Skill based routing in telephone exchanges

Henrik Edlund and Mikael Nilsson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2017

MASTER THESIS

Skill based routing in telephone exchanges

A model concept and user interface design

Henrik Edlund and Mikael Nilsson

LUND UNIVERSITY

Skill based routing in telephone exchanges

A model concept and user interface design

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

Subject: Interaction Design (MAMM01)
Division: Division of Ergonomics and Aerosol Technology
Supervisor: Joakim Eriksson
Co-supervisor: Henrik Thorvinger, Telavox
Examiner: Héctor Caltenco

Abstract

Private branch exchange systems have been used to connect clients with enterprises for over 40 years and have over time progressed from manual switchboards to modern, internet based technologies. Despite technological progress, many clients still feel great frustration when contacting customer service, not being able to reach the person they seek. As part of Telavox enhancements to private branch exchanges, skill based routing introduces a new way to connect incoming telephone calls to the best suited agent. Using available information about the client's call and the agent's different skillsets, the call can be routed and connected automatically. This project identifies different components needed to create a system model allowing for skill based routing. Based on this system model, two user interface prototypes were developed through iterative user centered design processes: One administrator interface for setting up a skill based routing system, as well as an agent interface for displaying the new routing functionality. Through user surveys indications were found showing clients' willingness to wait slightly longer to reach an extra qualified agent, especially if on hook waiting is offered. The final high-fidelity prototype was evaluated using unmoderated remote testing, validating that it can be used as a basis for implementing a skill based routing system. This project seeks to motivate future work in the field of skill based routing, providing an improved user experience for all users in the future.

Keywords: skill based routing, remote testing, usability, PBX (Private Branch Exchange), prototyping, Telavox

Sammanfattning

Abonnentväxlar, eller private branch exchanges, har använts av företag för att nå ut till kunder i över 40 år. Över tid har systemen utvecklats från stora manuella växlar till molnbaserade internetlösningar. Trots teknisk utveckling upplever många kunder som ringer telefonsupport idag frustration över att inte nå den person de söker. Som en del i Telavox satsning på virtuella abonnentväxlar utforskar detta projekt konceptet skill based routing. Genom att använda tillgänglig information om kunder som ringer, kan de kopplas ihop med den telefonist som är bäst lämpad att svara på deras frågor. Detta projekt identifierar de många komponenter som behövs för att skapa en växelmodell som stödjer skill based routing. Utifrån denna modell har två gränssnitt tagits fram, genom tre iterativa och användarcentrerade designfaser: Ett gränssnitt för administratörer som sätter upp systemet, samt ett gränssnitt för de telefonister som använder systemets nya funktioner. Genom enkätundersökningar konstaterades det att inringande kunder kan tänka sig att vänta extra för att nå en kunnig telefonist, speciellt om de erbjöds att bli uppringda istället för att vänta. De slutgiltiga prototyperna utvärderades genom omodererade distanstester. Testerna visade att de föreslagna gränssnitten kan ligga till grund för en framtida implementation av ett skill based routing system. Detta projekt hoppas kunna inspirera fortsatta studier inom området, för en framtida förbättrad användarupplevelse för alla användare.

Nyckelord: skill based routing, distanstester, användbarhet, telefonväxel, prototyper, Telavox

Acknowledgments

We would like to thank Joakim Eriksson, our supervisor at Lund University for his guidance and valuable input throughout this master thesis project.

We are grateful to everyone at Telavox for supporting us through a fantastic work environment and valuable feedback. A special thanks to Henrik Thorvinger for making this master thesis possible through his invaluable guidance and giving us the freedom to steer the project according to our interests.

Finally, we thank our friends and family that have helped us throughout the project and kept our spirits up.

Lund, June 2017

Henrik Edlund and Mikael Nilsson

Skill based routing terminology

This glossary was created for terminology clarification and ensured that the naming of different entities stays consistent throughout the report.

Administrator	A person that manages a skill based routing system.
Agent	A person answering a call to a private branch exchange.
A-number	A callers (clients) phone numbers.
B-number	The called entities phone number.
Client	A person calling the system.
Pool	A queue with skills assigned to it.
Queue	A combination of agents to which calls are routed.
Skill	A characteristic that agents can have.
Skill based routing	The concept of routing a client’s call to an agent based on certain skills.
System model	The different features, matching algorithm and data structure of a skill based routing system.

Table of contents

1 Introduction

The introduction presents the background of the thesis project in the form of research questions, goals and delimitations; as well as the report structure. It also introduces the stakeholders of the skill based routing system and project.

