

P6

Hijab Nasir and Lojain Ajek

May 25, 2023



AALBORG UNIVERSITY
DENMARK

Title:
Bachelor Project

Theme:
Cyber Physical System

Project period:
01. Feb. to 26. May.

Project group:
SW6 Group 3

Abstract:

This project focuses on the development of a communication application designed specifically for elderly individuals who struggle with reading and writing. The application aims to simplify digital communication by providing a user-friendly interface with intuitive features and technologies such as Text-to-speech, ChatGPT. The application aims to improve social connectivity and quality of life for elderly users, empowering them to engage and connect with their loved ones more effectively.

Group member:
Hijab Nasir
Lojain Ajek

Supervisor:
Johannes Bjerva

Number of pages: 57

Due date:
May 26, 2023

Preface

Credits:

We would like to thank our supervisor Johannes Bjerva for helping to clarify the project's overall direction and for their assistance in this regard. We appreciate our supervisor's ability to provide guidance.

Another mention also goes out to our semester coordinator Rikke Hagensby Jensen, for her invaluable contribution during the Statusseminar and excellent feedback and also also introduced us to project proposals and guided us in the right direction.

Furthermore, we would like to thank Esther Ploeger for her invaluable feedback throughout the project. She willingly participated in every meeting, for us to learn how to improve in our process.

Resources:

The GitLab repository: <https://gitlab.com/Lajek20/bachelorproject.git>

Contents

1	Introduction	2
2	Problem Analysis	3
2.1	Problem Identification	3
2.1.1	Digital Communication	3
2.1.2	Why digital communication important for elderly people	4
2.1.3	The difficulties with digital communication	4
2.1.4	Understanding the need of target	4
2.1.5	Existing solutions	5
2.1.6	Summary	6
2.2	Interview Methodology	7
2.2.1	Data	8
2.2.2	Summary	14
3	Problem Statement	16
3.1	Problem Statement	16
4	Methodology & Design	17
4.1	Natural language processing	17
4.1.1	Prompt Engineering	17
4.1.2	ChatGPT API	18
4.1.3	Text-to-speech	19
4.1.4	Human-centered	19
4.1.5	Cyber physical system	19
4.2	Design	20
4.2.1	Rich Picture	20
4.2.2	Requirements	21
4.2.3	Low-fi and Hi-fi	22
5	System Architecture	27
5.1	System Architecture	27
6	Implementation	28
6.1	Implementation	28
6.2	Front-end	28
6.2.1	Home screen	28
6.2.2	Chat screen	29
6.2.3	Profile screen	32
6.3	Back-end	33
6.3.1	Authentication	33
6.3.2	Data models	35
6.3.3	GraphQL APIs	35
6.3.4	How is back-end connected with front-end	36
6.3.5	ChatGPT API	36

7 Testing	44
7.1 Usability testing	44
7.2 Software Quality Management	48
7.2.1 CodeScene	48
7.3 Snapshot testing	50
7.3.1 Integration testing	52
7.3.2 End-to-end testing	53
8 Discussion	54
8.1 Discussion	54
8.2 Future development	55
9 Conclusion	56
9.1 Conclusion	56

Summary

This report covers the process of a 6th-semester Software Engineering

The primary goal of this application is to make digital communication simpler for elderly users so they can interact and connect with their loved ones more efficiently. The application intends to increase social connectivity and improve the quality of life for elderly individuals by incorporating user-friendly interfaces and cutting-edge technology like Text-to-speech and the ChatGPT API.

To ensure the application's effectiveness and usability, a survey was conducted with elderly individuals residing in a nursing home. The purpose of this survey was to learn more about their particular communication needs and preferences. The group was able to better understand the demands of the target group by interacting with them directly. With this information, they were able to modify the app's functionality and interface. The comments from the interview influenced the design and functionality choices, resulting in a user-centric and intuitive end product.

This project focuses on developing communication that is specially made for elderly individuals who have trouble reading and writing. To make digital communication for older users simpler and promote social connectivity by utilizing the capabilities of new technologies like ChatGPT and adding features like Text-to-speech. The group gathered insightful feedback from actual consumers during interviews, making the final product more user-friendly and tailored to the specific requirements of the target market. This application tries to improve the quality of life and general well-being of elderly people by enabling them to interact and connect with their loved ones more effectively.

The application will also make use of the ChatGPT API, a cutting-edge tool that utilizes algorithms for natural language processing. This functionality will allow elderly individuals to participate in text-based conversations in real-time without having to have advanced typing or writing skills. Digital communication will become more engaging and user-friendly thanks to the ChatGPT API, which will enable interactive and dynamic dialogues.

The ultimate purpose of this initiative is to increase the elderly's quality of life and social connectivity. The application enables older people to connect and communicate with their loved ones more successfully by streamlining digital communication and offering straightforward features. In turn, this promotes a sense of community, lessens loneliness, and enhances general well-being.

Elderly individuals in nursing homes have been asked for their feedback and suggestions at different points of the development process. The functionality and interface of the application are customized to the target group's unique demands and preferences thanks to this close interaction. By considering their suggestions, the application can deliver a truly user-centric experience.

The goal of this project is to create a communication application that is specifically made for elderly individuals who have difficulty reading and writing. The application offers to increase social connectivity, simplify digital communication, and improve the quality of life for older users by integrating user-friendly interfaces and cutting-edge technologies like Text-to-speech and the ChatGPT API. The features and interface of the app have been improved by direct interaction with its target group.

1 | Introduction

In an increasingly interconnected world, digital communication has become a vital means of staying connected with loved ones, accessing information, and participating in various aspects of daily life. However, for elderly individuals, this ubiquitous mode of communication often presents unique challenges. Difficulties related to reading, writing, navigating interfaces, and authentication can significantly impede their ability to fully participate in and benefit from the digital real.

Recognizing the importance of inclusively and the need to address these challenges, our project aims to develop an innovative solution that empowers elderly individuals to overcome these barriers and engage in digital communication more effectively. By leveraging modern technologies, specifically Text-to-Speech and the ChatGPT API, we seek to create a messaging application that simplifies the communication process, making it more accessible and intuitive for older users.

Digital communication has transformed the way we interact and communicate with one another. For the elderly population, it presents a means of connecting with family and friends, accessing online services, and staying informed about current events. Additionally, it plays a crucial role in reducing social isolation, which is a significant concern among the elderly. However, traditional methods of digital communication, such as text-based messaging and complex user interfaces, often pose significant challenges for older individuals, leading to frustration, alienation, and a sense of exclusion.

Elderly individuals face a range of difficulties that hinder their ability to engage effectively in digital communication. One of the key challenges is related to reading and writing. Age-related visual impairments, cognitive decline, and limited literacy skills can make it challenging to comprehend text-based messages and respond with written messages. Additionally, navigating through complex interfaces and dealing with authentication procedures can be overwhelming and confusing, deterring older users from engaging in digital communication.

To develop an effective solution, it is essential to gain a deep understanding of the needs, preferences, and pain points experienced by elderly individuals in digital communication. Conducting interviews and engaging in direct conversations with the target demographic will provide invaluable insights and serve as a foundation for the design and development process. By listening to their firsthand experiences and challenges, we can tailor our solution to address their specific needs effectively.

While there are existing solutions and assistive technologies aimed at supporting elderly individuals in digital communication, many fall short of fully addressing the breadth of challenges faced by this demographic. Some applications may provide text enlargement or voice typing features, but they often lack the comprehensive integration of modern technologies that can significantly enhance communication accessibility and ease of use.

The primary objective of this project is to create a messaging application that leverages Text-to-Speech and the ChatGPT API to improve communication for elderly individuals. The methodology involves conducting a thorough problem analysis to identify the specific barriers faced by older users, understanding the importance of digital communication for this demographic, exploring existing solutions, and conducting interviews to gain valuable insights. Based on the findings, a robust implementation plan will be devised, incorporating the chosen technologies, and designing an intuitive user interface that caters to the unique needs of elderly individuals.

This project aims to bridge the communication gap and empower elderly individuals to participate fully in the digital world. The following sections of this project will delve into the problem analysis, methodology, implementation, testing, and future development, highlighting our progress in creating an application that addresses the difficulties faced by elderly individuals in reading, writing, and navigating interfaces. Through the integration of Text-to-Speech and the ChatGPT API, we strive to create an inclusive and accessible digital communication platform that enhances the quality of life.

2 | Problem Analysis

This chapter will begin by identifying the main issues within the problem domain of elderly individuals and communication problems. Section 2.1 will provide an overview of the global study on the value of communication for seniors Section 2.1.1, the unique challenges they encounter with modern technology and cellphones 2.1.3, and potential solutions to these problems section 2.1.5. Section 2.2 will present an interview study carried out in Denmark to acquire further insight into the communication habits of senior people who use smartphones. This study will look at the devices and applications that this group uses and give a thorough understanding of the communication difficulties that people in their age range face can be seen in section 2.2.1.

2.1 Problem Identification

This section explores the problem domain intending to address the communication challenges faced by elderly individuals. Aging is often associated with changes that can impact an individual's ability to communicate effectively, such as hearing loss, vision impairment, memory loss, and language processing difficulties. The section will examine these communication barriers and explore potential strategies and solutions for overcoming them.

2.1.1 Digital Communication

The ability to communicate with others and share thoughts, emotions, and experiences is a key component of human connection. However, as people age, they could go through several physical, cognitive, and social changes that could affect how well they communicate. Changes that can impact a person's capacity for communication are frequently linked to aging[1]. Therefore, communication is very important for elderly individuals since communication plays a crucial role in the mental and emotional well-being of elderly individuals. Social isolation and loneliness are common among older adults, and communication can help prevent these issues by providing a means for individuals to connect with others and maintain social relationships. Additionally, communication can help elderly individuals maintain their cognitive abilities, as engaging in conversation and mentally stimulating activities can help keep the mind sharp and active[1].

For elderly individuals, communication is essential as it helps them retain their social relationships and fight loneliness. A growing number of older adults are experiencing loneliness and social isolation, which is a serious public health concern. These conditions can have a variety of detrimental health effects, such as an increased risk of depression, anxiety, cognitive decline, and chronic diseases like heart disease and stroke. Also, older people who are lonely may experience more hospitalizations and falls [2].

Loneliness and social isolation pose serious health risks to elderly people, raising their risk of dementia and other serious medical conditions. According to a survey by the National Academies of Sciences, Engineering, and Medicine (NASEM), almost one-fourth of adults over the age of 65 report feeling socially isolated, and about one-third of people over the age of 45 report experiencing loneliness. Therefore, elderly individuals tend to be more vulnerable to the effects of a lack of communication, experiencing greater consequences in various aspects of their lives [2].

It can be challenging to evaluate social isolation and loneliness, but there is ample evidence that many persons 50 and older struggle with these problems, which can have a serious negative impact on their health. Comparable to smoking, obesity, and inactivity, the latest lookup has proven that social isolation can appreciably increase the chance of early demise from all causes. Poor social interactions characterized by social isolation or loneliness have been linked to a 29 percent higher hazard of coronary heart disorder and a 32 percent accelerated threat of stroke. It has additionally been linked to a 50 percent extended chance of dementia. Moreover, loneliness has been linked to a greater incidence of suicide, anxiety, and despair. Loneliness has been related to an almost four-fold increased hazard of death, a sixty-eight percent multiplied chance of hospitalization, and a fifty-seven percent multiplied threat of visits to the emergency

room in sufferers with coronary heart failure [2].

2.1.2 Why digital communication important for elderly people

As mentioned before, digital communication is important for elderly individuals because it may bridge the gap in distance between elderly people and their loved ones. Seniors' emotional and mental health depend on digital contact, especially those who may be lonely or live alone. It addresses several strategies for using technology, such as sending messages and voice messages and social media, to help seniors connect with their loved ones even when they are separated by distance.

Moreover, even if they are physically unable to participate in events or activities, internet communication can help seniors feel more involved and connected in their communities. Seniors can communicate with people who have similar interests and hobbies through social media, and they can utilize these resources to remain current on local events and activities.

Despite the numerous advantages of digital communication for seniors, it's critical to understand that not all of them have access to the tools they need or the skills to use them effectively. To help seniors learn how to use digital communication tools and maintain connections with their loved ones and communities, various organizations and community organizations offer technology training programs and services [3].

2.1.3 The difficulties with digital communication

For many elderly individuals, staying in touch with family and friends is a crucial aspect of daily life. Nonetheless, many seniors may find it challenging to keep up with the technology's quick growth given the increased use of digital communication tools like SMS messaging. Sending SMS texts to their grandchildren, for instance, may be a source of frustration and challenges for elderly individuals who may have issues using telephones or have visual or cognitive limitations that make texting more challenging [4] [5]. Some of the common communication problems that elderly individual experiences are hearing loss, vision problems, cognitive decline, and language barriers. As per age, people may experience changes in their hearing and vision abilities that can make it difficult to hear or see screens. Additionally, cognitive decline can impact the ability to process information, recall details and communicate effectively [5] [1].

Elderly individuals face difficulties using phones due to a range of factors. The small size of phone screens is a typical issue that can make it challenging for the elderly to read an SMS, view images, or navigate menus. Another difficulty that they are facing is small keyboards, which makes it difficult for them to write messages. Another common problem that elderly individuals are facing when using phones is hearing and vision loss. This can make it difficult for them to hear phone conversations and challenging for them to see small text on the phone screen [5].

2.1.4 Understanding the need of target

Importance of technology in improving communication between elderly individuals and their providers, technology can have a major impact on helping elderly individuals with communication problems. Technology such as messaging apps, and voice assistants can make it easier for elderly individuals to communicate with their friends and family members. Those technologies can help to overcome barriers such as hearing loss or difficulty with reading and writing messages [6].

2.1.5 Exiting solutions

Exiting applications require login to use the application, and many old people have difficulty remembering the login passwords and emails. They also have difficulty using such applications, since the interface is not very easy-to-use.

Lists of some of exiting applications:

- Facebook
- Whatsapp
- iMessage:
- NaturalReader
- Voice Dream Reader

Several programs, like Facebook and NaturalReader, need users to log in before they can use them. Yet, this might be problematic for older folks since they can have trouble recalling their login information, including passwords and email addresses. However, some of these applications may have complicated user interfaces that make it difficult for older folks to utilize them successfully and navigate.

Moreover, certain applications may have a challenging user interface despite not being primarily created for communication. Because of this, using these programs and finding the information they require may be difficult for older folks. These programs might also be restricted in terms of what they can do; for example, some of them might only let users read or write content, not both.

2.1.5.1 Facebook

Facebook is a social media platform, that allows users to connect and share information, as such any good news that wants to share with family and friend, sending and receiving messages on Facebook, where Messenger is another software that is a connection to Facebook, so user can log in in Messenger with Facebook ID. Facebook messaging software does speech-to-text technology.

2.1.5.2 Whatsapp

Whatsapp is a messaging application, which does not requires login with email, but has a very smart login system with phone numbers, Whatsapp allows user to send voice messages, and text messages, and share images, videos, documents, and GIFs with other users. Whatsapp does not have very new technology such as Speech-to-text and Text-to-speech technology.

