UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Improving audio training for Cochlear Implant users

Christer Hagenes Opdahl

INF-3990: Master's thesis in Computer Science - May 2022

Preface

A close relative got approval for a Cochlear Implant (CI) on the right ear in 2019. This individual had suffered from hearing loss, which increased in deficiency over several years. As a result, they were post-lingually (after language acquisition) deafened in the right ear, retaining some functionality in the left ear. Unfortunately, the left ear is also degrading over time.

Hearing deficiency is an invisible disability meaning the naked eye cannot see it. Invisible disabilities often come with a lack of understanding from other people, made apparent by people’s interactions. Getting the implant and regaining hearing can have a remarkable effect on the quality of life. However, the problem morphs into what you hear instead of not being able to hear. My relative reported that it was an incredible experience turning on the sound for the first time, hearing sounds they had not heard for several years. They could hear different sounding footsteps depending on what shoes people were wearing. Running water from sinks and other obscure sounds they had not heard for years. They reported to be somewhat overwhelmed with the experience, but the excitement was certainly more significant than the dread.

As the queue for getting an appointment with an audio therapist was long, they started doing simple audio training exercises at home. The training was mainly conversations with a partner stating different phrases, the CI-individual responding with either a separate statement or the exact phrase. Sometimes they would seek out noisy environments and reported the importance of having someone with normal hearing that could “translate” the sounds they were hearing. Once they had made the connection between a specific sound and its meaning, that sound became more tangible to interpret in the future without external assistance.

During training with software-based tools, my relative reported that the lack of diversity within the available tools was the main issue. Additionally, the software-based tools seemed to be tailored toward pre-lingually deafened CI users. While performing audio training exercises, my relative felt they relied on cognitive abilities instead of actively using the implant. For example, guessing the correct answers instead of actively participating in listening. Therefore, the primary motivation for this thesis is to create a digital training platform that may assist in audio training while avoiding the reported issues. My relative has gotten a second Cochlear implant approved, which is due in late 2022. My relative will then become a bilateral CI user. Therefore, it is the perfect timing to be able to write a thesis on the subject.

By witnessing the lived experience of my relative, I have seen what getting a Cochlear implant can do for someone and how it can positively affect people's lives. CI users are a marginal group of people relative to the larger population, but they deserve the best training tools possible to maximize the benefits of their training efforts. As the Norwegian CI community is relatively small, it is challenging to recruit end-users. Additionally, it may be difficult to gather feedback from individual CI users based on issues directly linked to their hearing disabilities. Nevertheless, recruiting end-users and listening to their feedback was crucial as I had no physical experience with CI. I had no experience with degraded sound and the intricate details of how it feels to hear through a sound processor. Without consulting the people I was trying to help, I would not be able to help. Therefore, end-users were involved throughout the thesis, and my relative was utilized as a user representative.

Acknowledgments

To everyone who helped me get here. To my supervisors, Eirik Årsand and André Henriksen, for your guidance. To everyone at the HiT lab, for making this year an experience I will never forget. To Kim, a loyal friend, and study buddy. To my parents, for your endless support and encouragement. To my sister, for pushing me toward new adventures! Finally, to Camilla, my partner in crime, for keeping me sane during this period of our lives!

Thank you!

"If you have your why for life, you can get by with almost any how"

Friedrich Nietzsche, Twilight of the Idols, Epigrams and Arrows #12

Table of Contents

Preface ii

Acknowledgments iv

Table of Contents vi

List of figures xiv

List of tables xviii

List of abbreviations xx

Abstract xxiv

1 Introduction 1

1.1 Cochlear implants 1

1.2 Motivation 3

1.3 Problem definition 3

1.4 State of the art 5

1.4.1 Traditional audio training methods 5

1.4.2 Digital training tools available in Norway 5

1.5 Deployment options 9

1.5.1 Cloud computing cost analysis 9

Linode Kubernetes Engine 10

1.5.2 Single-board computers 11

Preliminary testing 12

1.6 Security 13

1.7 Thesis structure 13

2 Method 15

2.1 Literature review 15

2.2 End-user involvement 15

2.3 Preliminary questionnaire 17

2.3.1 Data gathering 17

2.3.2 Data analysis 17

2.4 Primary questionnaire 17

2.4.1 Data gathering 17

2.4.2 Data analysis 18

2.5 Requirements specification 18

2.6 Development 18

2.7 User Testing 20

2.7.1 Alpha testing with supervisors and colleagues 20

2.7.2 Beta testing with end-users 20

2.8 Semi-structured interviews 20

2.8.1 Data gathering 21

2.8.2 Data analysis 21

2.9 Performance testing 21

3 Requirements 23

3.1 Literature review findings that impacted requirements 23

3.2 Questionnaire findings that impacted requirements 23

3.3 Minimum viable product 23

3.4 Functional requirements 25

3.4.1 Sound picture requirements: 25

3.4.2 Tone separation requirements 25

3.4.3 Word separation requirements 25

3.5 Non-functional requirements 29

3.5.1 Availability 29

3.5.2 Reliability 29

3.5.3 Usability 29

3.5.4 Scalability 29
3.5.5 Responsiveness 29
4 Design 31
4.1 Back-end design choices 31
4.1.1 Investigating possible database model systems 31
4.1.2 Replication and consistency scheme 31
4.1.3 Load balancing 32
4.1.4 Scaling 33
4.2 Front-end design choices 33
4.2.1 Sound picture 34
4.2.2 Tone separation 35
4.2.3 Word separation 36
4.3 The System as a whole 37
4.4 Attached and detached design 38
5 Implementation 41
5.1 Overview 41
5.2 Back-end 41
5.2.1 Data structure – The file index 42
5.2.2 Resolving paths 43
Files 43
Folders 44
5.2.3 Load balancing 45
5.2.4 Scaling 46
5.2.5 Adding and removing files 49
5.2.6 List of Norwegian words 50
5.2.7 Rhyming algorithm 51
5.3 Front-End 52

5.3.1 Implementing a responsive web design 52

5.3.2 The Audio player and its functionalities 53

Adding audio files 54

Volume control 56

Progress bar and timings 56

Play and pause 57

Dynamically generating sinewaves 57

5.3.3 Word separation 58

5.4 Deployment 59

5.4.1 User-testing deployment 59

5.4.2 Single-board computer deployment 59

5.4.3 Linode Kubernetes deployment 60

6 Tests and results 63

6.1 Literature review 63

6.2 Preliminary questionnaire results 72

6.3 Primary questionnaire results 72

Question one: Have you undergone a CI operation? 72

Question two: How often do you perform audio training? 72

Question three: Why do some CI users not perform audio training? 73

Question four: Is audio training typically done at home? 74

Question five: How much time do you spend traveling to and from training centers or experts to perform audio training? 74

Question six: What is the audio training content when training at home on your initiative? . 75

Question seven: What is the audio training content when following an expertly made training program? 76

Question eight: Are CI users happy with the training opportunities available today? 76

Question nine: Recording singular words or short sentences are better than longer sentences or stories for audio training content? 77

Question ten: Is recording people talking in their dialects better than someone reading text? 78

6.4 Semi-structured interviews 78

6.4.1 One interview session 78

6.4.2 Feedback from one audio therapist 80

6.5 Performance testing 80

6.5.1 A low number of clients sending sequential requests 80

6.5.2 A high number of clients sending asynchronous requests 83

The average throughput and number of dropped requests for the Linode Kubernetes Engine cluster 83

The average throughput and number of dropped requests for the Single node Raspberry Pi server 84

The average throughput and number of dropped requests for the Raspberry Pi cluster 85

The average throughput and number of dropped requests for the Raspberry Pi cluster with intercommunication for failure detection 86

Comparing performance for HTML response (landing page) 87

Comparing performance for file lookups 88

Comparing performance for the rhyme finding algorithm 88

Comparing cluster configurations' performance as file sizes increase 89

Comparing stress test performance for all configurations' 90

7 Discussion 93

7.1 Literature review 93

7.2 Preliminary questionnaire 93

7.3 Primary questionnaires 94

7.4 Semi-structured interview 96

7.4.1 Low turnout in the recruitment process 96

7.4.2 Interview findings 96

7.5 Performance tests 97

7.5.1 Low client count with sequential requests 97

7.5.2 High client count with asynchronous requests 97

7.5.3 Test finding summary and possible solutions 98

7.6 Solution 99

7.6.1 Final deployment 99

7.6.2 Changes during development 100

7.6.3 Security 101

7.7 Strengths 101

Novelty 102

7.8 Limitations 102

7.9 Future work 103

7.9.1 Security 103

7.9.2 Fault tolerance 103

7.9.3 Persistent storage for users 104

7.9.4 Creating native applications 104

7.9.5 Extending the features already implemented 104

8 Conclusion 107

9 References 109

Appendix I. 116

Appendix II. 118

Appendix III. 120

Appendix IV. 122

Appendix V. 126

Appendix VI. 129

Appendix VII. 131

Appendix VIII. 133

Appendix IX. 136

Appendix X. 138

Appendix XI. 140

Appendix XII. 144

Appendix XIII. 146

Appendix XIV. 148
Appendix XV. 151
Appendix XVI. 154
Appendix XVII. 157
Appendix XVIII. 159

List of figures

Figure 1: Cochlear implant overview, placement of the implant, and electrodes. Image from Thomas Lenarz [1]. 1

Figure 2: An individual wearing a sound processor, magnetically secured to the head with the microphone resting on the ear. Photo: Bent Mittet Opdahl....... 2

Figure 3: Screenshot from “CI Hva du hører (CI-what you hear)" [15]. 6

Figure 4: Screenshot from “lyttetrening etter CI (audio training after CI)” [16]. 7

Figure 5: Screenshot from Meludia [19]. 8

Figure 6: Annual fees for renting a virtual machine with 2VCPUS, 4GB RAM, and at least 20GB SSD storage. Prices from the most popular cloud providers. 9

Figure 7: Annual fees for renting Linode Kubernetes Engine with different node specifications. Single node versus three nodes. 10

Figure 8: Different server frameworks throughput when running on a Raspberry Pi 4 Model B. Upper and lower bounds. 12

Figure 9: The search string utilized within the literature review. 15

Figure 10: A timeline showing each end-user involvement: the practical procedures above and the start of data gathering below the line. 16

Figure 11: Screenshots from the landing page (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right....... 34

Figure 12: Screenshots of the sound picture exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right. 35

Figure 13: Screenshot of the tone separation exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right. 36

Figure 14: A screenshot of the word separation exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right. 37

Figure 15: A flowchart describing the dataflow and main logic of the “detached” design 39

Figure 16: The architecture of the Single-board computer cluster. 39

Figure 17: A flowchart describing the dataflow and main logic of the “attached” design 40

Figure 18: The architecture of the Linode Kubernetes Engine cluster. 40

Figure 19: A screenshot of the fileserver’s simplistic front-end. 42

Figure 20: Data structure of the file index. 43

Figure 21: Pseudocode describing the lookup process....... 44

Figure 22: Showing a folder structure. 45
Figure 23: The result after flattening folder 1, depicted in Figure 22. 45
Figure 24: Shows how a flattened folder is displayed within the simplistic front-end. Folder In black and files in blue. 45
Figure 25: A flow-chart describing the load balancing process from the masters perspective 46
Figure 26: A screenshot from the fileserver's admin panel. 47
Figure 27: A flowchart describing the scaling process 48
Figure 28: A flowchart describing the process of adding files to the fileserver. 49
Figure 29: Screenshot showing an administrator's view while browsing files on the fileserver's "simplistic" front-end. 50
Figure 30: A flowchart describing the process of removing files 50
Figure 31: Pseudocode describing the rhyming algorithm. 52
Figure 32: Showing how Cascading Style Sheets were used to implement a responsive design 53
Figure 33: A flowchart describing how the sound picture exercise functions 54
Figure 34: HTML code showing how the video tags were implemented 54
Figure 35: Showing one of the four audio players within the sound picture exercise after adding audio, subtitles, waveform data, and filename. 55
Figure 36: Showing how the color and icon change as audio files are added. 56
Figure 37: Showing how the duration change as files are added to the different audio players (HTML video elements). 57
Figure 38: The Progress bar in a playing and paused state 57
Figure 39: A flowchart describing the data flow and logic behind the tone separation exercise 58
Figure 40: A flowchart describing the data flow and logic behind the word separation exercise 58
Figure 41: Showing how the requests were routed to reach the single-board computer cluster 59
Figure 42: A Linode Kubernetes Engine deployment configuration file. 60
Figure 43: A Linode Kubernetes Engine service configuration file. 61
Figure 44: A PRISMA flow diagram showing the steps taken to eliminate records from the literature review. 63

Figure 45: Showing how often participants performed audio training. 73
Figure 46: Shows the participants' reasoning for not doing audio training. 73
Figure 47: Shows where the participants performed audio training. 74
Figure 48: Shows how much time participants spent traveling to and from audio training. ... 75
Figure 49: Shows the participants' audio training content when training at home. 75
Figure 50: Shows the participants' audio training content when following an expertly made training program. 76
Figure 51: Shows the participants' happiness with the audio training opportunities available. 77
Figure 52: Shows the participants' preference between short sentences and longer stories. ... 77
Figure 53: Shows the participants' preference between recordings of people talking in their dialect or someone reciting texts. 78
Figure 54: Comparing throughput of endpoint leading to the landing page with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients. 81
Figure 55: Comparing throughput of endpoint leading to root folder lookup with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients. 81
Figure 56: Comparing throughput of endpoint leading to rhyme finding algorithm with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients. 82
Figure 57: Comparing throughput of file lookup endpoint (file size: 350KB, folder depth: 2) with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients. 82
Figure 58: Average throughput for the Linode Kubernetes Engine cluster. One request per client, 50-1000 clients. 83
Figure 59: Average number of dropped requests for the Linode Kubernetes Engine cluster. One request per client, 50-1000 clients. 84
Figure 60: Average throughput for the Single node Raspberry Pi server. One request per client, 50-1000 clients. 84
Figure 61: Average number of dropped requests for the Single node Raspberry Pi server. One request per client, 50-1000 clients. 85
Figure 62: Average throughput for the four-node Raspberry Pi cluster. One request per client, 50-1000 clients. 85
Figure 63: The average number of dropped requests for the four-node Raspberry Pi cluster. One request per client, 50-1000 clients. 86

Figure 64: Four node RPi cluster with intercommunication performance. One request per client, 50-1000 clients. 86
Figure 65: The average number of dropped requests is for the four-node Raspberry Pi cluster with intercommunication enabled for failure detection. One request per client. 50-1000 clients. 87
Figure 66: Comparing average throughput for the landing page endpoint for all configurations...87
Figure 67: Comparing average throughput for the file lookups for all configurations (file size: 350KB, folder depth: 2). 88
Figure 68: Comparing the average throughput for the rhyme finding algorithm for all configurations...88
Figure 69: Comparing average throughput for file lookups (file size: 200KB, folder depth: 100). One request per client, 50-1000 clients..89
Figure 70: Comparing average throughput for file lookups (file size: 400KB, folder depth 100). One request per client, 50-1000 clients..89
Figure 71: Comparing average throughput for file lookups (file size: 800KB, folder depth 100. One request per client, 50-1000 clients..90
Figure 72: Comparing average throughput for file lookups (file size: 1.6MB, folder depth 100. One request per client, 50-1000 clients...90
Figure 73: Comparing average throughput from stress test performance for all configurations. Seven requests per client, 50-1000 clients. 91
Figure 74: Image of the alphabetic audiogram (The "speech banana") [118]..........................94
Figure 75: Mobile and desktop version of Lyttetrening.no (final deployment)....................100

List of tables

Table 1: The research problems for this master project. 4
Table 2: The pros and cons for "CI-hva du hører (CI-what you hear)" [15]. 6
Table 3: The pros and cons of "Lyttetrening etter CI (Audio training after CI)" [16]. 7
Table 4: The pros and cons of Meludia [19]. 8
Table 5: The requirements of the minimum viable product. 24
Table 6: The functional requirements of the sound picture exercise. 26
Table 7: The functional requirements of the tone separation exercise. 27
Table 8: The functional requirements of the word separation exercise. 28
Table 9: The literature review result 65
Table 10: Comparison between thesis result (lyttetrening.no) and similar digital tools. 102

List of abbreviations

ACID – Atomicity Consistency Isolation and Durability

API – Application Programming Interface

ARM – Advanced RISC Machines (RISC – Reduced Instruction Set Computer)

AWS - Amazon

CI – Cochlear implant

CPU – Central Processing unit

CSS – Cascading Style Sheet

DDR4 – Double Data Rate 4th generation

DB – Database

DBMS – Database Model System

DPO – Data Protection Officer

GB – Gigabyte

GHz – Gigahertz

HTML(5) – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IO – Input/Output

iOS – Operating System developed by Apple, earlier known as the iPhone Operating System

IP – Internet Protocol

ISP – Internet Service Provider

JS - JavaScript

JSON – JavaScript Object Notation

KB - Kilobyte

K8’s – The vanilla distribution of Kubernetes

LKE – Linode Kubernetes Engine

MVP – Minimum Viable Product

NAS – Network Attached Storage

NSD – Norsk Senter for Forskningsdata

OS – Operating System

PC – Personal Computer

Pre-lingual – Before learning a language

Post-lingual – After learning a language

RAM – Random Access Memory

REST – Representational State Transfer

RPi – Raspberry Pi

SSD – Solid State Drive

SSL – Secure Socket Layer

SQL – Structured Query Language

TA – Thematic Analysis

TB – Terrabyte

TCP – Transmission Control Protocol

TLS – Transport Layer Security

UiT – University of Tromsø

URL/URI – Uniform Resource Locator/Uniform Resource Identifier

USD – American Dollar

VCPU – Virtual Central Proccessing Unit

VM – Virtual machine

VRST – Volere Requirements Specification Template

WCAG - Web Content Accessibility Guidelines

WebVTT - The Web Video Text Tracks Format

WSGI – Web Server Gateway Interface

W3C – The World Wide Web Consortium

XML – Extensible Markup Language

YAML – YAML Aint Markup Language

Abstract

A Cochlear Implant(CI) is an implant that replaces the functionality of the inner ear with an electronic prosthesis. The prosthesis stimulates the auditory nerves within the cochlea so that people with auditory disabilities regain hearing. While the medical process of inserting the implant is relatively straightforward, learning how to use the implant may be difficult. This thesis proposes a digital training platform that can aid with the learning process for CI users and lighten the transition into the audible realm. Providing efficient training tools available at home without assistance from third parties can increase independence for CI users, as it becomes easier to conduct audio training. The results from a questionnaire performed during the thesis with CI users as participants found that participants spent, on average, one hour traveling to and from training centers or audio therapists. Introducing home-based training tools tailored toward post-lingually deafened CI users can help free up patients' time and alleviate the learning process. Therefore, a digital training platform was developed, tailored towards the needs of post-lingually deafened CI users. CI users were involved via questionnaires and semi-structured interviews throughout the development process, and their feedback was gathered to inform the design and increase usability and effectiveness. The participants' feedback and the findings from a literature review based on related research crafted the requirements specification for the proposed digital training platform.

Throughout the thesis, the main goal was to create something that could be deployed and maintained, accessible and reachable for those who may use it. Therefore, a cost analysis of different cloud computing services was carried out to find the cheapest deployment options to avoid adding a monetary barrier to entry. Linode Kubernetes Engine (LKE) was an affordable and efficient option. Along with LKE, the efficacy of deployment through single board computers was investigated.

The data gathered through performance testing done during the thesis indicates that the current generation of the Raspberry Pi platform could be used to provide a cheap alternative to deployment if the number of expected users is relatively low. However, a side effect of using single-board computers is that it puts the responsibility to provide the availability and reliability requirements on the developer. As a result, you can not rely on the monetary incentives of a cloud provider. Since the availability and reliability requirements of the proposed digital training platform could not be ensured when utilizing single-board computers, the final deployment utilizes LKE and is accessible through: www.lyttetrening.no

1 Introduction

1.1 Cochlear implants

A Cochlear implant (CI) is a treatment for sensory deafness and replaces the functionality of the inner ear. The acoustic signals of the ear are transformed into electrical stimuli, which activate auditory nerves within the cochlea to restore hearing. An electronic stimulus prosthesis replaces the functionality of the inner ear, effectively replacing the functionality of the inner hair cells [1]. The implanted prosthesis resides underneath the skin. Figure 1 shows an overview of the Cochlear implant placement and the electrodes going into the cochlea, which provides the stimuli to the hearing nerve [1].

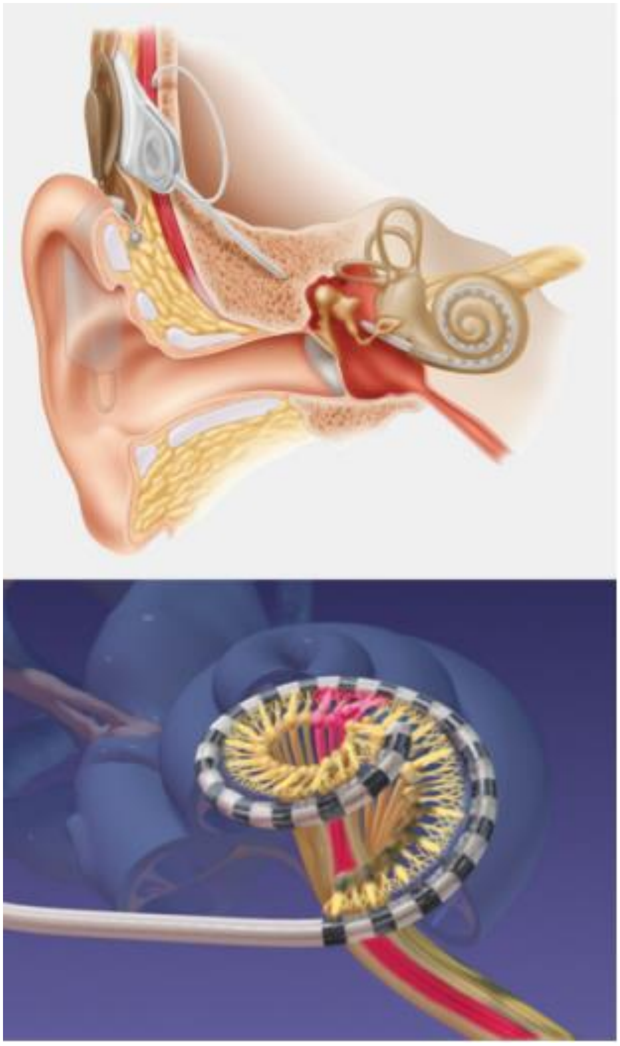


Figure 1: Cochlear implant overview, placement of the implant, and electrodes. Image from Thomas Lenarz [1].

The implant is connected to a sound processor with a microphone usually resting on the top of the ear. Figure 2 shows a person wearing a sound processor secured to the head by a magnetic connection to the implant. The sound processor transmits audio signals to the prosthesis, which transfers the electronic signals into the cochlea via the electrodes, finally stimulating the hearing nerve. Then, the stimuli are interpreted into sounds through neurological activity in the brain [1].

Figure 2: An individual wearing a sound processor, magnetically secured to the head with the microphone resting on the ear. Photo: Bent Mittet Opdahl.

Each year approximately 160 adult patients receive Cochlear implants in Norway [2], and in 2014, there were about 40 children approved for CI operations [3]. After the operation, the patients wait 4-6 weeks for the body to heal before turning on sound [4]. The process of hearing again can be particularly hard on an individual level. First, there is a relatively complicated procedure. Then there is the steep learning curve in retraining the brain to translate electronic stimuli into sounds and meaningful words. The steep learning curve stems from the fact that there is a limited number of separate channels that can convey the frequencies the external microphone is picking up. Post-lingually deafened CI users must learn to effectively map the degraded and distorted information onto stored representations collected via normal hearing [5].

Some factors can help flatten the extensive learning curve. First, reducing the amount of time spent in silence is beneficial. Green et al. [5] point to the time of onset of deafness and length of time post-implantation as contributing factors to studies' training outcomes. If not used, the ability to interpret and decode sounds is a skill we lose. As a result, the time to learn how to translate the electric stimuli will increase as the period spent in silence grows. Therefore, getting the CI operation approved as early as possible is essential. Second, the more training CI users do, the more generalizable benefits they may experience.

According to the national agreement on purchasing cochlear implants and accessories [6], Norwegian-speaking patients that undergo a cochlear implant operation can choose from Cochlear Norway [7], MED-EL Nordic [8], and Advanced Bionics [9]. Each provider has their implant with corresponding sound processors and accessories. The patients choose a Cochlear implant provider during pre-consultation, and the choice may differ due to preferences and need. All manufacturers have different applications and accessories for their hearing aids. The expected life expectancy of the implants is at least 25-30 years [6].

In the beginning stages of learning how to hear again, medical staff will be responsible for setting up the hearing aids, showcasing functionalities, and audio training. After this phase of turning on the sound and making fine adjustments, a user representative reported that the hospital would send a recommendation for audio training to the municipality where the individual resides. The communal health services will then recommend you to the nearest audio therapist, which will set up an appointment when they have the available resources. The audio therapists will go through exercises and help in the best way possible. Most audio training activities rely on having a second person available to read text or recite sentences and words. Eventually, the patients must be self-reliant and responsible for the audio training and the quality of the daily stimuli.

There are software-based tools available for audio training in Norwegian, but the tools seem to be tailored toward pre-lingually (pre-language acquisition) deafened patients. For patients that already know the language, the content and context of the audio training can be too easy. In some cases, this may result in the users relying on cognitive abilities instead of actively engaging in listening. There is a potential for better solutions tailored to post-lingually deafened individuals with CI. Learning to use the CI effectively can be complex and time-consuming. However, more tailored training tools with a broader scope may help alleviate some stress factors and increase user independence.

1.2 Motivation

Fogg [10] describes a persuasive design to focus on either increasing motivation and ability or triggering behavior. A persuasive design can provide one or more of these. This thesis aims to create a solution that can increase CI users' ability to perform audio training. The digital training platform should be a facilitator for triggering training behavior. Fogg describes the relationship between motivation and ability to be the critical factor in inducing behavior. Fogg also states that motivation and ability are not enough unless you have some trigger that makes people do the behavior you intend.

The motivational aspects of audio training are assumed to be there as CI users have undergone extensive surgery and are more than likely motivated to start using the implant. The solution created during this thesis will serve as a facilitator for the ability to perform audio training.

Triggers are the final ingredient within Fogg's recipe for persuasive designs. Fogg describes triggers to be some event that inevitably induces wanted behavior. Triggering audio training will not be the focus of this thesis. However, triggers could be provided within a closed feedback loop using reminders, as seen by Völter C. et al. [11]. The digital training platform could utilize push notifications for smart devices or send emails to desktop users. For the scope of this thesis, the end goal is to be a facilitator for audio training.

1.3 Problem definition

This thesis proposes a web-based digital training platform tailored toward people who have lost hearing after language acquisition. They have undergone a CI operation as a treatment for their deafness and need to perform audio training as a part of their learning process. The digital training platform should be a free and available web application where CI users can do audio training without needing assistance from a third party. The application must provide a sound library containing realistic environmental, background, animal sounds, and speech accompanied by subtitles. The audio training exercises will be based on techniques from current research and traditional audio training methods focusing on speech, tonal and pitch perception. Since a big part of the Norwegian language is the vast diversity when it comes to dialects, it is crucial to mirror the diversity within the collection of sounds available. The research problems which must be answered during the thesis are listed in Table 1. Problem number one is considered the main problem, whereas the subsequent numbers are considered sub-problems.

Table 1: The research problems for this master project.

Research problem	**Description**
1	How can we offer CI users the ability to perform audio training focusing on speech, tonal and pitch perception without the need for assistance from a second person or third party?
2	What is the best way to deploy such solutions to ensure that there are no monetary barriers to entry?
3	What is the best way of displaying training material to maximize the generalizable benefits for end-users?
4	How do we know that the solution is a good fit for the problem?

Several methods will be used to answer these research problems. First, the state-of-the-art audio training for Norwegian-speaking CI users will be established to get familiar with what other solutions have done. The acquired knowledge will be the foundation for analyzing the potential deployment solutions. Since the goal is to provide the result to the people who may need it without any barriers to entry due to cost, cost efficiency will be the most significant contributing factor to whether an option is viable.

Furthermore, the minimum viable product (MVP) and its requirements will be proposed based on preliminary interviews with a user representative, the findings from the literature review, and questionnaires. The effectiveness and usability of the MVP will be determined by end-user involvement. CI users will be recruited to test the solution while providing feedback through semi-structured interviews. Finally, I will analyze the collected data and discuss its results in the discussion section.

1.4 State of the art

1.4.1 Traditional audio training methods

The “audio therapist training handbook” [12] consists of several training exercises and methods used by audio-therapist to guide CI users through audio training. The pamphlet is a translated version of a training guide for English-speaking CI users by Med-el [8] and Geoff Plant [13]. It is accompanied by a user’s guide [14] directed towards guiding CI users through the exercises. These pamphlets were released in 2007 and have been an inspiration for the audio training program “CI-hva du hører (CI-what you hear)” [15]. After releasing “CI-hva du hører” the authors gave out a collection of texts and videos retaining information regarding audio training. The collection of information is called “Lyttetrening etter CI (audio training after CI)” [16]. Therefore, the training pamphlets published by Statlig Spesialpedagogisk Tjeneste (Statped) [17] may be an excellent place to understand how audio training has traditionally been done for Norwegian-speaking CI users.

The pamphlets focus on the analytical exercises from Geoff Plant’s [13] version. Each audio training activity provides instructions, training material, and general goals to accomplish while training. The main topics of exercises include:

- Phoneme exercises
    - A third-party reads phoneme-heavy text, sentences, words, or nonsense words (e.g., a:na, a:sa, a:da, a:ta), and the listener focuses on hearing specific phonemes. For example, the therapist will be reading, and the CI-user is listening for some particular consonant or vowel. Analytical exercises may include the user to determine whether the phoneme was present. Utilizing nonsense words is an excellent way to avoid the CI user guessing the correct answers [18].
    - For example, the audio therapist can read different words, repeat one, and have the CI user determine what word was spoken twice. The pamphlet explains several different versions of this type of training in detail.
- Synthetic exercises
    - These are exercises similar in nature to regular communication. These exercises challenge the CI users to use skills acquired from the phoneme exercises.
    - These exercises often take the form of conversations consisting of different topics between the therapist and the CI user. The training material has listed several questions that can start such conversations.
    - Number identification can be a synthetic exercise. For example, a CI user can identify dates or specific numbers.
    - Speech tracking exercises resemble regular communication with a coherent speech pattern. The CI user is prompted to repeat the phrases or sentences spoken by the therapist.

1.4.2 Digital training tools available in Norway

CI-hva du hører (CI-what you hear)[15] is an audio training tool that allows CI users to train with Norwegian speech from poems, sentences, individual words, and numbers. Additionally, the software provides different environmental sounds. “CI-hva du hører” also provides musical content from well-known Norwegian folk music. “CI-hva du hører” enables CI users to do audio training at home as it is available through a web browser. It has a listening mode and a training mode where the user can easily switch between the two. “CI-hva du hører” is a good tool for audio training. It is the perfect tool for pre-lingually deafened CI users who need to learn sentence building. The exercises provide good information on how the Norwegian language works. The pros and cons of “CI-hva du hører” are listed within Table 2. A screenshot of the “CI-hva du hører” web application can be seen in Figure 3.

Table 2: The pros and cons for "CI-hva du hører (CI-what you hear)" [15].

Pro	**Con**
The training tool teaches how to build sentences syntactically.	Some sounds can be perceived as non-realistic and of low quality.
The training tool provides a diverse spectrum of sounds.	The training can be too simple for post-lingually deafened CI users. Users may guess the correct answers and rely on cognitive abilities instead of actively engaging in listening
The training tool offers a diverse range of exercises.	

Figure 3: Screenshot from “CI Hva du hører (CI-what you hear)" [15].

Lyttetrening etter CI (Audio training after CI) [16] is a follow-up to "CI-hva du hører (CI-what you hear)". It provides helpful information relating to hearing deficiencies, Cochlear implants, and audio training. However, it is not a direct training tool such as "CI-hva du hører" as it does not have any repeatable training exercises you can do on your own. Instead, it provides videos and informational texts describing several audio training exercises. All audio training exercises listed require a second person to recite training content. The pros and cons of "lyttetrening etter CI" are listed in Table 3. A screenshot of the "Lyttetrening etter CI" web application can be seen in Figure 4.

Table 3: The pros and cons of "Lyttetrening etter CI (Audio training after CI)" [16].

Pro	Con
"Lyttetrening etter CI" has good information about hearing deficiencies, CI, and audio training.	All exercises rely on having a second person helping with the audio training activities.
"Lyttetrening etter CI" has several descriptive videos and text explaining several exercises.	"Lyttetrening etter CI" does not offer repeatable exercises through the web browser.
	Some links are outdated and no longer point to anything.

Figure 4: Screenshot from "lyttetrening etter CI (audio training after CI)" [16].

Meludia [19] is mainly used for practicing a wide range of musical concepts and is aimed toward normal-hearing people. Although the tool is designed for normal-hearing people, it can be applied to CI users as an audio training tool. Meludia offers a wide range of musical exercises that emphasize frequency separation. There are different themes for the audio training exercises, including density, rhythm, spatialization, stable/unstable, and melody. Using Meludia as an audio training tool can be challenging for CI users. The difficulty level is significant and may be too tricky in most cases. Meludia is available for Android and iOS devices and is a web application reachable through browsers. The pros and cons of Meludia are listed in Table 4. A screenshot from Meludia can be seen in Figure 5.

Table 4: The pros and cons of Meludia [19].

Pro	**Con**
Meludia is a good tool for musical exercises.	Starting at a difficulty level designed for normal-hearing people can be too hard for CI users.
Meludia is accessible in app stores and available as a web application.	Meludia only provides musical-themed training.
Meludia is an excellent place to draw inspiration regarding musical training.	

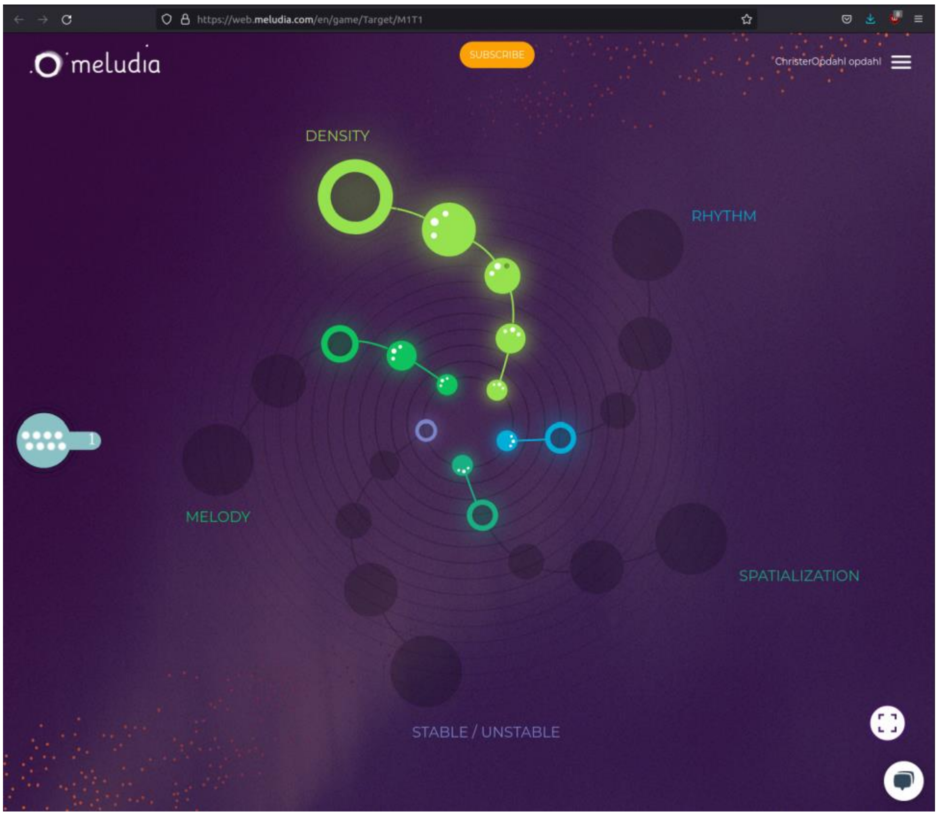


Figure 5: Screenshot from Meludia [19].

1.5 Deployment options

1.5.1 Cloud computing cost analysis

From chapter 1.4, it is clear that the typical audio training tool either stores or creates audio and displays the audio as training content. Therefore, the proposed training platform must be able to store and distribute audio files to users to be utilized as audio training content, such as seen in "CI-Hva du hører (CI-what you hear)" [15] and Meludia [19]. However, the number of audio files that need to be stored is limited, and each file will be relatively small in size and likely no longer than one minute in length.

Several one-minute-long audio files were recorded and inspected to calculate the minimum storage space required. It turns out that a minute of sound within the mp3 format is approximately equivalent to one megabyte of data. Therefore, with 20GB of storage space, the solution could contain 20 thousand one-minute-long audio files. Of course, a final solution holding 20 thousand audio files is not a realistic expectation, but the system should have at least 20GB of storage to ensure that the system is not running short on storage space any time soon.

Traditional cloud computing services can be relatively expensive depending on instance size and storage space. If the price for having the solution deployed was too high, it could undermine the ability to provide an audio training tool free of charge. Therefore, cloud provider prices were looked at to understand the available options better. Each provider's native price calculator was utilized to find these prices. The annual fees were calculated by providing the following specifications to each provider's native price calculator: 20GB solid-state hard drive (SSD) storage, two virtual central processing units (VCPU), and 4GB RAM. The prices listed in Figure 6 are the annual fees for renting virtual machines from the most popular cloud providers.

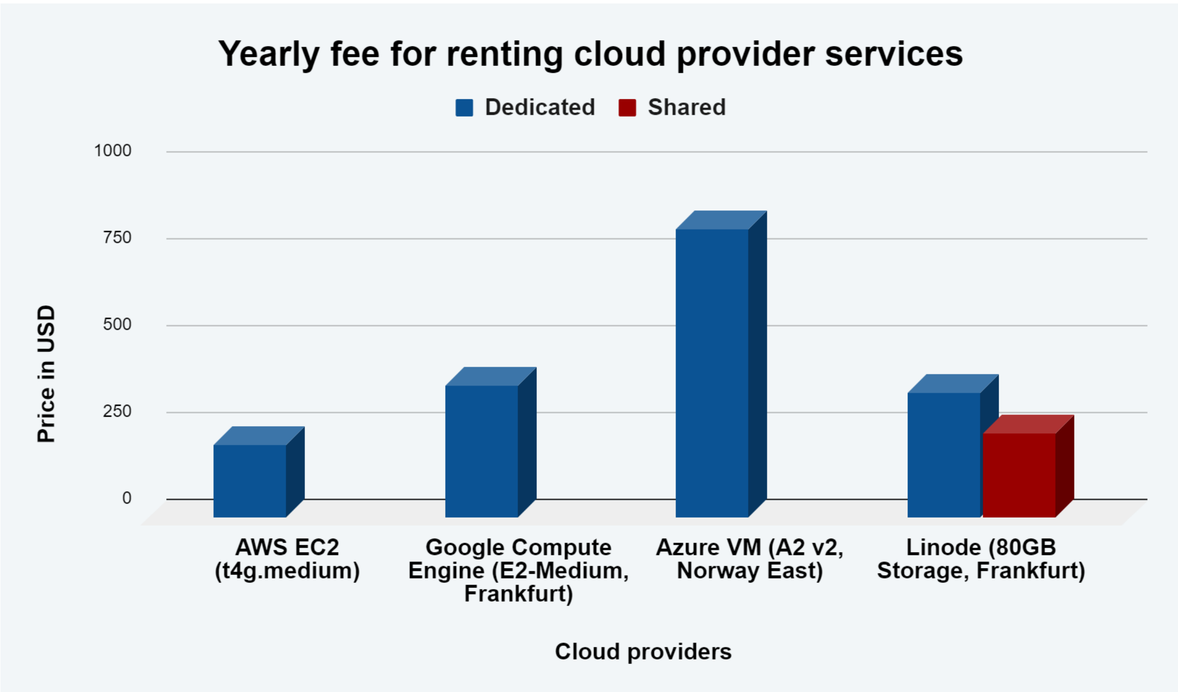


Figure 6: Annual fees for renting a virtual machine with 2VCPUS, 4GB RAM, and at least 20GB SSD storage. Prices from the most popular cloud providers.

Figure 6 shows that Amazon Web Services (AWS) [20] has the lowest annual fee with their t4g.medium instance. Linode [21] is the second cheapest as it has an option to sign up for a shared plan which gives you 80GB of storage for the yearly fee of 240 USD. Additionally, Linode has a service called Linode Kubernetes Engine (LKE) [22], where you can set up a Kubernetes cluster with as many nodes as preferred. Using a clustered solution for file storage may decrease the cost of deployment as the load can be balanced across several instances with cheaper specifications. With an LKE cluster, the master node is free, so you only pay for the worker nodes. The number of worker nodes (replicas) that should be running at any time is stated within a deployment configuration file, and the load balancing service will balance the incoming load across the replica servers. The price of one LKE worker node compared to three with different node specifications is compared in Figure 7.

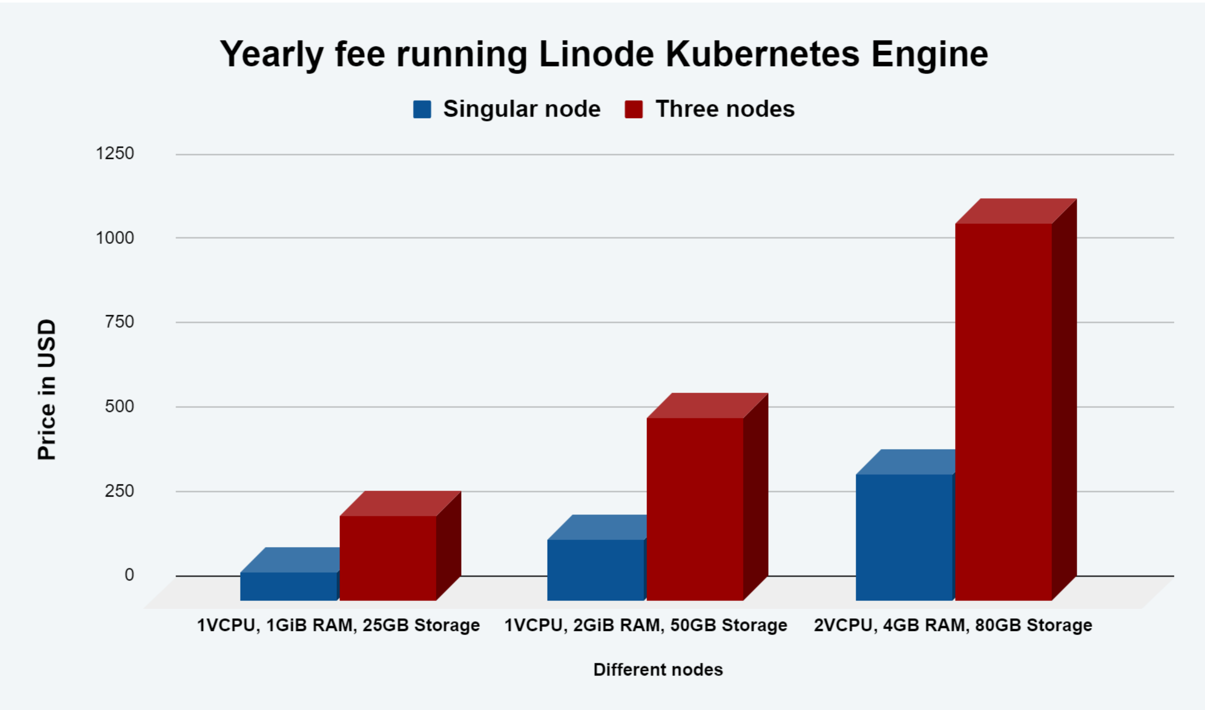


Figure 7: Annual fees for renting Linode Kubernetes Engine with different node specifications. Single node versus three nodes.

Running three nodes with 1VCPU, 1GiB RAM, and 25GB storage gives the annual fee of 253 USD. That is 45 USD more than running a single 2VCPU, 4GB RAM, and 20GB AWS t4g.medium instance. However, having automatic scaling, load balancing, and failure handling may be worth the extra expenses.