1.1 Background

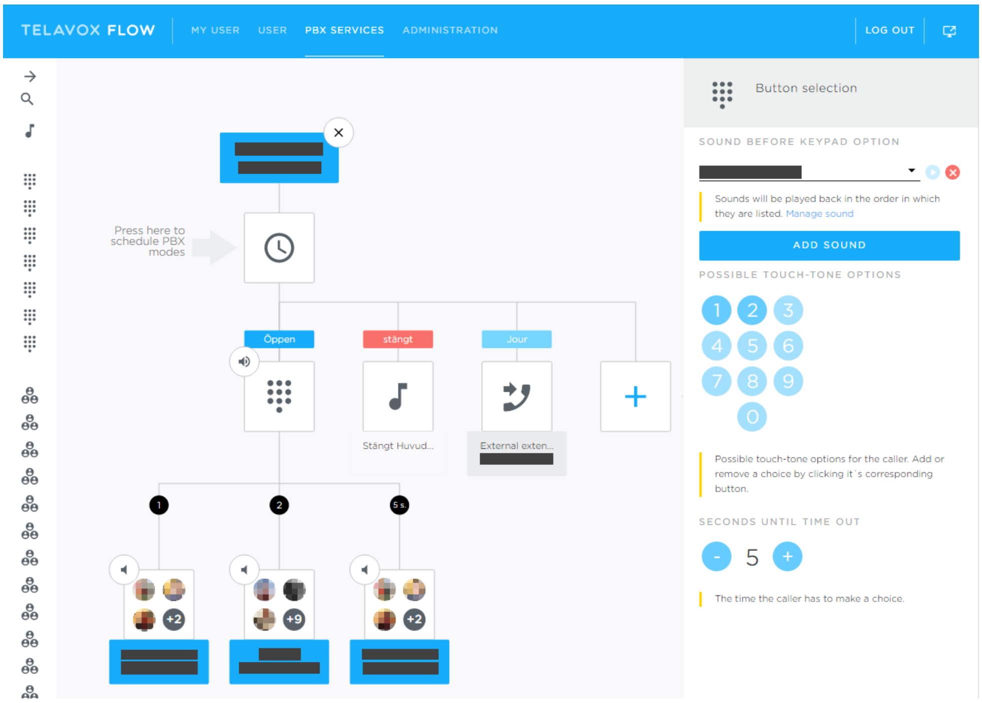
Private branch exchange (PBX) systems have been used to connect clients with enterprises for over 40 years and has over time progressed from manual switchboards to modern, internet based technologies. In recent years, PBX systems have become more and more integrated with cloud services, opening for a whole new spectrum of features and uses. Many of these features are still unexplored and there is much potential left to discover. PBX systems are often used in call centers to provide support and act as enterprises' point of contact with clients. Customer service is one of the biggest factors in brand and organization loyalty, making it a key success factor in retail (Microsoft, 2015).

Being passed between agents before reaching someone who can resolve the issue at hand is one of the most frustrating parts of reaching out to customer service, tightly followed by having to call multiple times about the same issue (Microsoft, 2015). One solution to this issue is skill based routing (SBR). SBR enables PBXs to route calls directly to the most suitable agent, based on the specific characteristics of the incoming call. While an annoyance to clients, transferring calls is also a loss of productivity for the agents not using their time to solve the issues at hand. This together makes SBR a valuable field of research for the sake of both consumers and businesses.

1.2 Telavox AB

Telavox AB is a Swedish company providing PBX and mobile services for businesses. They have a comprehensive market focus with customers ranging from small entrepreneurs to big enterprises. Telavox began as a start-up company in 2002 and has grown to a total of 23 offices in Sweden, Norway and Denmark. They are now servicing over 250 000 users daily.

Telavox's main product is called *Flow*, an adjustable PBX solution with chat and voice functionality. *Flow* is created with a mobile first approach and is available both on mobile and web. It allows users with an account to get a localized telephone number with an exchange, allowing simple customization for opening hours and routing to agent groups based on button presses. Figure 1 shows how an example PBX can be configured. At the top of the tree in Figure 1 the incoming number can be seen, followed by three scheduling states and two button choices with a timeout after 5 seconds.



Sounds will be played back in the order in which they are listed. Manage sound

Possible touch-tone options for the caller. Add or remove a choice by clicking it´s corresponding button.

The time the caller has to make a choice.

Figure 1. An example PBX setup page in Telavox's web-app *Flow*. Names and phone numbers are covered for privacy reasons.

When a client calls the number for this PBX, different options are presented to the client depending of the current schedule state. If the PBX service is open, a voice message is read and they have the option to press 1 or 2 on their telephone to be routed to a specific group of agents. In this specific case, if the client doesn't press any key before the timeout of five seconds, the client will automatically be redirected to a third group of agents.