2.1.5.3 iMessage

iMessage is a messaging service developed by Apple for IOS deceives, such as iPhones, iPad, and Macbooks. iMessage works also with Siri, for IOS deceives. iMessage also supports video and audio calls and iMessage only has Speech-to-text technology.

2.1.5.4 NaturalReader

NaturalReader is a software that works to convert text to spoken language so it does text-to-speech technology. The software also has a mobile application, an online application, and a Chrome extension, but the interface is not very easy-to-use and requires login.

2.1.5.5 Voice Dream Reader

Voice Dream Reader is a text-to-speech application, users can listen to written content using text-to-speech technology. Voice Dream Reader can also read the longer file as such PDF, Microsoft Word, EPUB, and any other web pages.

2.1.6 Summary

Due to a variety of factors, including language barriers, cognitive decline, hearing loss, and vision issues, many older people have trouble communicating. Feelings of isolation and loneliness may result from this. Also, some people could have trouble utilizing new technologies, which might further impair their capacity for effective communication. To strengthen one's social relationships and general well-being, it's critical to address these issues and look for support.

2.2 Interview Methodology

For this project, a survey methodology was chosen as the primary data collection method. The survey was designed to collect quantitative data from participants who met the inclusion criteria. The survey questions were carefully developed and pre-tested to ensure their clarity and relevance to the research objectives.

The survey was administered online using Google Surveys. The inclusion criteria for this research were individuals between the age of 60 and 80 who are considered to be elderly. The purpose sample was chosen to ensure that the participants had relevant experience related to the research question.

There were both closed-ended and open-ended questions in the survey. Whereas open-ended questions allowed for more in-depth and nuanced responses, closed-ended questions contained predetermined response alternatives.

To ensure the reliability and validity of the survey results, various measures were taken. Firstly, the group went to many of the nurse homes to be able to ask the question directly to the participants and write down the answers in the survey to save time and effort. So we interviewed some of the elderly but the quantity of the answers was insufficient to make data from it. The reason behind this was there are a few elderly who use the telephone in the nursing houses. Therefore, the survey was shared in a Facebook group, where the group aims to improve the welfare of the elderly. The group names "Alles kamp for bedre velfærd for de ældre". The goal of sharing the survey in this group was to ensure the largest reach of the required segment. Thirdly, statistical methods were used to analyze the survey data and identify any significant findings.

Overall, the survey methodology was a successful means to gather quantitative data on a variety of subjects connected to the study issue from a large number of participants. The survey's findings offer insightful information that aids in finishing this project. It should be emphasized that there are several drawbacks to the survey approach, such as the chance for response bias and the inability to collect more in-depth, qualitative data.

While developing an application for elderly people, interviews can be a useful method for gathering information so that we can better understand their needs. Identifying problems with usability You can have a more thorough understanding of the difficulties and experiences of elderly people by speaking with them. By doing so, you can develop empathy for your users and develop a more user-friendly and pleasant application.

2.2.1 Data

In this section, data will be explored which gathered from the interview with the elderly people. The purpose of the interview was to gain insights into the challenges the elderly face while using communication applications, as well as to gather their perspectives on various topics related to the field. The data collected from this interview provides valuable insights into the current state of both communication applications and the challenges.

Age



Figure 2.1: How old are you?

Age groups from the survey are shown in a diagram 2.1. The age range in the diagram runs from 55 to 82. Yet, we find that surveys that begin from age 60, where 10.3% of the population is, are the most beneficial.

The diagram shows that the highest goal ages are 60, 66, 68, and 72. The diagram is crucial because it reveals which age group has the most trouble interacting with friends and relatives.

How often elderly communicate

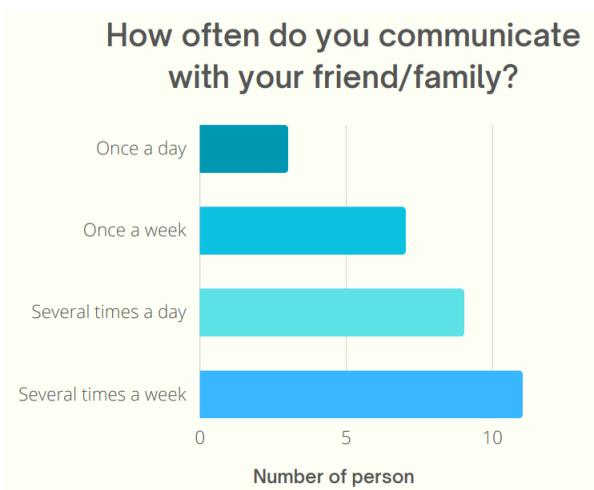


Figure 2.2: How often do you communicate with your family/friends?

The diagram 2.2 shows how often people communicate with their family and friends. The diagram is a bar chart, that displays four answer options for the question "How often do you communicate with your family/friends".

The answer options are:

- Once a day
- Several times a day
- Once a week
- Several times a week

From the diagram, we can see that the majority of respondents communicate with their family and friends "Several times a week" with 37.9% and 24.1% respondents communicating "Several times a day". The diagram provides insight into the communication habits of people, by understanding how often people communicate.

Difficulties with communication

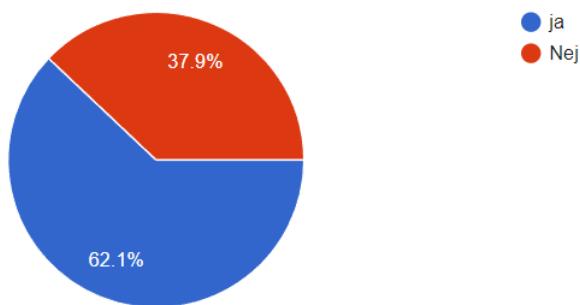


Figure 2.3: Do you have difficulties communicating with your family/friends

The percentage of people who have difficulty communicating with their relatives and friends is shown in the diagram 2.3.

The diagram is a pie chart with the questions "Do you have Difficulty communicating with your family/friends?" and two answer choices, yes or no. Each response has a distinct color option; the blue one indicates a "yes" response, while the red one indicates a "no" response. According to the diagram, 62.1% of respondents said they had difficulty connecting with their family and friends, while 37.9% said they have no difficulty communicating

Reasons for difficulties with communication

As part of our survey, a question was asked respondents to our study on communication with family and friends to provide a reason for their communication difficulties. Respondents were allowed to give their responses to the question. The question is "Why do you have difficulty communicating with your family/friends".

According to the survey results, the majority of respondents find it difficult to communicate with their family and friends for a range of reasons, such as forgetting passwords, using a small keyboard, having a difficult-to-use interface, and having trouble writing and reading.

Difficulty with reading

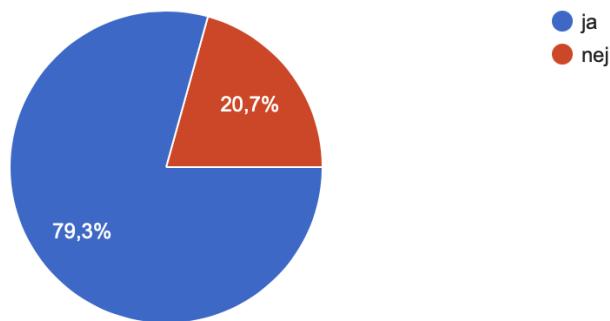


Figure 2.4: Do you have problems with reading messages

The percentage of people who have problems with reading messages is shown in the diagram 2.4

The diagram is a pie chart with the question "Do you have problems with reading messages?" and two answer choices, yes or no to be able to show how many people have difficulty with reading messages. According to the diagram, 79,3% of respondents said they have problems with reading messages, while 20,6% said they have no problems with reading messages.

Difficulty with writing

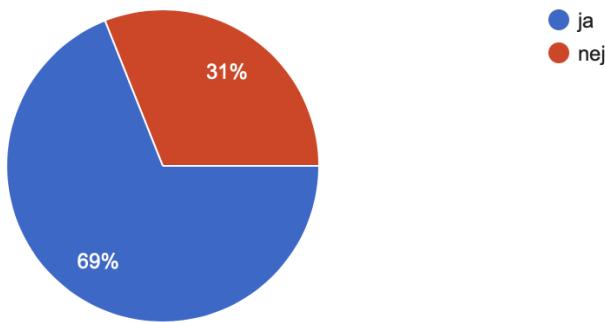


Figure 2.5: Do you have problems with writing messages

As in the previous section, the percentage of people who have problems with writing messages is shown in the diagram 2.5

The diagram is a pie chart with the question "Do you have problems with writing messages?" and two answer choices, yes or no to be able to show how many people have difficulty with writing messages. According to the diagram 69% of respondents said they have problems with writing messages, while 31% said they have no problems writing messages.

Languages

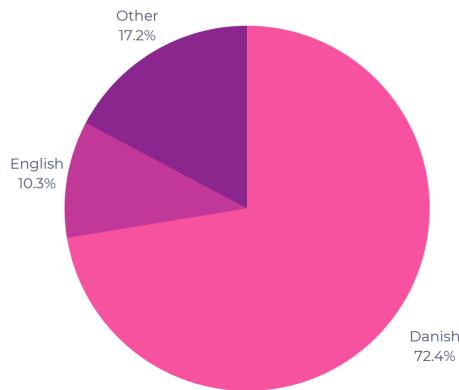


Figure 2.6: which language do you use to communicate with your family/friends?

From the diagram 2.6, we can see that the majority of respondents are using Danish with 72.4% to communicate with their friends and family. 10.3% of respondents said that they use English to communicate, which was the second most used language. The rest participants answered that they use another language.

2.2.1.1 Applications

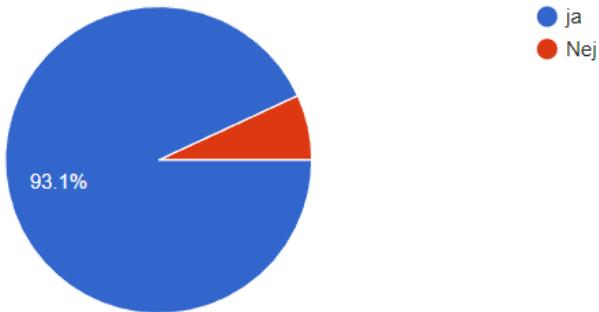


Figure 2.7: Do you use Facebook, WhatsApp or Message

as diagram 2.7 shows that major of respondents uses the applications such as Facebook, Message, and WhatsApp.

More specifically about which application they use the most. See diagram below 2.8. This shows that majors are using WhatsApp with 33.3% and Facebook with 43.3%. Some respondents use Message with 10% and the remaining are using others.

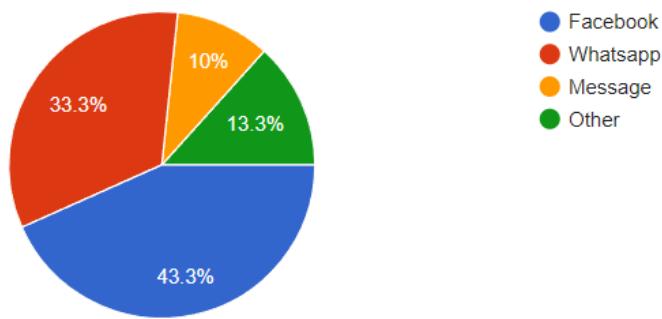


Figure 2.8: which app do you use to communicate with your family/friends?

Challenges of using the mentioned application

To find out what are the challenges that people face when using communication applications such as WhatsApp and Facebook, an open question was asked, "What are the problems that you face when using Facebook and WhatsApp?" This question aims to collect as much data as possible to use as an aid at the design stage and not to fall into the same current problems that older people suffer from when using these applications.

According to the survey results, the majority of respondents find it difficult to use communication applications such as WhatsApp and Facebook for a range of reasons, such as difficulty remembering the password, some interface challenges, some of the participants have difficulty writing messages, and also their eyesight is weak.

In addition to this question, another question was asked specifically about the interface to indicate whether the elderly find difficulty in the interface.

So, the percentage of people who have difficulty using an interface of the mentioned application is shown in the diagram 2.9

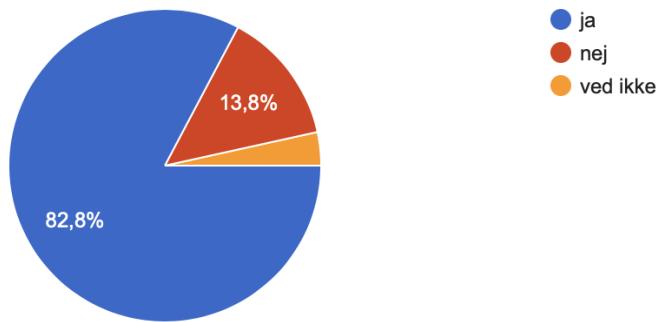


Figure 2.9: Is the interface difficult to use?

The diagram is a pie chart with the question "Is the interface difficult to use?" and three answer choices, yes, no, and other. According to the diagram, 82,8% of respondents said they have difficulty using the interface, while 13,8% said they have no difficulty using the interface, and 3,4% said they don't know.

Email and passwords

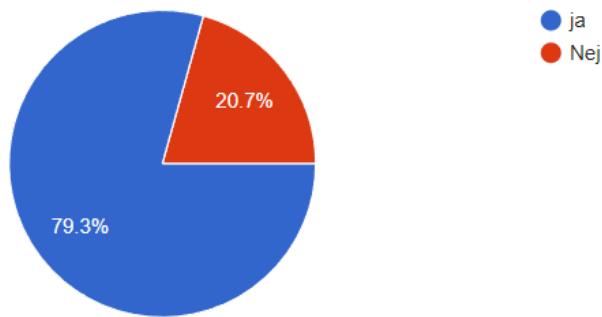


Figure 2.10: Do you have e-mail? is it hard to remember email and password for apps?

Participants in the survey were asked about having an email account, how difficult it is for them to remember their passwords for apps, and whether they have an email account. The survey's two-part question was structured as follows: "Have you got email? Is it difficult to remember your email and application passwords?" question included a Yes or a No response choice for the participants. A bar graph was used to represent the results. 2.10.

According to the survey's findings, 79,3% of respondents said it was difficult to remember their apps' email addresses and passwords, while only 20,7% said it was not. All participants also acknowledged having an email address.

Colorblindness

As an auxiliary question in the design phase, a question related to color blindness was asked, to find out what is the percentage of people who have either color blindness or vision problems. In this way, we ensure that the software that will be done is suitable for the elderly and people who have vision problems. The question is "Are you colorblind or have any other problem with seeing colors/texts?" the results are shown in the diagram2.11

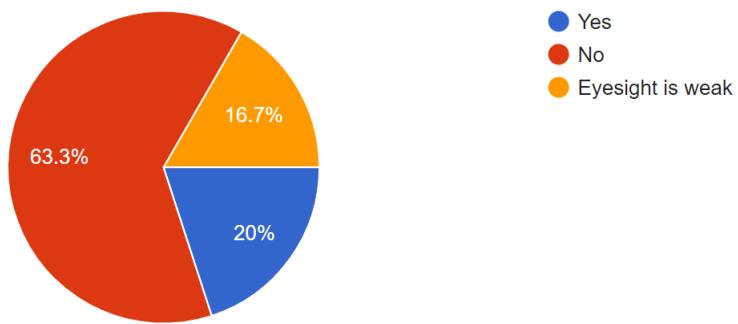


Figure 2.11: Are you colorblind or have any other problem with seeing colors/texts?

