprehending the distinctions between these platforms is crucial for understanding the distinct characteristics and benefits that each one provides.

Aspect 1: Accessibility

Mobile app: The mobile application offers the convenience of offline accessibility, enabling users to document vocabulary without the need for an internet connection. This characteristic is especially advantageous in regions with restricted connectivity or for individuals who frequently travel. The provision of offline access guarantees uninterrupted data collection, thereby enhancing the convenience and flexibility of users in contributing to the language corpus, irrespective of their location or network availability.

Web app: The web application, on the other hand, necessitates an internet connection for both accessing and recording words. Although the web application's utility may be restricted in regions with inadequate or absent connectivity, it confers benefits in terms of instantaneous access to updates and the most recent iteration of the application. The web application can be accessed by users through contemporary web browsers, rendering it a convenient option for individuals who predominantly operate on desktop computers or

favour interfaces based on browsers.

Aspect 2: Functionality for Uploading Data

Mobile app: In the context of mobile technology, the recorded words are directly saved to the storage of the mobile device by the mobile application subsequent to the recording process. Subsequent to the initial recording, individuals have the option to upload said recordings at a later time, contingent upon the availability of a WiFi network connection. The utilisation of a two-step process enables streamlined administration and uploading of recorded vocabulary, affording versatility and adjustability to diverse connectivity scenarios.

Web app: The web application utilises Firebase storage as the primary repository for storing recorded words. The aforementioned cloud-based storage solution guarantees the secure administration and retrieval of the recorded verbal data. The implementation of Firebase storage obviates the necessity of local storage on the user's device and offers a scalable and dependable solution for managing a considerable number of recordings in the web application.

Aspect 3: Compatibility of Devices

Mobile app: The mobile application has been developed to ensure compatibility with particular operating systems, such as Android. The implementation of this focused strategy guarantees enhanced efficiency and a cohesive user interface across compatible devices. The mobile application can optimise the recording process and improve the overall user experience by utilising the functionalities of particular operating systems, such as hardware integration and device features.

Web app: The web application provides a higher degree of device compatibility in contrast to the mobile application. It is accessible via contemporary web browsers across diverse devices and operating systems. The ability of the web application to function seamlessly across multiple platforms allows users to access it on various devices such as desktop computers, laptops, tablets, or smartphones, irrespective of the underlying operating system. The extensive accessibility of the web application enables a wider user demographic to participate in the language corpus, without being constrained by the restrictions typically associated with particular operating systems.

The successful execution of our particular project in Ghana is contingent upon the deployment of both the mobile and web applications. The provision of both mobile and

web application alternatives enables us to accommodate a broader spectrum of users and circumstances, thereby promoting inclusivity and optimising the efficacy of our language corpus development endeavours.

To conclude, the mobile application and web application demonstrate notable variations with regards to accessibility, upload capability, and device adaptability. The mobile application facilitates offline recording, whereas the web application is dependent on an internet connection. The mobile application stores recordings directly onto the device's storage, allowing for greater adaptability in terms of future uploads. On the other hand, the web application employs Firebase storage as a means of centralised and streamlined data management. Furthermore, the mobile application exhibits compatibility solely with particular operating systems, whereas the web application provides cross-platform compatibility.

Comprehending these distinctions enables us to make knowledgeable determinations regarding which platform to employ, contingent upon user contexts and prerequisites. Through a careful analysis of the distinctive characteristics and benefits of each platform, we can optimise the efficacy of our language corpus construction initiative and facilitate the ability of users to make valuable contributions to the advancement of the artificial intelligence model.

5.3 Design and Usability Rules

One of the primary objectives established by the team was to ensure that the system, particularly the interface and functionality, would be straightforward and comprehensible to users of varying levels of expertise, given the diverse user base. Both the mobile and web apps adhere to Nielsen's ten heuristics for user interface design and National Aeronautics and Space Administration (NASA)'s web and app design guidelines(7). The aforementioned techniques, namely the heuristic evaluation and the Human Integration Design Processes (HIDP), were respectively introduced by Jacob Nielsen in the 1990s and by the National Aeronautics and Space Administration International Space Station Program Johnson Space Center Houston, Texas in 2014 (8). Adhering to the 10 Nielsen heuristics for user interface design is imperative in order to develop a prosperous website that offers a favorable user experience, facilitates the attainment of project objectives, and distinguishes the website from its competitors. Following these heuristics can enhance user experience, minimize user frustration, improve usability, increase business success, and result in time and cost savings.The following is a comprehensive description of the design and usability guidelines that were established and executed on the platform:

Visibility of system status: The system provides users with clear and concise feedback on the recording process, including progress indicators and success messages. During the uploading process, for instance, there is a progress bar and a message indicating that the recording was successfully submitted.

Match between the system and the real world: The application employs user-familiar language and design elements that reflect how users interact with the real world. For instance, the system employs recognizable icons and buttons to represent actions such as recording (the microphone icon), playing (the play icon), and saving (the floating disk icon). An example of the utilisation of the icons in the system can be also seen in the Figure 5.13

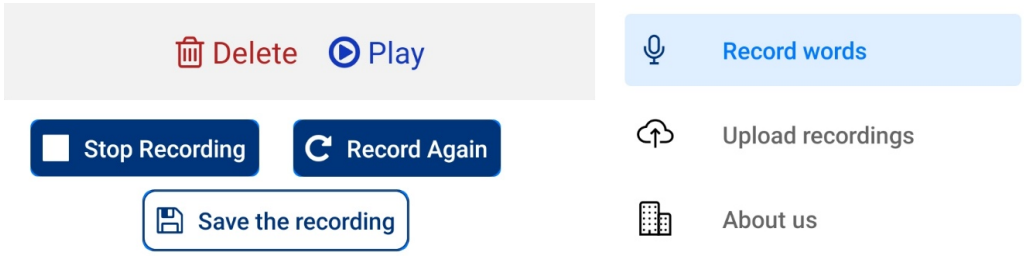


Figure 5.13: Example icons of rule: "Match between the system and the real world"

User control and freedom: The app gives users the ability to undo or cancel actions if they make a mistake or change their minds. For instance, if a user realizes they have made a mistake, the system provides a button to reorder the words.

Consistency and standards: The app adheres to established design patterns and standards to facilitate user navigation. For instance, the system uses consistent button design and placement throughout the application.

Error prevention: The application prevents errors by providing users with clear instructions and prompts. In some instances, the system prevents the user from performing an action they are not authorized to perform. For instance, the user is required to provide a username and gender; therefore, it is impossible to record a single word without providing this information.

Recognition rather than recall: Utilizing recognizable and simple-to-understand design elements, the application reduces the need for users to remember information. For example, the system labels buttons and controls with clear and descriptive names.

Flexibility and efficiency of use: The app enables users to perform tasks in a variety of ways and at their own pace, thereby maximizing their flexibility and efficiency. For instance, both systems offer two distinct options, 'word' and 'list, which ultimately record the same words, but their distinction is based on the user's personal preference.

Aesthetic and minimalist design: The application's aesthetic and minimalist design is visually appealing and does not detract from its primary function.

Help users recognize, diagnose, and recover from errors :Assist users in identifying, diagnosing, and recovering from errors: The application provides users with helpful error messages and recovery suggestions. For instance, if a user attempts to upload an audio file from the mobile app without an active internet connection, they will receive a pop-up prompting them to try again. The system will then recheck for an internet connection and recover from the previous error.

Help and documentation: Help and documentation: If users require additional assistance, the app provides access to help and documentation. For instance, there is documentation on how to use the application effectively, as well as an email address where users can send questions.

5.4 Helpful JavaScript Files

The organisation of categories and vocabulary in mobile and web applications is streamlined through the utilisation of JavaScript and Node.js. The aforementioned technologies offer a sturdy infrastructure for augmenting the user experience and optimising data management. The code can be accessed via the Github hyperlink and the subdirectory labelled "mobile-app/javascript_scripts". The utilisation of JavaScript and Node.js has facilitated the integration of diverse features to facilitate the management of categories and words in the mobile and web applications. This section delineates the salient characteristics and particulars of the implementation, along with pertinent details of the implementation procedure.

1. **Viewing Categories:** JavaScript and NodeJS enable the retrieval and presentation of pre-existing categories in the application. The utilisation of the file system module in JavaScript code enables the extraction of essential information from the available category files. The presented console output displays categories and their

corresponding word lists, thereby enabling users to navigate through them. In order to ensure a user-friendly experience, the system displays suitable messages when a selected category is not found. An instance of this phenomenon is observable in Figure 5.14 at the begginig of the default printing after exectuting the JavaScript file.

2. **Creating a New List of Words:** The seamless creation of new word lists is facilitated by the implementations of JavaScript and Node.js. By leveraging the console and file system modules within the Node.js environment, the JavaScript programming language is able to acquire user input and subsequently persist it to a text file, thereby constituting a novel lexicon. The implementation encompasses the incorporation of error handling mechanisms and provision of suitable feedback messages to facilitate successful creation of lists. As illustrated in Figure 5.14, an instance of generating a fresh lexicon, titled "Seasons," is presented. The user inputs the value of 0 to initiate the creation of a novel word list, followed by the input of the corresponding category name. The Node.js server furnishes the requisite routes and endpoints for managing data and executing file writing operations. This methodology guarantees the effective generation and retention of novel lexicons, obviating the requirement for a visual user interface.

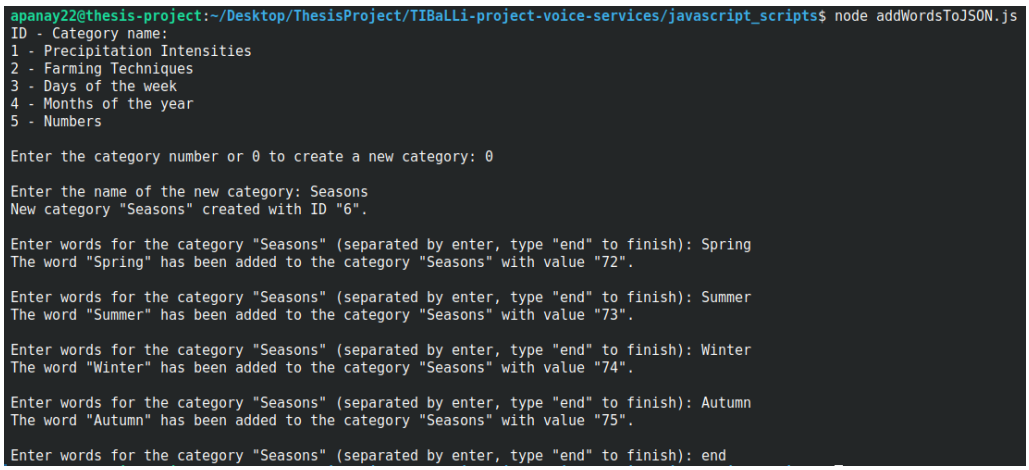


Figure 5.14: Example of the console screen after adding the new category named "season"

3. **Adding Words to the List:** The incorporation of words into pre-existing lists is facilitated by the utilisation of JavaScript and Node.js implementations, which streamline the process for users. By means of the console, users are able to input

novel words, which are then subjected to validation and subsequently appended to the corresponding word list file by the JavaScript code. The utilisation of the file system module in Node.js is employed for the purpose of reading and writing data in the implementation. Incorporation of error handling mechanisms and provision of feedback messages are crucial in ensuring a seamless process of adding words. An instance of this phenomenon is observable in Figure 5.15. Upon executing the "addWordsToJSON" file, the user proceeds to designate a category and subsequently provides words as input via the console. The aforementioned procedure is iterated until the term "end" is provided as an input. Figure 5.14 illustrates that upon creating a new category, the programme prompts for potential words that align with said category.

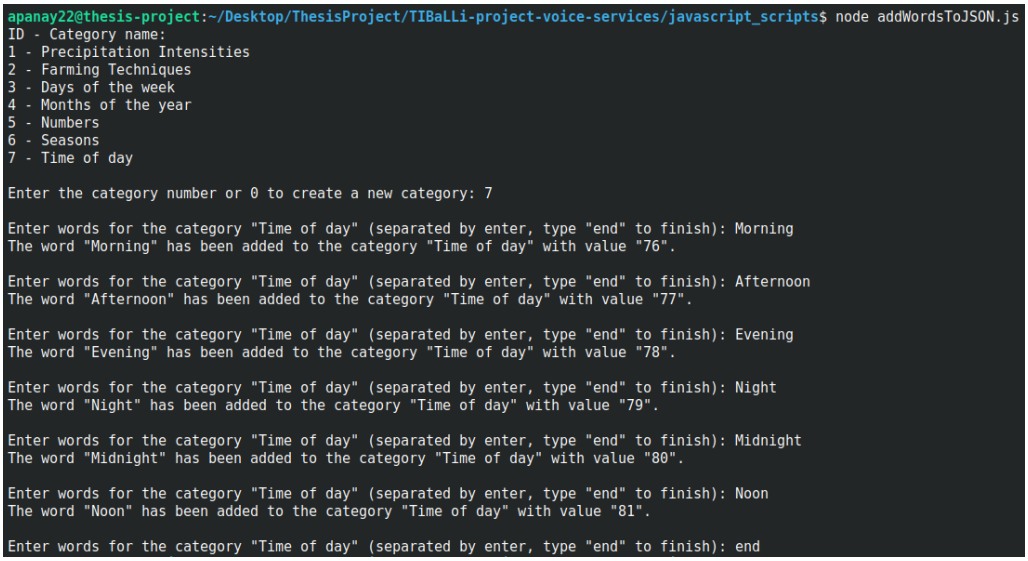


Figure 5.15: Example of the console screen after selecting a category and adding new words

4. **Detailed Information about Uploaded Recordings:** JavaScript and Node.js are crucial components in the retrieval and display of comprehensive data pertaining to uploaded recordings. The utilised approach involves the utilisation of the file system module for the purpose of accessing the recording files and extracting pertinent information, including author particulars and upload timestamps. Upon execution of the "downloadFirebaseData" file, copious amounts of information are presented on the user's console and are also exported to a text file. The JavaScript programme utilises console formatting techniques to effectively present and provide users with extensive information pertaining to the uploaded recordings, as shown in Figure 5.16.

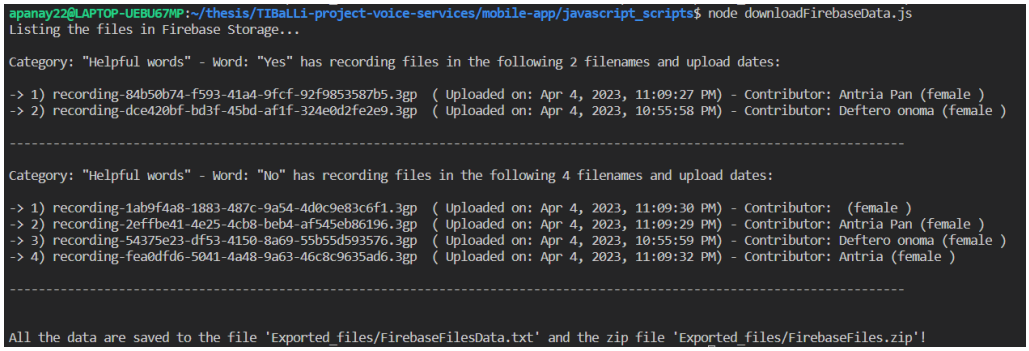


Figure 5.16: Example of the console screen after getting the metadata of the uploaded recordings

To summarise, the utilisation of JavaScript and Node.js offers a robust framework for proficient management of categories and words within the project. The system facilitates the creation of new word lists, addition of words, retrieval of existing categories, and display of detailed information about uploaded recordings through the utilisation of advanced technologies. The implementation of console interactions and file system operations that export txt files is designed to achieve effective data management and enhance user experience.

Having successfully implemented the language preservation platform, the "Implementation" chapter culminates with an extensive exploration of the mobile and web app's functionalities, JavaScript files, design rules, and usability guidelines. The next chapter, "Field Evaluation," delves into the assessment phase, where the platform's performance, functionality, and usability undergo rigorous testing. Subchapters include Experimental Design, Functionality Testing, Usability Testing, Performance Assessment, and comprehensive evaluation of results and insights derived from the evaluation process.

6

Field Evaluation

A set of experiments was devised and implemented to assess the efficacy and efficiency of the crowdsourcing application developed for the purpose of preserving and revitalising minor indigenous languages. This section provides an analysis of the experimental design, presents the acquired findings, and illustrates their significance in assessing the assertions posited in the introduction. The evaluation results are subsequently employed to substantiate design decisions and evaluate the impact of various elements of the application's design on the overarching objectives.

6.1 Experimental Design

The primary objective of the experimental design was to evaluate fundamental elements of the crowdsourcing application, encompassing functionality, usability, performance, and user satisfaction. A series of experiments were carried out:

6.1.1 Functionality Testing

A comprehensive set of functional tests was performed to validate the app's core functionalities, such as word recording, playback, selection, categorization, and offline capabilities. By meticulously examining these functionalities, we aimed to ensure that the app operates as intended and meets the specific requirements of language resourcing.Presented below is an exhaustive compilation of potential functional testing scenarios for the crowdsourcing application:

1. **User Registration:**

- Test for proper validation of user input fields (e.g., radio button - gender, text - name).
- Ensure that appropriate error messages are displayed for invalid inputs.

2. **Word Recording - Single Word Category Selection:**

- Conduct an examination of the feature that allows for the re-recording of a specific word selection subsequent to the initial recording, without the need for immediate submission.
- Conduct an assessment of the operational capabilities pertaining to the auditory reception of spoken language.
- Conduct an examination of the operational capabilities pertaining to the modification of the recording option in the presence of an ongoing recording process.
- The ability of users to choose a word from a given list and subsequently document their pronunciation is to be confirmed.
- It is imperative to ensure the accurate preservation of the recorded audio in relation to the designated word.
- It is imperative to ensure the efficacy of protocols in managing errors or interruptions that may occur during audio recording.

3. **Word Recording - Category of Multiple Words:**

- Conduct an assessment of the operational capabilities pertaining to the recording of multiple words within a designated category.
- The functionality of allowing users to choose a category and subsequently access a list of words falling under that specific category is to be confirmed.
- Assess the capacity to successfully traverse to the subsequent and preceding word without encountering any errors.
- It is imperative to establish a clear correspondence between each recorded audio and its respective word within the designated category.
- Conduct an assessment of the operational capabilities pertaining to the auditory playback of recorded speech.
- Conduct an assessment of the operational capabilities pertaining to the modification of the recording setting while a recording is in progress.

- This study aims to assess the efficacy of protocols in managing errors or interruptions encountered during audio recording.

4. **Offline mode:**

   - The objective is to ascertain the capability of users to access and perform offline operations such as listening to and deleting previously recorded words and categories.
   - Evaluate the capacity to capture verbal expressions in an offline setting and store them within a local storage medium.
   - It is imperative to synchronise offline recordings with the server upon restoration of internet connectivity.

5. **Upload Functionality:**

   - Conduct an assessment of the upload functionality by testing the process of uploading recordings to the Firebase platform.
   - The ability for users to select multiple recorded words for upload should be confirmed.
   - Conduct an evaluation of the upload procedure, with the objective of verifying the successful transfer of files to the Firebase platform.
   - The validation process ensures that the uploaded recordings are appropriately linked to the corresponding user and category.

6. **Error Handling:**

   - Conduct testing to assess the application's response to different error conditions.
   - Validate that appropriate error messages displayed in the event of unsuccessful recordings or uploads.
   - Conduct testing to evaluate the efficacy of error handling mechanisms in the context of network connectivity issues that may arise during the processes of recording or uploading.
   - Verify the generation and documentation of error logs for the explicit purpose of troubleshooting.

6.1.2 Usability Testing

A usability evaluation was conducted, wherein usability testing sessions were carried out with a diverse group of participants who were representative of the target user base. The participants were assigned specific tasks to complete within the application, and their interactions and feedback were systematically observed and documented. The evaluation facilitated the assessment of the app's ease of use, intuitiveness, and overall user experience, thereby ensuring its accessibility and user-friendliness for individuals who contribute to the language database.

6.1.3 Performance Assessment

The efficacy of the application was evaluated based on its responsiveness, speed, and resource consumption. On a variety of devices and operating systems, the app's efficacy and responsiveness under varying conditions were evaluated via performance evaluations. On a variety of mobile devices and operating systems, performance tests were conducted to evaluate the app's efficacy and responsiveness under varying conditions, ensuring its optimal performance for users. Some examples can be shown in Figures 6.2 and 6.1. Specifically, figure 6.2 showcases a compilation of screenshots that serve as illustrations of the web application's ability to adapt and respond effectively to different screen resolutions. This feature guarantees a smooth and uninterrupted user experience across a wide range of devices. In addition, figure 6.1 exhibits a collection of screenshots that exemplify the web application's uniform rendering and operational capabilities across various web browsers (e.g., Google Chrome, Microsoft Edge, and Safari), thereby guaranteeing compatibility and accessibility for a wide range of users.

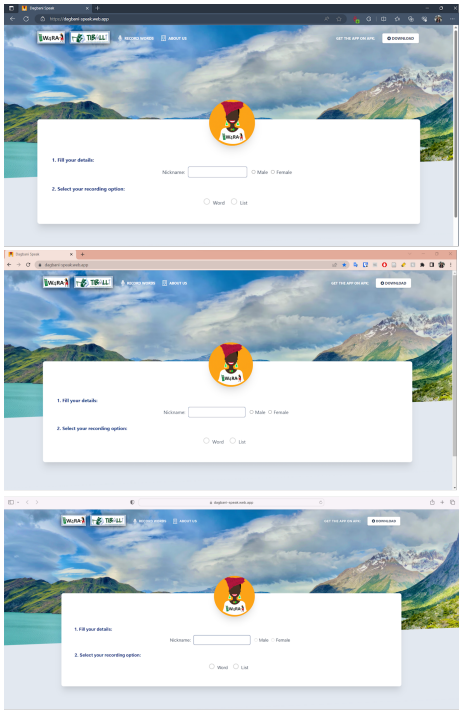


Figure 6.1: Web-app screenshots showcasing compatibility with different browser environments

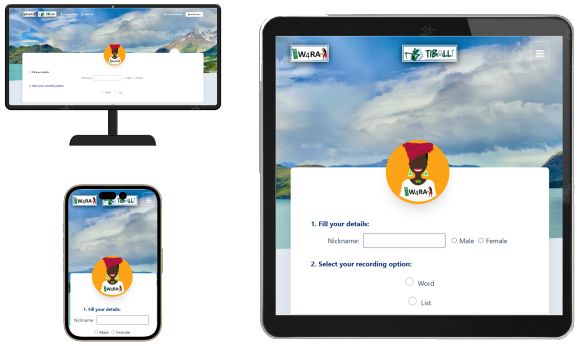


Figure 6.2: Web-app showcasing responsive design with various screen resolutions

6.2 Results and Evaluation

The experimental findings yield significant insights regarding the performance and efficacy of the crowdsourcing application that was developed. The results are employed to assess the assertions posited in the introduction and to evaluate the overall efficacy of the application in achieving its intended goals.

6.2.1 Functionality Evaluation

The results of the functionality testing indicate that the application effectively performed tasks such as recording and saving user pronunciations, correctly associating recordings with chosen words or categories, and efficiently managing error situations. The obtained results serve to confirm the efficacy of the application, thereby substantiating the assertion that it offers a dependable platform for language resourcing.

6.2.2 Usability Evaluation

The usability evaluation encompassed a series of testing sessions that yielded significant insights into the user interface, navigation, and overall user experience of the application. The participants expressed a significant degree of satisfaction regarding the app's intuitive nature and user-friendly interface. The feedback received from these sessions proved instrumental in identifying areas that required improvement, subsequently resulting in refinements being made to the design of the application and ultimately enhancing its usability.

