

The Effects of Threads in a Chat Application and How to Design Them

Arvid Bengtsson and Filip Tran

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2023

MASTER THESIS



The Effects of Threads in a Chat Application and How to Design Them

Arvid Bengtsson
ar5780be-s@student.lu.se

Filip Tran
1e5116tr-s@student.lu.se

22 september 2023

Master's thesis work carried out at Telavox AB.

Supervisors: Günter Alce, gunter.alce@design.lth.se
Niklas Bruce, niklas.bruce@telavox.com
Elias Vernersson, elias.vernersson@telavox.se

Examiner: Joakim Eriksson, joakim.eriksson@design.lth.se

The Impact of Threads on a Chat Messaging Application

Copyright ©2023 Arvid Bengtsson, Filip Tran

Published by

Department Design Sciences
Faculty of Engineering LTH, Lund University
P.O Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design MAMM01
Division: Ergonomics and Aerosol Technology
Supervisor: Günter Alce
Examiner: Joakim Eriksson

Abstract

The usage of online communication has increased heavily in recent times, especially chat applications. It both makes communication faster and fits well with remote work. As the chats of large chat groups can be messy with many writers there is potential for improvement in regard to the structure and organization of messages. This is the focus of this master thesis, what kind of effects using threads may have, and how to design the implementation. The research questions that were posed were the following: 1) What effects do chat threads have on text communication? 2) How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability? 3) How do different communication systems affect the efficiency of the teams/people using them?

These were answered by a user-centered and iterative process, during which multiple methods were used to gain both qualitative and quantitative data. Such as heuristic evaluation, user tests, interviews, and surveys. This showed that threads would be appreciated by users in larger chat groups and what to keep in mind while designing this feature. The iterative process also showed which aspects of the design of threads needed to be improved while also displaying which aspects worked. Furthermore, the results indicate that if implemented correctly, with the user in mind, threads would be intuitive to learn and use. The text communication seemed to be affected in a positive manner by threads as it organized and structured messages which allowed for countermeasures against message obscuration. Lastly, the structure and organization allowed for more efficient work as it encouraged more public conversations and eased finding older and relevant messages.

Keywords: Chat, Chat application, Chat threads, Usability, User-centered design

Sammanfattning

På senare tid har användningen av online kommunikation ökat drastiskt, speciellt angående chatapplikationer. Det gör både kommunikation snabbare och passar in väl med distansarbete. Då chatter i stora grupper kan bli röriga då det är många användare som skriver finns det potential för förbättring gällande strukturen och organiseringen av meddelandena. Det är målet med detta examensarbete att hitta vilken effekt användandet av trådar kan ha och hur implementation av dem ska designas. Frågeställningen blev följande: 1) Vilka effekter har chattrådar på textkommunikation? 2) Hur kan man effektivt designa/implementera trådar för att göra det tillfredsställande enligt ISO 9241-11:2018 angående användbarhet? 3) Hur påverkar olika kommunikationssystem effektiviteten hos teams/människor som använder dem?

Detta blev besvarat av en användarcentrerad- och iterativ process där flera metoder användes för att få fram både kvalitativ och kvantitativ data. Metoder som heuristisk utvärdering, användartester, intervjuer, och formulär. Detta visade att trådar hade varit uppskattat av användare i större chattgrupper och vad man ska hålla i åtanke när man designar denna funktion. Den iterativa processen visade också vilka aspekter av tråddesignen som behövde förbättras och vilka aspekter som fungerade. Resultaten indikerar att om trådar implementeras korrekt och med användaren i åtanke, kommer trådar att vara intuitivt att lära sig och använda. Textkommunikationen verkar bli positiv påverkad av trådar då det organiserar och strukturerar meddelanden vilket förebygger obskuration av meddelanden. Slutligen tillåter strukturen och organiseringen mer effektivt arbete då det uppmuntrade fler publika konversationer och underlättade att hitta gamla och relevanta meddelanden.

Nyckelord: Chatt, Chattapplikation, Chattrådar, Användbarhet, Användarcentrerad design

Acknowledgements

We would like to thank Günter Alce from the Department of Design at LTH for being our supervisor during this thesis and giving us valuable guidance, feedback, and input. We would also like to thank Elias Vernersson and Niklas Bruce for being our supervisors at Telavox for welcoming us to their company and helping us through the administrative work to conduct our thesis at their company. Furthermore, they aided us in providing all the necessary assets and resources for us to complete our thesis. Lastly, we would like to thank all the participants who have been part of our usability testing and pilot study for taking their time to help us and giving valuable input and feedback.

Contents

1	Introduction	4
1.1	Background	4
1.1.1	Related work	5
1.1.2	Global Sustainability Goals	5
1.1.3	Limitations	5
1.1.4	Purpose and Goals	6
1.1.5	Research Questions	6
2	Methodology	7
2.1	User centered design	7
2.2	Norman's Design Principles	7
2.3	Usability	9
2.3.1	Heuristic Design Evaluation	9
2.4	Data Gathering	10
2.4.1	Interview	10
2.4.2	Questionnaires	11
2.4.3	Quantitative Data	12
2.4.4	Qualitative Data	12
2.4.5	Affinity Diagram	13
2.4.6	Brainstorming	13
2.5	Prototype	13
2.6	Testing	14
2.6.1	Test Plan	14
2.6.2	Comparative Analysis	15
2.7	Programs and Software	15
2.7.1	Miro	15
2.7.2	Figma	15
2.7.3	Sketch	16
2.7.4	Telavox Chat Application	16
2.7.5	Google Meet	17

3	Iteration 1: Prototyping	18
3.1	Pilot Study	18
3.1.1	Heuristic Design Evaluation	18
3.1.2	Purpose and plan of action	19
3.1.3	Results	20
3.2	Prototype	20
3.2.1	Idea generating process	20
3.3	Prototype Testing	26
3.4	Results	30
3.5	Result Analysis	35
4	Iteration 2: Refinement	38
4.1	Idea generating process	38
4.2	Development	39
4.2.1	Prototype 1 - Inline	40
4.2.2	Prototype 2 - Segregated	41
4.3	Prototype Testing	42
4.4	Results	42
4.5	Result Analysis	44
5	Discussion	49
5.1	General	49
5.1.1	Process	49
5.1.2	Embracing change	50
5.2	Testing	50
5.2.1	Usability test	50
5.2.2	Integrity of the test	51
5.2.3	Subjective quantifiable data	51
5.2.4	Sample Sizes	52
5.2.5	Finding participants	53
5.3	Research Questions	53
5.4	Future Work	55
6	Conclusion	57
	Bibliography	58
	Appendix A Questionnaires and Questions	61
	Appendix B Affinity Diagrams	67
	Appendix C Prototypes	79

Chapter 1

Introduction

1.1 Background

Chat applications are becoming more and more prevalent within the workplace [15]. Compared to most alternatives they can be found to be fast and less formal, allowing quick conversations between colleagues to exchange information or lending services [8]. These chats can connect two individuals and even entire teams in the form of a chat room or group chat allowing them to reach all individuals quickly. While this can be very beneficial it is not without issues or challenges. Problems can arise if multiple conversations regarding different topics are held in the same space [20]. If these conversations are held at the same time, it would be more difficult to follow them all or if someone did not stay tuned in while the different conversations were taking place, going back to read and sift through previous messages might be tedious and ineffective. Especially if there is a need to find a discussion that was held a long time ago there might need for some intensive scrolling and searching.

We wish to explore if threads are a possible solution to this. Graphically tying messages regarding a certain topic to the root message may increase readability [23]. This could in turn increase the productivity of the team, as the members can go to a specific thread to read or say what they need. If a topic is irrelevant to them for the time being threads could also help them by minimizing the distraction this topic would cause.

Telavox

Telavox, a telecommunications software company that offers their services to improve communication within companies and their external communication, offered their help by providing workstations, tools, and mentors [25]. Even the application that will act as the foundation for our research is their internal chat application used by their workforce. The chat application is one of their main ways of communication through text and is developed and maintained internally. Due to this reason, the purpose and work of this thesis as well as

Telavox's own interest in developing their internal chat app made the collaboration possible.

1.1.1 Related work

The work done and the article written by Gilbertsson and Malmström [5] were taken into account during this project. There are many similarities between their work and ours, being conducted and collaborated with the same company, Telavox, for starters. Both projects had the purpose of researching how digital communication is affected by different tools or functions provided by applications and used the Telavox chat application as a foundation. In the course of their research they went through three iterations, during each they developed more and more advanced prototypes. Each prototype was developed by forms of idea generation, implemented, and then tested to gain data and insight to take away for future iterations or work. Their findings were that features that improve the organization and structure of messages, namely threads, in an application could help with collaboration between users. While we took their work into consideration we tried to stay independent to not just replicate what they did. This is because if we came to similar conclusions on our own the validity of both their and our claims would be strengthened.

Another article that explored a similar to ours, i.e. threads, is Shen's article written in 2010[23]. It presents his research on how instant messaging (IM) tools are used in the corporate sector can be improved. It describes how IM provides a free and casual form of communication that can cause confusion among individuals, which in turn means it is unsuited for corporate environments. Roddy and Epelman-Wang also bring up the issues that IMs have when they scale up with more users [20]. They mention in their article that chat rooms start to break down as more users are introduced and more conversations are held concurrently, making it hard to separate between conversations. The solution proposed by Shen is to have a topic-centric thread-based way of organizing the chat. This led to the development of a prototype and preliminary usability testing which showed that it was capable and effective in supporting simultaneous group conversations.

1.1.2 Global Sustainability Goals

The aim of this thesis is to strive towards the eighth goal of the 2030 Agenda for Sustainable Development, which entails "promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all" [19]. It does this by researching the impact that organization has on text communication as it could encourage more effective work and as a result cause a higher degree of productiveness. Furthermore, the impact of organizing text messages would hopefully encourage other chat messaging apps to implement similar features and create a competitive market that promotes economic growth.

1.1.3 Limitations

This project will span over 20 weeks, and the limit of time will bring some direct limitations and some limitations that are chosen to fit the schedule. The development of the system is directly limited as the time is considered to allow for two iterations. Going from a lo-fi prototype to a mid/hi-fi prototype and finishing with an implementation of the system.

Further limitations that are brought on by the time constraint is that the research will be done in regard to the web application. The application version used for phones will be disregarded as the platform is different and would most likely require a similar project of its own.

Another limitation as a result of the time constraint is that no A/B testing will be applied in the second iteration that will be previously used in the first iteration. This is mainly due to the nature of numerous tests that will be conducted as well as the implementation of the feature in the second iteration. Thus, only one prototype will be implemented in the second iteration. In terms of testing the results of the increase or decrease of effectiveness in teams, it is difficult to accurately quantify because of the limitation that it is not fully tested in an environment where many users use the feature of threads. It will instead be limited to a smaller sample size of the overall users by testing them on scenarios that were created beforehand.

1.1.4 Purpose and Goals

The overall goal and purpose is to investigate the effects that threads can have on communication. To do this, the thesis will investigate the desired features that users wish to have and then create prototypes that introduce said desired features. With the information gathered from users testing the prototypes, we will investigate the effects threads have on text communication, their impact on group communication, and how to design threads to make them user-friendly.

Goals

- Develop and implement a chat thread functionality in a chat app.
- Evaluate design of chat threads and their impact on communication, with a user-centered design process.

1.1.5 Research Questions

- What effects do chat threads have on text communication?
 - What kind of redundancy do threads introduce in text communication?
- How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability [9]?
 - How do you design/implement threads conveniently so users would benefit from making and using threads?
 - How should the GUI or layout be designed to smoothly navigate between different threads and chat rooms?
- How do different communication systems affect the efficiency of the teams/people using them?
 - What constitutes effective communication?

Chapter 2

Methodology

The methodology chapter aims to give insight into the processes behind the methods that were used during the thesis. It specifically covers the user-centered process of data gathering and analysis as well as the implementation of the prototypes.

2.1 User centered design

In order to research and find possible answers to the research questions, the process will use a user-centered design approach. Meaning solutions to given problems will have the user as a focal point to find and design the possible solutions [17]. In order to achieve this, users will be involved early on in the process when gathering data to find solutions on how to implement a feature and be a part of testing design choices that were created.

The goal of this thesis, as a result of the user-centered design, will make use of an iterative process where different stages of development will be done several times in order to find solutions and further make the user included in the process. This process can be seen in figure 2.1. The first step in the process, which will not be repeated, is the pilot study, where data gathering if chat threads are desired or not. The idea-generating process is the next step which is about coming up and creating solutions to requirements. The prototyping is the third step in the process that implements the solutions that were generated which are then tested by users in order to evaluate the prototype. When all of this has been completed, the process is repeated from the idea-generating process.

2.2 Norman's Design Principles

Norman in his book "The Design of Everyday Things: Revised and Expanded Edition" introduced the fundamental principles of interaction design and with it were seven principles

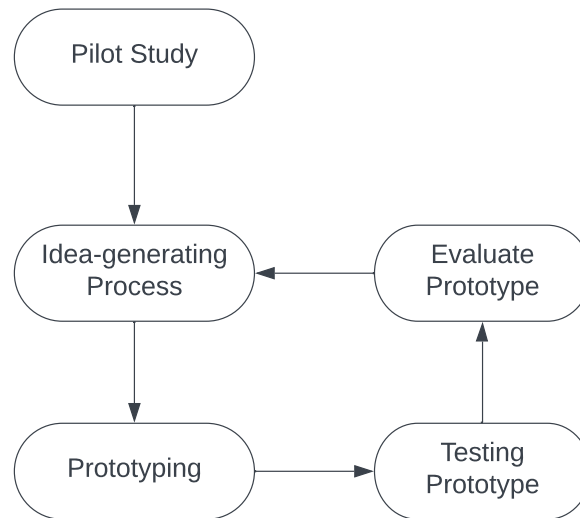


Figure 2.1: The iterative user-centered design process.

of design [14]. The seven principles mentioned by Norman are the following: affordance, signifier, feedback, constraint, mapping, discoverability, and conceptual model.

- **Affordance** is a term that describes the relationship between an interactive object and a person. In essence, it is about how the interactive object is able to convey to a person possible ways to interact with it. When an object is said to have great affordance is when a person is able to determine how to use it instantly.
- **Signifiers** could be confused with affordance, but what signifiers describe is where the action takes place rather than what actions are possible with an object. An example of a signifier is a sign on a door that says "push".
- **Mapping** is the relationship between two sets of things and is important to consider when designing layouts for controls and displays. An example of mapping could be when turning the steering wheel of a car clockwise, it moves the car to the right, and turning the steering wheel counter-clockwise, would move the car to the left.
- **Feedback** is when an object communicates the result of an action. An example of this would be when a person turns on the TV when they press the on button on their remote controller and the TV displays something on its otherwise black screen.
- **Conceptual model** is what Norman describes as "an explanation, usually highly simplified of how something works" where for example a folder on a computer is an explanation of how documents or files could be stored within it.
- **Constraint** is when restrictions are made in order to guide the interaction to make it easier to interpret. The restrictions or constraints can be done through different methods such as creating physical, logical, semantic, or cultural restrictions.

- **Discoverability** is when affordances, constraints, feedback, signifiers, and mapping are properly applied and allows the user to understand how the product works, what it does, and what is possible to do with it. In essence, discoverability is if the user can determine the current state of the device and what actions are possible.

2.3 Usability

Usability is described by Rubin and Chisnell as when a product is truly usable: "the user can do what he or she wants to do the way he or she expects to be able to do it, without hindrance, hesitation, or questions"[21]. Ways to increase usability are, for example, during development, use the 10 heuristic design principles while designing the product.

2.3.1 Heuristic Design Evaluation

Jakob Nielsen proposes, in his article "10 usability heuristics for user interface design" [13], 10 heuristic design principles called heuristics. These can be used as a rule of thumb while designing a UI and were taken into consideration during this project.

- **Visibility of system status** refers to how well the system gives feedback and shows the user what is happening. This could be highlighting, pop-ups, descriptive texts, or things that help the user understand their situation.
- **Match between system and the real world**, this is how well the UI is designed for the user. Making it more familiar for them by using concepts, icons, and words that they used to. Doing this and having information shown in a natural and logical order makes the system more intuitive and the learning process easier.
- **User control and freedom**, is about helping the user feel free and in control of the system and not vice versa by making it easy to rectify mistakes. Accidentally pressing the wrong button could mean getting sent to a new window could easily happen, it should be easy to go back again. Having clearly coded signifiers for affordances regarding exit or back actions is necessary to improve upon this.
- **Consistency and standards** states that the designer should follow industry standards and conventions while being consistent in the design. Having the icon mean different things in different windows will make the system unnecessarily complicated to learn. Being consistent and using concepts conventionally used in the industry may decrease how much a new user needs to learn before being able to use a system.
- **Error prevention**, in addition to giving the user easy ways to get back after being sent to a new window after accidentally clicking a button, the design can help prevent these mistakes. Asking for confirmation or removing options at certain moments can help the user avoid making mistakes. This could be a warning message when the user is about to close a text editor with unsaved data.
- **Recognition rather than recall**, the user should be able to recognize what they should do to achieve their goal, rather than remember. Making the information that is shown

understandable through intuition will decrease the amount that users will have to remember. This will make it easier to learn how to use the system.

- **Flexibility and efficiency of use**, after the users have become more experienced with the UI their work could be sped up through shortcuts and customization. This should be hidden in some way and be optional to use when ready. Having these kinds of options makes the work for both new and frequent users.
- **Aesthetic and minimalist design**, this describes how the UI should not have unnecessary items of information. Every piece of graphic in the interface competes for the user's attention. Having a minimalistic design, making the less relevant information less distracting, while making the more relevant information more attractive will help the user know what to focus on.
- **Help users recognize, diagnose, and recover from errors**, if the program encounters an error during runtime or some action the user took failed there should be a message. The message should clearly describe why the error occurred in a way the user understands and provides a solution to the problem.
- **Help and documentation**, even if the program is intuitive and easy to use documentation can still necessary to describe how to do tasks etc. The documentation should be concise and easy to follow, and it should be easy for the user to find what they are looking for.

2.4 Data Gathering

One crucial aspect of this thesis is the method of gathering the data that is needed in order to answer the research questions that were posed. There are numerous methods of doing so, but the ones that were used during the course of this thesis are described below. Furthermore, the data that is gathered from the data gathering methods can also be categorized into two separate types.

2.4.1 Interview

Interviews are one way of gathering data by conversing with the other party and asking questions in order to receive answers to the questions that the interviewer posed to the interviewee. There are several ways of holding an interview in the sense of how the questions are asked to the interviewee/interviewees and how the questions themselves are formulated in order to receive answers in a certain way. The three methods that were used in this thesis were: unstructured, structured, and semi-structured [4]

Unstructured

Unstructured interviews are open-ended in nature because the questions are asked in a way that promotes elaborate answers rather than yes or no answers. Furthermore, due to the nature of open-ended questions in unstructured interviews, probing is also used in order to

steer the user toward the topic [17]. The use of unstructured interviews allows answers to be more in-depth regarding a topic, in this case, threads. This in turn makes the data more rich as it is more complex and gives a better understanding of the topic at hand. The data that is gathered is qualitative data.

Structured

Structured interviews are often the preferable choice when gathering data regarding a particular design feature. This is due to the nature of structured interviews being more closed as the questions are standardized and more specific than the open-ended interviews [17]. The interviews themselves follow a strict script where there are few or none deviations from said script [4]. Thus, structured interviews will be used after a user is testing the prototype of an iteration.

Semi-structured

Semi-structured interviews are a combination of the previously mentioned styles of interviewing [17]. It makes use of the open-ended questions that are prevalent in unstructured interviews to probe the interviewee for data and predetermined questions that are more close-ended in nature. The predetermined questions are there so the interviewees are all asked the same questions and when they do give an answer that would benefit from further clarification, probing is used.

2.4.2 Questionnaires

Questionnaires are another way of collecting data as it is a well-established technique. They are a way to easily reach a larger number of people as it is easier to distribute said survey over the internet instead of conducting interviews in person [17]. This also eliminates the problem that people cannot attend because they are located in separate locations. Questionnaires themselves can be done similarly to interviews in the sense that they could have open-ended questions or close-ended ones. Usually, if there is no need for the moderator or anyone present to answer questions, questionnaires are a good option to use to collect data [17]. As there is no one present to clarify questions, it is important for the questions to be clear of ambiguity and concise.

System Usability Scale

The System Usability Scale (SUS) is a ten-question questionnaire with five different grades in each question that are ranked from strongly disagree to strongly agree. SUS is meant to measure the usability of the subjective experience of a system that has been tested by a participant [1].

SUS has a grading score where the different questions, depending on if the question is odd or even in the order it is presented, will yield different results. It is calculated by the odd-numbered questions (1, 3, 5, 7, 9) being the score of the answer and subtracted with one, for example, if the participant answers four on the first question, the score of the first question will be three as it is calculated through $4-1$. On the other hand, the even-numbered questions

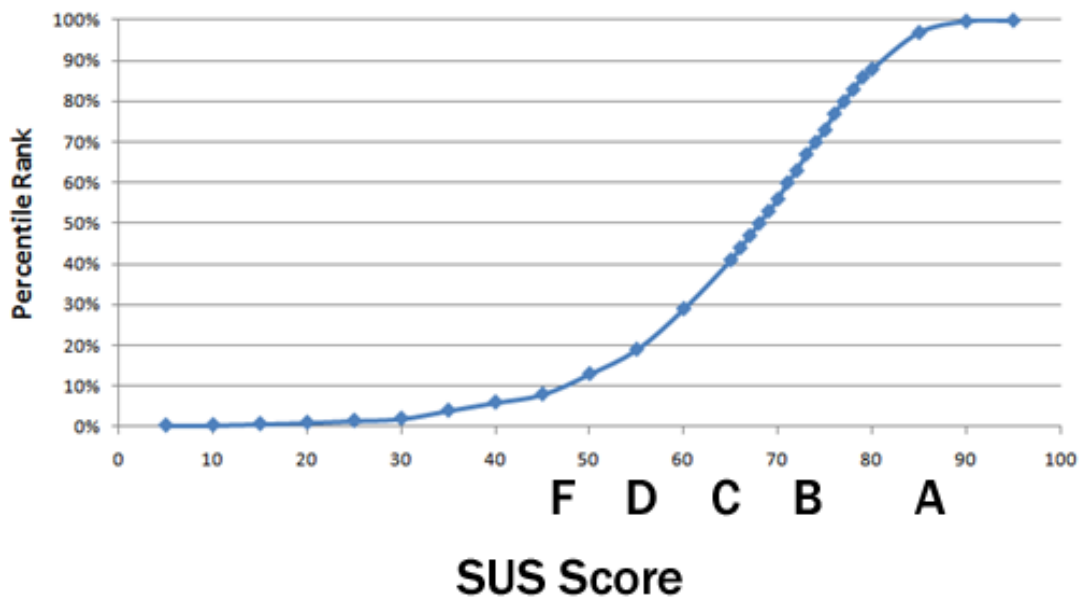


Figure 2.2: A SUS curve depicted by Jeff Sauro on how to interpret the score obtained from SUS [18].