According to the diagram, 20% of respondents said they have colorblind or have problem with seeing colors/texts, while 63,3% said they are not colorblind and don't have problem with seeing colors/texts, and 16,7% said their eyesight is weak.

Types of devices

We inquired about the devices people use to communicate with their family and friends as part of our survey. Designing communication tools and apps that work with older folks' devices will be easier if we know which types of gadgets they use the most frequently.

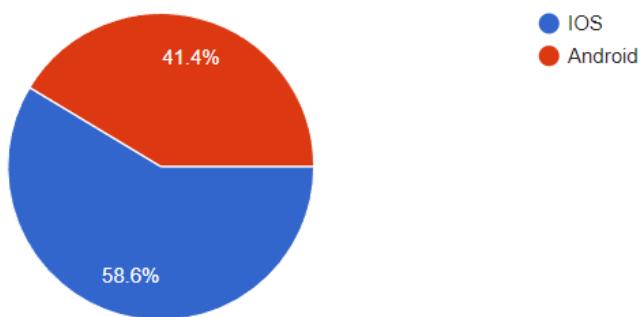


Figure 2.12: Which type of device do you have

To identify the type of device individuals owned, survey respondents were asked to select between iOS and Android. Participants were given two answer choices, and the outcomes were displayed as a bar chart.

According to the diagram, 58.6% of participants claimed to own an iOS smartphone, whereas 41.5% claimed to own an Android device. This suggests that the two categories of gadgets are distributed rather evenly.

2.2.2 Summary

In a conclusion, several themes emerged regarding the participants' experiences with communication applications. One theme was that most elderly people communicate several times a week with their family/friends. Data has been collected from 8-9 different nursing homes around in Sjælland and from a Facebook group "Alles kamp for bedre velfærd for de ældre" and 29 participants answered the survey.

Another theme that emerged was that most elderly people have difficulty communicating with family and friends. Participants described also the reasons behind their difficulty while communicating with their fam-

ily/friends, such as forgotten passwords, a difficult interface, etc. In addition to difficulty with reading and writing messages.

A third theme was the challenges they face when using communication applications, such as Whatsapp and Facebook. some of these challenges are very similar to the reasons behind the difficulty while communicating with family/friends. It's also about difficult interfaces and forgotten passwords.

A fourth theme was they have difficulty with emails. this difficulty can be hard to remember the email or the passwords.

If we study at the surveys, we find that approximately 55% of the respondents difficulty with both reading and writing. The applications' user interface looks to be the main problem, with emails and passwords appearing to be the secondary problem. For older people who might have trouble remembering them, this is especially tough.

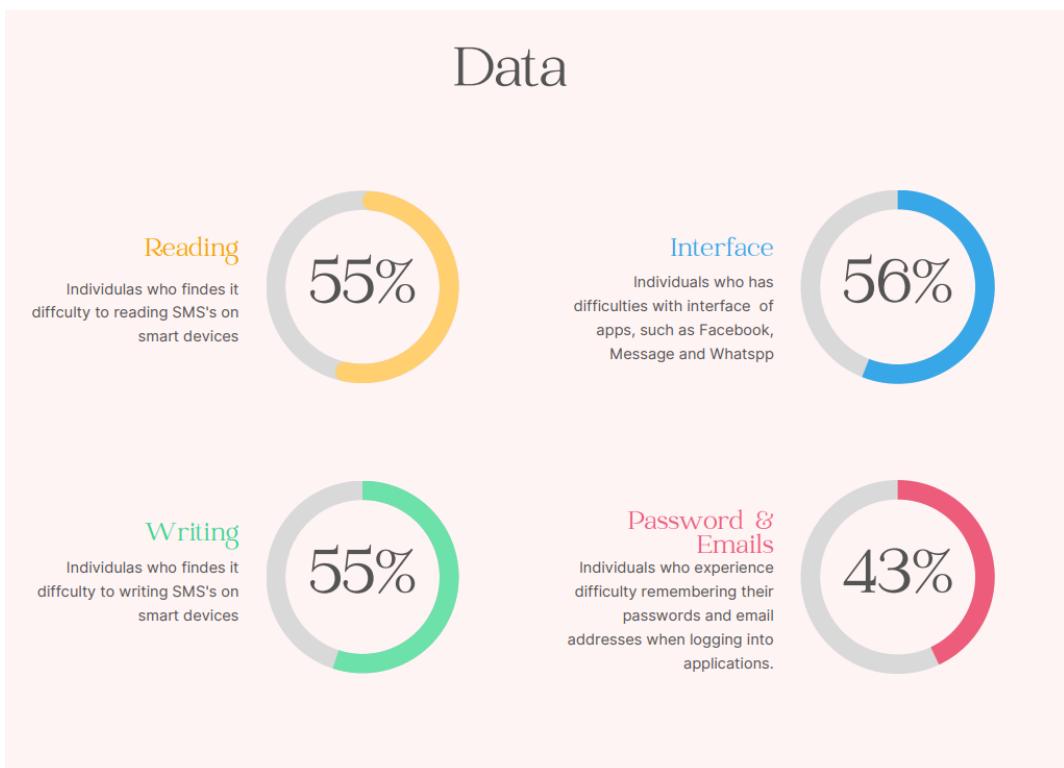


Figure 2.13

Also as a part of the interview, some questions related to the design phase were asked, such as the question regarding the colorblind or any problems with seeing colors and texts, the type of device they use, the language they use to communicate, and which applications they use.

Data has been extracted from 14-13 individuals who experienced difficulties with the four main problems out of the 27 individuals surveyed.

3 | Problem Statement

3.1 Problem Statement

Elderly individuals experience difficulties in digital communication with their family and friends due to challenges related to reading, writing, interfaces, and authentication. The issue at hand is how to develop and put into use an application solution that takes advantage of modern technologies, such as Text-to-Speech and ChatGPT API, to make communication more accessible for elderly individuals.

How can we develop a messaging application that leverages technologies, including Text-to-Speech and ChatGPT API, to improve communication for elderly individuals?

4 | Methodology & Design

This chapter provides a comprehensive overview of the methodology and technological framework used in our project. The primary objective of the project is to integrate a Problem-Based Learning approach with elements derived from Human-Centered Design (Section 4.1.4), cyber-physical systems (Section 4.1.5, and prompt engineering (Section 4.1.1). In addition to this, the chapter also presents an in-depth exploration of the design process that led to the development of the final product. It encompasses the meticulous steps undertaken to conceptualize, refine, and ultimately deliver a comprehensive solution. Additionally, the chapter focuses on key elements of the design process, including the creation of a rich picture and the meticulous clarification of requirements. The aim is to effectively incorporate new technologies into individuals' lives to enhance their overall well-being. This chapter delves into the methodical approaches adopted to address the aforementioned issues and presents the structures implemented throughout the project.

Computer science research area

Human centered interaction with focus on prompt engineering and ChatGPT API

4.1 Natural language processing

Natural language processing is a subfield of artificial intelligence that focuses on the interaction between computers and humans, particularly in understanding and processing human language. Natural language processing techniques have become increasingly important and prevalent due to the rise of natural language models.

In this section, there will be discussed about prompt engineering, ChatGPT API, Text-to-speech and Human-centered.

4.1.1 Prompt Engineering

Prompt engineering is a technique used within natural language processing (NLP) to design and optimize prompts for language models that focuses on creating and improving prompts for use with large language models. Basically, it is the process of creating requests or statements to elicit valuable responses from large language models [7]

Prompt engineering is the process of meticulously developing and improving prompts to modify language models' behavior and output. Example: *A prompt is a question, statement, or request given to an AI system to generate a specific response or output* In the context of OpenAI's GPT models like ChatGPT, prompts are the initial text or instructions provided to the model to generate a response.[8]

A prompt is a question, statement, or request that is given to an AI system in order to generate a particular response or output. Prompt engineering in the context of Natural language processing entails creating prompts that successfully communicate the desired task or topic and encourage coherent, accurate, and pertinent responses from the AI model.[7]

Prompts can also affect how specific the responses that are generated are. Engineers can regulate the model's level of detail and scope of information by customizing the prompts.

4.1.2 ChatGPT API

OpenAI has made substantial contributions to the field of artificial intelligence, especially in the area of natural language processing. Modern language models, such as the GPT (Generative Pre-trained Transformer) series, have been created by them.

GPT

Generative Pre-trained Transformer, such as GPT-2, GPT-3, and GPT-4, refers to a collection of advanced language models. These models can be fine-tuned to perform tasks like translation and summarization because they were trained on huge amounts of text, but they are unable to 'chat' with a user directly. The GPT models, such GPT-3 and GPT-4, are made to produce text that resembles human speech based on input.. These models can generate suitable responses and create logical and contextually relevant responses because they have been trained on vast volumes of text data from the internet.[\[9\]](#)

ChatGPT

The ChatGPT API is an interface provided GPT language model by OpenAI that allows developers to integrate and interact with the ChatGPT language model in their own applications. It is helpful for creating conversational agents, chatbots, or other natural language processing applications since it enables to send a sequence of messages to the API and receive model-generated responses in real-time.

The OpenAI language model ChatGPT was developed for dialogue-based applications like chatbots since it has been trained on a large volume of conversational data. It is intended to make user interactions easier by allowing users to converse or ask questions without having to adjust the model for a particular activity. When using ChatGPT as an interface, you can take advantage of its strengths as a language model to create conversational agents or include it into your program to allow for user interaction.

With the ChatGPT API, you interact with the model by sending a series of messages as prompts.'Role' and 'content' are the two components that make up each message. The "role" can be "system," "user," or "assistant," and the "content" is the message's actual text.

Normally send a system message to the assistant to configure its behavior before beginning a discussion. For instance, you could tell the assistant to imitate Shakespeare. Then you engage in a back-and-forth dialogue by switching between user and assistant messages.

You get an output after sending the messages that is a model-generated response. By adding more messages to the list and sending them again, something can carry on the conversation.

Using the API, it can have discussions that are dynamic and interactive while keeping track of context and prior messages. By defining appropriate messages and directing the model's behavior using system or user messages, it may regulate the conversation's flow. Openai train large language models.[\[9\]](#)

These AI programs are designed and generate human-like language. OpenAI's AI models can read and comprehend text, answer questions, and even write stories or articles.[\[9\]](#)

OpenAI's AI models have been trained using a lot of information from the internet, so they access to many facts and can provide helpful information. They are also able to have conversations with people.[\[9\]](#)

These models were trained by OpenAI, which also hosts the API.

4.1.3 Text-to-speech

Text-to-speech technology (TTS) is a technology of turning written text into spoken sounds. Computers and other devices may now produce speech the application of artificial intelligence and natural language processing.[10]

Sentences, paragraphs, or complete papers can be entered into a text-to-speech system as written text. The system checks the text, applies linguistic models and rules, and then combines the data to create spoken word audio output.[10]

Text-to-speech technology has a wide range of uses, including voice assistants, audiobooks, language learning tools, accessibility for those with visual impairments, and many more. It enables the audio conversion of written content, improving accessibility and communication for those who prefer or need spoken content.[10]

Text-to-speech transforms written text into generated audio. There are various steps in the procedure. The input text is first examined to determine its overall structure, including the words, phrases, and paragraphs. Punctuation, capitalization, and speech patterns are all made clear through this examination.

Text-to-speech technology turning written text into spoken sounds after linguistic analysis, prosody modeling, and speech synthesis. Together, these stages turn written text into spoken audio, which is important for voice assistants, audiobooks, language learning aids, and accessible software for those with visual impairments. Text-to-speech technology is pertinent to the project as it has the potential to assist elderly individuals with reading difficulties.

4.1.4 Human-centered

A human-centered approach focuses on creating software solutions that give users' requirements, objectives, and experiences first priority and places humans at the center of decision making and problem solving processes. It entails comprehending customer needs, carrying out user research, and taking user feedback into account at every stage of the development process. The creation of intuitive, user-friendly interfaces and overall user experience optimization are the goals. In order to make sure that software solutions match the skills, tastes, and behaviors of the desired user base, human-centered design approaches are frequently applied. Human-centered is relevant for our project because we are creating an application for elderly users to address their issues and provide a user-friendly interface.

4.1.5 Cyber physical system

To incorporate cyber-physical systems into the application, will utilize physical input in the form of user input. Physical components obtained from users will be used as part of the input. We will specifically incorporate a Text-to-Speech facility to let users enter using spoken language. Our application improves the user experience and makes it possible for seamless interaction with the software's cyber-physical systems by exploiting this physical component. Text-to-speech technology turns text into speech, offering a different way to enter data while promoting accessibility and allowing for users who might prefer or need voice-based engagement. This integration serves as another evidence of our dedication to creating software that is both physically and digitally inclusive and user-centered.

4.2 Design

This section provides an exploration of the design process, including essential components such as the creation of a rich picture and other crucial elements. Additionally, it outlines the fundamental requirements that must be incorporated during the application development phase. The section further delves into the presentation of prototypes, both in their low-fidelity and high-fidelity forms, accompanied by a thorough explanation of the decision-making process underlying their design.

4.2.1 Rich Picture

The rich picture approach is a highly valuable tool for comprehending complicated systems and situations. It enables a full perspective of the system's dynamics and possible difficulties by offering a holistic and visual picture of the many components and interactions inside a system.

In the context of a communication app, a rich picture can be used to capture the various aspects of the app, including its features, functionalities, target users, and user experience. The rich picture can also depict the stakeholders involved in the project, including the development team, investors, and users.

For this reason, Figure 4.1 shows the Rich Picture model constructed in this project.

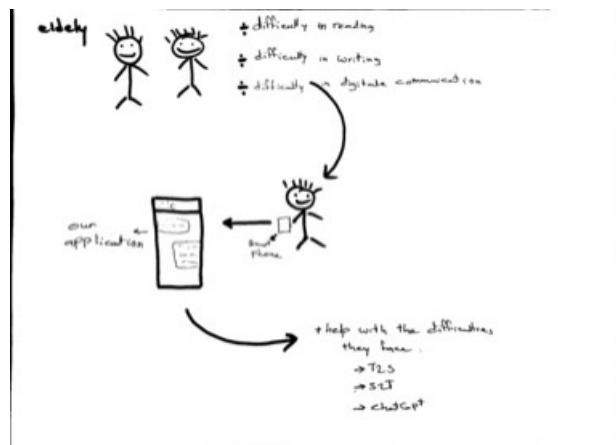


Figure 4.1: Rich picture

The model illustrates the story of elderly individuals who have difficulty communicating with their family for many reasons, such as difficulty with writing, reading, and difficult interface with many communication applications. Initially, as also detailed in Chapter 2.2.1, over half of the results in the interviews we made show the difficulty for the elderly individuals to use the nowadays application for communication. Therefore we aimed to solve this problem by creating an application with technology such as Speech To Text, Chatgpt to help them with communication.