Amazon, Google, and Azure have similar services that provide Kubernetes clusters, AWS EKS [23], Google GKE [24], and Azure AKS [25]. However, these services are enterprise-oriented, and the prices start at 70 USD a month for AWS EKS, only increasing for the other options. Therefore, these prices are too high even to consider.

Linode Kubernetes Engine

Kubernetes [26] is a robust and powerful framework for container orchestration. It comes in many forms, and the open-source framework has spawned many different versions, trying to either increase the framework's capacity or simplify its installation process. K8’s (vanilla

Kubernetes) [26], K3's [27], Minikube [28], Red Hat OpenShift [29], VMware Tanzu [30] to mention a few.

Similarly, the Linode Kubernetes Engine (LKE) is a container orchestrator that simplifies cluster creation and management. Linode provides the underlying infrastructure, and all you need to do is deploy containers and set up the management rules. Deployment is done by regular Kubernetes deployment configuration (YAML) file, where you specify an image and the number of replicas. Give the configuration file to LKE, and the underlying framework initializes the replica nodes. The LKE cluster will always ensure that the specified numbers of replicas are up and running, and if a container stops, LKE automatically launches a new one. In addition, if you want to update a deployment, a rolling update can be initiated. A rolling update will shut down a container, create a new one with the updated image and continue to do so until all outdated containers have been updated.

After deploying a cluster, you can set up load balancing services to enable access from the outside world. A load balancing service is as easy to set up as the deployment itself. You specify the service in a YAML file, transfer it to the cluster, and the underlying framework handles the initialization. LKE is incredibly easy to use for being a Kubernetes engine. It's well documented and abstracts cluster orchestration. In addition, it is affordable.

1.5.2 Single-board computers

Although the Linode Kubernetes cluster is relatively cheap, there may be other deployment options than cloud provider services. For example, single-board computers are cheap and power-efficient and may be an alternative as the expected number of users are relatively low. Any software developed for single board computers would also run in more traditional cloud computing environments. Additionally, the solution could always be transferred to a cloud computing environment should there be an influx of users. Furthermore, utilizing single-board computers could avoid the costs related to storage, and the front-end could be deployed by cheaper cloud provider options such as AWS Lambda [31] or similar options.

A Raspberry Pi (RPi) [32] is a single-board computer platform that is easy to set up and use. It can run Linux operating system (OS), making development a one-to-one comparison to previous experiences. More importantly, there is evidence pointing to the efficacy of RPi as a deployment platform. For example, Schot et al. [33] looked into running the Hadoop [34] database model system (DBMS) on a cluster of RPi 2 nodes to investigate the platform's efficacy for big data and video streaming. In addition, Shrivastava et al. [35] showed that RPi was a viable option for running home-based network-attached storage (NAS).

The RPi platform is on its 4th edition and has gotten several hardware upgrades since the RPi 2. The RPi model 4 B now runs a 1.5Ghz, four-core ARM processor with 4GB of DDR4 RAM. The price of one RPi 4 Model B on eBay is 63 USD (without shipping). Add on the price of a MicroSD card, and it becomes 78 USD for one deployment, assuming you already have a router, USB-C, and ethernet cables. Even when rounding up the price to 80 USD, it would still be cheaper than the yearly price of the weakest LKE node. Furthermore, if an RPi lasts longer than a year, it would reduce the annual fee to a one-time purchase.

Although utilizing the RPi platform would technically give us the cheapest deployment option, I would have to pay for Static IPs from the internet service provider (ISP). The average monthly fee for a static IP address was investigated by contacting several ISPs and averaged 5.69 USD (50NOK) each month. The average annual cost was approximately 63 USD.

There is an additional cost to utilizing the RPi platform for deployment. Namely, availability and reliability. Since the RPi is physical hardware placed in a fixed location, it will inevitably be victim to power outages or connection problems due to common day-to-day issues. Additionally, there is no way to spawn a new instance if one should fail since its physical hardware, and any failure that does not resolve itself must be fixed by manual inspection. Whether this price can be paid remains to be seen and will likely always be subject to analysis.

Preliminary testing

Preliminary testing was done to see whether a Raspberry Pi 4 Model B could provide enough throughput to be a viable deployment platform. With a small number of potential concurrent users, the system does not need much throughput. But if single board computers could not provide satisfactory results for the most simplistic server, there would not be any point in further research.

The preliminary testing was carried out by writing a simplistic web server in the top server framework for the Python [36], Rust [37], and Go [38] programming languages. A client program sent out 10,000 concurrent requests, and the throughput was calculated by $tput = \frac{Nrequests}{\text{time}}$. To get a good view of each framework upper and lower bound in relation to throughput, the test was carried out 50 times for each framework.

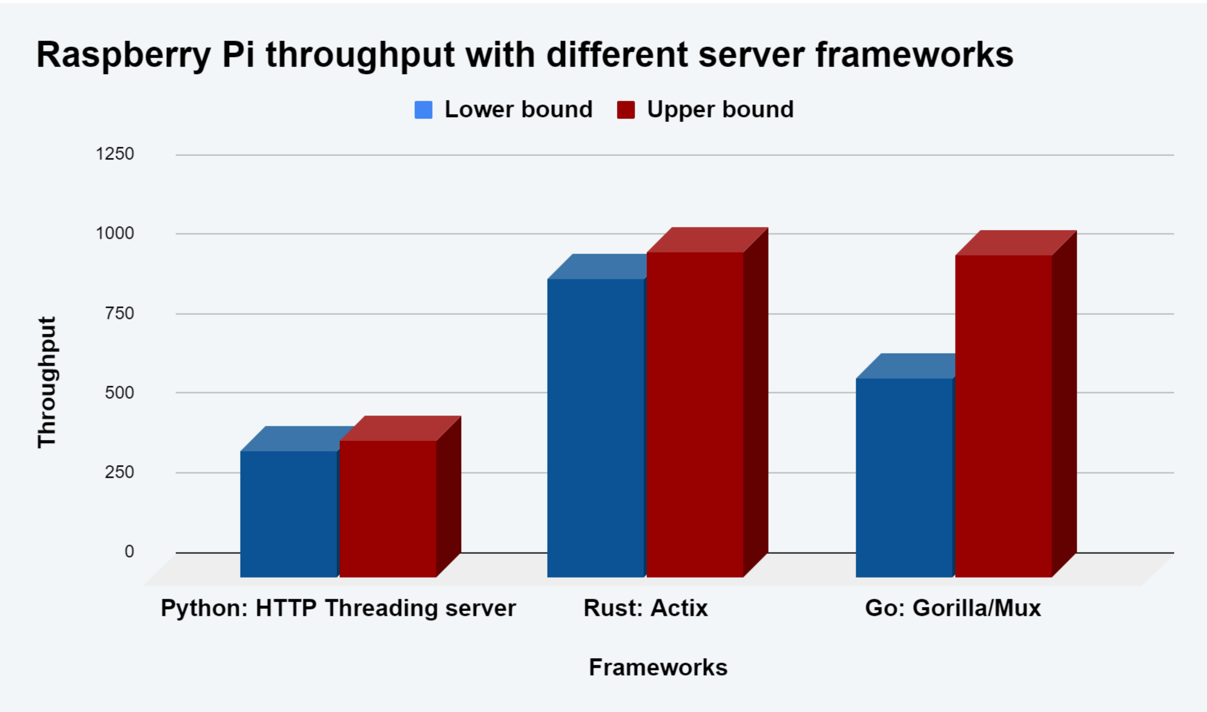


Figure 8: Different server frameworks throughput when running on a Raspberry Pi 4 Model B. Upper and lower bounds.

Figure 8 shows that the Rust programming language combined with the Actix server framework had the highest and most stable throughput out of the three languages and framework combinations. As a result, the decision to write the back-end in Rust was made. However, the results are based on a simplistic throughput test. Compared to more robust machines, the results are not significant. The overall performance and potential compute power could be increased by harnessing the modular nature of single board computers and aligning them in a clustered configuration. The RPi's biggest strength may be the ability to start with one single-board computer and scale out as needed. In addition, it's cost and power-efficient. Although the RPi platform is essentially a tiny computer, the platform is highly moldable and easy to use.

If the necessary steps are taken to avoid bottlenecks in compute power, a cluster of RPi nodes might be able to provide a cheap alternative to deployment. Load balancing the cluster will be a crucial feature to ensure efficiency as it scales. The solution developed will be tested to run on an RPi cluster, to investigate the efficacy of the Raspberry Pi 4 Model B as a deployment platform. The RPi cluster will be compared to the performance of an LKE cluster running an equal number of nodes.

1.6 Security

The demographic of the proposed digital training platform is vulnerable as the usage is directly linked to their medical condition. Therefore, significant efforts should be made to protect the users from the leakage of sensitive information. For example, the digital training platform could let CI users add audio files to the sound catalog. However, this would imply several software complications and open up privacy concerns. Therefore, such features will not be implemented, and the training platform will only offer dynamic digital exercises with no possibility of adding user data to the system.

Additionally, every necessary step to ensure CI users' privacy must be taken during end-user recruitment and feedback gathering. Storing and analyzing the data must be done while following the guidelines given by UiT’s Data Protection Officer.

1.7 Thesis structure

The thesis follows this format:

Chapter 1 – The introduction chapter describes the problem and the primary motivation for writing the thesis. Additionally, the chapter briefly describes the state-of-the-art audio training tools currently available to Norwegian-speaking CI users and discusses several deployment options.

Chapter 2 – The method chapter explains the methods used during the thesis.

Chapter 3 – The requirements chapter lists the functional and non-functional requirements of the proposed digital training platform.

Chapter 4 – The design chapter describes why design choices were made and explain the architecture of the proposed digital training platform.

Chapter 5 – The implementation chapter describes the steps taken to implement the architectural and design choices made in the design chapter.

Chapter 6 – The tests and results chapter displays the results from the questionnaire, literature review, and deployment testing.

Chapter 7 – The discussion chapter discusses the results seen within the test and result chapter, the digital training platform created during the thesis, the future work needed to optimize the solution, and argues the strengths and limitations of the thesis.

Chapter 8 – The conclusion chapter briefly summarizes how the research questions were answered to conclude the thesis.

2 Method

2.1 Literature review

The literature review addresses common methods, practices, techniques, and platforms for audio training, technology, and similar projects. Search queries were done by utilizing the digital databases PubMed [39], ACM DL [40], and IEEE Xplore [41]. Similar search strings were applied to all databases, with some syntactic differences between the three. The search string is seen in Figure 9.

```
("audio training"      OR
 "auditory training"   OR
 "frequency training"  OR
  training)
AND(
 "post op"             OR
 "after getting"       OR
  after)
AND(
 "cochlear implant"    OR
  CI                   OR
  ci)
NOT(
 "prelingual deaf"     OR
  syndrome             OR
  pediatric            OR
  elderly)
```

Figure 9: The search string utilized within the literature review.

There were four exclusion criteria, research related to prelingually deafened CI users, syndromes, pediatric and elderly, and papers published before 2015. First, Pre-lingually deafened CI users were excluded as these individuals face several additional challenges relating to learning how to hear with CI. For example, learning language syntax and sentence building. Second, when performing searches related to CI, many articles were directly linked to specific medical conditions or syndromes. Since this was deemed irrelevant to the thesis, these articles were excluded. Third, pediatric and elderly CI users were excluded as the thesis focused on the adult population. Additionally, there are related cognitive issues for both age groups. Finally, papers published before 2015 were excluded from the literature review to get the newest relevant research.

The literature review result can be seen in Table 9, presented within chapter 6.1. In addition, some literature review findings directly affected the requirements specification. These findings are displayed in chapter 3.1 and later discussed in chapter 7.1.

2.2 End-user involvement

As the end-user is in focus, it was essential to reach out to people who have implemented a Cochlear Implant and audio training into their daily lives. The CI users' feedback was utilized during the development. Creating something similar to what CI users already know may make the result more familiar and easy to use.

Two questionnaires helped answer research problems three and four, see Table 1. Namely, what is the best way of displaying training material to maximize the generalizable benefits, and how do we know that the solution fits the problem. Figuring out what the content contained within the training material should look like is imperative. If the content that makes up the training material cannot generate benefits from training (problem three in Table 1), the solution will not fit the problem, and there is no point in displaying the content in the first place. Problem four in Table 1 relates to the solution being a good fit for the problem. Any credible answer to this has to come from end-users. Therefore, qualitative interviewing was performed in a semistructured manner as described by Brinkmann [42] to gather user feedback after testing the digital training platform.

Recruitment of end-users was done utilizing the Facebook group “CI-Gruppa (CI-Group).” “CI-Gruppa” is the largest Facebook community for Norwegian-speaking CI users and currently has more than 1400 members. Consulting the UiTs Data Protection Officer (DPO) ensured the participant's privacy rights. Norsk Senter for Forskningsdata (NSD) [42] was consulted whenever the DPO could not confirm whether the steps taken to ensure the participants' privacy rights were sufficient. Consulting NSD is done through applications, which will only be approved when all necessary steps have been taken.

Audio therapists were recruited to test the digital training platform and give feedback. A recruitment message was posted within the Facebook group “Audiopedagog – er/ skal bli/ ønsker å bli? (is/ becoming/ want to become audio therapist)”. In addition, a CI researcher was reached out to via email to see if the researcher were willing to test the digital training platform to give an opinion on it. Unfortunately, I received no reply from the researcher, but one audio therapist gave feedback through an email response after trying the digital training platform. The questionnaires and semi-structured interviews will be explained in more detail during the following sub-chapters. In addition, a timeline of each end-user involvement is seen in Figure 10.

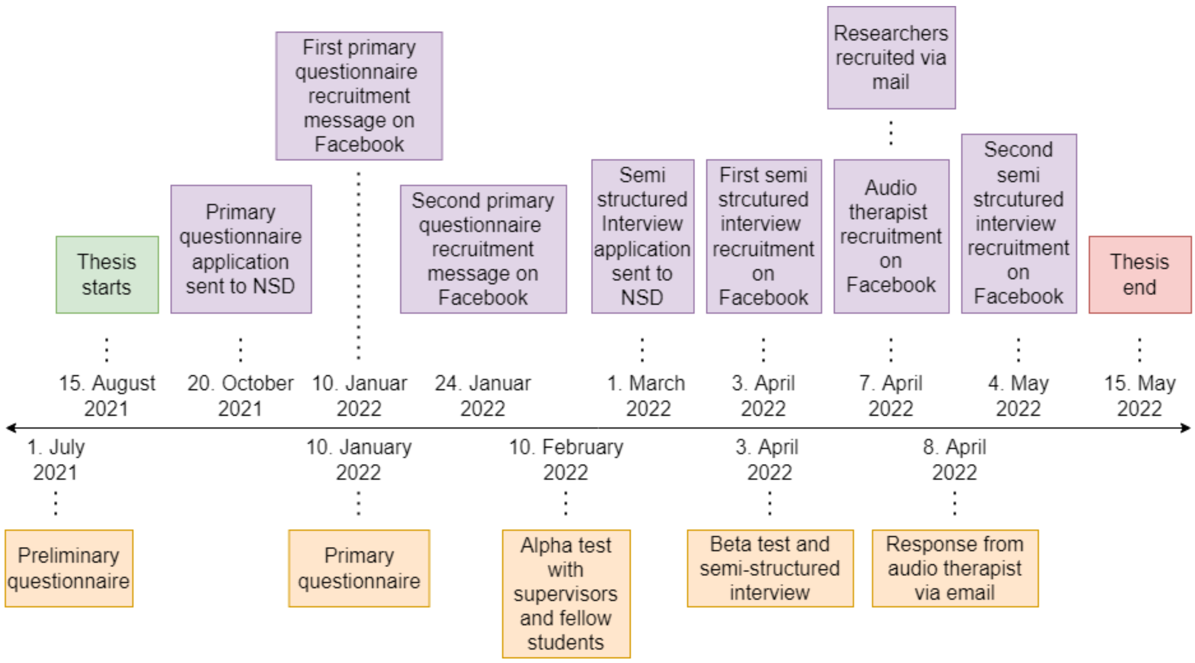


Figure 10: A timeline showing each end-user involvement: the practical procedures above and the start of data gathering below the line.

2.3 Preliminary questionnaire

Before the thesis started, a preliminary questionnaire was performed in the summer period. Unfortunately, the initial questionnaire was not subject to inspection by UiT's Data Protection Officer. Therefore, the results will not be shown, but the conclusions drawn from the questionnaire will be offered in chapter 6.2 and discussed in chapter 7.2.

2.3.1 Data gathering

The preliminary questionnaire consisted of six questions and had 35 participants. The questionnaire was posted utilizing Google Forms [43] and contained the following questions:

- Statement: I have undergone a CI operation (yes or no).
- Statement: I am or answer on behalf of (adult, teenager, child)
- What platform do you use the most?
- Statement: A digital training tool should focus on (Voices with dialects or specific sounds).
- Mention sounds or words you find challenging.
- Mention situations or scenarios you find challenging.

The questionnaire information, questions, and recruitment message are listed throughout the following appendices: Appendix I - Appendix III.

2.3.2 Data analysis

The quantitative data gathered from participants' answers were analyzed using descriptive statistics. Any answers given within the text input areas were grouped into generalizable terms after the fact.

2.4 Primary questionnaire

The privacy of the primary questionnaire participants was ensured by consulting UiT's Data Protection Officer (DPO). However, the DPO could not confirm that there were no identifiable questions and advised that an analysis of anonymity should be conducted to know whether to apply to Norsk Senter for Forskningsdata (NSD) [44]. After consulting with supervisors, an analysis of anonymity was performed. Since the analysis identified potential risks, we decided to apply to NSD as it was the safer option. NSD approved the application, and the questionnaire was published within the Facebook group "CI-gruppa (CI-group)." The results from the primary questionnaire are shown in chapter 6.3 and discussed in chapter 7.3.

2.4.1 Data gathering

The primary questionnaire consisted of 10 questions and had 28 participants. The questionnaire was posted utilizing nettskjema.no [45] and contained the following questions:

- Statement: I have undergone a CI operation (yes, no).
- How often do you do audio training?
- Why do you not perform audio training? (only asked if stating that they did not perform audio training)

- Where do you perform audio training?
- How much travel time is involved with your audio training?
- What is the audio training content when you train on your initiative?
- What is the audio training content when you follow a program made by audio therapists or other experts?
- Are you happy with the audio training tools available today?
- Statement: Short audio clips with short sentences or singular words are better than longer clips with longer stories or sentences (different levels of agreement as answers).
- Statement: A person talking with their dialect is better than someone reciting text (different levels of agreement as answers).

The questionnaire information, questions, DPO correspondence, analysis of anonymity, NSD approval, the first and second recruitment messages, and feedback from one audio therapist are listed throughout Appendix IV to Appendix XVII.

2.4.2 Data analysis

The quantitative data gathered from participants' answers were analyzed using descriptive statistics. Any answers given within the text input areas were grouped into generalizable terms after the fact.

2.5 Requirements specification

The Volere Requirements Specification Template (VRST) [46] was used to help craft the initial requirements. The VRST thinks of requirements as belonging to different types, functional, non-functional, constraints, project drivers, and project issues. The requirements specification of this thesis will focus on functional and non-functional requirements and list them utilizing a derived version of the VRST.

In addition, several features and audio training activities were crucial to exist within the digital training platform to provide a minimum viable product (MVP) [47]. The features and audio training activities shown in Table 5 laid the foundation for the functional and non-functional requirements discussed in chapters 3.4 and 3.5. The functional requirements are listed in Table 6, Table 7, and Table 8, and the non-functional requirements are listed in chapter 3.5.

The initial requirements were based on a user representative's reported experience, the literature review, and questionnaire findings. Throughout the development process, the requirement specification was subject to change.

2.6 Development

The tools and hardware utilized during the development process were the following:

The development platforms used during the thesis were Raspberry Pi (RPi) 4 Model B [32] single-board computers and a Linode Kubernetes Engine cluster.

Docker [48] was used as a deployment tool, which simplified the deployment process. Docker files were used to build the Docker images, which would run in detached mode to ensure that the images kept running after ending the Secure Shell (SSH) sessions [49].

Kubernetes configuration files [22] were used to deploy and set up the Linode Kubernetes Engine cluster services.

Visual studio code [50] was used throughout the development process as a text editor.

The development followed an agile-based iterative approach where the requirements specification were updated as the software was developed [51]. After crafting the requirements specification, it was clear that the digital training platform proposed needed a storage manager, preferably capable of conveying the hierarchical structure of the underlying file structure.

Several database model systems (DBMS) were investigated, and the result is seen in chapter 4.1.1. However, most DBMS solutions did not fulfill the requirements established as they are generally designed for more complex tasks and operations. Therefore, both the back and front-end were developed in-house during the thesis. The back-end stores a collection of audio files, and the front-end displays the available audio training exercises and content.

The sound collection was created by downloading Creative Commons [52] licensed videos from YouTube [53]. The video sources used are listed in Appendix XVIII. After downloading the videos, the sounds were edited by a musical editing software called Ableton live [54]. Next, I asked fellow students with different dialects to record themselves reciting texts and, in some cases, superficial conversations. Then, to get more story-based and longer content, I wrote two texts emphasizing the letter S and recorded myself reading them. The letter "S" was mentioned as challenging by participants of the preliminary questionnaire (see chapter 6.2). Finally, there are four Norwegian poems read by three different speakers, all provided by Tekstforslag.com [55].

One audio training activity that came out while crafting the requirements specification was a word separation exercise. Users are prompted to listen to a spoken word before finding the word amongst similar choices. A potential solution to create such an exercise dynamically was inspired by Rimsmia [56]. Rimsmia is a web application where you enter terms as textual input and get rhyming words as output. Unfortunately, Gyldendal (the publisher of rimsmia) does not provide an application programming interface (API) or other tools that enable you to use their algorithm within your applications. The solution was to create an algorithm that could iterate over a list of Norwegian words to find rhyming words similar to rimsmia.

I created a list of around 15 thousand Norwegian words, which the rhyme finding algorithm could iterate over. Several steps were taken to create the list, and all steps are explained later in chapter 5.2.6. At its simplest, the process consisted of gathering a large body of Norwegian text that could be split into single words. Then, a simplistic rhyme finder algorithm iterates through each word within the list, comparing all words' endings against some term given as input. Finally, the output contains the words with the most significant number of equal letters.

2.7 User Testing

2.7.1 Alpha testing with supervisors and colleagues

Before testing with CI users, an alpha test phase took place, testing the platform in-house utilizing supervisors and fellow students as testers. The interview guide was crafted and refined during this stage with the help of my supervisors. As a result, the supervisors provided good insight into the interview questions and how to conduct semi-structured interviews. The semi-structured interviews are described in further detail in chapter 2.8.

In total, there were three sessions with the supervisors. The first two sessions were done over a Teams [57] call where the supervisors would share their screens while going through the digital training platform. During the testing phase, the only input the supervisors were given was a scenario. They were told they had just gotten their sound processors turned on and were told by friends that the digital training platform was suitable for audio training. We found good ways to improve the training platform by tweaking visuals and adding features, making the training platform more intuitive and easier to use.

The first two sessions were done one-to-one, implementing changes based on the feedback between each session. Implementing changes between the sessions prevented the supervisors from commenting on the same thing. The first session focused on the desktop browser version, while the last session focused on smartphone testing. After implementing the ideas from sessions one and two, the supervisors went through the training platform for the third time but focused on the mobile version.

After utilizing the supervisors as test subjects to understand how the process of doing the tests should be, fellow students were asked to test the digital training platform and gave feedback verbally. The feedback gave further indication of the usability of the platform.

2.7.2 Beta testing with end-users

After several development iterations based on the feedback from the alpha test, a recruitment message was posted in the Facebook group “CI-Gruppa (CI-Group)” to recruit CI users as participants. Anyone could try the digital training platform independent of participating in giving feedback through interviews. The CI users who signed up to be user testers would test the digital training platform before participating in a 15-minute semi-structured interview session over the phone.

2.8 Semi-structured interviews

The semi-structured interview sessions were recorded and transcribed to avoid taking notes during the interviews if the participants consented to the session being recorded. The interview participants' rights to privacy were ensured by consulting UiT’s Data Protection Officer (DPO). The DPO pointed to UiTs guidelines for audio recording, which we referred to while applying to NSD. Knowing the policies led to a straightforward application process. As a result, Norsk Senter for Forskningsdata (NSD) approved the application without further requirements since the application could refer to the guidelines pointed to by the DPO. The semi-structured interview information, DPO correspondence, approval from NSD, and the first and second recruitment message can be seen throughout Appendix XI to Appendix XVI.

In addition, the results from the semi-structured interviews are shown in chapter 6.4 and discussed in chapter 7.4.

2.8.1 Data gathering

After the participants had conducted enough testing to form an opinion on the different exercises, they participated in semi-structured interviews. As long as the participants consented, the sessions would be recorded, and the recordings were deleted after manually transcribing them. The interviews were semi-structured, meaning the participants were encouraged to talk freely, without interruption from the interviewer. Participants were asked a new question from the interview guide when exhausting a topic.

2.8.2 Data analysis

The qualitative data gathered from the interviews were subject to thematic analysis (TA), a widely used method by health researchers Braun et al. [58]. TA offers an accessible and theoretical flexible approach to analyzing qualitative data. Braun and Clarke [59] describe a TA as a method for identifying, analyzing, and reporting patterns or themes within data. It minimally organizes and describes your data set in detail. Patterns within the gathered data were found by utilizing the six-step guide defined by Braun and Clarke [59]. These include data familiarization, code generation, searching for themes, reviewing themes, defining and naming themes, and finally, producing the report.

2.9 Performance testing

From chapter 1.5, it was decided that the most viable deployment options were Linode Kubernetes Engine (LKE) and the single-board computer platform Raspberry Pi 4 Model B (RPi). Two designs tailored toward each option were developed, deployed, and subject to performance tests to investigate and compare their performance. The deployment with the best performance and capability to provide the availability and reliability requirements stated in chapter 3 became the final deployment. The testing was carried out by putting the two different deployments under heavy loads generated by sending concurrent requests. Finally, the result was tallied, plotted, displayed, and discussed. The test results are shown in chapter 6.5 and are discussed in chapter 7.5.

Two different test methods were utilized. One consisted of several concurrent clients, each sending one thousand sequential requests. This test compared a single node RPi server against a four-node RPi cluster with and without intercommunication for failure detection. The results from this sequential test are heavily dependent on the requests' roundtrip times due to being sent sequentially. Nevertheless, it gave an excellent metric to compare different RPi configurations as the nodes have comparable roundtrip times. However, this is not true for the LKE deployment. The LKE deployment is physically placed in Frankfurt and has much higher roundtrip times. Therefore, testing with a few concurrent clients sending sequential requests to get a performance metric will not give comparable results between the LKE and RPi clusters. Thus, to compare LKE and RPi deployments, another test was performed.

The second test pushes the client number as high as possible to see how many concurrent clients the different server configurations can handle simultaneously. Each client sent one request, and the number of clients started at 50 and incrementally increased to 1000. This test is less dependent on the roundtrip times as all requests are sent simultaneously rather than sequentially.

The following endpoints were tested:

- GET “/”: points to the landing page and returns HTML5 [60] content.
- GET “/”: with request header “content-type: application/json” involves transforming the indexing structure into a JSON object. This endpoint is only tested within the sequential test as this is removed from the attached design used within the LKE deployment.
- GET “/rhyme”: iterates through the list of words to find rhyming words.
- GET “/<path>”: File lookup requires reading a file and transferring it to the clients.

Both tests involve measuring throughputs for the individual endpoints mentioned above. Additionally, the throughput for file lookups was tested with increasing file sizes to see the impact of folder depth and file size. Finally, the last test suite consisted of a stress test where each client would send a request to each endpoint mentioned above, plus additional file lookups with a folder depth of 100 and file sizes from 200KB increasing to 1.6MB (200KB, 400KB, 800KB, and 1.6MB). The stress test clients sent seven asynchronous requests each.

3 Requirements

3.1 Literature review findings that impacted requirements

From the literature review shown in Table 9, it was clear that most research agrees on some aspects related to audio training. This sub-chapter will outline the key findings from the literature review that directly affect the proposed solution's requirements. The literature review findings can be seen in chapter 6.1 and discussed in chapter 7.1.

Giving CI users the ability to train for complex scenarios where there are multiple talkers or multiple sounds layered on top of each other is a recurring topic in current research [5], [61], [62]. Therefore, it is crucial to implement some form of this training exercise in any audio training tool. Most of these articles have a maximum of two sounds layered on top of each other. The proposed digital training platform aims to take this one step further and enable CI users to do further layering.

There seems to be a consensus amongst researchers that using musical concepts in audio training to increase tonal, pitch, and or speech perception is an efficient training method [63], [64], [65], [66], [67], [62], [68], [69]. Therefore, having some form of musical training is crucial to any audio training tool.

Utilizing spoken words, phrases or sentences is important audio training activities. Traditionally this seems to be the most basic form of audio training [65], [70], [5], [71], [11], [72], [73], [18], [74].

Additional effort should be made to give the participants all the information they need to understand and interpret the sound they are hearing. There should be a focus on implementing as many audio-visual queues as possible [66], and efforts towards removing the possibility of guessing the correct answers should be made.

3.2 Questionnaire findings that impacted requirements

This sub-chapter outlines the key findings from the questionnaires that directly affect the proposed solution's requirements. The questionnaire results can be seen in chapters 6.2 and 6.3. Finally, the result is discussed in chapters 7.2 and 7.3.

There was no consensus on a single preferred platform. However, there was a slight preference for iOS platforms but nothing significant. Therefore, the digital training platform must be available on the most popular platforms, iOS, Android, and desktop. The solution will be developed as a responsive web application.

The participants reported several scenarios to be challenging, and there was no "one size fits all" type solution for audio training content. Additionally, there was no significant difference in content from an expertly made training program versus training done on the participant's initiative. Therefore, the scope of the sound library should be as broad as possible.

3.3 Minimum viable product

The minimum viable product (MVP) requirements were based on the user representative's reported experience, the literature review, and the questionnaire findings. The requirements are listed in Table 5. The term minimum viable product comes from Eric Ries's book called The Lean Startup [47] and is based on a “build-measure-learn” feedback loop. The MVP is the minimum baseline product offered to users or customers. The MVP is a version of the product that may have just enough features to be usable by customers. The MVP is the earliest stage where customers can use the product and give feedback, explaining their user experiences for future product iterations [47].

Table 5: The requirements of the minimum viable product.

Req #	*Description*	*Importance*
1	The MVP must be a web application reachable from desktop, iOS, and Android devices.	Crucial
2	The MVP must provide users with a comprehensive catalog of sounds to cover most users' needs. In addition, the sounds must be realistic and should include a bare minimum of environmental sounds, speech, and singular words.	Crucial
3	Users must be able to simulate challenging scenarios such as the sound of a busy restaurant with multiple talkers and background noise. The sounds should be gathered into a catalog. The displayed catalog must be easy to navigate.	Crucial
4	Users must be able to practice tone separation.	Crucial
5	Users must be able to practice separating similar words	Crucial
6	The MVP must be deployable in a cost-efficient manner.	Crucial

Since we are looking to investigate the efficacy of using single-board computers as a platform for deployment, the architecture must be deployable in a clustered configuration. Enabling a clustered configuration is vital as single-board computers have relatively weak processors. Therefore, taking advantage of the granular nature of single-board computers may make the system more robust.

The other deployment option deemed viable in chapter 1.5 was a Linode Kubernetes Engine (LKE) cluster. Kubernetes cluster functions differently from the suggested Single-board computer cluster as it generally takes singular docker images and duplicates them across nodes to create replicas. Therefore, the system must be adaptable to run within an LKE environment.

Suppose Single-board computers and LKE environments are deemed non-viable in terms of deployment. In that case, the solution must be deployable as a single node monolithic server in a traditional cloud computing environment.

3.4 Functional requirements

3.4.1 Sound picture requirements:

A sound picture is audio signals layered on top of each other to convey enough information to simulate realistic environments, such as a busy restaurant. Conceptually this feature must offer users the ability to combine and layer audio files. The audio files must be able to be played as a single entity or alongside other audio files as a part of a bigger sound picture. The users must be able to browse the collection of audio files and add them to the player. The sound picture requirements specification can be seen in Table 6.

3.4.2 Tone separation requirements

The tone separation exercise is intended to train CI users in separating tones with different frequencies. At least two distinct sine waves must be displayed to the user. The user should be prompted with corresponding input buttons to be able to choose one of the two. The feature should be a game-like experience where the user can keep track of their scores to have a sense of how they performed at the end of the training session to encourage and motivate users. The tone separation requirements can be seen in Table 7.

3.4.3 Word separation requirements

The word separation exercise is intended to train CI users in separating similar words from each other. There must be an audio file audibly displayed to the user containing the voicing of a single word. The user should be prompted to hear the audio file and find the word amongst similar choices. The text on the displayed input buttons must be similar to the word within the audio file, as the exercise should force the end-user to separate the first distinctive frequencies of a word. This feature should also be a game-like experience to encourage and motivate users. The word separation requirements are listed in Table 8.

Table 6: The functional requirements of the sound picture exercise.

Req #	Description	Rationale	Originator	Fit Criterion	Priority
1	To be able to search through a catalog of sounds freely.	Enable users to find the sounds they find the most challenging.	Author	The available content within the back-end is easily viewable and as easy to navigate through as the underlying folder structure of the back-end	1
2	Layer multiple audio files	Enables users to simulate audible scenarios that they find challenging.	Questionnaire and literature findings	Four audio files can be inserted into the players simultaneously.	2
3	Play multiple layered audio files at the same time or singular audio files.	To increase the difficulty level as the user layers more audio files on top of each other.	Author, questionnaire, and literature review findings.	Multiple audio files can be inserted into the players and started simultaneously as the user clicks play.	3
4	All audio files must have their content transcribed and contained within subtitles files that can be displayed along with audio content	To enable users to read along as the audio progresses	Litterature review findings	Subtitles are dynamically inserted and shown when adding audio files to the players.	4
5	Audio wave forms that can display the sound audio-visually to users.	To enable users to identify spikes as speech and lower aggregates as background noise	Litterature review findings	Audio waveforms are dynamically generated and displayed as the user inputs audio files.	5
6	An audio file progress bar can be modified by user input to skip forward or backward.	Enables users to see and modify where the audio is concerning the audio file's duration.	Litterature review findings	Progress bar displaying the current place within the audio files' overall duration. Thumb button, which can be interacted with to change the current place within the audio file duration.	6
7	Toggle subtitles	Enable users to turn on or off the subtitle tracks depending on whether they want them on. May increase or decrease the difficulty level.	Litterature review findings	Subtitles can be turned on or off.	7

Table 7: The functional requirements of the tone separation exercise.

Req #	Description	Rationale	Originator	Fit Criterion	Priority
1	Dynamically generate two frequencies	To be audibly displayed to the users	Author	Audio is dynamically generated, and the frequencies of the sine waves are adjustable.	1
2	Display the audio files audibly to the users	Enable the users to listen to the sine waves	Author	The sine waves can be played independently from each other.	2
3	Input buttons to choose one or the other sound.	To know whether the user is correct or not	Author	Clickable buttons will give the users feedback on whether the choice was correct.	3
4	Session statistics must be tallied and displayed to the user	To enable users to see the result of the training session to compare the progress to previous attempts	Supervisors	Users' session performance statistics are shown	4
5	The exercise should be a game-like experience	To motivate training recurrence and user retention	Literature review findings	Keep score of user performance and reset it if the user chooses the wrong option.	5
6	Give the users the option to set the frequency range of the sine waves.	Enables users to set the frequency range to the range they find most challenging.	Literature review findings	Input sliders are in place that will change the range of the frequency of the generated sinewaves.	6

Table 8: The functional requirements of the word separation exercise.

Req #	Description	Rationale	Originator	Fit Criterion	Priority
1	Dynamically generate similar words	To be displayed as input buttons	Author	Similar sounding words are dynamically found and inserted into input buttons.	1
2	Have one single word able to be played to know what word the users are looking for within the options	Enable the users to listen to the word and find the same word within the similar input buttons	Author	The word that is to be the correct choice is present within the input button choices and can be played as audio.	2
3	Session statistics must be tallied and displayed to the user	To enable users to see the result of the training session to compare the progress to previous attempts	Supervisors	Users' performance statistics are shown	3
4	The exercise should be a game-like experience	To motivate training recurrence and user retention	Literature review findings	Keep score of user performance and reset it if the user chooses the wrong option.	4

3.5 Non-functional requirements

3.5.1 Availability

The system must be available whenever the users want to use the platform.

3.5.2 Reliability

If deployed in a clustered configuration, the system must be robust enough to avoid crashing on a system-wide level if node failure should occur.

3.5.3 Usability

The user interface must be intuitive and follows accessibility concepts such as the Web Content Accessibility Guidelines (WCAG) [75] to provide a smooth user experience. Additionally, independence and a fun environment to maintain interest in training are essential concepts for user retention. Finally, motivate the users with gamified content and training methods to see statistics over training sessions and overall progress.

3.5.4 Scalability

As the realistic numbers of concurrent users are relatively small, the system must be adaptable and easy to scale in the event of an influx of users. For example, suppose the back-end is deployed as a clustered system of single-board computers. In that case, it must be able to add nodes without taking down or interrupting the rest of the system.

3.5.5 Responsiveness

There was no clear platform preference among the questionnaire participants. Therefore, the solution must fit the most common platforms. Thus, the web application must follow a responsive design to allow users to use the website from any device as long as there is a web browser present.

4 Design

4.1 Back-end design choices

4.1.1 Investigating possible database model systems

The storage manager responsible for storing the audio files used as audio training content must be able to serve files concurrently to clients while avoiding bottlenecks to maintain efficiency and correctness. Relational database systems were considered, such as MySQL [76] and PostgreSQL [77]. These database model systems (DBMSs) are tried and true. Still, there is no need for SQL compliance(adhering to SQL rules), ACID transactions (atomicity, consistency, integrity, and durability), or rollback features [78]. At its simplest, the file server must be able to receive a path and return the corresponding file if it exists within the file server's root folder.

On the NoSQL [79] side, there are MongoDB [80] and CouchDB [81]. Both can function as key-value stores, for example, by storing filenames as keys and the file location as value. Although this would work, MongoDB needs a 64-bit system which may lead to compliance issues when investigating single board computers. CouchDB is probably the most viable option here, as CouchDB works for Raspberry Pi and has native support for file storage.

All the DBMSs investigated have one thing in common, they are designed to solve more complex issues with more complicated operations than needed for the use case. The system requires a simplistic fileserver that can be responsible for storing, managing, and sharing audio files. Ideally, the server should share them in a hierarchical structure so that a human third party can sort through the content effortlessly.

As the investigated DBMSs do not fit the use case, a simplistic distributed file server was created. The server can do file lookups based on relative paths and will distribute load amongst nodes. The file server does file lookups on an indexing structure generated at server startup and stored in memory for easy and fast access.

4.1.2 Replication and consistency scheme

The Raspberry Pi 4 B CPU is powerful for its size and runs a 1.5Ghz ARM processor with four cores. The platform has 4GB of DDR4 RAM. Although it is powerful for its small size, it is relatively weak compared to traditional processors. Distributing load across the cluster and optimizing memory usage is vital to increase efficiency. Therefore, the master node distributes load within a cluster of replicas. By treating each node as a replica, the load can be distributed equally to reduce the amount of CPU load on a system level. The master node will not do an extensive lookup to find the node holding the specific data, as done within a Chord [82] circle, but rather delegate to any available replica. The system balances the load in a round-robin schema and will redirect requests as they come in to detach the master node from the response. This replica-based distributed system was chosen as it is easy to maintain and set up and avoids routing traffic through the master node.

The back-end file server is a distributed replica-based system. The fileserver can run in a a single node or clustered configuration on a collection of Raspberry Pi single-board computers. Each node in the system is an identical replica, and each node can be either a master or a worker node.

The master node has these main tasks:

1) Distribute load to maximize efficiency and minimize the system's overall load.

2) Distribute added files uploaded by the system administrator, or make sure removed files are deleted across all nodes.

3) Is responsible for accepting and contacting new nodes.

There is no need for a strict consistency scheme. For example, users may see that there are audio files available in the list of content but be unable to download the content from a node that is not yet updated. Other than being a slight annoyance, there will not be any part of the system where having updated replicas is crucial. Therefore eventual consistency will be leveraged.

4.1.3 Load balancing

Most traditional clusters pipe the communication through the master. However, this may not be a viable option for single-board computers due to performance-related issues. Instead, a pairing scheme may be more fitting, where the master essentially delegates work in a “delegate and forget” manner. Even though the master node delegates work, we still want the master node to answer requests and contribute resources to the overall system. Therefore, the master node will utilize a round-robin schema to know whether it should delegate or do the work itself.

Delegating work is done utilizing HTTP redirections, a well-defined standard within the HTTP protocol. However, redirecting to worker nodes will hurt the system transparency [83], as the URL change when browsing the fileserver. Therefore the front and back-end will be detached from each other to justify using redirection to avoid routing traffic through the master. The transparency issue may be avoided if the back-end is only used as storage space. For example, suppose the users only interact with the front-end. And the HTML served by the front-end contains enough Javascript functionalities to dynamically fetch data from the storage space without the users knowing it. In that case, the cluster transparency will no longer matter. The back-end will only be indirectly interacted with through JavaScript running within the users' browser. Since the users will never see any of the steps taken by the underlying architecture, the end-user will be oblivious to the actions happening “under the hood.” Therefore, I will knowingly sacrifice transparency for functionality in the hopes of increasing performance by not routing traffic through the master node.

By utilizing redirection, the master node will make worker nodes respond to the requests detaching itself from the response. All redirection protocols reside within the 300’s HTTP status range. There are different redirection protocols within the HTTP protocol. Realistically the Temporary Redirect (HTTP 307) is the only fit as this protocol does not change the original request, except by redirecting to the new location set in the response header.

Additionally, we need to ensure that the node receiving the delegated work is up and running. Liveness can be assured by pinging the receiving node before delegating the work. If the node does not reply, the master node performs the work instead.

4.1.4 Scaling

Scaling is done by adding nodes to the cluster. The scaling process is initiated by entering a new nodes contact information into the scaling field present within the master nodes admin panel. Next, the master node sends its file index to the new node so that the new node can duplicate it to be it's own. After copying the file index, the new node will have enough information to go through the replicated index to download each file and eventually become a replica. The first duplication request is the only request unique for the scaling process. Therefore, the only request that the master must handle itself is the first request that will start the scaling process. After that, the rest of the work can be delegated to other worker nodes.

Whenever a new node has gone through the master’s file index and has caught up to the master, the new node will notify the master that it is ready to be a part of the cluster. The master will broadcast the notification, and each node will add the incoming node as its neighbor.

4.2 Front-end design choices

The front-end is a RESTful server, serving responsive HTML files containing JavaScript functionalities and stylesheets based on which endpoints are requested. The term REST (representational state transfer) was introduced in 2000 by Roy Fielding [84] and provided practical constraints to guide Application Programming Interface (API) designs. At its most rudimental form, a RESTful API explicitly uses HTTP methods for communication between server and client. It is stateless, it handles every incoming request uniquely, and all information needed to handle the request properly is present in the HTTP request. In addition, it exposes URIs and transfers data via XML or Javascript Object Notation (JSON) or both [85]. The front-end is written in Python utilizing Flask [86]. Figure 11 shows a screenshot of the front-end’s landing page.

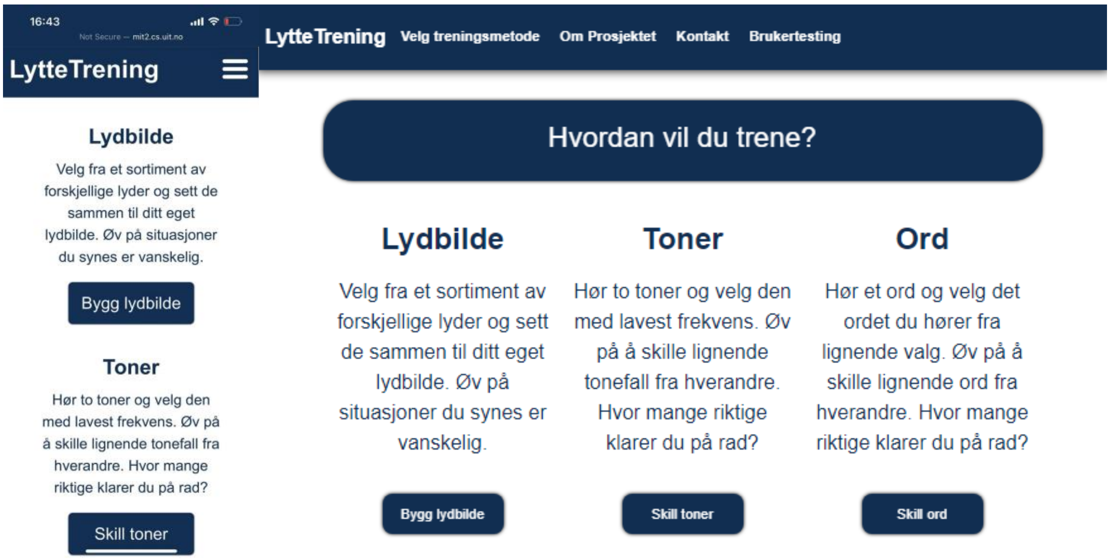


Figure 11: Screenshots from the landing page (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right.