(2, 4, 6, 8, 10) are calculated by subtracting the score they chose in the questionnaire from five, for example, if a participant answers three on question two, the score for that question will be calculated as two due to the calculation being $5-3$. When all the scores from the questions have been tallied, they will then be multiplied by 2.5 in order to achieve a 0 to 100 score [1].

Jeff Sauro mentions in his article *Measuring Usability with the System Usability Scale (SUS)* [18] what a good SUS score would be. He depicts that a score of 68 is average and anything below or above would be classified as above-average or below-average. Furthermore, he ranks the raw SUS scores with an A-F ranking as well as a percentile rank based on normalizing the total score (see figure 2.2).

2.4.3 Quantitative Data

Quantitative data is the information that has been gathered that can be expressed in form of numbers [17]. Some typical examples of quantitative data are for example the age of a participant or how many times they have used a product in the past. It is through these numbers that you can analyze the magnitude, size or amount of the data you have gathered [17]. In terms of example testing the degree the test participants believe a feature is designed, it gives designers numerical data for them to analyze and draw conclusions in order to improve the design of a feature.

2.4.4 Qualitative Data

Different from quantitative data which is in the form of numbers, qualitative data is instead the data that is not expressed numerically (while it is possible to be quantitative, it is not deemed as necessary/useful most of the times) [17]. The data that is seen as qualitative data

could be, but not limited to, the data that is gathered from interviews from the descriptions that the interviewee has said, quotes from interviews and images that the participants have drawn.

2.4.5 Affinity Diagram

An affinity diagram is described by *Andres Lucero* as a versatile tool used for organizing and to get an overview of unstructured and dissimilar data [11]. In the report it is described that there are different versions and variations of affinity diagrams and how to proceed while using them. A common way, and the procedure used during this project, is described in four steps. The *first step* is the creation of notes, for example, sticky notes or some digital tool could be used for this. Each note contains some piece of data, possibly collected during an interview regarding the evaluation of a prototype. In the *second step* these notes are grouped together into clusters based on their relation to each other and overarching topics. When the clusters are formed the *third step* is done as a review of the note clusters created. This is done by seeing if there are clusters that can be merged, are unnecessary at this point or if they can be rearranged after importance. The last and *fourth step* is focusing on documentation of the findings. Picking notes that are the most relevant to ones project and counting notes enables quantification of the data making it possible to distinguish the main findings.

2.4.6 Brainstorming

A quick way to get ideas into the open is brainstorming, there are multiple ways of going about doing this. One is saying the first thing that pops into ones head as a reaction to something. Another is to speak in order, the first person starts of with an idea and the rest builds upon this foundation with their own ideas and additions. Brainstorming is not an exclusive technique to interaction design and Preece et al says in his book "Interaction design: beyond human-computer interaction" that brainstorming is a technique to generate, develop, and refine ideas [17] that is used in many different areas of expertise. In terms of interaction design, brainstorming is mostly used to suggest new and improved ideas and to create alternative designs to an already existing product or design.

2.5 Prototype

During testing, the developer can craft a prototype that will act like a simulation of the final product, meaning testing of the system is possible without the full implementation [17]. This allows the developers to quickly test the viability of a system with lower resource investment compared to testing the full implementation. Testing early can show you what is working, what is not, and what is in need of improvement in the design.

There are different kinds of prototypes in terms of how advanced and how like the final product they are. The two most commonly used groups are Low fidelity (lo-fi) and High fidelity (hi-fi). Low fidelity is very primitive, has no moving parts, and is often made of drawing and paper [21]. Testing with these is done very early in the process as a form of proof of concept and a way to get early feedback [21]. High fidelity prototypes however look like the final product and can offer more functionality and immersive scenarios [17]. A high

fidelity prototype for software can be created with different kinds of programs which allows more interaction for the tester. For example, letting them press buttons, scroll through lists, etc. with the expected outcome actually happening [17].

2.6 Testing

During the development of a system or a product, the developers can let users test their prototypes in order to gain feedback and insights, which can be used for improvement [21]. This can be invaluable as it makes the user's voice heard, and if done early can save a lot of resources.

2.6.1 Test Plan

When there is a need for testing it is useful to have a plan. This will help define and keep track of what resources are needed and what the goals are. In their book, Rubin and Chisnell, describe the 9 sections or parts a test plan is typically consistent of [21]. The following steps in a test plan are listed where the first step will be listed at the top and the last step at the bottom.

- **Purpose, goals, objective of the test** - What you want to achieve or find, an abstract goal
- **Research questions** - According to Rubin and Chisnell is the most important one [21]. This is because the research questions describe the issues or questions that need to be resolved. This focuses the research and shapes the planning, designing, and conducting of a test. For the research question to be effective they should be as precise, accurate, and measurable or observable as possible.
- **Participant characteristics** - This is where the end user is described, who is going to use the system when finished, and in turn who the testers' groups are and how many from each group are necessary.
- **Method (test design)** - This is described as a detailed description of how the research with the participants will be carried out. It should give a clear overview of each part of the test, from when the participant is first arriving until they leave. This makes it easier to understand what kind of resources are needed and for cooperation. A clear test design will allow others to understand what is meant to happen and they can more easily see possible improvements, or even allow them to carry out the test as well.
- **Task list** - Contains the tasks that the participants will perform during the test. When presented to the participants the task will serve as scenarios. These are meant to give details and context enough for the participants to be able to do the tasks with as little intervention from the moderators as possible.
- **Test environment, equipment, and logistics** - A description of the resources needed to conduct the test. It describes the environment that will be simulated and the equipment needed by the participants during the test.

- **Test moderator role** - What they will do both in regular cases and under certain circumstances. This entails describing the scenarios to the tester, what to do if some error occurs, and when to ask follow-up questions.
- **Data to be collected and evaluation measures** - Description of the data that is meant to be collected during the test. One group is the objective part, performance data, which includes measures of participants' behavior, error rate, performance time, etc. The other is the subjective part, preference data, which is the testers' thought process during the tasks, answers to questions, rankings of elements, etc.
- **Report contents and presentation** - This is a summary of the main parts of the test report and how the results are intended to be communicated.

2.6.2 Comparative Analysis

Comparative analysis is defined by Pickvance as "having an explanatory question of why observed similarities and differences between cases exist" and having a "reliance on the collection of data on two or more cases, ideally according to a common framework" [16]. In the case of the first statement, this thesis applies this with the introduction of chat threads in a chat application. The second statement is applied through the testing of the two prototypes that were created during the iterations.

When conducting a comparative study, some methodological considerations should be taken as the quality of the studies is dependent on internal validity (study setting, intervention, participants, measures, analysis, and interpretations) and external validity (other settings). One such factor that could affect the study is the selection or allocation bias [10]. In this thesis, this bias was minimized by having one group start with one prototype, and the other group start with the other and when done, test the other prototype they did not start with.

2.7 Programs and Software

2.7.1 Miro

In essence, Miro is a software tool created by Miro that imitates the use of an actual whiteboard in real life. Miro calls it a "visual collaboration platform" due to the fact that people can use their tool for, but not limited to, brainstorming ideas and creating affinity diagrams as well as allowing people to be involved in the work remotely from home [12]. In their product, they have functions that allow users to create, edit, and move around diagrams and post-it notes that the user has created.

2.7.2 Figma

Figma is a software that was created by the company themselves and is a collaborative tool that allows users to create designs and make them interact each other while other users can access the file at the same time and cause changes to the design in real time [3]. The functions

allow the users to create a prototype of high fidelity as it allows them to manifest the end product without committing development time or development resources in order to test the functionality of said prototype. Lastly, one function that it has that will be useful for developers is the fact that the design can be easily translated so developers can make use of it without them having to guess their way to recreate the same design.

2.7.3 Sketch

Sketch is a Mac software application that is similar to Figma as it allows users to create prototypes due to it being a collaborative tool [24]. It was created by themselves and has similar functions to Figma in which they can also create interactions between different designs, create prototypes, as well as ease the design hand-off to developers.

2.7.4 Telavox Chat Application

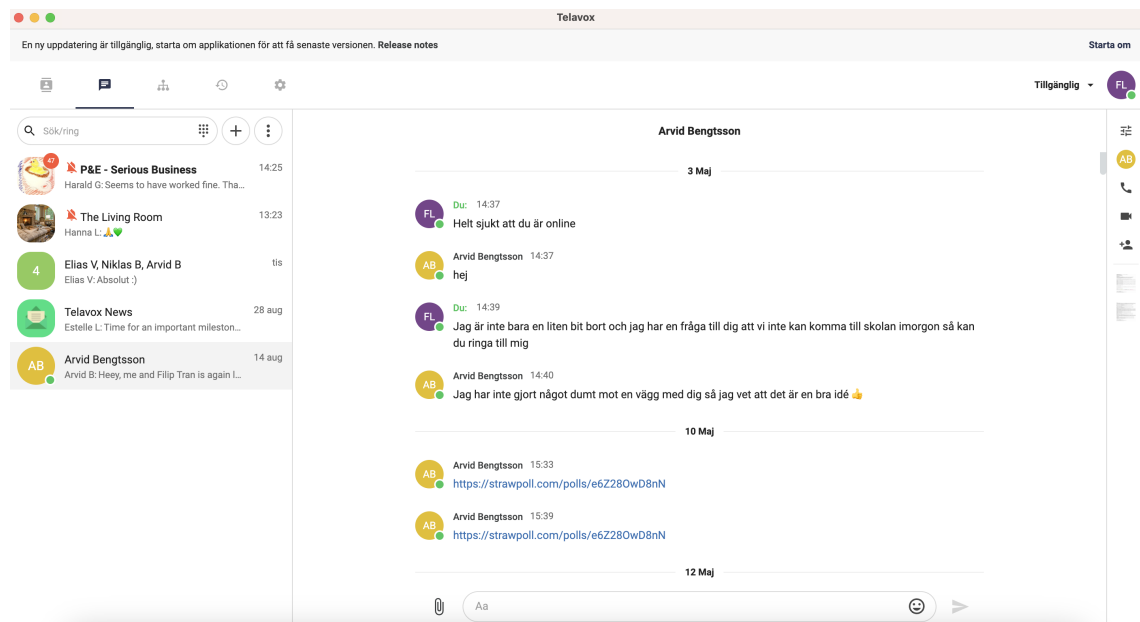


Figure 2.3: The Telavox chat app when messaging another user.

The thesis will be based on the Telavox chat app (see figure 2.3) that the company themselves have developed. One of the most notable parts regarding the app itself is the functionality or lack of functionalities that is usually present in modern chat apps. Functionality that includes reactions, replies, and search functionality are some common features that are missing but compensate for it with the feature of creating a post (that has a reply functionality) as well as a mention feature by using the @ symbol. However, the lack of functionalities presents an opportunity to research the effects that thread would have on a relatively clean slate of a chat app.

2.7.5 Google Meet

Google provides an online meeting service called Google Meet. This can be used to have video meetings[7] and allows the user to share their screen making the other participants able to see what they are showing[6]. It is also possible to record the meetings, including camera and shared screen feeds [7]. This was used for a few interviews during the pilot study but primarily during the testing phase of iteration 2.

Chapter 3

Iteration 1: Prototyping

This chapter aims to explain the results of the first iteration of the thesis from methods that were previously described in the methodology chapter. The iteration was the first of two and focused on what features the user wanted as well as the creation of the first prototype based on the data gathered from interviews.

3.1 Pilot Study

3.1.1 Heuristic Design Evaluation

In the initial phase of the pilot study, there was a heuristic design evaluation of the current Telavox chat application in order to both evaluate potential improvements as well as becoming more familiar with the product itself. The approach was to evaluate the usability of the product concerning messaging and reading messages from other users. Mostly the priority was to find usability issues regarding messages whether it was reading or typing messages and not evaluate the call system for example.

Messages

- When a message is sent by the user themselves, they have no method of interacting with the message after sending it. This issue is a part of error prevention which the application does not fulfill. Furthermore, the messages cannot be edited either if the user for example writes a typing error or wishes to rephrase their message.
- The emoji menu does not have an option to close it once open other than pressing on the same icon again. This limits the user's control and freedom as the user is restricted to only using the menu to close it or pressing on another conversation/group chat to

exit it. Pressing outside the emoji menu in the current active conversation i.e. on the main message feed, does not close the menu.

- A functionality that exists in Telavox's app is the ability to create posts similar to a Facebook post where users can comment and like on a post. However, the corresponding icon to the post functionality does not exist in any chat other than a public chat room which makes it not satisfy the category recognition rather than recall.
- There is no way of sending multiple attachments at once, only being able to send one at a time, which is a hindrance to efficiency and makes it not fulfill the flexibility and efficiency of use evaluation.

UI

- There is a button in the group chat that does nothing and the icon is not consistent with creating posts even if it is in the same spot. Thus, the consistency regarding that button beside the chat input field does not satisfy consistency and standards.
- Regarding the text, that is the font of the application, there is no option to customize the font size in case the user wants to increase or decrease the text. This could be an accessibility issue, but also does not fulfill flexibility and efficiency of use.
- When the user first opens the chat application when they log in, they are directed to a page where they are notified of all the groups with new messages and the latest message sent from said groups. The user is also able to navigate to the group by pressing on the message. However, there is no clear indicator of navigating to this view unless you press on the chat icon above the search bar, making the user recall rather than recognizing it.

Miscellaneous

- There is no way of searching for specific messages which makes the app somewhat inefficient to use.
- Only way of connecting messages is by using @username, but not to the messages themselves as they are a method of notifying the user. People have to make the connection that the message is related to another message sent by the user who is used, making it inefficient to use when wanting to find the related message.

3.1.2 Purpose and plan of action

There was an initial study of Telavox's internal chat app to investigate if there was a demand for chat threads or any type of organization of messages. To investigate this, interviews were held with the company employees to hear what their thoughts were. The questions that were created and asked to the employees can be seen in A.1 where they were both asked in Swedish and English depending on the interviewee's preferred language. They were about what the employees believed could be improved in terms of organizing messages as well as what their thoughts were about chat threads in general. The interviews themselves were semi-structured

to find specific information regarding threads in addition to open-ended questions to see if the interviewee had any additional thoughts about threads.

3.1.3 Results

The initial study had ten participants who were interviewed. Among the participants, one was within the age range of 18-23, four were in 24-29, one in 30-34, three in 35-39, and lastly one was in the age range of 40-44 (see figure 3.1). Additionally, figure 3.1 shows that eight of the participants were men, and the rest, i.e. two were women. All the interviewees were a part of Telavox where seven of the participants had a technical role as developers and the other three, had a role in marketing, customer support, and economics. Nine participants were positive towards chat threads in the internal chat app but also expressed that they wished for other chat-related functions. Those functions varied, but the most common when excluding chat threads were reactions, message replies, and the ability to edit sent messages. They believed that chat threads would have a positive effect due to them giving context to messages as to what message they are replying to, and allowing other people to easily follow the conversation or find information regarding a topic as it gathers several messages in one place. However, there was a participant who conveyed that they did not believe that chat threads would be beneficial to the chat app due to how cluttered it could become and cause users not to be able to tell who replied to whom. Additionally, they also expressed that the functionality of threads already exists in the form of the post function that the chat app has.

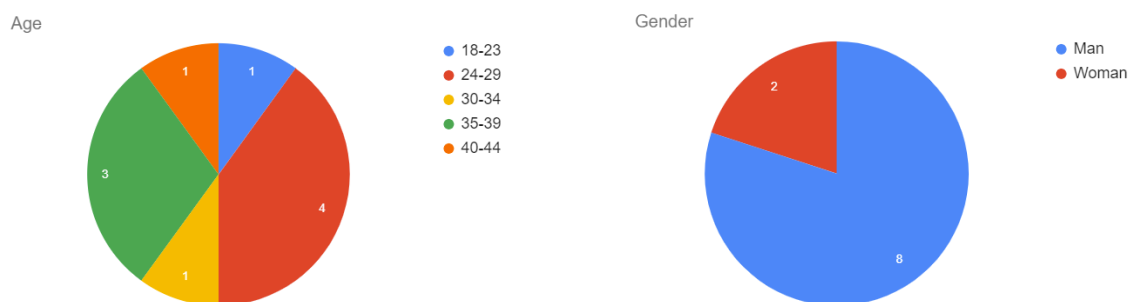


Figure 3.1: The age and gender distribution of the interviewees during the pilot study.

In order to organize the qualitative data gathered from the interview, affinity diagrams were used which can be seen in figure 3.2. The data were categorized into four, conceptual model, design, advantages with threads, and disadvantages with threads. Within the four categories, the data were further categorized to group the information from the interviews.

3.2 Prototype

3.2.1 Idea generating process

The method of generating the ideas for the new design was the use of brainstorming. However, the ideas were grounded in the data that has previously been gathered from the heuristic

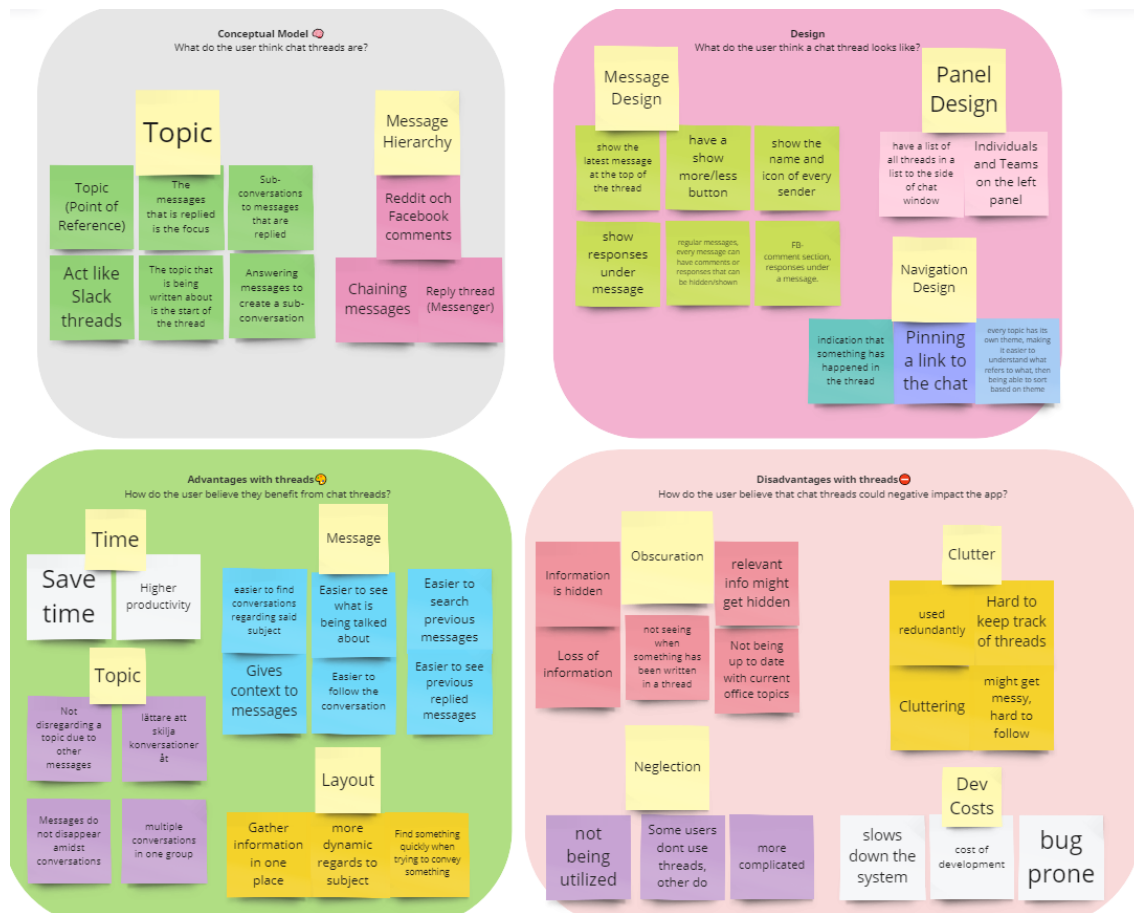


Figure 3.2: Affinity diagram of the data from the pilot study. Each aspect is shown individually at B.1, B.2, B.3, and B.4.

design evaluation and the pilot study conducted on the employees of Telavox. Before brainstorming, Telavox already had assets that the UX design team for their internal chat app. They were kind enough to allow us to use their assets and save time creating our own in order to replicate their chat app in Sketch.

The ideas were thought upon separately and then when the process was done, the two of us discussed the reason behind our ideas. In order to properly explain the ideas, quick and small prototypes that were of mid-fi fidelity were created in Figma. As previously mentioned, with the help of the employees of Telavox who had assets made, the prototypes were later created in Sketch in order to accurately portray the ideas in higher fidelity. The prototypes, however, were not implemented with functionality until discussing and deciding which prototypes to go forward with. As for the discussion, an internal ranking based on the categories from the pilot study was formed and depending on the score, would be narrowed down to two prototypes which then will be converted into hi-fi prototypes for comparative testing. The reason for using comparative testing is due to the scope of this thesis as testing one single prototype on a large number of participants is infeasible. Furthermore, it would allow for testing on a greater range of features and design choices which in turn would yield more data to answer the research questions. Additionally, as we would be using a version of comparative analysis where the participant would test both prototypes, where some would start with prototype 1 and the rest with prototype 2, more data would be gathered from a smaller group

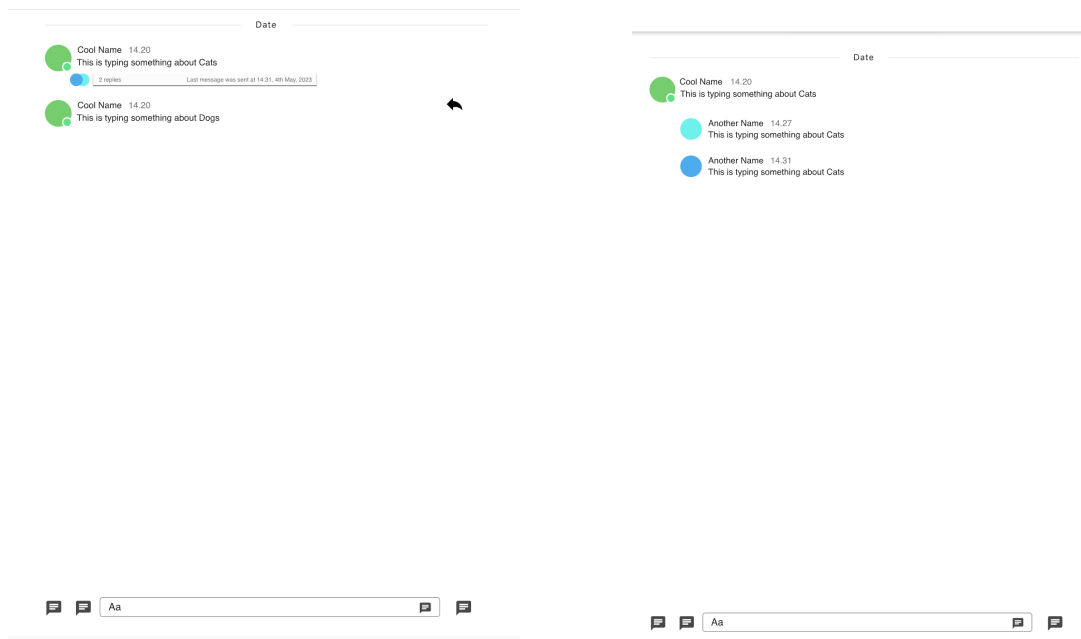


Figure 3.3: The first prototype that was created for brainstorming during iteration 1.

of participants.