4.2.2 Requirements

Requirements are an essential component of every project since they define what has to be accomplished and what the project will provide. They form the basis for project planning, design, development, testing, and execution. Stakeholders may make educated decisions about project scope, money, and schedules if the requirements are well-defined. Clear criteria also aid in the prevention of confusion and misunderstandings among team members, clients, and other stakeholders, ensuring that everyone is on the same page about project goals and objectives. Finally, having well-defined requirements increases the likelihood of project success while decreasing the risk of project failure.

Therefore we started gathering requirements by doing interviews in the nursing homes. Then we have documented the requirements we got in a clear, concise, and unambiguous manner. And they include **functional and non-functional** requirements, user stories, and acceptance criteria.

According to the problem analysis 2, many older people have trouble communicating and this leads to feel of isolation and loneliness. Additionally, individuals with limited access or familiarity with existing technologies may encounter challenges in utilizing them, further hindering their ability to communicate effectively. Therefore to address these challenges, it may be helpful to provide older people with phones that have simple interface and easy to use.

As mentioned above, the requirements are divided into functional and non-functional requirements:

- **The functional requirements** are those specific features and capabilities that a system or software application must perform to meet the needs of its users. These requirements describe what the system is supposed to do in terms of its functionality and how it should behave in response to user inputs or system events. In order to our application, the functional requirements are the ability for users to send and receive text from their contacts. Another functional requirement can be the ability for users to use Text-to-speech. In order to help users with writing difficulty, there will implement Speech-to-text, along with ChatGPT API. ChatGPT will be used in order to help them correct their spelling mistakes.
- **The non-functional requirements** which my include aspects such as performance and usability. For example, our application needs to be fast and responsive, have an intuitive user interface. In additional, the application should have easy to use interface. This means that the buttons used in the application should be large, with lesser text and simple interface. The application should also run on both IOS and Android devices to include a wide range of our target. Since our target are elderly people so we need to take into consideration that the ageing effects the vision. As one ages, typically around 40 years old, the lens of the eye undergoes changes that lead to a loss of flexibility and hardening. Consequently, the ability of the older eye to discern low contrast patterns is reduced. Studies by the National Eye Institute in the US indicate that contrast sensitivity begins to decline from the age of 40 and may decline by up to 83% by the age of 80. This poses significant challenges when designing technology for seniors. For instance, some combinations of text color and background color may prove challenging to read for older individuals with reduced contrast sensitivity, see figure 4.2.<https://eldertech.org/color-in-designing-technology-for>.

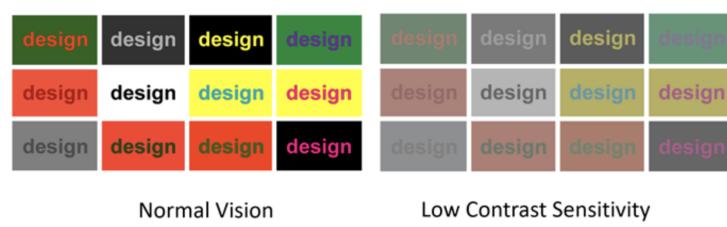


Figure 4.2: Normal vision vs. low contrast sensitivity

Therefor the colorpattern used in our application called Summer Splash color combination to ensure a better

user experience. We have made it in Canva based on the requirements and research we have done.

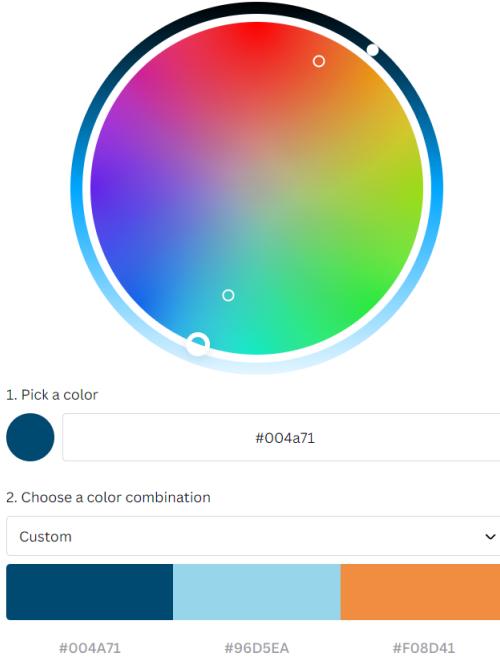


Figure 4.3: Summer Splash which used in designing the application

All of the requirements identified in this report are based on the outputs obtained from the interviews conducted with the target group . These interviews provided valuable insights into the needs of the end-users, and helped in identifying the key features and functionalities required in the software application. By gathering feedback from the users, we were able to ensure that the requirements accurately reflect the needs of them. This approach helps to ensure that the final product meets the expectations of the users [11].

4.2.3 Low-fi and Hi-fi

Low-fidelity (lo-fi) and high-fidelity (hi-fi) are two terms commonly used in the field of design to describe the level of detail and fidelity in a prototype or mockup. A low-fidelity prototype usually has minimal functionality and details, while a high-fidelity prototype is more detailed and closely resembles the final product.

Low-fidelity prototypes are frequently used in the early stages of design to quickly explore concepts and ideas.

On the other hand, high-fidelity prototypes are more detailed and may include interactive features, realistic visual design elements, and even be functional. They are beneficial for testing specific interaction patterns and user interface details.

The choice between low- and high-fidelity prototypes depends on the design process stage and the project's objectives. In our communication application project, we utilized both low- and high-fidelity prototypes based on the results obtained from our interviews with prospective users 2.2.1.

In the initial stages of the project, we created low-fidelity prototypes to quickly explore various layout and user flow ideas. As we gained more insights from our interviews, we iterated on the designs and gradually increased the fidelity of our prototypes. We employed high-fidelity prototypes to test specific interaction patterns, color schemes, and typography choices with our target audience.

4.2.3.1 Sign Up Screen

The process of making the low-fidelity prototypes started with sketching the idea of an sign up screen, with the requirements in mind. This ended up like this:

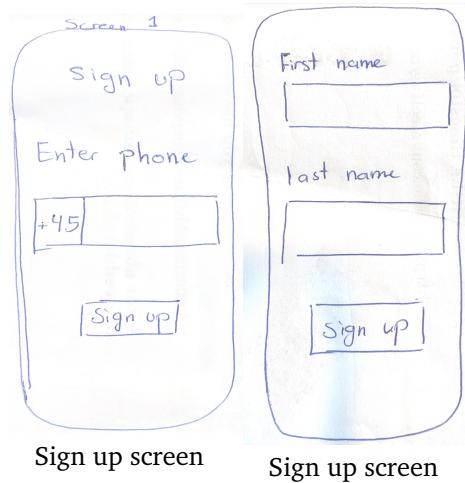


Figure 4.4: Low-fidelity prototypes for Sign Up screens

Once our low-fidelity prototype was approved by our supervisor, we proceeded to develop a high-fidelity prototype using a similar process. For this stage, we utilized Figma, a web-based tool that specializes in creating user interface designs and prototypes in an intuitive manner. Our high-fidelity prototype incorporated more refined details and interactive elements to closely resemble the final product. The resulting prototype can be viewed below.

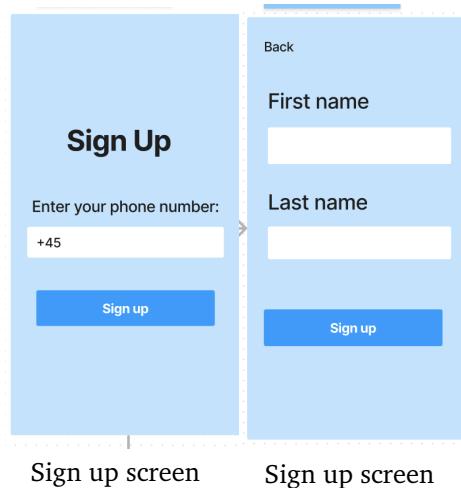


Figure 4.5: Hi-fidelity prototypes for Sign Up screens

4.2.3.2 Home Screen

The home screen showed a list of names of that person the user have contacted them. The main focus while creating lo-fi prototype was getting a screen with simple and easy overview, without too many features. This was implemented by creating dynamically a list of the names, where each name has a name and an image.

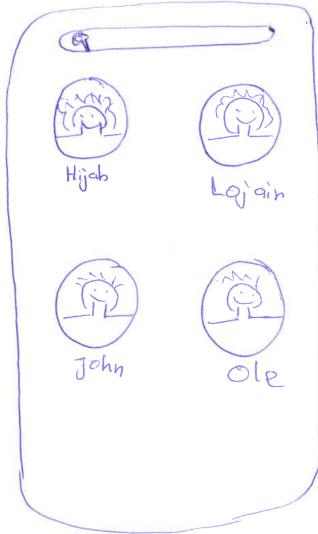


Figure 4.6: Low-fidelity prototype for Home Screen

After the low-fidelity prototype had been created and the overall design had been shown and approved by the supervisor, the next step was to develop high-fidelity prototype for home screen of the app. This was done using the online design tool Figma, which specializes in creating prototypes in a user-friendly and intuitive way.



Figure 4.7: Hi-fidelity prototype for Home Screen

4.2.3.3 Chat Screen

For the chat screen of our application, we created a low-fidelity prototype in the same way as we done with other screens. We sketched out various versions of the chat screen on paper, considering different options for displaying messages, user avatars, and input fields. We also tested out different layouts for the chat and the message input area. Once we had several sketches to work with, we evaluated each one to determine which would work best for our target audience. Based on our evaluation, we selected the one we thought is the best based on the target's needs that enable in volume, reduce complexity and simple interface.

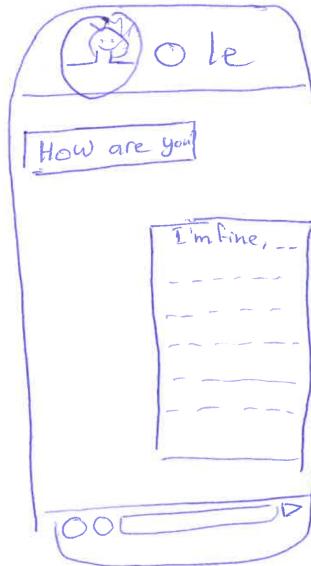


Figure 4.8: Low-fidelity prototype for Chat Screen

In the high-fidelity prototype for the chat screen, we aimed to create a more detailed and realistic representation of the final product. We used Figma to create a polished and interactive prototype that closely resembled the actual user interface. We refined the visual design of the chat screen by adding more realistic color schemes and typography choices. Furthermore, we added navigation bar in the bottom of the screen to give users easy access to different screens such as profile and home screen. We also incorporated a back button to allow users to easily navigate back to previous screens.

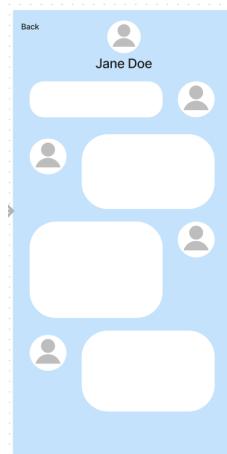


Figure 4.9: Hi-fidelity prototype for Chat Screen

4.2.3.4 Profile Screen

The group's primary objective for the profile screen design was to ensure that it provides users with a clear and straightforward overview, avoiding any confusing features. The demo of the profile screen was designed to incorporate a personalized element right from the moment the user accessed the screen. To achieve this, a profile picture was placed in the center of the screen, with the user's name displayed above it.

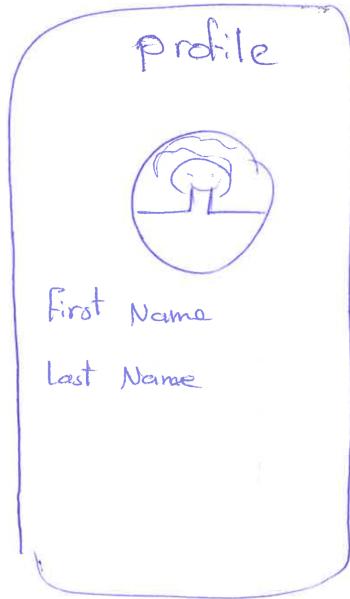


Figure 4.10: Low-fidelity prototype for Profile Screen

The high-fidelity prototype for the profile screen aimed to create a more polished and realistic version of the design that closely resembled the final product. The design focused on delivering a personalized experience for the user, with a clear emphasis on the user's name and profile picture at the center of the screen. The high-fidelity prototype included a more refined and visually appealing layout, with the user's profile picture and name presented prominently at the top of the screen.

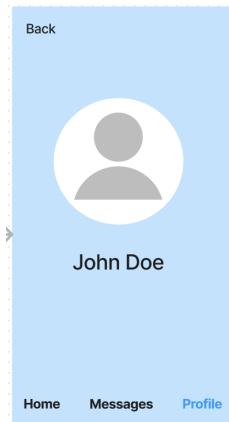


Figure 4.11: Hi-fidelity prototype for Profile Screen

By utilizing a combination of low- and high-fidelity prototypes, we were able to effectively test and refine our design ideas based on user feedback. This approach allowed us to create a more effective and user-friendly communication application that maybe satisfies the needs and expectations of our target audience.

5 | System Architecture

5.1 System Architecture

In this chapter, we will explore the system architecture of our application, which is responsible for enabling seamless and efficient communication for elderly individuals. We will delve into the structure, components, and their interactions, highlighting how they contribute to the overall functionality and performance of the application. By understanding the system architecture, we can gain insights into how the different parts of the application are organized, how data flows between components, and how various technologies and services are utilized to deliver a user-friendly experience.

The architecture overview of a system refers to the high-level structure and design of the system. Describing its elements, how they work together, and how information or processes move as a whole. A system's specific architecture can change based on its goal and domain.

A representation of a high-level architecture overview of the application can be seen in figure 5.1.

The application utilizes AWS Amplify, specifically AWS Cognito for user authentication and AWS AppSync for defining a GraphQL API schema. The AppSync API uses resolvers to communicate with particular data sources and map requests and interact. The system includes data models for messages, users, and chat rooms that are organized in accordance with the needs of each. Overall, AWS Amplify simplifies development by providing pre-configured services and abstracting the underlying infrastructure. AWS AppSync simplifies the process of building APIs for applications, enabling effective data retrieval, real-time updates, and integration with different data sources and authentication frameworks.