4.2.1 Sound picture

As this feature aims to enable users to layer audio, multiple audio channels must be present. From an HTML perspective, this means either using audio or video tags. Video tags were chosen due to having the option of enabling subtitles. W3C [87] has developed a framework for implementing subtitle tracks for the HTML5 video tag. The framework is called WebVTT: The Web Video Text Tracks Format [88]. The subtitles are created manually by transcribing audio files into the VTT format and inserted into HTML track tags as users add audio to the different channels. Then, waveforms are generated and displayed using the wavesurfer.js [89] library.

The users had to be able to browse the sound catalog. The catalog is made browsable by displaying the content in a drop-down list. The available content is pulled from the back-end and transformed from a JSON response to displayable HTML content by the front-end. After transformation, the HTML is inserted into an HTML template before serving the template to the user.

The users can add and remove audio files, toggle subtitles, change volumes, play and pause audio, and interact with the progress bar. Figure 12 shows a screenshot of the sound picture exercise.

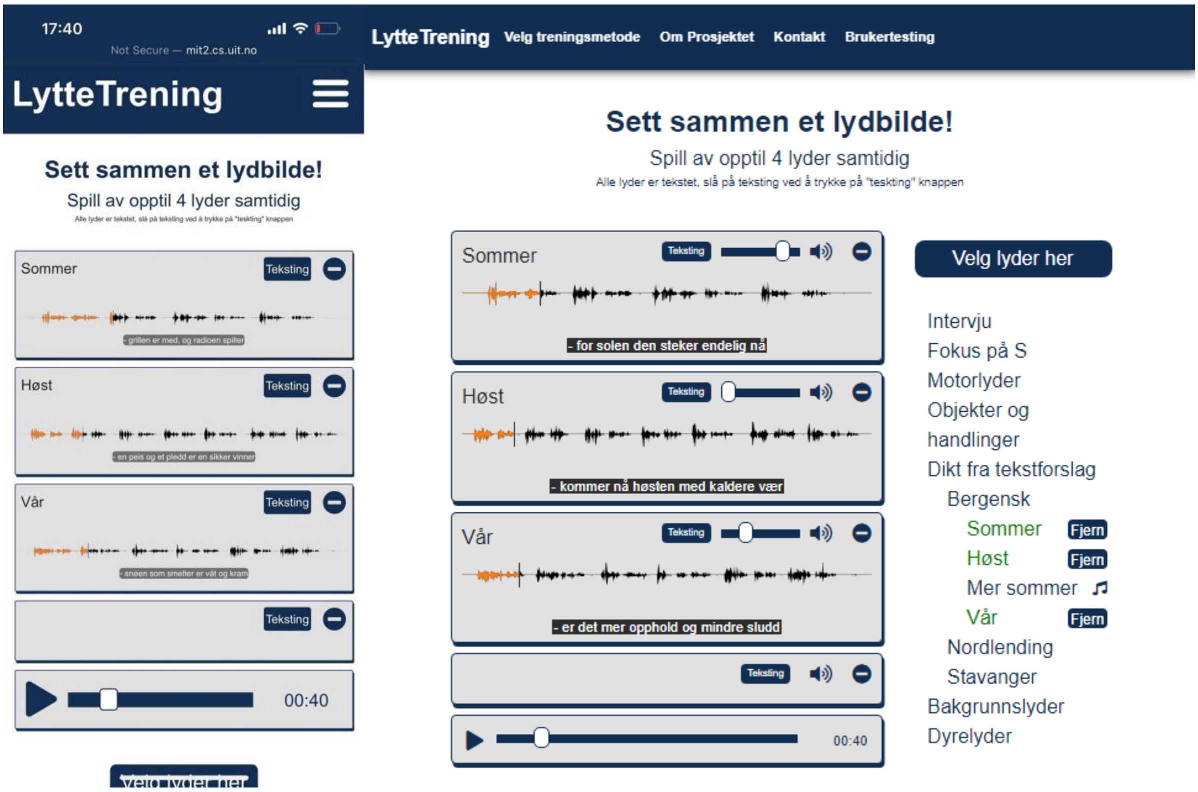


Figure 12: Screenshots of the sound picture exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right.

4.2.2 Tone separation

The sinewaves used in the tone separation exercise are generated within the user's browser utilizing JavaScript. The sinewave frequency range is set to random by default but can be changed by interacting with input sliders within the settings menu. By default, the users must find the sinewave with the lowest frequency. The users can change this to find the highest tone by interacting with the settings menu. Next, the users must choose from two buttons (see Figure 13 buttons a and b), and each button corresponds to one of the tones. After choosing one of the tones, the user will be prompted by a message displaying if they choose the right or wrong answer. Progress and session results are tallied dynamically and communicated to the users. In addition, there are dropdown menus for instructions and settings. Figure 13 shows a screenshot of how the tone separation exercise works.

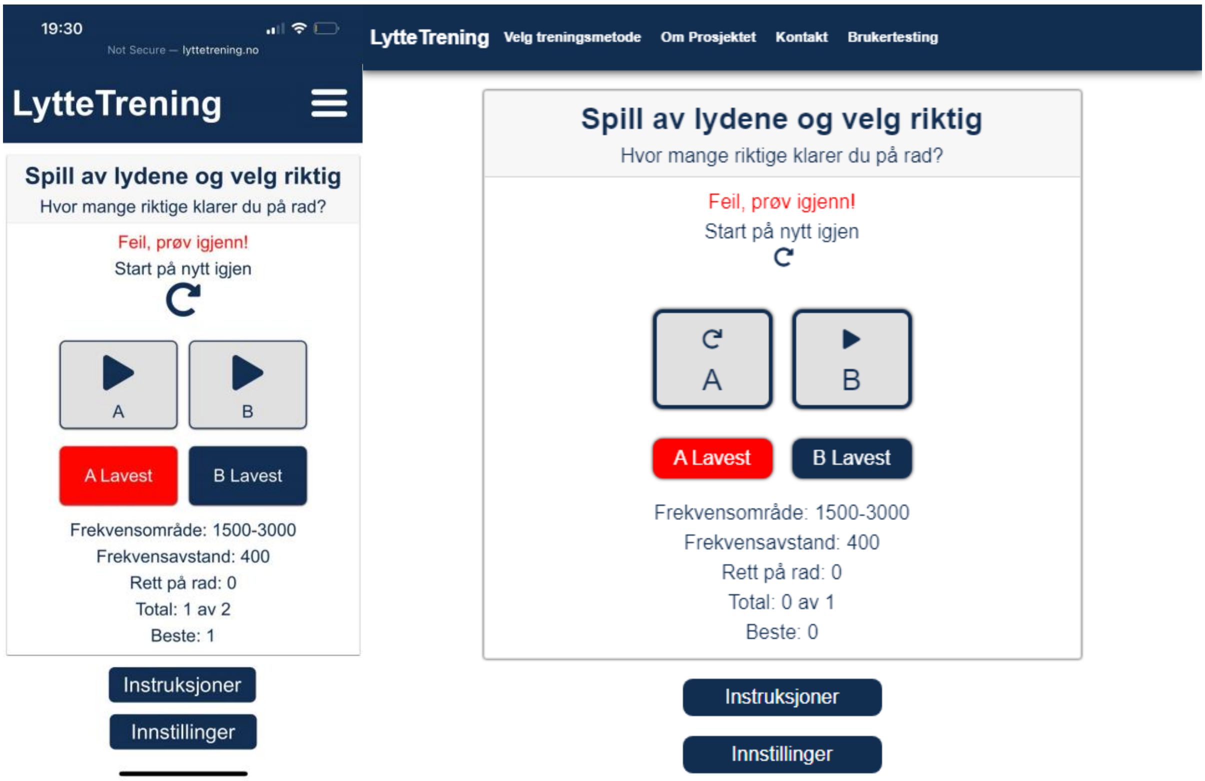


Figure 13: Screenshot of the tone separation exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right.

4.2.3 Word separation

The word separation exercise is a synthetic exercise inspired by Statped's [17] handbook for audio therapists [12]. The exercise focuses on separating similar words such as "stake," "make," "take," etc. A possible dynamic solution is utilizing rhyming words. On paper, finding rhyming words is not complicated. You compare the ending of two words, and they rhyme if they are equal. However, determining if two words rhyme phonetically is more complicated. For example, the words "stove," "move," and "love" rhymes on paper, but phonetically they are not similar.

Soundex [90] is an algorithm that gives a score indicating the level of phonetic similarity between two words. Double Metaphone is even better at it, as it is an improved version of Soundex. However, the algorithms were initially intended to work for English pronunciations and did not give sufficient results when testing with Norwegian words. Furthermore, consistently finding phonetically rhyming words was a more complex problem than initially thought. Therefore, the solution became the simplistic algorithm described earlier.

Such an algorithm would need to iterate over an extensive list of words, and the quality of the search would be directly linked to the number of words present in the list. So, for example, say the word "excellent" is the input, but the only word present within the list is "thesis," there would be no output. Therefore, a relatively extensive list of Norwegian words was created, and an algorithm could iterate over it to find rhyming words. The rhyme finding algorithm and the list of words reside within the back-end and will respond to an HTTP GET /rhyme request.

Google text-to-speech [91] converted the words contained within the list from textual terms to audio. Google's algorithm for text-to-speech works sufficiently well for Norwegian text. It can be weirdly voiced when converting longer sentences, but the API works sufficiently well for single words. Using Google text-to-speech was the most efficient solution to the problem, as manually recording 15 thousand words was not a viable option.

When sending an HTTP GET /rhyme request to the back-end, the response contains a list of rhyming words and the path needed to fetch the audio file containing the voicing of the term used as input for the rhyming algorithm. Each request starts by retrieving a random term present within the list, which becomes the input of the rhyme finding algorithm. After finding words that rhyme, the information is wrapped and sent as a JSON response. All terms within the list were downloaded and stored to avoid contacting Google text-text-to-speech API while running the server. Figure 14 shows a screenshot of how the word separation exercise works.

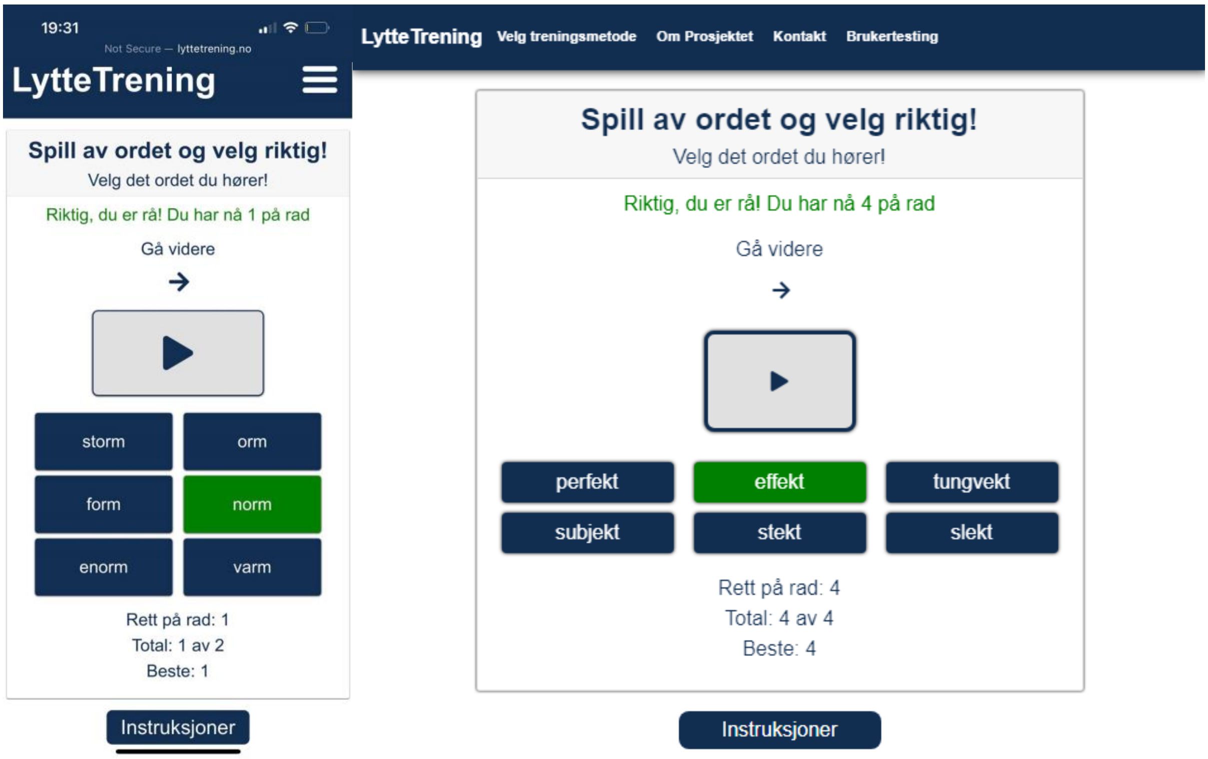


Figure 14: A screenshot of the word separation exercise (lyttetrening.no). The mobile version is on the left, and the desktop version is on the right.

4.3 The System as a whole

The overall system consists of two main parts, the front and back-end. The back-end is a centralized distributed data storage that stores audio files in a hierarchical folder-like structure. The back-end can run in a clustered or standalone configuration. The master will balance load among equal nodes utilizing redirects to detach the master from the response. Scaling is done by adding new nodes to the cluster.

The front-end will display the available content present within the back-end to enable users to browse the catalog of sound. Individual audio files will be fetched and displayed when interacting with the content present within the web application. Fosr example, when the users click a specific audio file or start using the word separation exercise, the individual audio files will be fetched. The files are fetched from the back-end by Javascript running within the users' browser. JavaScript subsequently updates the content contained within the web page to display the fetched content.

The level of control the user can excerpt by their inputs is limited to choosing specific audio files, playing audio files, or using input buttons that give users feedback from predetermined answers. The definition of content in this scope is either specific audio files or generated audio. The only type of content that does not exist and must be generated locally within the web browser are sinewaves used within the tone separation exercise.

4.4 Attached and detached design

As mentioned in chapter 3.3, the system must be deployable utilizing single-board computers, Linode Kubernetes Engine (LKE), and traditional cloud computing environments. Therefore, the design must be split into two versions: one “attached” design intended for the LKE deployment and a “detached” design for the single board computer deployment.

The “detached” design is described so far. It detaches the front and back-end from each other, as discussed in chapter 4.1.3. This choice was made to circumvent the transparency issues from utilizing redirection as a load distribution method. By separating the two components, the users only interact with the front-end, and the underlying cluster will be hidden (transparent) from the users. The front-end serves HTML files containing the needed JavaScript functionality to dynamically pull files from the back-end to update the content within each exercise. Figure 15 shows the data flow and logic behind the “detached” design.

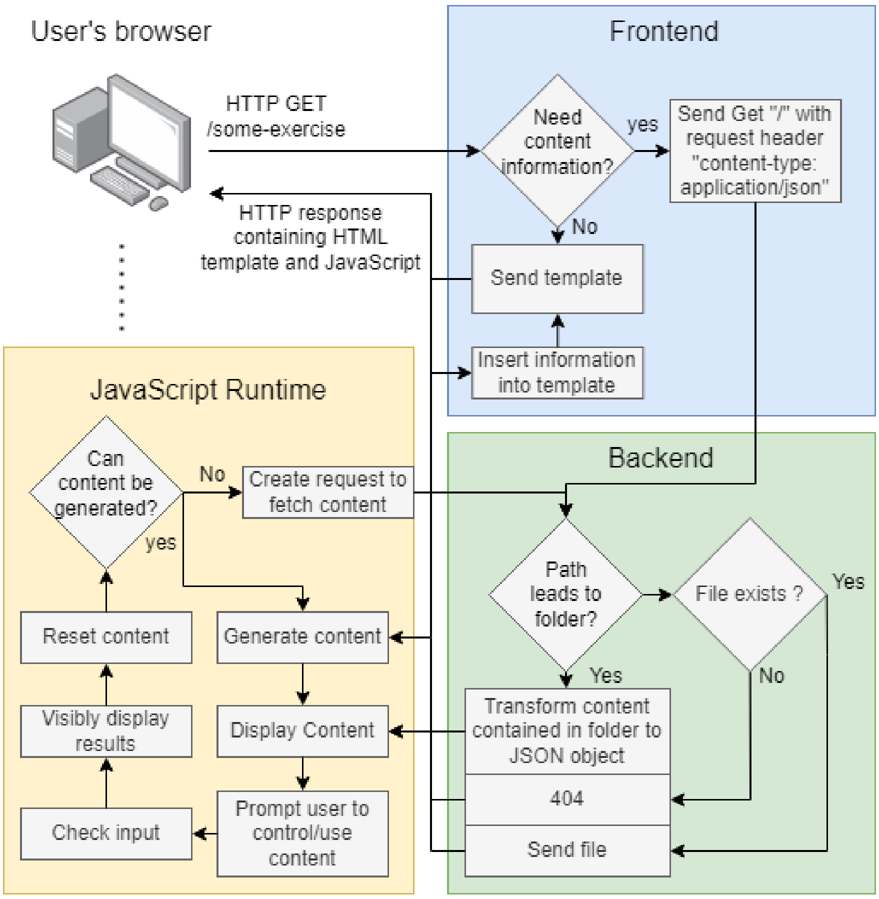


Figure 15: A flowchart describing the dataflow and main logic of the "detached" design

Figure 16 shows the architecture of the detached design, where the front and back-end is two separate servers.

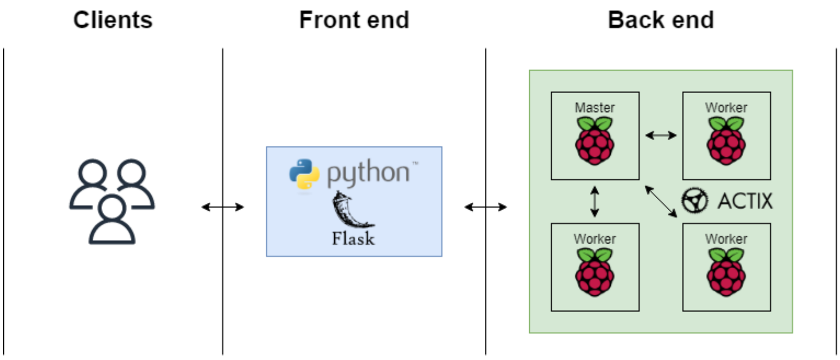


Figure 16: The architecture of the Single-board computer cluster.

The same design will be used within the Linode Kubernetes Engine (LKE) deployment, except that the front and back-end no longer need to be separated. This is because the underlying Kubernetes framework handles transparency, and separating the two components is no longer needed. Therefore, the design can be simplified by attaching the front and back-end, hence the "attached" design.

An LKE deployment consists of one or several nodes where each node has a container running a docker image. Inside the docker image is the "attached" design. It serves the same endpoints as the "detached" design and functions as a single node monolithic server. The design is equal to the "detached" design, with the only exception being that it comes as one singular complete "package" instead of being two separate entities. Kubernetes will duplicate the images across nodes, and each node acts as a replica. The Kubernetes framework handles load distribution, updates, and reliability. Therefore, any cluster functionality in the "detached" design is disabled within the attached design. Figure 17 shows the data flow and logic behind the attached design.

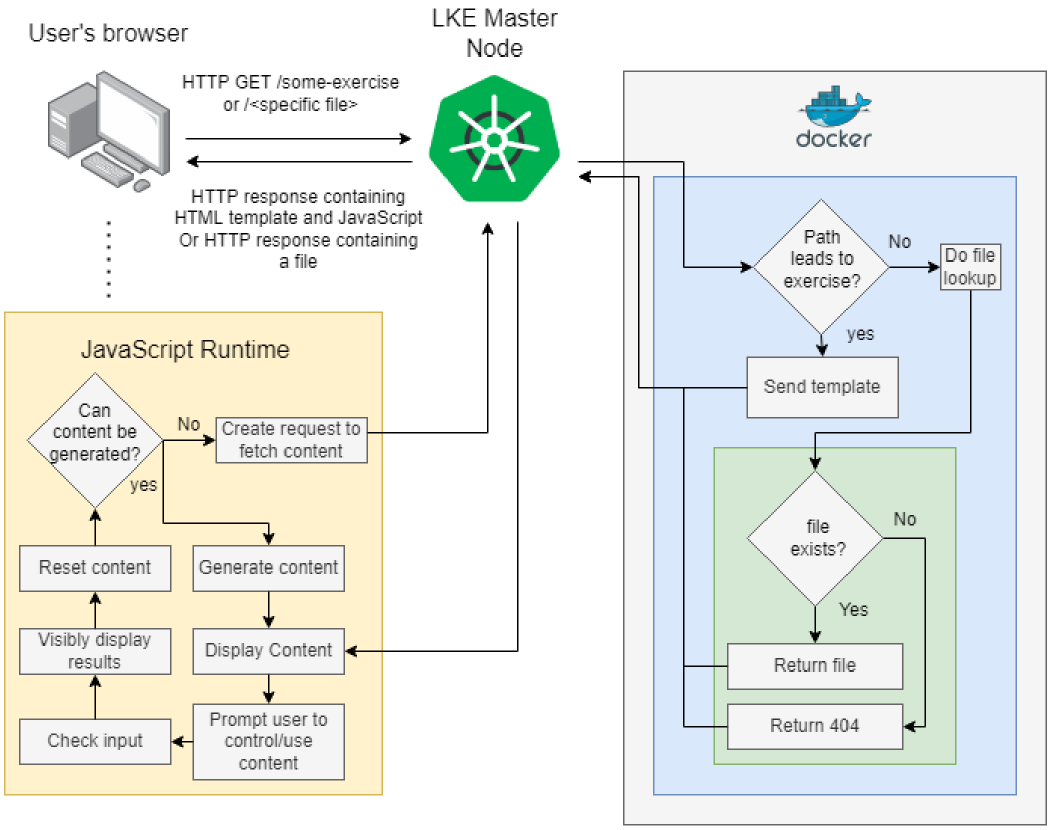


Figure 17: A flowchart describing the dataflow and main logic of the "attached" design

Figure 18 shows the LKE architecture, where the front and back-end are contained within the same server and functionally equal to the "detached" design.

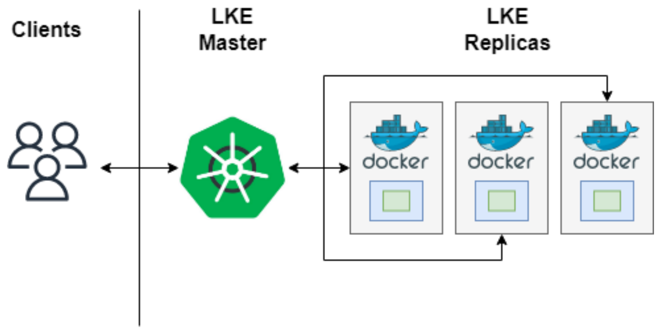


Figure 18: The architecture of the Linode Kubernetes Engine cluster.

When comparing the LKE and single-board computer cluster architecture, both handle transparency similarly. An intermediary node is placed in between the cluster and the client. In the LKE architecture, the intermediary node is the LKE master, while within the RPi cluster, the intermediary node is the front-end server. The difference between the two is how the data are routed. Since the single-board computer cluster detaches the front-end from the cluster due to the redirections, the front-end will receive fewer requests, and no traffic is piped through the master node. In contrast, the LKE master node sees all incoming traffic as it is directed through the master node. Seeing how the two different architectures affected performance was vital in determining which solution was better and what design was chosen as the final deployment platform.

5 Implementation

5.1 Overview

The following sub-chapters describe the steps taken to implement the design stated in chapter 4. First, the back-end is described in chapter 5.2. Then, after outlining the steps taken to implement the front-end in chapter 5.3, the different deployment methods used throughout the thesis are explained in chapter 5.4.

As described in chapter 4.4, the “attached” and “detached” designs have two functionally similar designs. However, implementing the “attached” design was considerably more straightforward. The Linode Kubernetes Engine (LKE) provides the required cluster functionalities such as load-balancing, adding and removing files, failure handling, and fault detection. Therefore, the implementation chapter focuses on the “detached” design, and the most significant differences between the two designs are briefly described.

In the “attached” design, the storage manager and the front-end have been merged to provide one server capable of serving the static HTML files without contacting an external source (the back-end). In contrast to the “detached” design, where the front and back-end are separated, the front-end must contact the back-end to know what content is stored within the storage manager. The storage manager is implemented similarly for both designs and can look up files and share them with users. However, the implementation does differ somewhat from the “detached” design. Since there is no need for a “simplistic” front-end, as depicted in Figure 19 (chapter 5.2), the folder flattening process described in 5.2.2 is removed.

Furthermore, the “attached” design will only respond with byte-streams of specific files. Therefore, it will not handle incomplete paths leading to folders. It is unnecessary as the “attached” design already has the information about folders through the underlying file structure. Finally, since the “attached” design is one single server holding all of the functionality needed, the “attached” design was written entirely in Python [36].

Chapter 5.3 outlines the steps taken to implement the front-end’s design. In contrast to the back-end, there is no difference between the “attached” and “detached” implementation.

Chapter 5.4 describes the different deployment methods used throughout the thesis. There have been three deployments: One for testing the single-board computer cluster (detached design) and another for testing the LKE cluster (attached design). A third deployment where a server was borrowed from UiT was utilized as a deployment for user testing (detached design).

5.2 Back-end

The back-end server is written in Rust [37] and utilizes the Actix [92] server framework. It is reachable through a browser as it has a simplistic front-end displaying its available content in a hierarchical structure using the templating engine Tera [93], as shown in Figure 19.

Figure 19: A screenshot of the fileserver's simplistic front-end.

Tera is inspired by Jinja2 [94] and functions similar to the python library. The fileserver has an admin panel hidden behind a login page for scaling, adding, and removing files. The back-end functions as a RESTful API and serves the following endpoints:

- HTTP(S) GET /<path>
  - Initiates a file lookup.
- HTTP(S) GET /login
  - A Login form that sends the form data to /auth when submitted.
- HTTP(S) GET /admin
  - Redirects to /login if not authenticated.
  - If authenticated, it leads to an administrator page displaying information about the cluster and the option to add or remove files and nodes.
- HTTP(S) GET /rhyme
  - To find rhyming words.
- HTTP(S) GET /ping
  - To check for node liveness.
- HTTP(S) POST /auth (form-data)
  - Validates credentials from the login form.
- HTTP(S) POST /fileupload (multipart form-data)
  - For adding files. Must be authorized.
- HTTP(S) POST /add (IP-Address, Port)
  - For adding nodes, scaling. Must be authorized.
- HTTP(S) POST /Update (indexing structure)
  - Part of the scaling process. Must be authorized.
- HTTP(S) DELETE /removefile/<path>
  - The inverse process of the file-upload endpoint.
- HTTP(S) DELETE /remove/<ip and port>
  - The inverse process of the /add endpoint

5.2.1 Data structure – The file index

The Indexing structure is a data structure containing nested maps. Each map represents the content contained within a folder, and the content can be folders or files. The resulting data structure is a structure containing file and folder names linked to their corresponding paths, as shown in Figure 20.

```
struct Folder {
   String: name,
   String: path,
   Map<String, Folder> : subfolders,
   Map<String, File> : files,
}
struct File {
   String: name,
   String: path,
}
```

Figure 20: Data structure of the file index.

When launching a server, it will recursively go through all folders and files within its root folder. The paths and names of files and folders are mapped into the corresponding place in the data structure. The maps maintain the original hierarchical structure of the underlying filesystem. The root folder's absolute path is defined in a configuration file and is easily changeable.

5.2.2 Resolving paths

When a worker node receives an HTTP GET /<key/file.mp3>, it will resolve the path. A path is resolved key by key. Each key within the path is checked to determine whether it corresponds to a file or folder present within the server's file index. Therefore, the server will start with the leftmost key and recursively go deeper, making sure each key within the path exists within the file index.

Files

Consider this path leading to the file wind.mp3, /root/environmental/wind.mp3. First, we check if there is something called "root" within the index structure subfolders field. *index.subfolders["root"]* exist and contain a folder structure as depicted in Figure 20 with the name root. The next key is "environmental" and is validated by looking through the Folder structure from the previous lookup process. root.subfolders["environmental"], also exists. The last part contains a period (".") in the key, indicating it is a file, which means that it is the last key, and there should be a file called wind.mp3 within the map called "Files." Environmental.Files["wind.mp3"] exist and contain a file structure, as depicted in Figure 20. The file structure contains the path "/root/environmental/wind.mp3" and the corresponding name "wind.mp3".

The path has been successfully resolved, and the system can read the file before sending it as a byte stream. If a key should fail to be resolved or is non-existent, a 404 HTTP status message will be sent as a response. The file lookup process is described with pseudo-code in Figure 21.

```
resolve_path(indexing_structure: Folder, path: String) {
	If (path == "/") {
		return indexing_structure
	}
	else {
		keys = path.split("/")
		result = try_get(indexing_structure, keys)
		return result
	}
}
try_get(indexing_structure: Folder, keys: Array){
	if( keys.len() == 0 ){
		return indexing_structure
	}
	else {
		new_index = try_get(indexing_structure.subfolders, keys[0])
		if(new_index == Some(result)){
			keys.pop(0)
			return try_get(new_index, keys)
		}
		else {
			new_file = try_get(indexing_structure.files, keys[0])
			if(new_file == Some(result){
				return new_file
			}
			else {
				return error_msg(404)
			}
		}
	}
}
```

Figure 21: Pseudocode describing the lookup process

Folders

Folders lookups follow the same process, with some exceptions in the response. There are two types of responses for folders. Either as a JSON object describing the folder's content or a representation of the "flattened" folder.

If the content-type request header is set to "application/json" the result from resolving a path will be serialized into a JSON object using Serde and sent as a response. "Serde is a framework for serializing and deserializing Rust data structures efficiently and generically" [95].

If the request header content-type is not set or different from "application/json", the server will assume you are trying to access the server's simplistic front-end. Therefore, the server will "flatten" the resolved folder, removing all information about data not directly within the specified directory. For example, suppose we have this file structure as shown in Figure 22.

```
root
….file0
….folder1
……..file1
……..file2
……..folder2
………...file3
```

Figure 22: Showing a folder structure.

The response from the folder “flattening” process after resolving the path /root/folder1 would look like Figure 23. Content contained deeper than the folder1 directive is removed from the response. The returned content will be the content contained directly residing within the folder1 directive: file1, file2, and folder2, but no information about file0 and file3 will be contained within the response.

```
{
        Name: “folder1”,
        Path: “/root/folder1”,
        Files: {“file1” : ”/root/folder1/file1”, “file2” : “/root/folder1/file2”},
        Folders: {“folder2” :  “/root/folder1/folder2”},
}
```

Figure 23: The result after flattening folder 1, depicted in Figure 22.

Firstly, the “flattened” response enables clients to request content that is present within certain directives while excluding deeper folders and files. Secondly, it simplifies inserting folder content into HTML templates utilizing the Tera templating engine. The templating engine can iterate over the contents of both maps, inserting file names as list items and creating event functions that will link further into the indexing structure or directly to specific files. The resulting HTML response will look like Figure 24 in a browser.

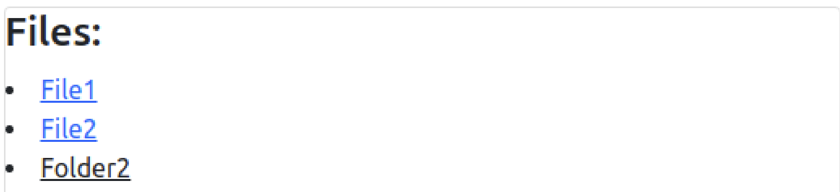


Figure 24: Shows how a flattened folder is displayed within the simplistic front-end. Folder In black and files in blue.

5.2.3 Load balancing

The distribution of load is done in a round-robin, turn-based schema. Each incoming request increments an index counter, and the index is used to determine which node should be responsible for handling the incoming request. Each time the index equals the number of worker nodes present within the cluster, the index is reset, and the master node will handle the request instead of delegating it.

In addition to this, there is an option to set a master's offset so that the master will only handle requests after delegating to the worker nodes *x* number of times, relating to the offset value. For example, if the offset value is equal to one in a cluster of three worker nodes. Then, the master node will be responsible for handling every fourth request. If offset is equal to two, the master will handle every eighth request. Figure 25 shows the logic behind the load balancing and how the master utilizes an offset value to know whether it should delegate or do the work itself.

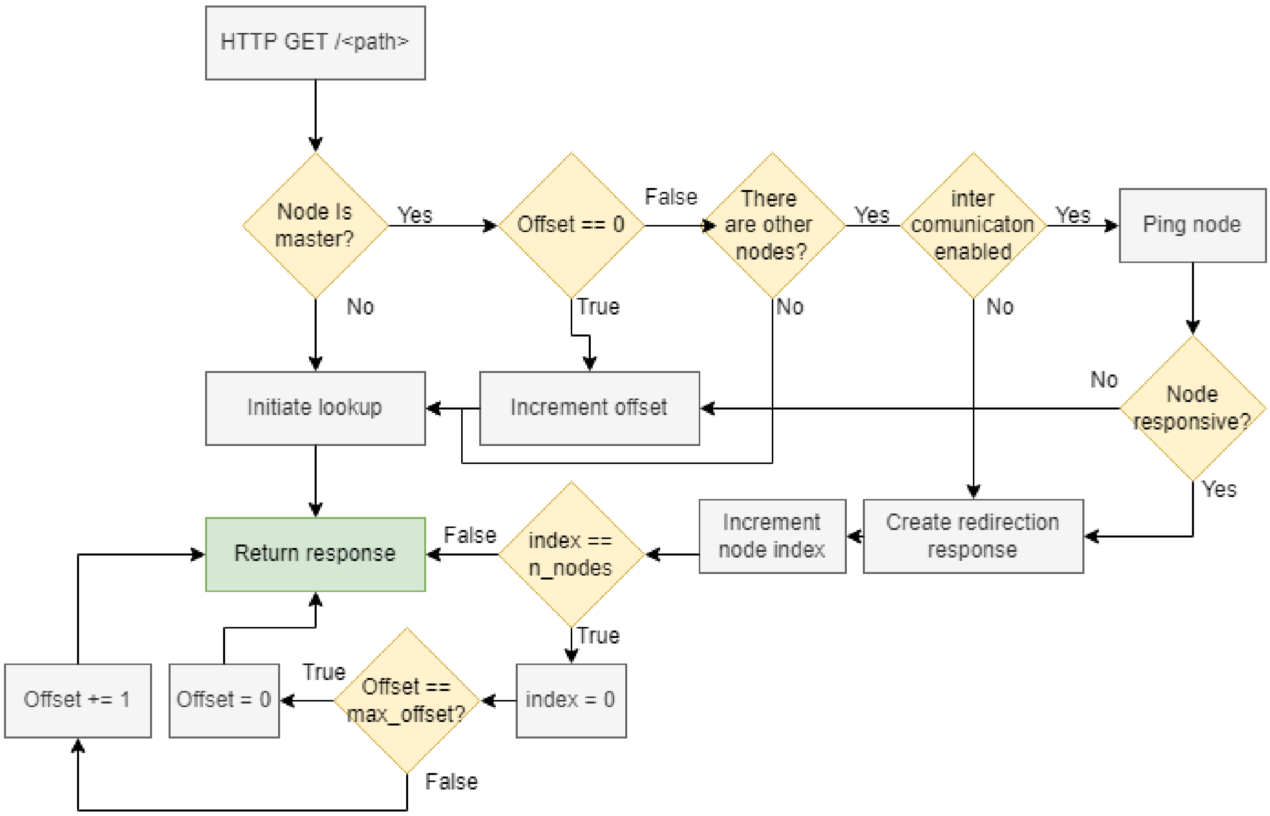


Figure 25: A flow-chart describing the load balancing process from the masters perspective

5.2.4 Scaling

Scaling is done by adding additional single-board computers to the cluster. Scaling was implemented by first adding an admin panel for the master node where a new node's contact information could be added, as seen in Figure 26.

The system administrator must log in via the login form to reach the admin panel. The login process is protected by utilizing the Secure Sockets Layer (SSL) [96]. SSL encrypts data sent between the client and browser. In addition, the credentials are validated against information stored as environment variables on the server-side. If the credentials are correct, the client will receive an identification token. The token is based on the JSON Web Token(JWT) Rust crate [97]. The token's validity is verified for each request, and the user will be redirected to the index page if not valid or expired.

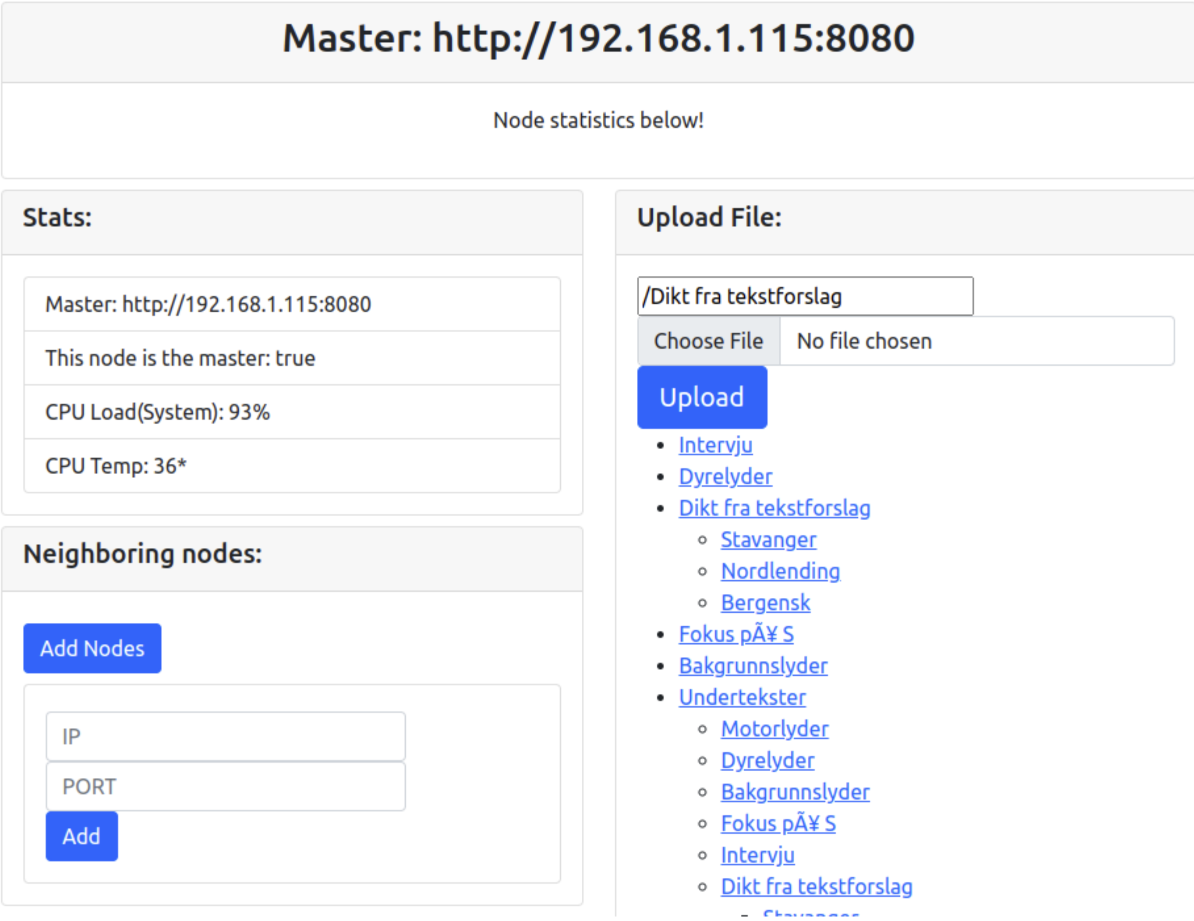


Figure 26: A screenshot from the fileserver's admin panel.

There is an “add node” input field within the admin panel (lower left Figure 26). The scaling process is started by adding the new node’s contact information into the input fields and submitting the form. An HTTP POST /add request is sent to the master node when submitting the form. Next, the master node checks the correctness of the IP address utilizing a regex (regular expression) [98] and checks that the port number is inside the ephemeral port range (between 1024-65535). If the information submitted contains a valid IP address and port number, an HTTP POST request is sent from the master to the new node containing the master node’s file index. The new node will receive the indexing structure and duplicate it to be its own before starting the next phase. At this point, the main task of the master node is done, and the new node will go through the replicated index to pull the master's content from the cluster. From the perspective of the master, the requests coming from the new node to download all available files look just as if they were coming from regular clients and will be handled as normal requests.

Before responding to the duplication POST request, the new node spawns a new thread that will iteratively go through each file contained within the newly duplicated file index. First, each file is fetched from the pre-existing cluster and written to the corresponding paths listed within the indexing structure. When all files have been downloaded and written to storage, the new node will notify the master that it is done duplicating content. Then, the master sets the new node as ready, indicating it is a part of the cluster before notifying all worker nodes of its arrival. The other worker nodes will then add the new node to their list of neighbors. Figure 27 describes the scaling process in a flowchart.

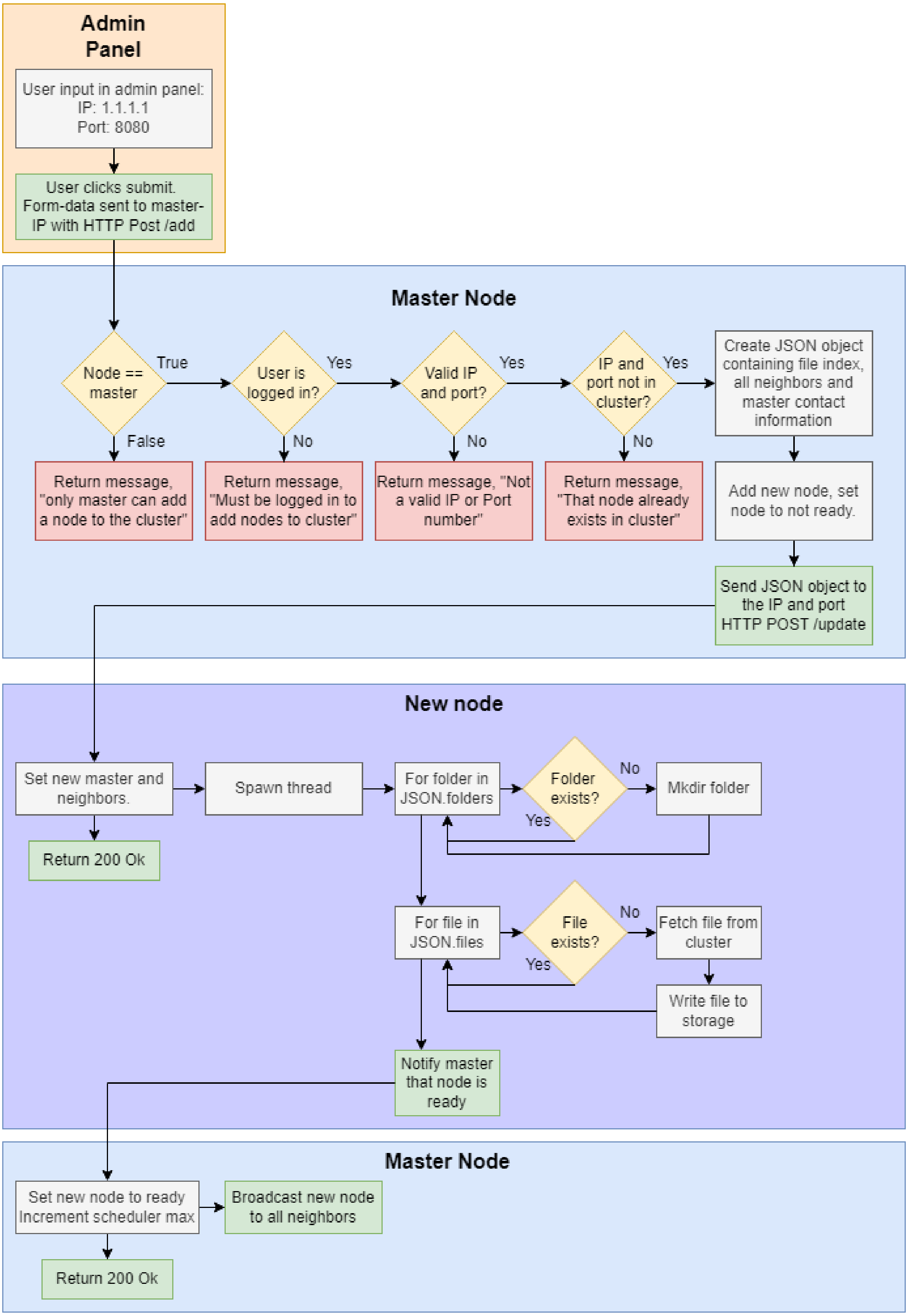


Figure 27: A flowchart describing the scaling process

5.2.5 Adding and removing files

Adding a new file is done through the master node's admin panel. If you look at the right side of Figure 26, there is an upload file field. Additionally, all folders within the fileservers root folder are displayed underneath the Upload button. By clicking one folder, you specify that the file chosen for upload is to be written to that folder's directive in the underlying filesystem. Changing the path is done via the on-click event functions present for each list item.