Prototype 1, Segregated button threads

Prototype 1 (see figure 3.3) was created with heavy influence from the already existing function chat threads by Slack. The idea behind their design is minimalism while at the same time allowing the users to see when the thread has been updated. Additionally, it shows the users in a group chat how many unique people have replied to the message as well as how many replies exist within the thread. It has two affordances, the rectangle and the thread reply icon on the right of the message, which acts as an affordance and a signifier. One difference is the layout of the chat thread when it opens as it redirects the user to a whole new page dedicated to only the thread messages. This constrains the user and forces the user to only type in the thread. In the same layout, the messages are slightly indented to signify that the messages belong to the message being responded to and to show a hierarchy similar to Facebook messages.

Prototype 2, Segregated forum threads

Prototype 2 (see figure 3.4) was created on the basis that messages are shown that they are chained or connected to each other. In order to showcase this, lines connect to the indented thread replies below the message being replied to. Furthermore, only the two latest messages in the thread will be displayed on the "main" message feed and will continuously be updated as more messages are added to the thread. Both the messages will be acting as affordances as when you press them, the user will open a side panel that will shrink the main message feed and show the thread replies with the topic message being on the top. This is to allow the user

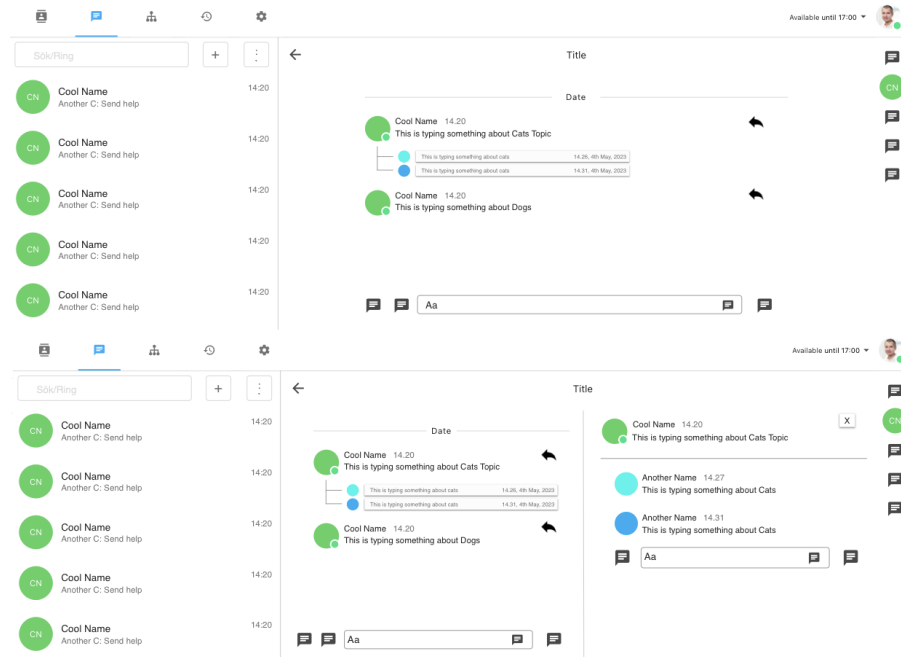


Figure 3.4: The second prototype that was created for brainstorming during iteration 1.

to scroll through the main message feed while viewing the thread and its replies at the same time.

Prototype 3, Color threads

The idea for this prototype (see figure 3.5) was to build upon the existing functionality of the Telavox chat app. This was done by adding a form chat threads, in a way that would keep the option to use the app similarly to the original version. The way it was intended to work was that a user could reply to any message in the chat room by clicking a button tied to the message in question. This would tag both messages with a color marker to indicate that they are tied together. Any further messages replying to either would then be tagged in the same way as well. This way the feed would keep its chronological order while making it easier to distinguish what messages are related to each other. In order to make the threads or topics easier to read even further the user would be able to choose a color tag to view. The feed would then only show messages with that tag.

Prototype 4, Inline threads

The fourth prototype (see figure 3.6) was developed with similar ideas as the third, allowing the users to have the option of viewing a chat log in chronological order and the ability to view just a specific thread. In this prototype this was done by having three kinds of views the user could enter. One is the full log of a certain chat where everything is chronological. The second view is similar but here messages that are tied to a thread are compressed to just show the latest message below the start of said thread. At such a thread header there is the option to show all messages through a pop-down visual. The last view is meant for singular threads,



Figure 3.5: The third prototype that was created for brainstorming during iteration 1.

this is similar to the first view but the messages shown are just from a selected thread. This prototype also had the functionality of showing a list of the headers to the threads that exist in the chat. This was to give an overview regarding new messages and quick access to the singular thread views.

Prototypes of choice

The internal ranking criteria that were formed as a result of the data from the pilot study, were created with the purpose of helping decide which prototypes to add interaction for testing. The points are distributed by comparing the ability of the prototype to meet a criterion against the Telavox chat app's ability to meet the same criterion. As for the point range for each criterion, most were within the ranges of -5 to 5 points while three criteria (obscurity through navigation, cluttering, and neglect) had ranges between -5 to 0. When a criterion is given a negative value, it is seen that the Telavox app is better at that particular criterion, while a positive value is seen that the prototype performs better.

As a result of ranking the prototypes, the third and fourth prototypes were ranked the highest (seen in table 3.1). However, due to the two prototypes that were created by each person, being similar to each other in comparison to the other person's ideas, we chose to disregard the third prototype as it ranked lower than the fourth and instead chose the third

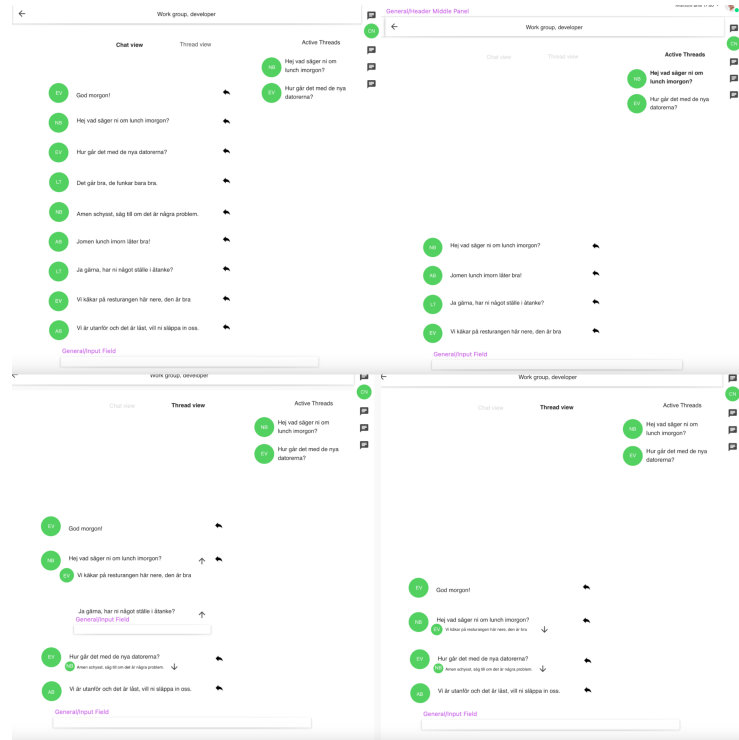


Figure 3.6: The fourth prototype that was created for brainstorming during iteration 1. Larger pictures of the views are in the appendix at C.4, C.5, C.6 and C.7.

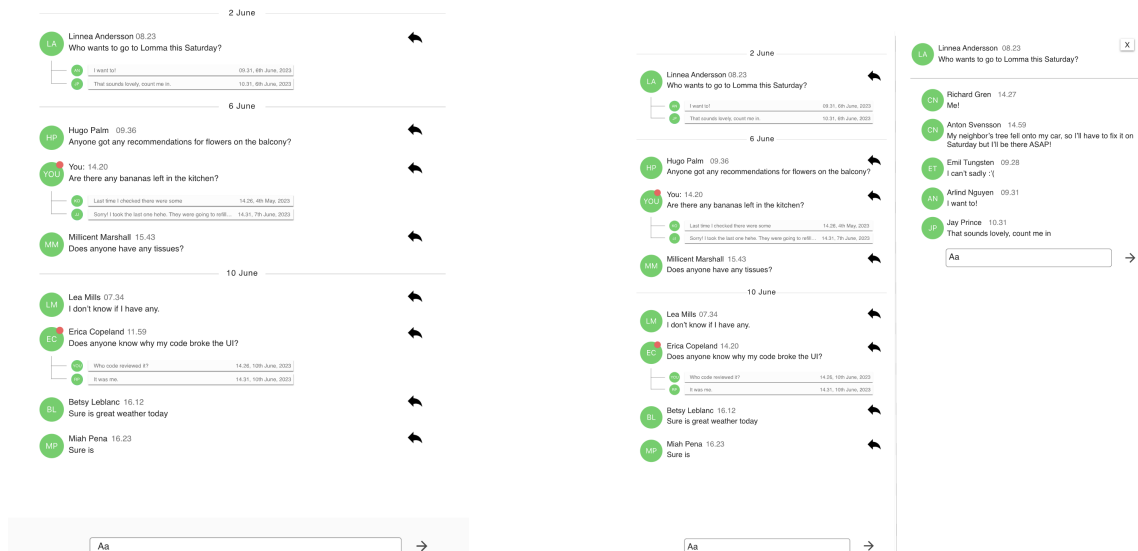


Figure 3.7: The final design of the segregated prototype.

highest ranked i.e. the second prototype. This is because we want to see a broader result from testing two prototype concepts and the impact of different functions and designs instead of small changes. Furthermore, the design of the two chosen prototypes (segregated forum and inline) was more aligned in terms of the conceptual model that the user already had regarding chat threads. The two prototypes that were chosen would then proceed to be further

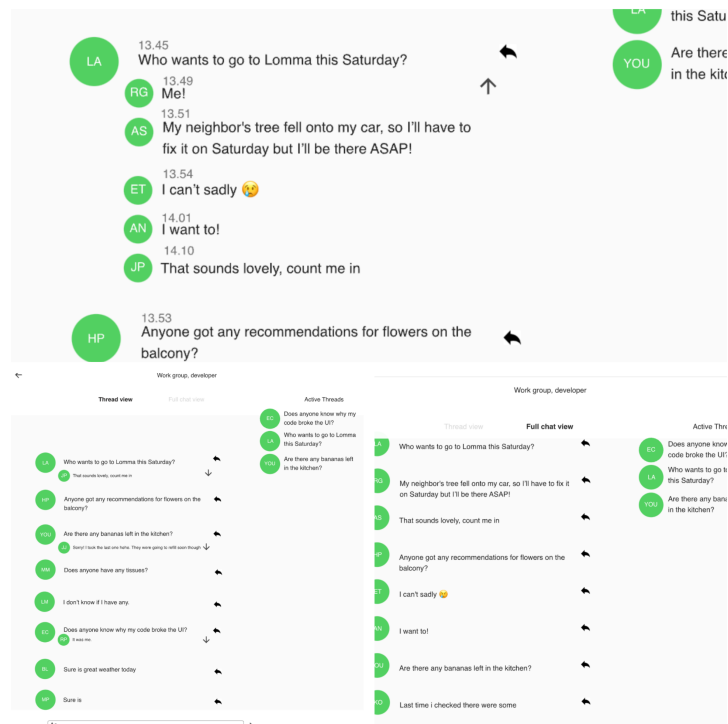


Figure 3.8: The final design of the inline prototype.

polished in terms of design and once that was complete, add interaction to the prototypes. The inline prototype will be referred to as prototype 1 and the segregated forum prototype will be referred to as prototype 2 in this iteration. Figure 3.7 and 3.8 were the final versions of the two prototypes once they were ready for testing.

3.3 Prototype Testing

Purpose and goal

The purpose and goal of testing the prototypes was to find out what design the participants preferred when it came to chat threads. As a result, some objectives or questions were created in order to structure the test plan and make it easier to receive qualitative data that would aid in answering the research questions posed in this thesis. The test plan should in some way cover what kind of navigation the user prefers, what ways could a design achieve high readability, the ease in which users could find the messages they are searching for, what design could achieve a higher level of intuitiveness, and lastly how useful would the functionality of chat threads in a workplace environment?

Research questions

- What effects do chat threads have on text communication?
- How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability?

Table 3.1: The ranking system and the categories that were made during the decision of which prototype to test where a ranking of 0 is the same as the Telavox application.

Segregated Button	Points
Time efficiency writing	-1
Time efficiency navigation	-3
Time efficiency reading	3
Message Cohesion	3
Topic Distinguishability	4
Message Layout	5
Obscuration through messages	3
Obscuration through navigation	-2
Cluttering	-1
Neglection	-1
Total	10

Color	Points
Time efficiency writing	0
Time efficiency navigation	-1
Time efficiency reading	3
Message Cohesion	2
Topic Distinguishability	3
Message Layout	4
Obscuration through messages	4
Obscuration through navigation	0
Cluttering	-2
Neglection	-1
Total	12

Segregated Forum	Points
Time efficiency writing	-1
Time efficiency navigation	-2
Time efficiency reading	3
Message Cohesion	3
Topic Distinguishability	4
Message Layout	5
Obscuration through messages	3
Obscuration through navigation	-1
Cluttering	-2
Neglection	-1
Total	11

Inline	Points
Time efficiency writing	-1
Time efficiency navigation	-1
Time efficiency reading	3
Message Cohesion	2.5
Topic Distinguishability	4
Message Layout	5
Obscuration through messages	4
Obscuration through navigation	-0.5
Cluttering	-2
Neglection	-1
Total	13

- How do different communication systems affect the efficiency of the teams/people using them?

Data to be collected

The data that was collected from the users can be categorized into four categories, objective quantitative and qualitative data, and subjective quantitative and qualitative data. Within objective qualitative the data was gathered by analyzing trends and patterns of behavior when interacting with the prototypes as well as within the subjective data collected from the quantitative and qualitative data. For subjective quantitative we used SUS and for qualitative we used a post-test interview with a predetermined set of questions and potential follow-up questions. Table 3.2 summarizes the categorized data and the means to collect said data.

Table 3.2: The data to be collected for each research question and what measures are used.

RQ	Objective Qualitative	Subjective Quantitative	Subjective Qualitative
#1	Analyze trends and patterns	System Usability Scale	Post-test interview
#2	Analyze trends and patterns	System Usability Scale	Post-test interview
#3	Analyze trends and patterns	System Usability Scale	Post-test interview

Participant characteristics

To stay within the scope and to make the testing feasible, 4 participants would be currently working at Telavox, and 6 would be friends and family that have volunteered to participate in our study. Furthermore, to ensure that the privacy and anonymity of the participants would be maintained, they will be referred to as participant #x or as a collective. This would limit the broad range of experience that would typically be desired when testing as different backgrounds and experiences would affect the way the participant interacts with a product i.e. everyone's conceptual model is different. Most of the participants will therefore have some kind of technical background and be within the age range of 24-29 with some exceptions.

The two chosen prototypes were divided into two parts of a comparative testing schedule. The subject groups were divided into internal participants from Telavox and external participants not affiliated with Telavox. Both these groups were then once again split into two groups where one group began with prototype 1 and the other started with prototype 2 depending on the order in which they tested the prototypes, i.e. every odd-numbered participant would start with prototype 1, and the even-numbered participant would start with prototype 2. This is meant to counteract the possibility that participants would rate the first prototype higher than the second that they would test.

Test environment and equipment

The equipment that was used when conducting the tests was the following:

- Microphone for recording audio (OnePlus 5 and OnePlus 9 was used during testing).
- Two MacBook Air M1 laptops, one for interacting with the prototype and recording the screen, and a second for taking notes of the participant's answers to the interview questions.

The test environment was fairly flexible as it only required the equipment stated above. Preferably, the test environment would be in a room where the microphone would be able to pick up the audio that was recorded and allow the participant to focus on the task at hand without any disruptions. Furthermore, it would be most optimal if only the moderators were present in the room with the participant as other people would not affect the participant's performance while performing the various tasks given to them. The moderators themselves should do nothing more than ask the questions written in the post-test question list and if

relevant ask follow-up questions for clarification to answers given by the participant, give the tasks to the participant, and clarify when the tasks given are unclear to the participant.

Procedure

Participants were first given an introduction to who we, the moderators, were as well as the thesis we were doing. They were informed of the general procedure of the test to allow the participant to receive an overall understanding of what would happen during the test. The procedure of the test was that they would begin with one prototype, and when the participant was done with the tasks belonging to the prototype, they would answer a questionnaire that would ask about their age, gender, and current occupation. Additionally, they would also answer an SUS questionnaire regarding the prototype they tested. This will then repeat once again, but they test the other prototype and skip directly to the SUS questions for the new prototype they tested. Furthermore, they were also informed that the participant would not be tested on their ability to perform certain tasks, but on the interaction of the system itself. Lastly, we also asked the participant if we had their permission to record their interaction with the prototypes as well as their audio for later usage and also informed them that everything would be anonymous and their name would in no shape or form appear in the thesis. Once all the steps regarding the prototypes are done, the test will round off with some questions to gather subjective qualitative data. The overall estimated time for the procedure was roughly 30 minutes and all of the steps are summarized in table 3.3.

Table 3.3: The overall procedure by which the usability evaluation was conducted.

Phase	Description
Briefing	Background info and recording consent
First prototype	Completing given tasks
Answering questionnaire	Answering background information and SUS questions
Second Prototype	Completing given tasks
Answering questionnaire	Answering SUS questions
Post-test interview	Answering questions regarding both prototypes

Tasks

The tasks that were given to the participants were the same with the exception of prototype 1 as it had an additional task to showcase the active thread feature as well as the views. Table 3.4 shows the tasks that both prototypes had, and the additional tasks exclusive to prototype 1. Overall, the tasks were chosen in order to enable the tester to experience the functionality of the prototypes in order to help us answer the research questions.

The first task was to find a given message and then respond to it which would then create a thread with the reply message in it. This task was to determine the usability of creating a thread and the affordance. When the first task was completed, they would move on to find a

specific message that was hidden in a thread, this was to simulate the scenario when the user vaguely recalls a message and would need to navigate towards it. This scenario was meant to investigate the navigation behind the prototypes and how the user would naturally act. If the participant is testing prototype 1, this is where an extra task would be added and they would be given the task to find the same message again, but instead do so in an alternative view i.e. the chat view instead of in the thread view. The third task was to find a new reply that had been sent to one of the threads or messages that the user had sent. This was to investigate the user's ability to find new messages in the scenario where they have been inactive from the chat application and came back to check new messages and/or when they receive notification about an update. Lastly, the user was asked to find a message that was sent to a thread the user had previously written in. This was to check the navigation and notification of how the user would be able to find new messages in the scenario where the new messages would be relevant, unlike the case where other irrelevant threads would be updated and the user would not receive updates.

Table 3.4: The tasks that the participants were asked to complete to explore the two prototypes and their functionalities and design.

Tasks	Description
#1	Create a thread by responding to a message
#2	Find a specific message that is inside a thread
#2a	Do the task #2 again, but in an alternative view (exclusive to prototype 1)
#3	Find the new message to a question the user has posed
#4	Find the new message to a question that the user has answered

3.4 Results

The participants were divided into two groups depending on whether they were from Telavox or not. Telavox employees were put in the internal group while participants from outside the company were placed within the external group. As per comparative analysis, half of each group started with one prototype while the other half started with the other. The age and gender distribution of all the employees as well as the external participants can be seen in figure 3.9.

In general, a majority of the participants, with the exception of two, rated prototype 2 higher than prototype 1 regardless if they started with prototype 2 or not. This can be seen in table 3.5 with all participants' individual scores as well as the box plot given in figure 3.10. Furthermore, when analyzing the data of the scores for both prototypes, the statistics in table 3.6 suggest that participants, regardless of group perceived prototype 2 to have better usability. The internal participants valued the second prototype more than the first while in the external group only participants 5 and 10 were outliers. Participant 5 ranked both equally as good, while participant 10 valued the first prototype more than the second one, despite testing prototype 2 first.

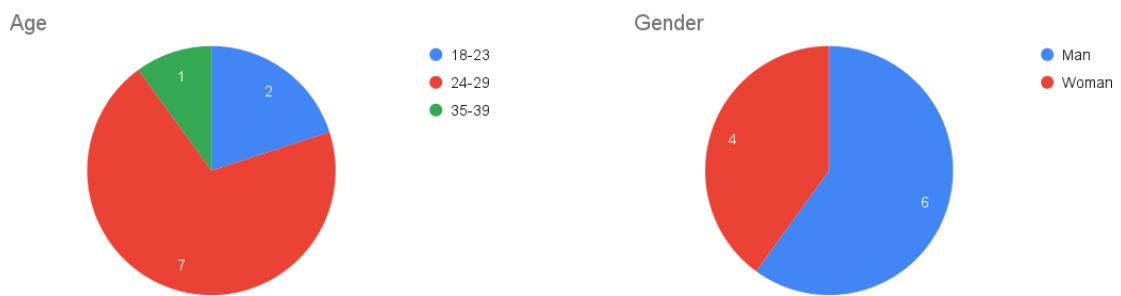


Figure 3.9: The age and gender distribution of the participants.

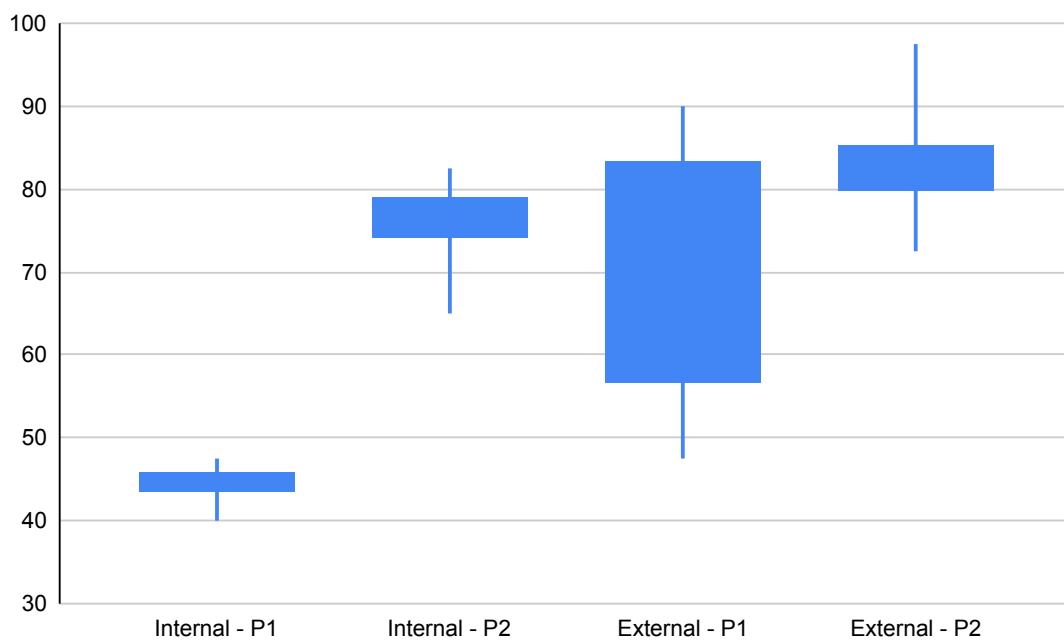


Figure 3.10: Box plot of the SUS score in iteration 1 for each group where P1 and P2 stand for prototype 1 and prototype 2 respectively.