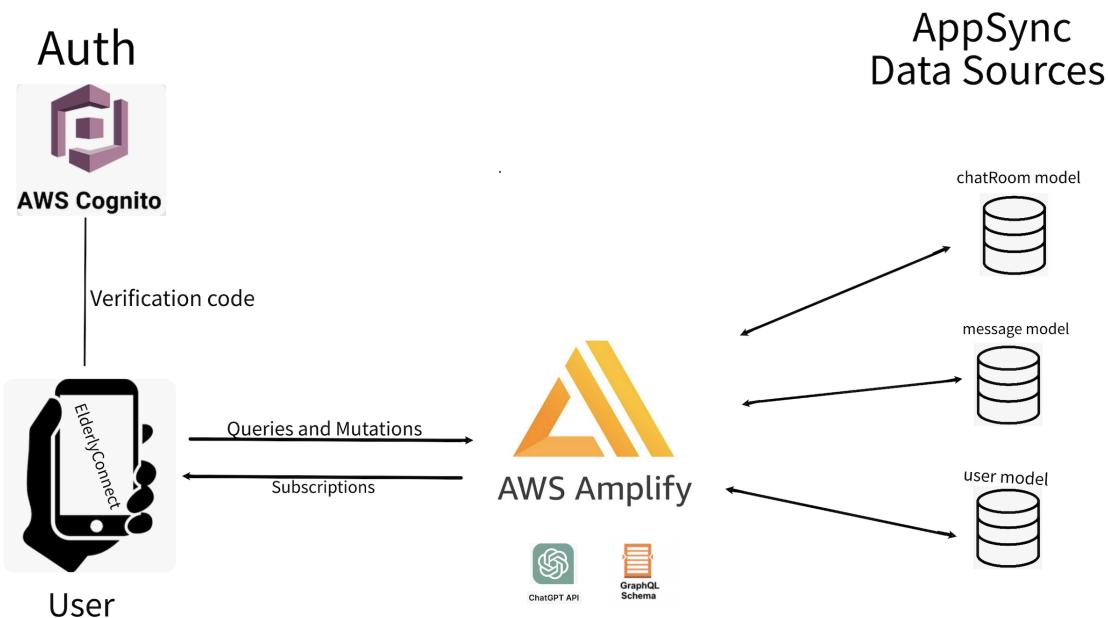


Figure 5.1: Architecture overview

6 | Implementation

6.1 Implementation

This chapter provides an overview of the implementation process for the communication application designed for elderly individuals with reading and writing difficulties. It focuses on the front-end and back-end aspects of the system and highlights the frameworks used. The chapter outlines the implementation approach and key considerations, ensuring the app's usability and functionality.

6.2 Front-end

It's crucial to comprehend the framework we used to create our application before getting into the specifics of how our front-end was implemented. Our application will be built on the open-source React Native framework, which was created by Facebook. By utilizing JavaScript and React to build user interfaces and features that closely match native experiences on both the iOS and Android platforms, this framework makes it possible to create cross-platform mobile applications.

Several considerations went behind the choice to use React Native. Firstly, the development process was made easier by our significant expertise and knowledge of React Native. Additionally, React Native has the option to simultaneously construct interfaces for both platforms, which is what we needed to build an application that serves users of both iOS and Android.

By utilizing React Native, we can make use of its powerful features, draw on our knowledge, and create a mobile application that provides native-like experiences on several platforms.

6.2.1 Home screen

To enhance code organization and facilitate a comprehensive understanding, our approach involved breaking down the code into distinct components. This strategic decision offers several benefits, including improved code maintainability, reusability, and enhanced comprehensibility. These components are designed to be utilized consistently across multiple screens throughout the application. By implementing this modular structure, we ensure that updates, modifications, and troubleshooting can be performed efficiently while promoting a streamlined development process and fostering code that is easier to grasp and work with.

An illustrative example can be observed in the Home Screen 6.1, where the presence of the ChatList component showcases the rendering of individual chat list items, each encompassing vital user details such as their image and name. Additionally, this component assumes the responsibility of facilitating seamless navigation to a dedicated chat screen upon user interaction, specifically when an item within the chat list is pressed. This exemplifies the effective utilization of components to encapsulate discrete functionalities, promoting code modularity and enhancing user experience within the application.

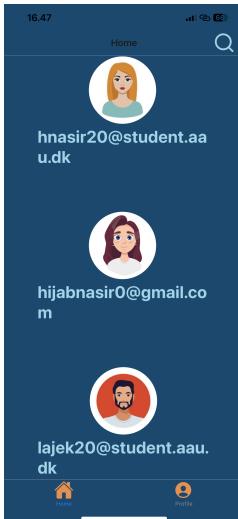


Figure 6.1: Home screen

6.2.2 Chat screen

Within the chat screen, our focus was on enhancing accessibility for elderly individuals by integrating technologies that facilitate ease of writing and reading lengthy messages. To achieve this objective, we incorporated text-to-speech, speech-to-text, and ChatGPT functionalities.

These advanced technologies are seamlessly integrated into the lower section of the user interface, specifically in the input box area. This integration enables users to input their messages while taking advantage of the integrated functionalities of ChatGPT.

To enhance the user experience and streamline communication, we have incorporated two distinct buttons that leverage the capabilities of ChatGPT. The first button, highlighted in orange, serves the purpose of sentence rewriting. When users encounter a sentence with errors or misspellings, they can simply click on the orange button. ChatGPT then processes the input and rewrites the sentence correctly. For example, if a user inputs "Hwo ar uor" and clicks the orange button, the corrected sentence "How are you" will appear on the screen.

The second button, highlighted in blue, offers assistance in crafting messages. By clicking on this button, users can access the guidance and suggestions provided by ChatGPT. This feature assists users in formulating their messages, helping them overcome any difficulties they may encounter while expressing their thoughts.

By incorporating these intuitive functionalities into the user interface, we aim to simplify the communication process for elderly individuals. The seamless integration of ChatGPT's capabilities within the application's input box area empowers users to effortlessly engage in digital conversations, ensuring their messages are accurate and effectively conveyed.

ChatGpt serves as a powerful tool for rewriting messages and generating creative ideas. By leveraging the capabilities of ChatGpt, users can receive suggestions and alternative versions of their messages, thereby enhancing their overall communication experience. This integration aims to foster a more efficient and engaging interaction between the user and the application.



Figure 6.2: Chat screen

Text to speech

Considering that most devices already provide speech-to-text capabilities within their keyboards, we opted to utilize this existing functionality to save time and allocate resources toward further application improvements. Consequently, our implementation focused on text-to-speech functionality using the "Speech" module from the expo-speech package. This module offers a comprehensive set of APIs that seamlessly integrate text-to-speech capabilities into our Expo mobile application.

By leveraging "Speech.speak()" method to convert text into speech, allowing users to listen to messages instead of reading them.

Below is an example showcasing the implementation of this functionality:

```

1 import * as Speech from "expo-speech";
2
3 dayjs.extend(relativeTime);
4
5 const Message = ({ message }) => {
6   const [isMe, SetIsMe] = useState(false);
7
8   useEffect(() => {
9     const IsMyMessage = async () => {
10       const authUser = await Auth.currentAuthenticatedUser();
11
12       SetIsMe(message.userID === authUser.attributes.sub);
13     };
14
15     IsMyMessage();
16   }, []);
17
18   const handlePress = () => {
19     Speech.speak(message.text, { rate: 0.75, language: "da-DK" });
20   };

```

Listing 6.1: Text to speech implementation

Through the integration of these technologies, we aimed to enhance the overall user experience by providing elderly individuals with alternative methods to engage with the chat application, enabling them to both listen to and generate messages effectively.

Speech to text

Speech-to-text technology is an innovative solution that utilizes machine learning and artificial intelligence to convert spoken language into written text. By enabling devices and applications to comprehend and process human speech, it offers users the convenience of interacting through voice input rather than relying on traditional typing methods. Recognizing the potential benefits of this technology, we considered integrating it into our application to assist a specific target group that faces challenges with writing, as revealed through our conducted interviews.2.2.1.

Our objective was to leverage speech-to-text technology within our application, allowing users to compose messages with ease. By utilizing this technology, individuals who struggle with writing can rely on voice input to convey their thoughts effectively. Consequently, it would enhance accessibility and inclusivity, catering to the needs of our target audience. In addition, we can use it in the application so they can write the message with the help of it, and the input we get from this technology we can use in the ChatGPT button to generate different prompts.

However, it is important to note that speech-to-text technology is already present in a multitude of devices, particularly in modern smartphones. For instance, devices like the iPhone XR, released as early as 2018, have built-in speech recognition capabilities that facilitate dictation of text messages, web searches, and email composition. Considering that the desired technology is readily available on the users' devices, we made a strategic decision not to invest significant resources and time into implementing a feature that is already accessible to them.

As a result, we opted not to integrate speech-to-text technology directly into our application. Instead, we recognized the existing functionality provided by the device's keyboard, which already supports speech input. By leveraging this built-in feature, users can conveniently utilize speech-to-text capabilities without requiring additional implementation efforts on our part.

Therefore, we took advantage of the already-found technology and work on it how can we use it to help the user. So the user can write a message with the help of s2t from the keyboard, and use the given technology in the application to rewrite it, since it might be the message has some writing mistake.

6.2.3 Profile screen

The profile screen, deliberately designed with simplicity in mind, offers a straightforward user experience. It primarily showcases essential information such as the user's name and profile image, along with a discrete sign-out button. The intentional simplicity of this screen aims to minimize user confusion and ensure seamless interaction.

To prevent accidental sign-outs and ensure continuous user access, the sign-out button is intentionally designed to be less prominent. This approach mitigates the risk of users inadvertently signing out and experiencing difficulties related to retaining their email or password information. By prioritizing user convenience and maintaining a seamless sign-in experience, we strive to enhance overall user satisfaction and reduce potential disruptions.



Figure 6.3: Profile screen

6.3 Back-end

We chose AWS Amplify for the back-end with a database with a GraphQL API based on the data model we created. we chose AWS Amplify because it simplifies development and is integrated. Amazon Web Services (AWS) offers a development framework and suite of tools called AWS Amplify that make it easier to create mobile applications. It provides functionalities for database-related back-end services.

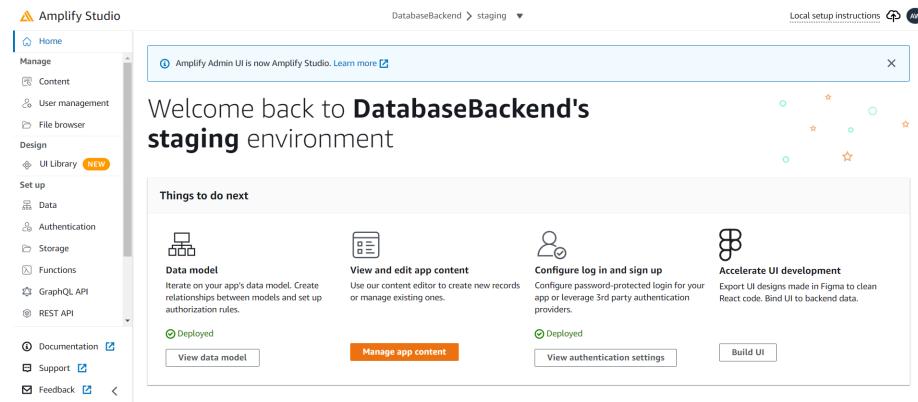


Figure 6.4: AWS amplify studio

6.3.1 Authentication

For our mobile application, we have implemented email-based authentication with a verification done using AWS Amplify. With authentication method allows users to sign in to the mobile application using their email address and will receive a verification code to confirm their identity. The verification code adds an extra layer of security and ensures that only authorized users can access the application.

AWS Amplify offers a high-level authentication API that makes it simple to integrate with authentication methods, such as email-based authentication with a verification code. Through the Amplify CLI or the Amplify Console, we could configure the authentication settings with an email verification template. We have used pre-built UI components from AWS Amplify that handle the authentication flow.

```

1 import { Amplify, Auth, API, graphqlOperation } from "aws-amplify";
2 import awsconfig from "./src/aws-exports";
3 import { withAuthenticator, AmplifyTheme } from "aws-amplify-react-native";
4
5
6 //Configure Amplify with the provided AWS configuration
7 Amplify.configure({ ...awsconfig, Analytics: { disabled: true } });
8
9 function App() {
10   useEffect(() => {
11     const syncUser = async () => {
12       //Get Auth user
13       const authUser = await Auth.currentAuthenticatedUser({
14         bypassCache: true,
15       });
16
17       //query the database using Auth user id(sub)
18       const userData = await API.graphql(
19         graphqlOperation(getUser, { id: authUser.attributes.sub })
20       );
21
22       //if there is no users in db, create one
23       if (!userData.data.getUser) {
24         console.log("user already exists in the DataBase");
25         return;
26       }
27
28       const newUser = {

```

```

29     id: authUser.attributes.sub,
30     name: authUser.attributes.email,
31   };
32 
33   await API.graphql(graphqlOperation(createUser, { input: newUser }));
34 };
35 
36 syncUser();
37 }, []);
38 return (
39   <View style={styles.container}>
40     <Navigator />
41 
42     <StatusBar style="auto" />
43   </View>
44 );
45 };
46 
47 
48 const styles = StyleSheet.create({
49   container: {
50     flex: 1,
51     backgroundColor: "#BD8451",
52     justifyContent: "center",
53   },
54 });
55 // Define a custom theme for authentication screens(amplify UL component)
56 const customTheme = {...AmplifyTheme,
57   button: {
58     ...AmplifyTheme.button,
59     backgroundColor: '#004A71',
60     borderRadius: 10,
61   },
62   ....
63 };
64 export default withAuthenticator(App, {theme: customTheme});

```

There has also implemented a sign-out functionality using Auth.signOut from AWS Amplify's Auth module. So users can sign-out out of the application whenever they want to.

```

1 / import { Auth } from "aws-amplify";
2 
3 const ProfileScreen = () => {
4   return (
5     <View style={styles.container}>
6       <View style={styles.profileContainer}>
7         <Ionicons
8           name="person-circle"
9           size={250}
10          color="#F08D41"
11          style={{ marginBottom: 10 }}
12        />
13        <Text style={styles.name}>Lojain</Text>
14      </View>
15      <View style={styles.buttonContainer}>
16        <Button
17          onPress={() => Auth.signOut()} //Auth.signOut is called when the user press on the
18          title="Sign Out"
19          color="black"
20          size={500}
21          style={styles.button}
22        />
23      </View>
24    </View>
25  );
26};

```

6.3.2 Data models

To manage the data within an application, the data model is important. It offers a logical representation of the data entities, properties, and connections within the application. The goal of the data model is to logically and consistently arrange and store data so that information may be easily retrieved, modified and maintained. We can lay a solid basis for creating a chat application that enables seamless user communication by defining the structure and relationships of our data elements.

We created data models for ChatRoom, User and Message.

- ChatRoom Data Model: The ChatRoom Data Model represents a chat room in the application. It has a one-to-many relationship with the Message model and a many-to-many relationship with the User model.
- Message Data Model: The Message data model represents a message in the chat application. It has a one-to-many relationship with the ChatRoom model and a one-to-many relationship with the User model.