6.2.3 Performance Evaluation

The performance assessment revealed that the application consistently demonstrated high levels of responsiveness and efficiency, even when subjected to fluctuating network conditions. The application's offline functionalities were deemed dependable, enabling users to locally record and store their contributions and subsequently synchronise them with the server when an internet connection became accessible.

In general, the outcomes derived from the assessment experiments offer significant substantiation for the assertions posited in the introductory section. The app's effectiveness in facilitating language resourcing and preservation efforts is affirmed by the positive feedback it has received, as well as its successful functionality, usability, and performance. The assessment outcomes also played a pivotal role in providing justification for particular design decisions and evaluating the impact of various elements of the application's design on the overarching objectives. The feedback obtained from usability testing played a significant role in the iterative improvement of the application's user interface and navigation, with the aim of enhancing user satisfaction and optimising the overall user experience. Through the implementation of these experiments and subsequent analysis of the obtained outcomes, we have acquired significant and valuable knowledge pertaining to the inherent strengths and potential areas for enhancement of the crowdsourcing application. The evaluation process served to validate the efficacy of the application, while also offering valuable feedback to inform its ongoing development and improvement.

The "Field Evaluation" chapter culminates with a comprehensive assessment of the language preservation platform through functionality testing, usability evaluation, and performance assessment. It also addresses pertinent research questions regarding audio cleaning techniques and their influence on ASR accuracy. The following chapter, "Getting Started

with Machine Learning," delves into an extensive exploration of audio cleaning methods, historical ASR developments, and the impact of various factors on ASR accuracy, presenting valuable insights for the language preservation platform's improvement.

7

Getting Started with Machine Learning

Improving the quality of the audio data is crucial to this study, as it has a direct impact on the performance of the machine learning models and subsequent language preservation efforts. The research aims to ensure that the collected audio files meet the highest standards for seamless integration into the machine learning dataset by implementing advanced audio cleaning techniques, such as noise cancellation and removal of empty noise. Achieving superior data quality lays the groundwork for precise and dependable processing, paving the way for the success of language preservation and voice-based agricultural assistance initiatives.

7.1 Audio Cleaning Techniques of User's Recordings

Imagine a world where every spoken word could be effortlessly transformed into an accurate written text, revolutionizing communication and accessibility across languages and cultures. The aforementioned accomplishment is facilitated by ASR systems, which transform oral communication into textual representation. The swift progressions in ASR technology have propelled us towards an era where spoken language can be effortlessly converted into precise written text. The improvement of ASR systems and resolution of intricate speech scenarios have emerged as a central area of focus in research and development. The process of converting speech to text is accompanied by a number of obstacles, encompassing a variety of speech patterns and difficulties arising from environmental factors, background noise, and variations in pronunciation.

Nevertheless, when considering low-resource settings and indigenous languages with limited resources, the process of converting speech to text encounters various obstacles. The aforementioned data sources encompass modest data corpuses, data procured via mobile devices amidst environmental noise, and dialectal discrepancies arising from regional dissimilarities. The restricted accessibility of data poses a challenge to the training of models, whereas audio captured through mobile devices may exhibit noise and distortions. Accurate transcription of regional dialects necessitates language models that are capable of capturing subtle nuances. Mitigating these challenges requires novel methodologies for minimising noise interference, devising effective data gathering approaches, and adapting to diverse dialects. The present study investigates the difficulties encountered in the process of audio cleaning for the purpose of converting speech to text. The study places special emphasis on identifying strategies to alleviate these challenges and integrating them to tackle intricate speech scenarios.

The act of speaking is a fundamental means of human communication that facilitates the transfer of concepts, information, and affective states. The emergence of ASR systems has significantly transformed our capacity to transcribe spoken language into written text, thereby facilitating diverse applications such as transcription services, voice assistants, and other related domains. The precision of ASR is significantly contingent upon the excellence of the audio input, which can frequently be undermined by intricate speech situations. ASR is a multidisciplinary field that involves various areas of study, including linguistics, signal processing, machine learning, and human-computer interaction. The process entails unraveling the complex correlation between acoustic waves and linguistic expressions, analyzing the nuances of human verbal communication, and utilizing advanced technological tools to enhance the precision of automatic speech recognition.

The quest for precise transcription has been a driving force behind human progress throughout the course of history. Throughout history, there has been a persistent drive to establish a connection between oral and written forms of communication, ranging from the carving of symbols into stone by ancient civilizations to the development of the printing press. Currently, we are at the forefront of a new era, utilizing the capabilities of machine learning and artificial intelligence to reveal the mysteries of spoken language and facilitate effortless conversion of voice-to-text.

The present study aims to examine the key determinants that exert a substantial impact on the accuracy of ASR. The study places a specific emphasis on the influence of extraneous noise, speaker diversity, contextual factors, and phonetic mispronunciations. The aforementioned factors present significant obstacles in attaining dependable conversion of

speech to text. Furthermore, the present study investigates the benefits and compromises linked to the utilization of diverse amalgamations of audio cleansing methodologies in the preprocessing of audio data for machine learning algorithms utilized in ASR.

The objective of this study is to offer useful insights and practical strategies to improve the accuracy of ASR in real-world applications by conducting a comprehensive analysis of intricate speech scenarios. By means of thorough examination, systematic experimentation, and meticulous assessment of extant literature, this study makes a valuable contribution to the progress of ASR technology and provides a basis for enhanced voice-to-text conversion. Our objective is to narrow the divide between oral communication and written language, with the aim of unleashing the complete capabilities of automatic speech recognition systems. This will facilitate improved communication, accessibility, and innovative applications across a broad spectrum of domains, including transcription services and voice-activated technologies. With every progressive stride, we are approaching the realization of a seamless and precise voice-to-text transcription system.

The paper is structured as follows. In Chapter 7.2, we present an introduction that emphasizes the importance of audio cleaning in the context of voice-to-text conversion and provides a captivating overview of the research topic. Chapter 7.3 focuses on the research questions, delving into the primary factors that significantly affect the accuracy of ASR systems and discussing strategies to mitigate their impact. In Chapter 7.4, we conduct a comprehensive literature review, comprising two subchapters. The first subchapter offers background information on ASR systems, exploring their functioning, limitations, and challenges faced in complex speech scenarios. The second subchapter analyzes the primary factors affecting ASR accuracy, drawing from existing research and studies in the field. Chapter 7.5 details the methodology employed in this study, encompassing data collection, experimental setup, and evaluation metrics. In Chapter 7.6, we delve into audio enhancement techniques for ASR systems, examining the effectiveness of approaches such as noise reduction, speaker variability training sets, speaker normalization techniques, the integration of a variety of speakers, and language models. Chapter 7.7 presents the evaluation methodology, highlighting the experimental results and engaging in a thorough discussion and analysis. Chapter 7.8 offers a comprehensive examination of the results, discussing the advantages and trade-offs associated with the employed techniques. Finally, in Chapter 7.9, we conclude the paper by summarizing the key findings, reflecting on the learnings derived from this research, and proposing potential avenues for future improvements in audio cleaning techniques for voice-to-text conversion applications.

7.2 Historical Overview of ASR

ASR technology has a lengthy and extensive chronicle that encompasses numerous decades.The following graphic, Figure 2 illustrates the evolution of ASR from 1950 to 2020. The inception of ASR can be attributed to the initial years of the 1950s, during which scholars commenced investigating the feasibility of utilizing computers for speech recognition. Davis et al. (1952) reported that Bell Laboratories developed an ASR system in the 1950s that utilized a formant-based methodology to identify vowel sounds (9). Initially, speech recognition systems were primarily designed to recognize numerical inputs rather than linguistic ones. A decade subsequent to its predecessor, IBM unveiled "Shoebox," a language processing system capable of comprehending and generating responses to a vocabulary of 16 English words.

In the subsequent decades, the technology of ASR underwent further advancements, as scholars delved into diverse methodologies for speech recognition. The field of ASR experienced a significant advancement during the 1970s, when the technique of Hidden Markov Models (HMMs) was introduced for speech recognition by Baker in 1975(10). HMMs are a type of statistical model that enables the modeling of temporal dynamics in speech. This represents a significant improvement over prior methods that treated individual speech sounds as distinct entities. The HMM employed a probabilistic approach to estimate the likelihood of the unidentified phonemes constituting lexemes, rather than relying solely on phonetic features and auditory cues.

During the 1980s and 1990s, advancements in ASR technology were made through the creation of novel algorithms and models. A significant development during this timeframe pertained to the utilization of neural networks in the domain of speech recognition, as noted by Bourlard and Morgan (1994)(11). Neural networks are a machine learning model that is capable of recognizing patterns in data. Studies have demonstrated their efficacy in speech recognition, particularly when compared to HMM.

Currently, ASR technology is employed in a diverse array of applications, including but not limited to virtual assistants, speech-to-text transcription, voice search, and language translation. The field of ASR technology remains a dynamic domain of investigation, with persistent endeavors to enhance the precision, resilience, and efficacy of ASR frameworks.

As of 2001, the accuracy rate of speech recognition technology had reached 80%. Throughout the majority of the decade, there were limited technological advancements until the introduction of Google Voice Search in the 2010s. This innovation facilitated speech recognition for a vast number of individuals via an application and delegated processing capa-

bilities to data centers. Google has utilized data from a vast number of searches to enhance the precision of its services. Specifically, its English Voice Search System has integrated a corpus of 230 billion words. During this period, voice recognition applications such as Apple's Siri[1] were introduced, leading to a rise in consumer acceptance of conversing with machines through devices such as Amazon's Alexa[2] and Google Home[3]. Currently, there is a competition among prominent technology corporations to attain the most precise speech recognition system, with Google asserting a minimal error rate of 4.9 percent. The graphical representation presented herein has been extracted from Mary Meeker's 2017 Internet Trends report. The Figure 1 depicts the word accuracy rate of Google, which has recently surpassed the 95% benchmark for human accuracy.

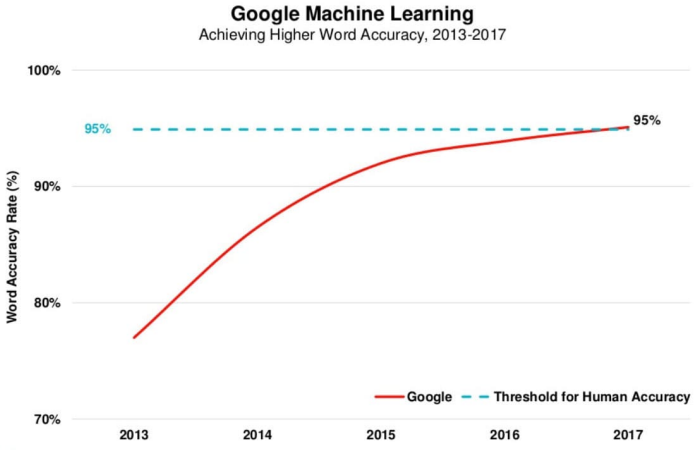


Figure 7.1: The evolution of Word Accuracy Rate through the years 2013-17

In recent times, the advancement of deep learning-based models, combined with transfer learning methodologies, has contributed significantly to the progress of ASR technology. Deep learning is a machine learning approach that employs intricate neural networks to perform data processing and analysis. Research has demonstrated its remarkable efficacy in the domain of speech recognition, as evidenced by studies conducted by Hinton et al. (2012)(12) and Hannun et al. (2014)(13). The utilisation of transfer learning in the instruction of ASR models for languages with limited resources not only enhances their efficacy but also surmounts the obstacles of inadequate data by utilising the abundant resources accessible for more prominent languages. Furthermore, transfer learning facilitates the customization of models to the particular linguistic features and accents of low-resource

[1] https://www.apple.com/siri/
[2] https://alexa.amazon.com/
[3] https://home.google.com/welcome/

languages, thereby augmenting the precision and resilience of ASR systems across a range of linguistic settings. The advancement of ASR technology has been facilitated by the increased accessibility of voluminous datasets and the enhanced computational capabilities of contemporary computing systems, commencing from 2014. Convolutional Neural Networks (CNNs) and Recurrent Neural Networkss (RNNs) are currently employed in the field of speech recognition. Specifically, CNNs are utilized for efficient feature extraction from speech signals (14), while RNNs are applied for sequence modeling and classification. The utilization of transfer learning methodologies has been employed to enhance the performance of ASR. This involves training a model on a vast dataset and subsequently fine-tuning it on a smaller dataset that is specific to the task at hand (15). ASR technology is currently being employed in various industries such as healthcare, legal, automotive, and education. Recent advancements in deep learning and transfer learning have led to substantial progress in ASR technology, resulting in the development of more precise and resilient speech recognition systems. These developments have also created new prospects for ASR technology in diverse applications.

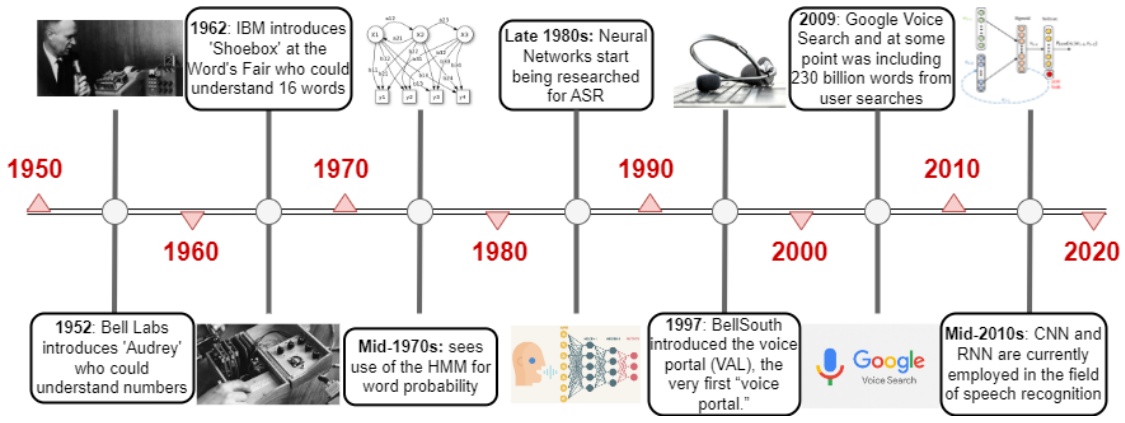


Figure 7.2: History of Automatic Speech Recognition through the years 1950 - 2020

7.2.1 Historical Overview of Audio Cleaning

The procedure of audio cleaning, which is intended to enhance the quality of speech signals for the purpose of ASR, has undergone notable progressions throughout the years, as shown in Figure 3. This segment presents a comprehensive historical survey of the development of audio cleaning methodologies, emphasising significant landmarks and contributions within the discipline.

During the initial phases of audio cleaning, the primary emphasis was on fundamental techniques for reducing noise. During the 1970s, scholars initiated an investigation into statistical models, such as spectral subtraction, as a means of mitigating noise interference in speech signals(16, 17). The aforementioned methodology entails the estimation of the noise spectrum and its subsequent subtraction from the spectrum of the speech signal that is contaminated with noise, thereby improving the intelligibility of the speech. Although spectral subtraction exhibited promising results at first, it exhibited constraints in addressing non-stationary and reverberant noise environments.

In the following years, progressions in Digital Signal Processing (DSP) methodologies facilitated the emergence of more intricate techniques for audio purification. During the 1980s, the noise reduction capabilities were enhanced by the emergence of adaptive filtering algorithms, such as the Wiener filter, which were designed to adapt to the statistical properties of the noise and speech signals(18). The utilisation of these techniques resulted in a more resilient noise reduction and facilitated the enhancement of ASR accuracy.

The emergence of machine learning and artificial intelligence has brought about a significant transformation in the field of audio cleaning techniques. The utilisation of neural networks for the purpose of noise reduction and speech enhancement has been investigated by researchers. During the 1990s, there was an introduction of RNN and TDNN which aimed to capture temporal dependencies and contextual information in speech signals. This development led to notable improvements in noise reduction and speech quality(19).

The field of audio cleaning has undergone a significant transformation in recent years, owing to the advent of deep learning models. The utilisation of CNN and Long Short-Term Memory (LSTM) networks has been extensively applied in the domain of noise reduction and feature enhancement. The aforementioned models demonstrate exceptional proficiency in acquiring intricate patterns and interdependencies within speech signals, thereby facilitating superior noise reduction and improved ASR efficacy(20, 21, 22).

The utilisation of GAN for the purpose of audio cleaning has garnered significant interest. GAN utilise a hybrid approach that involves both discriminative and generative networks to acquire knowledge of the relationship between noisy and clean speech signals. The utilisation of GAN has exhibited encouraging outcomes in the domain of unsupervised audio denoising. Specifically, GAN have been observed to acquire the ability to generate clear speech signals from noisy inputs(23, 24, 25).

The current research contributes to the field of audio cleaning by addressing previously identified limitations and gaps in the literature. In contrast to prior investigations that have

concentrated on particular facets, the present study adopts a holistic perspective on audio cleansing by taking into account the wider context of ASR and its influence on the efficacy of cleansing methodologies. This study endeavours to determine the optimal combinations of audio cleaning techniques by means of thorough experimentation, rigorous evaluations, and a systematic review of existing literature. This study investigates the potential synergies resulting from the use of multiple audio cleaning techniques by analysing their respective advantages and trade-offs. The ultimate goal is to contribute to the advancement of more effective and resilient audio cleaning methods.

Additionally, the present study focuses on the correlation between ASR and audio cleaning, acknowledging the pivotal function of audio quality in attaining precise speech recognition. This study provides significant contributions to the fields of audio cleaning by thoroughly examining the challenges and opportunities involved. The study's implications have far-reaching consequences for a variety of applications that depend on ASR technology, including virtual assistants, speech-to-text transcription, and voice search, as it improves the precision and practicality of these systems. To summarise, the study's comprehensive approach, exploration of combination techniques, and practical insights make a significant contribution to the field of audio cleaning. This provides valuable guidance for both researchers and practitioners and enhances the overall performance of ASR systems.

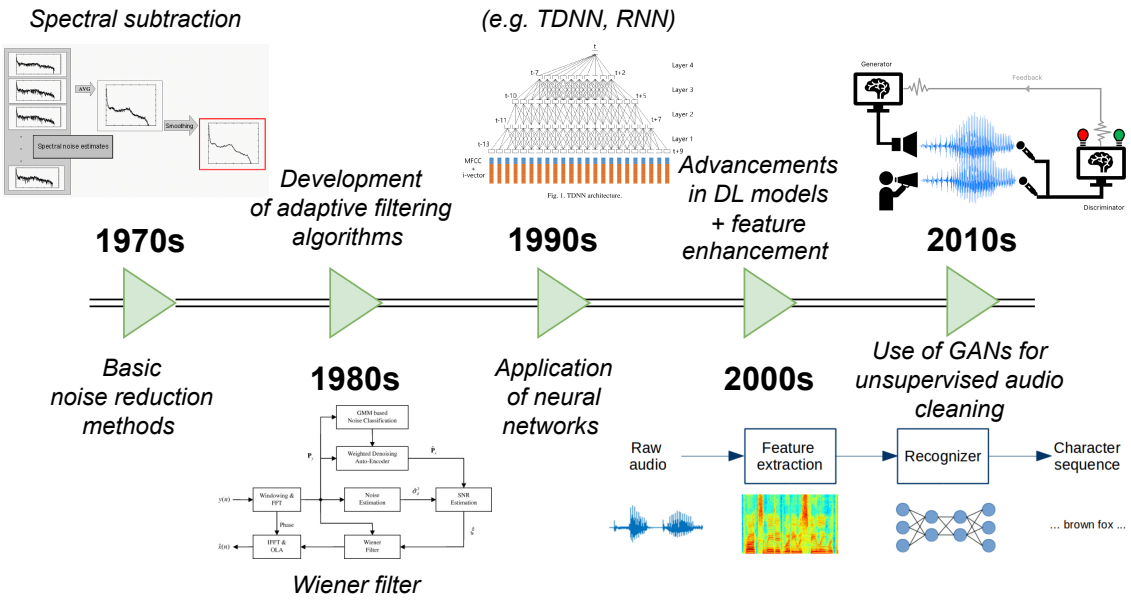


Figure 7.3: History of Audio Cleaning through the years 1970 - today

7.2.2 Primary factors affecting ASR accuracy

The efficacy of ASR systems is significantly contingent upon the caliber of the audio input. There are multiple variables that can impact the fidelity of auditory signals, ultimately resulting in diminished ASR precision. Various factors can impact speech recognition, such as environmental factors, speaker variability, speech styles, pronunciation errors, accents and dialects, and background noise. This subsection delves into each of the aforementioned factors comprehensively, utilizing relevant literature to offer perspectives on their influence on the accuracy of ASR. The objective of this study is to enhance comprehension of the difficulties that may emerge during audio data collection and to offer techniques for purifying and prepping audio data prior to its utilization in training ASR models. The most frequent issues with audio recording are listed below, along with how they may affect ASR functionality.

CHA-1: Background noise The term "background noise" pertains to any undesired auditory stimuli that co-occur with the intended speech signal, and it is a formidable obstacle that poses a significant threat to the precision of ASR systems(26). The existence of ambient noise can considerably diminish the Signal-to-Noise Ratio (SNR) of the speech signal, resulting in inaccuracies in speech recognition(27). The phenomenon of speech signal interference by noise can manifest even at low noise levels, owing to the possibility of the noise being present within the frequency range of the speech signal, as noted by Hirsch and Pearce (2011)(28).

The presence of background noise introduces extraneous acoustic energy to the primary speech signal, leading to a modification of the speech waveform. Figure 4 provides an illustration of background noise, wherein the left side of the image exhibits auditory activity while the right side remains devoid of sound. The presence of noise may result in the superimposition of the noise signal onto the speech signal, leading to the possibility of distortion or partial masking of certain segments of the speech signal.

The presence of noise has a differential impact on the amplitude and frequency components of the speech signal, thereby posing a greater challenge for the ASR system to effectively extract the requisite information from the speech signal. Furthermore, the presence of noise has the potential to induce variations in both the amplitude and duration of the speech signal, thereby resulting in alterations in the temporal and spectral attributes of the signal as reported by García-Perera et al. (2020)(26).

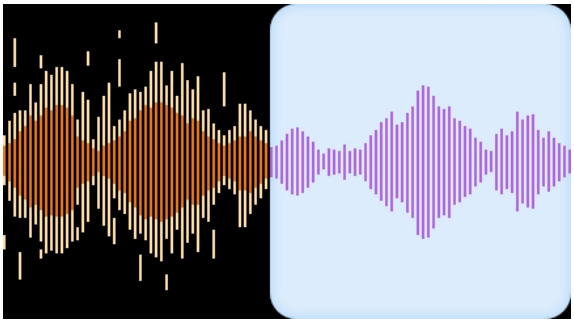


Figure 7.4: The presence of background noise in an audio file; The left side of the image is audible while the right side is silent.

CHA-2: Speaker variability The term "speaker variability" pertains to the dissimilarities in speech patterns, accent, pronunciation, and other attributes that may differ across various speakers. The existence of speaker variability can present a substantial obstacle for ASR systems, as it may lead to diminished precision and heightened rates of errors. Park and Kim (2019) conducted a study which revealed that speaker variability, particularly in the case of non-native accents, had a significant impact on ASR performance. Hori et al. (2018) conducted a study to investigate the impact of speaker variability on the accuracy of ASR in a multilingual context. The findings of the study revealed that an increase in the number of non-native speakers resulted in a decrease in the accuracy of ASR systems.

Variations in speech tempo, pitch, and intonation can also arise due to speaker variability. The presence of these discrepancies can result in waveform distortion, thereby affecting the precision of speech recognition mechanisms. An instance of a speaker with a high-pitched voice could potentially generate a waveform that exhibits a dissimilar frequency distribution in comparison to a speaker with a lower-pitched voice. Likewise, a speaker who articulates at a rapid pace may generate a waveform with a dissimilar temporal dispersion in comparison to a speaker who enunciates at a leisurely pace. The fluctuations in frequency and temporal dispersion pose a challenge for speech recognition systems to precisely detect and transcribe speech.
Figure 5 depicts an example of three audio wave files containing the same word spoken by three distinct individuals. Evidently, the three representations exhibit a similar pattern in their respective low and high points. However, there exist notable dissimilarities in certain particulars, such as the magnitude of the pause and the requisite duration for each syllable. These differences are created due to unique pronunciation characteristics that each person has.