Something to note with the testing order of the prototypes was that the participants were not equally distributed regarding which prototypes they started with. As a result of this, the internal group lacked one participant starting with the second prototype, and the external lacked one participant starting with the first prototype. However, since the results are heavily skewed towards prototype 2, which prototype the participants start with does not seem to have a seemingly large impact on their evaluation of the two prototypes. Another factor to take into consideration during this iteration of usability evaluation was that the prototype design could have affected the result as it was not a one-to-one copy of the already existing Telavox chat app. This could be seen when some participants noted the increased difficulty of differentiating users in the chat due to them being able to do so more easily when there are unique profile pictures for each user.

The results from the conducted usability evaluations can be categorized depending on the prototype in question. For prototype 1, the categories of the data gathered were features,

Table 3.5: A compilation of the SUS scores for all the participants for each prototype. External participants are marked with "*". Odd-numbered participants started with prototype 1 and even started with prototype 2.

	Prototype 1	Prototype 2
Participant 1	40	77.5
Participant 2	45	82.5
Participant 3	45	65
Participant 4*	47.5	72.5
Participant 5*	85	85
Participant 6*	70	80
Participant 7*	77.5	85
Participant 8*	52.5	97.5
Participant 9	47.5	77.5
Participant 10*	90	80

Table 3.6: The statistics of the SUS score retrieved from the tests with the mean, standard deviation, and the confidence interval of 95% in both groups and the values for both prototypes.

	Internal - P1	Internal - P2	External - P1	External - P2
Mean	44.38	75.63	70.42	83.33
SD	3.15	7.47	17.28	8.32
95% CI	[39.37, 49.38]	[63.75, 87.50]	[52.28, 88.55]	[74.61, 92.06]

layout, and ease of use. On the other hand, prototype 2 had the same categories with the exception of features due to the fact that prototype 2 did not have any results that could be categorized in features in comparison to prototype 1 which had the active threads functionality. These categories can also be seen in figure 3.11 and the overall affinity diagram can be viewed in figures B.5, B.6, and B.7.

Prototype 1

Features was one of the categories from the data gathered and one of the more appreciated features amongst the participants was the existence of the auxiliary feature active threads, a method of showcasing threads that existed in the group chat and allowed the user to access all the saved threads. However, there were many concerns regarding this feature as the display of "active threads" text caused confusion regarding what the functionality did. Some preferred that it would instead display the threads that the user participated in themselves, and even its display of the latest message was not clear to the user. In addition to the active threads feature, having view alternatives of the chat was seen positively as it gave the user a degree of freedom to choose what view they wished to display, but the functionality between the alternatives, that is the thread view and full chat view was not apparent to the users at first glance. There was also a participant who expressed that the full chat view was unnecessary.

The layout design had the most feedback from the participants about the prototype. One



Figure 3.11: Affinity diagram of the data in iteration 1. A larger image can be seen at B.6 and the affinity diagrams for general comments and prototype 2 can be viewed at B.7 and B.5.

feedback was that the thread view functionality caused the app to be less cluttered when it comes to the messages. Furthermore, it was also a boon that the user was able to see all the threads that exist in the group conversation. Participant 5 expressed that the design of the prototype was simple and easy to use at the start while participant 3 said that it was clean. However, there were also concerns regarding the layout as there was no back arrow that worked, contributing to the navigation being convoluted. Furthermore, the redirection of navigating into a thread, both when creating and navigating to view the replies, was confusing according to participant 1. Additionally, the overview was confusing as there were too many views for the user to keep track of, as well as the views themselves had a low affordance to the user as they had a hard time keeping track of where they were located. Coupled with the fact that with the many alternatives the user had, participant 1 said that it caused the prototype to have too much freedom and too many roads that could take the user to too many places, which caused further confusion to the user.

The data regarding the ease of use of prototype 1 pointed towards that the idea behind the implementation was good, but caused frustration for the user in practice. Participants 8 and 10 conveyed that the design was familiar as it was closer to a normal chat. In combination with the fact that threads act more similarly to a drop-down menu in thread view, it was more integrated into the normal chat which in turn could cause the chat to feel more familiar to the user. It was easy to start a thread and the concept of threads was appreciated as it made the view less cluttered overall and allowed participants to organize messages and then find them later on. However, the design of the prototype did seem to be perceived as unintuitive to some of the participants and participant 8 stated that it would take time, but would eventually learn how to use it.

One common feature that the participant wanted to have was a search functionality to be included in some way to search for specific messages. Another that was seen that could have steered the prototype in a more positive direction is to have the active threads panel be threads that the user has participated in instead of displaying every single thread that has occurred in the chat room. The functionality in the dropdown menu could also be improved in the thread view so that the user would be able to write in the panel that opens. Additionally, when navigating into a thread, a back button within that view would have improved the navigation as it would allow the user to go back to their previous message. It would also be an improvement if the user would not be directed to a new page when trying to respond to a message. The two views could be improved in terms of conveying to the user what they would do, as well as limiting the number of views as both of them were seen as redundant. In addition to that, giving the user feedback on which view would most likely increase the user experience. Lastly, a way to tie the messages in full chat view to their thread would ease the navigation as well as give the user a greater overview and a better sense of where they are in the application.

Prototype 2

As prototype 2 had no additional auxiliary features to threads, only the threads themselves, all the focus was put on the overview and layout design to be simple for the user to understand. The overview gave info at a glance to the user which seemed to not have a large effect as a countermeasure against obscurity. However, the overall design did give a sense of professionalism according to participant 5, and in general, was seen as a simple overview by most

of the participants which in turn made it a good overview for the participants. The design also allowed the participants to find the relationship and context between topic messages and responses which the participants seemed to have no problem doing. Another comment from the participants was that the overview when entering a thread (a side panel opens) was overall appreciated as it gave the user the freedom to read the messages outside the thread while reading the contents of the thread at the same time.

However, due to the nature of moving the whole view and making it smaller to accommodate the side panel, it did introduce cluttering which in some scenarios could be a downside. The cluttering in general was overall more than opposed to prototype 1 as a side panel was introduced. Furthermore, the icons for creating a thread were at the side of every message causing it to be more cluttered as well as redundant as the user does not need to see the option everywhere on every message. The thread overview in the message feed and its elements were also seen as both confusing and redundant as the thread line was unnecessarily thick and the thread reply display was unclear and had a small font making it difficult to read. Furthermore, the replies that were displayed were unclear whether or not it was the most two recent messages or not. There is also a possibility that the thread button had a low signifier as most participants did not enter the thread by pressing on it.

One improvement that could be made to prototype 2 was that the thread messages in the main feed could be combined into one larger message instead. This is to eliminate the small font size that made the message hard to read, increase the affordance of the reply as well as make it more clear to the user, and add more functionality to the thread messages such as the number of replies in the thread. Another improvement could be to introduce a feature that is similar to active threads in prototype 1 that displays to the user the threads they have replied in and/or save threads so that they will receive notifications in spite of not being a part of it. Lastly, a way of giving feedback to the user when they are viewing the thread to show which thread they are currently in.

3.5 Result Analysis

This section is an analysis of the results from iteration one based on how they answer the research questions.

What effects to chat threads have on text communication?

What was gathered from the results of the usability tests that had been conducted, points towards that the users had an easier time finding information and understanding the context behind messages, i.e. what messages belonged to which conversation. Furthermore, it allowed the user to also identify and separate different topic messages from each other with the added benefit of which messages responded to the topic message.

The threads themselves do not seem to introduce any redundancy unless it is directly added to the UI. This can be seen when creating affordances in the UI where the user has the ability to enter the thread by simply pressing on the icon. However, threads did still introduce redundancy in form of the cluttering the message feed as it introduced more elements to the user. That in turn, forced the user to keep track of more elements and what functionality the added UI had. In other aspects, the functionality of prototype 2 reduced the redundancy as

it allowed for two different conversations to occur at the same time, effectively eliminating one problem while at the same minimizing the number of issues introduced.

How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability?

One of the things that was noted early on making it as satisfactory for the user as possible was minimizing the number of times the user is redirected to new pages, views, and windows and only to do so when there is an explicit need for it. Something to keep in mind when designing threads is to "move the world around them" which is to be interpreted as creating threads in a way so that the navigation of threads is kept consistent, and grounded as well as simple and clear. This is to make the user have a better user experience in terms of being able to understand where they are present in the program. To further enhance it, a good overview of the application will contribute to the user having a better understanding of where they currently are. It also helps when the user is in control of the program and not the other way around. However, the design should not give the user too much freedom and many ways to navigate through the app as it can lead to confusion instead of a clear and intuitive way of moving through the application.

Another factor to keep in mind when designing threads is to create and introduce a design that is familiar to the user which in turn will make the feature easier for the user to use as they are already familiar with it. It was also noted that a way to save threads and keep track of threads when they are updated would improve the experience of using threads as users might want to keep track of more important chat messages/conversations as older conversations would be lost when newer messages and conversations are written in the chat. Lastly, the threads should have a way to show a clear relation between the topic message and its replies.

In order to promote and encourage users to use threads, the user has to first and foremost experience that the act of creating a thread is simple and intuitive. In addition to that, it is crucial that the overview also be adequate and the navigation should also be straightforward and satisfactory. This is to hopefully encourage and help the user to experience that the making of threads is useful as well as not forcing them to extensively navigate to proceed to the thread. One avenue that could encourage the users to use threads more often is to constrain the user by making the message reply button a thread creation and implementing a way for normal replies, one such example is using the @ sign followed by the user's name. Another way that could promote more thread usage is by not having replies as the only way to navigate to threads, but also other ways to navigate to the thread replies through a functionality. This could for example be done by having a button underneath the thread message. Threads could also be more encouraged to be used if there is a way for the user to receive notifications from threads, both old or new when they are updated with new messages.

There have been mentions of having the navigation to be smooth and intuitive, but in order to achieve it, the design should have some kind of overview of the threads themselves. Two examples could be to list all threads and display them and another would be to display the threads that the user themselves have participated in. Active threads made use of displaying all threads and not only threads that the user has participated in. This could counteract obscurity and minimize the traveling between chat rooms as it would also allow the users to navigate between chat rooms if the threads exist in a different chat room. One noted discovery when navigating between threads in the current chat room the user was in, was that

the side panel allowed the user to both read the content of the message while also navigating between other threads as the main feed was shown in parallel with the content of the thread. The UI should not be overly cluttered with buttons and icons, but at the same time be subtle enough with its interaction for the navigation to be more intuitive and not overwhelm the user.

How does different communication systems affect the efficiency of the teams/people using them?

While it was not tested during the usability evaluation, the participants conveyed that the prototypes could improve efficiency compared to a bare-boned application and that it overall increased the efficiency of the communication. Compared to the Telavox chat app which is bare with barely any functionality in the chatting portion of the app, this was seen as an improvement and would affect communication positively as it would be easier to communicate with other people as well as read their messages. Furthermore, threads encourage communication that would be sent to a person privately to be seen in the public space of the chat room, allowing other users to follow along and possibly contribute to the conversation.

The point that needs to be taken into consideration when creating communication systems is to understand what effective communication in practice entails. Text communication in chat applications should have the ability to easily find information and messages. Additionally, the system should have the ability to respond to messages in order to create context for other users to determine what conversation the message belongs to. Furthermore, the context that is then created should also be easily seen by other users. Effective communication in the context of chat applications is also the ability to easily direct other people to specific conversations. To summarize, the points that are some factors of effective communication are the ease of finding information and messages, the ease of responding to messages and seeing the context of said messages, and lastly be ability to direct other people to specific conversations.

Chapter 4

Iteration 2: Refinement

This iteration covers the choices that were made based on the results from the feedback gathered in the previous iteration. It also goes over the development of a potential hi-fi prototype and then the decision to create and use two hi-fi wireframe prototypes instead. Lastly, it covers the changes in the testing plan and the data gathered from the usability testing.

4.1 Idea generating process

Taking the result from the previous iteration we wanted to develop either a hi-fi or evaluation prototype. The hi-fi prototyping in this case would consist of upgrading the prototypes from the previous iteration. The evaluation prototype would be a constrained implementation of the preferred prototype using the code for the Telavox chat app.

The benefits of upgrading the existing prototypes were first and foremost that the time investment seemed to be smaller compared to the implementation. This smaller time investment would also allow us to, if necessary, upgrade both prototypes and then test these again.

The benefits of implementing the evaluation prototype are that it would be more life-like and give the tester a more authentic understanding and feeling for the system. Their opinions would then be more grounded in reality instead of imagined.

The result of the previous iteration brought forth multiple opportunities for improvement, both abstract and detailed things. For example, fewer views in the UI to improve navigation and a better sense of location. From these potential improvements, the most wanted and desired ones were selected as goals for this iteration's prototype.

With these goals decided we came up with different ways to reach them to then evaluate between us allowing us to choose the ones thought to be the best, or ones that could yield valuable data.

The decided goals were a mix between individual ones for each prototype and some were desired in both. The first goal decided for both of them was to introduce a *hover menu* for

the button to reply or create a thread. This menu would appear to the side of the message being hovered. The second was to improve the icon used for thread creation, this icon should represent the concept of threads more. The third goal was to improve upon the function of being able to see threads that the user had taken part in. This turned into the goal of being able to save any thread with a pin function. The fourth goal was to show the time sent for each message.

The first goal for the inline prototype was to reduce the number of views to be more comprehensible, the full chat log was unnecessary. To further the flow of the prototype, the second goal was to be able to write directly under the chosen message or thread, under the pop-down area. The third goal was to improve the navigation and sense of location and the fourth was to make the prototype less convoluted. The threads should only have a pop-down area and not their own view to improve navigation and to make it easier for the users to know where they are.

The first goal of the segregated prototype was to have better highlighting of which thread is being viewed. This would make it easier for the user to understand what is going on and what they are doing. The second goal was to have some way of handling replies in old threads. The third was to show fewer messages in the thread header that is shown before a thread is opened. The fourth goal is also related to the header, but this was that the header should show the number of replies in the thread.

4.2 Development

The implementation of the prototype of this iteration was planned to be through coding by using the already existing Telavox code and developing a prototype that would not affect the current production of their app. However, due to the nature of the architecture and also the lack of knowledge on our part, there were problems being able to create a fully responsive prototype that would have the expected behavior of sending messages based on the user input. There was progress on the visual aspects of the prototype as well as functionality that would result in this iteration's prototype 2, but we decided to scrap this approach as the time we had was not enough for us to complete a prototype that would be ready for testing within the time frame we had. Furthermore, the only solution we could think of regarding the messages was to send in an already predetermined message that would not be the same message that the user would input into the chat. As a result of this, we decided to switch to prototypes created in Sketch because the interaction would be the same as the software-developed chat with the predetermined messages. Additionally, the time it would take to create the prototypes in Sketch would be quicker than coding as well as allowing us to create two prototypes and allowing us to have the opportunity to gather more data regarding different designs and implementations of threads.

The development of the prototypes was done first and foremost in Sketch, new assets were acquired making the process quicker and the prototypes better looking. These assets were symbols and icons already created to make a likeness of the Telavox chat app. With these, we could easily rebuild our prototypes making them more lifelike and appealing. This also meant that adding the improvements would be easier compared to not having these newfound assets.

One feedback that was gathered from the result of the first iteration of prototypes was the

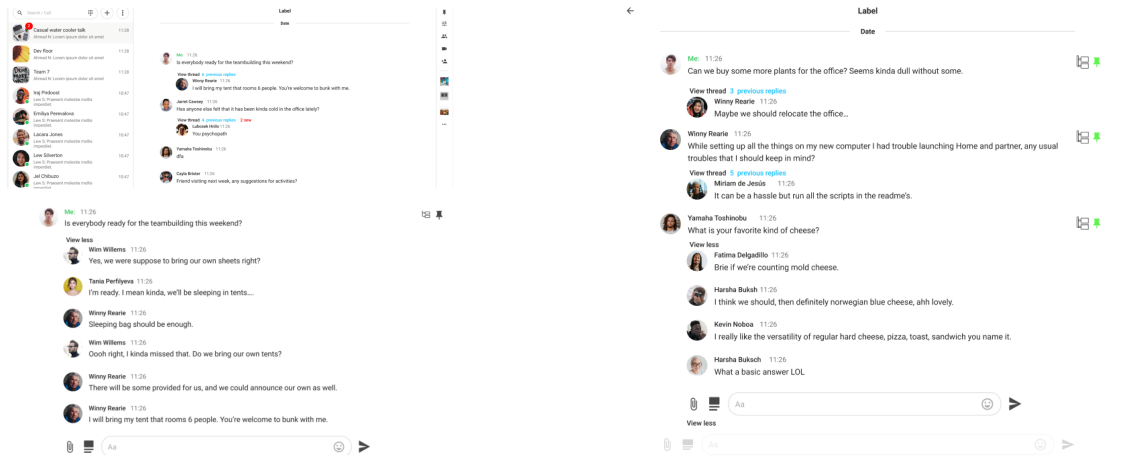


Figure 4.1: The final version of inline which was used for the usability evaluation.

thread create button that was cluttering the screen because they were attached to every single message that was displayed. To counteract this, we brainstormed a hover functionality that would instead make the thread button only appear when the mouse was hovered above the message. However, one issue that arose from this was that Sketch did not support interaction that would trigger hover functionality. What we looked into in order to resolve this was to use third-party plugins for Sketch as some had the functionality to support hover effects. In spite of this, there were many hurdles as most plugins either did not support a hover functionality or required payment to use up to a certain degree. This caused a shift in what program was used as we changed from using Sketch to using Figma in order to create the hover functionality. Furthermore, we also discovered during this shift that it was possible to interact with the created prototypes through digital means.

4.2.1 Prototype 1 - Inline

Since the prototypes in this iteration were updated versions of the previous they will be described in regards to their differences compared to the last version. Both of them got new graphics in the UI but again differ when it comes to the implementation of functionality.

How the new version of the inline prototype looks can be seen in 4.1. In order to respond to a message to create a thread the user must now hover over said message for the thread button to show. Instead of it being visible all the time for every message. While hovering pin icon appears as well, pressing this will save the thread to a list of pinned threads. This is used instead of having the concept of active threads. This makes the user able to save the threads they like and then find them in a list among other pinned threads. To find this list there is a pin icon in the top of the right column. When pressed the chat view is replaced by just the pinned threads. To then get back from this view there is a back arrow in the top left of the chat. And instead of there being a short list in the top right that gets updated every time there is a new message in a saved thread, there is just a red indicator on the top right pin icon.

The header containing thread information, located in the chat log and in the pin window is also changed. The information shown is the number of previous replies in blue, the amount

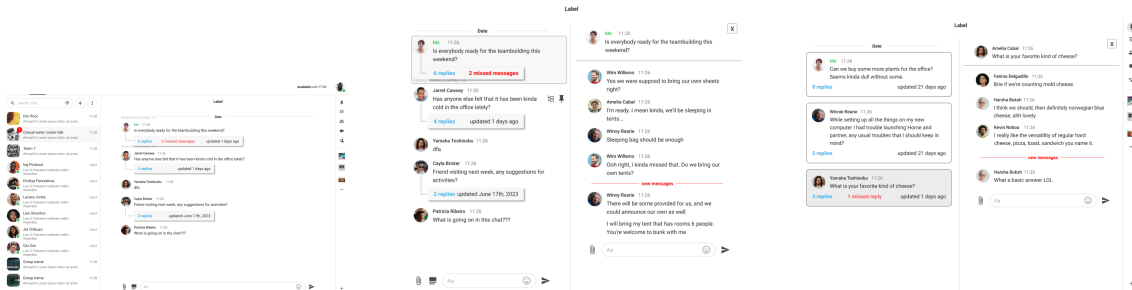


Figure 4.2: The final version of segregated which was used for the usability evaluation.

of new ones in red, and the latest reply sent. To pop down the the user can press either of these info texts as well as the text saying "view thread". When pressed the messages of the thread will appear beneath the header, and the info texts along with "view thread" are replaced by "view less". When this is pressed the thread is collapsed or closed again, this affordance is also located beneath for convenience.

The concept of being able to see every thread in their own view has been removed, it is only possible to see them in the pop-down area beneath their original message. This is also where responses to a thread are written. The view to see all chat, which shows every message in chronological order, has also been removed.

4.2.2 Prototype 2 - Segregated

The new version of the segregated prototype can be seen in 4.2. One core difference between the segregated prototype of this iteration and the one from the first iteration is the introduction of pins. In regards to the user interface elements of the pin, they are identical to the inline prototype. However, the segregated prototype differs when a user enters the pin functionality as a panel opens similar to threads, and in it are the saved threads. When the user then attempts to view the thread, the panel is moved to the left with the contents of the thread are shown to the right in another panel. This prototype also differs in the back navigation when inside the pin functionality as the only way of going back is by pressing the pin icon again or pressing the x-button on the right panel. Lastly, the pin icon in the message feed stays when the user pins threads.

Another change was the button that is displayed underneath the thread as it now has no messages shown in the button. Instead, the button now shows the amount of replies and the amount of missed messages as well as the last time the thread was updated. Similar to the inline prototype, in order to respond to a message to create a thread the user must now, in this prototype as well, hover over said message for the thread button to show. Instead of it being visible all the time for every message. The last notable change was the two buttons displayed underneath the message, merging into one larger one.

4.3 Prototype Testing

The test procedure of this iteration was conducted similarly to the previous one and can be summarized again with table 3.3. First, the testers received the same introduction as last time. They then got tasks to complete using one of the prototypes and then answering a SUS form. This is repeated for the second prototype before finishing up with a semi-structured interview.

Participants consisted of Telavox staff, which were found through their internal chat groups, and friends and family. Returning participants were divided into internal and external testers, the same was done with new participants. The separation between new and returning participants resulted in a total of four subgroups. Lastly, following the comparative analysis method, every other person in each subgroup started with one prototype while the rest started with the other.

The tasks themselves were also similar during this test, however, the specifics of the tasks themselves were different in order to accommodate new features. The first was to respond to a message in order to create a thread. The second was to find new messages that were located in a different chat and in two different threads. The third was to find a described message hidden in one of four threads. The fourth task was then to save three of the four threads by pinning them and then proceed to the pin window to view them, and then go back. The fifth and last task was to find a new message sent in one of the threads that previously had been pinned. A summary of the tasks can be found in table 4.1.

Table 4.1: The tasks that the participants were asked to complete to explore the two prototypes and their functionalities and design in iteration 2.