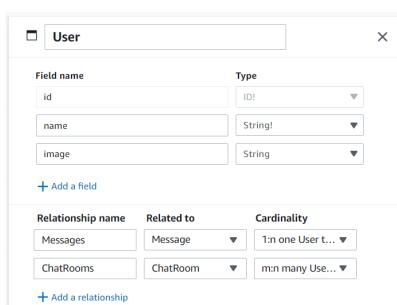


Figure 6.5: Data model for user

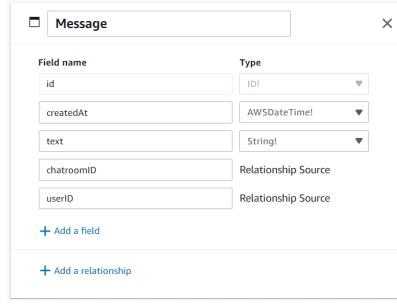


Figure 6.6: Data model for message

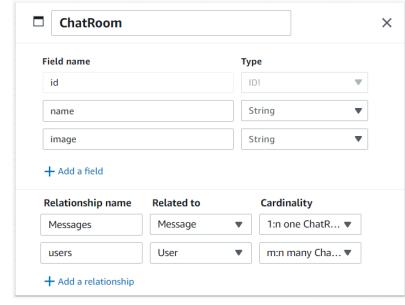


Figure 6.7: Data model for ChatRoom

6.3.3 GraphQL APIs

Using AWS Amplify, we defined the Graphql APIs based the the data models that we created which specified the structure and relationships between the entities, the data models are ??.

GraphQL API allows the front end to access data from one or more data sources with a single network request using GraphQL.

We have implemented GraphQL queries, mutations, and subscriptions to interact with your back-end services.

- Queries in GraphQL are used to fetch data from the server.
- Mutations in GraphQL are used to modify or create data on the server.
- Subscriptions in GraphQL enable real-time data updates by establishing a persistent connection between the client and the server. With subscriptions, can receive data updates in real-time as they occur on the server in the application

6.3.4 How is back-end connected with front-end

The application's front-end and back-end are connected through APIs and data fetching.

In the front-end components and screens, we implemented the necessary AWS Amplify modules, such as Auth, API, and graph operation, to interact with the back-end services.

Storage

The application uses a storage service provided by AWS Amplify To effectively store and handle different sorts of data. The storage service plays a crucial role in handling data related to chat rooms, messages, and user information.

6.3.5 ChatGPT API

In the application, we have implemented ChatGPT API by OpenAI to enhance the chat functionality and provide advanced text generation capabilities. The API key obtained from the OpenAI platform is utilized to establish a secure connection with the ChatGPT service.

Our application integrates OpenAI's ChatAPI to help elderly individuals who have writing difficulties communicate more effectively. We help people get their ideas across effectively despite language barriers through the use of two well-crafted prompts. Through advanced natural language processing, our application provides prompt and accurate responses, fostering inclusive and meaningful interactions. Our goal is to empower the elderly community and enhance their communication capabilities, making a positive impact on their lives.

To interact with the ChatGPT API, the application makes use of the fetch function to send HTTP requests to the API endpoint. The API endpoint 6.8 is accessed with the appropriate HTTP method, headers, and request payload.

To implement ChatGPT API in our application, we utilize the provided API key from OpenAI. By sending HTTP requests to the ChatGPT API endpoint 6.8, we leverage its language generation capabilities. The "Any prompt" function demonstrates how we utilize the API.

Implementation

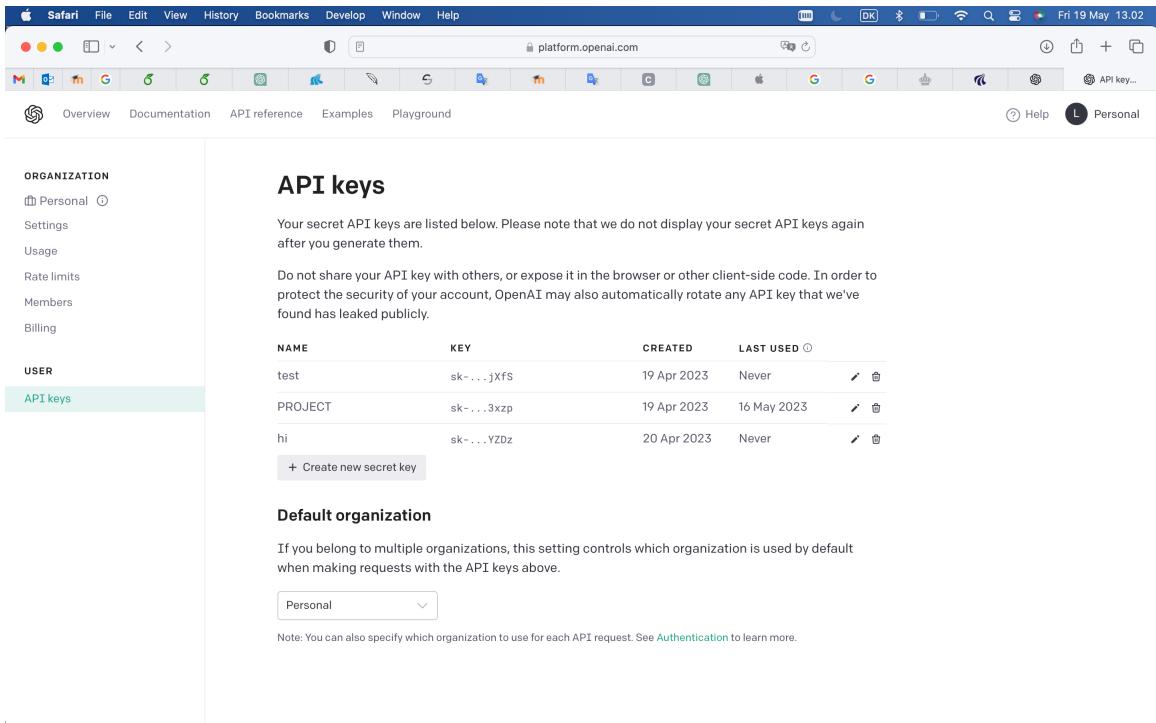


Figure 6.8: API keys

```
1 /   const OPENAI_API_KEY = "sk-Li4Hawd6Rkv2isvDbfSoT3B1bkFJCopMn8jGU3xGXjB43xzp";
2
3 async function generateRewrittenText(newMessage) {
4   const response = await fetch(
5     "https://api.openai.com/v1/engines/text-davinci-002/completions",
6   {
7     method: "POST",
8     headers: {
9       "Content-Type": "application/json",
10      Authorization: `Bearer ${OPENAI_API_KEY}`,
11    },
12    body: JSON.stringify({
13      prompt: `Rewrite this sentence: ${newMessage}`,
14      max_tokens: 50,
15    }),
16  }
17 );
18
19 const data = await response.json();
20 return data.choices[0].text;
21 }
```



```
1 /   const OPENAI_API_KEY = "sk-Li4Hawd6Rkv2isvDbfSoT3B1bkFJCopMn8jGU3xGXjB43xzp";
2
3 async function generateRandomText(newMessage) {
4   const response = await fetch(
5     "https://api.openai.com/v1/engines/text-davinci-002/completions",
6   {
7     method: "POST",
8     headers: {
9       "Content-Type": "application/json",
10      Authorization: `Bearer ${OPENAI_API_KEY}`,
11    },
12    body: JSON.stringify({
13      prompt: `I need some assistance in crafting a message that I'm about to send. Can
14      you provide me with some creative ideas: ${newMessage}`,
15      max_tokens: 50,
16    }),
17  }
18 );
```

```
16     }
17 );
18
19 const data = await response.json();
20 return data.choices[0].text;
21 }
```

Prompt implemented

Two specific functionalities are implemented using the ChatGPT API: text rewriting and generating random text suggestions.

For text rewriting, the application sends a prompt containing the sentence to be rewritten to the API. The response received from the API contains the rewritten text, which is then displayed within the application. This feature allows users to obtain alternative versions of their input text.



Figure 6.9: "Rewrite this sentence" in the application

The generation of random text suggestions involves sending a prompt requesting creative ideas for a message. The API response includes a randomly generated text suggestion, which can serve as inspiration or assistance in crafting a message. This functionality enables users to explore different text options and enhance their communication within the chat rooms.

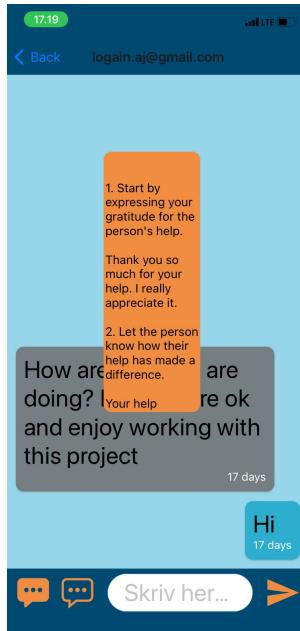


Figure 6.10: "I need some assistance in crafting a message that I'm about to send. Can you provide me with some creative ideas" in the application

Exploration of Different Prompts

During the implementation of the ChatGPT API, we faced several experiments with implementing different prompt to achieve the desired results. we experimented with several prompts to enhance the functionality of our application.

- Provide a brief summary of conversation.
- Generate a list of potential reasons for this message
- Give 3 important bullet points of latest received message.
- Can you tell me if I have a meeting scheduled
- I wanted to confirm the time of our appointment or meeting

The table can be seen 6.1, which provides a comprehensive overview of various prompts, along their corresponding outputs and the reasons why they are considered unsuitable for our case.

Prompts	Outputs	Reasons
Provide a brief summary of conversation	"The conversation was about the speaker's experiences with studying abroad" See figure 6.11	Output is not correct
Generate a list of potential reasons for this message	"The speaker and the person they are talking to discuss the concepts of change and how people often resist it even when it is necessary. the speaker gives the example of how people often don't like change in their routine even when it would be beneficial" See figure 6.11	The output is very different from what the prompt is requesting for.
Give 3 important bullet points of latest received message	"I could not find a transcript pf the conversation" See figure 6.12	The output is significantly different and does not match with the expected response
Can you tell me if I have a meeting scheduled	"i cannot tell you if you have meeting scheduled because the information is undefined" see figure 6.12	The reason provided by this prompt is not relevant in our case since it fails to determine if a meeting is scheduled,despite the conversation mentioning specific timing for the meeting..
I wanted to confirm the time of our appointment or meeting	"Our meeting is scheduled for 3:00 pm" see figure 6.13	The output is incorrect in its meeting or appointment information.

Table 6.1: Unsuccessful prompts

These prompts have several limitations and difficulties that made them useless for our application are lack of contextual understanding, incomplete, incoherent responses and difficulty in extracting specific information.

Prompt outputs

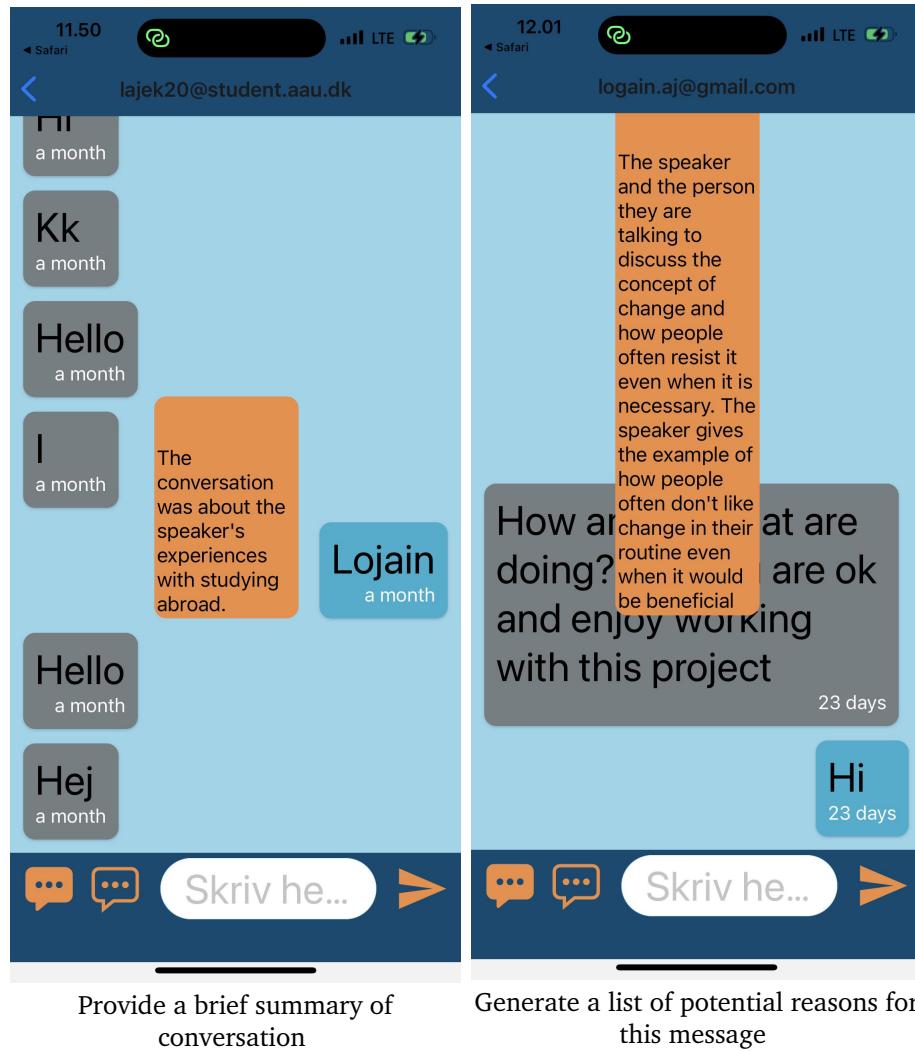
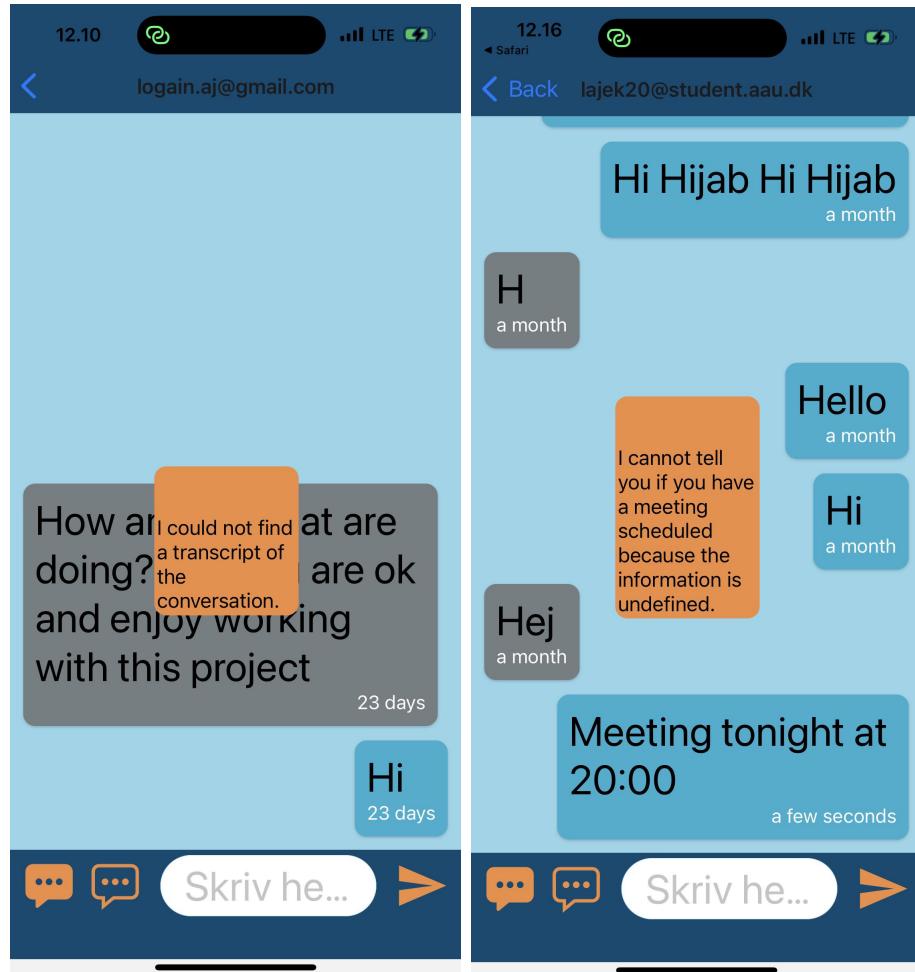


Figure 6.11: Different prompts

Implementation



Give 3 important bullet points of latest received message

Can you tell me if I have a meeting scheduled

Figure 6.12: Different prompts

Implementation



Figure 6.13: I wanted to confirm the time of our appointment or meeting

7 | Testing

This chapter focuses on conducting comprehensive testing to evaluate every aspect of the application. A series of tests will be performed to ensure the functionality, usability, and performance of the application. Usability testing is the first step in the testing process, which also includes software quality control, unit testing for the front end, and integration testing for the GraphQL APIs.