CHA-2A: Accents and Dialects The acoustic properties of speech are significantly affected by accents and dialects, which presents a notable obstacle for ASR systems. The presence of accents and dialects can lead to significant variations in speech patterns, including alterations in the duration and stress of phonemes, thereby contributing to the occurrence of ASR errors. Research has indicated that ASR systems' recognition accuracy may decline by as much as 20% when processing non-native accents in contrast to native accents([29], [30]). Scholars have conducted studies on the influence of dialectal variation on the performance of ASR, including the distinctions between British English and American English, as reported by Nina Markl(2022)([31]). The aforementioned studies underscore the necessity of training ASR systems on a varied spectrum of accents and dialects in order to enhance their resilience.

This variation is included for the sake of the paper because the audio files contain words that were recorded in two different dialects. There are several pronunciation differences between the Greek and Cypriot Greek dialects of the Greek language. The way some sounds and letters are pronounced is one notable difference. For instance, the letter "τ" (tau) is frequently pronounced as a "tch" sound in Cypriot Greek, but a "t" sound in Standard Greek. Furthermore, vowel sounds not found in Standard Greek are frequently used in the Cypriot dialect, such as the "u" sound in words like "uranos," which means "sky."

The two dialects' intonation and stress patterns also differ from one another. The intonation of Cypriot Greek frequently has a more sing-song quality and a higher pitch at the ends of sentences. While Standard Greek typically stresses the penultimate (second-to-last) syllable, Cypriot Greek frequently emphasises the final syllable of a word ([32]).

CHA-2B: Speech Styles ASR systems encounter difficulties in recognising speech patterns that are introduced by various speech styles, such as whispering, shouting, and fast speech, due to their inherent variability. As an illustration, the act of whispering and shouting can have a notable impact on the amplitude and periodicity of speech signals, thereby rendering them more arduous to perceive with precision. Rapid speech, conversely, may cause phonetic information loss and speech sound blending, thereby resulting in errors in recognition. According to Grozdić et al. (2017), research has indicated that the precision of ASR systems may diminish by as much as 25% when attempting to recognise speech

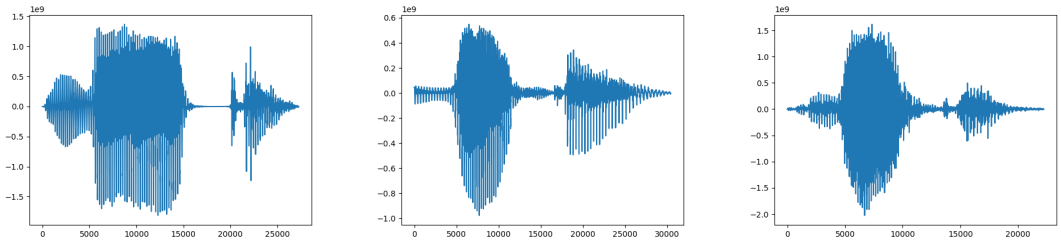


Figure 7.5: A sample of audio wave files containing three individuals saying the same word.

that is either whispered or shouted in comparison to speech that is spoken at a normal volume (33). The field of speech properties encompasses a comprehensive spectrum of vocal modes, including but not limited to whispering, soft speech, normal speech, loud speech, and shouting. The study of Zelinka et al. (2012) demonstrates the influence of variability in vocal effort on the efficacy of an isolated-word recognition system(34). Furthermore, the study evaluates various techniques to enhance the system's resilience. When trained on normal speech, the accuracy for normal speech was 80%. The results of the test indicate a significant reduction in performance, with a decrease of up to 40%, when evaluating whispered speech. This observation provides a rationale for the divergences that may arise among distinct trained models featuring diverse speech patterns.

CHA-3: Environmental factors The performance of ASR systems can also be influenced by environmental factors. The clarity of a recorded speech signal can be impacted by factors such as the quality of the microphone and the acoustics of the room. This can pose a challenge for ASR systems in accurately recognising spoken words. Furthermore, the spatial separation between the speaker and the microphone can potentially deteriorate the ratio of the signal to noise, thereby impeding the efficacy of the ASR system. The degradation of speech signal and the consequent negative impact on ASR performance can be attributed to reverberation, which is the result of sound waves reflecting off various surfaces in the environment.

An instance of the influence of environmental factors on ASR can be observed in the research carried out by Allen et al. (1979)(35). The study examined the impact of room acoustics on the efficacy of ASR systems. The study conducted by the authors revealed that the precision of ASR systems was considerably affected by the

duration of reverberation and the intensity of ambient noise within a given space. The outcomes indicated that heightened levels of reverberation and noise were associated with a decline in the performance of the ASR systems.

Matthias et al. (2009) conducted a study that investigated the impact of microphone type and placement on the accuracy of ASR(36). The study conducted by the authors revealed that the utilisation of directional microphones in close proximity to the speaker's mouth resulted in a significant enhancement in the performance of ASR systems, in comparison to the utilisation of omnidirectional microphones positioned at a greater distance from the speaker. In order to ensure optimal sound capture in an ASR system, it is imperative that the microphones be situated in a stationary position in close proximity to the sound source, which is typically the speaker's mouth. Accordingly, body-mounted microphones, including headsets and lapel microphones, offer superior sound quality.

Environmental factors can have adverse impacts on the recording, which can be observed through various manifestations in the waveform display. In instances where there exists a notable presence of reverberation, the waveform may exhibit extended decay durations, leading to a potential reduction in the lucidity and comprehensibility of the speech signal. Figure 6 depicts an illustration of this phenomenon. In the event that the recording is captured remotely from the speaker, the waveform may exhibit a decreased amplitude and an increased level of noise. This can pose a challenge in discerning the speech signal from ambient noise.

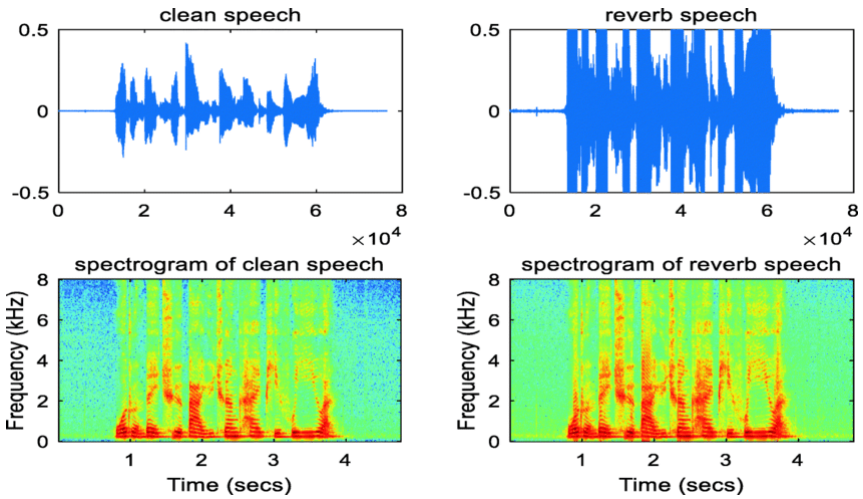


Figure 7.6: Effect of reverberation on speech waveform and spectrogram from El-Moneim's et al.(2020) paper(11)

CHA-4: Pronunciation errors The transcription of speech into text in ASR systems is contingent upon the precision of the acoustic models utilised. However, when a speaker pronounces a word differently from the way it is pronounced in the training data, ASR accuracy can be significantly affected. This is known as a pronunciation error. There are multiple variables that can potentially influence the occurrence of pronunciation inaccuracies, such as the speaker's mother tongue, accent, and manner of speaking. Studies have shown that non-native speakers of a language are more likely to make pronunciation errors, especially when they are less proficient in the language (Gibbon et al.)(37). Numerous research studies have explored the influence of pronunciation inaccuracies on the precision of ASR systems. For example, one study by Lee and Hon (2019) found that mispronunciation of specific phonemes resulted in a significant decrease in ASR accuracy, particularly for non-native speakers(38).
Pronunciation errors can affect the waveform in several ways. Initially, the waveform may exhibit a deficiency in distinctness or accuracy in the articulation of specific phonemes or lexemes, resulting in a more erratic configuration. This can result in the waveform being harder to distinguish, and thus harder for ASR systems to accurately interpret. Moreover, the presence of mispronounced words or sounds can result in the manifestation of entirely distinct words or sounds in the waveform, thereby causing perplexity in ASR systems. The aforementioned phenomenon may be interpreted as a modification in either the frequency or amplitude of specific segments of the waveform, which can result in imprecisions in the process of speech recognition. The occurrence of pronunciation inaccuracies can potentially generate additional interference in the waveform, thereby impeding the ASR system's ability to differentiate between the intended speech and the erroneous phonetic or lexical units. This can result in a more complex waveform with more variability in amplitude and frequency, further complicating the speech recognition process.

7.3 Research Questions for Audio cleaning

Despite being widely used, ASR technology still has a number of issues that could decrease its accuracy and dependability. The wide range of speech patterns and acoustic conditions is one of the biggest obstacles. The accuracy of ASR systems may be impacted by this variation, resulting in mistakes and misinterpretations. In this paper, we explore two crucial ASR-related Research Questions (RQs) in order to address these issues:

RQ1: *What are the primary factors that significantly affect the accuracy of automatic speech recognition, and what strategies can be employed to mitigate their impact?*

The significance of addressing this research inquiry lies in the fact that the precision of ASR systems is often influenced by diverse factors, including but not limited to background noise, speaker variability, accents, and other related factors. Through the identification of key factors that exert a substantial influence on the precision of ASR and the formulation of corresponding mitigation strategies, it is **possible to enhance the dependability and efficacy** of such systems.

In order to address the research inquiry, a systematic review of the extant literature will be undertaken to ascertain the principal factors that exert a significant impact on the precision of ASR. Subsequently, experiments will be carried out to evaluate the effects of aforementioned factors on the precision of ASR systems. The proposed experiments will entail the utilization of diverse datasets comprising of recordings featuring distinct accents, speaking styles, ambient noise, and other pertinent variables. Subsequently, the obtained outcomes will be scrutinized to discern the pivotal variables that exert a substantial influence on the precision of ASR and ascertain the optimal tactics to alleviate their repercussions.

Several strategies that can be investigated to mitigate the influence of the primary factors that impact ASR precision encompass methods for reducing noise, adapting to the speaker, expanding the vocabulary, and normalizing accents. The techniques under consideration will be applied to a consistent corpus of audio data in order to assess their efficacy in enhancing accuracy and to determine the optimal approaches for mitigating the influence of the primary factors that impact automated speech recognition.

RQ2: *What are the advantages and trade-offs of employing various combinations of audio cleaning techniques in the preparation of audio files for machine learning models used in automatic speech recognition? Which cleaning techniques are most effective in improving the final results of the automatic speech recognition system?*

The significance of this inquiry stems from the fact that the precision of ASR systems is contingent upon the caliber of the audio data utilized for the purpose of training the machine learning algorithms. Audio recordings are frequently subject to various types of interference, such as background noise, reverberation, and other forms of distortion, which may have an impact on the precision of ASR) systems. The implementation of diverse

amalgamations of audio cleansing methodologies can **enhance the caliber of audio data utilized for the training** of ASR models, thereby **augmenting their precision**.

In order to address this research inquiry, a comprehensive examination of the current body of literature pertaining to audio cleansing methodologies employed in the preprocessing of audio data for machine learning algorithms in the domain of ASR will be conducted. Subsequently, we shall assess the benefits and compromises of diverse amalgamations of audio refinement methodologies through a sequence of trials utilizing distinct sets of data. The study will additionally examine the effects of various audio cleaning methodologies on the precision of ASR systems. The objective of our study is to ascertain the optimal amalgamations of audio cleansing methodologies that can enhance the caliber of audio documents utilized for the instruction of ASR models, thereby elevating their precision.

7.4 Audio Enhancement Techniques for ASR Systems

Subsection 7.2.2 elucidates that there exist several factors, namely **CHA-1** to **CHA-4**, which can exert a significant impact on the efficacy of ASR systems. In order to address these challenges, scholars have devised diverse methodologies to alleviate the influence of these factors on ASR efficacy. The present chapter aims to examine various techniques and evaluate their efficacy in enhancing ASR accuracy in challenging acoustic conditions. The two broad categories of these methods are feature-based and model-based approaches. Feature-based methodologies encompass the extraction of more resilient features from the speech signal, whereas model-based methodologies involve the training of more advanced models that can more effectively manage the variability in the speech signal. The subsequent section will delve into the intricacies of these methodologies and analyse their respective merits and drawbacks.

7.4.1 Exploring Essential Libraries

This section will delve into the code implementation, with a particular emphasis on the libraries employed. Libraries are essential in addressing diverse tasks, including but not limited to data analysis, audio processing, visualisation, and other related functions. Libraries offer a range of functionalities and tools that streamline the development process and optimise the overall efficiency of the code. This paper will provide a comprehensive analysis of the libraries in question, encompassing their intended function, characteristics, and their role in fulfilling the distinct demands of the project.

1. **soundfile (Source: PySoundFile, Developer: Bastian Bechtold)**: This library provides an interface to read and write audio files. In the code, soundfile is imported as sf, and it is used to read the audio files for further processing.

2. **re (Regular Expressions, Source: Python Standard Library, Developer: Python Software Foundation)**: The re module allows you to work with regular expressions in Python. In the code, it is used to extract ID numbers from the file paths and file names.

3. **pandas (Source: pandas, Developer: pandas Development Team)**: Pandas is a powerful library for data manipulation and analysis. In the code, it is used to create and manipulate DataFrames. It is utilized it to perform operations such as grouping, counting, and visualizing data.

4. **seaborn (Source: seaborn, Developer: Michael Waskom)**: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. In the code, seaborn is used to create count plots and bar charts to visualize the frequency and distribution of categories, words, and gender in the audio recordings.

5. **matplotlib (Source: Matplotlib, Developer: Matplotlib Development Team)**: Matplotlib is a widely used plotting library in Python. It provides a flexible and comprehensive set of functions for creating static, animated, and interactive visualizations. In the code, matplotlib is used to create various plots, such as bar charts, count plots, and pie charts, to visualize the distribution of categories, words, and gender in the audio recordings.

6. **firebase_admin (Source: Firebase Admin SDK, Developer: Firebase team)**: Firebase Admin SDK allows you to interact with Firebase services from a server environment. In the code, firebase_admin is used to initialize the Firebase Admin SDK and interact with the Firebase storage service. It enables the downloading of files from a specified bucket and folder path.

7. **google.cloud.storage (Source: google-cloud-storage, Developer: Google)**: The google.cloud.storage library provides a client interface for interacting with Google Cloud Storage. In the code, it is used to work with the storage bucket and list the files in a specified folder path.

8. **os (Source: Python Standard Library, Developer: Python Software Foundation)**: The os module provides a way to interact with the operating system. In the code, it is used to create directories for storing downloaded and modified files.

9. **pickle (Source: Python Standard Library,Developer: Python Software Foundation)**: The pickle module allows you to serialize Python objects to a byte stream and deserialize them back into objects. In the code, it is used to store and load the metadata list as a pickle file.

10. **pydub (Source: pydub, Developer: James Robert)**: PyDub is a simple and easy-to-use library for audio processing in Python. In the code, pydub is used to load, manipulate, and export audio files. It provides functions for applying noise reduction, compression, silence detection, and normalization to the audio data.

11. **numpy (Source: NumPy, Developer: NumPy developers)**: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. In the code, numpy is used for various operations, such as converting audio data to NumPy arrays, computing score values, and performing mathematical computations.

12. **wave (Source: Python Standard Library, Developer: Python Software Foundation)**: The wave module provides a convenient interface to the Waveform Audio File Format (WAV) file format. In the code, it is used to open and read the audio files in WAV format for further processing.

7.4.2 Mitigating the factor CHA-1

The presence of ambient noise poses a significant obstacle for ASR systems, as it can considerably diminish the precision of speech recognition. There are various techniques that can be employed to mitigate ambient noise in audio recordings intended for ASR purposes. Utilising noise reduction algorithms is a viable approach for mitigating background noise in audio files intended for ASR purposes. The algorithms function by conducting an analysis of the audio file and discerning the ambient noise, subsequently eliminating or decreasing it while maintaining the integrity of the speech signal. Numerous noise reduction algorithms exist, such as spectral subtraction, Wiener filtering, and adaptive filtering. Utilising a DSP technique that entails the application of a low-pass filter and a high-pass filter to individual segments of audio is a highly efficacious approach for eliminating background noise from

speech signals. This methodology can prove to be especially advantageous in instances where speech signals are marred by broadband noise or persistent noise throughout the recording. Hence, it is deemed as the optimal choice given that one of the aims of this project is to enhance the quality of speech signals for the purpose of advancing ASR.

The purpose of the low-pass filter is to eliminate high-frequency noise that could potentially exist in the signal, such as electrical noise or hiss. The high-pass filter is designed to remove low-frequency noise that may be present in the signal, such as hum or rumble. The utilisation of dual filters enables the retention of a spectrum of frequencies located in the central region, which is known to contain significant speech-related data, while simultaneously eliminating noise present at the extremities of the spectrum.

7.4.2.1 Implementation mitigating techniques for CHA-1

Figure 7.7 presents the noise reduction Python code about a function called "noise_reduction" that performs noise reduction on an audio file and saves the filtered audio as a new WAV file. The code first extracts the audio data from the input audio file as a numpy array, while also retrieving information about the file, such as sample rate, number of channels, and sample width.

Subsequently, the audio data is reshaped into a two-dimensional array, where each row represents a channel. Using the "split_on_silence" function from the "pydub" library, the code splits the audio file into chunks based on detected periods of silence. By splitting the recording into smaller chunks and applying the filters to each chunk separately, the noise reduction can be more targeted and effective.

Next, for each chunk of audio, the code applies a low-pass filter to remove high-frequency noise above 4000 Hz and a high-pass filter to remove low-frequency noise below 200 Hz, using the "low_pass_filter" and "high_pass_filter" functions from the "pydub" library. The filtered chunks are then concatenated back into a single audio file.

Finally, the filtered audio is saved as a new WAV file in a directory named "modified_files", with the name of the new file based on the input file name, with "_noise_red" appended to the end.

Figure 7.8 contains the code for a function that applies an oversmoothing filter to an audio signal using a moving average filter. This function applies oversmoothing using a moving average filter to an input audio signal specified as an ndarray in the audio_signal argument. The window_size parameter determines the degree of oversmoothing by specifying the size of the moving average window. window_size is set to 5 by default and is defined as a numpy array of ones divided by the window size. This results in a uniform

```python
def noise_reduction(audio,name):

    # Extract the audio data as a numpy array
    audio_data = np.array(audio.get_array_of_samples())
    sample_rate = audio.frame_rate
    num_channels = audio.channels
    sample_width = audio.sample_width
    audio_data = np.array(audio.get_array_of_samples())

    # Convert the audio data to a numpy array
    audio_data = np.frombuffer(audio_data, dtype=np.int16)

    # Reshape the audio data into a two-dimensional array (one row per channel)
    audio_data = audio_data.reshape(-1, num_channels)

    # Split the audio file into segments using silence detection
    chunks = split_on_silence(audio, min_silence_len=10, silence_thresh=-30)

    # Apply noise reduction to each chunk using the built-in filter function
    for i, chunk in enumerate(chunks):
        filtered_chunk = chunk.low_pass_filter(4000)
        filtered_chunk = filtered_chunk.high_pass_filter(200)
        chunks[i] = filtered_chunk

    # Concatenate the filtered chunks back into a single audio file
    filtered_audio = None
    if chunks:
        filtered_audio = chunks[0]
        for chunk in chunks[1:]:
            filtered_audio = filtered_audio + chunk

    if filtered_audio is not None:
        # Export the filtered audio to a new WAV file
        filtered_audio.export("./modified_files/" + name + "_noise_red.wav", format
    ="wav")
    else:
        audio.export("./modified_files/" + name + "_noice_red.wav", format="wav")
```

Figure 7.7: Python code: Noise reduction function.

weighting of samples within the window. The scipy.signal library's lfilter function is used to apply the moving average filter to the input audio signal. As filter coefficients, the window array is utilised, and a normalisation factor of 1 is specified for the denominator. The output is stored in the variable oversmoothed_signal. The final step is to export the oversmoothed audio signal as a WAV file using the export() method. The file is saved in the./modified_files/ directory with the name "_overmoothing.wav" appended to the original name argument.

7.4.2.2 Fine-Tuning Low-Pass and High-Pass Filters

The present code employs a low-pass filter for the purpose of eliminating high-frequency noise that surpasses the threshold of 4000 Hz. This approach has been found to be effica-

```python
def oversmoothing(audio,name, window_size=5):
    audio = np.asarray(audio)
    audio = audio.squeeze()  # Remove single-dimensional entries from the shape of the array

    window = np.ones(window_size) / window_size
    oversmoothed_signal = lfilter(window, 1, audio)
    oversmoothed_signal.export("./modified_files/" + name + "_overmoothing.wav", format="wav")
```

Figure 7.8: Python code: Oversmoothing function

cious in mitigating auditory disturbances emanating from sources such as electronic noise or ambient conversations. The high-pass filter is utilised for the purpose of eliminating low-frequency noise that falls below the threshold of 200 Hz. This technique has been found to be efficacious in mitigating noise emanating from various sources, including but not limited to wind, traffic, and air conditioning.The selection of cutoff frequencies for low-pass and high-pass filters is aimed at attenuating noise to a maximum extent while retaining the intended audio signal. The selection of the cut-off frequencies for low-pass and high-pass filters, namely 4000 Hz and 200 Hz, respectively, is determined through an examination of the frequency spectrum of the audio file. This process involves identifying the frequency range in which the noise exhibits the highest prominence. The frequency spectrum of the audio file was visualised through the utilisation of MATLAB[1] software. Achieving a balance between minimising noise and maintaining the integrity of the audio signal is a crucial consideration. Consequently, a prevalent approach employed to determine the optimal equilibrium for audio files involves experimenting with numerous pairs. The filters were subjected to a variety of spectrums, and the most favourable points were selected. The low-pass range comprises the values [2000, 3000, 3500, 4000, 4500, 5000, 6000], whereas the high-pass range encompasses the values [50, 100, 150, 200, 250, 300, 350, 400, 500].

7.4.2.3 Observations after mitigating the factor CHA-1

Prior to the implementation of noise reduction techniques, the waveform representation of the audio file displayed numerous anomalies that posed challenges in identifying the speech signal. The graph exhibited noteworthy fluctuations in magnitude and a considerable amount of ambient interference that obscured the vocal signal. The discernment of speech segments from ambient sound was found to be arduous due to their apparent submergence

[1] https://www.mathworks.com/products/matlab.html

within the noise. Furthermore, the noise exhibited a broad spectrum, encompassing both high and low frequencies, thereby exacerbating the discernment of the speech signal. The red lines in the visual representation indicate the uppermost and lowermost points of the ambient noise, particularly at the onset and conclusion of the recording. Meanwhile, the green region denotes the segment of the recording that exclusively comprises background noise beyond the absence of sound.

Upon the implementation of noise reduction techniques, the waveform representation of the audio file exhibited substantial enhancements in both clarity and signal-to-noise ratio. The signal's amplitude exhibited a greater degree of uniformity, with reduced fluctuations and an overall smoother profile. The ambient noise was considerably diminished, thereby enhancing the prominence of the speech signal. The speech segments have been observed to manifest as identifiable and distinct patterns on the graph, exhibiting clearly defined peaks and valleys that correspond to the vocalisations of the speaker. Additionally, the audio signal's frequency spectrum exhibited a greater concentration around the speech frequencies, as evidenced by the absence of certain portions at the beginning and end of the signal, with fewer noise components present in the high and low frequency ranges.