Tasks	Description
#1	Find Yamaha Toshinobu and create a thread
#2	You have new messages that you have not yet read, find the people who wrote the new messages and the message content.
#3	You are gonna set up a new computer as your old one broke down. While setting up you had some issues, but you remember someone saying that you should spam a command in the terminal. Find the message.
#4	You want to save the threads that you want to keep track of when they update. Pin the first three threads starting from the top and going down. When done, go into the pin function. Feel free to check the threads that are displayed.
#5	Now, imagine some time has passed and new messages have been written. One of the saved pinned threads you have, was recently updated. Please find the new message in the updated thread by using the pin function and read the message.

4.4 Results

The participants in this iteration of usability evaluation measured up to 20 people in total, where 14 were people outside of Telavox and the other six were employees from the company.

Of the 20 participants, half of them were women and the other half were men as seen in figure 4.3. In regards to the age distribution, 15 were within the age range of 24-29 while 2 were in 30-34, and 1 each in the ranges of 18-23, 35-39, and 40-44 (see figure 4.3). Additionally, all the external participants answered that they have never used the Telavox chat app while all employees of Telavox answered that they very frequently use it. All of the participants' individual SUS scores for both prototypes are in table 4.3.

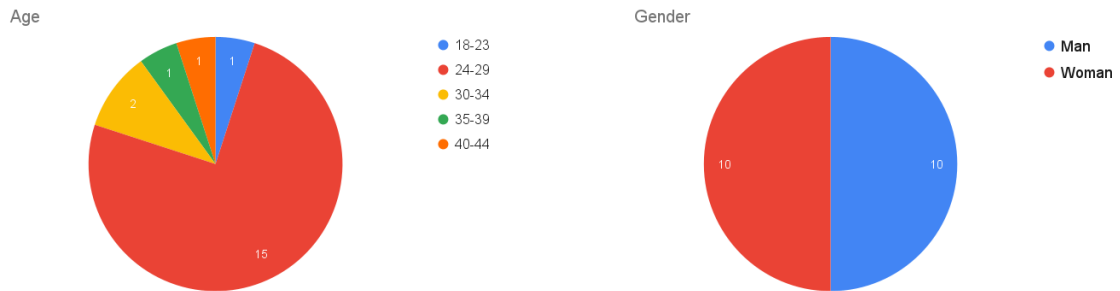


Figure 4.3: The age and gender distribution of the interviewees during the pilot study.

From the SUS score gathered from the usability evaluation, the inline prototype (prototype 1) seemed more appealing to the external users while the internal mostly found the segregated prototype (prototype 2) appealing. This can be interpreted through the box plots of the SUS score in each group in figure 4.4. This is further reinforced when looking at the confidence interval in table 4.2 as the range is overall higher in prototype 1 than prototype 2. Furthermore, the standard deviation is lower for prototype 1 than prototype 2 in both groups. The plots in figure 4.4 suggest that the external group, which mostly consisted of students and people who worked in small offices or in small companies, may not have felt the need for the segregated threads.

Table 4.2: The statistics of the SUS score retrieved from the tests in iteration 2 with the mean, standard deviation, and the confidence interval of 95% in both groups and the values for both prototypes.

	Internal - P1	Internal - P2	External - P1	External - P2
Mean	77.92	85.00	83.21	79.46
SD	12.79	16.05	17.44	19.04
95% CI	[64.50, 91.34]	[68.16, 101.84]	[73.14, 93.29]	[68.47, 90.46]

All the data was collected and summarized in an affinity diagram similar to the affinity diagram in figure 3.11 which can be seen in figure 4.5 with the exception of the categories as layout was changed to user interface. Overall, a majority of people who participated in the usability evaluation preferred the inline prototype, more people still expressed in the post-test interview that the segregated prototype was both more professional and structured than inline threads. However, the inline prototype was seen as more familiar and intuitive. Another feature that the participants in general seemed to appreciate was the clear indication of new messages. The inline prototype lacked the red line that indicated where the new messages started within the thread which was expressed by the participants to be somewhat

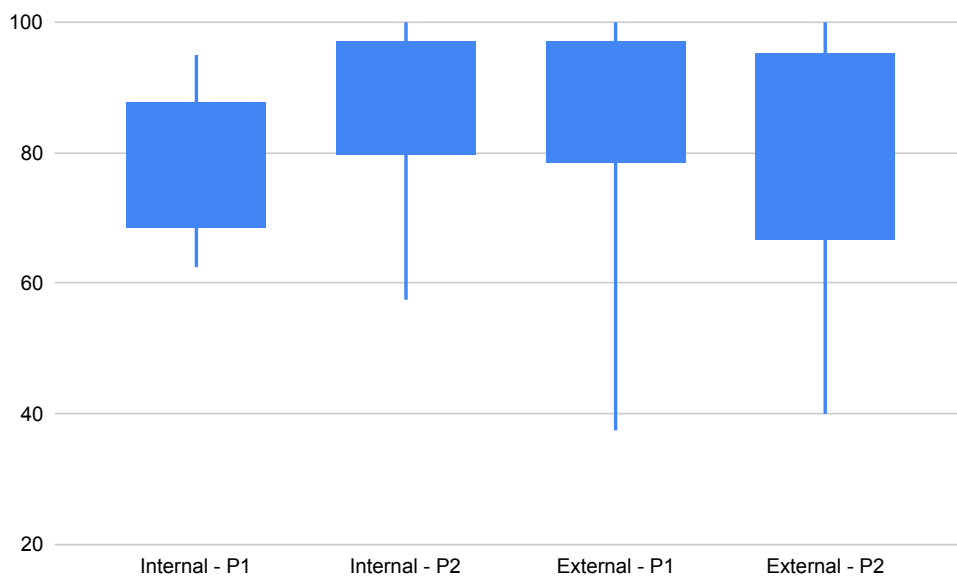


Figure 4.4: Box plot of the SUS score in iteration 2 for each group where P1 and P2 stand for prototype 1 and prototype 2 respectively.

difficult to discern which messages were new. Contrary to that, the segregated prototype did have a red line as an indicator which participants conveyed was an overall better feature and allowed for easier discernment. On the other hand, the inline prototype had the text "new" instead of "missed messages" in the segregated prototype. Some participants noted that they preferred "new" instead of "missed" as it did not make them feel judged for having not read the new messages.

Prototyping had issues where the functionality did not work as intended and did not show on the participant's screen when in reality it worked as usual when observing through Google Meets. In one instance of this, the participant said that they like the currently chosen chat to be highlighted in some way. When viewing their screen directly it was not highlighted, when viewing it through the share function in Google Meets it was highlighted as intended. How and why this occurred we could not find out, possibly how the computers visualized color codes. However, that should still make the highlighted area a different color from its surroundings.

4.5 Result Analysis

This section is an analysis of the results from iteration two based on how they answer the research questions. This was done separately from the previous analysis, 3.5, and they will be compared in the discussion, 5.3.

What effects do chat threads have on text communication?

From the responses during the interviews, it was brought forth that the introduction of threads could bring many new effects to a thread-less system. These effects could be pos-

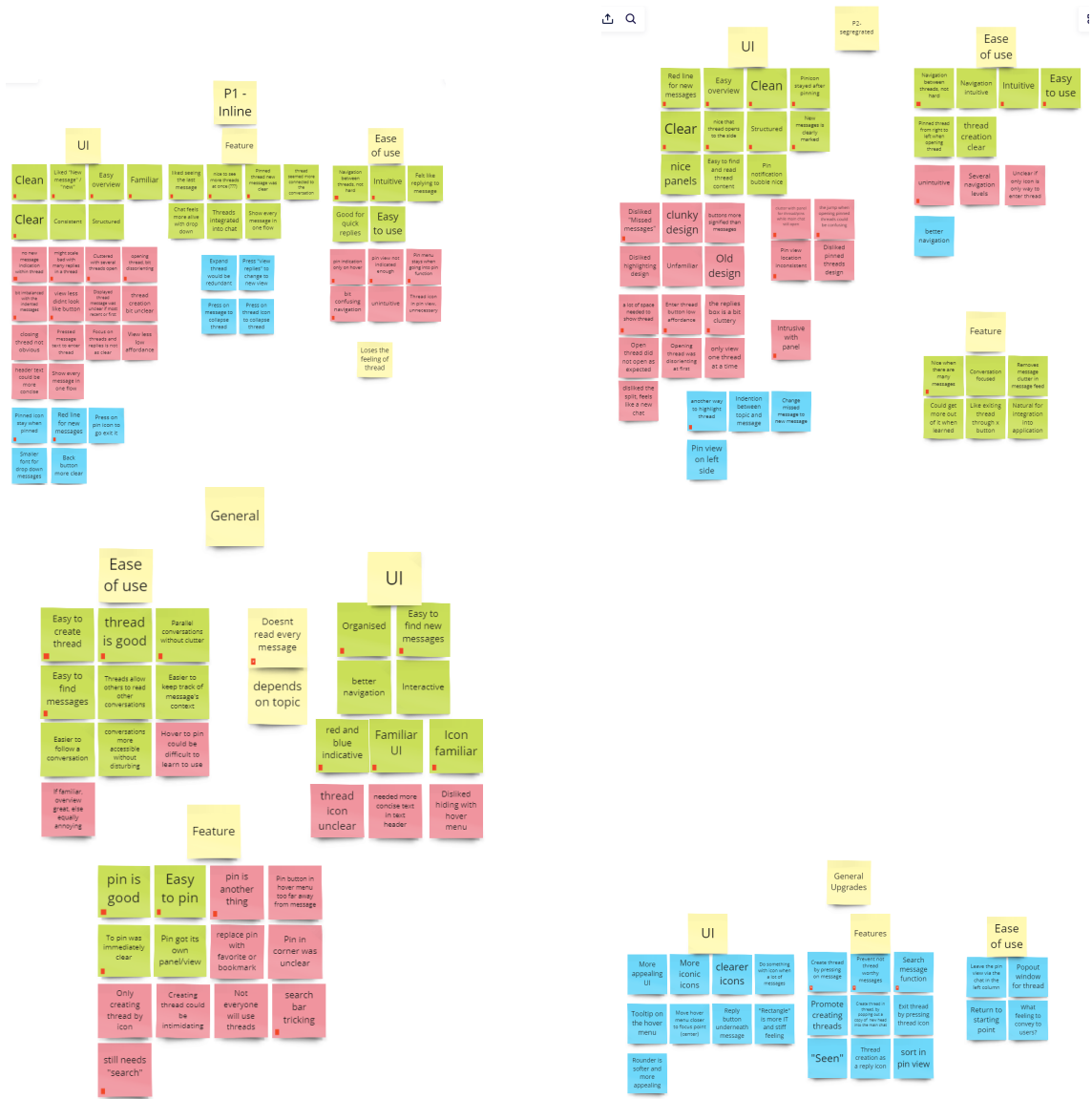


Figure 4.5: Affinity diagram from the usability evaluation in iteration 2. Larger pictures of prototype 1, prototype 2, general comments, and general upgrades can be viewed at B.8, B.9, B.10, and B.11 respectively.

Table 4.3: The scores of each participant during the second iteration of usability evaluation. Previous participants are in the repeat tables.

Internal - New	Prototype 1	Prototype 2
Participant 1	67.5	97.5
Participant 2	72.5	80
Participant 3	95	95
Participant 4	90	80

External - New	Prototype 1	Prototype 2
Participant 1	82.5	60
Participant 2	97.5	97.5
Participant 3	97.5	95
Participant 4	60	52.5
Participant 5	37.5	80
Participant 6	85	40
Participant 7	77.5	87.5
Participant 8	95	77.5
Participant 9	72.5	65
Participant 10	92.5	90

Internal - Repeat	Prototype 1	Prototype 2
Participant 1	62.5	100
Participant 2	80	57.5

External - Repeat	Prototype 1	Prototype 2
Participant 1	82.5	95
Participant 2	100	97.5
Participant 3	87.5	72.5
Participant 4	100	100

itive, negative, and/or neutral. As subjectivity plays a part, all three opinions could be held at the same time by different people regarding the same function. Still, there are trends that show what was generally liked by the users. During this iteration all the testers appreciated threads and felt like using them would make communication more organized and structured. Responding to a certain message or topic would be easier as it would be possible to write in direct connection to what you are responding to.

One benefit that might not be too obvious is having one-on-one conversations in public. During the interviews, it was mentioned that if a person needed help with something they might write directly to someone who can help them, so as to not spam the chat. Then this exchange would be held in private unnecessarily. Would it instead be held in a thread inside a group chat then others could take part in helping or finding help themselves, this would minimize the need for repetition and relieve some workload for the people that are helping others. Elaborating on this idea threads could potentially be helpful for future help-seekers as well, if someone asks for help they could be directed or referred to the appropriate thread.

A negative effect that was brought up is that it might be a bit jarring to create threads or have your message be a thread starter. It can feel more official or "important" as your message is not one in the log compared to just writing in a chat.

How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability?

The UI should provide enough visual information for the user to be able to intuitively complete their goal. Using icons that clearly show what affordances are being provided is necessary and makes usage more intuitive. For example a pin icon for a pin-function, both for the pinning part and the show pinned messages part. Using colors to indicate affordances or

news is necessary as well. It is a way to guide the user to their goal, using red to indicate new messages worked well in the tests. The red markers should be present at every step, first to indicate which chat, then the relevant thread, and lastly to show the specific messages.

The blue color used to highlight the number of replies in total there is in a thread worked well as it showed there was something unique here but did not grab the attention of the user excessively. Informative text that is pressable to do some function would benefit from some color coding or help icon. In the inline prototype, the way to close threads was not indicated enough even though the text explicitly said: "view less". Having it be a different color to nearby text or it being paired or even replaced with an icon could help. This icon could for example be an "X" or an arrow pointing backwards. If the color was switched it could be switched to the same blue similar to the number of replies. As the amount of reply texts would disappear when a thread is opened, there would be a vacancy in that color theme that the "view less" text could fill.

The UI needs to be clean and not excessively filled with icons and text. If there are many options or affordances in regards to a message, for example, pin, favorite, respond, edit, etc. The functions can be hidden to alleviate the clutter. This could be through having the signifiers be invisible until their message is being hovered over. They can be hidden within a sub-menu linked to their message. The overall appearance of the UI is important in regards to helping the user understand where they are. Highlighting icons, boxes, and headers during interaction makes the user more aware of what they are doing and what is happening. Text in the UI should be guiding and indicative while not being aggressive towards the user, for example using "new message" is better than "missed message". A user would probably understand what both of these texts are conveying, but the "missed message" was reportedly more aggressive and judging, making the user feel as if they were to blame.

It seems logical that the fewer clicks that are needed to perform a task the faster the task can be done, improving the efficiency. This was somewhat tested in this iteration as the number of clicks was kept as low as possible without affecting the navigation of the system. Something that can affect efficiency is the distance between where the user's attention is and where the system wants them to put it. In the prototypes, when a message or thread was hovered the buttons showed were quite far away and small. This worked as the testers managed their tasks without problems. But, being more in accord with Fitt's law would probably improve the efficiency, as it states that the bigger the goal and the shorter the distance is to said goal, the faster the user can reach it[2].

Moving users around between screens, windows, and views might make it unnecessarily hard and increase the risk of them becoming confused and getting lost. To keep efficiency up it is important to keep the user grounded in the UI. If a thread is opened the system should still show where in the chat log you are. If the thread is opened to the side or inline the user has not moved, instead the system has moved according to the user.

How does different communication systems affect the efficiency of the teams/people using them?

The inline approach seems to be more suited for smaller groups and out-of-office use. This is strengthened by the inline prototype getting the higher SUS score by the external group which can be seen in figure 4.4. This group was mostly students and small office workers. During the interviews it was described as familiar and intuitive, it was believed to be better

for shorter threads and being able to see all threads in one flow. An issue brought up is if the threads become too large showing it all inline might be too intrusive. This could be handled by having a thread be shown inline still but now in a scrollable box. Another way would be to show only a set amount of messages first and then allow the user to click a "view more" function to see further into the thread.

The segregated approach seems more suited for larger organizations and in-office use. This prototype had the higher SUS score among the internal testers, see figure 4.4. It was also said to be intuitive but to be better for long threads viewing them would not take up space within the chat log. This makes sense as the internal group would be more used to having up to hundreds of people in one chat and to have topics or conversations can be quite long. It might be lacking in regards to overview as you can only view one thread at a time. This could be rectified by having the ability to have threads in tabs or being able to "pop out" threads into their own window.

From the interviews during the test the overarching impression was that having a chat application where you can create threads on previous messages would allow for an increase in efficiency when introduced to an application without. Being able to structure and organize messages like this makes it easier to go backward to find a conversation, as it would be collected in the thread, instead of having to string messages together on your own. For example, if you were to reply to a question you would not have to type in the chat what question you are answering, nor would you have to do it in private, letting others see your answer.

Being able to save threads either through some pin or favorites system would further the potential for efficiency even more. Being able to quickly get back to a thread that started long ago would for starters save some scrolling time as you look for the thread. This would in turn promote the usage of long-lived threads. For example, your team has a recurring topic that is discussed in the chat. Instead of switching back and forth between this and other topics, the recurring one could be held in a thread. It would not distract from other topics and it would be easier to find the messages regarding it as well. Pinning or saving the thread would then increase its longevity.

Chapter 5

Discussion

In this chapter, there will be a discussion of the various parts of the thesis written prior to this chapter. It will first cover the overall process and general aspects of the thesis and what parts went well or could have been improved. After that, we will discuss the important results that are relevant from both iterations to answer the research questions. Future works will then be the closing part of this chapter as it covers future potential designs, features, and improvements.

5.1 General

In this part, we will discuss the overall aspects of the thesis that could have improved the work done during the thesis. It will also cover other aspects regarding the impact of threads when users do not make use of them.

5.1.1 Process

The process was overall great, with the pilot study in the beginning to evaluate the flaws and potential for improvements which then led to the idea-generating process, prototyping, and usability evaluation, which were repeated for both iterations. This user-centered process allowed the data to revolve around the user and not the developer or designer. Consequently, this caused the data to constantly focus on improving elements that users felt were lacking.

In iteration 2 we developed some code with the goal of creating one hi-fi prototype for the usability test to simulate the end product with threads. However, due to lack of time, we decided to dismiss the development of the hi-fi prototype through code and instead created two improved prototypes from the previous iteration. Within the scope of this thesis, this was infeasible, but something to consider is that testing the actual end product would yield more valuable and rich data than simple prototypes where the interaction is somewhat limited.

One aspect that should have been taken into consideration or done earlier is coming into contact with relevant teams. Examples of this would be when first going into the first

iteration, we also had a meeting with the UI/UX team at Telavox and received an overview of how they approach creating designs and prototypes. During this thesis around the start of iteration 2, we found out that progress had already been made regarding the implementation of threads. While they still were in the starting phase with only the design done, it could have eased the process and resulted in better data in iteration 1.

Meeting with the UI/UX team earlier could have possibly resulted in receiving the assets for prototyping that were later used in iteration 2, for iteration 1. The reason for this is that we did not know that there were any other assets that we could have access to and received assets that were limited to those used in iteration 1. We later found out that once we received our licenses for Sketch by Telavox, there were numerous more assets for us to use that was identical to the current Telavox chat application. This could have been a result of miscommunication or simply ignorance from both parties.

5.1.2 Embracing change

When the changes to an application are substantial, there is a risk that users may not even make use of the new additions to the application. One thing to possibly mitigate this is to introduce the changes slowly or bit by bit instead of a tremendous change at once. With the introduction of threads, the chat familiarity might shift to be more professional which might deter some users from making use of threads as they could be either intimidated to create threads or simply not familiar enough to make use of it and default to sending normal messages instead. In general, it is hard to force users to use a functionality if they do not embrace the change themselves.

5.2 Testing

Testing is comprised of many different parts and areas, and as it is vital during development, it is important that the different parts work well together and individually. In this section we will discuss how these parts could have been improved, were affected by circumstances, and why they were chosen.

5.2.1 Usability test

Something that can improve the testing of this project is some of the collected performance data. Being able to see the time of completion for each task would allow us to see more specifically potentially lacking features. This would also make it easier to see improvement between iterations. It could also be beneficial to modify some of the tasks for each prototype. The task where the testers were supposed to find a given message hidden in one of four threads, they had to pretend not to know where the message was. We wanted as little difference as possible between the tasks for the prototypes, but changing the text of each message or changing the sought after message could have affected the experience. Otherwise, the test was useful as it gave the tester enough understanding of the systems in order to give valuable feedback.

5.2.2 Integrity of the test

The tests of the second iteration were mostly conducted remotely, the testers used their own equipment and were in their own environment. This could affect the integrity of a test as the moderators would have less control of the surroundings, and the testers could be distracted by things off-screen. Overall testers got noticeably distracted only a couple of times in total and those that did were able to go back to the task at hand without problem. This probably did not have a significant effect on our testing, but it definitely is worth having in mind for future projects.

Speaking of things that can affect the integrity of the test is the moderator itself. When do you give hints or ask questions without steering and putting too much pressure on the tester. During our testing we intervened when there were technical issues with the equipment or the prototype. We also intervened if they reached the goal of a task without realising it, then we would give them a leading hint or question. And we asked questions when there was a need for clarification about something they had done or mentioned that they were thinking. This approach felt like a good balance as we did not interrupt the testers during an interaction unless necessary for the test to work. And we were able to get deeper insight into their thought process.

Mistakes and adversity

Every human makes mistakes, even a moderator of a test is capable of this. Stumbling on one's words or accidentally saying the wrong task can easily happen. These mistakes can be quickly rectified during the tasks.

Some adversity cannot easily be predicted or avoided, during the testing period of the first iteration one of us caught a cold. This led to the other conducting the tests alone. In order to adapt to a situation like this it is very helpful to have a detailed test plan that is easy to follow.

Even with a detailed test plan, there is room for mistakes, one person was asked to start with the wrong prototype. This can simply be fixed by reversing the order of another tester in the same group. However, this was not caught until it was too late. But as it was during the first iteration and because of the fact it was just two people, this would not have a significant effect.

5.2.3 Subjective quantifiable data

Different surveys

To get subjective quantifiable data from the users during the tests in this project SUS forms were used. There are alternatives to SUS that are worth keeping in mind. Two others are *Usability Metric for user Experience-Lite Version* (UMUX-Lite) and *User Experience Questionnaire-Short Version* (UEQ-S), these were compared in the article by Schrepp, Kollmorgen, and Thomaschewski [22]. They explained how all three try to measure the usability or pragmatic quality of a product but that they differ in what they cover. SUS focuses only on the usability and ease of use. UMUX-Lite also focuses on usability but is not as detailed as SUS. However, it brings a unique factor in the sense it also covers the usefulness of the tested system.

UEQ-Ss is more detailed than UMUX-Lite but not as detailed as SUS. It does however cover how fun it is to use a system, making it unique among these three.

They conclude from their result that UMUX-Lite can be a substitute for SUS if just an overall UX quality indication is needed. And if you want to rank different products based on their usability or UX quality there is no big difference between the surveys. However, SUS still gives more information through the 10 questions which allows for finding more detailed or specific issues.

As we going to have two prototypes "compete" with each other it would on a surface level have sufficed with either of the surveys. But as SUS could provide more detailed information about the different aspects of the prototypes this was the survey chosen.