When the system administrators click the "choose file" button, they are prompted to search through the filesystem of their computer to choose a file to upload to the cluster. A multipart form[99] is sent to the cluster's master node with an HTTP POST /fileupload message when the administrator clicks the upload button.

When received, the multipart form is split into specific parts. First, the path and filename are found. Then, the final part of the multipart form is the byte-stream, which is written to the path with the filename found earlier. Then, if the writing was successful, the file is added to the fileserver's file index. Next, the master node will broadcast the same file upload message to all worker nodes, which will go through the same process. Figure 27 describes the file upload process in a flowchart.

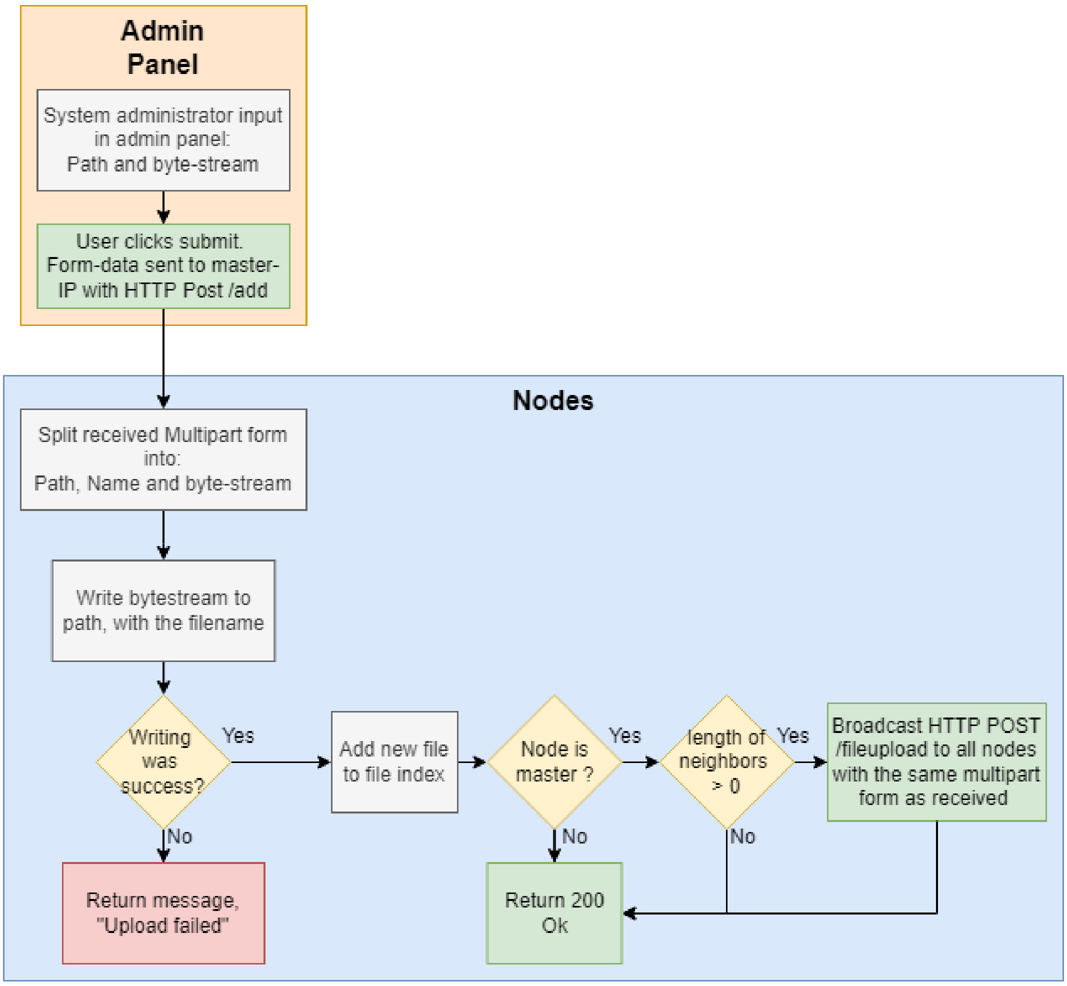


Figure 28: A flowchart describing the process of adding files to the fileserver.

Removing files is done as the inverse process of Figure 28. When logged in as an administrator on any node(worker or master node), you get an additional option of removing files. The delete button only appears if logged in as an administrator, and the removal buttons can be seen in Figure 29.

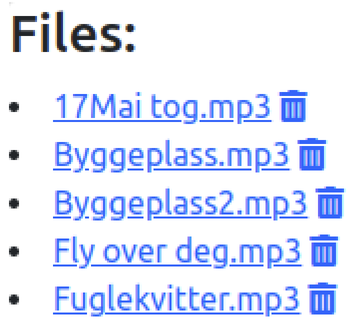


Figure 29: Screenshot showing an administrator's view while browsing files on the fileserver's "simplistic" front-end.

An HTTP GET /removefile<path> is sent to the same node as the administrator is logged in to when clicking the remove buttons. Again, the request is sent by utilizing on-click event functions. When a node receives a removal request, the specific file contained within the request is removed from the underlying filesystem and the fileservers file index. If the node is a worker node, the same removal HTTP GET message will be sent to the master, which will go through the same removal process but ends with broadcasting the message to the other nodes. Figure 30 describes the removal process as a flowchart.

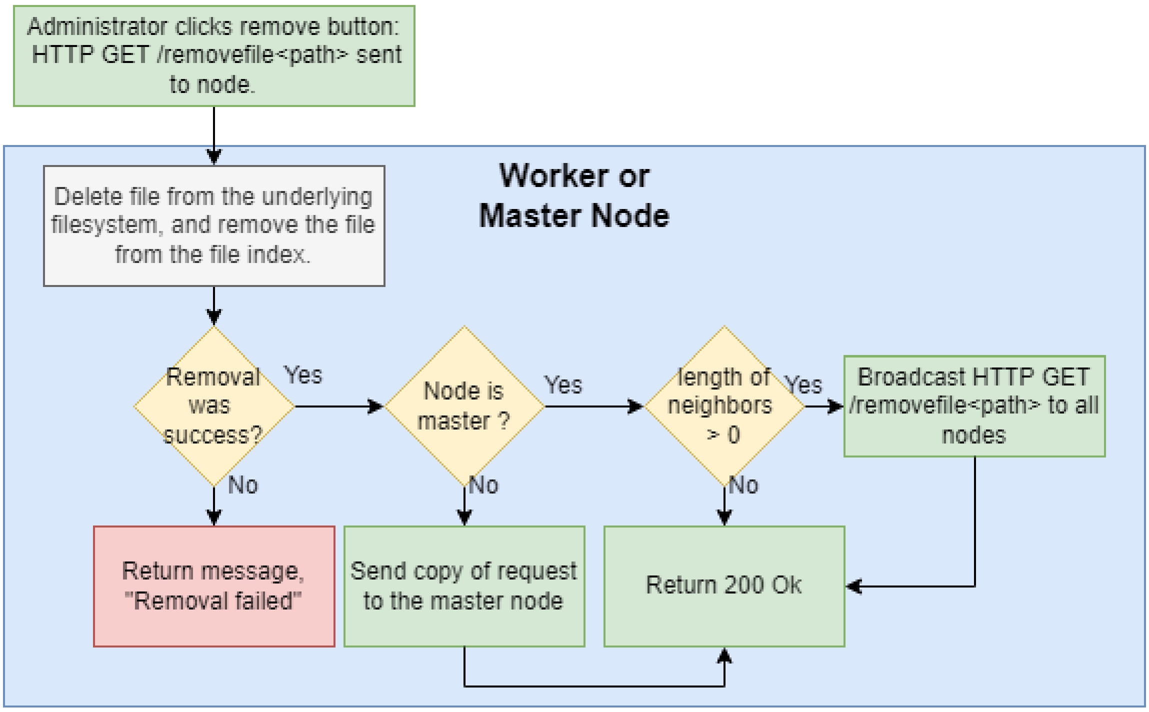


Figure 30: A flowchart describing the process of removing files

5.2.6 List of Norwegian words

The list of Norwegian words was created by accumulating a large body of Norwegian text later split into singular words and inserted into a list. First, the text was gathered by using

Norwegian text collections found online [99], [100], [101]. Next, the text was pruned for special characters, numbers, and unwanted words like slurs or profanities programmatically.

Each line within the document was split into singular words by separating lines of text on spaces and inserted into an array for storage. Then, each word was validated one by one by querying an online dictionary [102]. If the words were confirmed to be Norwegian and grammatically correct, they were written into a final document containing the list of words. After sorting the result alphabetically, the list was extensive enough to get sufficient results from the naive rhyming algorithm.

Even though unwanted words were dynamically pruned in the starting phase, an additional manual inspection of the words was performed. Some of the text collections used were old and contained outdated or Danish-looking words. Some of these words would go through the online dictionary even though they are not used within the Norwegian language. The manual inspection took some time but was worth it. Being confident that there are no unwanted words within the list is essential. The manual inspections were also a one-time process and will only be done again if there is a need to extend the list to include more words.

Each word within the list was converted to audio files utilizing the Google text-to-speech [91] API. Converting the words from textual terms into audio was done by running a Python script that takes words as input and creates an API call before handing it over to Google. The response is an audio file with the voicing of the requested term. The API is not free-to-use, but Google's pricing is free for the first million characters each month. By downloading all words within the list and storing them locally within the back-end, the server avoids additional costs from using the API and potential bottlenecks from contacting the Google text-to-speech API during runtime. Each audio file is a few KB in size, and storing them was not an issue.

5.2.7 Rhyming algorithm

The rhyming algorithm is straightforward. It compares the ending of two terms and decides if they rhyme based on the last letters being equal. If they are equal, the specific word and the number of equal letters will be inserted as a tuple into an array. Next, the collected tuples are sorted from high to low based on the number of equal letters. Finally, the five most similar words, the word used for input, and the path needed to reach the voiced audio file are returned as a single JSON response. The rhyming algorithm works decently, despite limitations concerning the phonetic similarity. Figure 31 shows the rhyming algorithms in pseudo-code.

```
Rhymefunction(String: inputword){
    for word in listofwords {
        count = 0
        for character in reverse(word) {
            index = inputword.len() - count
            if character == inputword[index]{
                count ++
            }
            else {
                break
            }
        }
        Scores.push((word, count))
    }
    return sorted_array(scores)
}
```

Figure 31: Pseudocode describing the rhyming algorithm

5.3 Front-End

The front-end server is written in the Python [36] programming language and utilizes the Flask [86] server framework. Flask utilizes Jinja2 [94] as a templating engine to dynamically change or insert HTML content within templates. The front-end functions as a RESTful API serving the following endpoints.

- HTTP GET /Index: landing page displaying the different training modes available.
- HTTP GET /Player: synthetic exercises.
- HTTP GET /Tone-separate: tonal separation, musical exercise.
- HTTP GET /Word-separate: phoneme exercise.

Each exercise has a JavaScript singleton Class initialized at runtime within the user's browser. During training, results and valuable information will be stored, tallied, and displayed to the users as performance metrics.

5.3.1 Implementing a responsive web design

A responsive web design was implemented by utilizing different scalings for the HTML content depending on the user's screen size, as seen in Figure 32. Cascading style sheets [103] have media queries that take expressions as input. For the responsive web design, the "max-with" expression will apply styles only if the browser's viewport width is equal to or narrower than the expression.

```
@media (max-width: 1500px) {
    .object {
        Margin-left: 10rem;
    }
}
@media (max-width: 1000px) {
    .object {
        Margin-left: 5rem;
    }
}
```

Figure 32: Showing how Cascading Style Sheets were used to implement a responsive design

5.3.2 The Audio player and its functionalities

When the front-end receives an HTTP GET /player request, it will request the available content from the back-end and display the response to the user in a viewable manner. First, the front-end creates an HTTP GET "/" with "content-type: application/json", before sending it to the back-end utilizing the python library Requests [104]. Then, the back-end receives the HTTP request and initiates a lookup process on the root folder indicated by the backslash ("/"). It will return a JSON object describing the contents available within the root folder. The returned JSON object contains all available content stored within the back-end and the URLs needed to reach each specific audio file.

The front-end receives and transforms the JSON object into a bootstrap [105] drop-down list. The structure of the dropdown list resembles the hierarchical folder structure of the underlying file system within the back-end. The transformation is done to enable users to navigate the available content easily. It should be as easy to navigate as the original folder structure of the back-end. After transformation, the HTML content is inserted into an HTML template using Jinja2 [94] before serving the template to the user who requested it. The data flow and logic are seen in Figure 33.

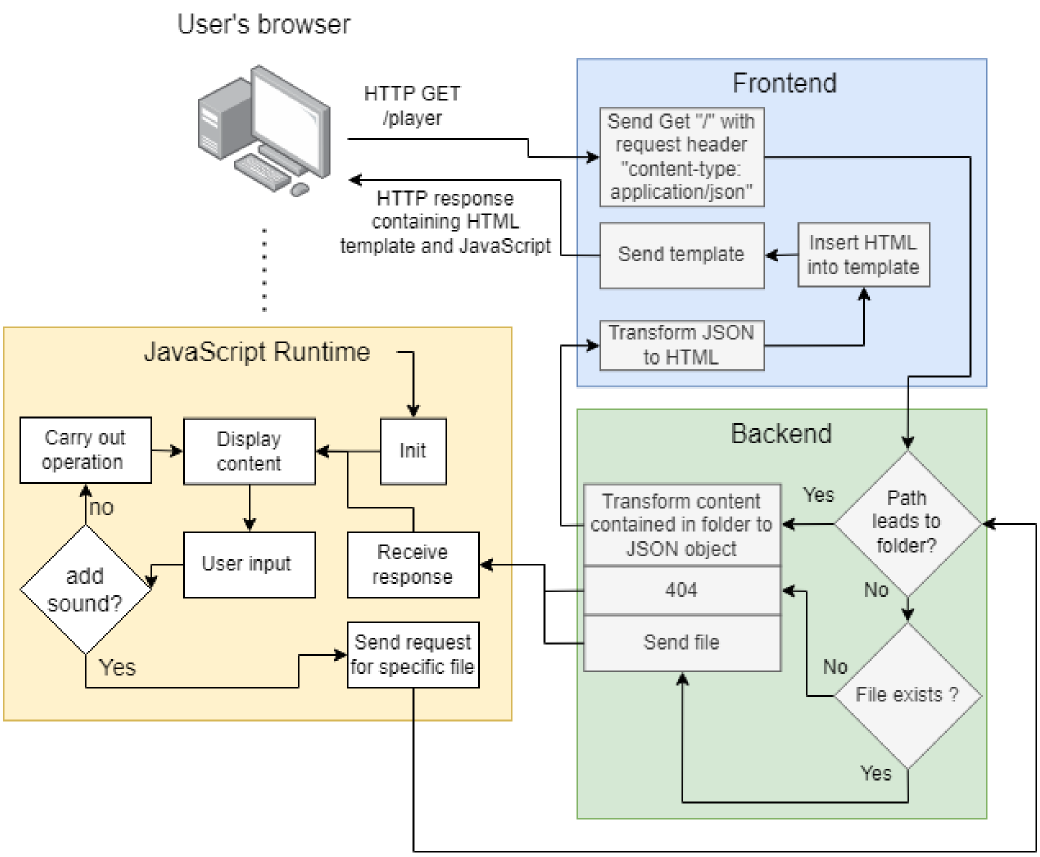


Figure 33: A flowchart describing how the sound picture exercise functions

Adding audio files

There are four video players available, which the users can insert audio sources and play simultaneously. Each player has its unique id, corresponding video source, and track tag for subtitles. The video tags HTML code is shown in Figure 34.

```
<video id=”audio1” crossorigin=”anonymous” style=”display: none;” playsinline>
        <source id=”audiourl1” src=”<audio source>” />
        <track id=”subtitle1” src=”<subtitle source>” default />
</video>
```

Figure 34: HTML code showing how the video tags were implemented

The video and subtitles source is dynamically inserted into the first available video player source tags. Inserting audio happens when the user clicks specific list items within the drop-down list. Each list item gets an on-click event function while transforming the JSON object received from the back-end. The event function takes two URLs and a name as input. The two URLs point to audio and subtitle files.

Subtitle paths are not contained within the JSON object received from the back-end. Instead, the subtitle paths are derived from the audio file paths. Subtitles have the same path as the audio files except for the root folder and file extension. For example, imagine a file called keyboard.mp3, stored at the relative path of “/Instruments/keyboard.mp3”. The corresponding subtitle file will be stored at “/subtitles/Instruments/keyboard.vtt”. The folder placement and static file extensions make converting audio file paths to their corresponding subtitle path easy and are done during the transformation of the JSON object. Furthermore, deriving the subtitle paths from audio file paths enables excluding subtitle files within the drop-down list as they are not present within the JSON response.

The on-click event function will insert the audio URL into the specific player's source tag (line number 2, Figure 21). Next, a new HTML track tag (line number 3, Figure 34) with the subtitle URL as the source is created and inserted into the video tag. After inserting the sources, the underlying HTML5 architecture will automatically load the content so they are ready to be played. Loading audio from the inserted sources happens as long as the cross-origin parameter is set within the video tag to avoid problems relating to the CORS [106] mechanism. “CORS is an HTTP-header-based mechanism that allows a server to indicate any origins(domain, scheme, or port) other than its own from which a browser should permit loading resources. “ [106].

The tracks “default” parameter (line number 3, Figure 34) is set to ensure that subtitle tracks will always be shown within the video element. The subtitle buttons toggle specific video elements on or off by changing the video tags cascading style sheet (CSS) display value. If the display value is “none,” the video element will be hidden, and the subtitles cannot be seen. Vice versa, if the display value is equal to “block,” the video element will be displayed, and the subtitles will be made visible.

The waveform data from the audio files are displayed using the JavaScript library wavesurfer [89]. It is easy to use and functions as an additional audio player. When the source is set and loaded, the wavesurfer library will automatically generate and display the waveform data through an HTML canvas element. The wavesurfer source is set and loaded within the same event function described earlier, and the wavesurfer library will automatically load the data. Figure 35 shows how one of the four audio players looks after adding audio, subtitles, waveform data, and filename.

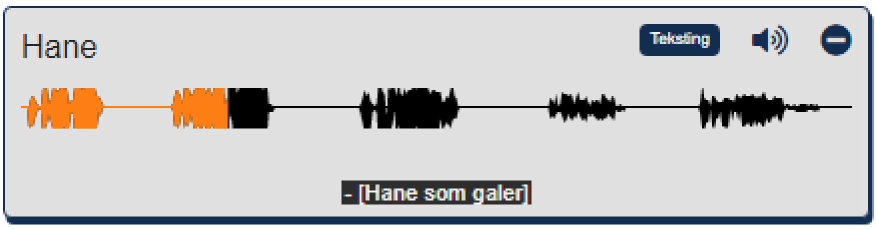


Figure 35: Showing one of the four audio players within the sound picture exercise after adding audio, subtitles, waveform data, and filename.

When adding audio files, the color of the list item corresponding to the audio file is changed to green. Additionally, the icon displaying that it is an audio file (a musical note) will change to a “remove” button to decrease the amount of navigation that is needed to add and remove audio. For example, Figure 36 shows the color of the list item “Sommer” has been changed to green after adding it to the player.

The audio files are removed from the player by resetting the video tag source. The underlying HTML5 architecture will handle the job of resetting the audio itself. Subtitles are removed by setting the default tag to disabled, which will make the subtitles invisible. The track will still be present until remade by adding a new audio file into the same player. When adding new subtitles, the subtitles track must be destroyed and remade to flush the subtitle queue to avoid lingering subtitles from previous tracks.

Figure 36: Showing how the color and icon change as audio files are added.

Volume control

The volume attribute of each video tag can be modified by interacting with input sliders that appear when clicking the volume icon. The corresponding video tags volume attribute is also changed whenever the input sliders are changed. The volume control feature is removed from smart devices as it took too much space and was quite tricky to use on smaller screen sizes.

Progress bar and timings

As there are multiple audio files, the progress bar and maximum duration are chosen to correspond to the duration of the longest audio file. When users interact with the progress bar by clicking and dragging the thumb button, the current time attribute for each audio file is changed. If the current time attribute of a specific player is bigger than the audio file's duration, the audio from that file is paused. Users can change this by interacting with the settings section and setting the loop audio option to true. Suppose the looping option is set to true. In that case, the current time for each file will be relative to the audio file with the longest duration, calculated by the remainder of the specific audio file's total duration. The player will always display the duration of the longest audio file. The duration change is seen in Figure 37.

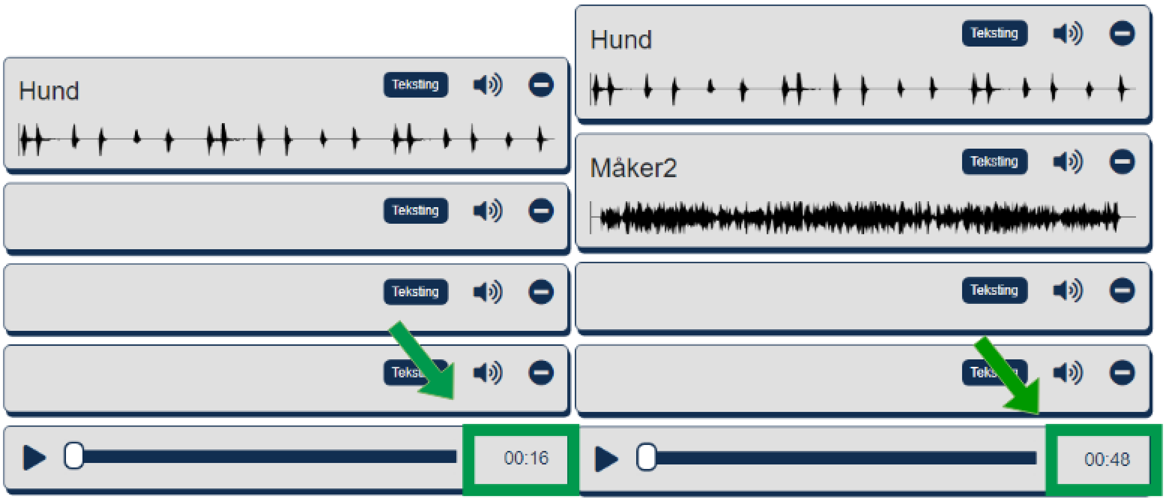


Figure 37: Showing how the duration change as files are added to the different audio players (HTML video elements).

Play and pause

As users push the play button, all video tags which contain a set audio source will be set to a playing state along with their corresponding wavesurfer.js canvas. The state is only changed if the audio source is set and enough data has been loaded. When the video tags are playing, the play icon is switched to a pause icon, and the on-click function is changed to be a pause function. When the player pushes the pause button, all sounds will be set to a paused state. The icon change is shown in Figure 38.

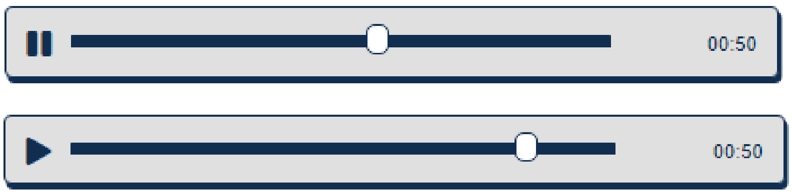


Figure 38: The Progress bar in a playing and paused state

Dynamically generating sinewaves

The generation of sinewaves used within the tone separation exercise utilizes the AudioContext [107] JavaScript interface. The AudioContext interface is a way to process audio programmatically, and It does so by using another JavaScript interface called AudioNode [108]. These interfaces enable the system to dynamically generate sinewaves with a predetermined frequency, which can be displayed to the users within the tone separation exercise. The audio generated is not inserted into HTML5 tags but rather played utilizing JavaScript.

After generating the sinewaves, the audio is presented to the user by clicking the play buttons. Next, the user chooses the audio's corresponding input buttons to select one of the two sinewaves. Finally, the input is checked for correctness, and the users are prompted with a message telling them if they were correct or not. Finally, the score is tallied, updated, and displayed to the user. The data flow and logic are shown in Figure 39.

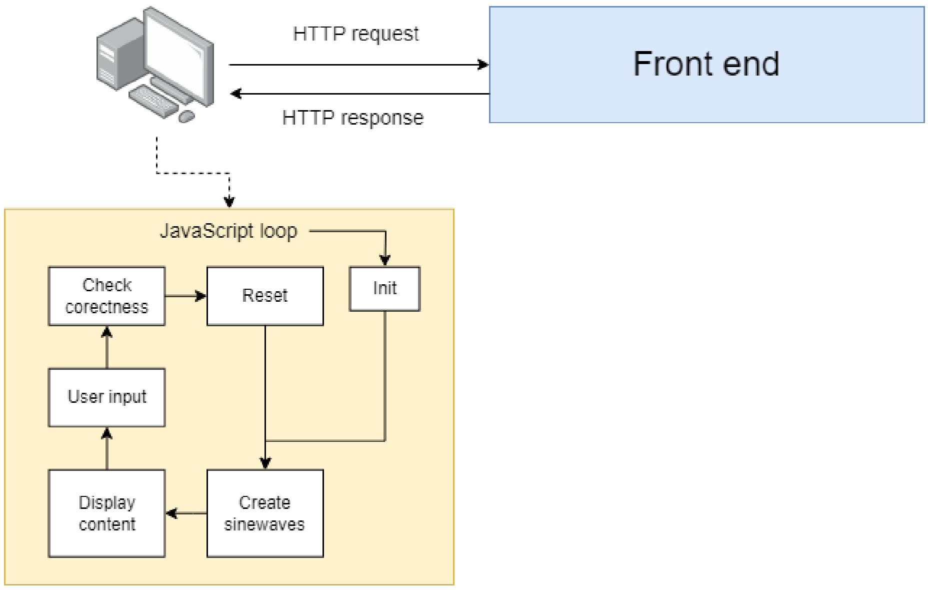


Figure 39: A flowchart describing the data flow and logic behind the tone separation exercise

5.3.3 Word separation

When users receive the HTML template sent from the front-end server, the JavaScript class contained within the template will go through an initialization function. The most important thing during the initialization process is that an HTTP GET /rhyme request is created and sent from the user's browser to the back-end. The JSON object received within the response is displayed to the users, who are prompted to choose amongst buttons similar to the spoken word. Finally, the key presses are checked for correctness, and a message is displayed to tell the user if the input was right or wrong. The data flow and logic are shown in Figure 40.

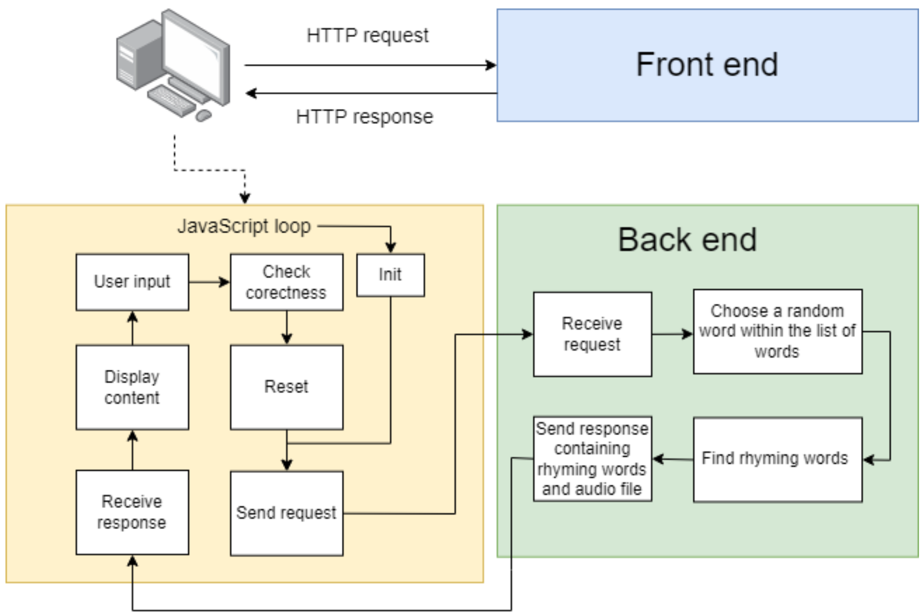


Figure 40: A flowchart describing the data flow and logic behind the word separation exercise

5.4 Deployment

5.4.1 User-testing deployment

The first step to user testing and getting feedback through semi-structured interviews was to make the digital training tool available for users. I was able to borrow a server from UiT, where I could deploy the front and back-end to allow access from the outside world.

The machine was a Dell PowerEdge R540, 2018 model with two Intel Xeon Silver 4114 CPUs. It had 64GB RAM with 2x 240GB SSD and 8x4TB HDD for storage. The server ran the Ubuntu 20.04 operating system.

The front and back-end were deployed by wrapping the “detached” design into Docker images and running them in detached mode [48]. The back-end was deployed as a single node monolithic server where all cluster functionalities were disabled. By borrowing the server, I did not have to use reverse proxy servers for the user testing, which would not have been a viable method for deployment as the lab computers reset themselves after a certain period.

The front-end docker image ran the flask server through Gunicorn [109]. Gunicorn is a Python WSGI (Web Server Gateway Interface) HTTP Server for Unix. It takes Python applications such as a flask server and enables the deployment of multiple flask servers working in unison to create a multithreaded web server.

5.4.2 Single-board computer deployment

The student housing where I live is connected to the internet via the same network as the University. The implication is that the Raspberry Pi (RPi) nodes are not reachable from the outside world without utilizing reverse proxy servers due to firewall issues. However, as I was able to borrow a server from UiT, I could route the traffic to my personal router's “public” internet protocol (IP) address by using reverse proxy servers. The traffic was routed through the borrowed server via a reverse proxy present within a lab computer, finally reaching my router. Figure 41 shows how the traffic was routed, utilizing Nginx [110] servers.

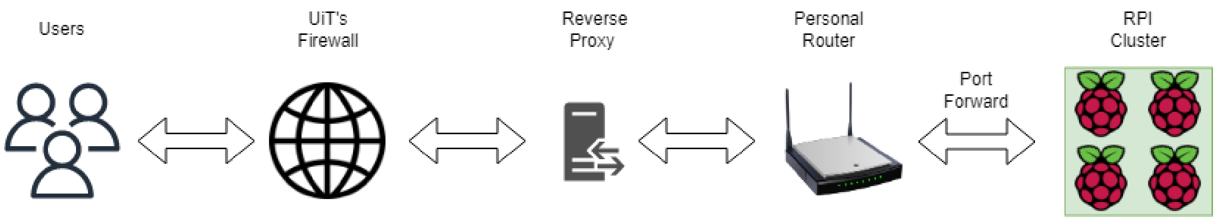


Figure 41: Showing how the requests were routed to reach the single-board computer cluster

Additionally, my personal router port-forwarded depending on what port number was pointed to by the requests to reach the local IP of each RPi node. Each RPi node ran a compiled binary file of the back-end server, listening for TCP traffic on different ports. There were four nodes within the cluster, one master distributing load to three worker nodes, simulating the same scenario as the deployed Linode Kubernetes Engine cluster. The testing was done by connecting to a different network outside my router. Each request was sent to the borrowed server's public IP address, which ran a reverse proxy server forwarding the request to the lab computer. The lab computer ran a reverse proxy server forwarding the request to my router, which would port forward to the cluster nodes. The port number was used to know which node a client was trying to reach. For example, port 8080 led to the cluster's master node. The master node would respond with a redirect to a new location containing a port pointing to a worker node.

5.4.3 Linode Kubernetes deployment

The Linode Kubernetes Engine (LKE) deployment consisted of three nodes plus the master node. The three worker nodes specification were 1VCPU, 2GB RAM, and 50GB of storage, costing 10 USD each (the monthly fee was 30 USD).

After Linode instantiated the cluster, the "kubeconfig" was downloaded, holding the security credentials needed to contact the LKE master node via kubectl (kube-controller) [26]. The number of replicas in the deployment was specified as three nodes, all listening on port 80 for TCP communication. Finally, the deployment image pointed to a docker hub account where an image containing the "attached" design was uploaded. The deployment configuration is seen in Figure 42.

```
apiVersion: apps/v1
kind: Deployment
metadata:
    name: lyttetrening-deployment
    labels: app: lyttetrening
spec:
    replicas: 3
    selector:
        matchLabels:
            app: lyttetrening
    template:
        metadata:
            labels:
                app: lyttetrening
        spec:
            containers:
            - name: lyttetrening
               image: <docker-hub-account/<img>:latest
    ports:
    - containerPort: 80
```

Figure 42: A Linode Kubernetes Engine deployment configuration file.

The deployment configuration is applied to the LKE by running "*$kubectl apply -f deployment.yaml*". Next, the load-balancing service was configured. Figure 43 shows the service configuration file.

```
apiVersion:
v1 kind: Service
metadata:
    name: lyttetrening-service
spec:
    selector:
        app: lyttetrening
    ports:
    - protocol: TCP
      port: 80
      targetPort: 80
type: LoadBalancer
```

Figure 43: A Linode Kubernetes Engine service configuration file.

After applying the service with *$kubectl apply -f service.yaml,* the service enabled the cluster to be reached from the outside world with a public IP, and Kubernetes will load-balance the incoming requests across the replicas. Moreover, since the specified numbers of replica nodes were set to three, Kubernetes will always make sure that there are three containers up and running, and if one should crash, it will automatically start a new one.

6 Tests and results

6.1 Literature review

Figure 44 shows the PRISMA [111] flow diagram describing the process of excluding records from the literature review. Table 9 lists the literature findings, which are summarized below.

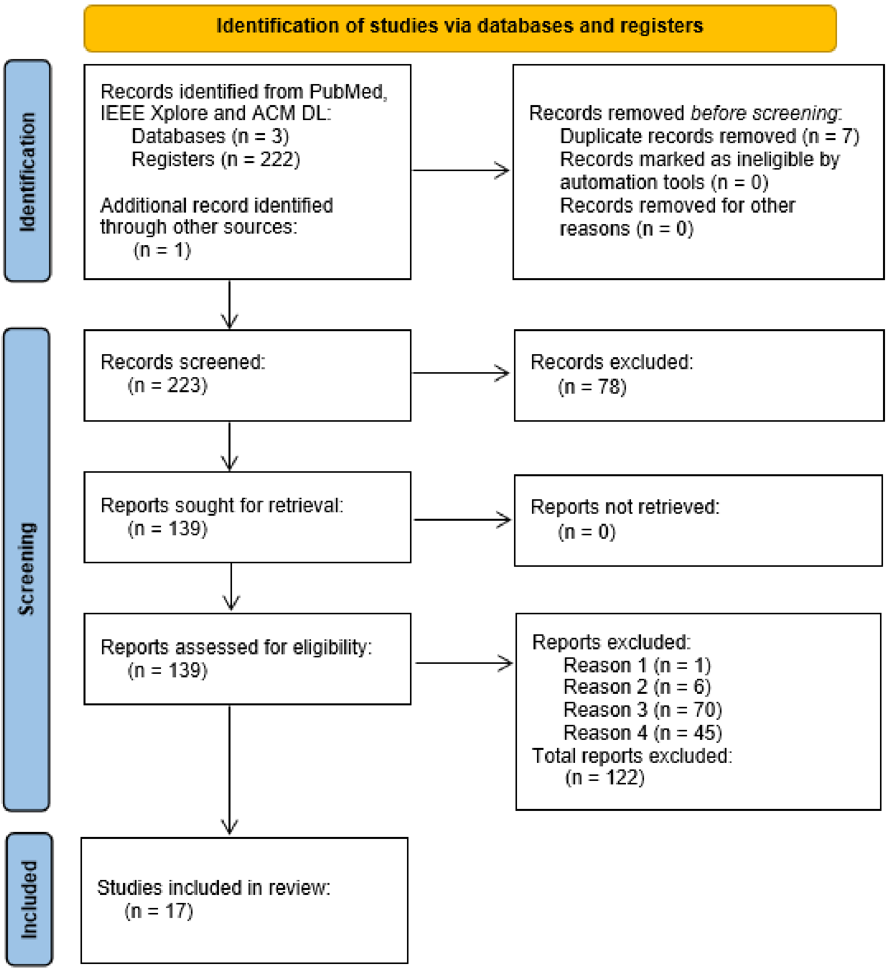


Figure 44: A PRISMA flow diagram showing the steps taken to eliminate records from the literature review.

Several papers within the literature have mentioned using environmental sounds or speech in babble (noise)s to train for speech or audio perception when there are multiple talkers or in scenarios with background noise [5], [61], [62].

Separating or segregating differences in frequencies has been pointed to by several papers within the literature review [63], [64], [65], [66], [67], [62], [68], [69]. Most of these papers are studies based on musical training techniques to improve musical, tonal, pitch, and speech perception. Out of all results reported within the literature review, the studies focusing on musical or tonal approaches towards audio training reported significantly better results than papers with other training approaches.

The method of trying to increase word perception differs from paper to paper. The traditional approach to audio training is focused on having a participant repeat or understand stated phrases, sentences, or words [70], [5], [71], [11].

Jiam et al. [65] saw training effects in the participant's time utilizing audiobooks as training content.

Research has been done to see the effects of utilizing contrast or nonsense words/syllables for audio training [72], [73], [18]. The common denominator for these articles is that they focus on similar words or sentences that participants must separate from each other. Additionally, substantial efforts are made to remove any possibility of participants guessing the correct answers.

Vandali et al. [66] strongly emphasized giving their participants the most information possible for the participants to make sense of what they were hearing. Therefore, they provided as many audio-visual queues as possible, positively affecting the training.