In general, the process of noise reduction had a notable effect on the waveform representation of the audio recording, rendering the speech signal more readily distinguishable and amenable to analysis. The findings indicate that the noise reduction methods employed were successful in eliminating a substantial portion of the ambient noise while retaining the fundamental characteristics of the speech signal, as evidenced by the enhanced clarity and signal-to-noise ratio. The wave graph that ensues from the aforementioned process serves as a valuable visual aid in evaluating the audio file's quality and gauging the efficacy of the noise reduction technique.

7.4.3 Mitigating the factor CHA-2

Speaker variability is a prevalent challenge encountered when working with ASR systems. There are several methodologies that can aid in addressing this issue. **Signal normalisation** is a technique that can be employed to mitigate speaker variability. The term "signal normalisation" pertains to the procedure of altering an audio signal in a manner that results in a more uniform amplitude and frequency distribution. The utilisation of this technique can prove to be advantageous for ASR systems, as it has the potential to mitigate the influence of speaker variability on the precision of the system. The normalisation of the signal enhances the system's ability to identify the phonemes and words uttered, irrespective of the speaker's accent, dialect, or speech style.

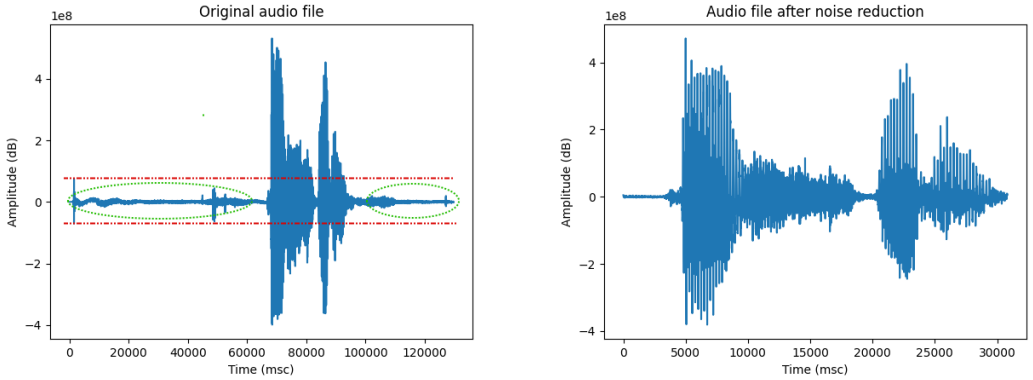


Figure 7.9: Changes in the audio wave spectrum after noise reduction has been applied [red: maximum point of noise outliers; green: area containing noise outliers].

There are various methodologies for signal normalisation, each possessing unique merits and demerits. Cepstral Mean Normalisation (CMN) is a frequently employed method in signal processing [1]. It entails the subtraction of the average value of the cepstral coefficients of a signal from every frame of the signal. This technique has the potential to mitigate the influence of fluctuations in the speaker's vocal tract length and other variables that may impact the frequency distribution of the signal. Feature Space Maximum Likelihood Linear Regression (fMLLR) is a technique that entails the training of a regression model to map the acoustic features of a signal to a standard reference signal [2]. This technique can aid in accounting for variations in the speaker's articulation and other variables that may impact the phonetic characteristics of the signal.

The technique of CMN is a commonly employed and straightforward method that entails the deduction of the mean of the feature values over time for every frequency band. The implementation of this methodology has the potential to mitigate channel inconsistency and enhance resilience across diverse recording scenarios. The computational efficiency of CMN renders it a versatile tool that can be seamlessly integrated into any speech recognition system without necessitating supplementary training.

In contrast, fMLLR is a sophisticated approach that entails the training of a transformation matrix tailored to the speaker, which serves to map the feature space onto a more discerning space. The application of fMLLR has the potential to facilitate the adaptation of an extant acoustic model to a particular speaker or environment, thereby leading to a

[1] More information about CMN https://speechpy.readthedocs.io/en/latest/content/postprocessing.html

[2] More information about FMLLR: https://dbpedia.org/page/FMLLR

discernible enhancement in recognition precision. Nevertheless, the utilisation of fMLLR necessitates training data that is specific to the speaker and may incur significant computational costs.

The selection of a suitable normalisation technique is contingent upon the particular demands of the application and the resources at hand. The present project has opted for the utilisation of the CMN technique, which is considered a suitable initial approach for speech recognition systems due to its ease of implementation and potential to enhance recognition accuracy across various scenarios.

Utilising a **varied range of speech data** presents a potential benefit in enhancing the overall precision of the system. This phenomenon can be attributed to the fact that the system's proficiency in speech recognition improves with an increase in the amount of speech data it is trained on. Through the exposure of the system to a diverse array of vocal characteristics and speech modalities, it enhances its capacity to acclimatise to novel voices and speech patterns that it may encounter within the actual context.

Moreover, incorporating a varied range of speech data can enhance the end-user's experience of the ASR system. The reason for this is that users exhibit a higher likelihood of contentment with a system that demonstrates precise speech recognition capabilities, irrespective of their accent or dialect. Through the process of training the system on a varied and inclusive corpus of speech data, the system's ability to accurately identify and comprehend a broader spectrum of speech patterns and styles can be enhanced. This, in turn, can lead to an improved user experience and increased accessibility for a more diverse user base. The reason for this is that the characteristics of speakers can exhibit substantial variation from one individual to another. Consequently, if an ASR system is trained on a restricted corpus of speech data, its performance may be suboptimal when processing speech from different speakers. The incorporation of varied speech data into the training set can enable ASR developers to ensure that the system is capable of recognising a broad spectrum of speech variations and accommodating the diverse speaking styles of users. Furthermore, the incorporation of a varied training set can enhance the overall generalizability and resilience of the ASR system, thereby increasing its efficacy in practical scenarios and expanding its user base. To succinctly encapsulate, the inclusion of a varied training set is of paramount importance in the development of an ASR system that can effectively discern speech from a diverse pool of speakers and adjust to the intricacies inherent in human language.

7.4.3.1 Implementation mitigating techniques for CHA-2

Figure 7.10 presents the CMN algorithm in Python code that performs normalization on an audio file and saves the filtered audio as a new WAV file. The objective of this code is to mitigate the impact of speaker variability by implementing CMN on an audio signal. The initial step of the process involves the conversion of the input audio file into a numpy array, which is achieved through the utilisation of the get_array_of_samples() function from the pydub library. Subsequently, the frame rate of the initial audio signal is obtained by utilising the frame_rate attribute.

Subsequently, the mean of the complete signal can be calculated by utilising the np.mean() function from the numpy library. Subtraction of the mean value from each frame of the audio signal is performed by subtracting the mean of the numpy array from the original signal utilising the - operator.

The CMN algorithm is capable of enhancing the performance of ASR systems by eliminating channel-specific fluctuations in the signal through the subtraction of the signal's mean. The resultant signal is a standardised rendition of the initial signal, which is anticipated to be less impacted by variations in the speaker's characteristics.

Ultimately, the altered signal is reconstituted as a pydub AudioSegment by means of the tobytes() function from the numpy array, and subsequently persisted as a novel audio file via the export() method.

The utilisation of the pydub library in Python for implementing CMN presents a straightforward and effective approach for the normalisation of an audio signal. The utilisation of numpy arrays and the pydub AudioSegment class is substantiated by their efficacy as data structures for the manipulation of audio data in the Python programming language. The utilisation of the tobytes() method for the purpose of converting the altered numpy array into a compatible format for the pydub library is a straightforward and efficient approach.

Train ASR systems on diverse speech data

In addition through extensive and diverse dataset training, ASR systems can effectively accommodate the multifarious manners in which individuals communicate, encompassing variations in pronunciation, regional language, and speaking patterns. The utilised recording set for this project encompasses a heterogeneous group of speakers, comprising individuals with varying gender identities, age ranges, and socio-cultural backgrounds. The recording set comprised of both male and female speakers, spanning across various age groups from young children to older adults. Furthermore, the recording set comprised

```python
def normalization(audio,name):
    # Convert the audio to a numpy array
    signal = np.array(audio.get_array_of_samples())
    rate = audio.frame_rate

    # Apply CMN across the entire audio signal
    signal_cmn = signal - np.mean(signal)

    # Convert the modified signal back to a pydub audio segment
    modified_audio = AudioSegment(signal_cmn.tobytes(), frame_rate=rate,
    sample_width=2, channels=1)

    modified_audio.export("./modified_files/" + name + "_normalizedCMN.wav", format
    ="wav")
```

Figure 7.10: Python code: Normalization function

participants from Greece and Cyprus, thereby ensuring a diverse range of dialects and speech patterns.

7.4.3.2 Observations after mitigating the factor CHA-2

The CMN technique is employed to normalise a signal by adjusting the amplitude values in order to mitigate the variability across the signal as it is displayed in Figure 9.

Prior to CMN normalisation, the decibel level exhibit an increase owing to the broader spectrum of amplitude values. Following the process of CMN normalisation, decibel level to decrease as a result of the diminished amplitude value range. Upon the application of CMN (i.e., cepstral mean normalisation) to an audio signal, it can be observed that the mean component of the signal is effectively eliminated. Consequently, all amplitude values exceeding the mean will be transformed into negative values. The outcome of this phenomenon lead to a modification in the total magnitude of the signal, whereby certain samples exhibit reduced values while others exhibit amplified values. The decrease in decibel level does not necessarily imply a degradation of information or signal quality. Instead, it signifies a decrease in variability that can enhance the accuracy of ASR.

It is noteworthy that the preservation of the relative amplitudes of distinct signal components is crucial during the normalisation process. It is imperative that the form and attributes of the signal are maintained, notwithstanding any alterations in the overall magnitude. The observed outcome can be attributed to the uniform application of modifications by CMN across all signal constituents, irrespective of their initial magnitude.

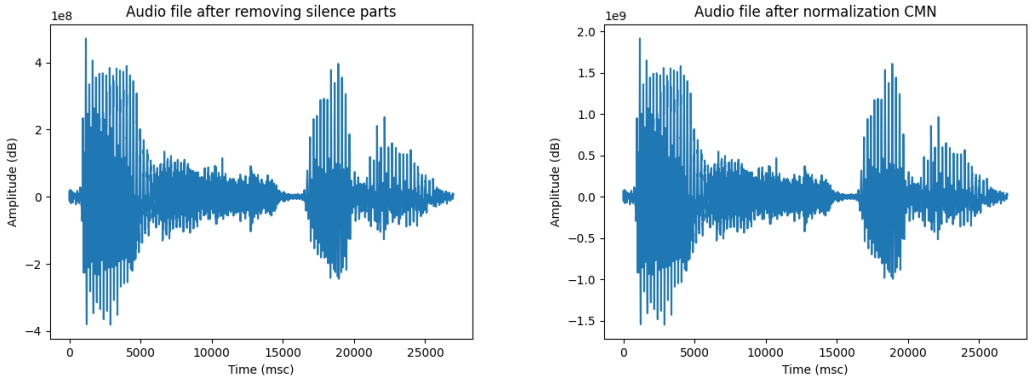


Figure 7.11: Changes in the audio wave spectrum after limiting the effect of speaker variability.

7.4.4 Mitigating the factor CHA-3

The precision of ASR systems can be impeded by environmental factors, particularly in a crowdsourcing context where users have the liberty to record audio files at their discretion, without stringent regulation over the recording devices or the environment. The quality of speech signals can be negatively impacted by various acoustic distortions such as background noise and reverberation, which can pose challenges for ASR systems in accurately transcribing them.

As a result, the management of environmental variables presents a considerable obstacle. The following justifications examine the inescapable influence of environmental variables on audio data for our ASR system, given the wide array of recording apparatuses and user surroundings.

The impact of uncontrollable recording environments on audio quality cannot be ignored, as factors such as background noise, reverberation, and ambient sounds inevitably affect the final output. Despite the utilisation of noise reduction techniques and sophisticated algorithms, the complete eradication of these factors from audio files remains a challenging task.

1. **Unmanageable Recording Environments:** The utilisation of diverse recording devices by users in our specific crowdsourcing system may result in variations in microphone characteristics and sensitivity levels. The audio quality and susceptibility to environmental factors may vary due to the distinct audio capturing mechanisms employed by these devices. It is imperative for the ASR system to exhibit robustness in order to effectively manage such inconsistencies.

2. **Variation in Recording Equipment:** The practise of enabling users to record audio at their discretion and in any location presents a convenient option, albeit with the drawback of restricted control over the surrounding environment during the recording process. Various factors, such as ambient conversations, vehicular noise, and personal speaking habits such as low volume or distance from the recording device, can considerably influence the quality of audio and, consequently, the accuracy of ASR systems.

3. **User Conduct and Recording Practises:** Acknowledging and addressing the challenges presented by uncontrollable environmental factors in audio files is imperative in our targeted crowdsourcing ASR system. Furthermore, our targeted crowdsourcing system provides users with the option to mitigate and tackle environmental obstacles. Additionally, users are afforded the liberty to review their audio recordings, remove them if deemed necessary, and subsequently re-record them. This functionality enables users to preempt unforeseen occurrences of substandard recordings, affording them an improved prospect of capturing audio of superior quality.

Despite the aforementioned challenges, **the system integrates diverse methodologies** to enhance the influence of ecological elements on the captured audio, as expounded in the preceding subsections 7.4.2 - 7.4.4. The aforementioned methodologies encompass normalisation, elimination of background noise, and additional signal processing techniques, as explicated in the preceding subsection. Although these techniques make a significant contribution towards enhancing the audio quality and reducing the impact of environmental factors, it is crucial to acknowledge that they are not a panacea for this problem.

In summary, despite efforts to mitigate the influence of environmental variables, their complete eradication remains a formidable undertaking. Consequently, we are actively pursuing innovations in ASR technology to effectively tackle these obstacles.

7.4.5 Mitigating the factor CHA-4

The precision of ASR systems, which aim to transcribe spoken language into written text, can be notably affected by inaccuracies in pronunciation. Fortunately, several strategies can be implemented to restrict pronunciation errors in audio files for ASR.

1. **Utilise superior audio recordings**: Utilising high-fidelity audio recordings is imperative for ensuring the precision of a ASR technology. The presence of extraneous noise and distortions in audio recordings of inferior quality can pose a challenge for

ASR systems, thereby impeding their ability to effectively identify and transcribe speech. Consequently, it is imperative to utilise audio recordings of superior quality that are devoid of any background noise, distortion, or other forms of interference.

2. **Train ASR systems with a variety of speech samples**: It is customary to train ASR systems on extensive speech datasets to enhance their precision. However, it is advisable to incorporate diverse speech data during the training process. In order to mitigate potential inaccuracies in pronunciation, it is crucial to incorporate a varied assortment of speech data within the dataset utilised for training purposes. The aforementioned may encompass individuals hailing from diverse geographical areas, exhibiting distinct dialects and languages, as well as those with differing speech patterns and impediments.

3. **Use language models**: The utilisation of language models has the potential to enhance the precision of ASR systems through the provision of supplementary contextual information. Language models are trained using extensive datasets of written language and can be utilised to estimate the likelihood of a specific word or phrase occurring within a given context. The integration of language models into ASR systems has the potential to mitigate pronunciation errors by furnishing supplementary context for speech recognition.

To summarise, the mitigation of pronunciation errors in audio files for ASR can be accomplished by utilising high-fidelity audio recordings, varied speech data for training, and the implementation of language models. Through the implementation of these aforementioned techniques, it is plausible to enhance the precision of ASR systems and mitigate enunciation discrepancies.

7.4.5.1 Implementation mitigating techniques for CHA-4

Prior to exploring the execution of language models, the incorporation of varied speech data, and the choice of audio file formats, it is crucial to underscore the importance of these elements in the creation of resilient and precise ASR systems. Chapter 7.4.4 outlines three potential methods for addressing environmental factors, which will be elaborated upon below. The project's implementation of these methods will also be discussed.

Selection of audio file format

The selection of the file format can have an impact on the overall quality of the recordings. The Third Generation for mobile Platform (3gp) format may be a viable option for recording audio files within a mobile application, as it presents various advantages.

1. **Compression and file size:** The 3gp file format is frequently utilised for mobile devices due to its compressed nature, which allows for reduced file sizes. The design of this technology is aimed at minimising the size of files without compromising the audio quality to a significant extent. The utilisation of this approach can confer benefits to mobile applications, as it enables the attainment of reduced file sizes, thereby mitigating the storage space demands on the device. Furthermore, it aids in the optimisation of bandwidth consumption during the transmission of audio files across networks, a critical aspect for individuals with restricted data plans.

2. **Device compatibility:** The 3gp format enjoys extensive compatibility with mobile devices, rendering it a dependable option for the development of mobile applications. The utilisation of the 3gp format for audio recording guarantees seamless playback across various mobile devices, irrespective of their hardware specifications or operating systems.

Conversely, in the context of audio recordings for websites, opting for the WAV format may be deemed a viable choice for the ensuing rationales:

1. **Uncompressed audio quality:** The audio file format known as WAV is characterised by its uncompressed nature, which allows it to preserve the entirety of the recorded audio's quality and fidelity. In contrast to compressed file formats such as 3gp, WAV files do not undergo any lossy compression algorithms, thereby yielding a superior level of precision in audio. This holds significant importance for websites that prioritise audio quality, particularly those that feature multimedia content or professional recordings.

2. **Broad compatibility:** WAV files enjoy extensive compatibility as they are widely supported by a majority of web browsers and media players, thus rendering them a dependable option for audio content on websites. The utilisation of the WAV format guarantees seamless playback of audio files across various devices and platforms, devoid of any compatibility discrepancies.

To summarise, the rationale behind opting for 3gp audio files for the mobile application and WAV files for the website is based on a careful analysis of the distinct demands and limitations of each platform. The 3gp format is optimised for mobile devices that have limited storage and bandwidth, providing a compressed format. On the other hand, WAV format offers a high-quality and uncompressed format that is compatible with a broad spectrum of web browsers. Optimising audio quality and compatibility for each platform can be achieved by carefully selecting the appropriate file formats.

Train ASR systems with a variety of speech samples

Chapter 7.4.3.1 has elaborated on the significance of integrating heterogeneous speech data into ASR systems. In order to guarantee the resilience and versatility of our models, we are currently employing a diverse array of speech data derived from multiple sources and speakers. As a result, this feature aids in mitigating environmental factors.

Employing language models

Incorporating language models into ASR systems offers significant gains in transcription accuracy, contextual understanding, and overall performance. By integrating language models, ASR systems benefit from improved handling of pronunciation errors, enhanced contextual understanding, and better correction of out-of-vocabulary (OOV) words. Language models leverage statistical patterns and contextual information to predict the most probable words or phrases based on the input speech, resulting in more accurate transcriptions. These models also enable ASR systems to capture nuances, idiomatic expressions, and domain-specific terminology more effectively, enhancing language understanding and producing contextually appropriate and linguistically coherent transcriptions.

Furthermore, it is worth mentioning that language models will be employed in a further step of the process when training models are built.

Speech enhancement

Speech enhancement is a technique that can substantially enhance the performance of ASR systems in the project. This technique converts spoken language into written text, and it heavily relies on the quality and intelligibility of the input audio. Speech enhancement techniques have a specific objective of improving the quality and intelligibility of speech signals through the reduction of noise and enhancement of the signal-to-noise ratio. The aforementioned techniques utilize diverse algorithms to attenuate or eliminate undesirable noise while retaining crucial speech data. Speech enhancement has the potential to offer various advantages to ASR systems by mitigating the effects of noise on the audio signal.

1. **Enhanced Accuracy:** Speech enhancement techniques can aid in achieving higher accuracy in recognizing and transcribing spoken words by reducing noise levels and improving the clarity of the speech signal. This is particularly relevant for ASR systems. Improved audio signals enable ASR models to concentrate on speech characteristics and minimize the probability of misapprehension due to noise disruption.

2. **Enhanced Stability:** The implementation of speech enhancement techniques enhances the robustness of ASR systems, particularly in real-world settings that present challenging acoustic conditions. The implementation of noise suppression or elimination techniques facilitates the optimal performance of ASR systems in environments characterized by high levels of ambient noise, such as crowded public spaces, street recordings, or telecommunication applications.

3. **Better Generalization:** Enhanced speech data has been observed to result in improved generalization of ASR models when tested on data that is either noisy or unseen. Through the utilization of speech signals that have undergone noise reduction enhancement during ASR model training, the models acquire greater resistance to noise fluctuations, thereby enhancing their performance when processing unprocessed, real-world audio.

Implementation technique for speech enhancement

Figure 7.12 contains Python code that implements a speech enhancement technique based on spectral subtraction. Utilizes spectral subtraction to perform speech enhancement on an audio file. The augmented audio is saved with the filename name + "_enhancement.wav" Sample_rate and audio data are extracted from the imported audio file at the outset. The code verifies that the audio has multiple channels (audio.ndim $>$ 1) to assure compatibility with stereo audio. If so, the audio is converted to mono using np.mean to calculate the average of all channels. In order to convert audio data from the time domain to the frequency domain, Short-Time Fourier Transform (STFT) is employed. STFT parameters, including frame size and frame displacement, are specified in seconds. The audio is divided into frames that overlap, and the Fourier transform is calculated for each frame using the fft function from the scipy.fftpack module. The noise spectrum is estimated using the initial 50 frames of the STFT (stft[:50]). This noise estimate is computed by using np.mean to take the mean of the magnitudes of the selected frames. The magnitude of the STFT is subtracted from alpha times the estimated noise spectrum to execute spectral subtraction. The result is then reduced to zero and combined with phase data to reconstruct the

improved STFT. Using the Inverse Short-Time Fourier Transform (ISTFT), the STFT is transformed back into the time domain. The enhanced audio is then reconstructed by merging the frames together, taking overlap and padding into account. Using slicing, the reconstructed audio is shortened to the original length of the input audio. The enhanced audio is exported as a WAV file using the export function, with the filename transformed by concatenating "_enhancement.wav" with the name parameter.

```python
def speech_enhancement(audio_file, name, alpha=1.0):
    # Load the audio file
    sample_rate, audio = wav.read(audio_file)

    # Convert audio to mono if it's in stereo
    if audio.ndim > 1:
        audio = np.mean(audio, axis=1)

    # Apply Short-Time Fourier Transform (STFT)
    frame_size = 0.025  # Frame size in seconds
    frame_shift = 0.01  # Frame shift in seconds
    frame_length = int(sample_rate * frame_size)
    frame_step = int(sample_rate * frame_shift)
    num_frames = int(np.ceil(len(audio) / frame_step))
    padded_size = num_frames * frame_step
    audio = np.pad(audio, (0, padded_size - len(audio)), 'constant')

    stft = np.empty((num_frames, frame_length), dtype=complex)
    for i in range(num_frames):
        frame = audio[i * frame_step : i * frame_step + frame_length]
        stft[i] = fft(frame)

    # Estimate noise spectrum using minimum statistics
    noise_frames = stft[:50]  # Use the first 50 frames as noise estimate
    noise_spectrum = np.mean(np.abs(noise_frames), axis=0)

    # Apply spectral subtraction
    enhanced_stft = np.maximum(np.abs(stft) - alpha * noise_spectrum, 0) * np.exp(1
    j * np.angle(stft))

    # Inverse Short-Time Fourier Transform (ISTFT)
    enhanced_audio = np.zeros(padded_size)
    for i in range(num_frames):
        frame = ifft(enhanced_stft[i]).real
        enhanced_audio[i * frame_step : i * frame_step + frame_length] += frame

    enhanced_audio = np.asarray(enhanced_audio[:len(audio)], dtype=np.int16)
    enhanced_audio.export("./modified_files/" + name + "_enhancement.wav", format="
    wav")
```

Figure 7.12: Python code: Speech enhancement function

7.4.6 Supporting Functions for Audio Enhancement

Apart from the mitigation techniques expounded in sections 7.4.2, 7.4.3.1, 7.4.4, and 7.4.5, there are a number of auxiliary functions that are instrumental in enhancing the calibre and applicability of audio files for ASR systems. The present subchapter centres on two fundamental operations, namely compression and the removal of silent portions located at the onset and offset of audio recordings. The aforementioned functions play a crucial role in producing audio files that are refined and streamlined, thereby augmenting the precision and efficacy of automated speech recognition procedures.