To determine whether the application effectively satisfies user requirements, usability testing is important. We can identify any usability flaws, navigational challenges, or areas that want improvement by watching how users interact with the program and gathering their feedback. The purpose of this test is to assess the user experience overall and make sure the program is simple to use.

The management of software quality is the following step. This test focuses on confirming that the program complies with accepted quality standards and best practices. It involves carrying out a codebase analysis, spotting potential security holes or performance hiccups, and putting safety, maintainability, and reliability improvements into action. We can reduce the likelihood of software errors and increase the overall resilience of the program by implementing thorough quality management.

To check the functionality of certain functionalities or components within the application, unit testing for the front-end is an important phase. This test includes separating and evaluating each function or component to ensure that they work as intended.

Integration testing will test the back-end, in addition, and concentrates on evaluating the GraphQL APIs that the application uses. This test wants to validate the proper functioning and effortless integration of the APIs into the larger application ecosystem. By emphasizing effortlessness, we seek to ensure that the integration of the APIs does not introduce any complexities or hinder the overall user experience. Assuring the overall stability and reliability of the program, we can identify any inconsistencies, compatibility problems, or mistakes that may develop during integration by validating the communication and data flow between various components and systems.

7.1 Usability testing

The purpose of usability testing is to evaluate a system's or application's usability by monitoring how users interact with it and gathering their comments. It aids in user behavior analysis, usability problem identification, design decision validation, and iterative improvement. In the end, usability testing seeks to develop a program that satisfies user needs, raises user satisfaction, and follows the intended requirements.

The group discussed whether it would be feasible to do the usability testing on the same user who has already been interviewed and at the same nursing homes.

The group created the following usability test to test it on real users in nursing homes and other early populations.

The user will then be required to complete several different tasks, which can be seen in table 7.1, each of which focuses on a different aspect of the application's functionality. Each task will be given as a case, with a setup reflecting what the user wants to achieve or do in the application. and an open-ended question that allows the user to interpret the task in their way. The group member who will serve as an observer for each activity makes notes regarding the actions and behaviors used to attempt to perform the task, whether or not it has been completed, and the time required to do it, if successful. The following tasks will be performed as a think-aloud test, which means that the user will be required to speak out at various points during the test to express what they are currently thinking, their findings on the task, as well as their reasons for taking a particular action.

Testing

Tasks:	Observations:	Completed:	Time to complete:
1) Create a new account using the Authentication. What do you do?			
2) Login in application with already created account.			
3) Use the search function to find a friend and add them to your contacts list. What do you do?			
4) Navigate to the profile and sign out. What do you do?			
5) Navigate to the between screens and start a new conversation with a contact. What do you do?			
6) send a message to the contact and receive a reply (without any technologies). What do you do?			
7) Send a message to the contact (with AI - rewrite technology) What do you do?			
8) receive a reply and use AI 3 bullets AI. What do you do?			
9) Use the text-to-speech functionality to listen to the received/send message. What do you do?			
10) Use the speech-to-speech functionality to write a message then use AI functionally to rewrite it. What do you do? What do you do?			

Table 7.1: Usability test

Testing performed

The forthcoming section will include the responses obtained from the test.

Tasks:	Observations:	Completed:	Time to complete:
1)	Easy to login	Yes	56 seconds
2)	Took some time to remember the email:	Yes	2 minutes
3)	Was very quick and liked the interface while searching for contact-list	Yes	around 1 minute
4)	Could easily go around in the app	Yes	22 seconds
5)	None	Yes	23 seconds
6)	Had spelling mistakes while typing a message and was thinking a lot	Yes	2 minutes
7)	got very excited to see, that this a thing and works so good/smart	Yes	58 seconds
8)	"So there is no need to read the whole message" said user	Yes	20 seconds
9)	Very nice, I have very hard time reading messages	Yes	7 seconds
10)	Liked the idea behind this idea and really found it very helpful to send message, since she had problems with writing messages, this technology helped a lot	Yes	1 minute

Table 7.2: Person age 63

Testing

Tasks:	Observations:	Completed:	Time to complete:
1)	The user could not remember his email addressee	yes	58 seconds
2)	Got confused between username and email, since it uses username, but you actually need to write email instead of username	Yes	49 seconds
3)	Easy to find the search function, because of the easy interface with less text	Yes	around 1 minute
4)	Want to add a picture, but application does not have that functionality and Sign-out was easy	Yes	18 seconds
5)	Very easy to navigate around between profile, contact and home screen, easily started a conversation	Yes	28 seconds
6)	Sent a message on normally speed, had a bit difficult writing a word	Yes	1 minute and 29 seconds
7)	Moderator told about the AI(rewrite) tech and the user found it very useful and, said will it also correct spelling, and got it very useful	Yes	47 seconds
8)	Like this tech a lot, found it useful for when the user could not understand the meaning of the received message	Yes	1 minute 10 seconds
9)	Found it very useful	yes	59 seconds
10)	No need to write the long message, which user always wanted, and the AI made it very easy, since the user should not care about the wording/spelling and structure of the message	Yes	2 minute

Table 7.3: Person age 57

Tasks:	Observations:	Completed:	Time to complete:
1)	Created an account without any problem, but a bit confused with that phone number option.	Yes	2 minutes and 34 seconds
2)	Got confused with username and email	Yes	1 minute
3)	like the interface/colors of the application	Yes	1 minute and 23 seconds
4)	Easily found the sign-out button because of less text	yes	59 seconds
5)	Found the contact very easily on home screen	Yes	40 seconds
6)	Was typing very slowly, had a hard time finding the letters on keyword. Then sent message half, did send a completed finished message	No	3 minutes
7)	Used 7. Half wrote message and got it completed by AI option, and got very shocked about how this works and liked it a lot	Yes	31 seconds
8)	liked this tech a lot as well, and saying so there is no need to read message, and the user can have a long conversation with her friends, which she always wanted but had problems with reading a lot	Yes	20 seconds
9)	The user found it very helpful because sometime the user needs to read old conversation in order to remember what they were talking, and the user has some problems with reading, so now with the help of these tech, the user don't need to read the message but instead listen to them both received and sent messages.	Yes	2 minute and 51 seconds
10)	Again, got shocked very this both tech, that even though you say something wrong or incorrect, so it will get corrected in this app with other correcting options.	Yes	2 minutes

Table 7.4: Person age 62

Tasks:	Observations:	Completed:	Time to complete:
1)	None	Yes	1 minute
2)	None	Yes	59 seconds
3)	Found it very easily	Yes	40 seconds
4)	Easily found it as well, liked the colors and easy interface. easily to navigate around	Yes	45 seconds
5)	Got there very easily	Yes	49 seconds
6)	Small keyboard	Yes	1 minute and 20 seconds
7)	Found it very useful and liked it a lot, also asked if this technology can be added on her phone	Yes	1 minute
8)	Liked it, found it easy to understand the messages	Yes	1 minute
9)	Found it very useful, since she could not read very messages	Yes	1 minute and 47 seconds
10)	She does not like speech-to -text a lot	Yes	3 minutes

Table 7.5: Person age 60

Summary

In conclusion, the results of the usability test were largely positive, showing that the application is quite effective at resolving people's initial problems. The use of OpenAI and text-to-speech technologies proved to be good solutions, with users particularly appreciating the text-to-speech feature's ability to do away with the necessity for message reading. Many users had difficulty with reading, however, OpenAI prompts like "Rewrite this sentence" and "I need some help crafting a message that I'm about to send" have helped. They found that asking, "Can you give me some creative ideas?" was a big help in getting past their writing obstacles. There is a significant distinction between task 6 and task 7 in terms of time efficiency. In task 6, users were required to send a message, whereas, in task 7, they were specifically asked to send a message using AL (rewrite prompt), resulting in a faster message delivery process. The interface was well received by the users, who found it to be simple to use and aesthetically pleasing because of the clever use of color. They also valued the application's clear wording, which improved usability.

The usability test results indicate that the authentication procedure might be enhanced. Due to time restrictions, the committee has discussed the potential of changing the process' current requirement for a phone number to one that only requires email addresses in the future. In addition, it has been suggested that the next updates include a choice for users to upload photos to the profile screen.

7.2 Software Quality Management

In this section, we will explore software quality management which is a critical aspect of software development. It has been used to ensure that we deliver a high-quality software product. For this purpose, we have used a code analysis tool called CodeScene.

7.2.1 CodeScene

As mentioned before, the tool we used to ensure the quality of our code is CodeScene. It provides several features, such as Hotspot Analysis and it generates visualizations that help us understand the evolution and complexity of our code.

The figure below shows a visual representation of the different parts of the code and their health ratings. The analysis was deemed to be excellent, as indicated by the high code health score.

According to the figure, the codebase, we have developed, consists of 2.791 lines of code and an average of 9.32 / 10 code health. This refers to the overall quality, maintainability, and robustness of the code. Code health includes several elements that determine the code's readability, performance, scalability, and reliability. It includes several elements that affect the readability, performance, scalability, and dependability of the code. However, the figure also shows that the worst performer out of all the files is a back-end-related file, which is an auto-generated file. The reasons behind this can be due to their complexity and prioritization of functionality over readability. However, we are generally satisfied with the extraordinarily good code of health.

Testing

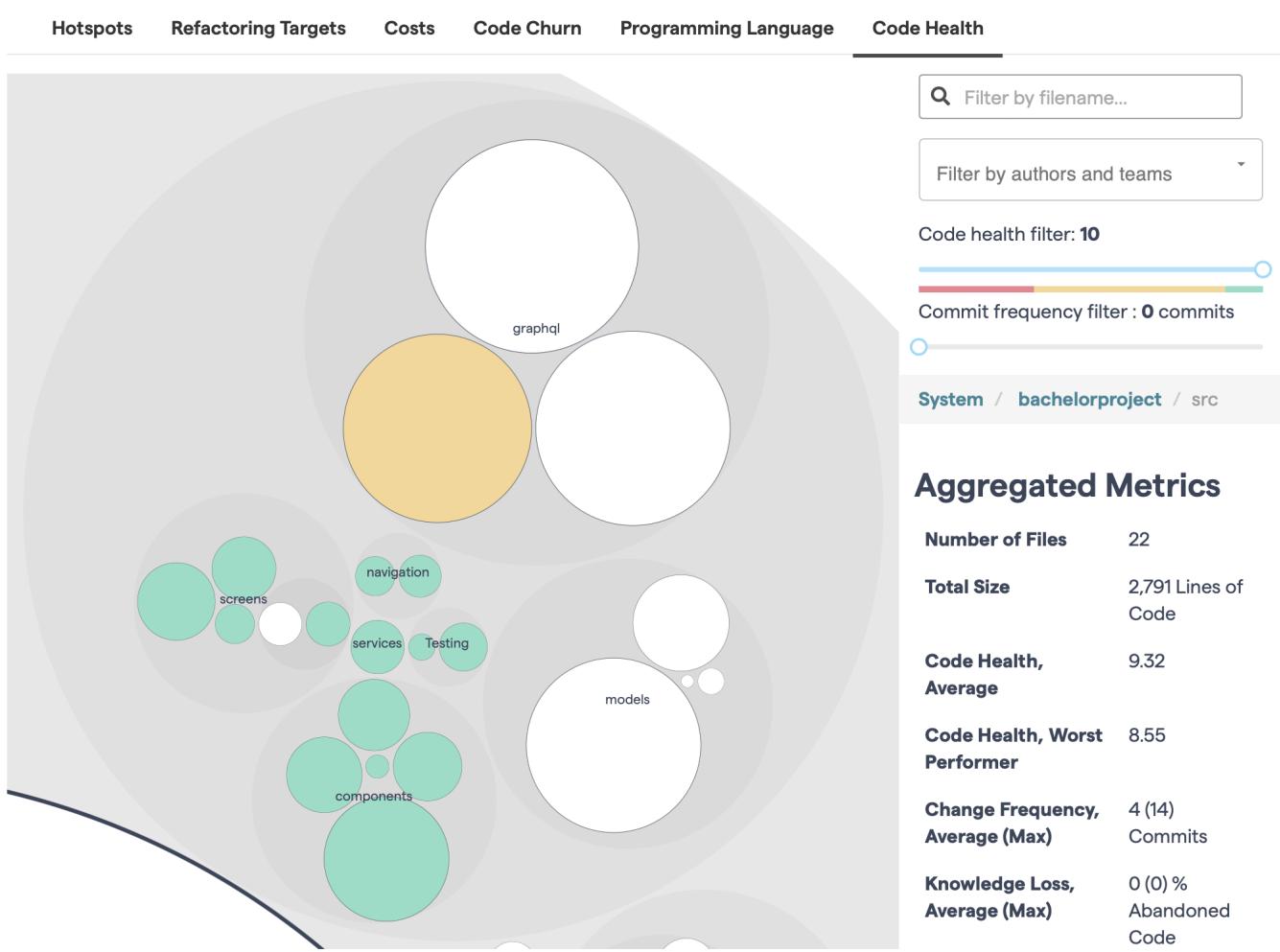


Figure 7.1: CodeScene result of code health

7.3 Snapshot testing

Snapshot testing is a popular software development testing technique that allows developers to test their code fast and efficiently. It entails collecting the output of a function or component, referred to as a "snapshot," and comparing it to a previously saved snapshot. This technique is especially effective for testing UI components since it helps developers to check that the UI remains consistent between codebase iterations.