Table 9: The literature review result

Authors	Methods	Technology	Targeted skill sets	Pre and post-training assessments	Location and duration	Training Method	Training Results	Take away	Language
Borel et al. [70]	Retrospective study to investigate audio training over the phone	Training based on using telecommunication through phones. A specialist would call a patient and have training sessions over the phone.	-Monosyllabic word comprehension. -Phone usage.	The training program was presented with a 75-page booklet to assist patients. 4 assessments were done in the hospital/speech center before training started—additional assessments after each session and a final assessment at the end of the six weeks of training.	Remote training sessions assisted therapists in the hospital and patients at home or work—rehabilitation was conducted over six weeks.	The program consisted of sessions where a therapist said phrases and words over the phone, and the patient would follow along with a provided booklet as guidance.	There were Positive results in understanding sentences over the phone or in recordings. Patients reported improvement in phone use for day-to-day life. Self-confidence was reported to increase throughout the 6-week training program.	Audio training done over the phone yields positive results in speech comprehension and may give the self-confidence to do further training and more use of cell phones in patients' day-to-day life.	French
Green et al. [5]	Assessment of training method	Software developed as an application used with computer tablets.	-Sentence recognition in babble (noise). -Consonant identification. -Phoneme identification. -Forward and backward digit span.	Performance was assessed after each session and after four weeks with no training.	The training was performed at home or in the workplace. In total, there were twelve hours of training over four weeks.	Recordings of connected narratives were divided into phrases from 2 to 10 words, with up to 2600 phrases per text. The phrases were presented in babble (20 different talkers). After each phrase, the listener identified keywords from the phrase among similar-sounding words.	Sentence recognition in babble improved. Improvements were seen in participants' speech reception threshold (approximately 2 dB). In addition, there was a minor improvement in consonant and phoneme identification. However, forward and backward digit span did not affect training.	The authors concluded that a moderate amount of computer-based text training at home significantly improved sentence recognition in babble. Although 2 dB does not sound like much, they achieved this result after a relatively low amount of training.	British English
Sousa et al. [71]	Single Case study, finding challenging points within phone usage which could be alleviated with training	Telecommunication using landlines between two labs.	-Audio at different thresholds -Monosyllabic words -Disyllabic words -Sentence recognition -Nonsense syllables	Client Oriented Scale of Improvement (COSI) [112]	In the lab alongside a therapist. Eight sessions total.	The therapist verbally presented lists of words and sentences over the phone(landlines).	Performance on the telephone with a speech processor in the telecoil function was equal to or better than that obtained in a soundproof booth. Authors showed an improvement rate of disyllables (6.7%), monosyllables (20%),	Using the CI's speech processor and telecoil was better than using speakers.	Portuguese

							and a list of nonsense syllables (14%). and list of sentences (43%)		
Shafiro et al. [61]	Qualitative Study, researching the effect of a short computer-based training regiment	Computer-based software was developed to run on computers to display audio training content.	-Environmental sound perception	Two pre-testing assessments one week apart. Finally, two post-test sessions separated by a week without training concluded the sessions.	Four training sessions were conducted on separate days for one week.	The test sessions included an environmental sound test consisting of 40 familiar everyday sounds. Furthermore, the test sessions consisted of a consonant nucleus consonant word test and revised speech-perception in noise sentence test.	The authors concluded that a home-based computer training regiment could improve environmental sound perception, as it did in this short four-session training regime. The most significant improvements were with the sounds included within the training set. The research underlines the need for an effective, low-cost approach for home-based systems that seek rehabilitation for CI users and other hearing-impaired populations.	Environmental sound perception could be improved by home-based training. Additionally, this could provide an effective and low-cost solution for audio training.	N/A
Sato et al. [74]	A qualitative study investigating the feasibility of in-home audio training.	A computer tablet with a prototype application was developed for the study.	-Speech intelligibility -Monosyllables	Assessments were performed 1, 2, and 3 months after training. The test contained 40 untrained words.	Training is done at home. The training was done three months before assessments.	Audiovisual speech stimuli. A trained Japanese female speaker recorded four hundred words.	Some users complained that the tasks were too simple and wanted more quantitative variations and qualitative. Nevertheless, significant improvements were observed in speech intelligibility. The authors conclude that repetitive in-home auditory training using audiovisual speech stimuli on a tablet could help improve speech intelligibility.	The quantity and quality of the stimuli and the training presentation are essential for an efficient application.	Japanese

Hutter et al. [68]	A pilot study to investigate the applicability of music therapy in CI rehabilitation	Training performed alongside therapists assisted by computer software developed for the study.	-Melody recognition -Timbre identification	Psychological and musical testing was done before and after training. Tests were carried out uni and bilateral.	Ten individualized 50-minute sessions. Assessment is done through a self-concept questionnaire and musical tests.	Training used music as training content utilizing several German folk songs. Tests consisted of having users separate musical contexts. For example, participants were asked to separate different tones from a piano.	Music seems to provide an effective treatment option for CI users in the early stages after the initial activation of the speech processor. The researchers saw improvements for Subjective sound, melody, and timbre identification.	Pitch perception and speech recognition are highly related to audio training. Therefore, improving the ability to separate frequencies makes CI users better understand speech.	German
Fuller et al. [63]	Comparison of different approaches to audio training	One group with computer-assisted training. One group with group sessions and one control group.	-Melody recognition -timbre identification -speech and music perception	Pre and post-assessments were done through questionnaires performed before week one and after week eight of training.	Both groups trained at a hearing center. The training was done over eight weeks. Each group had six weekly training sessions, and each session lasted 2 hours.	Three different groups underwent different training to investigate differences in results between the two. One group had pitch/timbre training, whereas the second had music therapy. The last was a control group.	There were no significant differences between the training methods, except for melodic contour identification, which had substantial pitch/timbre training results. Also, the self-reported perception appeared to improve across training sessions for music therapy.	Pitch/timbre VS. Music therapy seems to yield similar results. However, there are ups and downs for both. For example, the social aspect and emotional nature of music and differences in the results seen from pitch/timbre training.	N/A
Smith et al. [64]	A prospective study investigated whether a self-administered computer-based rehabilitation program could improve music appreciation and speech recognition.	Self-administered music rehabilitative software. HearTunes / MusicEAR	-Musical pattern perception -Pitch perception -Music perception and enjoyment -Speech perception	Post-training diagnostic testing compared to pre-training scores four weeks apart. A third post-training assessment six months later.	Training took place at home or at work using the training software. The training lasted over one week.	Self-administered music rehabilitative software to help improve the perception of musical patterns, pitch, and timbre perception. Testing was done before and after training. There were two groups, one with high musical abilities and one with low.	There were significant improvements in musical pattern perception. High musical abilities group findings: increased complex pattern perception. Not any significant improvement in music enjoyment, pitch, perceptual ability, and speech perception. Low musical abilities group findings: Increased musical enjoyment, short-term	Current technology in cochlear implantation may not provide the spectral resolution and fine structures necessary for music perception. However, with training, this can be improved significantly.	N/A

							improvement in pitch perception, increased perceptual abilities, and speech perception.		
Miller et al. [72]	Case study implementing pre-test - intervention - post-test design to examine whether multiple talker identification training enhanced phonetic perception in post-lingual CI users.	A custom-designed training package was verbally displayed.	-Multiple talker identification -Phonetic perception -Contrast identification	Pre and post-assessments were performed with a control group.	Training consisted of eight hours of identification training over a two-week training period. Training took place at the clinic.	Familiar talkers and unfamiliar talkers carried out the test mainly consisting of contrast identification.	The perception of the familiar talkers was greatly improved, which transferred to the unfamiliar talkers. However, these effects were not present in the control group. Phonetic perception and categorization improved overall.	Identification training is more naturalistic and encourages listeners to attend to higher, more abstract category-level differences across stimuli. The paper has an Interesting discussion relating to familiar talkers vs. unfamiliar talkers.	American English
Jiam et al. [65]	Randomized crossover study including normal-hearing people and CI users to see the impact of music training	Self-administered training using Meludia[19] and a non-musical audiobook of own choosing	-Pitch perception and discrimination -Timbre perception	Three assessments in total. Baseline assessment at the start, test scores after one month, and at the end of the training periods.	The training lasted for two months. Each participant did a minimum of two hours of training each week.	Two groups. One group did one month of online self-paced music training, followed by one month of audiobook listening as control. The second group did one month of audiobook listening, followed by a month of music training as control. Both groups included CI users and normal-hearing people.	Both training methods improved pitch discrimination among CI users and normal-hearing listeners. In addition, music training and audiobook listening were found to improve performance on untrained tasks of pitch determination.	Musical instrument recognition was improved with music training. Instrument recognition is a complex auditory task independent of pitch and loudness detection. It requires the listener to interpret short segments of highly dynamic information.	N/A
Schumann et al. [73]	A qualitative study to prove that phoneme discrimination training improves speech recognition in noise in experienced adult	Computerized, phoneme-discrimination training program.	-Speech recognition in noise	Assessments followed a pretest-post-test design. Baseline assessment at start, test scores at the end of the training period.	There were three weeks of training, 45-60 minutes twice per week, and nine sessions. The training was done in a clinic.	Training material consisted of three nonsense syllable combinations for vowel-consonant-vowel (VCV) and consonant-vowel-consonant (CVC) based on German	All participants completed the protocol. Indicating that the measures to improve engagement worked. Nonsense syllables proved to be an effective intervention to	By decreasing the semantic information of the material, the patient was forced to focus on the recognition of phonetic properties	German-based nonsense syllable combinations

	cochlear implant listeners.					pronunciation. Each set included 30 target sounds. The second set was designed as a multiple-choice identification paradigm with decreasing difficulty.	improve speech perception in moderate noise. Performances were maintained over an extended period.	in the speech signals.	
Lo et al. [69]	An evaluation of two different melodic contour training programs. Several speech perception tasks assessed their relative efficacy.	The training was designed as a computer program participants could use at home. The software would increase the difficulty and lower it after a correct response. The method of decreasing or increasing difficulty varied between the two training methods.	-Speech recognition in noise -Consonant perception in quiet -Question or statement prosody	Assessments followed a pretest-post-test design. Baseline assessment at start, test scores at the end of the training period.	A Six week training period. The training was performed at home on a computer.	Two different melodic contour training programs. The main difference was how the method controlled the difficulty of the training. One did so by manipulating interval size, the other by tonal/note duration. There were nine different musical patterns with five consecutive notes.	There was no significant difference between the results for either training program. Both training methods improved some but not all aspects of speech perception. For example, significant improvements were found in the perception of consonants in quiet and identifying questions and statements using only speech intonation.	Confusion between the place of articulation cues in voiced, unvoiced, and nasal stops was the most significant reduction, typically the poorest speech patterns for CI users.	N/A
Barlow et al. [62]	Investigation of a short psychophysical, auditory training program to see if there could be benefits to speech perception and cortical auditory evoked potentials.	The computer-based training was performed on laptops containing the audio training content.	-Speech perception -Changes in CAEPs	Objective and behavioral measurements were taken four times in total. Three measurements prior and one measurement after training.	There were seven days of training with one hour of practice each day. The training was done at home with provided laptops.	Participants had to identify the odd stimulus when three stimuli were presented. Tasks were: gap-in-noise (identify noise containing a gap), detection of amplitude modulations, discriminating noise, discriminating frequencies and spectral ripple difference.	Speech recognition in noise significantly improved for easy and hard monosyllabic LNT (lexical neighborhood test) words. Improvement in speech scores consistent with other investigations of short-term auditor training.	The paper has an interesting discussion on the consequences of hearing loss. A small amount of training still yielded good results.	Australian English
Vandali et al. [66]	An investigation to evaluate whether musical pitch	A computer-based training program	-Pitch discrimination	Assessments followed a pretest-post-test design. Baseline	The four-month training period consisted of 30	Training consisted of two phases, each lasting two months.	Single cues can improve the performance of pitch	The authors have a great discussion on why phase 2	N/A

	perception could be improved with specific training.	was developed for the study.	-Timbre discrimination	assessment at the start, test scores at the end of the training period, and an additional test three months after ceased training.	minutes of training per day. Training took place at home.	The first phase consisted of single cues. The second consisted of more complex sounds with multiple cues. The training was presented in a game-like format where users matched acoustic patterns of pitch and spectral timbre to visual patterns.	and timbre discrimination. However, diminishing returns were seen as the complexity of sounds increased.	(multiple cue training) did not affect the outcome. Interestingly, diminishing returns were found when increasing the complexity of the sounds.	
Völter et al. [11]	Prospective intervention pilot study to evaluate the possibilities of using tablet-based telerehabilitation instead of conventional face to face rehabilitation	Traditional face-to-face rehabilitation is done in a clinic and over telecommunication on computer tablets.	-Speech comprehension in noise -Speech tracking -Phoneme differentiation	Assessments followed a pretest-post-test design. Baseline assessment at the start, test scores at the end of face-to-face training as well as after the computer-based audio training period	Six weeks training period is split into two-three weeks of training methods. Three weeks were done in a traditional face-to-face setting in a clinic and the last three weeks at home.	Participants completed three weeks of conventional training and three weeks of computer-based auditory training at home.	The computer-based training proved to have good clinical outcomes while saving CI users and clinicians time. A cost-efficient solution to address the lack of human resources in health care and the global challenge of current or future pandemics.	While considering the economic advantages of such applications, the total cost of only having face-to-face methodologies for audio training is more than what can be counted in dollars.	
Arne Kirkhorn Rødvik [18]	A Ph.D. thesis, quantitative cross-sectional project.	N/A	N/A	N/A	N/A	A three-step project consisted of: -A systematic review and meta-analysis. -A study of consonant and vowel confusion in adult CI users. -An investigation of vowel confusions in children with CI. We are most interested in the second step (the pilot study) and its result—nonsense syllables used as	The study found a devoicing bias for stops. There was a high confusion rate for nasals [m, n, ŋ]. The recognition score of consonants in context was higher than the "I" and "u" characters. The highest scores were with vowel repetitions, second for unvoiced consonant repetitions. Voiced consonant repetitions had the lowest score. Consonants were mainly confused with other consonants	Removing the ability to be able to guess the words reinforces learning the relation between sounds instead of relying on intellectual abilities.	Norwegian

						stimuli in adult Norwegian speaking CI users.	having the same voicing and manner.		
Firestone et al. [67]	A Preliminary study investigating the effects of attentive music listening	Home-based training using music-streaming programs like Pandora [113])	-Speech perception -Frequency change detection	Pre and post-training assessments included hearing thresholds, speech, spatial and qualities of hearing scale (questionnaire), frequency change detection threshold, speech recognition test, and magnitudes of frequency changes.	The training consisted of 40 min sessions five days each week for either four or eight weeks.	A home-based music training program. Participants used music streaming programs to listen to musical genres and emphasized actively listening to melodies.	The results saw significant improvement in frequency change detection and speech perception. Additionally, there were improvements in cortical processing of both stimulus onset and stimulus frequency changes.	Being able to separate and segregate frequencies is imperative for speech perception.	

6.2 Preliminary questionnaire results

The results of the preliminary questionnaire were the following.

- The participants had no singular preferred platform. The number of participants who preferred Desktop, iOS, or Android devices was almost equal, with a slight preference for iOS devices.
- The most challenging scenarios for CI users were
  - Speech perception in situations with a high level of background noise
  - Noises high in volume
  - Understanding words with specific frequencies. The letter S was mentioned multiple times to be especially difficult.

6.3 Primary questionnaire results

The primary questionnaire contained questions concerning the CI user's audio training needs, the specific content, and the training context. The questionnaire was relatively short and included 10 questions. In total, 28 CI users participated.

Question one: Have you undergone a CI operation?

The first question contained within the questionnaire was related to whether the participants had undergone a Cochlear implant operation. All 28 participants answered positively.

Question two: How often do you perform audio training?

Figure 45 shows the questionnaire participant's answers when asked how often they performed audio training. The participants could only choose one of the predetermined answers. If participants answered never, they would be asked why in a subsequent question. If they were to select one of the other options, they would be asked about the location, content, and travel time. Four participants (14.3%) answered that they never did audio training, whereas 24 participants (85.7%) stated that they did audio training with varying frequencies.

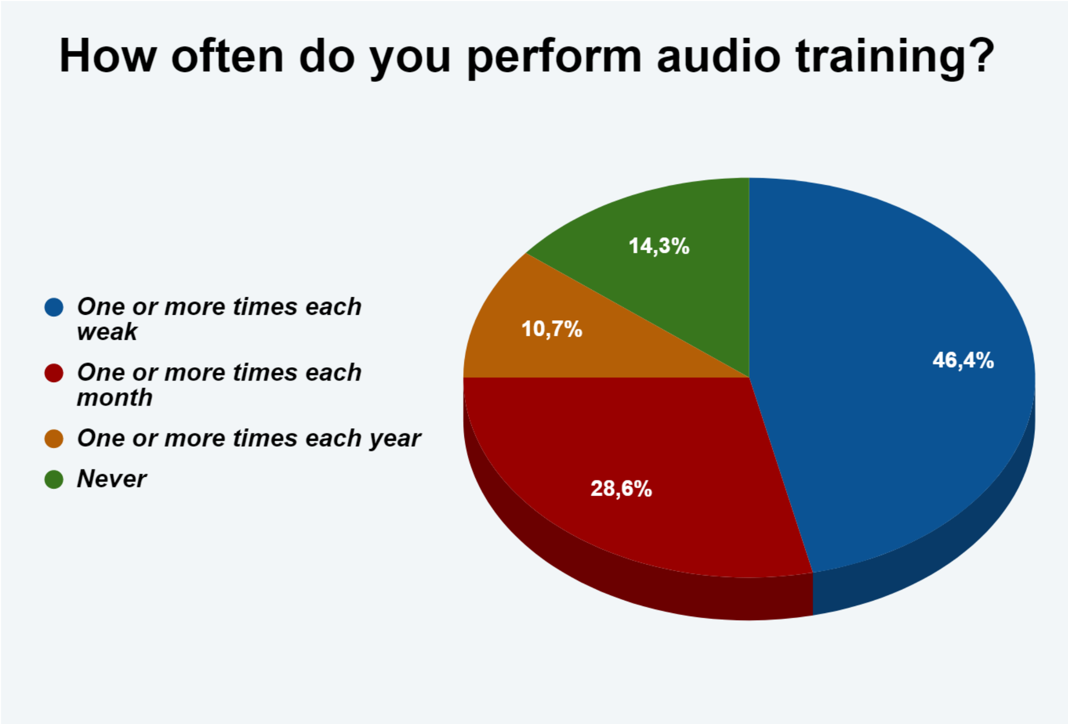


Figure 45: Showing how often participants performed audio training.

Question three: Why do some CI users not perform audio training?

The four participants who did not do audio training were asked why. The question was posed as a multiple-choice question with checkboxes. Figure 46 shows the four participant's answers. The participants could explain why they did not do audio training in further detail within a text input area. However, no additional reasoning was given.

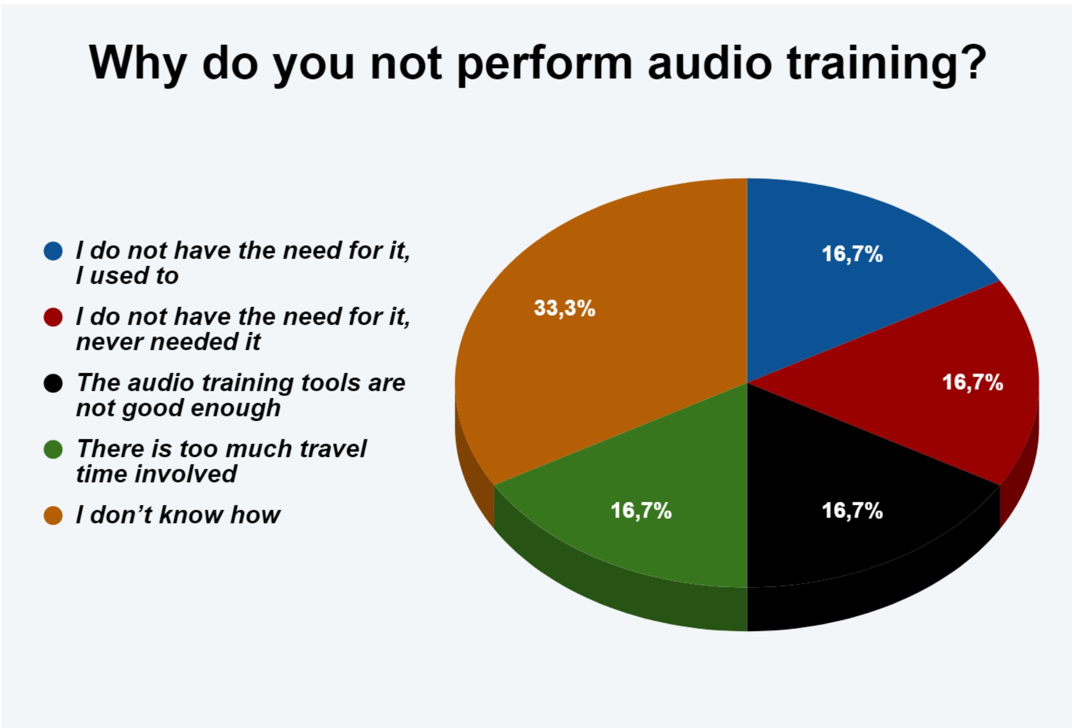


Figure 46: Shows the participants' reasoning for not doing audio training.

Question four: Is audio training typically done at home?

Figure 47 shows the participant's answers when asked about the location of their training. The question was posed as a multiple-choice question with checkboxes. The participants could provide additional locations within a text input area. However, no other locations were given.

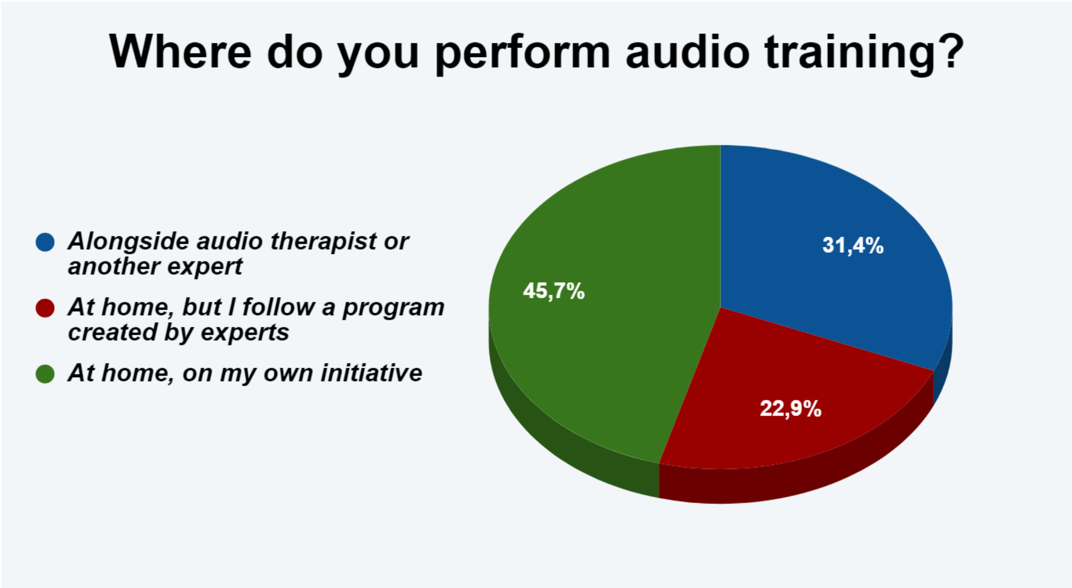


Figure 47: Shows where the participants performed audio training.

Question five: How much time do you spend traveling to and from training centers or experts to perform audio training?

Figure 48 shows the total time spent traveling to and from audio training for the questionnaire participants. The average travel time from the participants was 61 minutes. The answers were given within a text input area and were later grouped into 30-minute segments and tallied.

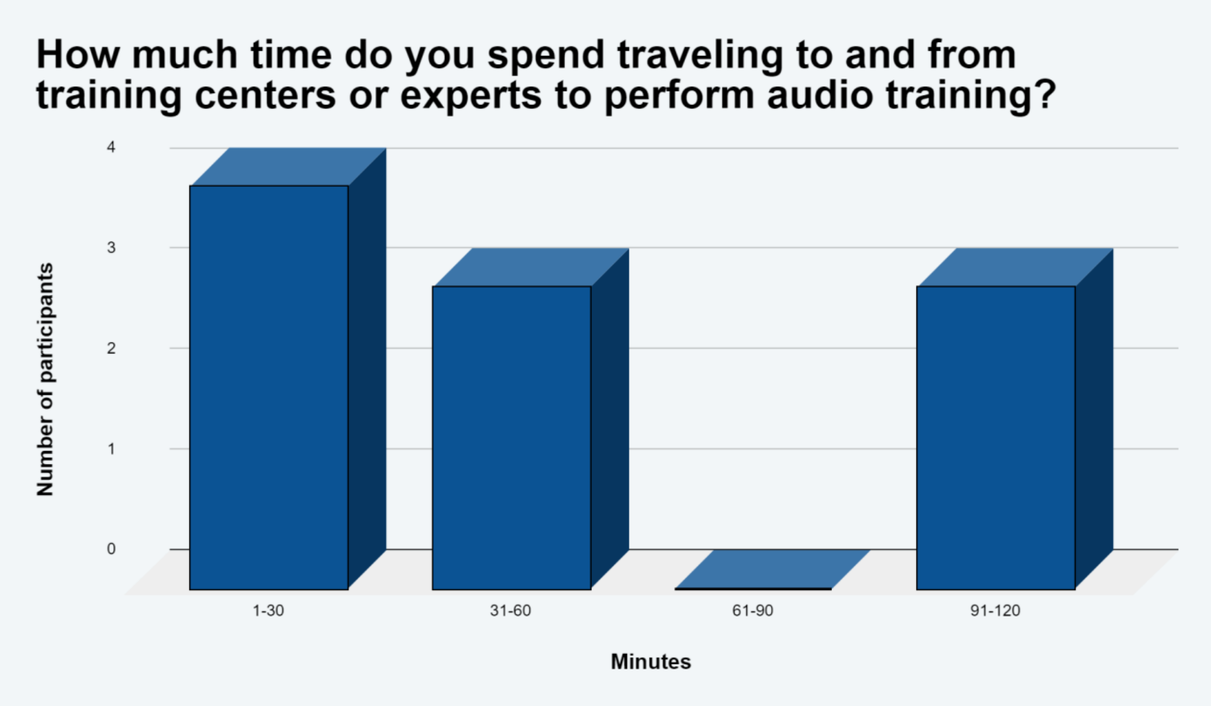


Figure 48: Shows how much time participants spent traveling to and from audio training.

Question six: What is the audio training content when training at home on your initiative?

Figure 49 shows the participant's answers when asked what the audio training content consisted of when training at home on their initiative. All answers were given within a text input area. The answers were grouped into generalizable terms after the fact, and the following groups were found:

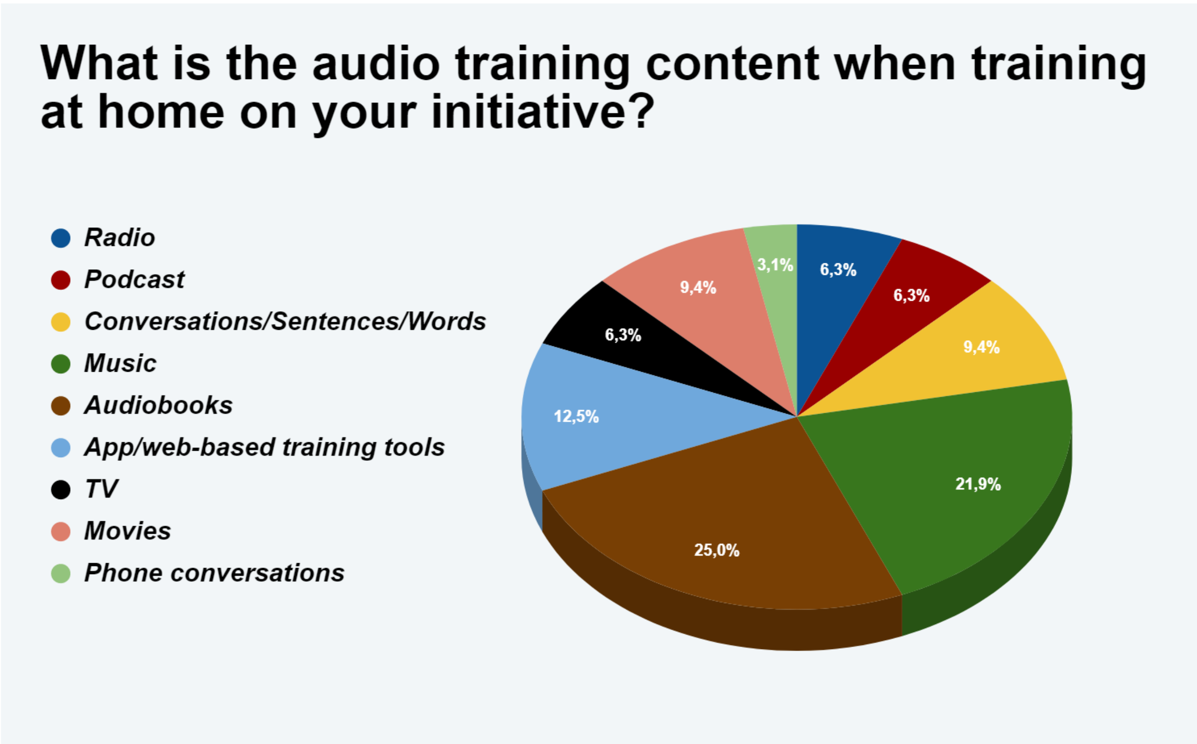


Figure 49: Shows the participants' audio training content when training at home.

Question seven: What is the audio training content when following an expertly made training program?

Figure 50 shows the participant's answers when asked what the audio training content consisted of while following training programs made by audio therapists or other experts. All answers were given within a text input area. The answers were grouped into generalizable terms after the fact, and the following groups were found:

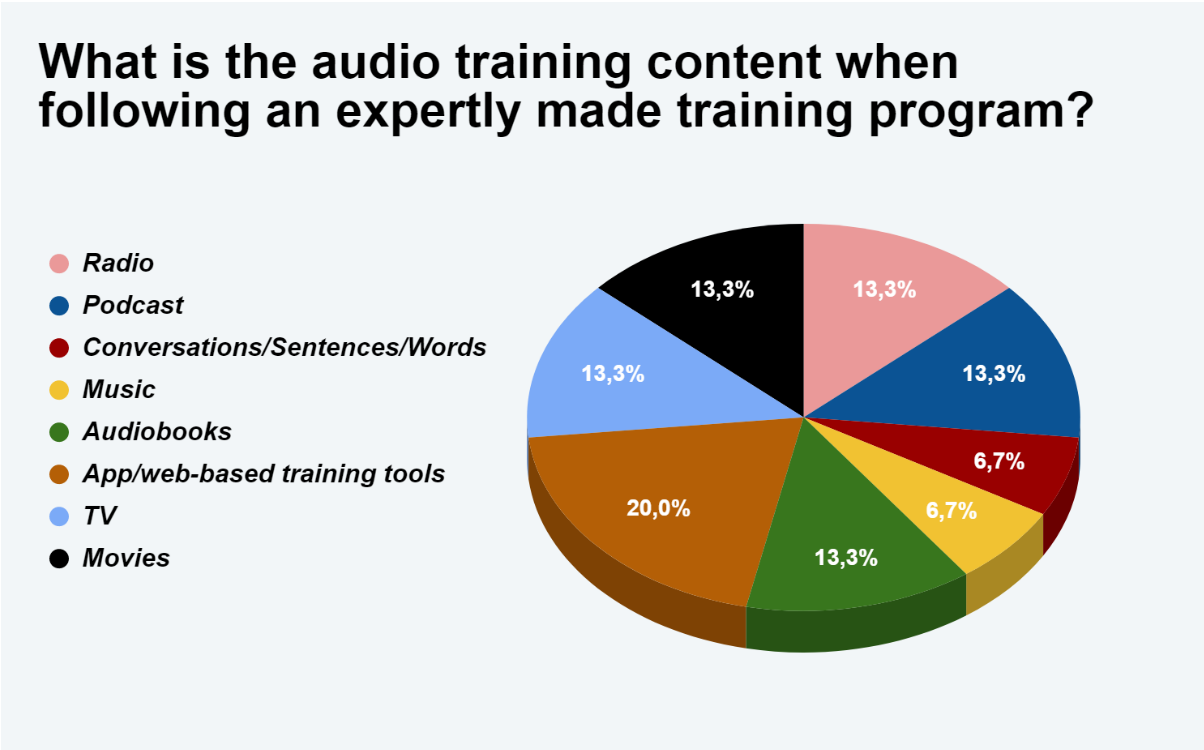


Figure 50: Shows the participants' audio training content when following an expertly made training program.

Question eight: Are CI users happy with the training opportunities available today?

Figure 51 shows the questionnaire participant's answers when asked if they were happy with the audio training tools available for them today. The participants chose one of the five predetermined answers.

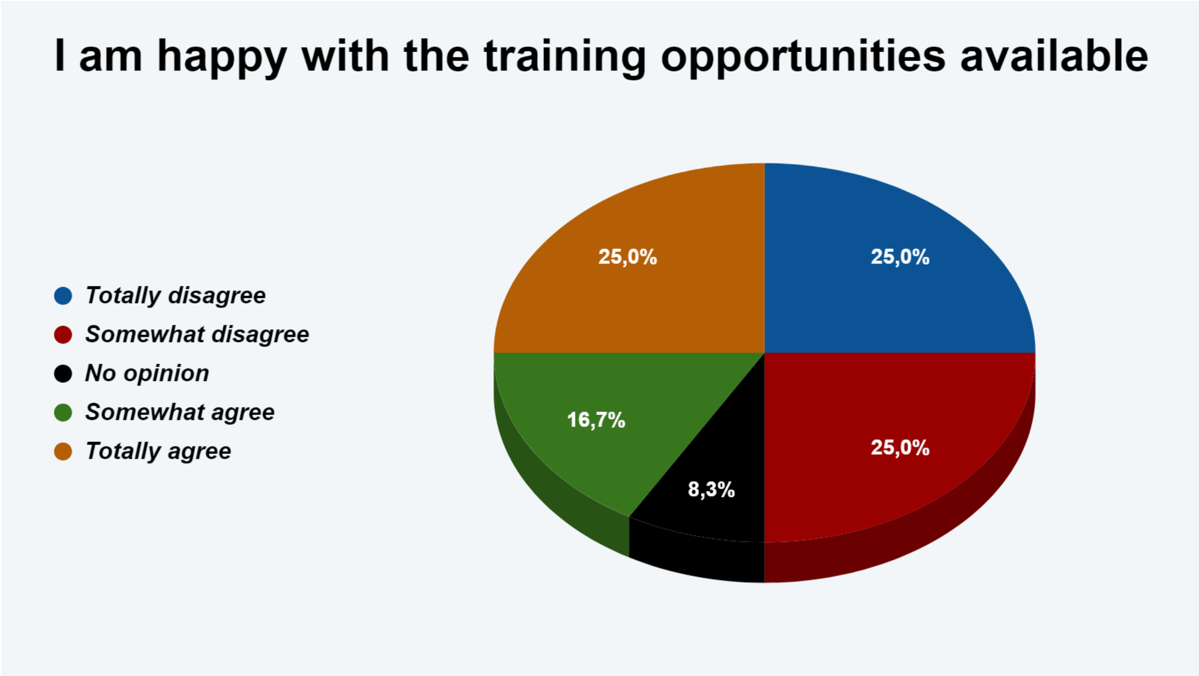


Figure 51: Shows the participants' happiness with the audio training opportunities available.

Question nine: Recording singular words or short sentences are better than longer sentences or stories for audio training content?

Figure 52 shows the questionnaire answers when asked whether the audio training content should be short sentences and words or longer sentences. Again, participants chose one of the five predetermined answers.

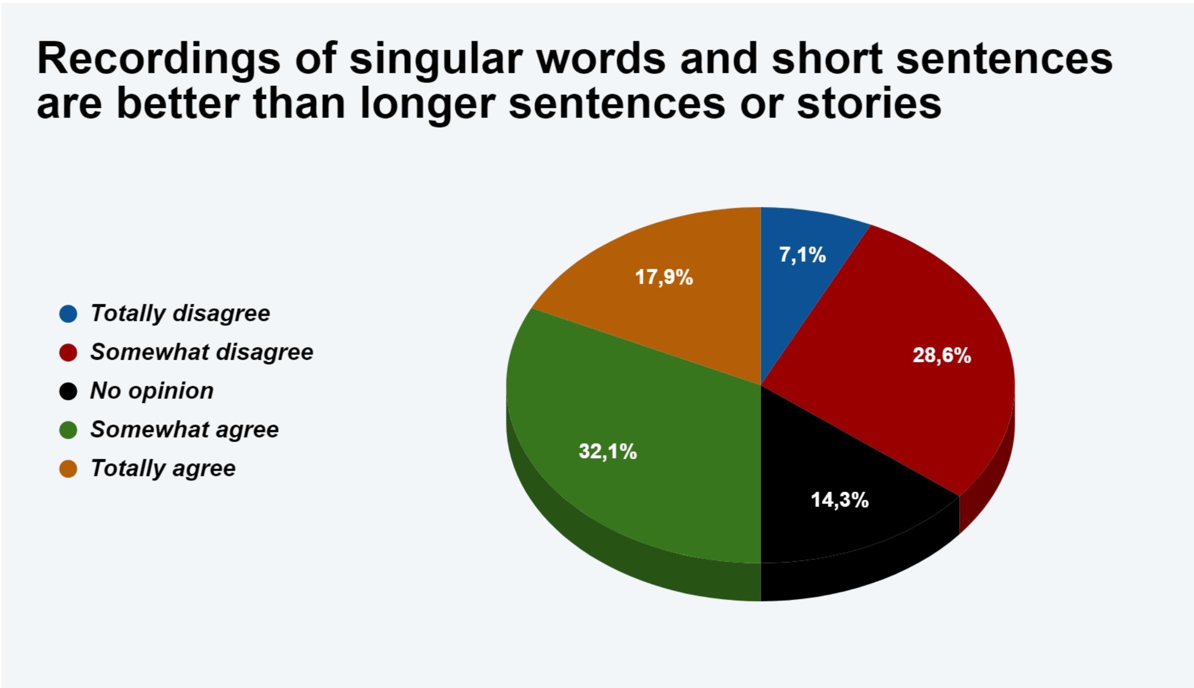


Figure 52: Shows the participants' preference between short sentences and longer stories.

Question ten: Is recording people talking in their dialects better than someone reading text?

Figure 53 shows the questionnaire participant's answers when asked if they would prefer recordings of people talking with their dialects or people reading text. Again, participants chose one of the five predetermined answers.

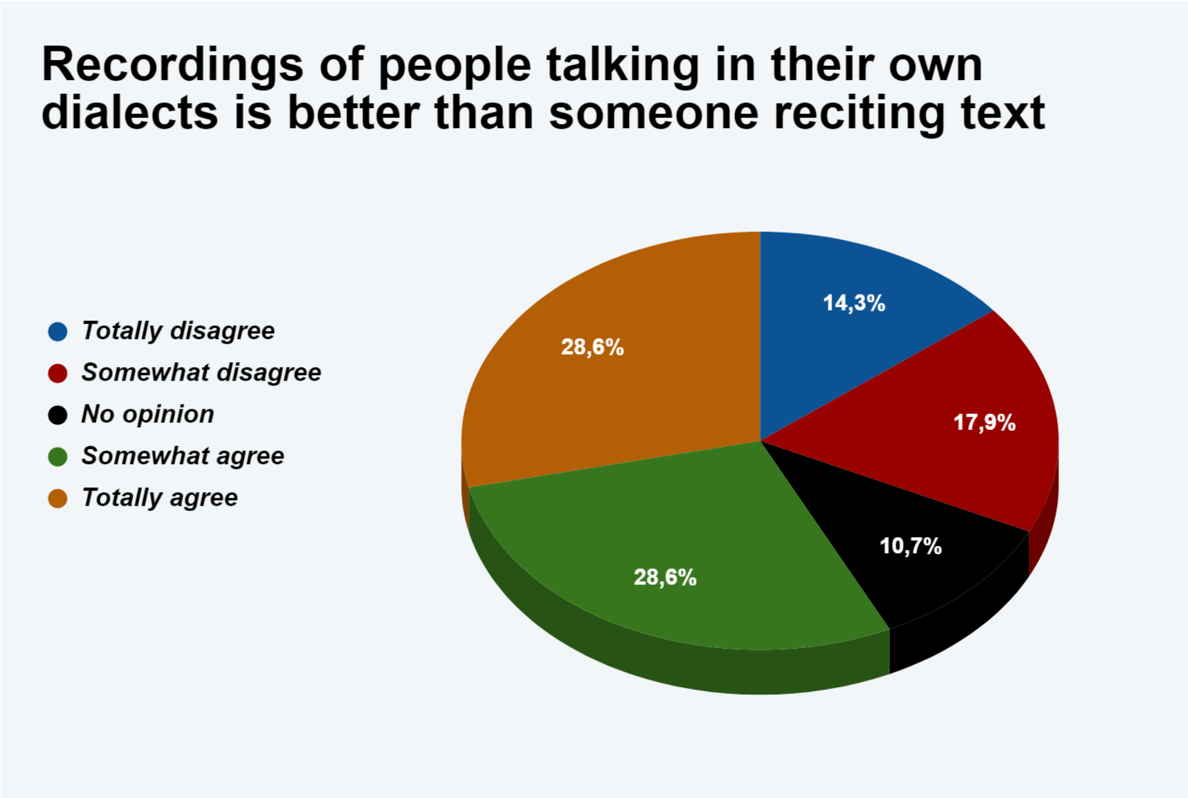


Figure 53: Shows the participants' preference between recordings of people talking in their dialect or someone reciting texts.

6.4 Semi-structured interviews

6.4.1 One interview session

One in-depth semi-structured interview was performed with a CI user who tested the digital training platform before the interview. The participant will be referred to as Jane Doe. These were the main takeaways from the interview:

Jane did reach the digital training platform through her preferred platform. She could wirelessly connect her sound processor to the desktop and preferred that method of listening to audible content. She could also use the desktop audio but felt that the audio was clearer while connected wirelessly. Jane said, *"I have heard other people having problems connecting the sound processors to their desktop, but I normally find this easier… It might be because I must be able to participate in online meetings for my work. It has become a habit."*

The following interview topic was aimed at jane's first impression, and the participant added that *"well… I was very excited to test it out, and when I reached the website, I was somewhat confused about where to click or what to do. So it felt natural to try one of the three buttons and landed on that first site (sound picture exercise)."*

Jane reported that the sound picture exercise was something she had never seen before. At first, she did not understand what to do. She stated: *"it took me a few seconds to understand what the exercise wanted me to do. After navigating the list where you could choose sounds, I saw that sounds were added as I clicked them. I tried playing them. It was nice to be able to add more than one sound. I thought it was a good idea."*

Jane eventually exhausted the topic, but she never mentioned trying the subtitles. Therefore, Jane was asked whether she had tried using the subtitle buttons. Jane stated that it took her a while to understand that subtitles were available. When asked if she had read the instructions, she said: "*I did not see the instruction button before navigating to the same exercise by mistake. I think the instruction button had been pushed to the bottom of the screen after opening the other one (the sound collection drop-down list). I never saw it when trying it the first time."*

Jane switched conversation topics and started to talk about the tone separation exercise. She had previous experience with such audio training activities but thought it was fun to have it in a game-like state where you could get a score. *"I have done quite a bit of this type of training before, so it wasn't anything new. The difficulty level was easy initially, but it was a good addition that it became more challenging as I went along... I don't think I have seen it in this way before, where you get scores, and it is more progression oriented."*

Similar to what had happened to the sound picture exercise, Jane had not seen the option to start at a more challenging level, so she never tried to change any settings or read the instructions. The conversation topic was altered toward the word separation feature. Jane said, *"that exercise was the most fun out of the three, in my opinion. It was easy to do, but I found myself clicking 'next' multiple times just to see what the next word was. It was enjoyable. I spent most of the time on that exercise."*

After talking about the specific exercises, Jane was asked whether she felt something was missing. She replied: "*Not really. You could… Maybe a quick guide and more information or something like that. Maybe I could have been more noticeable about the instructions on the first site (sound picture exercise), but after understanding that I had to choose sounds myself, it was okay after that… The other sites (exercises) were kind of self-explanatory. Maybe you should explicitly tell users to read the instructions before trying the exercises. If you read it one time, you don't need to see them again, and then it is not a problem that the instruction button gets pushed to the button".*

The conversation shifted toward similar digital training tools. When asked if Jane had tried similar digital tools, Jane replied: *"yes, there is another tool I have used online."* Jane could not remember what the name was, so I suggested that It might have been "CI hva du hører (CI-what you hear)"[15]. She replied: "*yes, that's the one… I can't remember if I could do the same exercises there. You could only click on buttons to hear what they had chosen for you. You could not choose sounds or build them yourself like in the first site (sound picture exercise). There were fewer choices there."*

After confirming that "CI-hva du hører (CI-what you hear)" was the only similar digital training tool Jane had used before, she was asked whether she could see herself using the digital training platform. She said: *"Absolutely. I mean, after having CI for a while, the tonal* *exercise and separating words becomes simpler, from my experience at least. So I would probably not use those two. I see the potential of the first site (sound picture exercise). So I think I will continue to try that more."*

Jane added, *"The most important thing for me is that the program (digital training platform) is usable when you turn on the sound for the first time until you feel that you don't need more training. You have created something that can provide that due to the first site (sound picture exercise). But you can start on the words, then separate tones before trying the first site (sound picture exercise)… the first site (sound picture exercise) was by far the most difficult one, and it is not as structured as the other ones."*

When asked if the training platform were something she would recommend to other people, she responded positively but stated that she would have to try it more before doing so. Furthermore, push notifications to encourage training were something she would be optimistic about as long as there was an option to opt-out. However, when asked about the frequency of such notifications, she would not do it more frequently than once a week.

The interview finished with Jane stating, *"I am happy to see that someone has created something new (talking about the training platform)… I appreciate that people develop these things so that we can have tools to teach us and have a good place to go for training."*

6.4.2 Feedback from one audio therapist

After posting the semi-structured interview recruitment message in the Facebook group "CI-Gruppa (CI-Group)," a similar message was published in a Facebook group called "Audiopedagog er/ skal bli / ønsker å bli? (is/ becoming/ want to become/ audio therapist)". The aim was to gather professional opinions to hear their thoughts on the digital training platform. One audio therapist responded and stated that the digital training platform seemed exciting and thought it had potential. Furthermore, the audio therapist said the digital training platform could be used as a supplement to regular audio training. The person did not work primarily with CI users but had experience with CI users as patients. The response can be seen in Appendix XVII.

6.5 Performance testing

6.5.1 A low number of clients sending sequential requests

The figures displayed in this sub-chapter compare the performance of four different endpoints with different Raspberry Pi (RPi) cluster configurations. This test was performed with 2-8 concurrent clients, sending 1000 sequential requests. The endpoints tested were the following:

- HTTP GET "/" pointing to the landing page with HTML and JSON response. Results are seen in Figure 54 and Figure 55.
- HTTP GET "/rhyme" pointing to the rhyme finding algorithm. Results are seen in Figure 56.
- HTTP GET "/Dyrelyder/Hane.mp3" including file lookup, reading, and transfer (file size: 350KB, folder depth: 2). Results are seen in Figure 57.

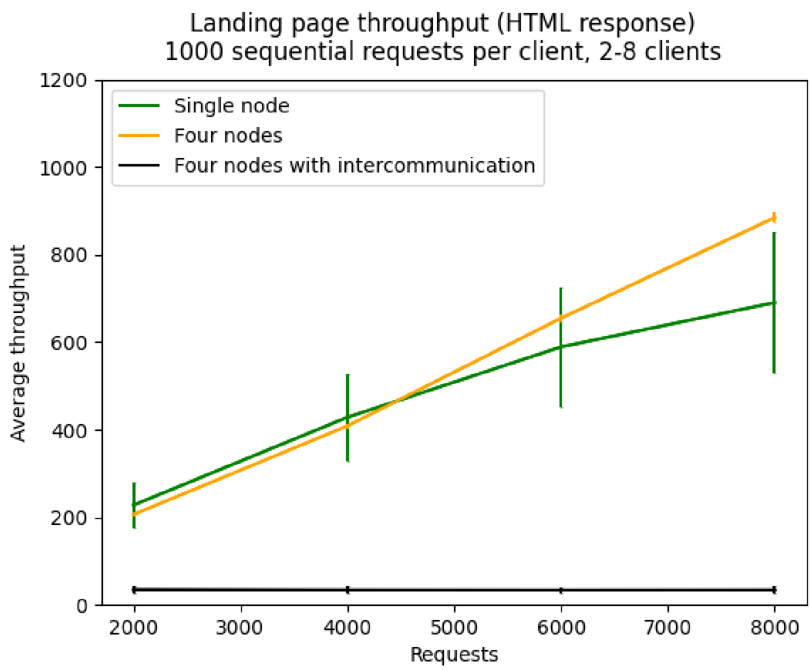


Figure 54: Comparing throughput of endpoint leading to the landing page with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients.