1. **Compression:** Compression is a technique in digital signal processing that is used to decrease the dynamic range of an audio signal. The mechanism involves the reduction of the amplitude of the high-intensity segments of the audio signal and the amplification of the low-intensity segments, leading to a more homogeneous and equitable auditory experience. The process of compression aids in the standardisation of audio levels, thereby mitigating the occurrence of excessively loud or soft segments in speech. Compression is a technique that reduces the dynamic range of audio signals, thereby mitigating the risk of distortion and clipping. This facilitates the accurate capture and interpretation of speech by the ASR system.

   The significance of compression is rooted in its capacity to augment the comprehensibility and uniformity of speech signals. The process of optimising the audio range facilitates the recognition and transcription of spoken words by the ASR system. In addition, the utilisation of compression techniques aids in mitigating the adverse effects of ambient noise by rendering it more uniform in relation to the audio levels at large. This leads to an enhanced signal-to-noise ratio and consequently, an improved performance of ASR systems.

7.4.6.1 Implementation technique for compression

The below code in Figure 7.13 shows the implementation of a function called "compression" that performs dynamic range compression on an audio file. The code starts by defining two parameters, "compression_ratio" and "compression_threshold." The compression ratio determines the amount of gain reduction applied to the audio signal, while the compression threshold specifies the level at which the compression starts to take effect. These values are adjusted based on the specific requirements of the audio file and the desired compression effect. Using those parameters, the code applies dynamic range compression to the input audio. The "compress_dynamic_range" function, likely from an audio processing library, is used to perform the compression. This function adjusts the amplitude of the audio signal, reducing the dynamic range and making softer parts of the audio more audible while limiting the peaks. After applying compression, the resulting audio is saved as a new WAV file. The code exports the compressed audio file to a directory named "modified_files" using the original filename with "_compressed" appended to it. This ensures that the modified audio file is stored separately and can be easily identified.

```python
def compression(audio,name):
    # Define the compression ratio and threshold
    compression_ratio = 4.0
    compression_threshold = -20.0

    # Apply the compression
    output_audio = audio.compress_dynamic_range(threshold=
    compression_threshold, ratio=compression_ratio)

    # Save the output audio file
    output_audio.export("./modified_files/" + name + "_compressed.wav", format
    ="wav")
```

Figure 7.13: Python code: Compression function

2. **Removal of silent:** The removal of silent or quiet portions from audio files is a common practise as these segments typically include extraneous noise or non-verbal sounds. The presence of silent intervals in speech does not augment the speech content and may result in undesired artefacts or prolonged periods of silence while processing ASR. The elimination of these inactive segments is a pivotal measure in the pre-processing of audio data for ASR systems.

   Through the elimination of the initial and trailing periods of silence, the audio file is efficiently condensed to encompass solely the pertinent speech material. This process effectively removes extraneous interruptions, ambient sounds, or non-verbal utterances, thereby optimising the data for ASR analysis. The act of removing periods of silence serves to not only decrease the computational resources necessary for v, but also to improve the precision and effectiveness of the system by directing attention solely towards the speech segments that contain relevant information. The significance of removing periods of silence is rooted in its ability to offer a more polished and succinct auditory input to the ASR system. The elimination of non-speech segments mitigates the risk of erroneous identification or misapprehension due to extraneous auditory material. Moreover, the act of trimming audio files enhances their overall usability by rendering them more manageable and convenient for storage, transmission, or subsequent analysis.

7.4.6.2 Implementation technique for removing the silence parts

The code below consists of two functions: "removeSilenceParts" and "detect_leading_silence."

- **removeSilenceParts:** The "removeSilenceParts" function is responsible for removing silence parts at the beginning and end of the audio file. The code first uses the "detect_leading_silence" function, which is explained below, to determine the duration of the silence at the beginning and end of the audio file. Using the information obtained from the previous step, the code trims the audio by excluding the silent parts at the beginning and end. The resulting trimmed audio is stored in the "trimmed_sound" variable. Finally, the trimmed audio is exported as a new WAV file. The code saves the file in the "modified_files" directory using the original filename with "_trimmed" appended to it.
- **detect_leading_silence:** The "detect_leading_silence" function is a helper function used by "removeSilenceParts" to determine the duration of leading silence. The function takes an audio file as input along with optional parameters such as the silence threshold and chunk size. The code checks the silence level of the audio before the first chunk using the specified silence threshold. If the silence level is above the threshold, indicating the absence of leading silence, the function returns 0. The function iterates over small chunks of audio until it finds the first chunk that exceeds the specified silence threshold, indicating the presence of sound. It keeps track of the accumulated time in milliseconds (trim_ms) to determine the duration of the leading silence. The function returns the value of trim_ms, representing the duration of the leading silence in the audio file.

The implementation of this procedure is given particular attention, as it exclusively involves the selection of the initial and final segments of the audio files for trimming purposes. The retention of potential silence in the middle sections of a recording is of utmost importance, as the elimination of initial and trailing silence serves distinct purposes, such as facilitating the comprehension of words with numerous compounds.

7.5 Visualization of Audio Files

The Figure 7.15 presented illustrates a heatmap that exhibits the Mel-Frequency Cepstral Coefficients (MFCC) obtained from a set of audio recordings. The arrangement of coefficients is organised in rows, denoting distinct features, and columns correspond to individual clips. The heatmap displays color-coded cells that correspond to coefficient values,

```python
def removeSilenceParts(audio,name):

    start_trim = detect_leading_silence(audio)
    end_trim = detect_leading_silence(audio.reverse())

    duration = len(audio)
    trimmed_sound = audio[start_trim:duration-end_trim]
    # Export the trimmed sound to a WAV file

    trimmed_sound.export("./modified_files/" + name + "_trimmed.wav", format="wav")

def detect_leading_silence(sound, silence_threshold=-50.0, chunk_size=10):
    trim_ms = 0 # ms

    assert chunk_size > 0 # to avoid infinite loop

    # Check the silence level of the audio before the first chunk
    pre_chunk_silence = sound[:chunk_size].dBFS
    if pre_chunk_silence >= silence_threshold:
        return 0

    # Iterate over chunks until you find the first one with sound
    while sound[trim_ms:trim_ms+chunk_size].dBFS < silence_threshold and trim_ms < len(sound):
        trim_ms += chunk_size

    return trim_ms
```

Figure 7.14: Python code: Removing the silence parts

with blue representing negative values, white representing zero, and red representing positive values. The present visualisation provides valuable insights regarding the coefficients' variations across diverse clips and features.

The initial trio of subfigures depicts the dispersion of an individual MFCC throughout numerous audio clips. In contrast, the final subfigure amalgamates twelve recordings, demonstrating a ubiquitous pattern that is shared among all of the recordings. The Mel-frequency cepstral analysis yields the MFCC coefficients that are extensively employed in speech recognition for the purpose of capturing the spectral attributes of sound signals. The dataset in question pertains to the Greek term "deka," which consists of two syllables separated by a brief pause.

The heatmap analysis reveals that the coefficient values in the bottom two rows are comparatively higher, which suggests a higher energy level in the higher frequency bands as opposed to the lower frequency bands. This trait indicates the existence of sounds with high frequencies, such as those generated by whistles or the vocalisations of females. It is imperative to acknowledge that the explication of MFCC values may exhibit variability contingent upon the contextual framework and the particular sound under scrutiny. It is

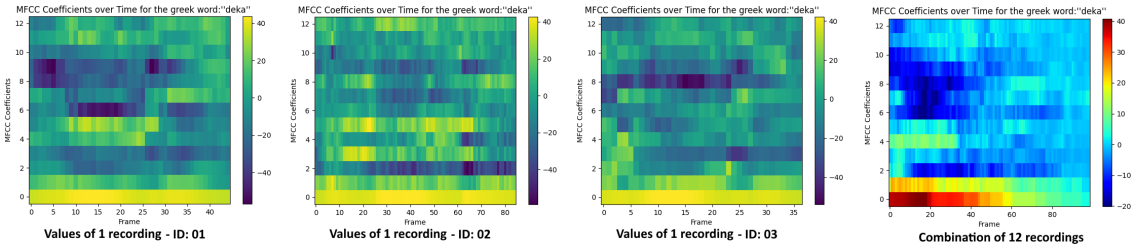


Figure 7.15: Heatmap of MFCC Coefficients for Greek Word 'Deka'.

plausible that the individuals involved in this particular dataset endeavoured to enunciate their words with utmost clarity, potentially leading to a perceived elevation in pitch.

The Greek language exhibits a varied set of vowels, which may be manifested in the allocation of energy across the upper frequency bands of the MFCC. The analysis of the spoken word "de-ka" indicates a notable concentration of energy in the higher frequency bands, as revealed by the visualisation. The possible cause of this phenomenon may be linked to the existence of sounds with high frequency, such as fricative or affricate consonants, or vowel sounds that fall within the higher frequency range.

Additionally, the heatmap illustrates a comparatively uniform dispersion of coefficient values, with the exception of rows 2, 8, and 9, which exhibit diminished values in contrast to the remaining rows. The decreased numerical values observed in rows 8 and 9 may imply an absence of energy within the associated frequency ranges, which could be interpreted as a momentary interruption or cessation of sound between the two syllables. This observation is consistent with the phonological structure of the recorded term "de-ka," which comprises of a disyllabic sequence separated by a brief pause.

To summarise, the aforementioned figure presents a thorough outline of the MFCC characteristics and their prospective value in the classification or recognition of speech. This statement underscores the efficacy of MFCC in capturing crucial details pertaining to the spectral envelope of an audio signal.

7.6 Evaluation of Audio Cleaning

The objective of this chapter is to assess and choose the optimal strategies for addressing the difficulties presented by the four essential parameters ranging from CHA-1 to CHA-4. The challenges discussed in the previous chapters have been comprehensively analysed and viable solutions have been proposed. The present discourse centres on a thorough

assessment of the aforementioned proposed tactics. The principal aim is to evaluate the compromises linked with every proposed resolution and appraise their pragmatic ramifications. The objective at the conclusion of the chapter is to ascertain and propose the most suitable methods of mitigation that exhibit the highest capacity to tackle the aforementioned parameters. The ultimate objective is to augment the comprehensive calibre and precision of ASR systems.

7.6.1 Evaluation of the parameter CHA-1

In the context of ASR, the trade-offs between noise reduction and oversmoothing techniques need to be carefully considered to enhance the accuracy and reliability of the speech recognition system. While both techniques aim to improve the quality of the audio input, their impact on ASR performance differs. The main objective of noise reduction methods is to address the issue of ambient noise, which poses a considerable obstacle in the context of ASR. The objective of these techniques is to improve the clarity and comprehensibility of speech by attenuating extraneous noise, thereby augmenting the signal-to-noise ratio. It is crucial to recognise that an overabundance of noise reduction techniques may unintentionally eliminate significant speech characteristics, ultimately resulting in the forfeiture of crucial information required for precise speech recognition. Additionally, aggressive noise reduction can introduce artifacts or distortions that could negatively impact the ASR system's performance.

Conversely, oversmoothing methodologies aim to mitigate sudden variations and fluctuations in the auditory signal. While oversmoothing can help improve the visual appearance of the signal and reduce high-frequency noise, it can also cause a loss of fine-grained details and distort the original speech characteristics. The loss of such details may negatively affect the ASR system's ability to accurately capture phonetic nuances, resulting in decreased recognition accuracy and potential misinterpretation of speech.

Given the trade-offs involved, as shown in Table 7.1 and Table 7.2, it is recommended that, in the particular scenario under consideration, the ASR system would be **better served by employing noise reduction techniques** as opposed to oversmoothing. Although oversmoothing may enhance the signal's visual appeal, noise reduction directly tackles the main obstacle of background noise, which has a substantial impact on the performance of ASR. The ASR system can improve its overall accuracy by enhancing the signal-to-noise ratio through a focus on noise reduction, which in turn allows for better recognition of speech patterns.

Persona Name	Limitations
Reduces background noise interference, thereby improving speech intelligibility.	Important speech details and acoustic cues may be lost.
Increased signal-to-noise ratio, reducing noise-induced errors.	Overreliance on noise reduction may result in speech signal distortion or alteration.
Improved speech recognition accuracy, particularly in noisy environments.	Inaccurate noise estimation or overzealous noise reduction may result in the introduction of artefacts.
Improved speech-to-noise separation improves ASR performance.	Insufficient noise suppression, resulting in noise-related ASR errors.
-	The presence of residual noise or artefacts in the processed audio, which inhibits the performance of the ASR.
-	Due to ineffective noise reduction, speech characteristics become distorted or unnatural.
-	Inefficient algorithms for noise reduction may increase computational complexity.

Table 7.1: Advantages and Limitations of noise reduction techniques for ASR.

Thus, considering the aim of improving ASR accuracy while mitigating the influence of ambient noise, the choice to prioritise noise reduction methods over excessive smoothing is rational. This decision guarantees that the ASR system can proficiently manage audio inputs from real-world scenarios that exhibit diverse levels of noise, leading to more dependable and precise speech recognition results.

7.6.2 Evaluation of the parameter CHA-2

In the realm of ASR, it is crucial to assess the benefits and drawbacks of speaker variability training sets and speaker normalisation techniques when making trade-offs. Both methodologies provide unique advantages that can substantially improve the efficiency of an ASR system.

The utilisation of a speaker variability training set confers various benefits. The ASR system can attain greater robustness and adaptability to diverse speaking styles and variations by subjecting it to a wide range of speakers during the training process. The aforementioned capability allows the system to exhibit a high degree of generalisation towards speakers who were not previously encountered, thereby mitigating the influence of

Advantages	Limitations
Potential reduction of background noise interference.	Loss of important speech details and acoustic cues.
Suppression of certain types of noise, such as stationary or continuous background noise.	Distortion or alteration of speech signal, leading to reduced intelligibility.
Reduction of overall noise level, which may enhance speech-to-noise ratio.	Potential introduction of artifacts or unnatural speech characteristics.
Improved speech-to-noise separation improves ASR performance.	Insufficient noise suppression, resulting in noise-related ASR errors.
-	Oversmoothing can lead to the loss of critical speech features, negatively impacting ASR accuracy.
-	Potential reduction in speech intelligibility due to the removal of important speech cues.
-	Introduction of unnatural or distorted speech characteristics, affecting ASR system performance.

Table 7.2: Advantages and Limitations of oversmoothing techniques for ASR.

speaker-dependent biases and enhancing the overall precision of the system. Moreover, the training set for speaker variability promotes speaker autonomy, thereby enhancing the ASR system's ability to effectively manage real-life situations characterised by significant variations in speakers' attributes.

Conversely, techniques for speaker normalisation present a distinct array of benefits. The objective is to minimise the impact of individual speaker traits on the ASR system by lessening the speaker-dependent variances. Through the process of speech signal normalisation, these techniques improve the system's capacity to model and identify speech patterns that are not limited to individual speakers, thereby facilitating generalisation to speakers who have not been previously encountered. The normalisation of speaker characteristics is a useful technique for comparing and analysing speech data from multiple speakers, rendering it advantageous in diverse applications.

Although both methodologies possess their own merits, it is imperative to recognise their respective constraints. The process of obtaining a training set for speaker variability necessitates a diverse and representative group of speakers, which can be a challenging and

resource-intensive task. The efficacy of speaker normalisation methods is contingent upon precise assessment of speaker attributes, which may not always be attainable, and there exists a conceivable hazard of introducing anomalies or deformations in the speech signal while executing the normalisation procedure.

Advantages	Limitations
Improved robustness to speaker-dependent variations.	Requires a diverse and representative set of speakers for training, which can be resource-intensive.
Better generalization to unseen speakers.	Difficulty in capturing all possible speaker variations.
Increased adaptability to different speaking styles.	Speaker variability alone may not address all sources of variation in the speech signal.
Reduces speaker-specific biases.	Potential risk of overfitting to training speakers.

Table 7.3: Advantages and Limitations of speaker variability training set.

Advantages	Limitations
Reduces speaker-dependent variations.	Speaker normalization techniques may introduce artifacts or distortions in the speech signal.
Helps improve speaker-independent modeling.	Requires accurate estimation of speaker characteristics, which may not always be feasible.
Enhances generalization to unseen speakers.	Incomplete or inaccurate normalization can lead to loss of important speaker-specific information.
Mitigates the influence of speaker variability.	Normalization techniques may not fully eliminate all sources of variation, such as pronunciation errors.

Table 7.4: Advantages and Limitations of techniques for normalization techniques for ASR.

Given the manifold benefits afforded by the **utilisation of both speaker variability training sets and speaker normalisation techniques**, as shown in Table 7.3 and Table 7.4, it is advisable to integrate both methodologies into the project. The collaborative utilisation of their respective strengths can lead to a synergistic enhancement of the performance of the ASR system. Through the utilisation of speaker variability training set,

the system can attain enhanced resilience, flexibility, and diminished speaker partialities. The utilisation of speaker normalisation techniques concurrently aids in the reduction of speaker-dependent variations, amplification of generalisation, and simplification of comparison among speakers.

The project aims to enhance the reliability and accuracy of the ASR system by incorporating both speaker variability training set and speaker normalisation techniques. This integration allows the project to leverage the benefits of each approach. The integration of these components facilitates the efficient management of diverse speech patterns, adaptation to speaker idiosyncrasies, and enhancement of global efficacy, in accordance with the objectives of the study to attain superior speech recognition results.

7.6.3 Evaluation of the parameter CHA-3

In the realm of ASR, it is imperative to assess the benefits and drawbacks of reverberation and noise reduction methods, as well as their potential efficacy in addressing CHA-3. Both methodologies possess the capability to tackle environmental variables such as ambient noise and sound reflection, however, their appropriateness for ASR necessitates a meticulous evaluation.

The implementation of techniques aimed at reducing reverberation presents benefits in terms of minimizing the influence of the acoustic properties of a room and ameliorating the deterioration resulting from the presence of echoes. The implementation of these techniques can potentially enhance speech intelligibility and alleviate the negative impact of environmental factors by minimizing reverberation. The precise determination of the Room Impulse Response (RIR) can pose difficulties, and the efficacy of reducing reverberation is contingent upon the particular attributes of the room. Moreover, there exists a potential hazard of introducing artifacts or distortions in the processed speech, which may have an adverse impact on the accuracy of ASR. In light of the constraints identified and their consequential influence on the accuracy of ASR, the project team has made the determination to abstain from implementing reverberation reduction methods in the project.

In contrast, the implementation of noise reduction techniques presents notable benefits in ameliorating the influence of ambient noise, a prominent environmental variable that detrimentally affects ASR efficacy. The implementation of these techniques results in an improvement of speech intelligibility and a reduction of noise, leading to a more precise and refined speech signal, as shown in Table 7.1. As previously stated, in subsection 7.6.2, the project team has opted to incorporate noise reduction methodologies into the project's

workflow. The aforementioned decision is consistent with the project's aim of enhancing ASR precision through the reduction of the impact of ambient noise.

Advantages	Limitations
Improved robustness to reverberant environments.	Accurate estimation of RIR can be challenging.
Enhances speech intelligibility in reverberant spaces.	The effectiveness of reverberation reduction depends on the specific room characteristics.
Helps minimize the impact of room acoustics.	Reverberation reduction techniques may introduce artifacts or distortions in the processed speech.
Mitigates the degradation caused by echoes.	Highly reverberant environments may still pose challenges for accurate speech recognition, despite reduction.

Table 7.5: Advantages and Limitations of reverberation technique for ASR.

To sum up, the utilization of both reverberation and noise reduction techniques exhibits the capability to alleviate environmental factors in ASR. Although reverberation reduction techniques have benefits in managing reverberation and room acoustics, their constraints and probable influence on ASR precision have prompted the project team to refrain from utilizing them. By way of comparison, the implementation of noise reduction techniques yields substantial advantages in ameliorating ambient noise, a pivotal environmental variable that impacts the efficacy of ASR. Hence, the implementation strategy **prioritizes the reduction of noise** as a dependable method for enhancing the precision of ASR in the face of environmental variables.

7.6.4 Evaluation of the parameter CHA-4

The investigation conducted by the project pertaining to the reduction of pronunciation errors has uncovered a number of efficacious methodologies, such as speech enhancement, the utilization of language models, and the incorporation of diverse speakers, as expounded upon in preceding sections. The employment of these methodologies in conjunction serves to effectively tackle the issue of mispronunciation in ASR systems.

The utilization of speech enhancement techniques presents benefits in the enhancement of speech intelligibility and the amplification of the signal-to-noise ratio. The process of

speech enhancement is of paramount importance in mitigating the influence of environmental factors on the precision of pronunciation. This is achieved through the reduction of background noise and the minimization of distortion and artifacts. It is crucial to take into account the constraints related to the precise differentiation of speech from ambient noise and the probable forfeiture of significant speech data while enhancing it. Furthermore, it is important to consider the computational complexity and potential latency that may arise from the implementation of speech enhancement algorithms.

The utilization of language models for the purpose of reducing pronunciation errors yields considerable advantages. Language models can enhance the contextual relevance of the ASR system by improving its accuracy in handling pronunciation variations, recognizing words with variations, and better handling regional accents and dialects. The careful consideration of the availability and quality of training data is imperative for the language model. It is plausible that valid pronunciations may be overcorrected or unfamiliar/non-standard pronunciations may be misinterpreted. Moreover, the incorporation and application of the linguistic model may necessitate augmented computational capabilities.

Moreover, the project acknowledges the noteworthy influence of employing diverse speakers to tackle pronunciation inaccuracies. As elucidated in preceding sections, the incorporation of varied speech patterns, accents, and dialects facilitates the conditioning of the ASR mechanism to identify and comprehend a broader spectrum of pronunciations. This methodology improves the resilience and versatility of the system in accommodating diverse speech patterns.

Advantages	Limitations
Improved speech intelligibility.	Difficulty in accurately separating speech from background noise or interference.
Enhanced signal-to-noise ratio.	Potential loss of important speech information during the enhancement process.
Reduction of background noise.	Sensitivity to the quality of the input audio and the specific characteristics of the noise.
Minimization of distortion and artifacts.	Computational complexity and potential latency introduced by the speech enhancement algorithms.

Table 7.6: Advantages and Limitations of speech enhancement technique for ASR.

After careful consideration of the trade-offs and time constraints associated with the project, the team has opted to integrate **multiple speakers**, as previously discussed in

Advantages	Limitations
Improved speech intelligibility.	Difficulty in accurately separating speech from background noise or interference.
Enhanced signal-to-noise ratio.	Potential loss of important speech information during the enhancement process.
Reduction of background noise.	Sensitivity to the quality of the input audio and the specific characteristics of the noise.
Minimization of distortion and artifacts.	Computational complexity and potential latency introduced by the speech enhancement algorithms.

Table 7.7: Advantages and Limitations of using language models.

subsection 7.6.2, and **employ language models** as the ultimate strategies. The aforementioned decisions were made after a thorough evaluation of the benefits of utilizing speech enhancement techniques to rectify pronunciation errors, while also taking into account the potential obstacles and constraints that may arise. The project endeavors to enhance the performance of the ASR system and provide precise outcomes in diverse pronunciation scenarios by utilizing the variety of speakers and exploiting the contextual comprehension offered by language models.