The goal of a snapshot test is to generate a textual representation of the displayed component. Because the written form is human-readable, we can easily grasp it and hunt for any flaws. This may be used to validate the code and test its behavior. Aside from that, we may utilize snapshots to compare renderings from different project iterations. Regrettably, we were able to test only one of the several screens in our program. This was mainly due to framework issues that hindered us from conducting comprehensive testing across all aspects of the application. We used the Jest framework to conduct snapshot testing. However, before to the commencement of the test case creation process, we needed to configure the framework.

Once we have done with configuration we can begin creating our test cases. As previously stated, we were unable to test all of our screens or components since they contained components that were incompatible with various frameworks. The test case below checks if the HeaderBox component renders correctly by creating an instance of the component using renderer and storing it in the 'component' variable. Then, the expect function is used with the 'toMatchSnapshot' method to compare the 'component' variable with previously saved snashot of the component. The 'toMatchSnapshot' method checks if the component output matches the stored snapshot. If there are any differences between the component and the snapshot, the test fails.

```

1 /import React from 'react';
2 import renderer from 'react-test-renderer';
3 import HeaderBox from '../components/HeaderBox';
4
5 describe('HeaderBox component', () => {
6   test('renders correctly', () => {
7     const component = renderer.create(<HeaderBox />);
8     expect(component).toMatchSnapshot();
9   });
10 });

```

Listing 7.1: Snippet of Jest Test Cases

The results of the test case for the HeaderBox component can be seen in the figure below 7.2. The test has passed and ensure that the component renders correctly.

```

● PS C:\Users\hijab\OneDrive\Dokumenter\Uni (software)\P6\bachelorproject-1> npm run test
  > bachelorproject@1.0.0 test
    > jest

    PASS  src/Testing/HeaderBox.test.js
      HeaderBox component
        ✓ renders correctly (39 ms)

    Test Suites: 1 passed, 1 total
    Tests:       1 passed, 1 total
    Snapshots:  1 passed, 1 total
    Time:        1.228 s, estimated 2 s
    Ran all test suites.

○ PS C:\Users\hijab\OneDrive\Dokumenter\Uni (software)\P6\bachelorproject-1>

```

Figure 7.2: Snapshot test getting passed

This verified that the output remains consistent and aligns with the expected outcome. And it provides confidence that the code is functioning as intended.

7.3.1 Integration testing

Integration testing is a kind of software testing that checks the functionality and interactions of several software modules or components within an application. To make sure that the software operates properly as a whole, it focuses on how different parts integrate and cooperate. This is particularly relevant for applications like ours that involve real-time messaging and require seamless communication between the back-end and front-end. Since our application relies on real-time messaging, where instant communication between users is essential. Integration tests help validate that the messaging functionality, including sending and receiving messages, works properly and that data is transmitted accurately and on time.

Integration testing of GraphQL APIs entails evaluating how the GraphQL API interacts and functions within the larger application environment. It focuses on verifying that the API correctly handles and processes GraphQL queries, mutations, and subscriptions and that it integrates effectively with the underlying data sources, external services, and other components of the application. By testing GraphQL APIs, we can ensure the correctness, reliability, and performance of their API implementations.

Here is an example of integration test cases we could use to test the application: By running the integration test, where it gets verify that ContactsScreen component interacts correctly with the API.

```

1  /
2  import React from "react";
3  import { render } from "@testing-library/react-native";
4  import ContactsScreen from "./ContactsScreen";
5  import { API, graphqlOperation } from "aws-amplify";
6  import { listUsers } from "../graphql/queries";
7
8  jest.mock("aws-amplify");
9
10 test("fetches users data from API", async () => {
11   API.graphql.mockImplementationOnce(() =>
12     Promise.resolve({
13       data: {
14         listUsers: {
15           items: [{ id: 1, name: "John Doe" }],
16         },
17       },
18     })
19   );
20   const { findByText } = render(<ContactsScreen />);
21   const user = await findByText("John Doe");
22   expect(user).toBeDefined();
23   expect(API.graphql).toHaveBeenCalledWith(graphqlOperation(listUsers));
24 });
25
26 test("renders contacts screen", () => {
27   render(<ContactsScreen />);
28 });
29
30 test("renders flat list with users data", () => {
31   const { getByTestId } = render(<ContactsScreen />);
32   const flatList = getByTestId("flatList");
33   expect(flatList.props.data).toEqual([]);
34   expect(flatList.props.renderItem).toBeDefined();
35 });

```

Listing 7.2: Integration testing using Jest framework

An integration test is a lower level than unit tests. Therefore we faced some issues transitioning from unit test to integration test and that led to an increase in the complexity of the code. Some of the issues were according to connect the test directly to our database since we didn't want the tests to interfere with the real data in the database. the reason behind this is that we have learned from previous projects that testing with an actual database over the network just leads to slow tests.

7.3.2 End-to-end testing

End-to-end testing is a useful testing approach that we may use to evaluate the efficacy of our software application. This thorough method examines the application's functionality, performance, and dependability throughout its whole lifespan. End-to-end testing enables us to verify the interactions between the various systems and components inside the application by replicating actual user situations.

End-to-end tests may be carried out using a reliable and flexible testing framework like Cypress. It provides us with a variety of tools and functionality to automate user interactions, make assertions, and check that our application behaves as intended.

We regretfully were unable to do end-to-end testing for our project owing to scheduling restrictions. However, we recognize its importance and think it's necessary for a complete assessment. We want to emphasize the advantages of end-to-end testing implementation in our final report and suggest its adoption as a recommended practice for further testing attempts. We will discuss the benefits end-to-end testing provides to our software development process as well as the potential processes and materials needed to carry them out successfully.

This Cypress test suite focuses on the "Home Screen" functionality. It begins by setting up a timeout of 10 seconds to ensure sufficient time for the chat rooms to load. The `beforeEach` hook ensures that the home screen is visited before each test case.

The first test, "displays a list of chat rooms," verifies that the chat room list is rendered on the screen. It checks for the existence of the chat room list element with the specified `data-testid` attribute. Additionally, it ensures that there is at least one chat room displayed by asserting the length of the chat room elements.

The second test, "opens a chat room when clicked," simulates a user clicking on the first chat room in the list. It checks that the URL is updated to include "/chatRoom," indicating a successful navigation to the chat room screen.

```

1 /describe("Home Screen", () => {
2   beforeEach(() => {
3     // Increase the default timeout from 4 seconds to 10 seconds
4     // to allow enough time for the chat rooms to load
5     Cypress.config("defaultCommandTimeout", 10000);
6
7     // Visit the home screen before each test
8     cy.visit("http://localhost:19006/");
9   });
10
11  it("displays a list of chat rooms", () => {
12    // Check that the chat room list is displayed
13    cy.get("[data-testid='chatRoomList']")
14      .should("exist")
15      .and("have.attr", "data", "chatRoomList");
16
17    // Check that at least one chat room is displayed
18    cy.get("[data-testid='chatRoom']")
19      .should("have.length.gt", 0);
20  });
21
22  it("opens a chat room when clicked", () => {
23    // Click on the first chat room in the list
24    cy.get("[data-testid='chatRoom']").first().click();
25
26    // Check that the chat room screen is displayed
27    cy.url().should("include", "/chatRoom");
28  });
29});
```

This Cypress test suite contributes to the overall quality assurance of the application by validating the expected behavior and user interactions on the home screen.

8 | Discussion

8.1 Discussion

Throughout the project, it became evident that several issues had an impact on the level of frustration and the development process. A general discussion on the development process will come next, and then a segment about the application's future development will be presented.

During the implementation part, the group went through some issues with phone number authentication (OPT). One of the most important parts of the application was the authentication procedure. The first need was to use AWS Amplify to implement phone number authentication. Due to time constraints, the group chose email authentication instead. For email authentication, it has been used a predefined user interface component from AWS amplifies React native library. By opting for email authentication, we streamlined the implementation process and focused on other critical functionalities of the application. Additionally, email authentication offers a more user-friendly experience, as it uses a communication channel that is commonly used.

The Comparison Chart between Facebook, Whatsapp and Elderconnect can be see in figure 8.1.

FUNCTIONALITIES	FACEBOOK	WHATSAPP	ELDERCONNECT	NATURAL READER	VOICE DREAM READER
Speech-to-text	✗	✗	✗	✗	✗
Messaging	✓	✓	✓	✗	✗
Text-to-speech	✗	✗	✓	✓	✓
ChatGPT API	✗	✗	✓	✗	✗

Figure 8.1: Comparison Chart

Our application, ElderConnect, provides a user-friendly and technologically advanced chat platform. For a more detailed overview of other chat applications, such as Facebook and Whatsapp can be found in the sections 2.1.5.2 and 2.1.5.1

The comparison chart 8.1, shows the functionalities and the exiting solutions from section 2.1.5 were evaluated alongside our application. While existing solutions, such as Facebook and Whatsapp do not offer any of the technologies, only messaging. NaturalReader 2.1.5.4 and Voice Dream Reader 2.1.5.5 both utilize cutting-edge Text-to-Speech technology. However, it's worth mentioning that these applications are not intended for chat-based conversations. our application goes beyond that by incorporating both text-to-speech (T2S) functionality and the advanced ChatGPT API. By leveraging these newer technologies, our application stands out as superior, providing a more comprehensive and innovative communication experience

8.2 Future development

In this section, we will provide an overview of the key areas addressed in the ElderConnect application. The focus will be on the following aspects:

- OPT authentication
- Speech-to-text
- Testing
- Possible upcoming features

OPT authentication

As mentioned earlier in the report as well as in our current version of the application, we have chosen email authentication as the login mechanism, due to challenges faced during the implementation of phone number authentication. However, we recognize the need of including phone number authentication in our application's future development. Implementing authentication with a phone number in AWS Amplify may incur costs. As a result, the authentication should change to to OPT authentication in the future.

Speech-to-text

Speech-to-text has not been implemented in the application, since react native does not provide built-in support for speech-to-text functionality. To implement the speech-to-text functionality in the application by utilizing third-party libraries. Due to time limitations, we were unable to work further resources to work on implementing speech-to-text functionality. However, it is essential to prioritize this feature as the first item in future development endeavors.

Testing

The group performed several forms of testing, as mentioned earlier in the report. since because of the short time the group could not perform end-to-end testing.

The group performed various forms of testing, as mentioned earlier in the report, to ensure the stability and functionality of the application. However, the group was unable to perform end-to-end testing because of time restrictions. Therefore, it is recommended that end-to-end testing be included in the future development cycle to ensure comprehensive quality assurance.

Possible upcoming features

An idea for upcoming possible and useful features for the application:

Adding groups option In future development, a group feature would be added to the application. Users will be able to create and join groups, facilitating collaborative discussions and connections with multiple people. So users can participate in group conversations, share ideas and make other plans together.

Enhancing profile pictures Another possible feature that could be implemented is to make enhance the user profile functionality. Integration of Profile Pictures: The application will be improved to allow users to manage and upload their Profile Pictures. With the help of this function, users will have a more personalized and visually appealing experience.

Enhancing user name Enhancing the user profile functionality is a further possible feature that might be made. The user's name will be made dynamic, allowing users to update their name within the application. This change will ensure that users can easily maintain and reflect their updated information.

9 | Conclusion

9.1 Conclusion

In conclusion, this project focused on addressing the challenges faced by elderly individuals in communicating with their family and friends due to difficulties related to reading, writing, interfaces, and authentication. Recognizing the importance of digital communication for elderly people and understanding the needs of the target demographic, our objective was to develop a messaging application that leverages modern technologies, specifically Text-to-Speech and the ChatGPT API, to simplify and enhance communication for older individuals.

Through a comprehensive problem analysis, we identified the specific barriers faced by elderly users in digital communication and explored existing solutions available in the market. To gain deeper insights and validate our findings, we conducted interviews with elderly individuals to understand their firsthand experiences and preferences.

Utilizing a well-defined methodology, we selected appropriate technologies and designed an application that catered to the unique requirements of the elderly. The implementation phase involved carefully integrating Text-to-Speech functionality and incorporating the capabilities of the ChatGPT API to provide a user-friendly and accessible messaging experience.

To ensure the quality and reliability of the application, rigorous testing was conducted, encompassing functionality, usability, and performance. This step allowed us to identify and address any potential issues or shortcomings, leading to an improved final product.

The discussion section of this project provided an opportunity to reflect on the outcomes and implications of our work. We highlighted the positive impact that our application can have on elderly individuals, enabling them to communicate more effectively and feel connected to their loved ones. Additionally, we explored future development possibilities, including potential enhancements and features that could further improve the user experience.

Before we conclude it's important to acknowledge and discuss the limitations of the application because understanding these limitations helps us with setting realistic expectations. some of the potential limitations of our application can be:

- Connectivity dependence: since some features of our app require an internet connection to function optimally, a poor internet connection may restrict the use of these features.
- Device Specifications: since we used some technology in devices launched in 2018, older devices may experience compatibility issues, which affects the overall user experience.
- Language: since our target group is primarily elderly Danes individuals, maybe we will face some issues with the used language and how this Technologies will deal with the

In conclusion, by harnessing the power of modern technologies such as Text-to-Speech and the ChatGPT API, we have successfully developed a messaging application that addresses the communication challenges faced by elderly individuals. This project not only contributes to improving the digital communication experience for older users but also promotes inclusivity and connectivity within the aging population. Moving forward, continued efforts to refine and expand the application will help us maximize its potential and ensure that elderly individuals can enjoy the benefits of digital communication in a more accessible and intuitive manner.

Bibliography

1. S. Kemper, (<https://www.ncbi.nlm.nih.gov/books/NBK97337/>) (2020).
2. C. for fisease control, precention, (<https://www.cdc.gov/aging/publications/features/lonely-older-adults.html>).
3. A. Kaniel, (<https://www.comfortkeepers.com/articles/info-center/senior-health-and-wellbeing/communicating-from-a-distance-seniors-and-technology/>) (2017).
4. P. HolBøll, (<https://mobilsiden.dk/nyheder/aeldre-kan-ikke-finde-ud-af-at-smse/>) (2010).
5. Artera, (<https://artera.io/blog/seniors-texting/>) (2022).
6. K. M. Yorkston, (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074568/>) (2011).
7. A. Vagh, (<https://dev.to/avinashvagh/understanding-the-concept-of-natural-language-processing-nlp-and-prompt-engineering-35hg>) (2023).
8. S. RAMLOCHAN, (<https://www.promptengineering.org/what-is-prompt-engineering/>).
9. openai, (<https://openai.com/research/overview>).
10. understood, (<https://www.understood.org/en/articles/text-to-speech-technology-what-it-is-and-how-it-works>).
11. Mark, (<https://www.understood.org/en/articles/text-to-speech-technology-what-it-is-and-how-it-works>) (2017).