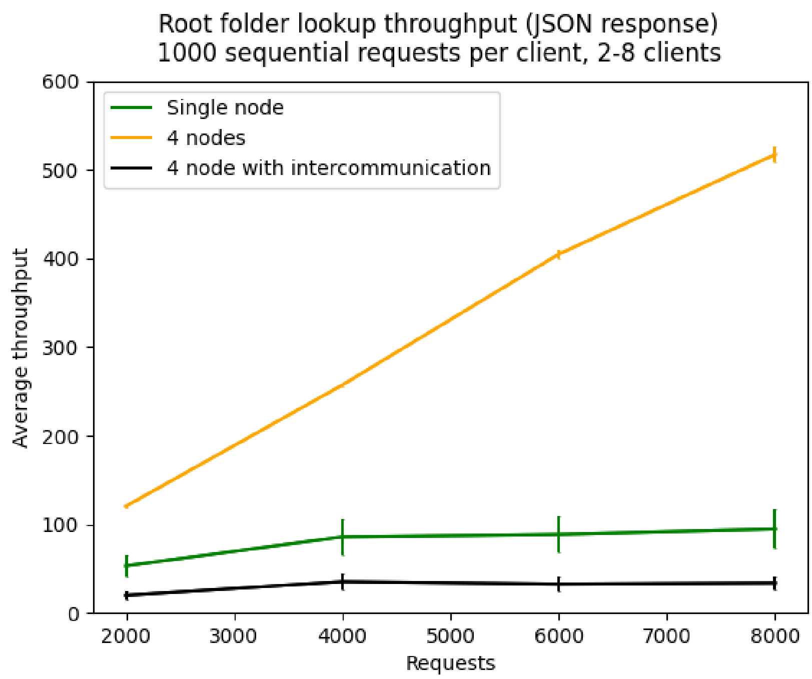


Figure 55: Comparing throughput of endpoint leading to root folder lookup with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients.

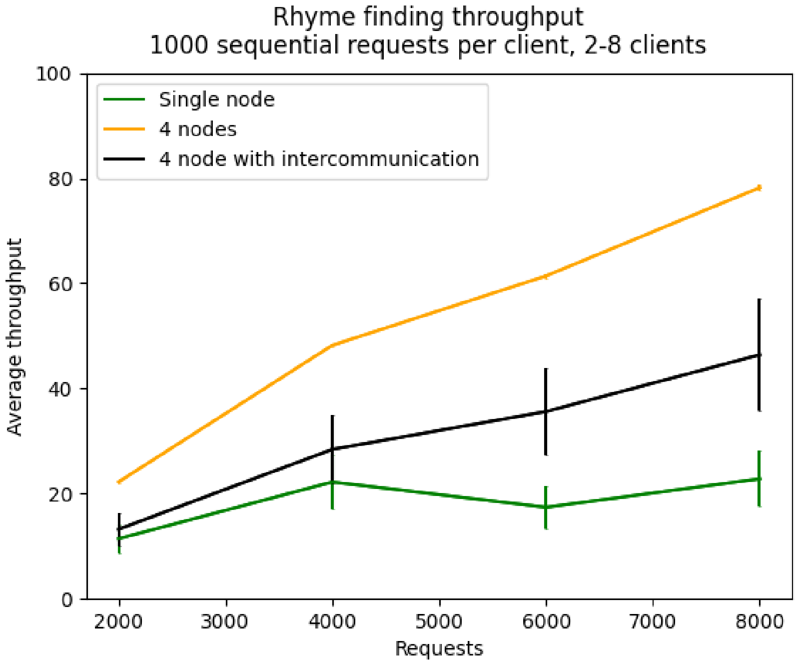


Figure 56: Comparing throughput of endpoint leading to rhyme finding algorithm with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients.

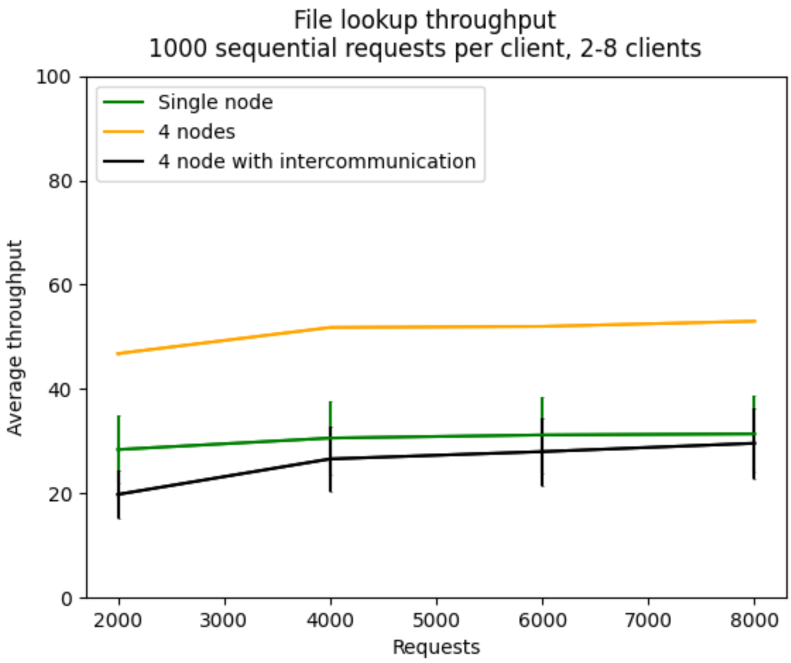


Figure 57: Comparing throughput of file lookup endpoint (file size: 350KB, folder depth: 2) with different Raspberry Pi configurations'. 1000 sequential requests per client, 2-8 clients.

6.5.2 A high number of clients sending asynchronous requests

The average throughput and number of dropped requests for the Linode Kubernetes Engine cluster

Figure 58 shows the performance of the LKE deployment for different endpoints. Figure 59 shows the number of dropped or declined requests. The test was performed with 50-1000 concurrent clients, sending one request each. The endpoints tested were the following:

- HTTP GET "/" pointing to the landing page with an HTML response.
- HTTP GET "/rhyme" pointing to the rhyme finding algorithm.
- HTTP GET "/Dyrelyder/Hane.mp3" including a file lookup process, reading, and file transfer (file size: 350KB, folder depth: 2).

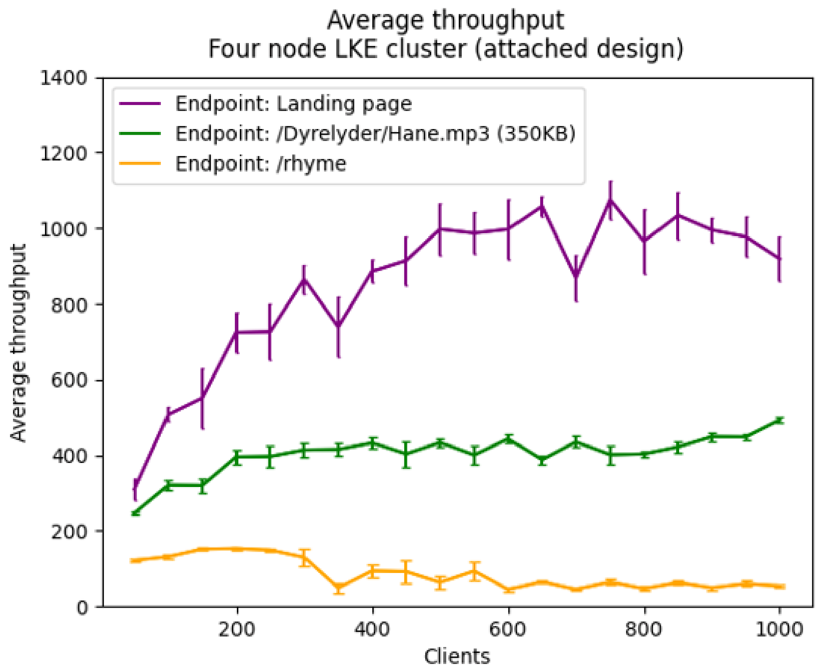


Figure 58: Average throughput for the Linode Kubernetes Engine cluster. One request per client, 50-1000 clients.

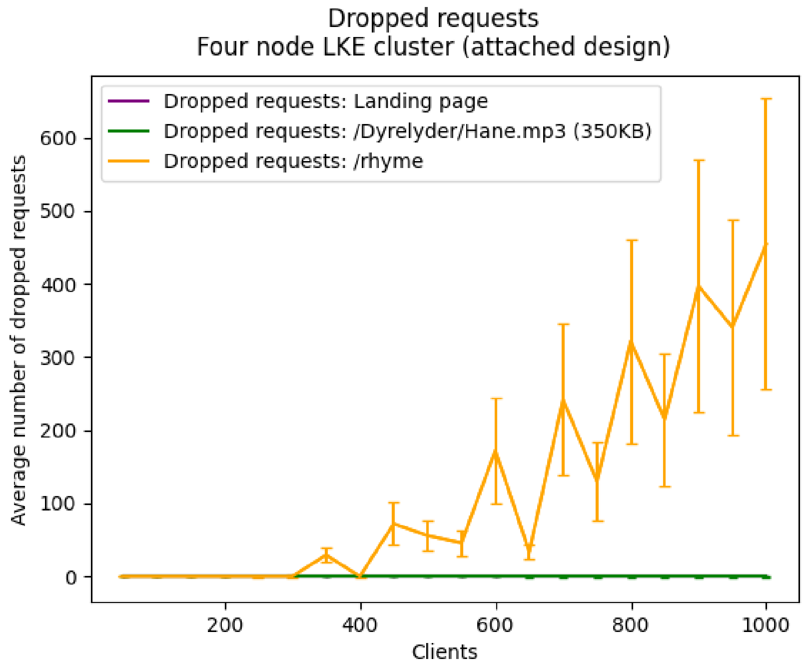


Figure 59: Average number of dropped requests for the Linode Kubernetes Engine cluster. One request per client, 50-1000 clients.

The average throughput and number of dropped requests for the Single node Raspberry Pi server

Figure 60 shows the single Raspberry Pi (RPi) node performance for different endpoints. Figure 61 shows the number of dropped requests. The test was carried out similarly and included the same endpoints as seen in Figure 58.

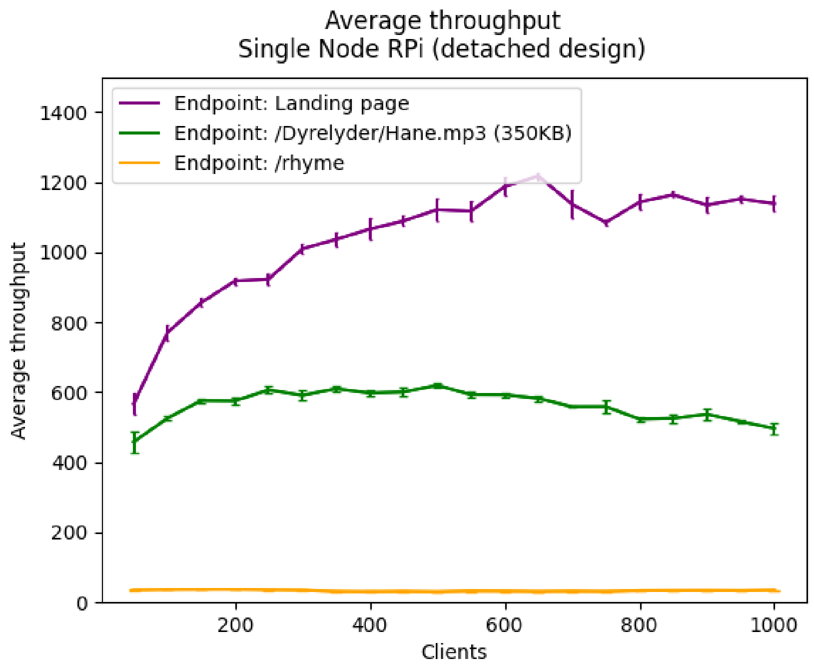


Figure 60: Average throughput for the Single node Raspberry Pi server. One request per client, 50-1000 clients.

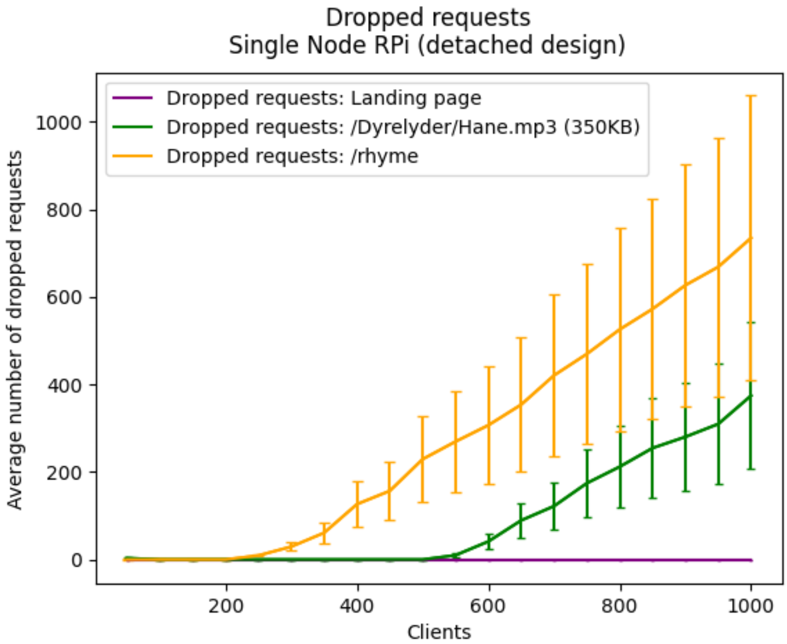


Figure 61: Average number of dropped requests for the Single node Raspberry Pi server. One request per client, 50-1000 clients.

The average throughput and number of dropped requests for the Raspberry Pi cluster

Figure 62 shows the RPi cluster's performance for different endpoints with four nodes. Figure 63 shows the average number of dropped requests. The test was conducted similarly and included the same endpoints as seen in Figure 58.

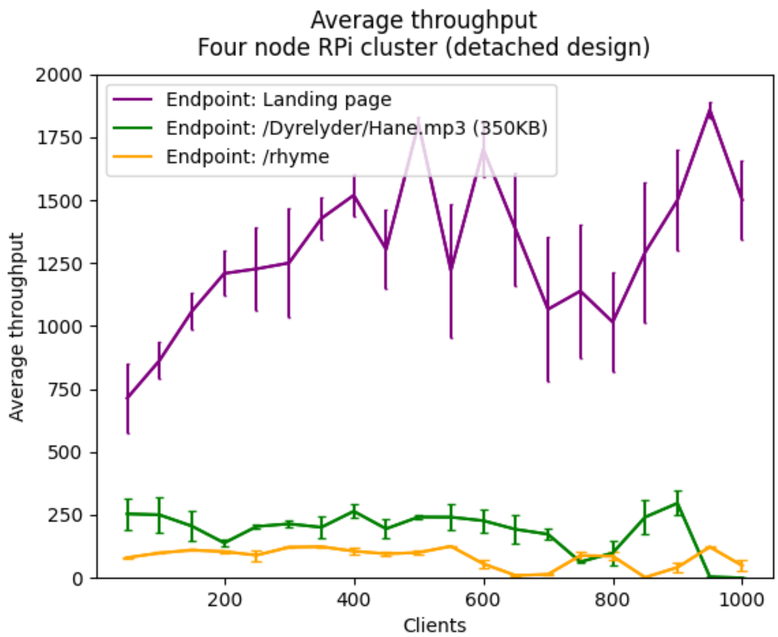


Figure 62: Average throughput for the four-node Raspberry Pi cluster. One request per client, 50-1000 clients.

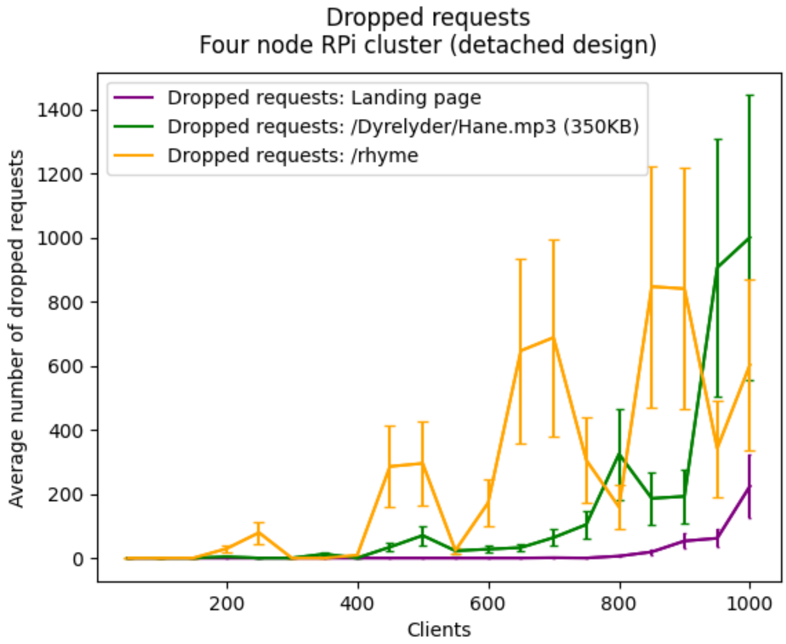


Figure 63: The average number of dropped requests for the four-node Raspberry Pi cluster. One request per client, 50-1000 clients.

The average throughput and number of dropped requests for the Raspberry Pi cluster with intercommunication for failure detection

Figure 64 shows the four-node RPi cluster deployment with intercommunication performance for different endpoints. Figure 65 shows the average number of declined requests (right). The test was conducted similarly and included the same endpoints as in Figure 58.

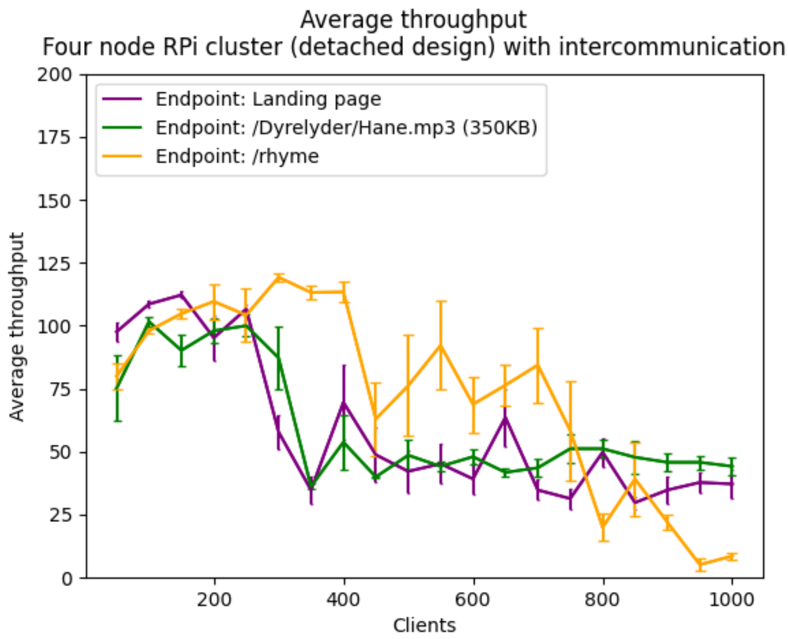


Figure 64: Four node RPi cluster with intercommunication performance. One request per client, 50-1000 clients.

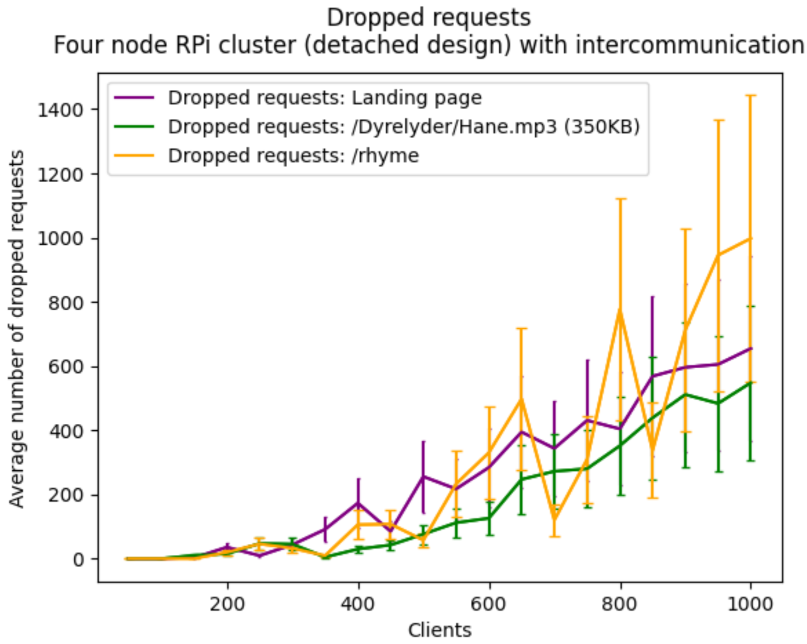


Figure 65: The average number of dropped requests is for the four-node Raspberry Pi cluster with intercommunication enabled for failure detection. One request per client. 50-1000 clients.

Comparing performance for HTML response (landing page)

Figure 66 compares the different cluster configurations' performances for the HTTP GET "/" endpoint which returns an HTML response of the landing page.

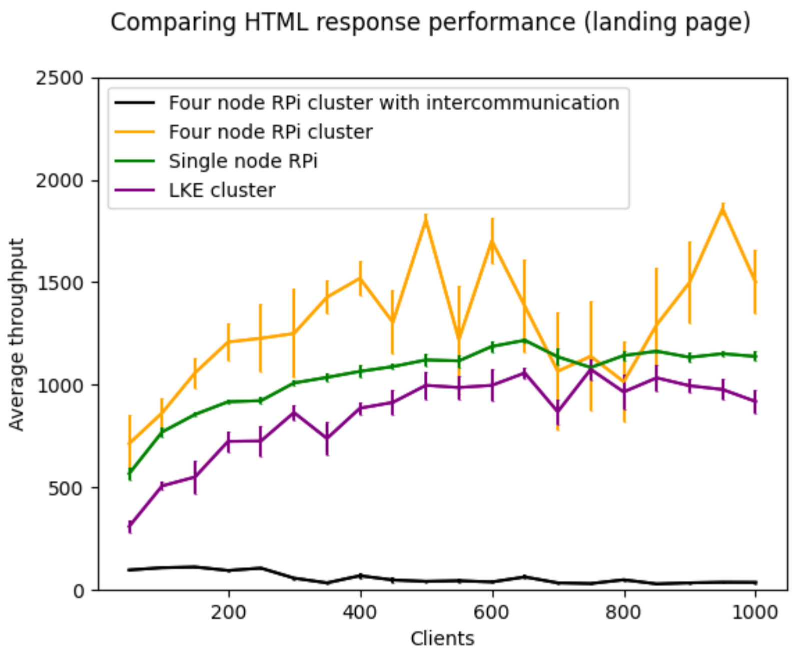


Figure 66: Comparing average throughput for the landing page endpoint for all configurations.

Comparing performance for file lookups

Figure 67 compares the different cluster configurations' performances for lookup on a file stored with a folder depth of 2 and file size of 350KB. The endpoint initiates a lookup process, reads the file, and returns a byte stream as the response.

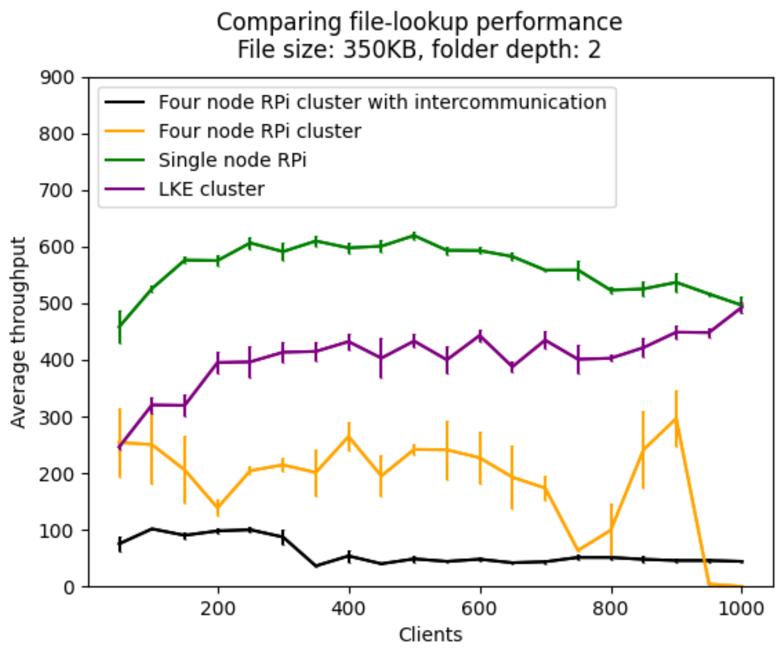


Figure 67: Comparing average throughput for the file lookups for all configurations (file size: 350KB, folder depth: 2).

Comparing performance for the rhyme finding algorithm

Figure 68 compares the different cluster configurations' performances for the HTTP GET "/rhyme" endpoint, which returns a JSON response.

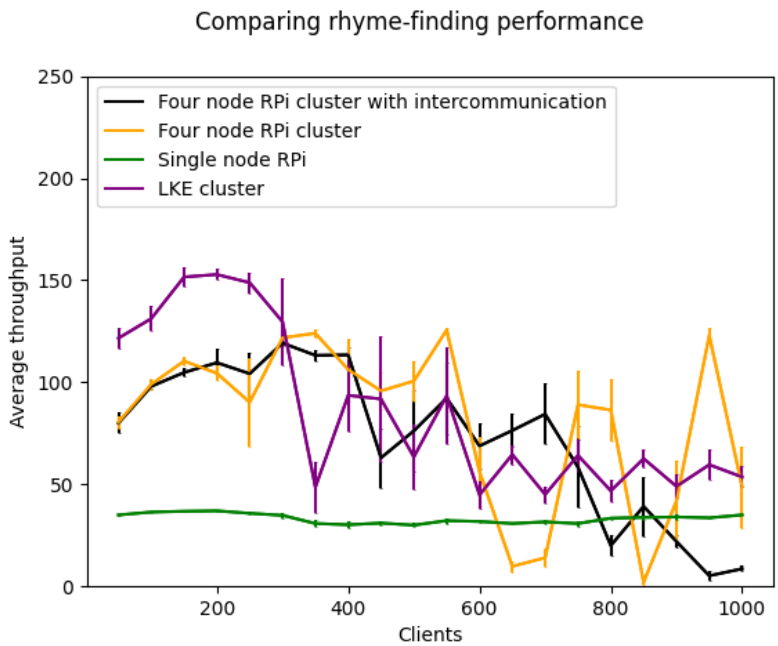


Figure 68: Comparing the average throughput for the rhyme finding algorithm for all configurations.

Comparing cluster configurations' performance as file sizes increase

Figure 69, Figure 70, Figure 71, and Figure 72 compare the different cluster configurations' performance for file lookups at different file sizes stored at a folder depth of 100. The file sizes start at 200KB and increase to 400KB, 800KB, and 1.6MBs.

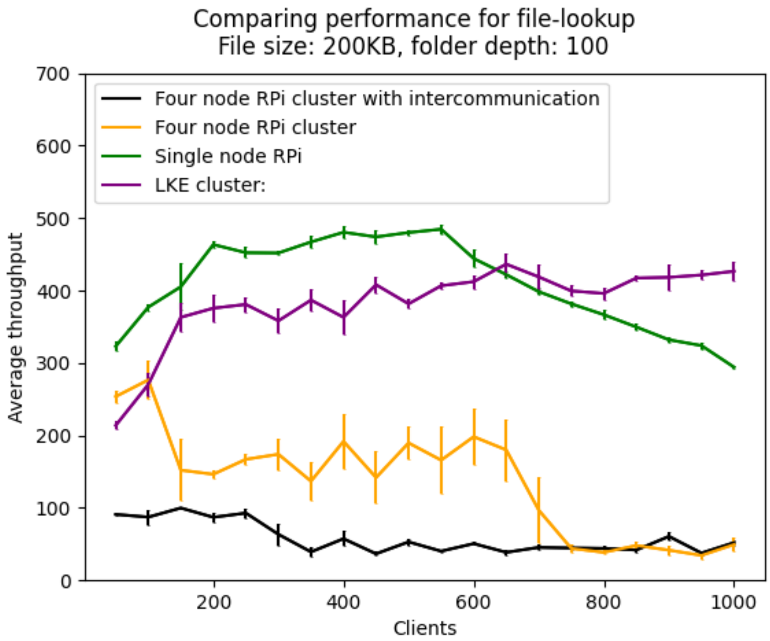


Figure 69: Comparing average throughput for file lookups (file size: 200KB, folder depth: 100). One request per client, 50-1000 clients.

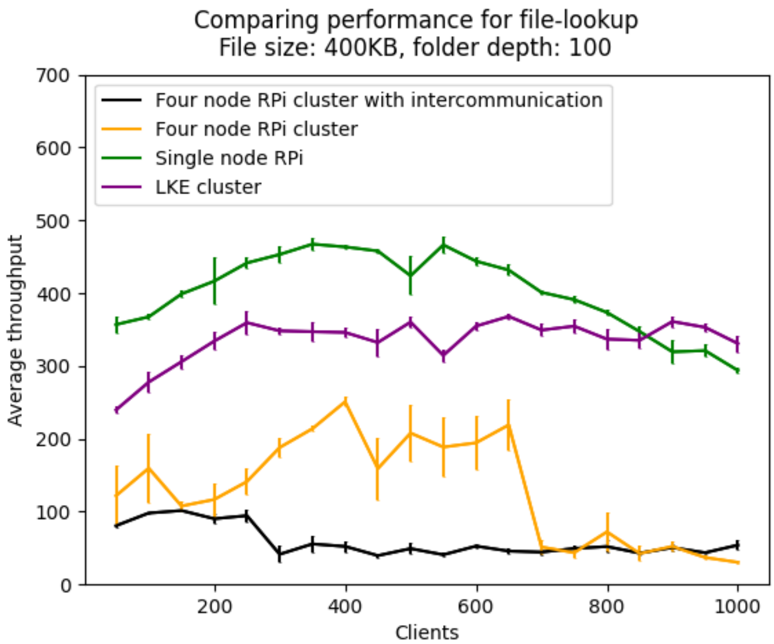


Figure 70: Comparing average throughput for file lookups (file size: 400KB, folder depth 100). One request per client, 50-1000 clients.

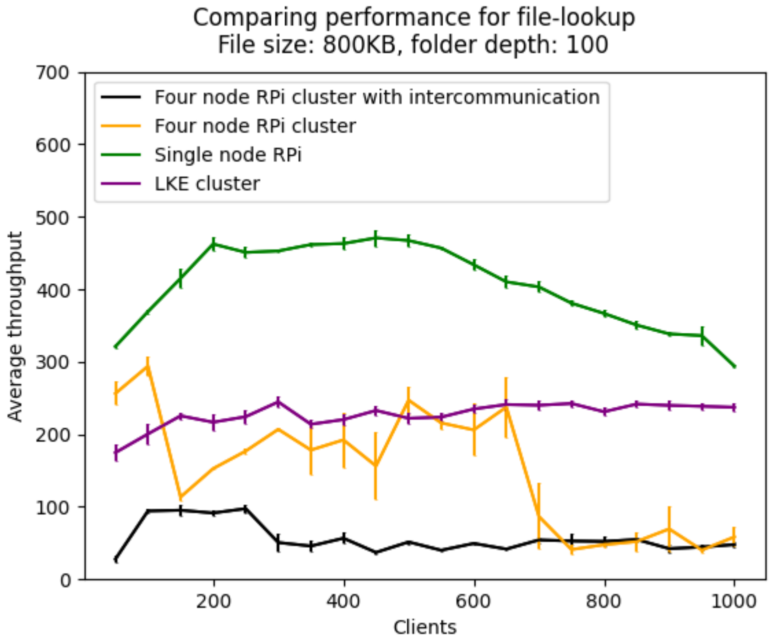


Figure 71: Comparing average throughput for file lookups (file size: 800KB, folder depth 100. One request per client, 50-1000 clients

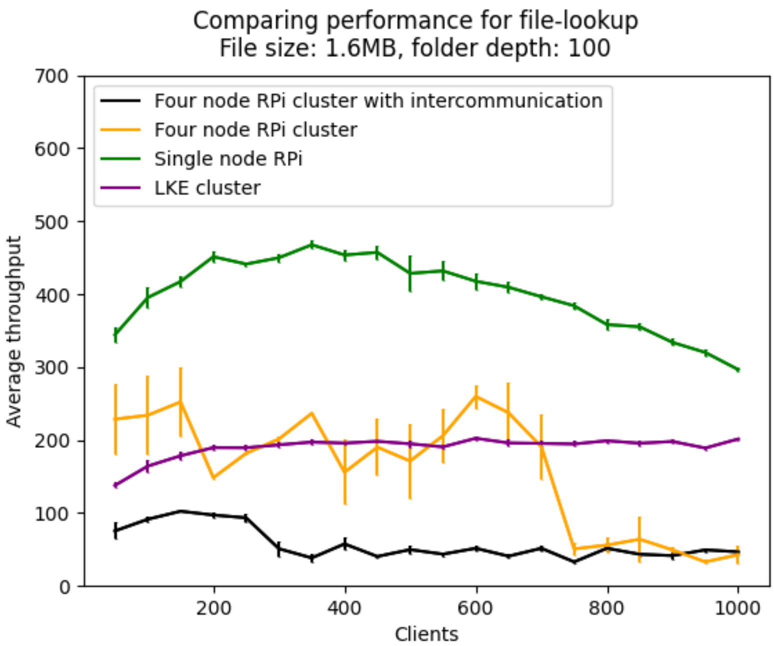


Figure 72: Comparing average throughput for file lookups (file size: 1.6MB, folder depth 100. One request per client, 50-1000 clients

Comparing stress test performance for all configurations'

Figure 73 compares the different cluster configurations' performance from the stress test. The test was performed with 50-1000 concurrent clients, sending seven requests each. The requests sent are the same as the requests utilized in the tests mentioned above:

- HTTP GET "/" – Landing page, HTML response
- HTTP GET "/Dyrelyder/Hane.mp3" – File lookup, byte stream response

- HTTP GET "/rhyme" – Rhyme finding algorithm, JSON response
- HTTP GET "/folder-depth-100/file-size-200KB" – File lookup, byte stream response
- HTTP GET "/folder-depth-100/file-size-400KB" – File lookup, byte stream response
- HTTP GET "/folder-depth-100/file-size-800KB" – File lookup, byte stream response
- HTTP GET "/folder-depth-100/file-size-1.6MB" – File lookup, byte stream response

All requests were sent asynchronously (simultaneously).

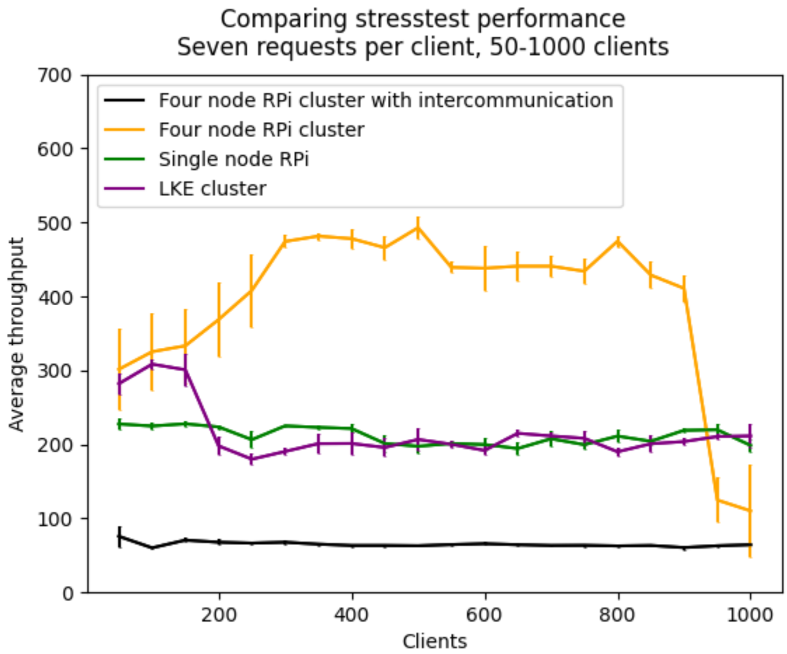


Figure 73: Comparing average throughput from stress test performance for all configurations. Seven requests per client, 50-1000 clients.

7 Discussion

7.1 Literature review

The literature review is shown in Table 9, conducted by utilizing PubMed [39], ACM DL [40], and IEEE Xplore [41]. Unfortunately, there were almost no relevant hits from the ACM and IEEE databases. ACM and IEEE focus on technological research, which may be why there was no relevant research found as Cochlear implants are medically related.

There was an attempt to distinguish the training tools and technology from the CI research. However, separating the topics was difficult, as most research papers and projects seem to be related to trying some new training tool or investigating the efficacy of a training exercise provided by some technology. Therefore, the literature review combined the two and included the most relevant research.

The main takeaways from the literature review are that most research focuses on the traditional way of conducting audio training. They are providing users with phonetic and synthetic exercises. However, the papers investigating the efficacy of musical training found promising results. There seems to be a consensus that musical training is an efficient way of training tonal separation and pitch perception for CI users.

7.2 Preliminary questionnaire

Only the conclusions drawn from the preliminary questionnaire have been shown throughout the thesis. The results have not been revealed because, as mentioned in chapter 2.3, the participant's right to privacy was not ensured. This happened as I was eager to get started on the thesis. In the summer before the thesis started, I decided to post the questionnaire to see if there were any platform preferences within the CI community. Additionally, I was interested in knowing more about difficult situations or scenarios for CI users. The initial thought was that the training tool could eventually simulate such scenarios.

As I had not consulted my supervisors and had never done questionnaires in the health domain before, I was unaware of the proper procedures. Therefore, only the conclusions from the preliminary questionnaire are stated, as it would be difficult to explain some of the earliest choices without mentioning the preliminary questionnaire.

The questionnaire and semi-structured interviews that followed the preliminary questionnaire ensured the right to privacy for its participants by reaching out to the correct channels with the guidance of my supervisors.

There were four main findings from the preliminary questionnaire.

- There was a slight preference for iOS devices, but most platforms were represented almost equally.
- Situations with a high level of background noise or with high volume were something the participants found to be difficult.
- The letter “s” was mentioned several times to be especially challenging.

After looking into the compatibility of different sound processors, the slight bias towards iOS devices could be explained as earlier sound processors were locked to either iOS or Android. For example, the Cochlear [7] made sound processors Osia 2 and Baha 5 [114] are only compatible with iOS devices. Furthermore, even earlier sound processors, such as the Nucleus six [114], were not compatible with any smart device and needed an intermediate link between the sound processor and sound source. Therefore, a Nucleus 6 sound processor needs a True wireless device [115] to link the sound processor toward an external sound source such as a television or GPS. These compliance-related issues might be the reason there is a slight preference towards iOS devices.

Multiple participants mentioned the letter “s” to be a problematic letter. Its frequency can explain why this is. Looking at the speech banana in Figure 74, the letter “s” is the letter with the highest frequency within the alphabet. The speech banana is an audiogram [116] of all letters within the alphabet [117].

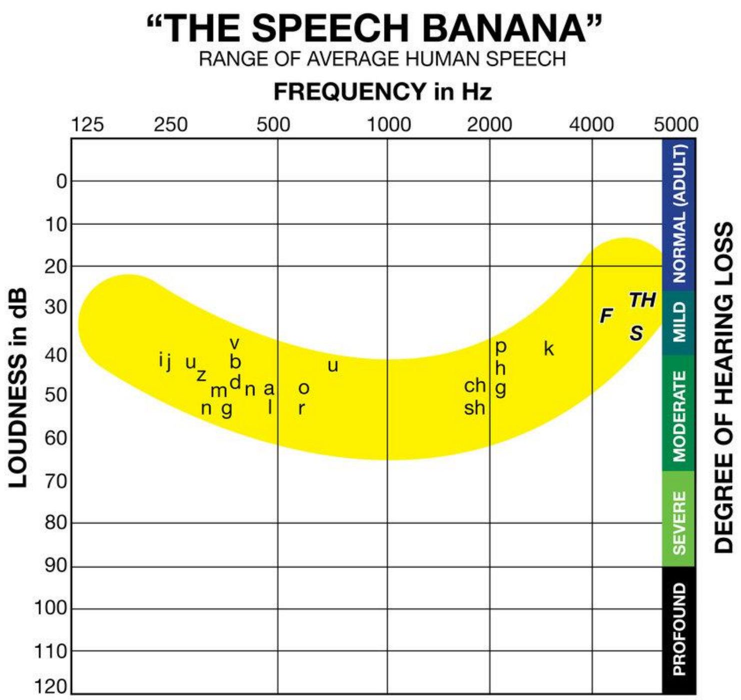


Figure 74: Image of the alphabetic audiogram (The "speech banana") [118]

Situations with high background noise and loud noises were challenging for the participants. These scenarios were also mentioned throughout the literature review shown in Table 9. Therefore, the sound picture exercise was created to simulate these scenarios where there are background noises or noises with high volume relative to other sounds.

7.3 Primary questionnaires

Four of the 28 participants stated that they never did audio training. However, most (85%) performed audio training with varying frequencies, as seen in Figure 45. In addition, most of the participants did audio training one or more times each week.

The four (14.3%) participants who did not do audio training had different reasons, all shown in Figure 46. The sample size is too small to make any definitive statement about the broader CI community, but it gives some hints about the development of the proposed system.

Knowing the location is an essential factor in understanding what audio training typically looks like, is it done at home, or is traveling involved? I hypothesized that most people would answer that they performed audio training at home while following a training program made by experts. However, I was wrong, as most participants answered, “At home, on my initiative,” as seen in Figure 47. I can only speculate why this is, and there should probably be more research done to investigate the reason for this.

When asked if the participants were happy with the currently available training tools, there was a slight tendency toward not being satisfied, as seen in Figure 51. However, the difference between the answers was minuscule. Bigger sample size is needed better to understand the general opinion of the CI community.

The only training method present in the own initiative training and not in the expertly made training programs was training using conversations over the phone, shown in Figure 49 and Figure 50. Audio training utilizing conversational techniques over phone communication was pointed out in Borel et al. [70] and Sousa et al. [71]. Interestingly, the difference in training content between expertly made training programs and training done on the participant's initiative is minuscule. Therefore, logically, one may conclude that the CI users will continue with the same training content provided by experts, perhaps less rigidly and on their initiative. If this is correct, it may also explain why most participants did audio training at home, as they may have continued the expertly made training programs. Sadly there were not enough clarifying questions asked to provide any definite answer.

From the participant's answers shown in Figure 52 and Figure 53, I could not determine whether there should be a strong emphasis on shorter sentences over longer ones. There was also no way of determining if there should be a focus on recording conversational speech containing different dialects or recording people reciting texts. The participants wanted different things, and every option seemed to be as viable as the other. The focus became covering most bases instead of focusing too heavily on one or two concepts. Therefore, there are recordings of people stating longer and shorter sentences, emphasizing their dialects, by reading shorter poems by Tekstforslag.com [55]. I recorded myself reading fictional writings emphasizing the letter “s” to provide longer content. Finally, the word separation exercise was implemented to offer singular words.

There was evidence of sociological consequences shown in Figure 48. For example, three participants reported having to travel two hours in total to be able to perform audio training alongside audio therapists or other experts. As pointed out by Völter et al. [11], CI users have sociological consequences as they have to travel varying distances to perform audio training alongside audio therapists. Völter et al. found that their participants “saved a mean of almost 4 hours of traveling” by doing audio training at home instead of traveling to training centers. In the questionnaire posed within the thesis, participants would save 61 minutes of travel time on average (Figure 48). Giving these participants the ability to train from home may save their time and free up significant resources and assets within the healthcare system. Additionally, it is essential to point out the need to provide home-based training if the Covid-19 pandemic should flare up again.

7.4 Semi-structured interview

7.4.1 Low turnout in the recruitment process

The semi-structured interview recruitment process did not go as expected. Despite posting the recruitment message multiple times, I could not recruit more than one CI user to participate in testing and provide feedback through an interview.

The communication media used within the interview process is likely the cause of this. Many CI users struggle with phone conversations, and there should have been an option to use video calls over Teams, Zoom, or Skype.

The decision to use phones as the communication medium was made based on these factors:

- The privacy concerns surrounding video calls are more extensive.
- My supervisors had previous experience with this method of gathering feedback.
- I underestimated CI users challenges when it comes to phone usage.