Each agent has an account with a profile and can communicate with other agents using the smartphone applications chat functionality. The PBX system can also be managed from the smartphone application, as seen in Figure 2.

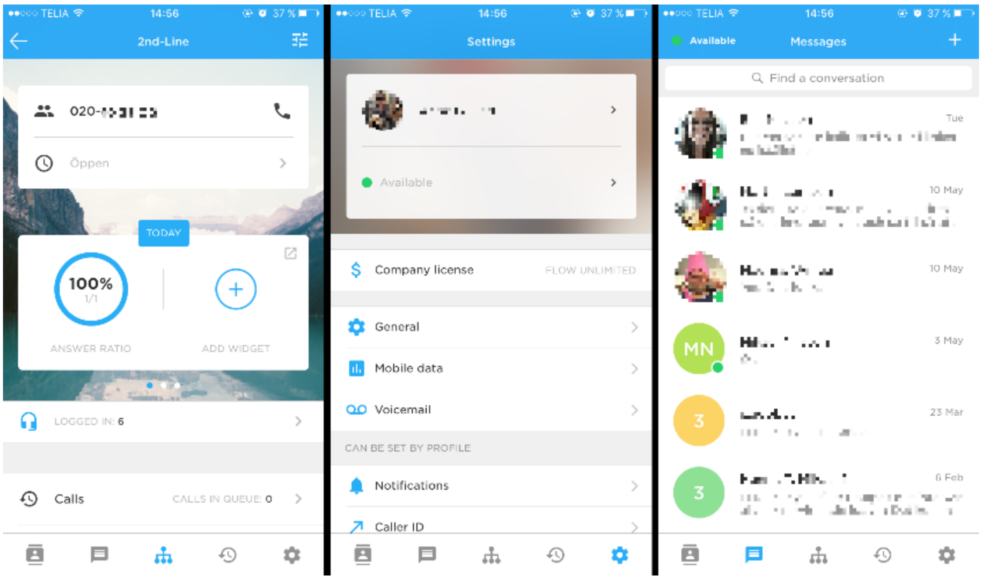


Figure 2. Examples of the mobile interfaces for *Flow*, with sensitive information covered up.

Agents can be put together into groups, and each agent group has a call strategy. Call strategies are predefined methods that specify how a call to the group will be routed. For example, one such strategy could be "call everyone in the group" or "call a random agent for 10 seconds then repeat the process with another agent". When a call comes into the PBX, it will be routed according to this strategy, and the corresponding agent’s smartphone rings.

1.3 Research questions and purpose

The purpose of the project was to research and evaluate different models and user interface (UI) design options for implementing SBR. The following research questions were defined:

RQ1. Which parameters are necessary to create a successful SBR system?

RQ2. How can a system model for SBR be created that is both forgiving and flexible?

RQ3. How should a UI be designed to present the setup of an SBR system?

RQ4. How should a UI be designed to present routing and calls to the agents in a transparent manner?

RQ5. Is unmoderated remote testing an effective way to evaluate high-fidelity prototypes?

RQ6. How do clients expect an SBR system to behave?

1.4 Goals and delimitations

The overall goal of the project was to answer the research questions. To do so, the following goals were identified in an early stage of the project:

- Identify relevant incoming call parameters from available phone call metadata.
- Create a model that if implemented would reduce call forwarding between agents.
- Prototype an easy to use wizard for setting up an SBR, and evaluate it.
- Prototype a UI for agents that enables them to understand the inner workings of the routing, and evaluate it.
- Design for the clients' best interest and introduce the SBR concept in a non-intrusive fashion.
- Identify expectations of clients calling an SBR system.

The following delimitations were set up to narrow the scope of the project and keeping it manageable in the agreed timeframe:

- The project will not be properly implemented into Telavox *Flow's* backend as the focus of the study is user experience (UX) and human computer interaction.
- Tests will not be performed on a broad representation of the population, but rather focus on the target audience of *Flow*.
- The project will be isolated from integration and development of *Flow*, to avoid limiting technical details.
- The system model will focus only on telephone routing.
- An evaluation of clients' experience in the proposed SBR system will not be carried out as they are not the primary users of the system.

1.5 Report structure

Following the introductory section, the different theoretical considerations and sources will be presented. The methods section will then in detail present the iterative design process of the project. From there on, the project will be presented in chronological order, followed by a discussion and finally the conclusion. The document is styled according to the guidelines provided by Lund University. APA was used as the adopted reference style, for better reference readability. Important figures and data will be presented in the text body while project related details and large images will be placed in Appendices at the end of the report.

2 Theory

This section will present the theoretical considerations of the project, discussing user interface and user experience design, queueing theory and routing model design considerations.