Affinity diagrams

One could argue that affinity diagrams are in a grey zone as the things written on the notes are subjective, but you are then able to count how many held what opinions. This would somewhat quantify the result though it would still be more qualitative. This approach was used during the project and allowed us to find items that users thought were weaknesses and strengths in our prototypes. And it helped us to differentiate between an individual's complaint and a large scale problem.

5.2.4 Sample Sizes

Usability evaluation is an important part of testing in order to find results. One thing that has the potential to affect the credibility of these results is the number of participants.

First iteration

In our case, the first iteration had a total of 10 participants. This has the potential to be too few participants to show any actual trends. As the scope of this project is small due to time and manpower limitations and because we still received very useful feedback, the number of participants was deemed to be adequate.

The distribution between internal and external was not equal in this iteration. It was skewed in favor of external which represented six out of ten participants. However, this was not seen as an issue since this iteration was mostly to get feedback on the prototypes on general improvements.

Second iteration

The second iteration had 20 participants which made it easier to see trends and this gave the result more credibility.

The distribution this time was even more skewed in favor of the external group, representing 14 out of 20 testers. It may be preferable to have an equal split between these two main groups. But we still managed to see distinct trends between them which could be used.

5.2.5 Finding participants

To find internal participants we posted messages in chats hosting around 310 people in total. Naturally, many other messages things were being said and posted. And as we found out during the tests we cannot assume that all of the people in the chat rooms read everything meticulously, especially in a room meant for topics unrelated to work. Meaning that some might simply have missed the posts related to the interviews and tests. The external participants were found by writing in group chats consisting of friends and acquaintances.

The external was easier to find possible due to our relation to them and that they wanted to help us out. In hindsight, we could have put more effort when marketing the tests for the internal people. But as it was during the summer and many were on vacation and since we still received useful feedback for our result, we thought the distribution and amount sufficed. It is however worth keeping in mind how to achieve an equally distribution and if it is important, for future projects.

5.3 Research Questions

This sections discusses how the research questions are answered by comparing the analyses of iteration one (3.5) and two (4.5).

What effects do chat threads have on text communication?

The overall consensus from both iterations is that threads make text communication better in the sense that messages become more organized and structured when compared to having no threads. It helps the user to distinguish what messages are responding to which conversations and differentiate between separate conversation topics. Threads also ease user replies to certain messages as the user creates a direct connection of their reply to the message being responded to. Additionally, threads encourage conversations that would otherwise occur privately between two users to be done in public chat rooms and allow for other people to participate in the conversation or make use of the information exchanged in the conversation. This minimizes the repetition of information and also alleviates the workload of people helping others as they could refer to specific conversations.

Threads do not introduce any redundancy, but integrating it into a chat application would inevitably do so as another functionality is introduced which in turn would clutter the user interface. However, threads compensate for this by reducing the clutter by moving messages within threads and cleaning up the user interface. Instead, creating threads could be intimidating to the user as threads could make it feel more official or important.

How do you effectively design/implement threads to make it satisfactory according to ISO 9241-11:2018 about usability?

Threads should minimize redirections as much as possible in order to ground the user and not make them confused about where they currently are in the application. Show the user where they currently are and move the system around the user, not the user around the system. A good overview would also give the user a better understanding of the current state of

the program. Consequently, this also has an indirect effect that there would be fewer clicks which would then improve the efficiency of the navigation as it would be overall faster. It could possibly also encourage more usage of threads in general as it would be straightforward and satisfying to the user. While threads should minimize the number of redirections, it should not be disregarded to add additional navigation for the user where it is convenient. The design itself should then be cautious not to introduce too much freedom as it could be confusing.

Achieving good navigation could also be through a method of saving threads or displaying threads the user has participated in. One method is to have a side panel where the user can navigate between threads while still reading the one they have already entered. Another way to do this is by allowing the user to save threads which would allow them to receive updates from old saved threads that are relevant and somewhat counteract obscurity. It would possibly also encourage users to use threads more when it is simple and intuitive to create them. An avenue to explore this aspect is to create potential constraints, where an example would be to automatically create a thread when replying to a message.

The user interface should also be taken into consideration when creating threads to be satisfying. By not having it overly cluttered with buttons and icons, it would allow for greater navigation. However, when there is a need for many buttons with different functionality and icons, they should be subtle and not overbearing which could for example be done with them being hidden until the user hovers over messages to show them. In that case, the distance between the buttons and icons should not be too far from the user's focus (roughly around the center of the screen) as it would make it difficult to find and use.

The user interface itself should introduce a familiar design to make it more intuitive to the user and also show a clear relation between a reply and a topic message. Furthermore, the icons should have good affordances to let the user know what it does. To highlight certain features that are based on text, changing the color of said text is a good method to indicate affordances. This can be seen when replies are in blue text and red text refers to new messages and if there are no text with no text, an icon could help with affordances instead. Even then, icons should be clean and not excessively used. Red markers should also be used at every step when a user receives updates of new messages so it indicates which chat the messages are in, which thread, and then the specific messages. Lastly, the text should also be guiding and indicative. This was shown when "missed messages" seemed judgemental or aggressive while "new messages" were more guiding and indicative which was preferred.

How does different communication systems affect the efficiency of the teams/people using them?

In both iterations, participants conveyed that threads would improve efficiency in comparison to having a chat application with none. It would also affect the communication in a chat application in a positive way as it would allow for easier communication and readability. Furthermore, similar to the answer in RQ1, it would encourage fewer private messages between two users and instead have the conversation public for everyone else to participate in. Threads also make it easier for users to determine what conversation the message is responding to making it easier to read through. It also allows people to direct others to certain conversations when they are searching for something specific that has been mentioned before. This is possible when messages are structured and organized. The user should also be

able to see the connection between messages without having to do it themselves. Additionally, having a way to save/bookmark threads would also potentially increase the efficiency as it promotes the usage of older threads.

In terms of the two communication systems that were created from the prototypes, inline and segregated, inline seemed to be more suited for smaller groups and out-of-office use while segregated seemed to be better for larger organizations and in-office use. Both were deemed intuitive by users, but inline seemed more familiar than segregated. This could ease the learning curve for inline while segregated would have a higher one. Something to note is that inline seemed to better fit when there are shorter threads and would like to see all threads in a flow, while segregated was for longer threads which could be more appropriate in a workplace environment as more detailed messages are hypothetically more needed than in everyday life.

5.4 Future Work

One functionality that would be worth considering is giving users the ability to move messages that are outside threads into them. This could be done by having another button in the hover menu that the create thread button is in and potentially adding another interaction by dragging and dropping the message into the thread. This would allow the user to correct mistakes and fulfill the error prevention as well as give the user freedom to use the product.

Other functionalities should not be disregarded as they should also be integrated together with threads to offer intuitive navigation and use. Examples of an additional feature could be reactions, which were deemed as a feature that some participants in our study as beneficial. Both themes of the prototypes that were tested, had some kind of feature beneath the topic message. Reactions then have to be displayed in a way that is adapted to the threads themselves, or threads could instead be adapted to reactions. One thing to consider then when implementing threads is how to design the affordances of it to accommodate other features while not compromising itself.

A way to improve the flexibility and efficiency of thread usage that was is to allow popping out a given thread into its own window. This would make it possible for the user to have multiple threads visible at the same time. This kind of feature would most likely be a more natural addition to the segregated version of threads than with inline as there already is an element of separation, which may make this less jarring.

The pin icon in the prototypes should be replaced with a star or a bookmark, to make it into a favorited messages function. It was brought up in the interviews that pinned threads or messages should not be individual and specific to the user but instead individual in terms of chat, they should be "pinned to the top". Adding on the proper pin function would be possible then as well, having both would most likely work if the visual difference is big enough. The pin or save function in the final prototypes was intuitive and all testers liked the idea of being able to save threads in this manner and the function should be kept.

There might be something to be said about the automatic saving of a thread that was used in the inline prototype during iteration 1. If implemented now it could be that a thread gets automatically saved for a user when they respond in a thread, or when thread is created from a message they have sent. This approach might however become an nuisance for the user if a lot of threads are saved that they have no wish for saving and forget to unsave. But using

this approach as the standard would likely lead to less obscuring of information as the users would get updated about threads that otherwise might be forgotten.

Chapter 6

Conclusion

The aim of this thesis was to find the effects of threads in a chat application and ways to implement them. This was done by exploring different ways of implementation through a user-centered design process.

The findings during testing suggest that using threads would make text communication more organized and structured. Consequently, it makes it easier to differentiate between two conversations or topics and easier to find the relation between which conversation a message is responding to. One-on-one conversations regarding work would also be encouraged to be held in public chat rooms which allows others to participate in the conversation.

The design of threads should be intuitive with well-placed signifiers and affordances. Together with clear indicators, both in regards to notifications and icons, but also highlighting selected elements, to improve the user's sense of location in the system. Another way to increase the sense of location is for the navigation to be minimal and grounded, only redirecting the user when explicitly needed and in clear ways as it can otherwise be confusing. The user interface should be somewhat familiar and not cluttered which can be done by hiding needed functionality at all times in a hover menu.

Lastly, depending on the focus group, they might prefer segregated design over inline and vice versa. Both designs improved efficiency and also lessened the workload of helpers as they could direct to older threads/conversations that have been saved.

Future research in regards to this could entail some form of performance data, with larger sample sizes in order to see if there is a significant time difference between the usage of different systems. There are also potential benefits from testing other functionalities as well. Such as being able to pop out threads into their own window and being able to both bookmark and pin threads. This would determine what degree of flexibility can be achieved regarding how well both new and experienced users are catered to.

Bibliography

- [1] J. Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.
- [2] R. Budiu. Fitts's law and its applications in ux. <https://www.nngroup.com/articles/fitts-law/>, 2022.
- [3] Figma. Creative tools meet the internet. [Online]. Available from: <https://www.figma.com/about/>. Accessed: 7th June, 2023.
- [4] A. Fontana and J. H. Frey. The interview: From neutral stance to political involvement. *The Sage Handbook of Qualitative Research, Thousand Oaks, Sage*, pages 695–727, 2005.
- [5] E. Gilbertsson and J. Malmström. The Impact of Instant Messaging Features on a Collaboration Platform's Usability. [Online]. Available from: <https://lup.lub.lu.se/student-papers/search/publication/9108789>, 2023. Master Thesis. Accessed: 3rd September, 2023.
- [6] Google. Present during a video meeting. [Online]. Available from: <https://support.google.com/meet/answer/9308856?hl=en&co=GENIE.Platform%3DDesktop>. Accessed: 20th September, 2023.
- [7] Google. Record a video meeting. [Online]. Available from: https://support.google.com/meet/answer/9308681?hl=en&ref_topic=13802358&sjid=6598435464060703836-EU. Accessed: 31st August, 2023.
- [8] E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano, and C. Kamm. The character, functions, and styles of instant messaging in the workplace. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 11–20, 2002.
- [9] ISO. Iso 9241-11:2018(en) ergonomics of human-system interaction — part 11: Usability: Definitions and concepts. [Online]. Available from: <https://www.iso.org/obp/ui/en/#iso:std:iso:9241:-11:ed-2:v1:en>. Accessed: 20th September, 2023.

-
- [10] F. Lau and A. Holbrook. *Handbook of eHealth Evaluation: An Evidence-based Approach [Internet]*, chapter 10. University of Victoria, 2017. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK481584/>. Accessed: 20th September, 2023.
- [11] A. Lucero. Using affinity diagrams to evaluate interactive prototypes. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, pages 231–248, Cham, 2015. Springer International Publishing.
- [12] Miro. The leading visual collaboration platform. [Online]. Available from: <https://miro.com/about/>. Accessed: 7th June, 2023.
- [13] J. Nielsen. 10 usability heuristics for user interface design. [Online]. Available from: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Accessed: 3rd September, 2023.
- [14] D. A. Norman. *The Design of Everyday Things: Revised and Expanded Edition*. y Basic Books, 2013.
- [15] K. Paerata. The use of workplace instant messaging since covid-19. *Telematics and Informatics Reports*, 2023.
- [16] C. Pickvance. The four varieties of comparative analysis: the case of environmental regulation. 2005.
- [17] J. Preece, H. Sharp, and Y. Rogers. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2015.
- [18] J. Sauro. Measuring usability with the system usability scale (sus). [Online]. Available from: <https://measuringu.com/sus/>. Accessed: 12th July, 2023.
- [19] United Nations. Transforming our world: the 2030 agenda for sustainable development. [Online]. Available from: <https://sdgs.un.org/2030agenda>. Accessed: 9th March, 2023.
- [20] B. J. Roddy and H. Epelman-Wang. Interface issues in text based chat rooms. *ACM SIGCHI Bulletin*, 30(2):119–123, 1998.
- [21] J. Rubin and D. Chisnell. *Handbook of usability testing : how to plan, design, and conduct effective tests*. Wiley Publ., Indianapolis, Ind., 2008.
- [22] M. Schrepp, J. Kollmorgen, and J. Thomaschewski. A comparison of sus, umux-lite, and ueq-s. *J. User Exper.*, 18(2):86–104, jun 2023.
- [23] H. Shen. Integrating topic-centric thread-based organized chat into web-based enterprise instant messengers. In *2010 IEEE 7th International Conference on E-Business Engineering*, pages 442–447, 2010.
- [24] Sketch. Sketch is where great design happens. [Online]. Available from: <https://www.sketch.com/>. Accessed: 7th June, 2023.
- [25] Telavox. How it works. [Online]. Available from: <https://www.telavox.com/how-it-works/>. Accessed: 7th June, 2023.
-

Appendices

Appendix A

Questionnaires and Questions

Swedish:

- Vad för funktion/funktioner, som organiserar meddelanden, önskar du att ha i den interna chatt-appen (desktop version)?
- Är du bekant med chatt-trådar?
 - Hade du kunna berätta för oss vad det är?
 - Hade du kunnat berätta vart du har stött på chatt-trådar innan?
- Tycker du att chatt-trådar är värt att ha i en chatt-app?
 - Varför?
 - Vilka potentiella fördelar skulle trådar kunna ge jämfört med att inte ha trådar?
 - Vad för risker ser du med trådar?
 - Hur skulle de kunna motverkas?
- Hur tror du att trådar kommer påverka ditt arbete i gruppsammanhang?
- I Telavox-appen, hur tror du trådar hade påverkat kommunikationen i publika rum?
- Hur ser en chattråd ut enligt dig? (1, 3, 5)
 - *Visa telavox appen där de får förklara*

English:

- What function/functions, that organizes messages, do you wish to have in the internal chat-app (desktop version)?
- Are you familiar with chat threads?
 - Could you explain to us what chat threads are?
 - Where have you seen chat threads before?
- Do you think that chat threads are worth it to have in a chat app?
 - Why?
 - What potential advantages do you think that threads have?
 - What risks do you see with threads?
 - How could they be mitigated?
- How do you think threads would affect your work in group context?
- In the Telavox app, how do you think threads would affect communication in public rooms?
- How does a chat thread look according to you?
 - *Show the Telavox app so they could explain*

Figure A.1: The questions asked to the interviewees during the pilot study.

TELAVOX

SUS Formulär

le5116tr-s@student.lu.se [Switch account](#)

Not shared

* Indicates required question

Have you filled this questionnaire before? *

Yes

No

Figure A.2: The first question in the survey for the usability evaluation

Background

Age *

18-23

24-29

30-34

35-39

40-44

45-49

50-54

55-59

60-64

65 and above

Figure A.3: The second question in the survey for the usability evaluation

Gender

Man

Woman

Prefer not to say

Other: _____

Figure A.4: The third question in the survey for the usability evaluation

What is your current occupation? *

Your answer

Figure A.5: The fourth question in the survey for the usability evaluation

How often do you use the Telavox chat app? *

- Never
- Very infrequently
- Somewhat infrequently
- Occasionally
- Somewhat frequently
- Very frequently

Figure A.6: The fifth question in the survey for the usability evaluation

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

Figure A.7: SUS Questionnaire and its questions and score choices.

Appendix B

Affinity Diagrams

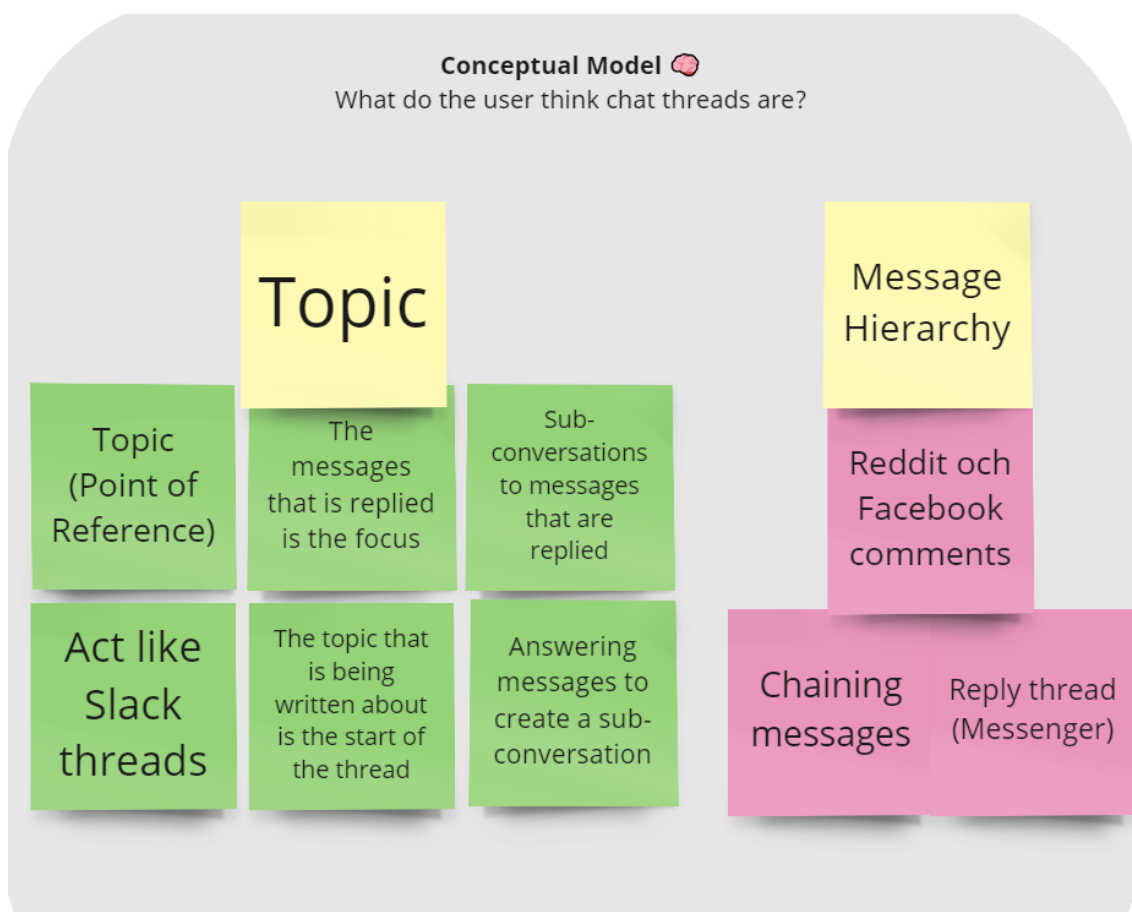


Figure B.1: Affinity diagram of the result from the pilot study regarding the participants' conceptual model.

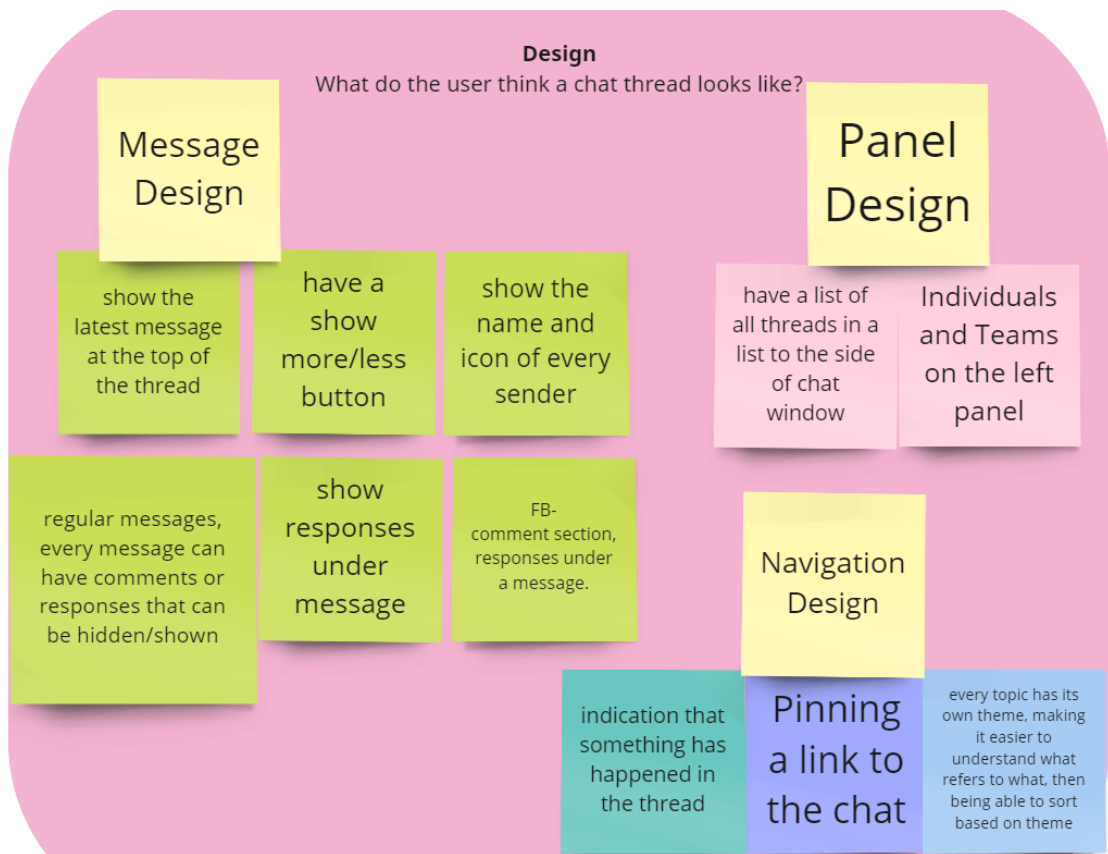


Figure B.2: Affinity diagram of the result from the pilot study regarding the participants' view on a thread design.