The fact that using phones as a communication device during the interview process could be a potential problem was raised by my supervisors before sending the application to NSD. However, due to my own experience with my relative, I underestimated the issue as I talk with my relative (a CI user) over the phone several times each month without significant problems. Therefore, I thought the potential of phone conversations being a big issue was negligible as the Facebook group used for recruitment has a relatively large number of CI users.

Immediately after posting the recruitment message on Facebook, a member pointed out that I excluded people who have issues using phones for communication. Unfortunately, this seems to be the case, as out of 1400 members, only one signed up. In addition, the application process of getting the interview approved by NSD took 30 days from sending the application to receiving the approval. Therefore, there was not enough time to re-apply for video calls.

7.4.2 Interview findings

The participant's experience showed that there needed to be more explicit information about knowing subtitles and information. Therefore, I put information about the subtitles as a secondary title within the sound-picture exercise, hopefully making it easier to understand how to toggle the subtitles on or off.

Concerning the instructions dropdown menu, I contemplated moving the button above the other buttons to remove the possibility of the instruction button being pushed to the bottom of the page. However, I decided against it as I think the "choose sound" button should be on the top. So instead, I did the same as done with the subtitles and added text to the secondary header. So now the secondary title of the sound picture exercise reads, "All sounds have subtitles, toggle them by clicking the subtitles button, find more information in the instructions tab.” Hopefully, this makes it more intuitive.

The positive feedback given was nice to hear. It also indicates that the training platform is a good fit for the problem (question 4 in Table 1).

7.5 Performance tests

7.5.1 Low client count with sequential requests

The results seen throughout Figure 54 to Figure 57 show that the average throughput increases as new nodes are introduced to the cluster. The four-node Raspberry Pi (RPI) cluster consistently has the best performance compared to the other configurations. For example, the JSON conversion throughput at eight clients spamming 1000 requests is significantly better on the four-node cluster than on the single node server (Figure 55). The increase in performance is not as big for the rhyming endpoint but has a significant performance increase by a factor of three (Figure 56). The endpoints relating to file lookups are dependent on reading speeds. As a result, the throughput for the four-node RPI cluster is better, but the performance increase is not as significant (Figure 57).

The results indicate that the load is successfully distributed amongst the nodes, and the system can harness the granular power of each RPI. However, when introducing intercommunication, the throughput is strangled and is seen to be performing worse than the single node server (black line, throughout Figure 54 to Figure 57). Therefore, utilizing pings to check for node liveness before delegating work is not viable for detecting a failure at the level of a worker node.

7.5.2 High client count with asynchronous requests

Figure 60 shows that as the number of clients increases, the throughput of the single node RPI server is very stable, and the number of dropped requests seen in Figure 61 follows a linear pattern for the file lookups and rhyme endpoint. In contrast, the landing page endpoint did not drop a single request. The single-node RPI server performs well at all endpoints except the rhyme finding endpoint at these loads due to the rhyme finding algorithm being a compute-heavy operation. When comparing the single node RPi server to the other configuration, we see that for the compute-heavy operations, the cluster configurations perform better in Figure 66 and Figure 68.

In contrast to what was seen in the sequential testing, the single node RPi server performs better for file lookups, indicating that the clustered version handles the sequential load from fewer clients better than the asynchronous load. Both the RPi and LKE master nodes struggle to keep up with the increase of concurrent clients at read-heavy loads. Figure 67 shows both cluster configurations handling the read-heavy load worse than the single node RPi server. The LKE cluster performs better than the RPi cluster for file lookups until the file sizes increase to 800KB. Figure 71 and Figure 72 show that the two cluster configurations have similar lookup performance when the file sizes increase to 800KB and bigger.

Figure 73 shows that the four-node RPI cluster performs the best for the stress test, with the LKE cluster and single node RPi being almost equal in performance. The Four node RPi cluster has the best stress test performance as it performs averagely better throughout the tests.

When comparing the performance from the four-node RPi cluster with intercommunication to the other configurations, it is clear that pinging for node liveness is not a viable failure detection method.

7.5.3 Test finding summary and possible solutions

The test results show that the four-node RPi cluster performs better at compute-heavy workloads (index and rhyme endpoints). This is because it can distribute the load effectively throughout the nodes within the cluster. However, the file lookups test results show that the load-balancing loses effectiveness as the time to handle requests increases.

At the very least, distributing load while detaching the master from the master by utilizing redirects was sufficiently efficient for distributing CPU heavy loads. However, when the RPI cluster comes under read-heavy operations that take a longer time to handle, the cluster can no longer distribute the load as efficiently. The efficiency loss is why the single node RPI server and LKE deployment perform better in Figure 67 and Figure 69, and Figure 70. Under the lookup loads, the four-node RPi master node would get overwhelmed and would not be able to distribute the redirects fast enough to balance the reads amongst the cluster effectively. This effect doubled as intercommunication where introduced. Pinging a node for each request doubles the number of requests the system needs to handle and strangles throughput. Moreover, it is not a good way to introduce fault detection to the cluster. Trying to detect failed nodes by pinging to ensure node liveness is not a viable solution.

While utilizing redirects did do the intended job of detaching the master from the response, it has the side effect of increasing the number of requests the router must handle by a factor of two and a factor of three as the intercommunication is introduced. This is because one additional request for redirection is added for every request, and if intercommunication is enabled, one additional request is needed for pinging. As the workload increased, the router struggled to keep up, and in some cases, the router would reset itself as it could not keep up with the number of simultaneous requests.

Interestingly, the single node RPi server kept up with the clustered deployments and was consistent throughout the testing. The impressive performance most likely comes from the fact that no redirects are utilized for the single node server. The only requests that the single node server dropped were due to operating system (OS) errors complaining that too many files weres being opened simultaneously. It is safe to say that the tests could push the single node server to the limit of the underlying hardware, whereas the clustered were pushed over its limit as it could not distribute the read-heavy operations quickly enough. Additionally, the clustered version dropped requests due to connection issues indicating that either the master or the router was overworked and unable to handle the requests.

Instead of pinging to detect a failed worker node, a second thread should be introduced, which could ping worker nodes at a set interval and remove them from the node pool if unreachable. Implementing the second thread would avoid pinging and lower the number of requests generated.

Utilizing redirection did do its intended job of detaching the master from the response, but it floods the system with additional requests, introducing problems elsewhere. Having a more powerful router capable of handling large numbers of requests might make a difference concerning cluster performance. Although this load distribution method induced some issues, we have seen that the concept works at lower loads.

Utilizing reverse proxies, as explained within chapter 5.4.2, to be able to contact the RPIs from the outside world may have influenced the performance. Although this may be true, the same tendencies were seen while testing inside the local area network. Additionally, the round trip times compared to the LKE deployment were lower.

Caching the most recent audio files should be implemented to reduce the read operations needed as lookup requests are received. Input/output (IO) operations are expensive, and caching would decrease the need for IO operations. In addition, implementing caching would more than likely result in a significant performance increase for the read-heavy loads.

7.6 Solution

7.6.1 Final deployment

Throughout chapter 7.5, there was evidence showing that single-board computers such as RPi's can be used as deployment platforms as the performance was impressive. However, there is more to a real-life deployment than just plain hardware. Using single-board computers to deploy the back-end will result in a compromise. On the one hand, you get the affordability to avoid monetary barriers to entry. On the other hand, there is no way to provide the availability and reliability requirements outlined in chapter 3.5. It is impossible to do so as the single-board computers would be physically placed within my home, meaning there would be no guarantees against power outages and unforeseen circumstances. Therefore, since the compromise would go against the requirements outlined in chapter 3.5, and there are no realistic ways to improve or fix the issues standing in the way of the requirements, the deployment that will be continued is the "attached" design utilizing Linode Kubernetes Engine.

The LKE cluster is stable and performs as expected. It rarely dropped requests except at the highest workloads. It may not consistently outperform the RPi configurations, but having automatic failure handling, ease of update, and availability from using a Kubernetes engine is priceless.

The RPI cluster cannot achieve the availability and reliability requirements stated in chapter 3.5. The LKE can provide these requirements "out of the box," and you do not have to do any additional development. Utilizing the LKE deployment means not dealing with reverse proxy servers and firewall issues. Even if I were to move, I would still have to deal with the ISP to get a static IP address, if even the available ISP allows private users to sign up for one. RPi's are incredibly effective for their size but dealing with the outside issues of using one as a deployment platform can be exhaustive.

Since the number of expected users is not as high as tested for within the performance tests, the number of nodes within the LKE cluster will be reduced to one node. Therefore, lowering the cost from 30 USD each month to 10 USD. Additionally, the domain name lyttetrening.no was purchased for a yearly fee of 12 USD (120 NOK) to make the URL easier to remember and the training tool more reachable. The final deployment annual fee is a total of 132 USD.

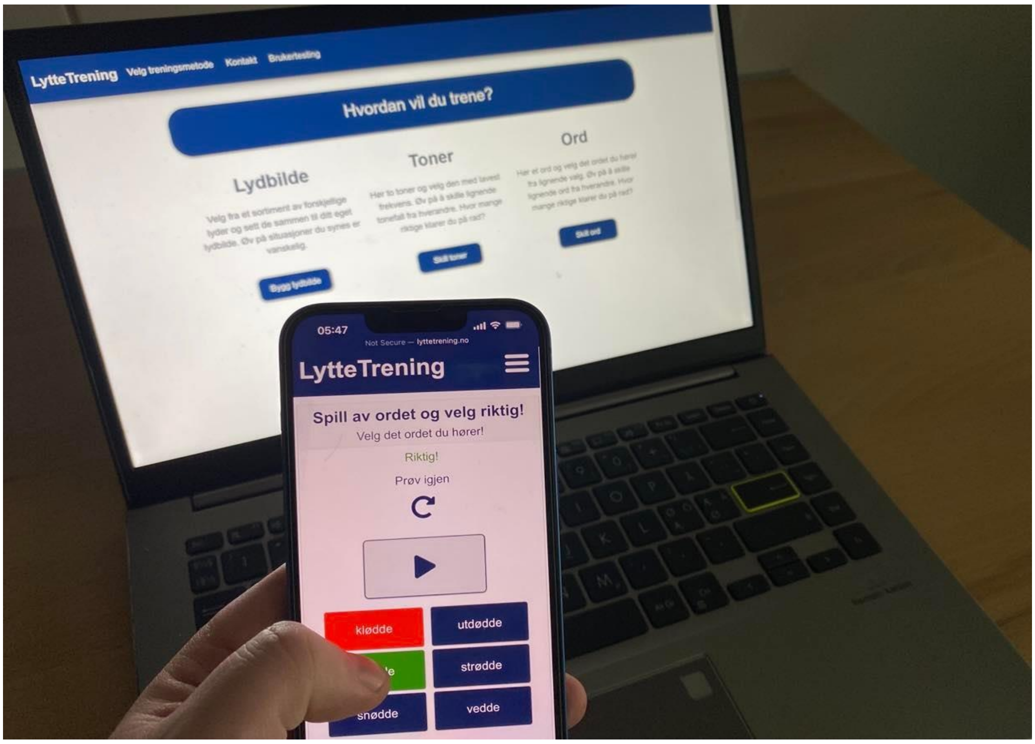


Figure 75: Mobile and desktop version of Lyttetrening.no (final deployment)

7.6.2 Changes during development

In the project's beginning stages, the initial idea was to design the front-end within Flutter [119]. Flutter is a Dart [120] framework that seeks to create web, iOS, and Android applications from one codebase. It was decided against because the framework was more complex than initially thought. Instead, I decided to use something more familiar and landed on Python with Flask. The development process was more efficient by switching to Python and using Flask. Python and flask were both a language and a framework that I had experience with from previous projects. By ensuring that the web application is responsive and viewable through a smart device, the web application can later be wrapped into native applications for each platform (IOS, Android).

I initially intended to do load balancing by measuring the CPU load of each node within the system and distributing the request towards the nodes with the lowest amount of current CPU load. However, getting an accurate measurement of CPU load is not something that can be done hastily, and most method of getting an accurate measurement of CPU load takes at least a second. Therefore, the introduced round-robin turned-based scheduler was straightforward but effective.

The biggest challenge of the tone separation feature was generating and controlling the different tones. I considered pre-recording different tones with a set increment between them. Then, the tones would be fetched from the back-end to function similarly to the other features. The initial solution was partially implemented by following this school of thought. However, this was overcomplicated and programmatically static.

Furthermore, it involved a great deal of manual labor in creating the sounds. Therefore, this solution was abandoned in favor of a solution that would programmatically generate sound with JavaScript. Creating the sounds turned out to be a simpler and better solution, where there was no need for any extra manual labor. In addition, the tone separation feature functions entirely without the need for any calls to the back-end due to not being dependent on stored files.

7.6.3 Security

There should be no real risk concerning security threats posed toward the system's users. First, there are no threats as users do not provide input other than controlling content. Therefore, there should be no risk involving them either. Furthermore, since the LKE deployment handles all cluster operations through the Kubernetes framework, including updating files, there should be no realistic security threats other than protecting the Linode authentication credentials. Therefore the final deployment using LKE is considered secure.

For the RPI cluster, there are several security risks. First, malicious third parties who are trying to breach the admin panel. The admin panel within the RPI cluster deployment contains information about the cluster, options to add and remove nodes for scaling, and adding and removing files to insert them into a running cluster. These features are essential to secure if the RPi version should be used in production environments. Additionally, authorization and encryption are currently leveraged to provide security. However, these functionalities have not been thoroughly vetted.

Additionally, the functionalities relating to the adding and removing files and the updating messages sent from the cluster’s master to new nodes for scaling are not subject to any authorization. As a result, it will perform its functionalities independent of who is sending the requests. The intercommunication must be secured before utilizing the fileserver within a production environment.

7.7 Strengths

The solution is tailored toward post-lingually deafened CI users and created based on input from CI users. CI users have been involved in the entire process from start to finish, from participating in questionnaires to providing feedback through semi-structured interviews.

The solution is deployed and released as a digital training platform readily available to users. The solution has audio training exercises that enable users to perform audio training without the need of any third party and focuses on repeatable exercises. In addition, the system provides statistical feedback to the users on their performances.

A hierarchical fileserver capable of storing and managing files was created during the thesis. It can convey its data while still maintaining the hierarchical structure of the underlying filesystem. It can be used in applications needing to display hierarchical structure, such as within a folder, as long as it is configured as a stand-alone monolithic with all cluster functionalities disabled.

Novelty

The sound picture feature is considered a Novelty. Being able to choose from a digital sound collection and layer them as you wish is something I have not seen in similar tools.

In addition, gathering sound pictures, phonemes, synthetic and musical exercises into one readily available digital training platform providing repeatable exercises tailored toward Norwegian-speaking CI users is also considered a novelty. Table 10 shows how the different digital training tools compare to the digital training platform created during this thesis.

Table 10: Comparison between thesis result (lyttetrening.no) and similar digital tools.

Tools	***Browsable sound catalog with the possibility of layering sounds***	***Phoneme exercises***	***Synthetic exercises***	***Musical content or exercises***	***Repeatable exercises, with no need for third parties***	***Audio-visual queues***
CI – hva du hører [15]		x	x	x	x	Subtitles
Lyttetrening etter CI [16]						
Meludia [19]				x	x	
Lyttetrening.no	x	x	x	x	x	Subtitles and waveforms

7.8 Limitations

Utilizing the wavesurfer.js library for generating and displaying waveform data means that the system loads the same audio file twice—one time within the video tag and one additional time within the wavesurfer.js player. The wavesurfer.js has implemented its framework to be a standalone audio player based on the HTML5 audio tag, with the addition of generating and displaying waveform data. Loading the data twice could be avoided by using the HTML5 video tag for subtitles and the wavesurfer.js player as an audio source. In addition, users may utilize data roaming to use the digital training platform, so a reduction in total bandwidth may be needed.

While using the word separation feature, there will eventually be some weirdly voiced words due to how Google text-to-speech works. The text is transformed into audio by a machine learning algorithm that takes text as input and converts it into audio. I would argue that Google's text-to-speech API is the best framework for Norwegian-sounding speech compared to similar options. But there is the possibility of words sounding weird or just plain wrong. Weirdly sounding words may make the digital training platform responsible for false teaching. The digital training platform is intended for post-lingually deafened CI users, but the potential is still there. If it were not for the text-to-speech framework, the exercise would not be created in this dynamic manner and probably would not exist. It would have to be developed more manually, where each word was recorded by utilizing manual labor. It may be worth it in the future, but it would be an unrealistic expectation within the scope of this thesis. I would argue that the potential risk is worth it, but if the false teaching is greater than the benefits from the training exercise, the exercise would have to be redesigned.

7.9 Future work

7.9.1 Security

The intercommunication must be subject to some authorization and encryption schema. Nodes must authorize the incoming request related to the intercommunication to ensure that the requests come from inside the cluster and not from a malicious third party. The nodes should only handle authorized requests. This could be done by utilizing SSL tunnels for encryption and only accepting requests from known sources. Although the cluster's intercommunication seems secure by implementing such features, the cluster may still be vulnerable to other threats such as IP spoofing, where a malicious third party will create fake packets containing a false IP address. Ensuring that no third party can insert malicious operations should be subject to further investigation and must be implemented before using the fileserver in production.

Furthermore, the traffic sent between the client and server should also be encrypted using SSL or TLS. However, using self-signed certificates is not a good practice when doing so. First, the website appears malicious when users visit the site, which may scare users away. Secondly, browsers don’t like self-signed SSL certificates when using content from other sources. For example, the Safari browser will not allow content from sources with self-signed certificates, while other browsers seem inconsistent with CORS handling [106]. Therefore, if the final deployed system uses SSL or TLS, it should use a validated certificate to remove any problems relating to self-signed certificates.

7.9.2 Fault tolerance

Fault tolerance is something that has not been implemented yet. A worker node can fail without totally crashing if the intercommunication pings are enabled. If a worker node is down, it will be none responsive to the master’s ping. If a pinged node does not respond, the master will do the work instead. If intercommunication is turned on, the master can detect that a worker node is down, but it will not take any steps to remove the node from the cluster or do anything to recover it. A side effect is that possible overworked or downed nodes can become functional worker nodes if the issue should resolve itself. This failure detection method was seen to have poor performance and should be revised in the future. Implementing an independent thread pinging neighbors on a set interval could avoid the problem seen with pinging. Removing unresponsive nodes from the thread pool to avoid delegating work to them could be a better way to introduce failure detection.

A failure of the master node within the RPI clusters would result in total system failure, as there is no re-election algorithm in place. Implementing a re-election algorithm to choose a new master would fix this problem. A possible solution could be some version of the bully algorithm [83].

Additionally, master failure during scaling will result in the new node being unable to join the rest of the cluster, as the new node is unable to contact the crashed master. There are some options here. Shifting the responsibility over to the new node would be one potential solution. Having the new node notify all nodes within the cluster instead of the master would fix this.

7.9.3 Persistent storage for users

In the future, the system should be able to provide users with more statistics and information on how the users are progressing. For example, offering users confirmation that the training work and that they are performing better than in previous attempts may be a good motivator for continuous training. The data could also be analyzed to see what exercise gives the most progression. Analyzing this data could provide valuable insights into the efficiency of each exercise. Such a system would need some form of login system backed up by a persistent database to keep track of the individual's login information and performances.

7.9.4 Creating native applications

During the thesis, I found no single preferred platform from the questionnaire participants. This was one of the main contributing factors to why the solution was designed to be a web-based application reachable on any device if a browser is present. In the future, there should be native applications for iOS and Android devices available through their respective app stores. The main reason for this is that it is easier to use as you do not have to remember an URL, and the app is present on your phone, which circumvents the need for browsers and reduces the number of steps a user needs to take before reaching the training platform.

Since there is already a website solution present, the steps taken to create a native application would entail wrapping the website within a native solution. There would be a native application pulling the website's content and displaying it in a native application.

7.9.5 Extending the features already implemented

The sound library should be extended to include more recordings of different content. There should be more recordings of other sentences and conversations. Right now, there are only three different dialects. Ideally, it should be more

There should also be more exercises included within the digital training platform, and there are several good exercises to draw inspiration from within the audio therapist handbook [12]. One example that could be implemented is when a CI user is prompted by a person reciting two different words but repeating one. The CI user must choose the word that was repeated. Most of the exercises listed within the audio therapist handbook would be relatively easy to implement as most of the infrastructure that must be present to implement them already exists. For example, one could rework the word separation feature to implement the exercise mentioned above.

Ideally, the users should be able to choose what words they want to use as content for the exercise within the word separation feature. Or be able to focus on words starting with specific letters. Most CI users have different frequencies they find the most challenging. The tool should accommodate these differences, similar to what was done for the tone separation feature.

The tone separation feature should also be extended to have more training modes. For example, introduce rhythmic sounds or have sounds resembling chords.

Previewing or being able to hear a small portion of a sound before adding it to the audio player should be implemented. It is a quality-of-life feature that could make it easier for users to find the correct sounds.

Lyttetrening.no works as intended for Google Chrome, Safari, Microsoft Edge, and Opera. Unfortunately, Firefox has issues with reloading subtitle tracks. Ideally, it should work for the most popular browsers, meaning Firefox-related issues should be fixed.

8 Conclusion

This thesis aimed to create a digital audio training platform tailored for post-lingually deafened Norwegian-speaking CI users. To ensure that the solution was fit for the problem (question 4 Table 1), CI users were involved throughout the thesis. Two questionnaires, a literature review, and the reported experience from a user representative helped craft the requirements of the audio training platform. It was decided that the best way to offer the training platform was to create it as a web application since there was no singular preferred platform from the questionnaire participants.

Questionnaire participants were asked what they would use as audio training content. Their answers, along with the main findings of the literature review, showed no singular “one size fits all” solution. Therefore, conversations with fellow students, story-based texts, and poems were recorded to offer conversational recordings as audio training content. The conversational recordings were added to a collection of sounds containing different realistic environmental sounds. The sound picture exercise lets users layer different sounds to simulate challenging scenarios. The word separation exercise offers singular words, and the tone separation exercise offers frequency separation training. Hopefully, the repeatable exercises containing different content and training methods maximize the training outcomes (question 3 Table 1).

A cost analysis of renting different cloud computing services was performed to find the cheapest options for deployment. The investigation found that Linode Kubernetes Engine was one of the most cost-effective cloud provider services. Additionally, the Raspberry Pi model B platform was an exciting option. Although some potential problems were raised, the single-board computer platform's cost and power-effective nature and its relatively good performance seen from preliminary testing justified further research.

Evidence pointed out that Raspberry Pi’s may be viable deployment options for systems with few expected users during performance testing. However, it was found that the Raspberry Pi platform could not provide the non-functional requirements outlined for the solution. Therefore, the deployment chosen as the final solution was the LKE cluster, costing an annual fee of 132 USD, which is sufficiently cheap to avoid any monetary barriers to entry (question 2 Table 1).

A digital training platform focusing on speech, tonal and pitch perception without needing assistance from third parties has been offered to CI users (question 1 Table 1). This was offered by developing a tailor-made solution and deploying it via Linode Kubernetes Engine.

The digital training platform can be reached by visiting www.lyttetrening.no

9 References

[1] T. Lenarz, "Cochlear implant - state of the art," (in eng), *GMS Curr Top Otorhinolaryngol Head Neck Surg,* vol. 16, p. Doc04, 2017, doi: 10.3205/cto000143.

[2] Helse Norge. "Cochleaimplantat, førstegangsoperasjon, voksne." Helse Norge. https://tjenester.helsenorge.no/velg-behandlingssted/behandlinger/ventetider-for?bid=205 (accessed 21.10, 2021).

[3] Statped, "CI - Og hva så?," Statlig spesialpedagogisk tjeneste, www.statped.no, 2014. [Online]. Available: https://www.statped.no/globalassets/fagomrader/horsel/horsel-2/dokumenter/ci---og-hva-sa---rapport-5-2-14.pdf

[4] Hørselhemmedes Landsforbund. "Hva er CI?" Hørselhemmedes Landsforbund,. https://www.hlf.no/horselsinfo/ci-cochlea-implantat/ (accessed 21.10, 2021).

[5] T. Green, A. Faulkner, and S. Rosen, "Computer-Based Connected-Text Training of Speech-in-Noise Perception for Cochlear Implant Users," (in eng), *Trends Hear,* vol. 23, p. 2331216519843878, Jan-Dec 2019, doi: 10.1177/2331216519843878.

[6] Sykehusinnkjøp HF. "Cochleaimplantater med tilhørende utstyr " Sykehusinnkjøp HF. https://sykehusinnkjop.no/nasjonale-avtaler/cochleaimplantater-med-tilhorende-utstyr#hvem-er-leverandorer (accessed 21.03, 2022).

[7] Cochlear Ltd. "Cochlear - Home page." Cochlear Ltd. https://www.cochlear.com/au/en/home (accessed 21.03, 2022).

[8] Medical Electronics. "Med-el - Home page." Medical Electronics. https://www.medel.com/ (accessed 01.02, 2022).

[9] Advanced Bionics AG and affiliates. "Advanced Bionics AG - Home page." Advanced Bionics AG. https://www.advancedbionics.com/com/en/home.html (accessed 21.03, 2022).

[10] B. Fogg, "A behavior model for persuasive design," presented at the Proceedings of the 4th International Conference on Persuasive Technology, Claremont, California, USA, 2009. [Online]. Available: https://doi.org/10.1145/1541948.1541999.

[11] Völter C., Stöckmann C., Schirmer C., and Dazert S., "Tablet-Based Telerehabilitation Versus Conventional Face-to-Face Rehabilitation After Cochlear Implantation: Prospective Intervention Pilot Study," (in eng), *JMIR Rehabil Assist Technol,* vol. 8, no. 1, p. e20405, Mar 12 2021, doi: 10.2196/20405.

[12] Statlig spesialpedagogisk tjeneste (statped). *Norsk tilpassing av deler av Geoff Plants Auditrain - Pedagogs håndbok*. (2007). www.acm5.com: Statlig spesialpedagogisk tjeneste. [Online]. Available: http://www.acm5.com/kompendier/Auditrain-Pedagog.pdf

[13] Hearing Loss Association of America (HLLA). "Geoff Plant, Founder and Executive Director, Hearing Rehabilitation Foundation." HLLA. https://www.hearingloss.org/presenter_bios/geoff-plant/ (accessed 01.04, 2022).

[14] Møller kompetansesenter. *Brukers håndbok*. (2007). www.acm5.com: Statlig spesialpedagogisk tjeneste. [Online]. Available: http://www.acm5.com/kompendier/Auditrain-Bruker.pdf

[15] Statlig spesialpedagogisk tjeneste. "CI - hva du hører." Statlig spesialpedagogisk tjeneste. https://www.acm.no/ci-hva-du-horer/webapp/#forsiden (accessed 21.10, 2021).

[16] Statlig spesialpedagogisk tjeneste. "Lyttetrening etter CI." Statlig spesialpedagogisk tjeneste. https://www.acm.no/lyttetrening/pdf/lyttetrening.pdf (accessed 19.03, 2022).

[17] Statlig spesialpedagogisk tjeneste (statped). "Statped - Homepage." Statped. https://www.statped.no/ (accessed 19.03, 2022).

[18] Arne Kirkhorn Rødvik, "Speech sound confusions in wellperforming adults and children with cochlear implants, measured by repetition of mono- and bisyllabic nonsense words," PhD, Institutt for spesialpedagogikk, University of Oslo, UiO, 2020. [Online]. Available: https://www.duo.uio.no/handle/10852/79178
[19] Meludia. "Meludia - Develop your listening skills through music." Meludia. www.meludia.com (accessed 21.10, 2021).
[20] Amazon. "Start Building on AWS Today." Amazon. https://aws.amazon.com/ (accessed 02.05, 2022).
[21] Linode. "Linode - Cloud Computing Developers Trust." Linode. https://www.linode.com/ (accessed 10.05, 2022).
[22] Linode. "Compute - Linode Kubernetes Engine." Linode. https://www.linode.com/docs/products/compute/kubernetes (accessed 03.05, 2022).
[23] Amazon. "Amazon Elastic Kubernetes Service (EKS)." Amazon. https://aws.amazon.com/eks/ (accessed 02.05, 2022).
[24] Google. "Google Kubernetes Engine." Google. https://cloud.google.com/kubernetes-engine (accessed 02.05, 2022).
[25] Microsoft. "Azure Kubernetes Service (AKS)." Microsoft. https://azure.microsoft.com/en-us/services/kubernetes-service/#overview (accessed 02.05, 2022).
[26] B. B. Joe Beda, Craig McLuckie,. "What is Kubernetes?" Kubernetes. https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/ (accessed 02.05, 2022).
[27] Rancher. "K3s - Lightweight Kubernetes." Rancher. https://rancher.com/docs/k3s/latest/en/ (accessed 02.05, 2022).
[28] Kubernetes. "Minikube start." Kubernetes. https://minikube.sigs.k8s.io/docs/ (accessed 02.05, 2022).
[29] Red Hat. "Red Hat OpenShift." RedHat. https://www.redhat.com/en/technologies/cloud-computing/openshift (accessed 02.05, 2022).
[30] VMware. "VMware Tanzu." VMware. https://tanzu.vmware.com/tanzu (accessed 02.05, 2022).
[31] Amazon. "AWS Lambda." Amazon. https://aws.amazon.com/lambda/ (accessed 21.10, 2021).
[32] Raspberry Pi Foundation. "Raspberry Pi 4." Raspberry Pi Foundation, . https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed 02.05, 2022).
[33] N. J. Schot, P. J. E. Velthuis, and B. F. Postema, "Capabilities of Raspberry Pi 2 for Big Data and Video Streaming Applications in Data Centres," in *Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems*, Cham, A. Remke and B. R. Haverkort, Eds., 2016// 2016: Springer International Publishing, pp. 183-198.
[34] Apache. "Apache hadoop." Apache. https://hadoop.apache.org/ (accessed 21.10, 2021).
[35] A. R. Shrivastava and J. Gadge, "Home server and nas using raspberry PI," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 13-16 Sept. 2017 2017, pp. 2270-2275, doi: 10.1109/ICACCI.2017.8126184.
[36] Python Software Foundation. "Python - Home page." Python Software Foundation. https://www.python.org/ (accessed 25.03, 2022).

[37] Rust. "Rust - A language empowering everyone to build reliable and efficient software." Rust https://www.rust-lang.org/ (accessed 26.03, 2022).
[38] Go. "Go - Documentation." Go. https://go.dev/doc/ (accessed 04.04, 2022).
[39] National Center for Biotechnology Informaiton. PubMed [Online] Available: https://pubmed.ncbi.nlm.nih.gov/
[40] ACM. ACM - Digital Library [Online] Available: https://www.acm.org/publications/digital-library
[41] IEEE. IEEE Xplore [Online] Available: https://ieeexplore.ieee.org/Xplore/home.jsp
[42] Svend Brinkmann, P. Leavy, Ed. *Unstructured and Semi-Structured Interviewing* (The Oxford Handbook of Qualitative Research). Oxford University Press, 2014.
[43] Google. "Google Forms - Get insights quickly, with Google Forms." Google. https://www.google.com/intl/en_en/forms/about/ (accessed 10.05, 2022).
[44] Norsk Senter for Forskningsdata (NSD). "Norsk Senter for Forskningsdata." NSD. https://www.nsd.no/ (accessed 06.04, 2022).
[45] Nettskjema. "Nettskjema - Spørreskjema, påmeldinger og bestillinger." Universitetet i Oslo. https://nettskjema.no/ (accessed 10.05, 2022).
[46] S. R. James Robertson, "Volere Requirements Specification Template," Edition 18 ed. https://www.volere.org/: the Atlantic Systems Guild Limited., 2016.
[47] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business (in English), 2011.
[48] Docker. "Docker documentation." Docker. https://docs.docker.com/ (accessed 01.04, 2022).
[49] Michael Goodrich and Roberto Tamasia, *Introduction to Computer Security: Pearson New International Edition*. Edinburgh Gate, Harlow: Pearson Education Limited, 2013, p. 514.
[50] Microsoft. "Code editing. Redefined." Microsoft. https://code.visualstudio.com/ (accessed 03.05, 2022).
[51] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915-929, 2015/10/01/ 2015, doi: https://doi.org/10.1016/j.chb.2014.10.046.
[52] Creative Commons Org. "Creative commons - Home page." Creative Commons Org. https://creativecommons.org/ (accessed 22.03, 2022).
[53] C. H. Steve Chen, Jawed Karim,. "Youtube - Home page." Youtube. https://www.youtube.com/ (accessed 22.03, 2022).
[54] Ableton. "Ableton - Home page." Ableton. https://www.ableton.com/ (accessed 22.03, 2022).
[55] Tekstforslag. "Tekstforslag - Tekster til alle anledninger." Tekstforslag. https://tekstforslag.com/ (accessed 22.03, 2022).
[56] Gyldendal. "Rimsmia." Ordnet. https://www.ordnett.no/rimsmia (accessed.
[57] Microsoft. "Velkommen til Microsoft Teams." Microsoft. https://www.microsoft.com/nb-no/microsoft-teams/log-in?rtc=1 (accessed 03.05, 2022).
[58] Braun V and Clarke V, "What can "thematic analysis" offer health and wellbeing researchers?," (in eng), *Int J Qual Stud Health Well-being*, vol. 9, p. 26152, 2014, doi: 10.3402/qhw.v9.26152.
[59] Braun V and Clarke V, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77-101, 2006/01/01 2006, doi: 10.1191/1478088706qp063oa.

[60] Mozilla Developer Network (MDN). "HTML5." Mozilla. https://developer.mozilla.org/en-US/docs/Glossary/HTML5 (accessed 03.05, 2022).
[61] V. Shafiro, S. Sheft, S. Kuvadia, and B. Gygi, "Environmental sound training in cochlear implant users," (in eng), *J Speech Lang Hear Res*, vol. 58, no. 2, pp. 509-19, Apr 2015, doi: 10.1044/2015_jslhr-h-14-0312.
[62] Barlow N., Purdy S. C., Sharma M., Giles E, and Narne V., "The Effect of Short-Term Auditory Training on Speech in Noise Perception and Cortical Auditory Evoked Potentials in Adults with Cochlear Implants," (in eng), *Semin Hear*, vol. 37, no. 1, pp. 84-98, Feb 2016, doi: 10.1055/s-0035-1570335.
[63] C. D. Fuller, J. J. Galvin, 3rd, B. Maat, D. Başkent, and R. H. Free, "Comparison of Two Music Training Approaches on Music and Speech Perception in Cochlear Implant Users," (in eng), *Trends Hear*, vol. 22, p. 2331216518765379, Jan-Dec 2018, doi: 10.1177/2331216518765379.
[64] L. Smith, L. Bartel, S. Joglekar, and J. Chen, "Musical Rehabilitation in Adult Cochlear Implant Recipients With a Self-administered Software," (in eng), *Otol Neurotol*, vol. 38, no. 8, pp. e262-e267, Sep 2017, doi: 10.1097/mao.0000000000001447.
[65] Jiam N. T., Deroche M. L., Jiradejvong P., and Limb C. J., "A Randomized Controlled Crossover Study of the Impact of Online Music Training on Pitch and Timbre Perception in Cochlear Implant Users," (in eng), *J Assoc Res Otolaryngol*, vol. 20, no. 3, pp. 247-262, Jun 2019, doi: 10.1007/s10162-018-00704-0.
[66] Vandali A., Sly D., Cowan R., and van Hoesel R., "Training of cochlear implant users to improve pitch perception in the presence of competing place cues," (in eng), *Ear Hear*, vol. 36, no. 2, pp. e1-e13, Mar-Apr 2015, doi: 10.1097/aud.0000000000000109.
[67] G. M. Firestone *et al.*, "A Preliminary Study of the Effects of Attentive Music Listening on Cochlear Implant Users' Speech Perception, Quality of Life, and Behavioral and Objective Measures of Frequency Change Detection," (in eng), *Front Hum Neurosci*, vol. 14, p. 110, 2020, doi: 10.3389/fnhum.2020.00110.
[68] E. Hutter, H. Argstatter, M. Grapp, and P. K. Plinkert, "Music therapy as specific and complementary training for adults after cochlear implantation: A pilot study," (in eng), *Cochlear Implants Int*, vol. 16 Suppl 3, pp. S13-21, Sep 2015, doi: 10.1179/1467010015z.000000000261.
[69] Lo C. Y., McMahon C. M., Looi V., and Thompson W. F., "Melodic Contour Training and Its Effect on Speech in Noise, Consonant Discrimination, and Prosody Perception for Cochlear Implant Recipients," (in eng), *Behav Neurol*, vol. 2015, p. 352869, 2015, doi: 10.1155/2015/352869.
[70] S. Borel, S. Dupré, M. de Bergh, O. Sterkers, I. Mosnier, and E. Ferrary, "Rehabilitation of telephone communication in cochlear-implanted adults," (in eng), *Eur Ann Otorhinolaryngol Head Neck Dis*, vol. 137, no. 5, pp. 381-386, Nov 2020, doi: 10.1016/j.anorl.2020.01.010.
[71] Sousa A. F *et al.*, "Telephone Usage and Cochlear Implant: Auditory Training Benefits," (in eng), *Int Arch Otorhinolaryngol*, vol. 19, no. 3, pp. 269-72, Jul 2015, doi: 10.1055/s-0034-1390301.
[72] Miller S. E., Zhang Y., and Nelson P. B., "Efficacy of Multiple-Talker Phonetic Identification Training in Postlingually Deafened Cochlear Implant Listeners," (in eng), *J Speech Lang Hear Res*, vol. 59, no. 1, pp. 90-8, Feb 2016, doi: 10.1044/2015_jslhr-h-15-0154.
[73] Schumann A., Serman M., Gefeller O., and Hoppe U., "Computer-based auditory phoneme discrimination training improves speech recognition in noise in experienced

adult cochlear implant listeners," (in eng), *Int J Audiol*, vol. 54, no. 3, pp. 190-8, Mar 2015, doi: 10.3109/14992027.2014.969409.
[74] T. Sato, T. Yabushita, S. Sakamoto, Y. Katori, and T. Kawase, "In-home auditory training using audiovisual stimuli on a tablet computer: Feasibility and preliminary results," (in eng), *Auris Nasus Larynx*, vol. 47, no. 3, pp. 348-352, Jun 2020, doi: 10.1016/j.anl.2019.09.006.
[75] World Wide Web Consortium (W3C). "WCAG 2 Overview." W3C. https://www.w3.org/WAI/standards-guidelines/wcag/ (accessed 12.05, 2022).
[76] MySQL. "MySQL - Home page." MySQL. (accessed 24.03, 2022).
[77] PostgreSQL. "PostgreSQL - Home page." PostgreSQL. https://www.postgresql.org/ (accessed 24.03, 2022).
[78] Hector Garcia-Molina, Jeffrey Ullman, and Jennifer Widom, *Database Systems - The Complete Book*, 2nd Edition ed. Pearson Prentice Hall, 2009.
[79] C. Strauch, U.-L. S. Sites, and W. Kriha, "NoSQL databases," *Lecture Notes, Stuttgart Media University*, vol. 20, p. 24, 2011.
[80] MongoDB. "MongoDB The application data platform." MongoDB. www.mongodb.com (accessed 21.10, 2021).
[81] Apache. "CouchDB - Hompe page." Apache. https://couchdb.apache.org/ (accessed 24.03, 2022).
[82] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, 2001, doi: 10.1145/964723.383071.
[83] A. S. Tanenbaum. and Herbert Bos, *Modern Operating Systems*, 4th ed. Pearson (in English), 2014.
[84] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D., University of California, Irvine, Ann Arbor, 9980887, 2000.
[85] A. Rodriguez, "Restful web services: The basics," *IBM developerWorks*, vol. 33, p. 18, 2008.
[86] Flask. "Flask - web development, one drop at a time." Flask. https://flask.palletsprojects.com/en/2.0.x/ (accessed 25.03, 2022).
[87] The World Wide Web Consortium (W3C). "W3C - Leading the web to its full potential." W3C. https://www.w3.org/ (accessed 11.05, 2022).
[88] The World Wide Web Consortium (W3C). "WebVTT: The Web Video Text Tracks Format." W3C. https://www.w3.org/TR/2019/CR-webvtt1-20190404/ (accessed 18.11, 2021).
[89] Thijs Triemstra (Main maintainer of github repository). "Wavesurfer.js - Quick start." wavesurfer.js. https://wavesurfer-js.org/ (accessed 26.03.2022, 2022).
[90] D. Holmes and M. C. McCabe, "Improving precision and recall for Soundex retrieval," in *Proceedings. International Conference on Information Technology: Coding and Computing*, 8-10 April 2002 2002, pp. 22-26, doi: 10.1109/ITCC.2002.1000354.
[91] Google. "Google - Speech-to-text." Google. https://cloud.google.com/speech-to-text (accessed 24.03, 2022).
[92] The Actix team. "Actix - A powerful, pragmatic, and extermely fast web framework for rust." The Actix team. https://actix.rs/ (accessed 26.03, 2022).
[93] Vincent Prouillet. "Tera " https://crates.io/crates/tera (accessed 04.04, 2022).
[94] A. Ronacher. "Jinja2 - Project description." Pypi. https://pypi.org/project/Jinja2/ (accessed 26.03, 2022).
[95] D. Tolnay. "Serde-rs." giithub. https://github.com/serde-rs/serde (accessed 04.04, 2022).

[96] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7 ed. Boston, MA: Pearson, 2016.
[97] Vincent Prouillet. "jsonwebtoken - repository." github. https://github.com/Keats/jsonwebtoken (accessed 26.03, 2022).
[98] Mark V. Lawson, *Finite Automata*, 1st Edition ed. New York: Chapman and Hall/CRC, 2004, p. 320.
[99] Dokumentasjonsprosjektet ved Universitetet i Oslo. "Tekstsamlingen - Ren tekst." Universitetet i Oslo. https://www.dokpro.uio.no/litteratur/txtfiler.html (accessed 02.05, 2022).
[100] Asbjørnsen og Moe. "folkeeventyr.no." Spennende Norske folkeeventyr for store og små https://folkeeventyr.no/ (accessed 05.02, 2022).
[101] Chief editor: Erik Bolstad. "Store Norske Leksikon." Store Norske Leksikon. https://snl.no/ (accessed 04.02, 2022).
[102] Universitetet i Oslo and Språkrådet. Bokmålsordboka [Online] Available: https://ordbok.uib.no/info/om_ordbokene_nn.html
[103] Mozilla Developer Network (MDN). "MDN Web docs - CSS: Cascading Syle Sheets." MDN. https://developer.mozilla.org/en-US/docs/Web/CSS (accessed 26.03, 2022).
[104] Kenneth Reitz. "Requests 2.27.1 - Project description." pypi. https://docs.python-requests.org/en/latest/ (accessed 26.03, 2022).
[105] Bootstrap Core Team. "Bootstrap v5.0 - quick start." Bootstrap. https://getbootstrap.com/docs/5.0/getting-started/introduction/ (accessed 26.03, 2022).
[106] Mozilla Developer Network (MDN). "Cross-origin Resource Sharing (CORS) - Documentation." MDN. https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS (accessed 26.03, 2022).
[107] Mozilla Developer Network (MDN). "AudioContext." MDN. https://developer.mozilla.org/en-US/docs/Web/API/AudioContext (accessed 26.03, 2022).
[108] Mozilla Developer Network (MDN). "AudioNode." MDN. https://developer.mozilla.org/en-US/docs/Web/API/AudioNode (accessed 26.03, 2022).
[109] Benoit Chesneau. " Gunicorn 20.1.0 - Project description." pypi. https://docs.gunicorn.org/en/stable/ (accessed 01.04, 2022).
[110] Ciara Stacke, Luca Comellini, and Stephen Hurwitz. "Nginx, Inc." Github. https://github.com/nginxinc (accessed 13.05, 2022).
[111] M. J. Page *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, 2021, doi: 10.1136/bmj.n71.
[112] H. Dillon, A. James, and J. Ginis, "Client Oriented Scale of Improvement (COSI) and its relationship to several other measures of benefit and satisfaction provided by hearing aids," (in eng), *J Am Acad Audiol*, vol. 8, no. 1, pp. 27-43, Feb 1997.
[113] Pandora Media Inc. "Pandora - Music & Podcasts." Pandora Media Inc. https://www.pandora.com/ (accessed 05.05, 2022).
[114] Cochlear Ltd. "Lydprosessor og app-kompatibilitet." Cochlear Ltd. https://www.cochlear.com/global/no/apps/sound-processor-and-app/compatibility (accessed 02.05, 2022).
[115] Cochlear Ltd. "Cochlear - True wireless devices." Cochlear Ltd. https://www.cochlear.com/no/no/home/products-and-accessories/our-accessories/true-wireless-devices (accessed 02.05, 2022).
[116] M. Ross, "The audiogram: Explanation and significance," *Hearing Loss Association of America*, vol. 25, no. 3, pp. 29-33, 2004.