7.6.5 Comprehensive Evaluation of ALL Challenges

This subsection provides a thorough assessment of various methods utilized for the purpose of ASR. The objective is to evaluate the efficacy of said techniques in achieving diverse evaluation objectives within the framework of ASR systems. The assessment comprises crucial facets such as comprehensibility of speech, existence of ambient noise, elimination of significant speech data, general enhancement in audio caliber, cost-benefit ratio, simplicity of integration and utilization, and duration of audio processing. The objective is to offer a comprehensive perspective on the methodologies and their efficacy in augmenting ASR capabilities. This assessment facilitates the process of making well-informed decisions pertaining to the choice and execution of suitable methodologies, taking into account their efficacy in addressing prevalent obstacles in automatic speech recognition. This resource is highly valuable for individuals involved in research, practice, and development who aim to enhance the functionality and user-friendliness of ASR systems across diverse applications and settings.

Legend:

✓ (Single Tick): Indicates a positive outcome or effectiveness in relation to the corresponding evaluation objective.

✗ (Single Cross): Indicates a negative outcome or ineffectiveness in relation to the corresponding evaluation objective.

✓✓ (Double Tick): Represents a higher level of positive outcome or effectiveness compared to other techniques in relation to the corresponding evaluation objective.

✗✗ (Double Cross): Represents a higher level of negative outcome or ineffectiveness compared to other techniques in relation to the corresponding evaluation objective.

The evaluation metrics chosen for assessing the performance of the audio cleansing techniques were meticulously chosen to address key aspects pertinent to the ASR system. Each evaluation metric serves a distinct function in assessing the efficacy and suitability of the techniques for ASR applications.

ID	Name of the technique
CT-1.	Remove Background Noise.
CT-2.	Remove Silence Parts.
CT-3.	Normalization.
CT-4.	Compression.
CT-5.	Use of Variety of Speakers.
CT-6.	Over-smoothing.
CT-7.	Reverberation.
CT-8.	Speech Enhancement.
CL-9.	Employ Language Models.

Table 7.8: Identifiers of various Cleaning Techniques(CT).

- **Understandability of Speech:** This metric seeks to evaluate the clarity and intelligibility of the speech after the cleaning techniques have been applied. It ensures that the processed audio remains understandable, allowing the ASR system to accurately transcribe the speech. A checkmark indicates enhanced speech clarity and intelligibility, whereas a cross indicates a detrimental effect on speech comprehension.
- **Presence of Background Noise:** Background noise can degrade the efficacy of ASR systems significantly. Therefore, evaluating the efficacy of noise reduction techniques is essential for minimizing the impact of background noise and enhancing the

overall quality of speech. A checkmark indicates that external noise has been effectively reduced or eliminated, whereas a cross indicates that noise reduction has been ineffective.

- **Removal of Important Speech Information:** When removing noise, it is essential to ensure that essential speech information is not eliminated or distorted unintentionally. This metric is used to evaluate a technique's ability to eradicate noise selectively while preserving crucial speech details. A checkmark indicates the effective preservation of crucial speech details, whereas a cross indicates the unintentional removal of such details.

- **Overall Improvement in Audio Quality:** This metric evaluates the overall improvement in audio quality brought about by the cleaning techniques, taking into account factors such as the speech's intelligibility, fidelity, and naturalness. A checkmark denotes an improvement in audio quality across the board, including clarity, intelligibility, fidelity, and naturalness. A cross signifies a deterioration in audio quality.

- **Cost-effectiveness of the technique:** Important considerations for "cost-effectiveness of the technique" include implementation feasibility and resource requirements. Evaluation of the cost-effectiveness metric identifies techniques that establish a balance between computational complexity, time efficiency, and financial costs, ensuring their applicability in real-world situations. Tick is a solution that is efficient in terms of computational resources, time, and financial expenditures. A cross represents a less cost-effective strategy.

- **Ease of Implementation and Use:** Usability and ease of implementation are crucial factors in the selection of cleaning techniques. This metric enables us to identify techniques that are simple to integrate into existing systems and require minimal user input. The checkmark denotes the technique's simple implementation and user-friendliness. A cross signifies implementation or usage difficulties or complexities.

- **Time Required to Process the Audio:** Processing time plays a crucial role in real-time applications. This metric assesses the effectiveness and speed of audio file cleansing techniques, ensuring that the selected techniques can process audio within acceptable time constraints. A checkmark indicates a quick and efficient processing time, whereas a cross indicates lengthier processing times or inefficiency in audio file management.

By selecting these specific evaluation metrics, we ensure a comprehensive assessment of the performance of the cleaning techniques in the context of ASR, taking into account various critical aspects such as speech understandability, noise reduction, speech information preservation, overall audio quality, cost-effectiveness, ease of implementation, and processing efficiency.

Table 7.9 presents a comprehensive tabular comparison of diverse evaluation techniques for ASR within the framework of distinct cleaning techniques. The aforementioned table presents a comprehensive summary of the efficacy of individual techniques in tackling primary evaluation objectives. The assessment methodologies comprise a total of seven and encompass a wide spectrum of facets that are pertinent to ASR systems. The tabular format facilitates a straightforward evaluation of the methodologies, revealing their respective efficacy. This thorough assessment facilitates informed decision-making when choosing the most appropriate methods for enhancing ASR accuracy in diverse contexts.

Evaluation objectives	CT-1	CT-2	CT-3	CT-4	CT-5	CT-6	CT-7	CT-8	CT-9
Understandability of speech	✓✓	✓	✓✓	✓	✓✓	✗✗	✓✓	✓✓	✓✓
Presence of background noise	✓✓	✓	✓	✗✗	✓ ✓	✗✗	✓	✓ ✓	✓ ✓
Removal of important speech information	✗✗	✗✗	✗✗	✗✗	✓✓	✓✓	✗✗	✓	✓✓
Overall improvement in audio quality	✓✓	✗✗	✓✓	✓✓	✓✓	✗✗	✓	✓✓	✓✓
Cost-effectiveness of the technique	✓✓	✓✓	✓	✓	✓✓	✓	✗✗	✓✓	✓✓
Ease of implementation and use	✓✓	✓✓	✓	✓	✓✓	✓	✗✗	✓✓	✓✓
Time required to process the audio	✓✓	✓✓	✓	✓	✓✓	✗✗	✗✗	✓	✓✓

Table 7.9: Evaluation of ASR Cleaning Techniques

The present illustration showcases the process of interpreting the evaluation table and acquiring valuable insights regarding the efficacy of individual techniques in mitigating

background noise to enhance ASR performance. Various methods were examined to determine their efficacy in reducing the impact of ambient noise during the assessment of its existence. The assessment of subsections 7.6.1, 7.6.2, 7.6.3, and 7.6.4 revealed that the methods of noise reduction and speech enhancement exhibited favorable results in mitigating the influence of ambient noise, as evidenced by a solitary checkmark in the respective cells of the table. In terms of efficacy, speech enhancement exhibited a superior level of achievement in mitigating ambient noise, as indicated by the presence of two check marks in its corresponding cell. This indicates that speech enhancement yielded superior results compared to noise reduction in mitigating the effects of ambient noise. Alternative methodologies, such as incorporating diverse speakers and utilizing linguistic models, demonstrated a degree of efficacy, albeit not comparable to that of speech enhancement. On the other hand, it was observed that methods such as over-smoothing and dereverberation exhibited lower efficacy in reducing the impact of ambient noise, as denoted by the symbol of a single cross in the corresponding cells. In general, the assessment underscores the differing levels of efficacy exhibited by distinct approaches in mitigating the impact of ambient noise, with speech enhancement being identified as the most efficacious technique in this regard.

7.6.6 Implementing Combined Audio Cleaning Techniques

The Python code presented in Listing 7.16 includes a function named clean_audio_files that is designed to enhance the quality of audio files utilized in automatic speech recognition machine learning models. The program sequentially traverses a collection of filenames denoted as metadata and executes a set of procedures aimed at enhancing the quality of each audio file. Subsequently, the code executes a series of cleansing methodologies on the audio file with the objective of improving its appropriateness for automated speech recognition. The aforementioned techniques are chosen based on the assessment of the preceding subsections. The code incorporates a visualization step that offers valuable insights into the impact of individual cleaning techniques on the audio file. The code facilitates a more comprehensive comprehension of the enhancements attained through the employed cleansing methodologies by means of visualizing the audio files at various phases.

7.7 RQ1: Factors Affecting ASR Accuracy

The first research question **(RQ1)** aims to identify the primary factors that significantly affect the accuracy of ASR and explores effective strategies to mitigate their impact. The

```python
def clean_audio_files(metadata):
    directory = "./downloaded_files/"
    for item in metadata:
        full_path = directory + item
        if not os.path.exists(full_path):
            print(f"File {full_path} does not exist. Skipping...")
            continue
        file_name_with_extension = os.path.basename(full_path)
        file_name_without_extension = os.path.splitext(file_name_with_extension)[0]
        sound = AudioSegment.from_file(full_path)
        print(full_path)
        visualization(sound,file_name_without_extension,"dB","Original audio file")
        noice_reduction(sound,file_name_without_extension)
        sound_noice_reduction = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_noice_red"+".wav")
        visualization(sound_noice_reduction,file_name_without_extension + "
    _noice_red", "dB", "Audio file after noise reduction")
        removeSilenceParts(sound_noice_reduction,file_name_without_extension)
        sound_trimmed = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_trimmed"+".wav")
        visualization(sound_trimmed,file_name_without_extension + "_trimmed", "dB",
    "Audio file after removing silence parts")
        normalization(sound_trimmed, file_name_without_extension)
        sound_normalized = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_normalized"+".wav")
        visualization(sound_normalized,file_name_without_extension + "_normalized",
     "dB", "Audio file after normalization")
        oversmoothing(sound_trimmed,file_name_without_extension, window_size=5)
        sound_overmoothing = AudioSegment.from_file("./modified_files/" +
    file_name_without_extension + "_overmoothing"+".wav")
        visualization(sound_overmoothing,file_name_without_extension + "
    _overmoothing", "dB", "Audio file after overmoothing")
```

Listing 3: Python code: Final cleaning techniques

Figure 7.16: Python code: Final cleaning techniques

investigation has revealed that various crucial factors have been identified, such as **background noise, speaker variability, environmental factor, pronunciation errors**. The aforementioned factors have been extensively acknowledged as significant obstacles in attaining a high degree of accuracy in ASR.

The presence of background noise presents a notable hindrance to precise speech recognition. In order to tackle this issue, various methods for reducing noise have been widely utilized. Through the implementation of noise suppression techniques and the enhancement of the signal-to-noise ratio, the capacity of the ASR system to identify speech patterns is augmented, leading to a subsequent improvement in accuracy.

The performance of ASR is significantly influenced by the factor of speaker variability. In order to reduce its impact, two primary strategies have been implemented, namely, the utilization of training sets that incorporate speaker variability and the application of techniques for speaker normalization. Training sets that incorporate speaker variability expose the ASR system to a range of speakers with distinct speech patterns, speaking styles, and individual characteristics. This enables the system to acquire knowledge and adjust to diverse speech inputs. The implementation of speaker normalization techniques serves to mitigate the influence of speaker-specific variations, thereby fostering equitable comparisons across speakers and augmenting the capacity for generalization.

ASR accuracy can be influenced by environmental factors, including reverberation. Although various methods for reducing reverberation have been explored, their impact on the accuracy of ASR has prompted us to refrain from incorporating them into our project. Our research has primarily centered on the implementation of noise reduction techniques. These techniques have been found to be effective in mitigating the issue of ambient noise, thereby leading to a substantial enhancement in the accuracy of ASR systems.

ASR systems encounter an additional obstacle in the form of inaccuracies in pronunciation. The employment of language models has demonstrated significant efficacy in reducing their impact. The accuracy of ASR can be enhanced by integrating contextual comprehension, grammar, and vocabulary knowledge into language models, which facilitate precise word recognition.

7.8 RQ2: Advantages and Trade-Offs of Audio Cleaning Techniques

The second research question **(RQ2)** delves into the advantages and trade-offs of employing various combinations of audio cleaning techniques in the preparation of audio files for

machine learning models used in automatic speech recognition. The aim is to ascertain the optimal cleaning methodologies that can enhance the ultimate outcomes of the ASR system.

By conducting a comprehensive assessment and juxtaposition of diverse cleaning methodologies, encompassing noise reduction, speaker variability training sets, speaker normalization, the incorporation of multiple speakers, and language models, the benefits and drawbacks of each approach were scrutinized.

The chosen cleaning techniques have shown notable benefits in effectively tackling particular difficulties. The implementation of noise reduction techniques has been observed to have a substantial impact on the signal-to-noise ratio, thereby resulting in the enhancement of speech pattern recognition and improved accuracy of ASR systems. The utilization of speaker variability training sets and speaker normalization techniques has been found to promote adaptability, versatility, and decreased speaker biases, while also decreasing speaker-specific variations and enhancing the ability to generalize.

The incorporation of multiple speakers has demonstrated significant advantages in managing a wide range of accents, dialects, and speech patterns, leading to a more resilient and versatile ASR mechanism. Finally, the integration of linguistic models offers contextual comprehension, grammatical accuracy, and lexical proficiency, thereby augmenting the precision of ASR systems.

Nevertheless, there are trade-offs associated with the utilization of these techniques. Reverberation reduction techniques, although advantageous in the realm of room acoustics management, impose certain constraints that may have an adverse effect on the precision of ASR. In light of the trade-offs involved and a thorough assessment of temporal limitations, our project has made the decision to abstain from employing reverberation reduction techniques.

To sum up, the utilization of a blend of methods for reducing noise, training sets for speaker variability, techniques for normalizing speakers, the incorporation of diverse speakers, and language models results in significant advantages for enhancing the precision and efficacy of automatic speech recognition systems. The aforementioned techniques are efficacious in mitigating crucial factors that exert a substantial impact on the accuracy of ASR systems, such as ambient noise, speaker diversity, and enunciation inaccuracies. Through the reduction of the influence of these variables and the optimization of the advantages of each method, the ASR system can attain greater precision, increased robustness, and

improved recognition aptitudes, resulting in more accurate results across a range of pronunciation contexts.

Moving forward, the subsequent chapter, "Discussion," will critically analyze the findings, draw insightful conclusions, and propose valuable recommendations for further improvement and future research in the domain of language preservation and speech recognition.

8

Discussion

The "Discussion" chapter examines the analysis and ramifications of the research findings, situating them in a broader context. This section aims to provide a comprehensive understanding of the implications of the proposed work for current and future research in the field, as well as for practitioners. Examining the results in relation to the research questions provide insight into the project's significance and potential impact. The discussion chapter contributes to the advancement of knowledge in the field and the direction of future research and practise through its analysis and interpretation of the findings.

8.1 Leveraging Crowdsourcing for Dagbani Language Database Creation

The fundamental characteristic of our ASR system is centered on a targeted crowdsourcing application that enables users to capture audio recordings of themselves uttering words in the Dagbani language. The transcriptions of these spoken words function as a repository of information used to educate a machine learning algorithm in the development of voice recognition technology. The application is accessible on both mobile and web platforms. The mobile application provides offline capabilities, enabling users to record words even without an internet connection. Additionally, it incorporates an automatic upload feature to the Firebase server, which activates once an internet connection is established.

Our crowdsourcing application is accessible to individuals who possess a high level of proficiency in the Dagbani language, with the objective of constructing an extensive repository of vocabulary. Nevertheless, it is important to recognise that individuals who have limited resources may encounter challenges when it comes to effectively utilising digital systems and gaining access to the necessary devices or internet connections.

In order to address these challenges, our ASR system integrates support mechanisms specifically designed to mitigate the digital literacy gap. The design principles of NASA are adhered to in order to ensure that the application is user-friendly, intuitive, and supplemented with informal pop-up notifications. In addition, our organisation offers tutorial videos and facilitates workshops in Ghana, providing comprehensive guidance on the utilisation and operational aspects of the application. Furthermore, in order to cater to the constraints of limited internet connectivity, the offline mobile application allows users to capture and store words offline, with the ability to later upload them once an internet connection becomes accessible.

The support mechanisms have two primary objectives. Our primary objective is to enhance the accessibility of our ASR system, thereby facilitating effective user engagement with the application among a broader range of users. Additionally, our efforts are directed towards enhancing the precision of transcribed text through the provision of streamlined user interfaces, support for multiple languages, and solutions that can be accessed offline or with limited bandwidth.

Through the implementation of these mechanisms, our objective is to address the digital literacy gap and enable individuals with limited resources to actively engage in our ASR system. It is our contention that the provision of tailored digital literacy training programmes, designed to meet the specific needs of individuals with limited access to resources, alongside simplified user interfaces and offline capabilities, can cultivate an inclusive atmosphere that empowers users to develop crucial digital literacy skills and effectively utilise our ASR technology.

In brief, the system aims to mitigate the digital literacy gap through the incorporation of various support mechanisms, including a user-friendly interface, offline capabilities, instructional videos, and workshops. The primary objectives of these mechanisms are to augment user accessibility, improve the precision of transcribed words, and ultimately enable individuals with limited resources to actively engage in the crowdsourcing initiative, thereby bolstering the overall efficacy of the ASR system.

8.2 A Comparative Analysis of the Mobile and Web Apps in the ASR System

The primary functionalities of the mobile application are centred on the capturing and storing of linguistic expressions. Individuals possess the capacity to document a solitary term or generate an inventory of terms classified under specific categories. Individuals

have the option to engage in activities such as reviewing their recorded materials, making additional recordings if deemed necessary, or submitting said recordings for the purpose of contribution. The verbal utterances are stored in a local repository on the user's device, such as a smartphone or tablet, thereby facilitating offline capabilities. The upload functionality enables users to assess their recordings and choose specific ones for transfer to the Firebase cloud storage service, contingent upon the presence of an internet connection. Furthermore, the mobile application incorporates informative sections such as the "About Us" page, which serves to furnish users with comprehensive project information.

In addition, the web application places emphasis on the documentation of vocabulary. Individual users have the capability to document individual words or generate word lists, with the added functionality of excluding specific words if desired. In a manner akin to the mobile application, individuals have the capability to engage in auditory playback of their recorded content, re-record as necessary, and subsequently submit said recordings for the purpose of contribution. In the web application, the recordings are uploaded to the Firebase service automatically upon submission, as it is assumed that web application users have access to internet connectivity. The web application also encompasses informative sections, namely "About Us," "Terms and Conditions," and a licensing agreement. In addition, the web application provides the added benefit of enabling users to download the mobile application directly through a designated button, thereby enhancing its accessibility to a wide range of users utilising various platforms.

The presence and utilisation of these features have a significant influence on user engagement and data contribution. The mobile application is designed to accommodate users who may experience restricted or sporadic internet connectivity, enabling them to capture and store words without an active online connection. After the establishment of an internet connection, individuals have the ability to selectively upload their recorded verbal expressions, thereby making a contribution to the collective dataset. The feature of offline capability facilitates increased user engagement and fosters active participation in regions with restricted access to internet connectivity.

The mobile and web applications of our ASR system provide unique characteristics and capabilities that are tailored to accommodate diverse user contexts and preferences. The observed disparities in functionality between the mobile and web applications are indicative of the diverse requirements and utilisation behaviours exhibited by our user base. The mobile application offers offline functionalities, catering to users who have restricted or sporadic internet connectivity. In contrast, the web application provides a more efficient

method for recording and submitting data, thereby offering convenience to users who have reliable access to the internet.

The ongoing data collection aims to acquire comprehensive information regarding user behaviour across various platforms, encompassing patterns of usage and trends in contribution. The present study endeavours to conduct a thorough examination of the distinct advantages and disadvantages linked to each application under investigation. By gaining insights into the preferences and patterns of user engagement, it is possible to enhance and optimise mobile and web applications in order to improve the overall user experience and increase the quantity and quality of user-generated content. By persisting in the process of data collection and analysis, we can acquire a greater quantity of valuable insights and enhance our system through the incorporation of user feedback and behaviour.

In the "Discussion" chapter, the research findings on audio cleaning techniques and crowdsourcing for Dagbani language database creation are thoroughly analyzed, assessing their impact on the ASR system's accuracy and the language preservation platform. Introducing the next chapter, "Related Project in Contrast with Existing Work", it provides a comprehensive review of existing literature and research in language preservation, automatic speech recognition, and crowdsourcing for linguistic databases. By situating the current research within the broader academic context, this chapter highlights its novel contributions to the field.

9

Related Project in Contrast with Existing Work

This chapter delves into the existing literature on the subject of crowdsourcing speech data for languages with limited resources. This study analyzes a collection of scholarly articles that share a mutual aim of utilizing crowdsourcing and machine learning methodologies to conserve and advance indigenous languages. By means of a comparative analysis, we aim to elucidate the similarities and differences between these papers, thereby highlighting their distinctive contributions and methodologies. Through an examination of current initiatives, valuable insights can be obtained regarding the difficulties and possibilities related to the acquisition of spoken data for languages with limited resources. In the following discourse, we shall examine the intricacies of said papers and assess their import in the progression of language technologies.

The papers chosen for the comparative analysis were selected through the utilization of the Connected Papers website[1]. The website generated a graph, as depicted in Figure 9.1. Srivastava's research was given priority among the papers due to its alignment with the objectives of our study, albeit with a focus on Indian languages as opposed to Dagbani. By utilizing this metric, we have successfully pinpointed the papers that are most closely associated with our research topic, thereby enabling us to present a comprehensive overview of the current research landscape within our field.

Within the framework of extant literature concerning the conservation of linguistic diversity in environments with limited resources, the present study introduces an innovative methodology that distinguishes it from prior investigations. Numerous scholars have conducted

[1] https://www.connectedpapers.com/

investigations into the application of crowdsourcing and machine learning techniques. However, this particular project stands out by centering its attention on the distinctive obstacles and prerequisites associated with the Dagbani language. It aims to address the community's needs and guarantee the feasibility of its implementation.

Crowdsourcing Speech Data for Low-Resource Languages from Low-Income

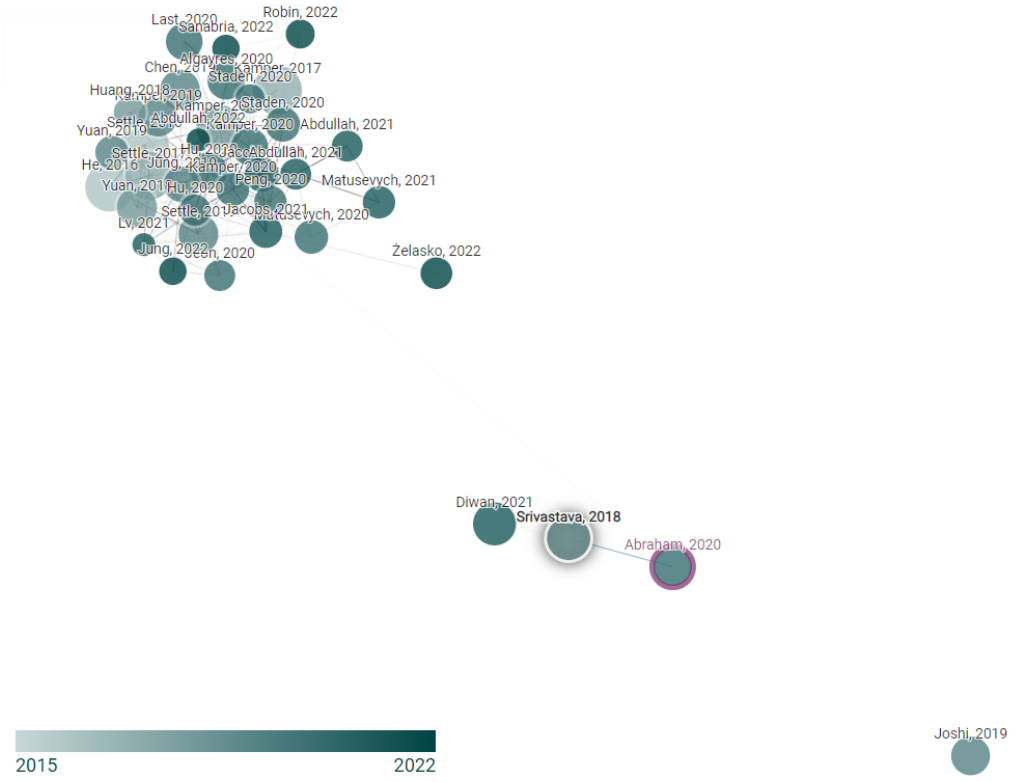


Figure 9.1: Graph of Related Papers Generated by Connected Papers

Workers:

Basil et al. present a series of user studies conducted in various Indian communities to acquire Marathi spoken data (39). The researchers created an Android application for users with limited digital literacy and limited experience. The application used hand-drawn icons to represent various actions, making navigation and task completion easier for users. Participants were recruited from rural villages, urban neighborhoods, and college campuses. The participants were provided with affordable smartphones and instructed on how to use the application to record Marathi sentences. The work was performed in a collaborative environment, with participants assisting one another, and was greeted with enthusiasm and pride for the participants' native language. The lack of reading material

in rural areas was emphasized, highlighting the significance of this initiative for preserving and promoting the Marathi language and culture.