2.1 Gestalt principles

Gestalt psychology was introduced in the 1920s and is still widely applied when designing and developing interfaces. The gestalt principles are different categories of how the human mind processes and acquires different perceptions of displayed elements. (Graham, 2008). The illustrations in the following sections are recreated images inspired by Graham (2008) and Johnson (2010).

2.1.1.1 Proximity

Elements that are located close to each other are perceived as part of a group, while elements that are placed apart are perceived as separate. Proximity helps informing which elements belong together by applying regular spacing, since irregular spacing may lead to interpreting material in unexpected ways (Graham, 2008). An illustration of the proximity concept is seen in Figure 3.

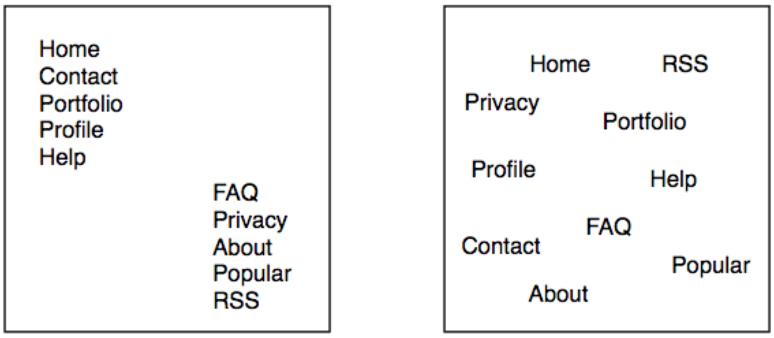


Figure 3. Proximity helps identify which elements belong together.

2.1.1.2 Similarity

Elements that are similar to each other are perceived as parts of a group. This property is illustrated in Figure 4. The law of similarity applies to elements that for example are similar in shape, color, size, proximity and direction. The elements don't necessarily need to be organized with regular spacing to be perceived as a group, as seen to the right of Figure 4 (Graham, 2008).

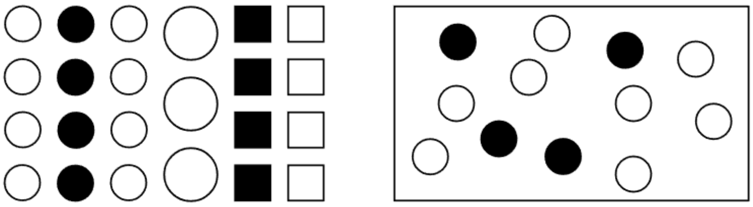


Figure 4. Similar elements are perceived as groups.

2.1.1.3 Continuation

When looking at shapes, the human eye tries to locate relationships between them. This can for example be seen in lines, curves and similar continuous shapes, see Figure 5. Continuation is also present in animation and sounds for example. Instead of interpreting each picture in an animation or each individual note in a song for itself, it is interpreted as an entire media experience as an animation or a song (Graham, 2008).

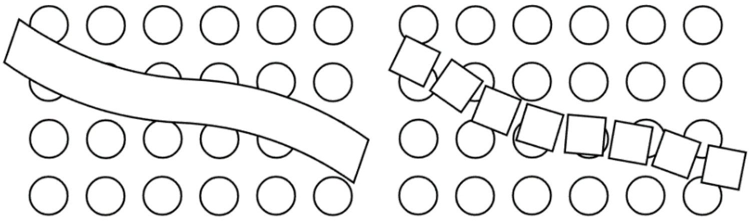


Figure 5. Similar shapes are seen as a continuation of a whole.

2.1.1.4 Closure

The human mind tries to create structures and close gaps in forms. This effect is even stronger with basic and familiar shapes. A typical gap closing example can be viewed in Figure 6. This effect is even more apparent in animations when the elements are moving in a regular and predictable pattern. The closure law and the law of continuation often work together to form a stronger experience to the viewer (Graham, 2008).

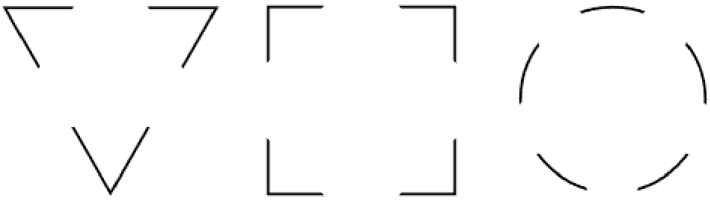


Figure 6. The human mind fills the gaps of shapes.