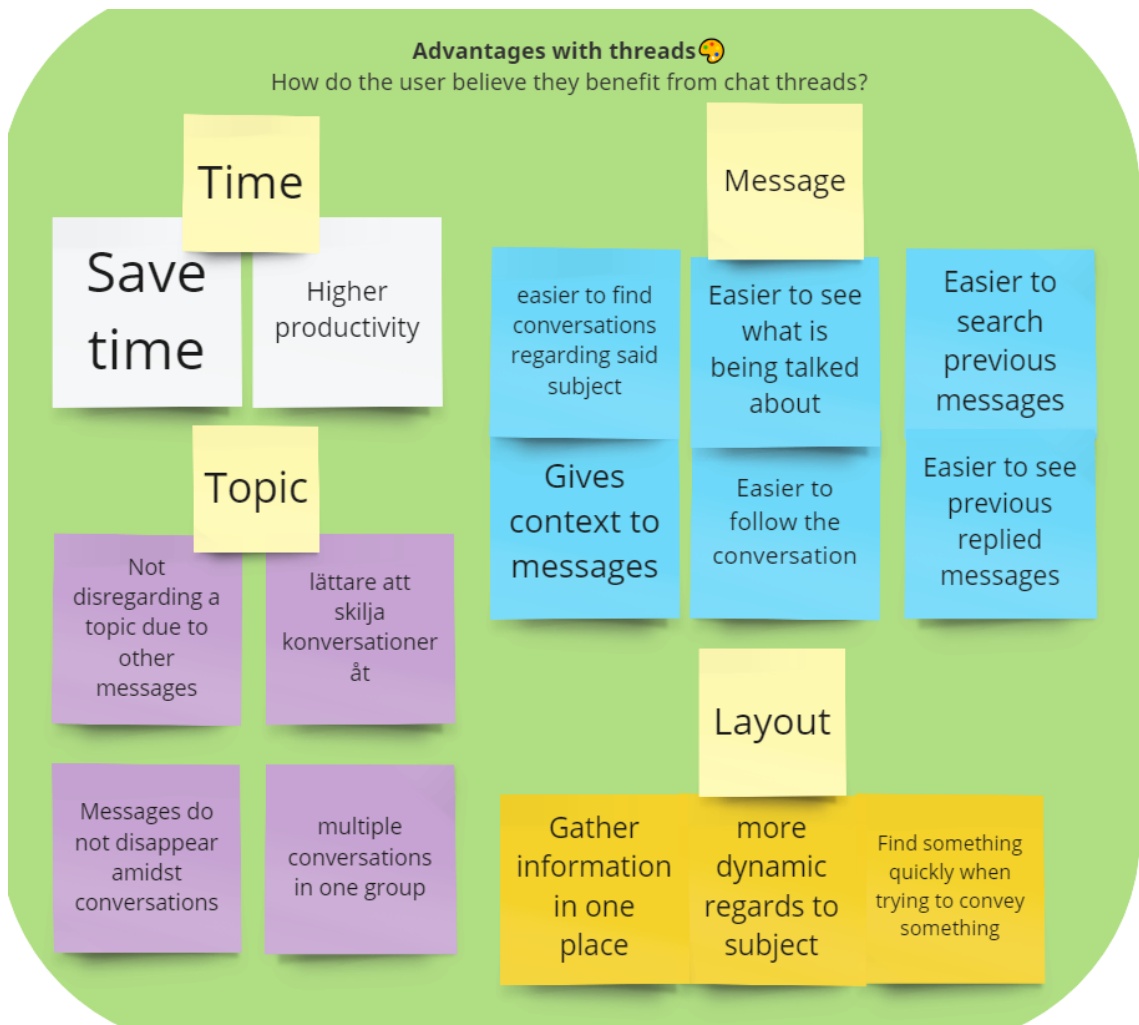


Figure B.3: Affinity diagram of the result from the pilot study regarding the participants' perceived advantages.

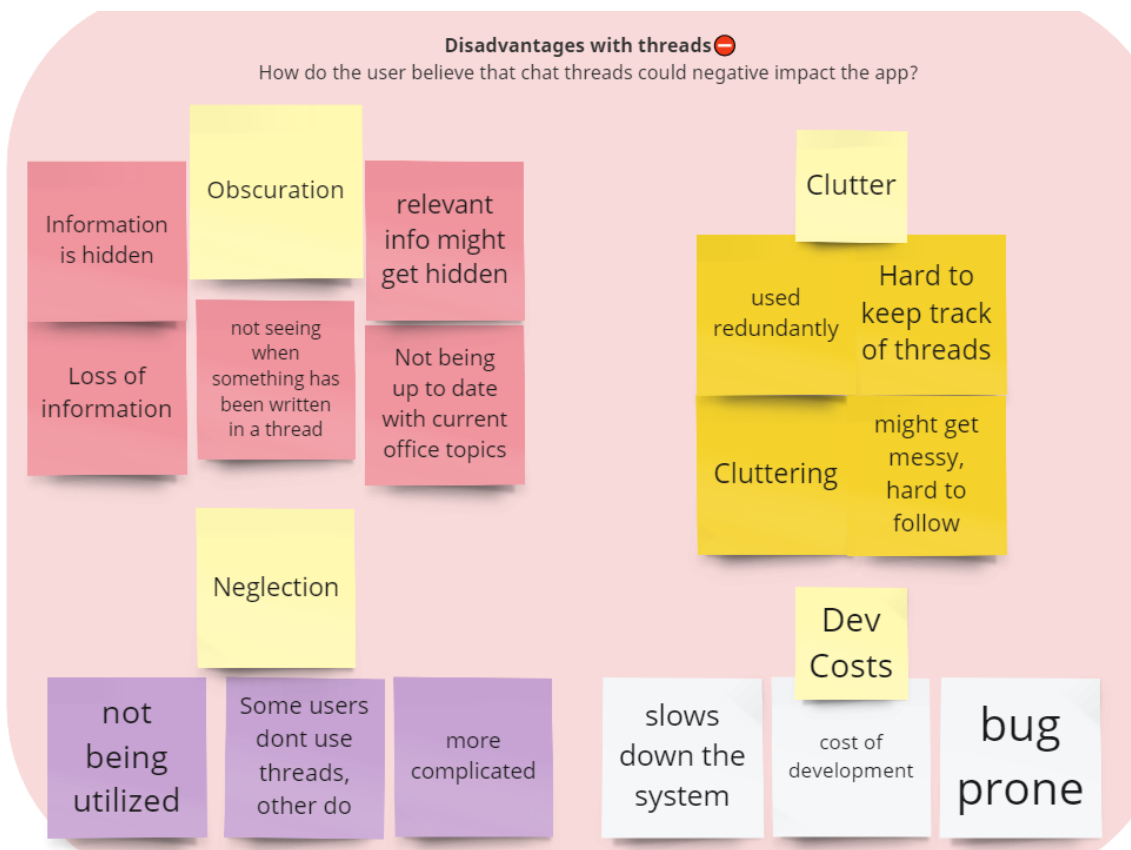


Figure B.4: Affinity diagram of the result from the pilot study regarding the participants' perceived disadvantages.

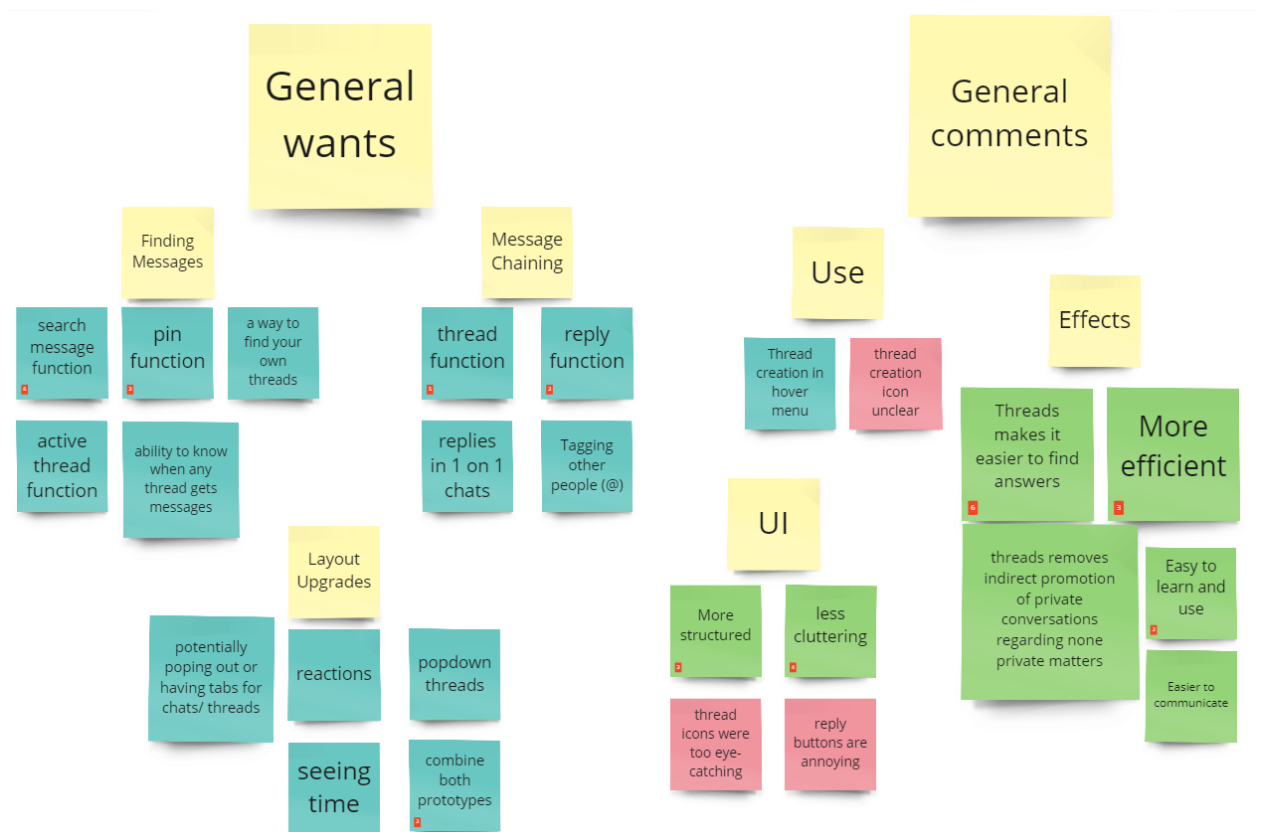


Figure B.5: The keywords and sticky notes from the results regarding general comments i.e. both prototypes in iteration 1.

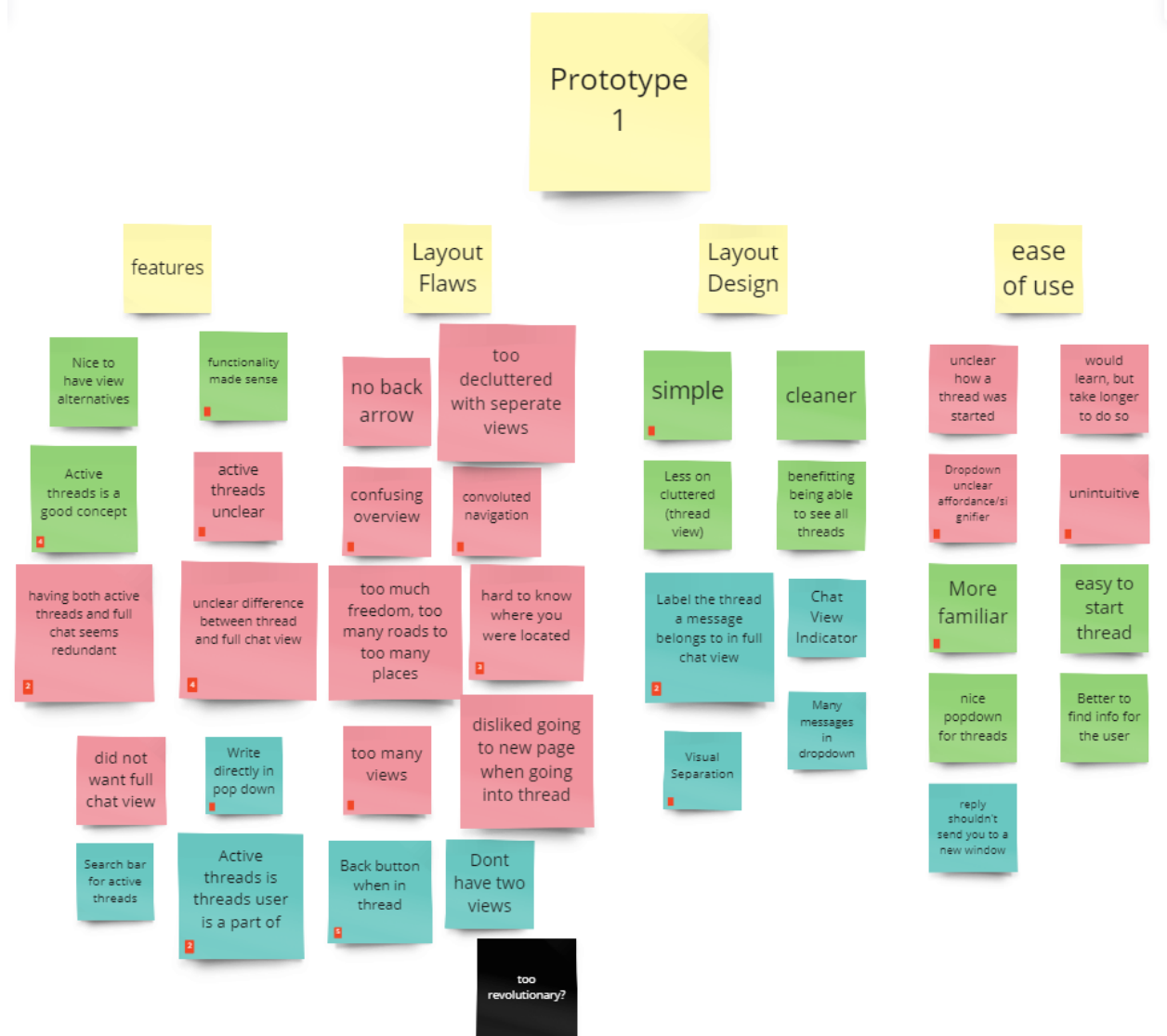


Figure B.6: The keywords and sticky notes from the results regarding prototype 1 in iteration 1.

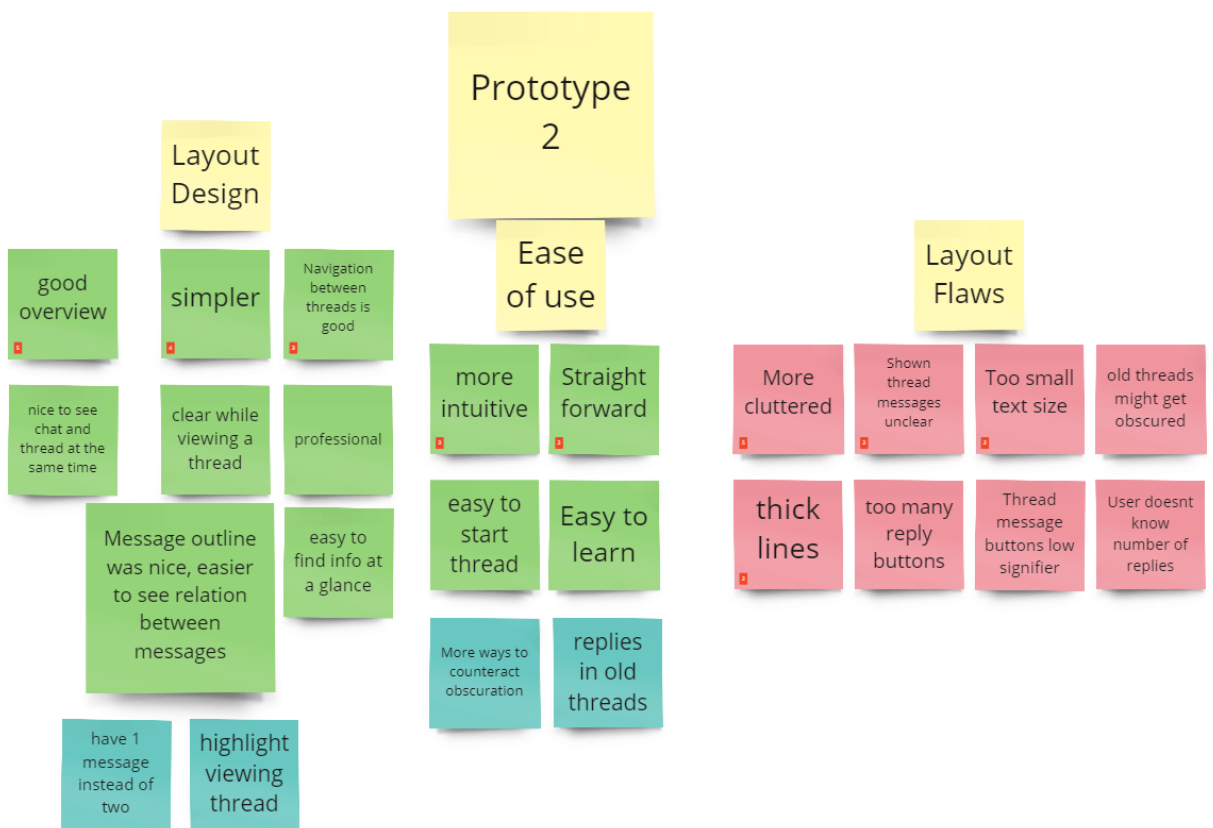


Figure B.7: The keywords and sticky notes from the results regarding prototype 2 in iteration 1.



Figure B.8: The affinity diagram of the inline prototype generated from the usability evaluation in iteration 2.



Figure B.9: The affinity diagram of the segregated prototype generated from the usability evaluation in iteration 2.



Figure B.10: The affinity diagram of the general comments from the usability evaluation in iteration 2.

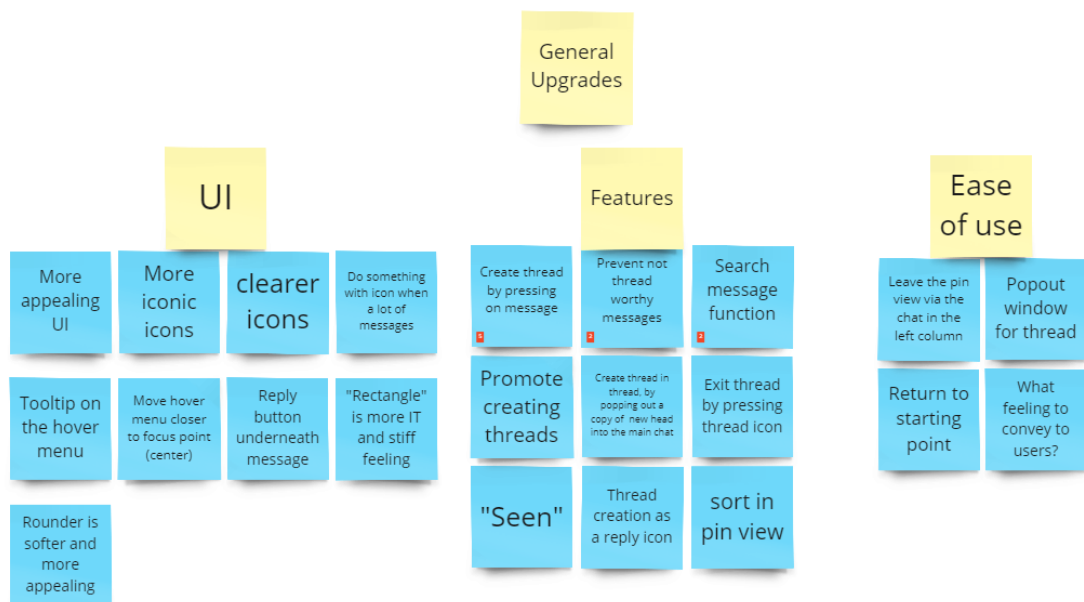


Figure B.11: The affinity diagram of the general upgrades from the usability evaluation in iteration 2.

Appendix C

Prototypes

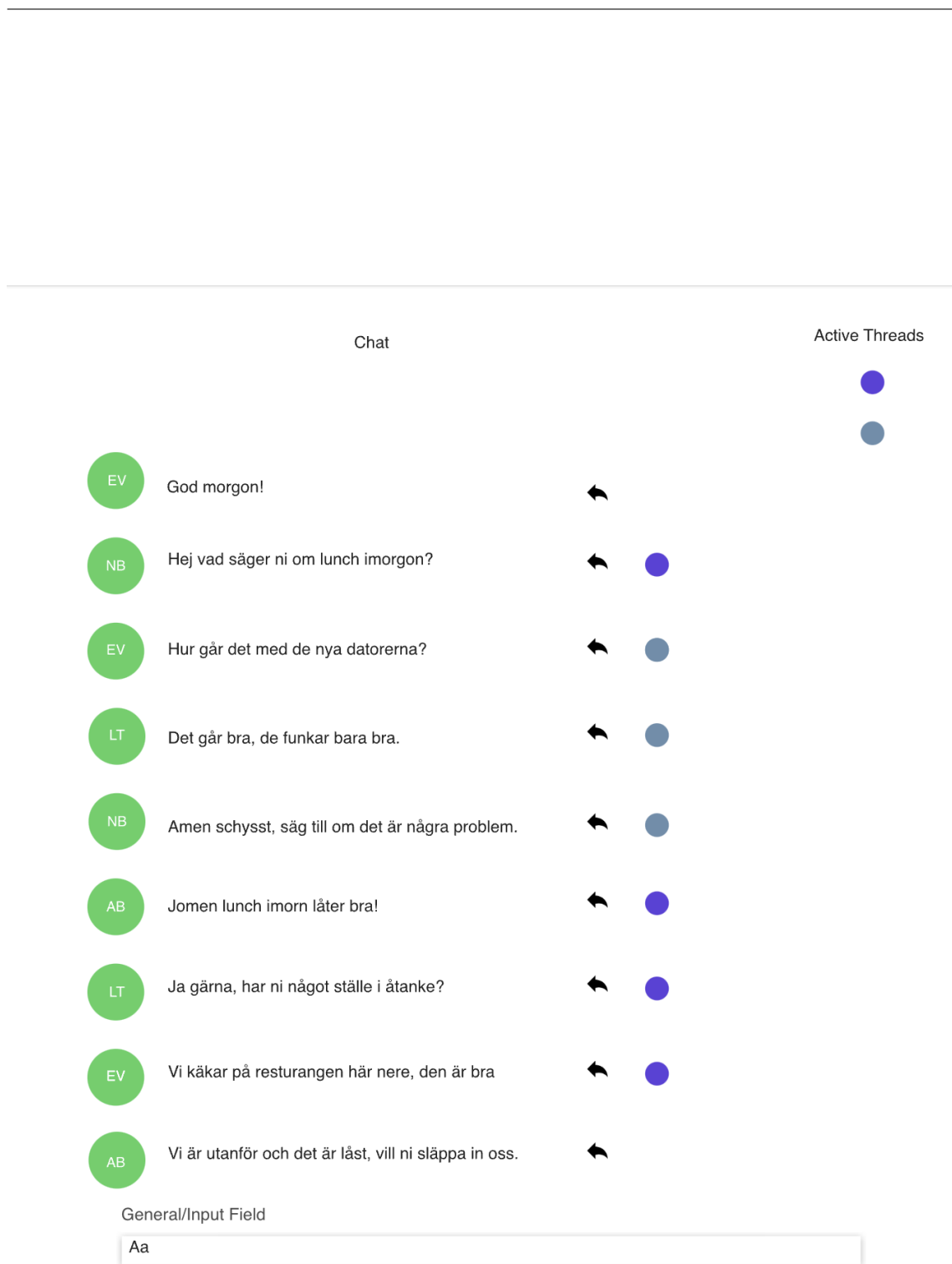


Figure C.1: Prototype 3, the color design in the overview.