While the previous initiative in India focused on the Marathi language, the Tiballi project targets the Dagbani language specifically. This emphasizes the distinctive cultural and linguistic context in which your work exists. The development of specific applications is another distinction. In the case of the project, it pertains to a mobile app and web app customized for "Resourcing Small Indigenous Languages in the Field," reflecting the specific requirements and difficulties of data collection in remote or field settings. In addition, this initiative places special emphasis on voice-to-text capabilities, indicating a particular emphasis on speech recognition and transcription for the Dagbani language.

Both projects involve the creation of mobile and web applications to capture spoken data for particular languages. Both organizations focus on preserving and promoting indigenous languages through crowdsourcing and recognize the importance of leveraging user-generated content to create knowledge bases that can be used in machine learning programs. Moreover, the ultimate objective of both initiatives is to use the collected data to develop advanced language technologies, such as voice-to-text systems.

A system for high quality crowdsourced indigenous language transcription: Munyaradzi et al. examine the Berkeley Open System for Skill Aggregation (Bossa), an open-source software framework that facilitates distributed thinking via crowdsourcing(40). Bossa enables project administrators to construct online tasks requiring human intelligence that volunteers can complete. Bossa operates on a volunteer basis without financial incentives, unlike platforms like Amazon Mechanical Turk. Providing tools and an administrative interface, it facilitates the creation of distributed-thinking initiatives. The framework supports job distribution and replication policies to assure volunteer accuracy and consensus. The text also discusses the transcription tool built on Bossa for transcribing texts in indigenous languages, emphasizing the login procedure, character encoding, transcription task, and inter-transcriber agreement evaluation of transcription accuracy.

While Bossa is a framework that facilitates distributed thinking initiatives across multiple domains, the Tiballi project is designed specifically for resourcing small indigenous languages in the field, with Dagbani as its primary focus. Your project may offer various motivations or incentives for users to contribute recordings, in contrast to Bossa, which operates on a volunteer basis without financial incentives. This initiative also includes the development of mobile and web applications, making it more accessible and user-friendly for those interested in contributing to the Dagbani language. In addition, the ultimate

aim of the Tiballi project is to enable voice-to-text capabilities, which distinguishes it from Bossa's broader scope of facilitating diverse crowdsourcing tasks.

In addition, this initiative, which focuses on resourcing small indigenous languages in the field, resembles the Berkeley Open System for Skill Aggregation (Bossa) framework. Both initiatives intend to collect and utilize human intelligence for language-related tasks through the use of crowdsourcing. Your application, like Bossa, allows users to contribute their voices and recordings to a knowledge base. This database of Dagbani user's recordings can be utilized to train a machine learning model for voice-to-text applications. Both initiatives emphasize the significance of coordinating the efforts of volunteers to resolve language-related issues and advance linguistic diversity.

Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages:

Srivastava et al. explain the ASR challenge for Indian languages with limited resources(41). The objective of the challenge was to advance Tamil, Telugu, and Gujarati speech recognition technology by providing participants with training and test datasets, baseline systems, and evaluation tools. Participants were permitted to utilize the supplied data to construct their ASR systems and submit their hypotheses for evaluation. In addition, the text emphasizes the participating teams and their methodologies, such as multilingual transfer learning, end-to-end models, and data sharing. The results demonstrated an advance over the baseline systems, with Time-Delay Neural Networks (TDNN)-based models exhibiting strong performance. The disclosed data will contribute to future research on speech recognition for Indian languages with limited resources.

While the Low Resource ASR Challenge provided existing datasets and focused on evaluating various ASR models, the Tiballi project entails the creation of a crowdsourcing mobile and web application. This application enables users to record themselves speaking Dagbani, thereby contributing new data for the purpose of constructing a knowledge base. This initiative emphasizes the grassroots effort of resourcing small indigenous languages, tailoring its efforts to Dagbani's particular requirements and characteristics. In addition, your ultimate objective of voice-to-text conversion demonstrates your intent to develop applications for common language use.

Both the Low Resource ASR Challenge for Indian Languages and the Tiballi Project seek to advance speech recognition technology for languages with limited resources. Collecting speech data from native speakers to create knowledge bases for training machine learning models is a component of both approaches. While the challenge targeted Tamil, Telugu, and Gujarati, this initiative focuses on the Dagbani language in particular. Both initiatives

acknowledge the significance of leveraging crowd contributions and the potential of machine learning to close the language resources gap.

A Review of Speech Recognition in Low-resource Languages:
Meng et al. provide an overview of speech recognition techniques, concentrating on both conventional methods and an end-to-end strategy(42). Traditional methods are comprised of four major components: feature extraction, an acoustic model, a language model, and decoding. The original audio signal is transformed into a series of feature vectors during feature extraction. Using hidden Markov models, the acoustic model determines the likelihood of a speech signal given a possible text sequence. Language models estimate the probability of word sequences, frequently utilizing N-grams or recurrent neural networks. Decoding entails using weighted finite-state transducers to seek for the most probable text sequence. The end-to-end approach investigates methods including Connectionist Temporal Classification (CTC) and Recurrent Neural Network Transducer (RNN-T) that directly map input audio to output text, thereby addressing the challenge of aligning input and output sequences.

Nevertheless, the Tiballi initiative diverges from conventional speech recognition techniques in a number of significant ways. Rather than relying solely on pre-existing audio corpora, you adopt a crowdsourcing strategy, enabling users to contribute their own recordings via a mobile and web application. This method permits a greater number of speakers to participate actively, thereby enhancing the Dagbani language's possible representation and coverage. Secondly, this project focuses specifically on the Dagbani language, emphasizing the need to address the unique challenges and nuances of a specific indigenous language, and combining the power of technology and community involvement to create a specialized voice-to-text system for Dagbani, highlighting the significance of preserving and supporting indigenous languages in practical applications.

Both initiatives collect audio data (recordings of Dagbani words) in order to construct a knowledge base for training a machine learning model, similar to the traditional speech recognition techniques discussed previously. The objective is to construct a voice-to-text system using this dataset, similar to the traditional approach. Moreover, both initiatives align with the objective of utilizing technology to preserve and support small indigenous languages, which is similar to the overarching objective of preserving linguistic diversity and providing resources for its maintenance.

"Small" language limited-vocabulary automatic speech recognition using Machine Learning:

Vlad's master's thesis discusses the rationale behind the development of a speech recognition system and the collection of language vocal data(43). It emphasizes concerns regarding the energy consumption, carbon footprint, and potential biases of large data models. The goal is to develop a system capable of recognizing words such as "yes" and "no" across diverse voice patterns and languages, with a particular focus on marginalized communities. The research question emphasizes on the viability of developing a low-resource application for data collection and identifying the minimum number of audio data points required for 90% accuracy. The research methodology includes a literature review, data acquisition, data processing, and the development of a machine learning model. In addition, the document describes the technology, characteristics, data collection process, data analysis, data processing stages, and the application of data augmentation techniques. Overall, the objective is to develop a speech recognition system that is inclusive, efficient, and compatible with platforms that support voice-based information services.

The specific domain of focus distinguishes this initiative from Vlad's voice-to-text project in a significant way. This initiative highlights the significance of resourcing small indigenous languages, with Dagbani as the language of primary interest. In contrast, Vlad's voice-to-text initiative may capture and transcribe speech in multiple languages or a specific mainstream language. In addition, whereas this project focuses primarily on data collection and constructing a knowledge base for the Dagbani language, the alternative project will likely entail a more extensive pipeline that includes speech recognition, natural language processing, and text generation. The ultimate goal of the voice-to-text initiative may be to provide accurate and real-time transcription services for a variety of applications, including transcription software and voice assistants.

This project on developing a crowdsourcing application for the Dagbani language and Vlad's project on voice-to-text technology share the objective of utilizing user-submitted data to enhance language-related applications. This paper concentrates on small indigenous languages, specifically Dagbani, whereas the other project seeks to develop voice-to-text capabilities. In both instances, machine learning algorithms play an essential role in training models to recognize and comprehend speech patterns. The development of mobile and web applications to facilitate data collection and interaction with the respective language datasets is included in both initiatives.

After conducting an extensive review of the available literature and examining the complexities of relevant research, our attention now turns towards the potential opportunities and future undertakings. As we proceed to the chapter titled "Prospects and Future Work",

a captivating narrative emerges, depicting the convergence of technological advancements with the profound elements of language and culture.

10

Prospects and Future Work

This chapter comprises three subchapters that centre on distinct facets pertaining to the future innovation of this project. The initial subchapter 10.1 delves into the discourse surrounding the advancement and instruction of machine learning models with the aim of achieving precise speech recognition in the Dagbani language. In the subsequent subchapter, denoted as 10.2, an in-depth exploration is conducted on the intricacies of language translation and voice synthesis. This examination places particular emphasis on the principles of user-centered design and the importance of adaptability within these processes. In conclusion, subchapter 10.2.1 delves into the difficulties encountered in low-resource environments and the corresponding approaches to surmount these challenges while upholding user requirements as a paramount concern.

10.1 Development and Training of Machine Learning Models

10.1.1 Improving Speech Recognition

The subsequent stage of the project entails enhancing the capabilities of the machine learning models in order to attain precise speech recognition in the Dagbani language. The subsequent actions to be taken are delineated based on the knowledge acquired from an interview conducted with André Baart, an expert in the respective field.

Baart proposes the application of Mel spectrograms as means to capture the distinctive attributes of Dagbani audio. The utilisation of Python libraries enables the generation of Mel spectrograms, which offer a more detailed depiction of frequency levels across temporal intervals. The spectrograms, characterised by their fluctuations in intensity, are utilised as valuable training data for the model.

One possible approach for training the speech recognition model involves the utilisation of transfer learning. Through the utilisation of pretrained models originally intended for diverse tasks such as object recognition, it becomes feasible to adapt them for the purpose of Dagbani speech recognition. This methodology mitigates the necessity of training the complete model anew and enables the customization of the model for the specific purpose of Dagbani speech recognition.

Baart proposes commencing the experimentation process by utilising a reduced dataset, specifically the existing 20 classes, and subsequently evaluating the accuracy of the model. The introduction of data augmentation techniques, such as the addition of silence and noise, can enhance recognition accuracy as the sample size is expanded. The utilisation of pretrained models in conjunction with data augmentation presents a promising approach for attaining resilient and precise speech recognition in contexts with limited resources.

10.1.2 The Process of Language Translation and Voice Synthesis

An additional pivotal component pertaining to forthcoming endeavours encompasses the domains of language translation and voice synthesis specifically in the Dagbani language. The identified use cases present a comprehensive plan for enhancing the functionalities of the system.

The primary focus in the initial use case is the development of an extensive language corpus and the training of artificial intelligence models for the purpose of language translation. The foundation for training a Neural Machine Translation (NMT) model is established by utilising a parallel corpus that comprises English sentences alongside their corresponding translations in Dagbani. The enhancement of the system's accuracy in providing translations can be achieved by capturing the linguistic nuances and context-specific translations that are necessary for weather-related queries.

Voice synthesis in Dagbani is an essential component that contributes significantly to the production of authentic and seamless audio responses for users, alongside the process of translation. Through the creation of a text-to-speech (TTS) synthesis model, it becomes possible to convert the translated responses into audio of superior quality in the Dagbani language. This step is implemented to guarantee that users are provided with audio responses that are linguistically precise and culturally suitable.

10.1.3 Enhancing and Incorporating Continuous Model Improvement

In subsequent endeavours, it is imperative to underscore the iterative and ongoing enhancement of the machine learning models. Through the integration of user feedback and the implementation of rigorous testing methodologies, it is possible to refine the models and improve their performance progressively. The augmentation of data, encompassing a wide range of dialects, accents, and speech variations within the Dagbani language, will contribute to the enhancement of the models' resilience and precision.

Integration is a crucial element that will play a significant role in future endeavours. The cohesive user experience is ensured by seamlessly integrating the trained language translation and voice synthesis models into the overall system. The integration of models facilitates the provision of weather information in Dagbani audio format, thereby enhancing accessibility and inclusivity for English-speaking users.

10.2 Considerations for Limited Resources and Design Focused on User Needs

10.2.1 Mitigating Challenges in Low-Resource Environments

It is imperative to take into account the difficulties and limitations associated with low-resource environments as the project progresses. In accordance with insights derived from an interview conducted with André Baart, a notable emphasis is placed upon the significance of comprehending the contextual factors and requirements of the intended user demographic. The utilisation of collaborative methodologies that entail active engagement with the local community and the formulation of pertinent inquiries facilitate the development of solutions that possess genuine relevance and feasibility.

Baart emphasises the importance of developing solutions that necessitate minimal or zero supplementary resources for end-users. This approach guarantees the accessibility and sustainability of the technology in low-resource settings, where challenges such as limited internet connectivity and extreme environmental conditions may arise. The ability to adjust to these circumstances and develop inventive strategies to surmount challenges will constitute a crucial element of future endeavours.

10.2.2 The Significance of User-Centered Design and Adaptability in Academic Contexts

The principle of user-centered design is expected to persist as a guiding framework in future endeavours. The active engagement of the Dagbani-speaking community in the design and development process, soliciting their input and feedback, is of utmost importance. By incorporating the viewpoints of individuals, the system can be tailored to conform to their requirements, inclinations, and cultural milieu.

In addition, the project will require adaptability in order to effectively navigate the intricacies of low-resource environments. The forthcoming research will prioritise the advancement of adaptable and resilient systems capable of enduring obstacles such as intermittent internet connectivity and severe weather conditions. By taking into account these factors during the design and implementation stages, it is possible to develop solutions that are more suitable for the specific context of the Dagbani-speaking community.

In summary, the forthcoming endeavours entail further enhancing the machine learning components of the undertaking, specifically focusing on the domains of speech recognition, language translation, and voice synthesis. The proposed methodology for model training, integration of limited scenarios, and consideration of resource-constrained settings offers a strategic framework for attaining precise and culturally significant results. The project endeavours to empower the Dagbani-speaking community and safeguard their linguistic heritage by placing emphasis on user-centered design principles and adaptability.

As the chapter on "Prospects and Future Work" draws to a close, we find ourselves on the cusp of potential, driven by the aspiration to safeguard linguistic heritage and adopt advancements in technology. As we direct our attention towards the future trajectory of the project, we commence the concluding phase of our endeavour, namely the "Collaboration: The Key to Empowering Linguistic Diversity" chapter.

11

Collaboration: The Key to Empowering Linguistic Diversity

In the context of an increasingly interconnected global society, the preservation of indigenous languages and cultures emerges as a matter of utmost significance. This is particularly true for a multitude of endangered languages, such as the Dagbani language, which is spoken in the northern region of Ghana. The potential for species extinction is heightened as a result of insufficient educational resources and declining levels of literacy. The research project responded to this urgent challenge by embarking on the endeavour of digitising indigenous languages, presenting a hopeful pathway for their conservation and rejuvenation.

The main aim of the study was to create a novel data collection methodology designed specifically for the Dagbani language. This methodology is an essential part of the larger research project called TIBaLLi. This initiative, operating within the framework of the Internet Society, sought to enhance the agency of the Dagbani community, preserve their linguistic heritage, and address the digital divide by utilising sophisticated artificial intelligence techniques such as machine learning and natural language processing. The overarching goal was to promote inclusivity on the Internet for communities residing in resource-constrained settings.

The study was based on the objective of utilising technology to protect and preserve linguistic diversity in environments with limited resources. By utilising a participatory co-design methodology in conjunction with individuals who are fluent in the Dagbani language, this study established a conducive setting that enabled the successful integration of stakeholders' and users' requirements and preferences. By adopting an iterative and

feedback-driven approach, the project underwent several rounds of testing and refinement, ensuring that the solutions developed were in line with the community's requirements. Throughout the course of this study, the research consistently demonstrated a strong commitment to comprehending and effectively dealing with the limitations, challenges, and technical requirements associated with the conservation of indigenous languages. This unwavering dedication further underscored the research's focus on safeguarding and rejuvenating the linguistic heritage of the Dagbani community.

The research heavily relied on an iterative approach, which was influenced by the participatory field experimentations conducted in collaboration with rural communities in northern Ghana as part of the TIBaLLi project. This approach facilitated ongoing improvement through the incorporation of community feedback and requirements, as evidenced by the effective deployment of the Dagbani data collection platform. Consequently, this initiative has led to the systematic acquisition of a valuable corpus in indigenous languages. By leveraging insights acquired through collaboration with the Dagbani community and experts, the research endeavour expanded the comprehension of the cultural and technical aspects entailed in the preservation of language. The acquisition of this valuable knowledge has greatly contributed to the successful resolution of diverse constraints and obstacles, effectively addressing the technical prerequisites for the preservation of indigenous languages.

The investigation into crucial factors for developing a language preservation system that aligned with the distinct requirements and principles of the local community resulted in the creation of a durable and culturally aware platform that successfully catered to the Dagbani community. In addition, a thorough analysis of the limitations, challenges, and technical specifications associated with the preservation of indigenous languages resulted in the adoption of an audio refinement procedure, which guarantees the superior quality of the gathered data. As a result, this approach enhances the precision and effectiveness of the machine learning algorithm.

In summary, this research endeavour highlights the importance of employing collaborative methodologies to protect and preserve linguistic diversity within resource-constrained settings. This statement highlights the importance of empowering indigenous communities by involving them actively in the preservation process. This approach helps to develop a deep understanding of the cultural and technical factors that impact language preservation. The acquired insights serve as a valuable resource for future endeavours, making a

significant contribution to the broader objective of safeguarding and commemorating linguistic heritage on a global scale. The success of the project can be attributed to several key factors, namely the iterative approach, stakeholder involvement, and the knowledge acquired from the existing TIBaLLi research. These factors have played a crucial role in contributing significantly to the preservation and revitalization of the Dagbani language. Furthermore, the project has set a precedent for similar initiatives on a global level.

References

. **Text-independent speaker recognition using LSTM-RNN and speech enhancement**. *Multimedia Tools and Applications*, **79**, 09 2020. xi, 105

. **User Centred System Design-New Perspectives on Human/Computer Interaction**. *J Educ Comput Res*, **3**, 01 1987. 31

Manifesto for agile software development. 2001. 32

. **The Importance of Hosting a Mobile Application on a Server**. *Journal of Mobile Application Development*, **1**(1):33–40, 2019. 54

. **Hosting Mobile Applications: An Overview of the Options**. *Mobile Application Development Journal*, **2**(2):45–53, 2020. 54

. **Firebase for Mobile Application Hosting: A Review**. *Journal of Mobile Application Development*, **2**(3):65–74, 2020. 54

. **Heuristic evaluation of user interfaces**. In *International Conference on Human Factors in Computing Systems*, 1990. 78

. **Human Integration Design Processes (HIDP)**. 2014. 78

. **Automatic recognition of spoken digits**. *The Journal of the Acoustical Society of America*, **24**(6):637–642, 1952. 96

. **The dragon system—An overview**. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **23**(1):24–29, 1975. 96

. *Connectionist speech recognition: A hybrid approach*. Kluwer Academic Publishers, 1994. 96

. **Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups**. *IEEE Signal Processing Magazine*, **29**(6):82–97, 2012. 97

. **Transfer learning from speaker verification to multispeaker text-to-speech synthesis**. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6965–6969, 2019. 97

. **Convolutional neural networks for speech recognition**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **22**(10):1533–1545, 2014. 98

. **Deep speech: Scaling up end-to-end speech recognition**. *arXiv preprint arXiv:1412.5567*, 2014. 98

. **Improving Speech Intelligibility in Noise Using Environment-Optimized Algorithms**. *Audio, Speech, and Language Processing, IEEE Transactions on*, **18**:2080 – 2090, 12 2010. 99

. **Enhancement of noisy speech by temporal and spectral processing**. *Speech communication*, **53**:154–174, 02 2011. 99

. **Wiener Filter and Deep Neural Networks: A Well-Balanced Pair for Speech Enhancement**. *Applied Sciences*, **12**(18):9000, 2022. 99

. **A time delay neural network architecture for efficient modeling of long temporal contexts**. pages 3214–3218, 09 2015. 99

. **Effect of Noise Suppression Losses on Speech Distortion and ASR Performance**. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 996–1000, Singapore, Singapore, May 2022. IEEE. 99

. **Deep learning for environmentally robust speech recognition: An overview of recent developments**. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **9**(5):1–28, 2018. 99

. **Should we always separate?: Switching between enhanced and observed signals for overlapping speech recognition**. *arXiv preprint arXiv:2106.00949*, 2021. 99

. **A comprehensive survey on generative adversarial networks used for synthesizing multimedia content**. *Multimedia Tools and Applications*, pages 1–40, 2023. 99

. **Adversarial audio super-resolution with unsupervised feature losses**. 2018. 99

. **Supervised and unsupervised speech enhancement using nonnegative matrix factorization**. *IEEE Transactions on Audio, Speech, and Language Processing*, **21**(10):2140–2151, 2013. 99

. **Speech recognition in noisy environments: An overview of the problem and the solutions**. *Applied Sciences*, **10**(16):5585, 2020. 101

. **Robust speech recognition in noisy environments: the 2001 IBM SPINEevaluation system**. **1**, pages I–53, 02 2002. 101

. **The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions**. In *ASR2000-Automatic speech recognition: challenges for the new Millenium ISCA tutorial and research workshop (ITRW)*, 2000. 101

. **Improving Language Identification of Accented Speech**. *arXiv preprint arXiv:2203.16972*, 2022. 103

. **Lexical modeling of non-native speech for automatic speech recognition**. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, **3**, pages 1683–1686 vol.3, 2000. 103

. **Language Variation and Algorithmic Bias: Understanding Algorithmic Bias in British English Automatic Speech Recognition**. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 521–534, New York, NY, USA, 2022. Association for Computing Machinery. 103

. **Linguistic practices in Cyprus and the emergence of Cypriot Standard Greek**. *Mediterranean Language Review*, **17**:15–45, 2010. 103

. **Whispered Speech Recognition Using Deep Denoising Autoencoder and Inverse Filtering**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **25**:2313–2322, 12 2017. 104

. **Impact of vocal effort variability on automatic speech recognition**. *Speech Communication*, **54**(6):732–742, 2012. 104

. **Image method for efficiently simulating small-room acoustics**. *The Journal of the Acoustical Society of America*, **65**:943–950, 04 1979. 104

. *Distant speech recognition*. John Wiley & Sons, 2009. 105

. *Handbook of Standards and Resources for Spoken Language Systems: Spoken language characterisation*. Walter de Gruyter, 1997. Google-Books-ID: 8cxtWcsAk5MC. 106

. **Speaker independent phonetic transcription of fluent speech for large vocabulary speech recognition**. pages 75–80, 01 1989. 106

. **Crowdsourcing Speech Data for Low-Resource Languages from Low-Income Workers**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2819–2826, Marseille, France, May 2020. European Language Resources Association. 152

. **A system for high quality crowdsourced indigenous language transcription**. *International Journal on Digital Libraries*, **14**:117–125, 2014. 153

. **Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages.** In *SLTU*, pages 11–14, 2018. 154

. **A Review of Speech Recognition in Low-resource Languages**. In *2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 245–252. IEEE, 2022. 155

. **"Small" language limited-vocabulary automatic speech recognition using Machine Learning**. 156

Appendix

11.1 Documentation

Table of Contents:

Project Overview

The crowdsourcing app permits users to record themselves pronouncing Dagbani words. These recordings will serve as valuable training data for a machine learning model that will accurately convert Dagbani speech to text.