2.1.1.5 Figure/ground

The human mind can identify different objects and is able to distinguish them from background objects. This is often used in interactive media designs such as on web pages to for example clarify interactable elements or display different elements' depths (Graham, 2008). An illustration where this is explained can be seen Figure 7. The concept of depth in flat surfaces is heavily used in design patterns such as Googles material design (Google Inc., 2017)

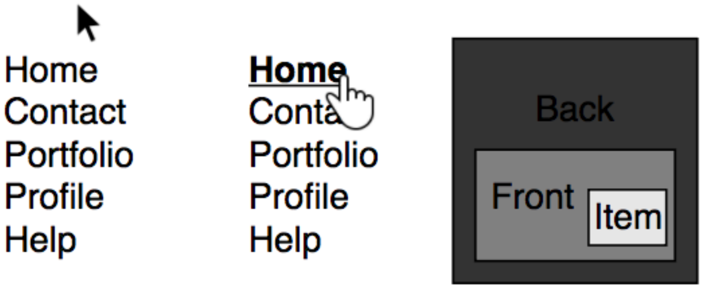


Figure 7. Some items are perceived as affordable.

2.1.1.6 Symmetry

When interpreting elements in a view, the mind tries to distinguish symmetrical elements. This is illustrated in Figure 8. where two identical squares are perceived as two squares with no filling color, and not as a shape with a small square on top of it nor two arrows pointing in each direction (Johnson, 2010).

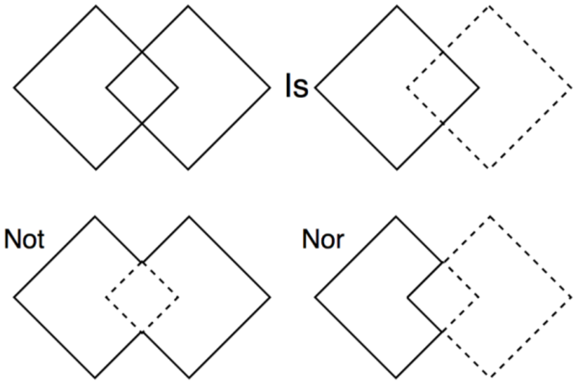


Figure 8. The mind tries to find symmetry in elements.

2.1.1.7 Common fate

According to Johnson (2010) this gestalt law is the only law that concerns moving objects. Common fate tells that elements that are moving are perceived as belonging or grouped together. Figure 9 explains how wiggling elements are perceived as grouped together.

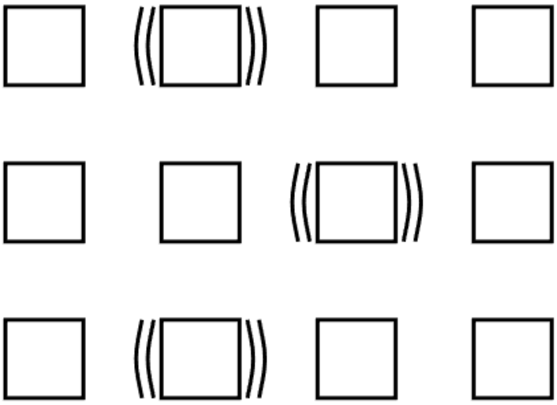


Figure 9. Moving objects are perceived as a group.

2.2 User-centered design

User-centered design, or human-centered design is an approach to developing and designing systems. The approach aims to make the systems more usable by applying knowledge and techniques about human factors, ergonomics and usability (International Organization for Standardization [ISO], 2010).

2.2.1 Norman's design principles

Donald Norman has introduced seven design principles and concepts for understanding different aspects of design (Norman, 2013). These design principles are today widely utilized in the development of projects where user-centered design and interaction design is of great importance. Norman (2013) described the seven principles as follows:

Discoverability In the discovered state of the device, it is possible to determine what actions are possible to perform.

Feedback Information about the actions taken is conveyed to the user. It should be easy to determine the new state of the device.

Conceptual model The information needed to create a good conceptual model of the design should be projected. This leads to a better understanding and feeling of control.

Affordances A good way of displaying affordance to provide a desired outcome should be present. Good affordance should intuitively convey how an object should be used.

Signifiers Signifiers are tightly bound to affordances, and help conveying affordance by indicating how an object should be used. A good use of signifiers enhances the discoverability and feedback.

Mappings Good mapping means that the relationship between controls and their actions are well communicated and intuitive.

Constraints Restrictions in the ways the user can interact with the design may help guiding the user in the right way and eases interpretation.

2.2.2 Responsive design and mobile first

Since mobile web browsing has recently been overtaking desktop browsing, responsive design is more important than ever (Chaffey, 2017). Responsive design is an approach to web development that allows dynamic changes to a website depending on the screen size it is viewed on. This allows programmers and designers to develop only one interface, compatible with multiple screen sizes and devices. Maintenance and development is therefore much easier. However, the user experience(UX) of every device must still be considered (Schade, 2014).