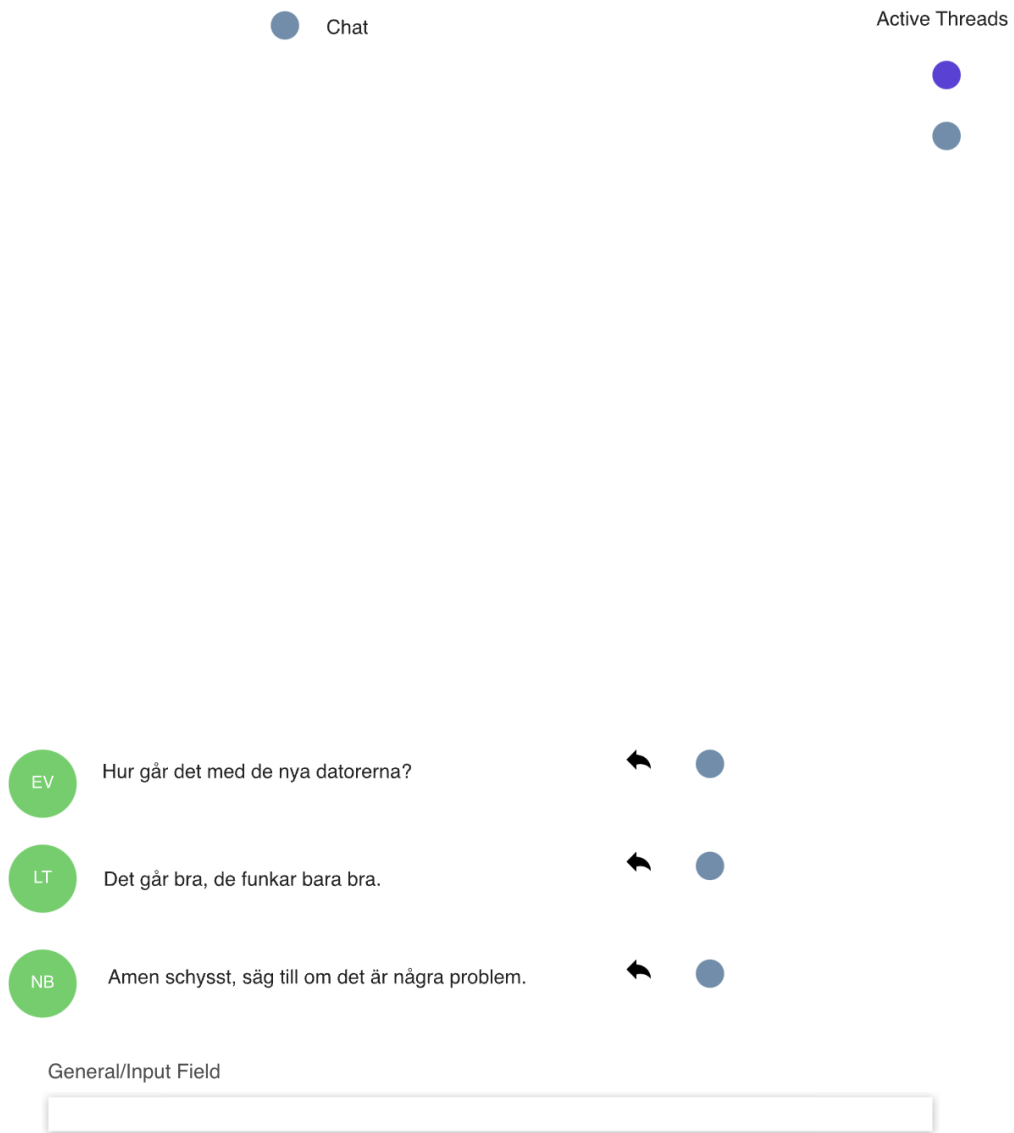


Figure C.2: Prototype 3, the color design when filtering and viewing the blue thread messages.

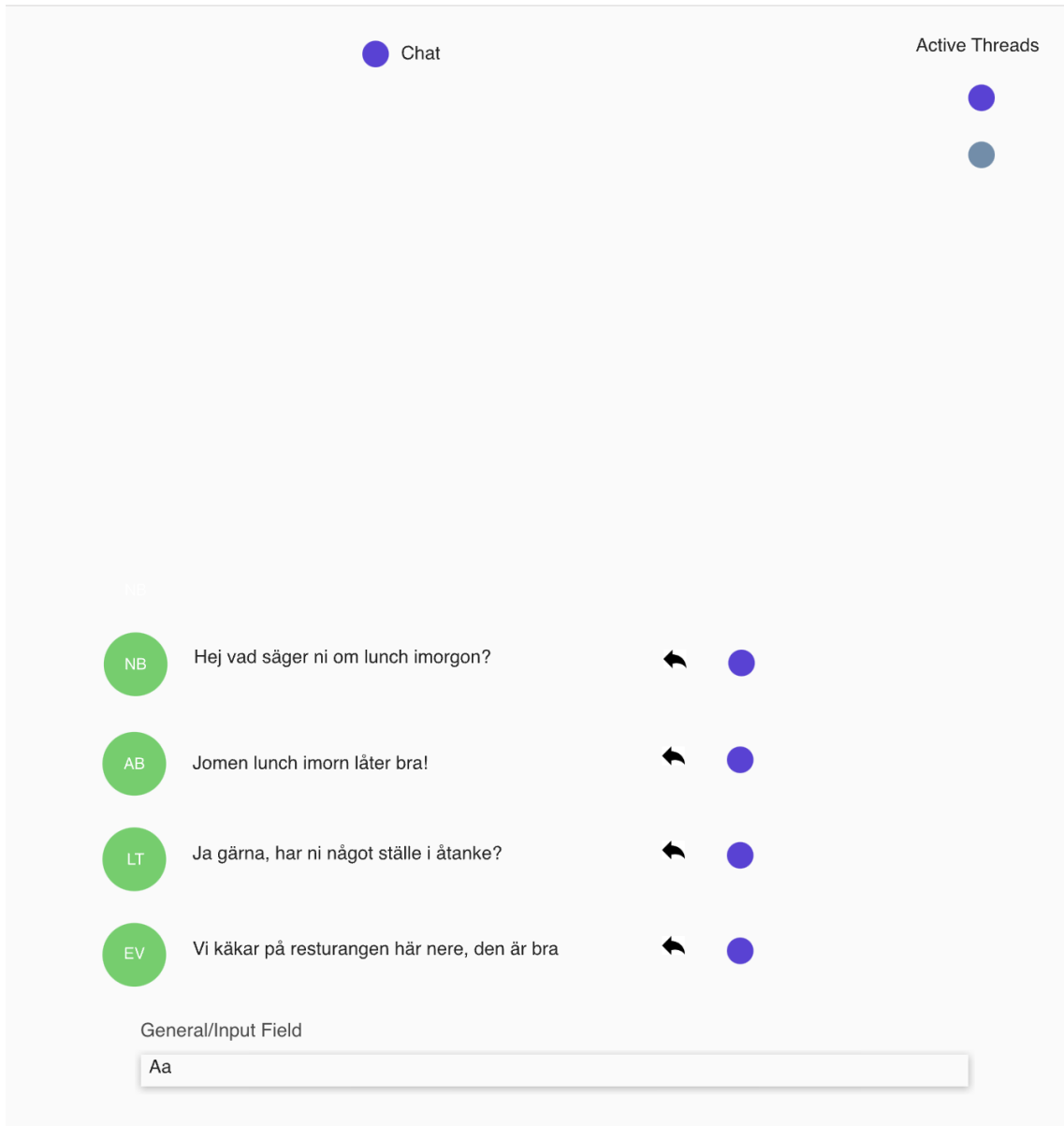


Figure C.3: Prototype 3, the color design when filtering and viewing the purple thread messages.

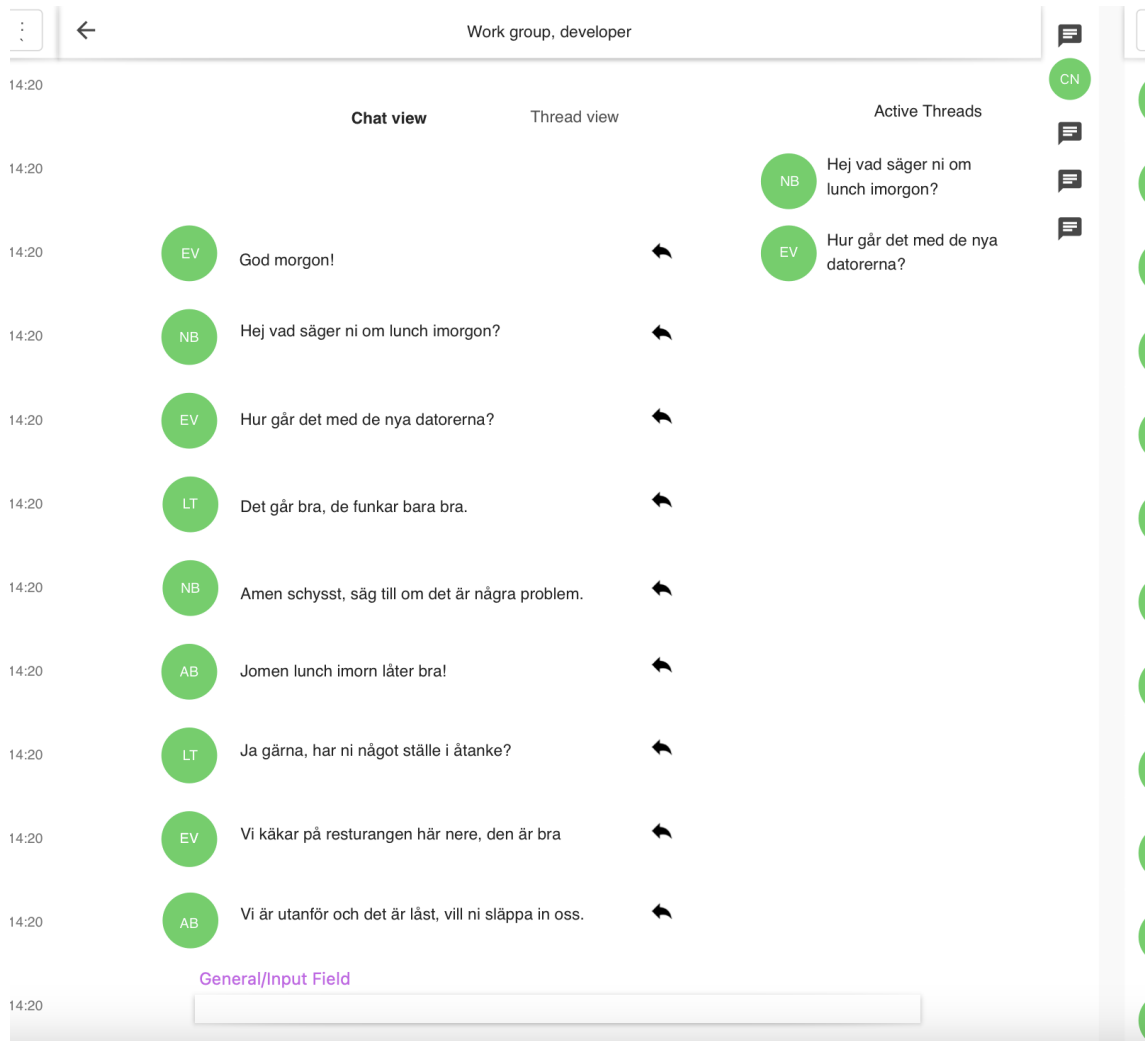


Figure C.4: Prototype 4, the inline design when inside the chat view.

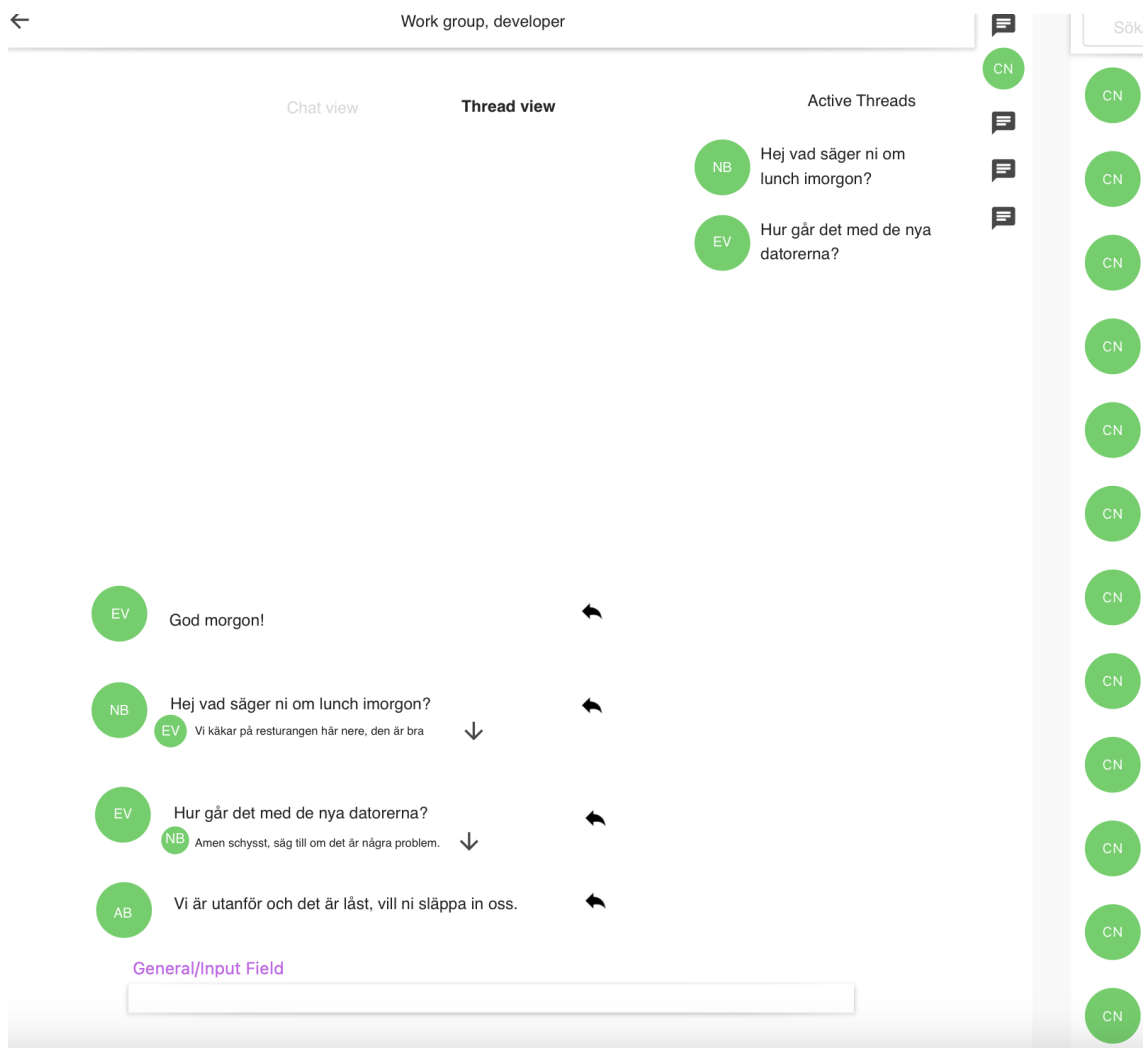


Figure C.5: Prototype 4, the inline design when inside the thread view with no threads open.

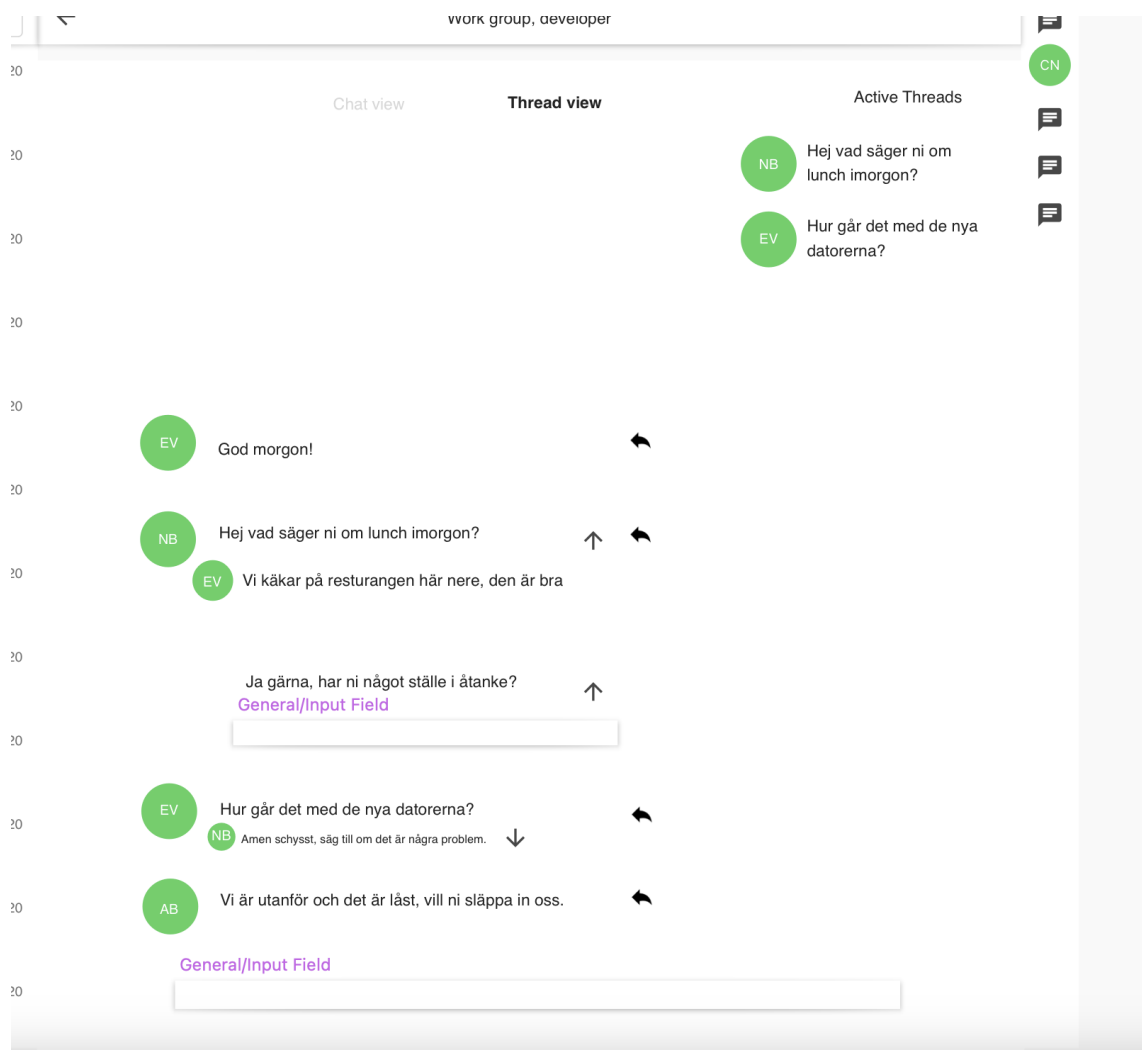


Figure C.6: Prototype 4, the inline design when inside the thread view with threads open.

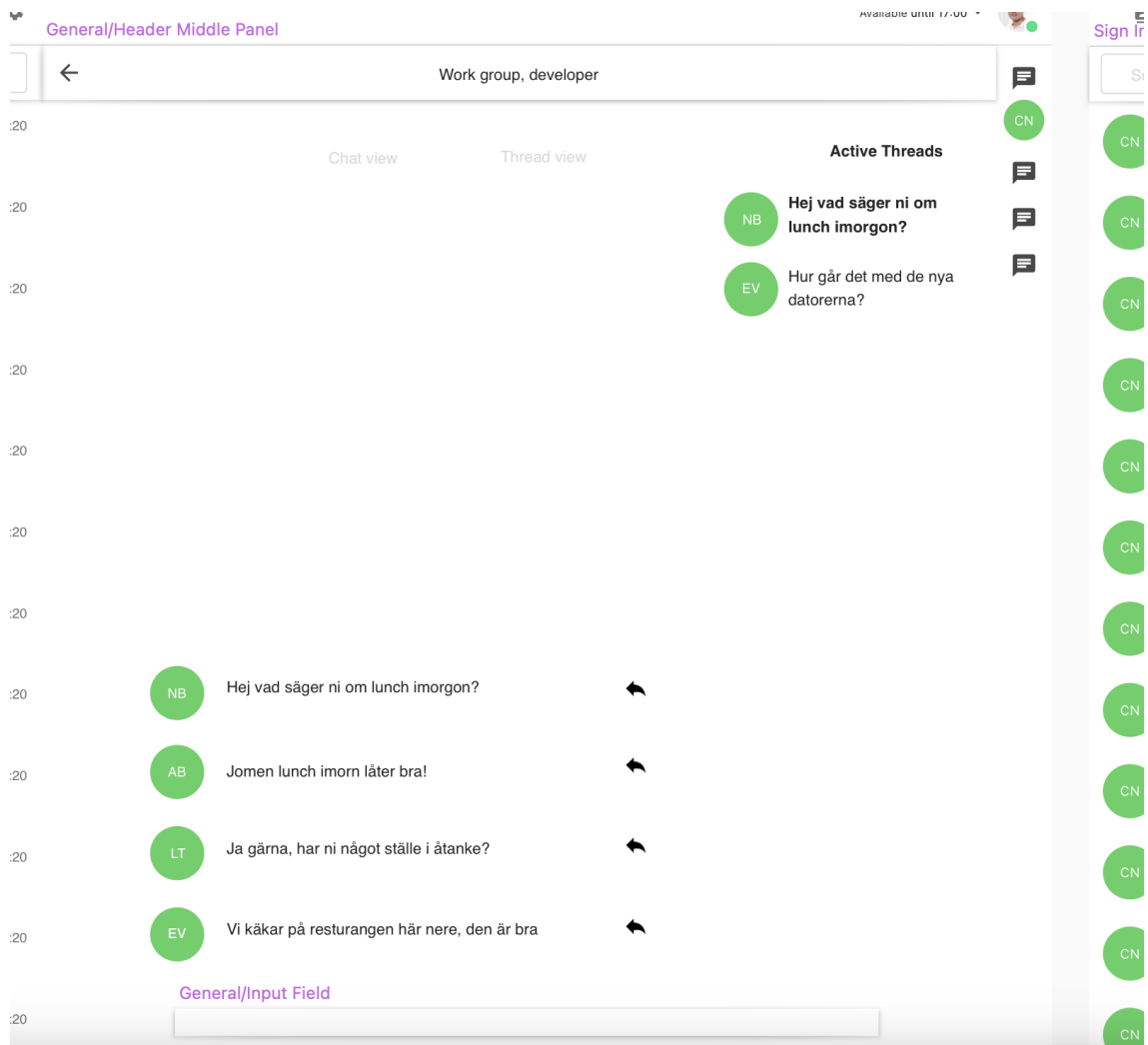


Figure C.7: Prototype 4, the inline design when inside the active threads.