There are two components to this project: a mobile app and a web app.

Mobile app: The mobile application is developed with React Native and Expo. It offers an intuitive interface for capturing and submitting audio samples of Dagbani words. Utilizing the capabilities of mobile devices, the application ensures high-quality recordings.

Web app: The web app, created using React and Tailwind CSS, complements the mobile app by offering a web-based platform for users to access and contribute to the crowdsourcing effort. Users can record their voices and submit Dagbani word recordings directly from their web browsers.

Installation and Setup

Before you proceed with the installation and setup, make sure you have the following software and tools installed on your system:

- **Node.js:** Download and install Node.js (version 14.0.0 or higher)
- **NPM:** NPM is typically installed along with Node.js, but make sure you have a compatible version (version 6.0.0 or higher)
- **Git:** Download and install Git

Mobile App Installation:

1. Clone the GitHub repository by running the following command in your terminal or command prompt:
   git clone https://github.com/AntriaPan/TIBaLLi-project-voice-services.git
2. Navigate to the project directory:
   cd mobile-app
3. Install project dependencies by running the following command:
   npm install
4. Start the development server:
   expo start

Web App Installation:

1. Clone the GitHub repository by running the following command in your terminal or command prompt:
   git clone https://github.com/AntriaPan/TIBaLLi-project-voice-services.git
2. Navigate to the project directory:
   cd web-app
3. Install project dependencies by running the following command:
   npm install
4. Start the development server:
   npm start
5. Open your web browser and enter the following URL:
   http://localhost:3000
   This will load the web app in your browser, and you can now explore its features and functionalities.

Real Time changes through users' device: Connect your physical device or start an emulator to run the app:

- For Android: Connect your Android device via USB and make sure USB debugging is enabled in the device settings. Alternatively, start an Android emulator.
- For iOS: Start the iOS simulator.

Run the app: For Android, run the following command: npx react-native run-android For iOS, run the following command: *npx react-native run-ios*

These commands will build the app and launch it on your connected device or emulator.

Updating Data on the Hosted Website

Prerequisites

Before updating data on the hosted website, make sure you have the following:

1. Firebase project credentials: These are obtained from the Firebase console when you create a Firebase project. You'll need the credentials to authenticate and access your Firebase project programmatically.

2. Access to the Firebase Firestore database: Firestore is a NoSQL cloud database provided by Firebase. Ensure you have access to the Firestore database associated with your Firebase project.

Steps to Update Data

1. Log in to the Firebase console (https://console.firebase.google.com/) and navigate to your project.
2. In the Firebase console, locate and select the Firestore database.
3. Browse the collections and documents in the database to find the data you want to update.
4. Make the necessary changes to the data within the Firestore console's interface. You can update values, add new fields, remove fields, etc.
5. Save the changes within the Firestore console.

   Upon saving the changes, the data on the hosted website will be automatically updated to reflect the modifications made in the Firestore database.

Updating Data via Terminal Commands and Deploying to Firebase Hosting Alternatively, you can update data in the Firestore database using terminal commands and then deploy the changes to Firebase Hosting. Here's how:

1. Install the Firebase command-line tools globally by running the following command:
   npm install -g firebase-tools
2. Authenticate with your Firebase account by running the following command and following the authentication prompts:
   firebase login
3. Navigate to your project directory using the terminal:
   cd web-app
4. To update data, you can use the Firebase CLI's Firestore commands. For example, to update a document in a collection, you can run the following command: firebase firestore:
   update [path_to_document] –data [updated_data]
5. Once you have updated the data, you can deploy the changes to Firebase Hosting by running the following command:
   firebase deploy –only hosting

   Upon saving the changes, the data on the hosted website will be automatically updated to reflect the modifications made in the Firestore database.

Access the files in Firebase

1. The user should open a web browser and navigate to the Firebase console website: https://console.firebase.google.com/. This website is the central hub for managing Firebase projects.

2. Upon reaching the Firebase console website, the user needs to sign in to their Firebase account using their credentials, as shown in Figure 11.1. This ensures that they have the necessary authentication to access their Firebase projects.

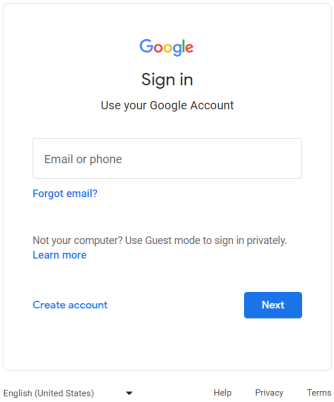


Figure 11.1: Login to the Firebase Console using the Google credentials

3. After signing in, the user will be directed to the Firebase console dashboard. This dashboard provides an overview of all the Firebase projects associated with their account. The user should locate and select the specific Firebase project that corresponds to the app they are interested in. In this case the Firebase project is called "Dagbani-Speak", as shown in Figure 11.2.

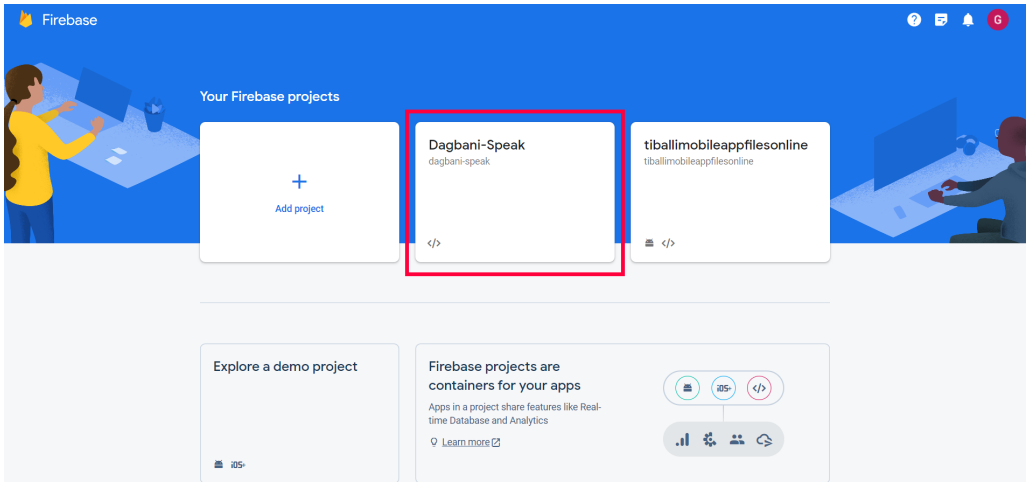


Figure 11.2: Default page of dashboard after sign in

4. Once the project is selected, the user will be taken to the project overview page. This page provides a summary of the project's settings and features, as displayed in

Figure 11.3. For example, the user can easily see the number of users that download the mobile app and how much storage is filled with user recordings' data.

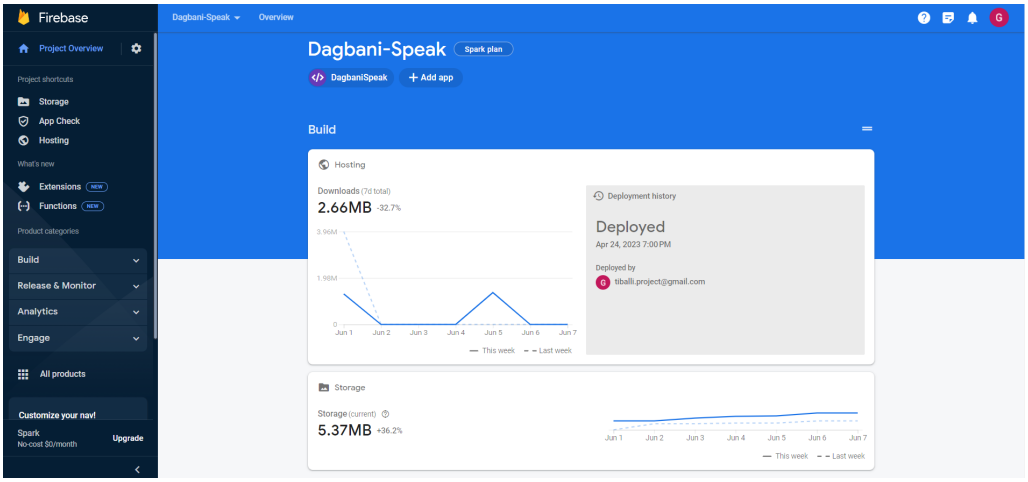


Figure 11.3: Overview page of "Dagbani Speak" project

5. To access the storage section, the user should look for the option labeled "Storage" in the menu on the left-hand side of the screen. This menu contains various options for managing different Firebase services, as shown in Figure 11.4.

Clicking on the "Storage" option will redirect the user to the Firebase Storage section. Firebase Storage is a powerful cloud storage service that allows users to store and retrieve files, such as images, videos, or documents, within their Firebase projects.

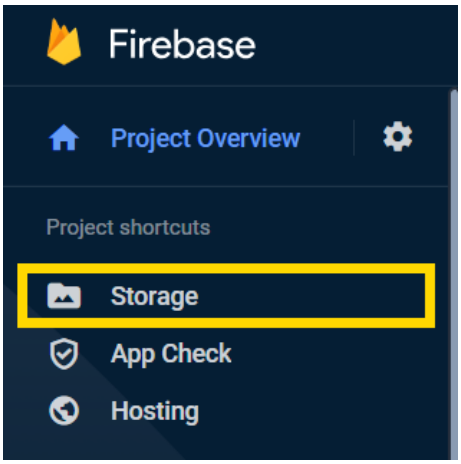


Figure 11.4: Selecting the storage option from the central menu

5. Once a storage bucket is selected, the user will be presented with a list of folders and files contained within that bucket. Folders help organize files and provide a hierarchical structure for better file management. The user can navigate through the folders by clicking on them to open and view their contents, as shown in Figures 11.5 - 11.8. The main folder that contains all the information is called "Recordings" as shown in Figure 11.5.

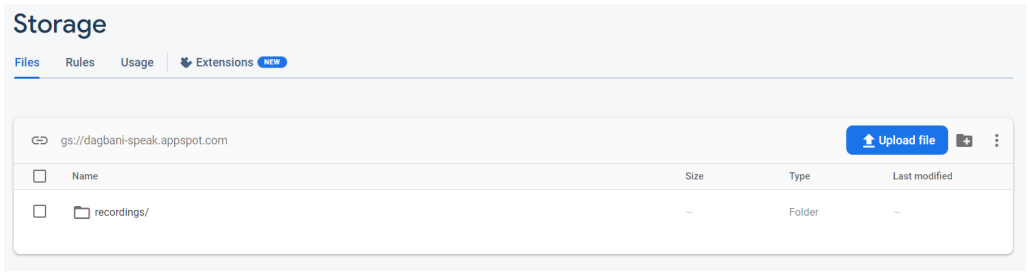


Figure 11.5: First step of the hierarchical structure

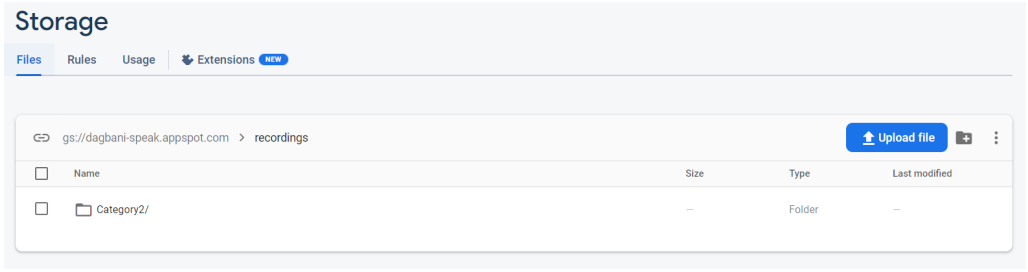


Figure 11.6: Second step of the hierarchical structure

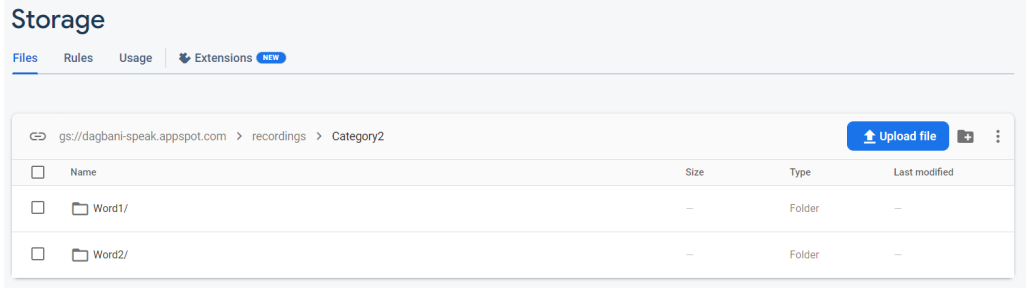


Figure 11.7: Third step of the hierarchical structure

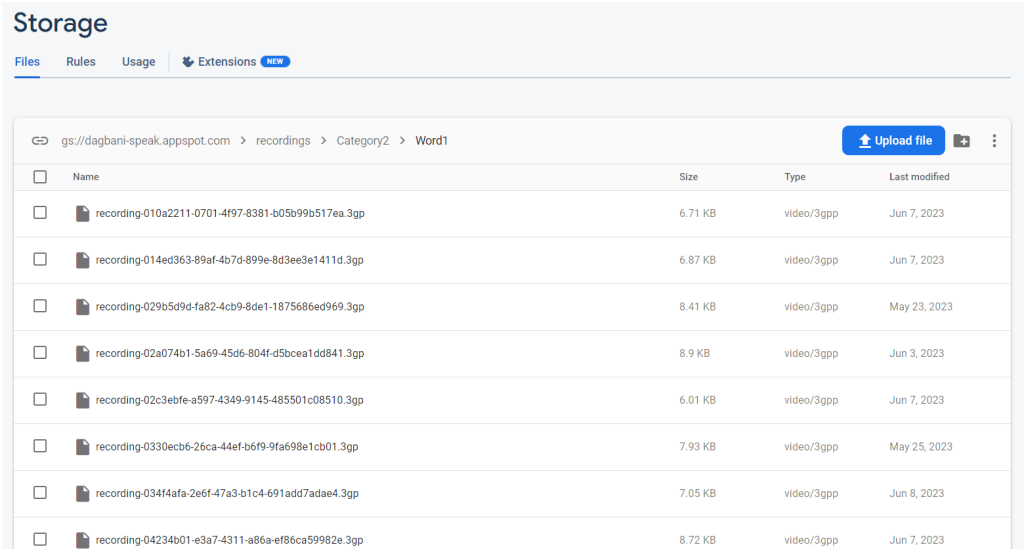


Figure 11.8: Last step of the hierarchical structure

6. To view the details or download a specific file, the user can click on the file name or select the file from the list. This action will provide access to additional options and information related to the file, such as its size, file type, and download link. An example of this can be found in Figure 11.9 The user can choose to download the file to their local machine or perform other actions based on their requirements.

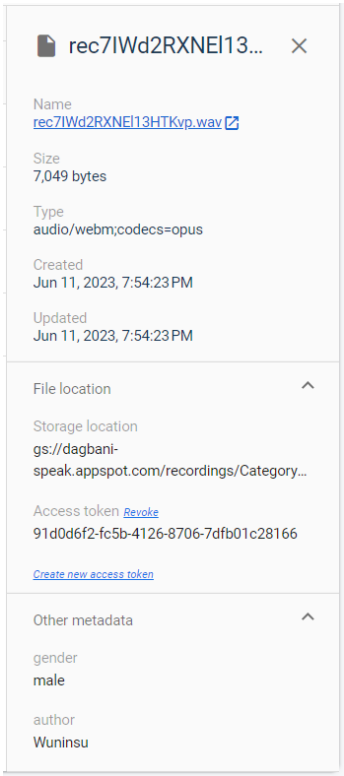


Figure 11.9: Access the metadata details of the file

It is important to note that the accessibility of files and actions within Firebase Storage is determined by the project's security rules. These rules define who can access the files and what operations they can perform. Therefore, the user should

ensure that they have the necessary permissions and access rights to view or interact with the files in the storage bucket.

License

This project is licensed under the MIT License.

11.2 Interviews transcription

11.2.1 André Baart: describing the procedure that machine learning's upcoming steps should take into consideration

So what we did in Vlad's paper, which I know somehow works, is that you have the audio files and you also create graphs from them, which you also create, but what I created may be the same thing with MFCC. We did create a Mel spectrogram, which is a way of converting frequency in hertz to something with more resemblance to how high we perceive this frequency to be for human hearing. So what we want is to look at them, something that you can do also with Mel's spectrogram with some Python library that can do it for you, and you are going to get something like the MFCC with a bit more granularity, and then the redder it is, the more intensity there is at a certain frequency level at a certain point in time. And I think for training, the model is not perceived to do a lot of cleaning. That's good that you cleaned the beginning and ending of the audio files; that's okay, and you can start with those samples; they are probably enough to create a model. What we usually do with this kind of model is increase the sample size, try to make it more robust, and try to add some noise. It's called data augmentation and adds silence and noise, but it's for a later stage. But what you can do with those Mel spectrograms is find models that are pretrained models and can, for example, disguise themselves and say this is a cat or this is a dog. And you can take one of those models, or you can take the pretrained model, in which the large part of the interpretation is already there, and you retrain it or fine-tune it for your classes, and you say, "This is the class for yes, and this is the class for no. And you should get an idea like that, and you should not train the whole model from scratch; you just cut off a large part of it and use your classes. This is what we used with Vlad and some other projects, and somehow it worked, but we didn't test it that much yet. So it is part of the experiment also, and just to summarise: Mel's spectrogram, data augmentation, and getting a pretrained model So you start with the 20 classes that you have, which will be the essentials to get something, and at the end you will also get a percentage of error on the accuracy. So you have 20 examples, let's say 5 for the test set and the rest for training, and will give the amount of errors and should go down this number. If it goes down to a good number, then you know that is probably going to work,

and then you do data augmentations. But you probably need more than 20 in order to get better results and feel confident. But you are going to see how this will happen, test yourself, and see if it's okay or not. But in any case, this needs a bit of experimentation. So I will really Google some Github examples, and there is also a lot of code. You probably need to do a transfer learning, cut off the last part, and add your own, and you can use Google Colab or something like that, which has a free GPU. Or you can try it on your own laptop, but it will take a while. So I hope that this gives you an idea of the whole process, and if you get stuck or something, you can send me an email.

11.2.2 André Baart: describing the procedure that machine learning's upcoming steps should take into consideration

What are some of the most important parts that you think of when you create a system with low resources?

Always considering the low resource environment, of course. There are so many examples on projects, including those envolved by myself, that don't consider this, and then there are the dangers that sound really great here at the office, and probably you can sell really well to people in other offices to make some money, but in reality you cannot implement that. Our methodology usually starts really small, and all businesses say that they do a lot of thinking without actually doing whatever they say. First, go to whoever you are designing something for and talk to them a lot and ask questions. Ask a lot of questions like "who are you, what do you do for a living, etc." in order to explore a little bit and better understand the context. For example, if you ask a person in Ghana exactly, "What do you want from your app?" the answer will most of the time not be that useful since they do not have the scope, and also here in the Netherlands, the same will happen, or you are going to get a full book of requirements that are never going to work, and you have to do this collaboratively. Think some kind of problems, then think some kind of solutions, then make some prototypes, and that way you will find your way, and that is what we usually also do in high resources but also in low resources. The difference will be that maybe you will have more challenges, like the internet not working all the time, it being 45 degrees, and your laptop breaking for some reason. You will not be able to fix it, but you have to figure it out since that is the reality there. A major mistake is when you build something that requires people to buy something new. For example, if they need a smartphone and they cannot afford it, we can buy it for them, but what will happen is that it will break or they will not use it because it is not close to their way of working. So we always try to make stuff that requires minor or, ideally, zero new stuff for people. and, of course, adaptability.

We are thinking of building an application in order to address the scope of

the project. What are your ideas about that?
I am not sure, since I don't know exactly the market in order to draw a conclusion on this. The biggest question is regarding who is going to be the target audience of this, and if there are people that use smartphones, then maybe we can draw a conclusion with this case. In my case, I would like to think a little bit more about ethics, market dynamics, and those kinds of things.

What are some functional or non-functional requirements for that kind of system?
From a technical perspective, it's probably not extremely difficult; you need an app, a website, or maybe both. You probably need some way to update the data that is there, probably some backend or access from your computer.

11.2.3 Francis Saa-Dittoh: describing the project overview and needs

So my research has been on building digital information systems for low-resource environments. One of the angles we explore is the idea of utilizing more advanced technologies. In our previous research, we typically relied on old technologies or something more adaptable to the current situation, such as radios or phones. However, in this case, we are interested in exploring innovations like AI, machine learning, and natural language processing and how they can be used in such environments. That's the general idea.

The focus of our research is to use natural language processing to build a corpus for local languages that currently lack resources. For example, in a country like Ghana, which is one of the 54 countries in Africa, there are over 350 languages. Many of these languages have limited or no resources available online or in written form. Our goal is to leverage the widespread use of mobile phones in Ghana, where there is about 139% mobile phone penetration, to reach communities that speak these languages.

We have already built solutions that work with local languages, but in most cases, we had to manually record the languages and gather information using DTMF (Dual-tone Multi-Frequency) input. For instance, when users called the system, it would answer in their language and prompt them to press one for "yes" or two for "no." However, we aim to automate this process so that users can respond using voice inputs in their own languages. The challenge is that we lack speech recognition models for these languages. Therefore, we need to find the easiest way to address this issue, considering the large number of languages.

To tackle this, our approach involves the following steps:

1. Crowdsourcing recordings of simple phrases like "yes" and "no" in specific languages.

2. Using machine learning techniques to train the system to recognize these phrases in

the respective languages.

3. Testing the trained system in the field and evaluating its performance.

In addition, we will expand this research to focus on climate data, as climate change is a significant issue, particularly for farmers who are the primary target of our work in rural areas.

To accomplish these goals, we need to:

1. Develop applications that enable crowdsourcing of recordings in various languages. This can involve engaging literate urban users through online chat rooms and dispatching personnel to local communities to collect data.

2. Utilize machine learning and AI techniques for training the system. We may explore different methods, including converting voice fragments into waveforms and leveraging AI models suggested by researchers in the field.

3. Integrate all the components into a comprehensive solution and deploy it in selected communities for real-world testing and evaluation. This involves linking audio recordings with the corresponding text, especially for the language we are focusing on, which is a language spoken in the northern part of Ghana. We will also explore collaborating with the Bible linguistics group, who have recordings in that language, as a potential source of crowd-sourced data.

We anticipate challenges along the way, such as the need to clean the data and ensure its quality. We also need to address the limited internet access in rural areas, where an offline mobile app would be useful for data collection. This app should allow easy data upload once an internet connection is available. Additionally, we may consider utilizing voice XML and React Native for developing voice applications that can work over GSM networks.

In summary, our research aims to leverage advanced technologies like AI and machine learning to build digital information systems for low-resource environments. By focusing on local languages and addressing challenges specific to these environments, we hope to empower communities and contribute to addressing critical issues like climate change.