Mobile first design is the approach of designing for the smallest available screen size and then working upward from that. It forces the designer to prioritize the most important features and content to fit them into the small screen space (Wroblewski, 2009).

2.2.3 **Users**

Identifying the users should be the first step of designing new systems and it can be helpful to distinguish between different types of users early in this process. According to Eason (2005), users can be defined and divided into three groups: Primary, secondary and tertiary (Eason, 2005). Primary users are the direct users that may be full time users of the system, secondary users are people who occasionally use the system, or work with the output of the system, and tertiary users are the people who are affected by the system but not direct users of it (Eason, 2005). Identifying the users and stakeholders is important to make informed decisions about their project involvement (Preece, Rogers, & Sharp, 2002).

2.2.4 **User stories**

User stories are short texts written from the perspective of a user regarding a functionality that they desire from a system. They are usually written in the form of "As a <role> I want to <action> so that <benefit>", adapted to the specific role, action and benefit of the story. User stories can be used as discussion points to discuss features and produce requirements for a system (Zeaaraoui, Bougroun, & Belkasmi, 2013).

According to Roman Pichler (2014), a good way to develop user stories is through a four-step process, as seen in Figure 10.

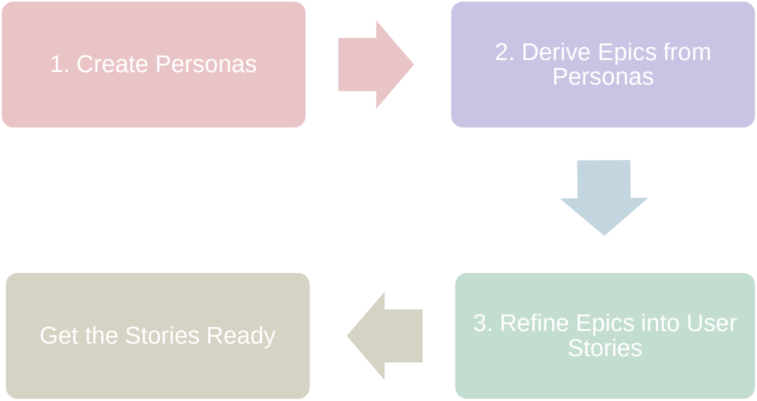

1. Create Personas
2. Derive Epics from Personas
3. Refine Epics into User Stories
Get the Stories Ready

Figure 10. Four-step model for generating user stories (Pichler, 2014).

The process starts with creating personas, in order to understand the target users and customers (Pichler, 2014). Flaherty (2015) defined personas as:

“representations of a cluster of users with similar behaviors, goals, and motivations. As such, personas are fictional, yet still realistic because they embody the characteristics and behaviors of actual people.” (Flaherty, 2015).

Personas can be created in a non-empirical way based of previous interaction with target users. This saves resources compared to data-intensive empirical approaches (Flaherty, 2015). Harley (2015) states that there are several common pieces of information to include when creating a persona (Harley, 2015). From these details, the most relevant ones to the project were identified as:

- Name, age and gender
- Experience level in the area of the product
- Context how they would use the product
- Goals and concerns when performing relevant tasks

From these personas, epics should be derived. An epic is a high-level user story, capturing the personas main goals. These epics are then refined to user stories in step three, which are finer grained than the epics. The last step is to ensure that the story is clear, feasible and testable before it is moved into development (Pichler, 2014).

2.2.5 **Prototyping**

Prototyping is an important part of user-centered design and development, making it possible to involve the user at an early design stage. The main benefit of a prototype is the inexpensiveness of creating a new iteration, as the price of developing a prototype is significantly lower than creating a final product. A UI prototype can be seen as a hypothesis, that can be rejected or accepted by testing it (Nielsen Norman Group, 2016).

A prototype may be high- or low-fidelity in terms of interactivity, visuals and content/navigation. Prototypes can be defined with in these categories according to the criteria defined in Table 1. When deciding which prototype type to develop, there are several factors to consider. High-fidelity prototypes are good for testing workflows, have realistic system response and do not require human assistance to be interacted with. Low-fidelity prototypes are often easy to change and feel easier for participants to criticize during tests (Nielsen Norman Group, 2016). The high- and low- fidelity of the prototypes are also a representation of the products maturity and depending on what needs to be evaluated, a certain prototype might be preferred.

Table 1. High- vs low-fidelity definitions (Nielsen Norman Group, 2016).