[117] Nasjonal Kompetansetjeneste for døvblinde. "Talebanan." Nasjonal Kompetansetjeneste for døvblinde, . https://www.dovblindhet.no/talebanan.556809.no.html (accessed 02.05, 2022).
[118] Wikipedia. "Image of audiogram called the "Speech banana"." Wikipedia. https://en.wikipedia.org/wiki/Speech_banana#/media/File:Speech_Banana_no_icons.png (accessed 02.05, 2022).
[119] Google. "Flutter - Build apps for any screen." Google. https://flutter.dev/ (accessed 29.03, 2022).
[120] Google. "Dart - Paint your UI to life." Google. https://dart.dev/ (accessed 29.03, 2022).

Appendix I.

Information – Preliminary questionnaire

Jeg er masterstudent ved universitetet i Tromsø, og skal etter sommeren begynne å skrive masteroppgave i informatikk. Min masteroppgave er å lage en lyttelærings applikasjon for personer med CI. Applikasjonen skal ha som mål å tilby mengdetrening på lyder, den skal være brukervennlig og tilgjengelig for alle. I den anledning ønsker jeg å spørre gruppe-medlemmene i CI-gruppen, om de kan være med å hjelpe meg med å spesifisere hva som er viktig å tenke på når man lærer å lytte igjen. Med deres hjelp kan jeg skreddersy et verktøy som bedre passer de behov en har etter en CI operasjon.

Appendix II.

Questions – Preliminary questionnaire

Jeg har gjennomført CI operasjon – radioknapper

- Ja
- Nei

Jeg er, eller svarer på vegne av (en/et) – radioknapper

- Voksen
- Ungdom
- Barn

Hvilke platform bruker du mest - (avkrysningsbokser)

- Datamaskin
- Android (telefon, nettbrett osv.)
- iOS (telefon, nettbrett osv.)

Dersom jeg skulle fått mest mulig utnytte av en lyttelærings applikasjon, måtte den hatt fokus på: - radioknapper

- Stemmer/ord - med forskjellige dialekter
- Spesifikke lyder

Dersom du har anledning, nevn lyder eller ord som var eller som du tror kommer til å bli ekstra vanskelig. (feks: torden, motorlyder, stemmer osv.) – tekstfelt

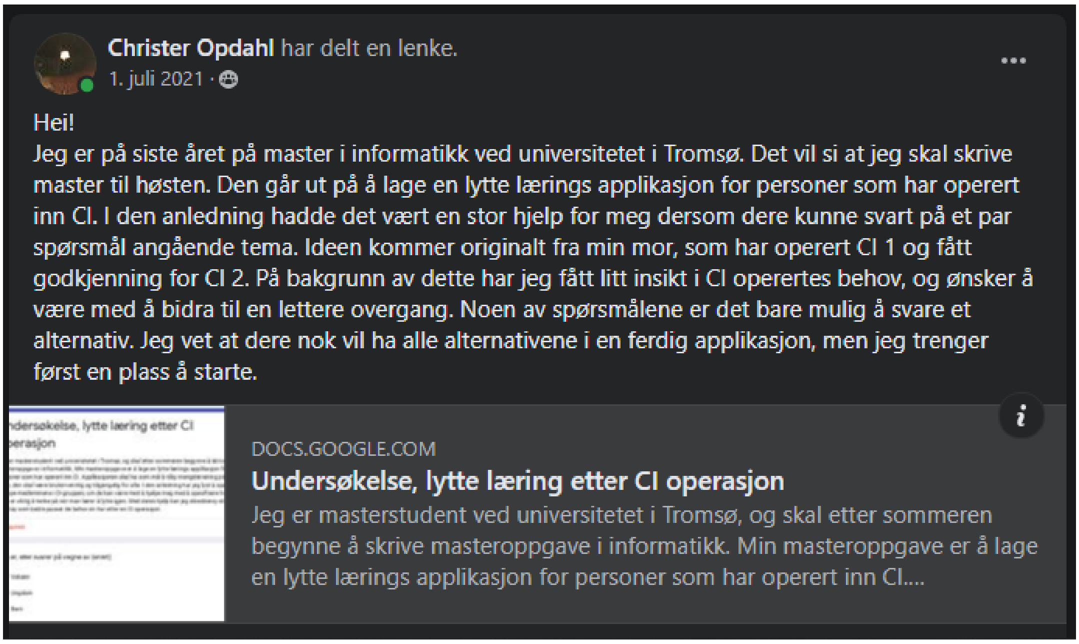
Dersom du har anledning, nevn situasjoner som var eller som du tror blir spesielt utfordrene. (feks: blant folkemengder, på restaurant osv.) – tekstfelt

Appendix III.

Recruitment message – Preliminary questionnaire

Hei!

Jeg er på siste året på master i informatikk ved universitetet i Tromsø. Det vil si at jeg skal skrive master til høsten. Den går ut på å lage en lytte lærings applikasjon for personer som har operert inn CI. I den anledning hadde det vært en stor hjelp for meg dersom dere kunne svart på et par spørsmål angående tema. Ideen kommer originalt fra min mor, som har operert CI 1 og fått godkjenning for CI 2. På bakgrunn av dette har jeg fått litt insikt i CI operertes behov, og ønsker å være med å bidra til en lettere overgang. Noen av spørsmålene er det bare mulig å svare et alternativ. Jeg vet at dere nok vil ha alle alternativene i en ferdig applikasjon, men jeg trenger først en plass å starte.



Appendix IV.

Information – Primary questionnaire

Vil du delta i forskningsprosjektet "Undersøkelse angående lyttetrening for personer med Cochleaimplantat"?s

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å få en bedre forståelse over behovene personer med Cochleaimplantat har når det kommer til lyttetrening. I dette skrivet gir vi deg informasjon om formålet med prosjektet og hva deltakelse i undersøkelsen vil innebære for deg.

Formål

Undersøkelsen er en del av en masteroppgave, som går ut på å utvikle en løsning for mengdetrening på lyd for personer med Cochleaimplantat. Løsningen skal være lett tilgjengelig i form av en applikasjon på telefon og nettleser, og den skal tilby realistiske lyder med et fokus på trening.

Formålet med undersøkelsen er å få en bedre forståelse over behovene personer med Cochleaimplantat har når det kommer til lyttetrening. Informasjonen som blir samlet inn vil være med på å påvirke funksjonaliteten i applikasjonen og resultatet av masteroppgaven.

Med lyttetrening menes det at man aktivt trener på gjenkjenning av spesifikke lyder der målet er å bli bedre til å separere lydbilder, gjenkjenne ord eller tonearter.

Hvem er ansvarlig for forskningsprosjektet?

Prosjektet gjennomføres på UiT Norgess arktiske universitet, institutt for Informatikk, og veiledes av Professor Eirik Årsand og universitetslektor André Henriksen.

Hvorfor får du spørsmål om å delta?

Du har fått spørsmål om å delta fordi du er medlem av Facebook gruppen "CI-Gruppa", som er den største Facebook gruppen for personer med Cochleaimplantat i Norge.

Hva innebærer det for deg å delta?

Dersom du velger å delta i prosjektet, innebærer det at du fyller ut et spørreskjema. Det vil ta deg ca. 5 minutter. Spørreskjemaet inneholder spørsmål om dine behov knyttet til lyttetrening. Spørsmål som hvordan, hvor og hvor ofte du utfører lyttetrening samt dine preferanser til innhold i lyttetreningen.

Det er frivillig å delta

Det er frivillig å delta i prosjektet, ingen spørsmål er direkte identifiserende. Dersom du velger å delta, har du rett til å trekke samtykket tilbake når som helst uten å oppgi noen grunn, dersom vi kan identifisere deg (du må i så fall sende oss en eksakt kopi av din besvarelse for at vi skal kunne identifisere deg). Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene du gir oss til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Ingen spørsmål er direkte identifiserende.

- Undersøkelsen utføres via nettskjema.no, som ikke innhenter identifiserende informasjon som IP-adresser eller annet. Den innsamlede informasjonen er beskyttet med 2-faktors autentisering.

- Bare de som er nevnt som ansvarlig for prosjektet vil ha tilgang til informasjonen som blir innsamlet.

- Vi samler **IKKE** inn informasjon som navn, adresse, kjønn, alder, seksualitet eller annen identifiserbar informasjon.

Du vil ikke kunne bli gjenkjent i masteroppgaven, der informasjonen blir sammenstilt og presentert. Informasjonen som blir innhentet, vil bli presentert som grafer eller figurer satt i en større sammenheng.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Alle opplysninger slettes når prosjektet avsluttes, som etter planen er 15. Mai 2022. Det vil da bare være de anonymiserte dataene som blir presentert i masteren.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

Dine rettigheter

Så lenge du kan **identifiseres** i datamaterialet (du må i så fall sende oss en eksakt kopi av din besvarelse for at vi skal kunne identifisere deg), har du rett til:

- Innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene

- Å få rettet opplysninger om deg som er feil eller misvisende

- Å få slettet personsopplysninger om deg

- Å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

- Christer Hagenes Opdahl: cop003@uit.no
- Hovedveileder og Professor Eirik Årsand: eirik.arsand@uit.no
- Biveileder og Universitetslektor André Henriksen: andre.henriksen@uit.no
- Personvernombud UiT: personvernombud@uit.no

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på e-post (personverntjenester@nsd.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Eirik Årsand

André Henriksen

Christer Hagenes Opdahl

Appendix V.

Questions - Primary questionnaire

- Jeg har gjennomført CI-operasjon (obligatorisk, radioknapper)
    - Ja
    - Nei

- Hvor ofte utfører du lytte-trening? (obligatorisk, radioknapper)
    - En eller flere ganger i uken
    - En eller flere ganger i måneden
    - En eller flere ganger i året
    - Aldri

- Hvorfor utfører du ikke lytte-trening? (obligatorisk, avkrysningbokser)
    - Jeg har ikke behov for det, gjorde det før
    - Jeg har ikke behov for det, aldri hatt behovet
    - Jeg har ikke tid
    - Det er for dårlig tilbud
    - Det er for langt å reise
    - Jeg vet ikke hvordan, for lite informasjon rundt tema
    - Jeg har ikke kommet i gang med det enda
    - Annet - Forklar kort: (ikke obligatorisk, tekst felt)

- Hvor utfører du lytte-treningen? (obligatorisk, avkrysningbokser)
    - Sammen med audiopedagog eller annen ekspert
    - Hjemme, men følger retningslinjer eller program satt opp for meg av audiopedagog eller annen ekspert
    - Hjemme, på eget initiativ
    - Annet - Forklar kort: (ikke obligatorisk, tekst felt)

- Hvor mye tid bruker du på å reise til audiopedagog eller annen ekspert? (ikke obligatorisk, tekstfelt)

- Hva er innholdet i lytte-treningen når du øver på eget initiativ (forklar kort) (ikke obligatorisk, tekstfelt)

- Hva er innholdet i lyttetreningen når du øver hjemme og følger oppsatt program fra audiopedagog eller annen ekspert (forklar kort)

- Jeg er fornøyd med de lytte-trenings tilbudene jeg har tilgjengelig. (obligatorisk, radioknapper)
    - Helt uenig
    - Litt uenig
    - Vet ikke
    - Litt enig

- Helt enig
- Jeg utfører ikke lytte-trening

- Korte lydklipp (enkle ord) vil være bedre enn sammensatte setninger. (obligatorisk, radioknapper)
  - Helt uenig
  - Litt uenig
  - Vet ikke
  - Litt enig
  - Helt enig
  - Jeg utfører ikke med lyttetrening

- Opptak av en person som snakker om et tema på sin egen dialekt, er bedre enn at personen leser fra et dikt eller annen tekst. (obligatorisk, radioknapper)
  - Helt uenig
  - Litt uenig
  - Vet ikke
  - Litt enig
  - Helt enig
  - Jeg utfører ikke med lyttetrening

Appendix VI.

Correspondance to DPO – Primary questionnaire

Hei Christer,
Flott at du tenker på personvernet knyttet til spørreundersøkelsen.

Personvernregelverket kommer bare til anvendelse dersom du skal behandle personopplysninger. Med personopplysninger menes enhver opplysning om en person som kan identifiseres direkte eller indirekte, se personvernforordningen art. 4 nr. 1. Det skal lite til for at behandling av opplysninger om personer ikke er anonyme. Man må ta i betraktning alle hjelpemidler som med rimelighet kan tenkes brukt for å identifisere vedkommende. Spørsmålene i spørreskjemaet er relevante å vurdere, men dette må også ses i sammenheng med øvrige prosjektdata, målgruppen, hvilke hjelpemidler som skal brukes og fagområdet. Det vil si at du og din veileder er nærmest til å gjøre en vurdering om behandlingen sin helhet er anonym eller ikke. For eksempel hvor mange er det i denne målgruppen, ungdom og med Cochleaimplantat eller som har gjennomført CI-operasjon?

Fint at du skal bruke Nettskjema. Nettskjema kan blant annet brukes ved anonyme undersøkelser.

Når det gjelder selve spørreundersøkelsen, så kan jeg ikke se spørsmål som utelukker at undersøkelsen er anonym, med forbehold om at jeg ikke kjenner til selve målgruppen. Fint at det ved fritekst bes om korte svar og at det er tydelig at svarene ikke skal inneholde personidentifiserende opplysninger.

Så skriv ned en vurdering av anonymiteten i undersøkelsen og ta vare på vurderingen som dokumentasjon. Om dere ender opp med tvil hvorvidt behandlingen er anonym eller ikke, så vil jeg anbefale at dere legger til grunn at det behandles personopplysninger og melder prosjektet til NSD for vurdering.

Med vennlig hilsen

[redacted](fjernet pågrunn av personvernshensyn)
[redacted](fjernet pågrunn av personvernshensyn)

UiT Norges arktiske universitet

[redacted](fjernet pågrunn av personvernshensyn)

Appendix VII.

Analysis of anonymity - Primary questionnaire

After consulting with the privacy representative at UiT - Norges arktiske universitet, we agreed that an analysis of anonymity was needed. This step is vital to evaluate the grounds for an application to NSD (Norsk Senter for Forskningsdata). The main concern is the target demographic and the aggregate of the data collected.

Target demographic:
The target demographic is young adults from around 18-16 years of age who have undergone a cochlear implant operation, uni or bilateral. The age restriction stems from related cognitive issues.

The questionnaire will be posted in the most popular Norwegian cochlear implant community-driven Facebook group. The questionnaire will be posted with the consent of the group leaders, and the questionnaire itself is voluntary.

The Facebook group is an active group that has over 1000 members. We, therefore, deem the group large enough to negate the risk of identification due to the small sample size if the aggregate data is also deemed non-identifiable.

The aggregate data:
To avoid gathering information relating to email and IP addresses, we will utilize nettskjema.no, designed to avoid doing so.

The text fields within the questionnaire are the most significant risk of direct identification of individuals through individuals oversharing personal information. This is negatable by the introductory text and specific encouragement to not do this. We will also suggest that any comments or questions be sent to a personal email and not submitted in the text fields.

All questions within the questionnaire are related to the use and need of their cochlear implant. Therefore, each question is deemed to be non-identifiable. In addition, there are no questions related to gender, sexuality, or any personal traits that might be identified within an aggregate of information.

Through the questionnaire, we might get information about an individual who has undergone a cochlear implant operation. The individual does audio training several times a year, using 50 minutes to travel to and from audio training specialists. The individual agrees with all the statements concerning the proposed contents of the new audio training tool.

We deem all this information non-identifiable if individuals are not oversharing in text fields. We could potentially remove the text input fields, but with a significant risk of losing valuable input, which can better the new audio training tool and the thesis. In addition, the subjects related to the text input fields are too broad to translate into non-identifiable checkboxes.

Appendix VIII.

Approval from NSD – Primary questionnaire

Behandlingen av personopplysninger er vurdert av NSD. Vurderingen er:

Det er vår vurdering at behandlingen vil være i samsvar med personvernlovgivningen, så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet 05.11.2021 med vedlegg. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige personopplysninger, særlige kategorier av personopplysninger om helseopplysninger frem til 01.06.2022.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 nr. 11 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse, som kan dokumenteres, og som den registrerte kan trekke tilbake.

For alminnelige personopplysninger vil lovlig grunnlag for behandlingen være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 a. For særlige kategorier av personopplysninger vil lovlig grunnlag for behandlingen være den registrertes uttrykkelige samtykke, jf. personvernforordningen art. 9 nr. 2 bokstav a, jf. personopplysningsloven § 10, jf. § 9 (2).

PERSONVERNPRINSIPPER

NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen:

- om lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet.

DE REGISTRERTES RETTIGHETER

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må prosjektansvarlig følge interne retningslinjer/rådføre dere med behandlingsansvarlig institusjon.

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilken type endringer det er nødvendig å melde: nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet i tråd med den behandlingen som er dokumentert.

Kontaktperson hos NSD: [redacted] (fjernet pågrunn av personvernshensyn), rådgiver.

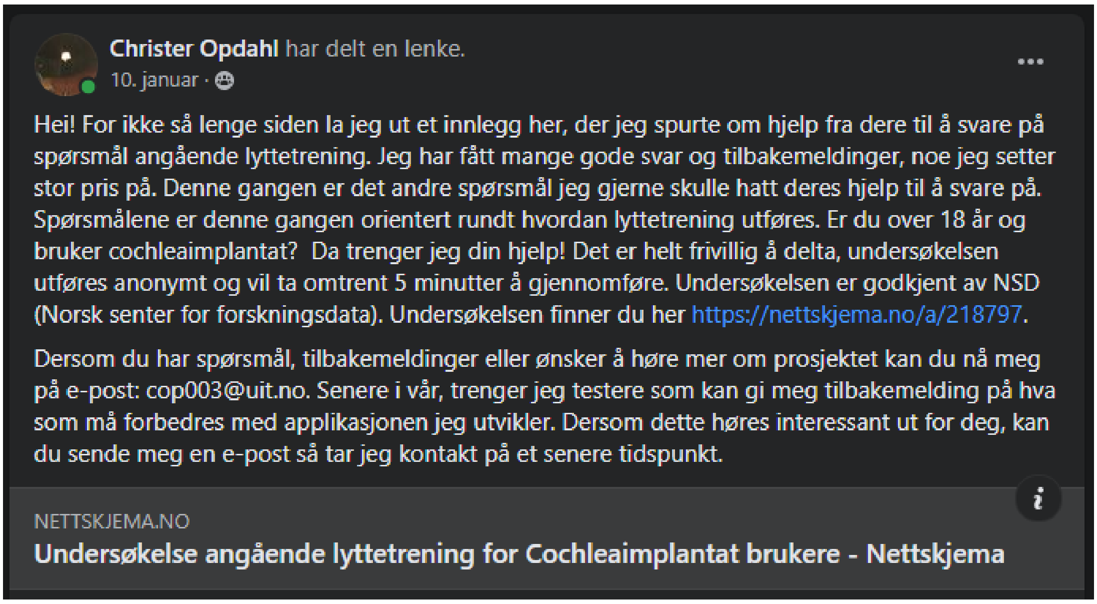
Lykke til med prosjektet!

Appendix IX.

First recruitment message – Primary questionnaire

Hei! For ikke så lenge siden la jeg ut et innlegg her, der jeg spurte om hjelp fra dere til å svare på spørsmål angående lyttetrening. Jeg har fått mange gode svar og tilbakemeldinger, noe jeg setter stor pris på. Denne gangen er det andre spørsmål jeg gjerne skulle hatt deres hjelp til å svare på. Spørsmålene er denne gangen orientert rundt hvordan lyttetrening utføres. Er du over 18 år og bruker cochleaimplantat? Da trenger jeg din hjelp! Det er helt frivillig å delta, undersøkelsen utføres anonymt og vil ta omtrent 5 minutter å gjennomføre. Undersøkelsen er godkjent av NSD (Norsk senter for forskningsdata). Undersøkelsen finner du her https://nettskjema.no/a/218797.

Dersom du har spørsmål, tilbakemeldinger eller ønsker å høre mer om prosjektet kan du nå meg på e-post: cop003@uit.no. Senere i vår, trenger jeg testere som kan gi meg tilbakemelding på hva som må forbedres med applikasjonen jeg utvikler. Dersom dette høres interessant ut for deg, kan du sende meg en e-post så tar jeg kontakt på et senere tidspunkt.



Christer Opdahl har delt en lenke.
10. januar ·

Hei! For ikke så lenge siden la jeg ut et innlegg her, der jeg spurte om hjelp fra dere til å svare på spørsmål angående lyttetrening. Jeg har fått mange gode svar og tilbakemeldinger, noe jeg setter stor pris på. Denne gangen er det andre spørsmål jeg gjerne skulle hatt deres hjelp til å svare på. Spørsmålene er denne gangen orientert rundt hvordan lyttetrening utføres. Er du over 18 år og bruker cochleaimplantat? Da trenger jeg din hjelp! Det er helt frivillig å delta, undersøkelsen utføres anonymt og vil ta omtrent 5 minutter å gjennomføre. Undersøkelsen er godkjent av NSD (Norsk senter for forskningsdata). Undersøkelsen finner du her https://nettskjema.no/a/218797.

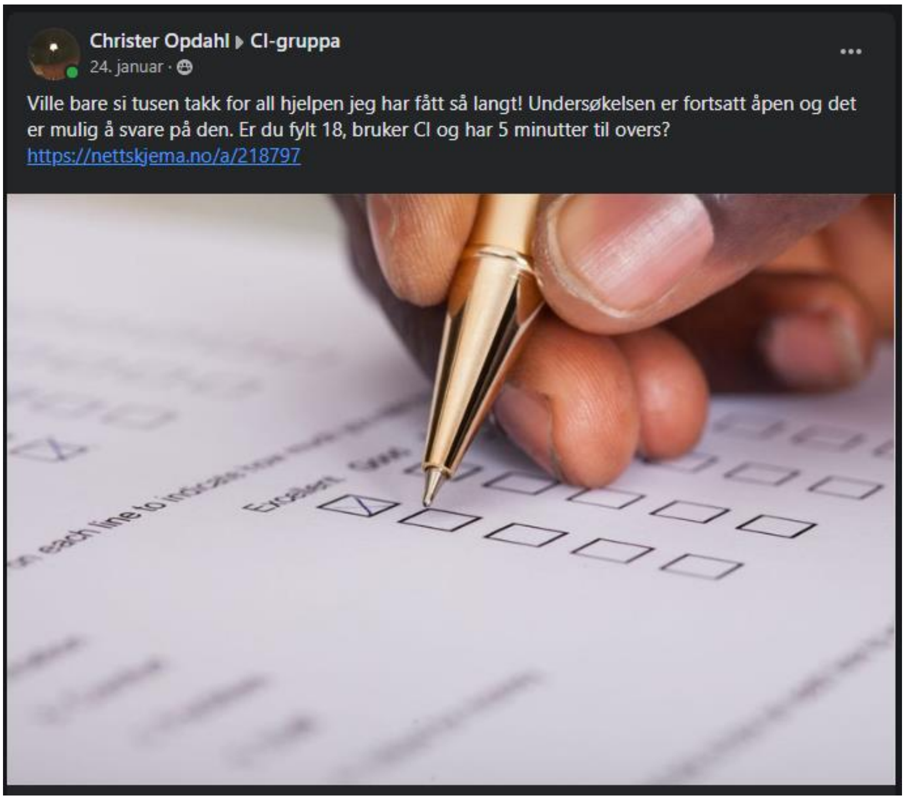
Dersom du har spørsmål, tilbakemeldinger eller ønsker å høre mer om prosjektet kan du nå meg på e-post: cop003@uit.no. Senere i vår, trenger jeg testere som kan gi meg tilbakemelding på hva som må forbedres med applikasjonen jeg utvikler. Dersom dette høres interessant ut for deg, kan du sende meg en e-post så tar jeg kontakt på et senere tidspunkt.

NETTSKJEMA.NO
Undersøkelse angående lyttetrening for Cochleaimplantat brukere - Nettskjema

Appendix X.

Second recruitment message – Primary questionnaire

Ville bare si tusen takk for all hjelpen jeg har fått så langt! Undersøkelsen er fortsatt åpen og det er mulig å svare på den. Er du fylt 18, bruker CI og har 5 minutter til overs? https://nettskjema.no/a/218797



Appendix XI.

Information – Semi-structured interview

Vil du delta i et Intervju i forbindelse med brukertesting av en nettløsning for lyttetrening?

Dette er et spørsmål til deg om å delta i et intervju hvor formålet er å hente informasjon rundt din brukeropplevelse av en nettløsning. Nettløsningen er utviklet av Christer Opdahl som en del av hans masteroppgave. Masteroppgaven går ut på å lage en løsning for mengdetrening på lyd for personer med Cochleaimplantat.

I dette skrivet gir vi deg informasjon om formålet med intervjuet og hva deltakelse vil innebære for deg.

Formål

Intervjuet er en del av en masteroppgave, som går ut på å utvikle en løsning for mengdetrening på lyd for personer med Cochleaimplantat. Løsningen kommer i form av en nettside tilgjengelig via nettleser på telefon og datamaskin. Den skal tilby realistiske lyder med fokus på trening av lydforståelse.

Formålet med intervjuet er å hente inn informasjon om din brukeropplevelse, slik at nettløsningen kan forbedres. Informasjonen du gir oss vil være med å påvirke utformingen av utseende og hvilke funksjonaliteter som er inkludert i nettløsningen.

Hvem er ansvarlig for forskningsprosjektet?

Prosjektet gjennomføres på UiT Norges arktiske universitet, institutt for Informatikk, og veiledes av Professor Eirik Årsand og universitetslektor André Henriksen.

Hvorfor får du spørsmål om å delta?

Du har fått spørsmål om å delta fordi du er medlem av Facebook gruppen "CI-Gruppa", som er den største Facebook gruppen for personer med Cochleaimplantat i Norge.

Hva innebærer det for deg å delta?

Dersom du velger å delta i prosjektet, innebærer dette at du prøver nettløsningen på egenhånd. Deretter vil du ta del i en telefonsamtale der det blir stilt spørsmål om din brukeropplevelse. Telefonsamtalen vil ta ca. 15 minutter og inkluderer 9 spørsmål. Dersom du tillater det, vil vi ta lydopptak av samtalen slik at vi senere kan transkribere opptaket. Dette gjøres for å spare tid under selve intervjuet. Lydopptaket vil bli slettet rett etter transkribering.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Dersom du velger å delta har du rett til å trekke samtykket tilbake når som helst uten å oppgi noen grunn. Alle personopplysninger og lydopptaket vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene du gir oss til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Intervjuet utføres over telefon. Lydopptak vil bare bli utført dersom du samtykker.
- Dersom du gir samtykke til å ta opp lyd, vil lydopptaket bare bli lagret via skylagring(www.nettskjema.no) beskyttet med flerfaktorautentisering.
- Bare mastergradsstudenten vil ha tilgang til lydopptaket.
- Lydopptaket vil bli slettet etter transkribering, senest 15. Mai. I forbindelse med transkriberingen vil all informasjon du har oppgitt bli anonymisert.

Du vil ikke kunne bli gjenkjent i masteroppgaven, der informasjonen blir sammenstilt og presentert. Informasjonen som blir innhentet vil bli satt inn som anonym informasjon i en større sammenheng i masteroppgaven

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?
Alle opplysninger slettes når prosjektet avsluttes, som etter planen er 15. Mai. Det vil da bare være de anonymiserte dataene som blir presentert i masteren.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

Dine rettigheter
Så lenge du kan **identifiseres** i datamaterialet, har du rett til:

- Innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene.
- Å få rettet opplysninger om deg som er feil eller misvisende.
- Å få slettet personopplysninger om deg.
- Å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

- Mastergradsstudent Christer Hagenes Opdahl: cop003@uit.no
- Hovedveileder og Professor Eirik Årsand: eirik.arsand@uit.no
- Biveileder og Universitetslektor André Henriksen: andre.henriksen@uit.no
- Personvernombud UiT: personvernombud@uit.no

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på e-post (personverntjenester@nsd.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen
Eirik Årsand
André Henriksen
Christer Hagenes Opdahl

Appendix XII.

Interview guide - Semi-structured interview

1. Nådde du web-applikasjonen på din foretrukne plattform(telefon/pc/nettbrett)?
2. Hva var ditt førsteinntrykk av web-applikasjonen?
3. Vanskelighetsgrad på de forskjellige tingene.
4. Er det noe du føler mangler i web-applikasjoner?
5. Hva synes du var bra eller dårlig?
6. Har du noen forslag til endringer som bør gjøres?
7. Har du sett noen lignende systemer?
8. Tror du at du ville brukt en slik applikasjon?
    a. Hva må eventuelt til for at du ville brukt den?
    b. Hvordan ville du ha brukt den?
        i. Ville du akseptert push-varsler
            1. Hvor ofte? en gang i uken, en gang for dagen?
9. Ville du anbefalt denne applikasjonen for andre?
    a. Hva måtte til for at du ville anbefalt denne applikasjonen for andre?

Appendix XIII.

Correspondence to DPO - Semi-structured interview

Hei Christer,
Gjennomføring av intervju med lydopptak innebærer behandling av personopplysninger. Det vil si at personvernregelverket og universitetets egne retningslinjer for behandling av personopplysninger i forsknings- og studentprosjekter kommer til anvendelse. Jeg tenker at dette utgjør en ny behandling i ditt mastergradsprosjekt og du må, i samråd med dine veiledere, melde behandlingen til NSD for ny vurdering. Det kan vurderes om dette kan meldes som en endring til spørreundersøkelsen du meldte først til NSD, men jeg tenker at det er mest ryddig å skille mellom disse to behandlingene. Når det nye delprosjektet meldes til NSD kan bør vise til den første vurderingen.

UiT har som et utgangspunkt fastsatt at det ikke tillates at tilsatte og studenter behandler personopplysninger med privat utstyr, se retningslinjer for behandling av personopplysninger i forsknings- og studentprosjekter punkt 11 fjerde ledd, Retningslinjer+for+behandling+av+personopplysningar+i+forskings-+og+studentprosjekt+ved+UiT+(oppdatert+300921).pdf

Men universitetet har fastsatt noen unntak fra dette forbudet, se Rutine+for+bruk+av+privat+utstyr+ved+behandling+av+personopplysninger+i+forsknings-+og+studentprosjekter+-+v1+-+vedtatt+10-11-21.pdf (uit.no)

Punkt 7 D er relevant i ditt tilfelle. Dersom du kan følge disse rutinene er det bare henvise til dette i meldeskjemaet til NSD. Alt annet bruk av privat utstyr enn det som tillates av universitetet i disse rutinene må godkjennes av Avdeling for it ved Faggruppe for informasjonssikkerhet og personvern. Ut over dette synes din fremgangsmåte å være i tråd med det som er vanlig praksis ved universitetet.

Med vennlig hilsen

[redacted](fjernet pågrunn av personvernshensyn)
[redacted](fjernet pågrunn av personvernshensyn)

UiT Norges arktiske universitet

[redacted](fjernet pågrunn av personvernshensyn)

Appendix XIV.

Approval from NSD - Semi-structured interview

OM VURDERINGEN

Personverntjenester har en avtale med institusjonen du forsker eller studerer ved. Denne avtalen innebærer at vi skal gi deg råd slik at behandlingen av personopplysninger i prosjektet ditt er lovlig etter personvernregelverket.

Personverntjenester har nå vurdert den planlagte behandlingen av personopplysninger. Vår vurdering er at behandlingen er lovlig, hvis den gjennomføres slik den er beskrevet i meldeskjemaet med dialog og vedlegg.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige personopplysninger og særlige kategorier av personopplysninger om helseforhold frem til 15.05.2022.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 nr. 11 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse, som kan dokumenteres, og som den registrerte kan trekke tilbake.

For alminnelige personopplysninger vil lovlig grunnlag for behandlingen være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 a.

For særlige kategorier av personopplysninger vil lovlig grunnlag for behandlingen være den registrertes uttrykkelige samtykke, jf. personvernforordningen art. 9 nr. 2 bokstav a, jf. personopplysningsloven § 10, jf. § 9 (2).

PERSONVERNPRINSIPPER

Personverntjenester vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen:

• om lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen

• formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål

• dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet

• lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet.

DE REGISTRERTES RETTIGHETER

Vi vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

Personverntjenester legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

Nettskjema er databehandler i prosjektet. Personverntjenester legger til grunn at behandlingen oppfyller kravene til bruk av databehandler, jf. art 28 og 29.

For å forsikre dere om at kravene oppfylles, må prosjektansvarlig følge interne retningslinjer/rådføre dere med behandlingsansvarlig institusjon.

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til Personverntjenester ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilken type endringer det er nødvendig å melde: https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema. Du må vente på svar fra oss før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

Vi vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Kontaktperson hos oss: [redacted] (fjernet pågrunn av personvernshensyn)

Lykke til med prosjektet!

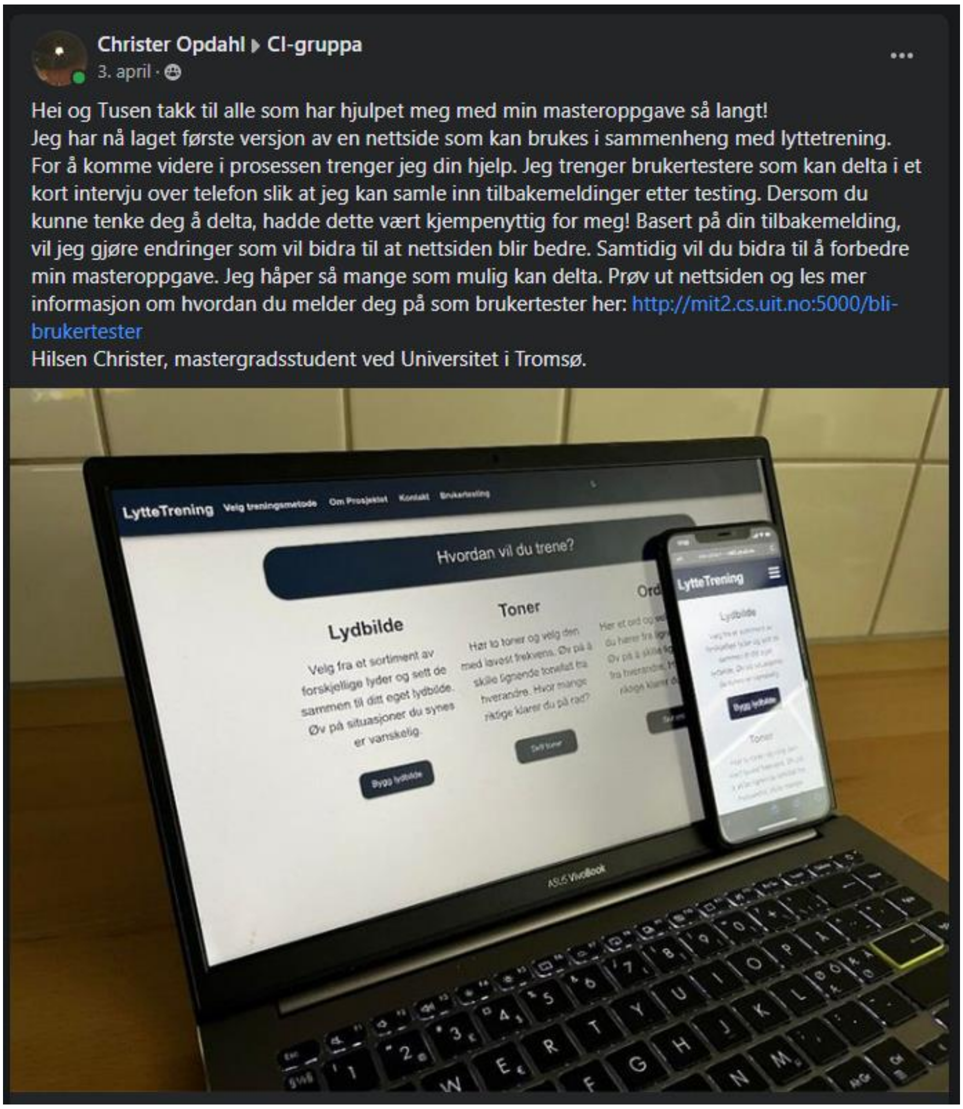
Appendix XV.

First recruitment message – Semi-structured interview

Hei og Tusen takk til alle som har hjulpet meg med min masteroppgave så langt!

Jeg har nå laget første versjon av en nettside som kan brukes i sammenheng med lyttetrening. For å komme videre i prosessen trenger jeg din hjelp. Jeg trenger brukertestere som kan delta i et kort intervju over telefon slik at jeg kan samle inn tilbakemeldinger etter testing. Dersom du kunne tenke deg å delta, hadde dette vært kjempenyttig for meg! Basert på din tilbakemelding, vil jeg gjøre endringer som vil bidra til at nettsiden blir bedre. Samtidig vil du bidra til å forbedre min masteroppgave. Jeg håper så mange som mulig kan delta. Prøv ut nettsiden og les mer informasjon om hvordan du melder deg på som brukertester her: http://mit2.cs.uit.no:5000/bli-brukertester

Hilsen Christer, mastergradsstudent ved Universitet i Tromsø.



Appendix XVI.

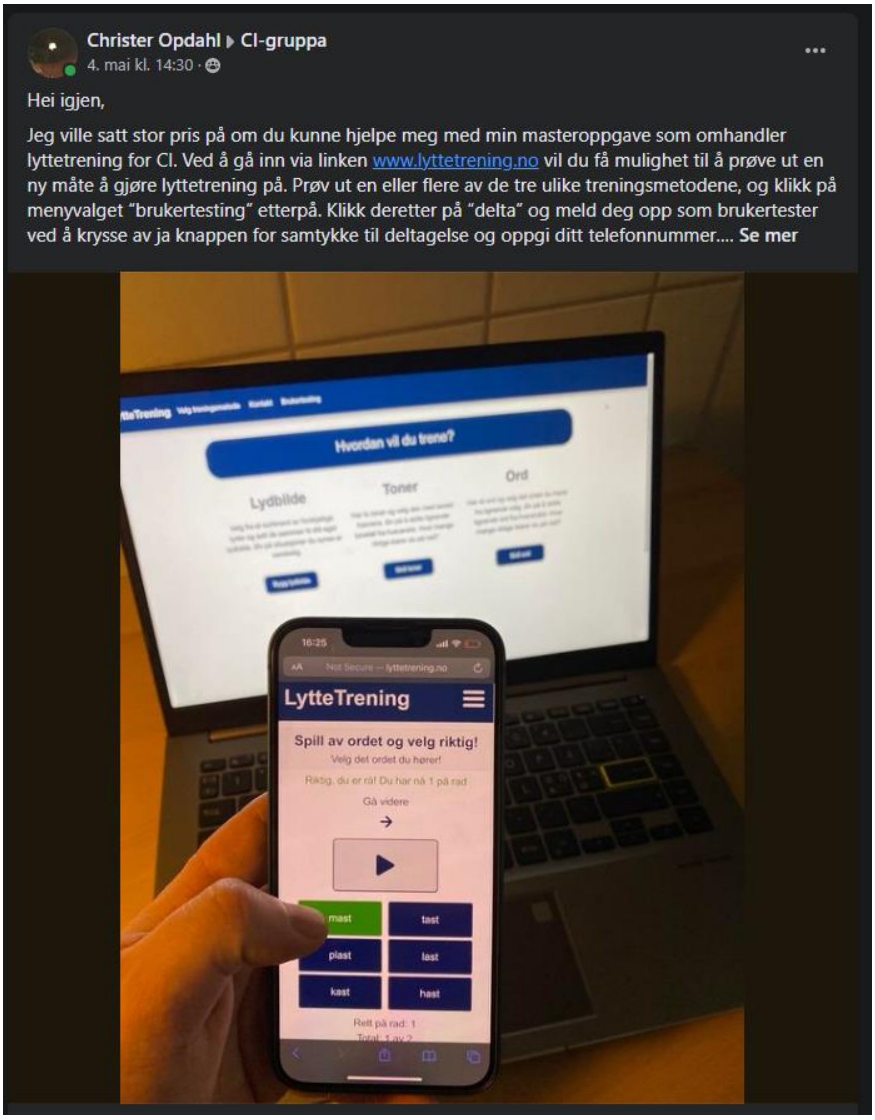
Second recruitment message – Semi-structured interview

Hei igjen,

Jeg ville satt stor pris på om du kunne hjelpe meg med min masteroppgave som omhandler lyttetrening for CI. Ved å gå inn via linken www.lyttetrening.no vil du få mulighet til å prøve ut en ny måte å gjøre lyttetrening på. Prøv ut en eller flere av de tre ulike treningsmetodene, og klikk på menyvalget "brukertesting" etterpå. Klikk deretter på "delta", meld deg opp som brukertester ved å krysse av ja knappen for samtykke til deltagelse og oppgi ditt telefonnummer. Deretter vil jeg ta kontakt slik at vi kan avtale tidspunkt for et kort intervju.

(På grunn av personvernhensyn er det dessverre bare mulig å delta via telefon, jeg forstår at dette ikke passer for alle)

På forhånd tusen takk!



Appendix XVII.

Response from an audio therapist

Hei Christer,

Verktøyet for lyttetrening som du har laget virker veldig interessant. Dette tenker jeg har stort potensiale. Jeg jobber ikke primært med ci brukere men dette ville ha vært et godt supplement som tilbud hvis det ble nødvendig. Jeg har noen ci brukere innimellom som også har tinnitus.

Med vennlig hilsen

[redacted] (fjernet pga personvernshensyn)

[redacted] (fjernet pga personvernshensyn)

Appendix XVIII.

Creative commons licensed youtube videos which were used to create the environmental sounds within the sound collection

kråke, frosk, hane, ugle, løve, hjort
https://www.youtube.com/watch?v=iCjWdoauUY4

måker
https://www.youtube.com/watch?v=Y4EZo7oTHSk

sauer
https://www.youtube.com/watch?v=dnHTSjLn8EU

hester
https://www.youtube.com/watch?v=k4VTSzxEqxY

klirrende flasker
https://www.youtube.com/watch?v=ADU6ZOsuy5w

byggeplass
https://www.youtube.com/watch?v=zSIwDQ1oCwY
https://www.youtube.com/watch?v=KET5ekyvpFU
https://www.youtube.com/watch?v=Nob1bpSXyrg

gravemaskin
https://www.youtube.com/watch?v=jmXFYond9jo

bulldoser
https://www.youtube.com/watch?v=m-sJz1wZbEE

jackhammer
https://www.youtube.com/watch?v=A0DXCptudXY

i skogen
https://www.youtube.com/watch?v=4FpA1-kgHDQ

ved havet
https://www.youtube.com/watch?v=wERteXig6Ok

restaurant bakgrunslyd
https://www.youtube.com/watch?v=IF5NJ_B6DX4

kaffi restaurant
https://www.youtube.com/watch?v=0pQyg8OIjp4

restaurant bagrunslyd

https://www.youtube.com/watch?v=u9WnP5GqtHw
fly
https://www.youtube.com/watch?v=-v-ZSrNhYEg
https://www.youtube.com/watch?v=gB-OA2pXOrk

sirener
https://www.youtube.com/watch?v=dnM_jtUw_TQ

https://www.youtube.com/watch?v=kNlUflz9-Ng

torden uten regn
https://www.youtube.com/watch?v=AN454nr6f_k

regn
https://www.youtube.com/watch?v=BY_lSm2gSzY

hunder, mange forskjellige
https://www.youtube.com/watch?v=MBSKB66mH_0

banking på dør
https://www.youtube.com/watch?v=fXLE4rxgO6M

knirking i dør
https://www.youtube.com/watch?v=ihugFC6vls0

papir
https://www.youtube.com/watch?v=vyFVHwfGwWw

papir knust
https://www.youtube.com/watch?v=p0M8CTQNvwg

rennende vann
https://www.youtube.com/watch?v=eRTTe56ZtyI

serverrom
https://www.youtube.com/watch?v=NeESf9aCZHQ

lighter
https://www.youtube.com/watch?v=EQE8Di2ysws

påhengsmotor
https://www.youtube.com/watch?v=ZKF381ydx4s

17.mai tog
https://www.youtube.com/watch?v=7G6yZZxHsx8

fugle kvittring

https://www.youtube.com/watch?v=ACKys9VPikg