	High-fidelity prototype	*Low-fidelity prototype*
	Interactivity	
Clickable links and menus	Yes: Many or all are clickable.	No: Targets do not work.
Automatic response to user's actions	Yes: Links in the prototype are made to work via a prototyping tool (e.g., InVision, PowerPoint).	No: Screens are presented to the user in real time by a person playing "the computer."
	Visuals	
Realistic visual hierarchy priority of screen elements and screen size	Yes: Graphics, spacing, and layout look like a live system would look (even if the prototype is presented on paper).	No: Only some or none of the visual attributes of the final live system are captured (e.g., a black-and-white sketch or wireframe, schematic representation of images and graphics, single sheet of paper for several screenfuls of information). Spacing and element prioritization may or may not be preserved.
	Content and Navigation Hierarchy	
Content	Yes: The prototype includes all the content that would appear in the final design (e.g., full articles, product-description text and images).	No: The prototype includes only a summary of the content or a stand-in for product images.

2.3 Usability testing

Evaluating the products and prototypes is one of the core principles of user-centered design and one of the most common ways to perform the evaluation is through user testing. Testing a product's usability should be done in a well-documented and consistent way, as a badly designed test will let problems through to the final product (Preece, Rogers, & Sharp, 2002).

2.3.1 **Test plan**

A test plan serves as the blueprint for a user test, specifying what should be done and who is responsible for doing it. It makes it easier to foretell what will happen and forces the writer to define clear goals for the test. Without a test plan, it is likely that details get ambiguous, and the test sessions are not consistent between participants (Rubin & Chisnell, 2008). According to Rubin and Chisnell (2008) nine typical sections included in a test plan are:

Purpose, goals and objectives Having a clear goal of the testing is important and should influence all the other parts of the test plan.

Research questions The single most important part of the test plan is describing the issues that need to be solved. It is essential to be specific and clear so that the questions can be answered from the test results.

Participants characteristics Determining the target users ensures the validity of the test. Testing on the wrong user group can in some cases render the test useless. For informal usability testing, four to five users per audience group usually exposes 80% of the products defects. For experimental design where statistically valid results are needed, 10 to 12 participants per condition are required.

Method (test design) An overview of how the test will be performed to keep it consistent for all tests. Two common types of test design are between-subjects and within-subjects. Between-subjects means that users test different parts of the system and within-subjects means that all users test all parts but in different orders to combat transfer of learning. Transfer of learning means that the users get better at performing tasks as they get more training in the system, possibly masking some issues with the product.

Task list A high level description of the tasks that later need to be detailed into scenarios. They should cover the most common tasks that will be performed on the product when released.

Test environment, equipment and logistics A list of resources needed for the testing in advance, so that it is not a surprise at a later stage in the testing.

Test moderator role A definition of the moderator's role and for instance how leading he/she should be during the test.

Data to be collected Definition of the performance and preference data needed to answer the research questions. It is good practice to limit the data collection to what is absolutely needed. Data can be in the form of qualitative or quantitative and subjective or objective.

Report content and presentation A short description of how the data will be reported, mostly used to communication with other stakeholders.

2.3.2 **Test methods**

Testing can be done in several different ways depending on requirements and available assets. They can for instance be done on-site or remotely, moderated or unmoderated and with single or groups of users. Each method has its own positives and negatives (Rubin & Chisnell, 2008).

2.3.2.1 Unmoderated remote testing

The unmoderated remote test is completed alone by the participant, without a moderator. Instead they follow instructions and pre-defined tasks, answering questions after each task or at the end of the session. A downside of unmoderated remote testing is the lack of support the participant can receive during the test (Schade, 2013). The advantages of unmoderated remote testing are that multiple test can be done simultaneously, lowering costs in terms of both time and logistics. A big drawback is that participants might not know if they finished a task correctly, which could affect the outcome. Unmoderated testing is most effective when a specific question needs to be answered (Soucy, 2010).

2.3.2.2 Moderated in-person testing

During a moderated in-person test, a test moderator is present to guide and observe the participant during the test. This allows the moderator to ask follow-up questions or help the participant. This naturally introduces the possibility of bias, as the moderator might affect the participant (Rubin & Chisnell, 2008).

There are several ways to get the most out of an in-person test. One such technique is to “think-aloud”, where the participant verbally says what they are thinking and experiencing. This can provide important insights in why a certain task proves difficult. One advantage of the “think-aloud” technique is that the users provides immediate feedback that they might otherwise forget. A big disadvantage is that performance of the participant is slowed significantly (Rubin & Chisnell, 2008).

2.4 Routing and queueing theory considerations

SBR is closely related to queueing theory and a badly designed system could increase queue times and reduce the efficiency of the routing. Therefore, it is important to consider some scenarios and how the system will handle load depending on different designs.

Notation in queueing theory is often done as λ for average arrival rate of customers and μ for average rate of service customers. Processes are described as series of symbols separated with slashes $A\ /\ B\ /\ X\ /\ Y\ /\ Z$. A indicates the arrival-time distribution, B the distribution of the service pattern and X as the number of parallel service channels. Y represents the capacity of the system and Z queue discipline used. For this project only first in first out queues will be considered and the last symbol can be ignored for simplicity (Gross, Shortle, Thompson, & Harris, 2008).

A queueing system can be either static or dynamic. A static system has fixed priorities while a dynamic system changes its routing depending on system load (Garnett & Mandelbaum, 2000). It is important to note that the routing rules of a static system may change over time and it will still be considered static, as long as it is not load dependent.

Garnett and Mandelbaum (2000) describes two possible routing policies, “Greedy 1” and “Greedy 2”, applied to a N-design M/M/2 queue. M/M/2 queues are basic Markov chains with Poisson distribution arrival, service rates and two servers. Figure 11 shows the N-design queue and parameters that Garnett and Mandelbaum used in their simulations. In “Greedy 1”, S_2 prioritizes λ_1 clients if S_1 is busy. In “Greedy 2”, S_2 prioritizes λ_2 clients if there are any and otherwise help with serving λ_1 clients. This in practice means that “Greedy 2” is inefficient in distributing the total workload of the system, as the queue for λ_1 grows at a faster rate. “Greedy 1” however prioritizes λ_1 clients, which instead increases overall throughput but is unfair to clients from λ_2. These policies assume that S_2 is capable of handling clients of both types, which might not always be the case, and that there is no abandonment from the queue.

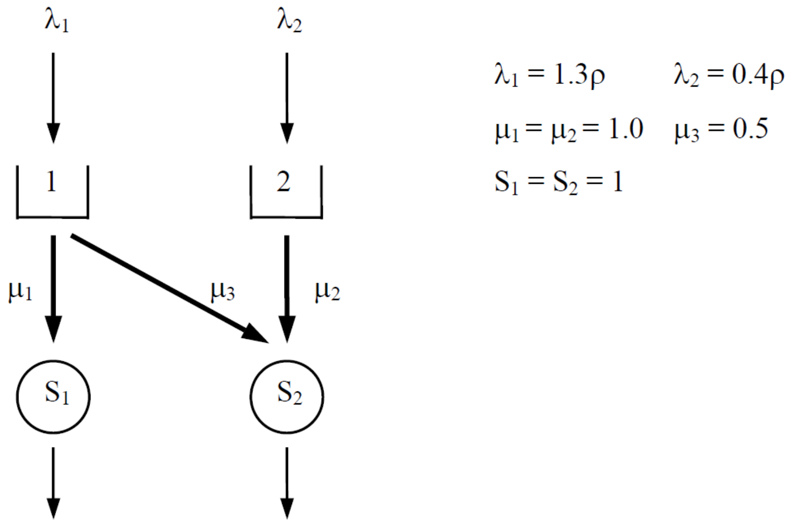


Figure 11. N-design queue with two servers. The two policies “Greedy 1” and “Greedy 2” control how S2 is populated by either prioritizing incoming calls from μ_1 or μ_2 (Garnett & Mandelbaum, 2000).

While SBR has the potential benefits described earlier, there are several potential issues that can arise when implementing SBR. Simulations done by Wallace and Whitt (2005) show that routing to agents with the experience of knowing two skills greatly improves call center performance, while adding additional skills after that only have some modest improvements. Their work and research in this case is based on a M_6 *$/$ M $/$ 90 $/$ 30* Erlang-C model. In accordance to the above notation, this means that they used a Poisson distribution arrival with 0-6 arrivals per time unit, Poisson distribution service time, 90 servers and a maximum of 30 customers allowed in the system at once. According to Wallace and Whitt (2005) it is advantageous for performance reasons to route calls to less flexible agents first, when agents do not have the same number of skills (Wallace & Whitt, 2005). This is also intuitive as flexible agents can answer more types of calls, and saving them for last increase the number of available people for each skill.

2.5 Data gathering

There are several methods of data gathering. This section will briefly introduce the concepts of interviews, heuristic evaluation and surveys as means to gather data. Good data is essential to make informed decisions about the development of a product and its features.

2.5.1 **Requirements elicitation**

The goal of requirements elicitation is to improve one's understanding of the requirements of a system. This can for instance be done by eliciting requirements from the different stakeholders through brainstorming sessions or interviews (Pohl, 2010). To gather several points of views, unstructured interviews can be used. Such interviews can be beneficial to gain new insights and aspects of a problem, with the risk of going into considerable depth about specific topics. It is recommended to take notes during the session and analyze the data quickly after the session (Preece, Rogers, & Sharp, 2002).

The actual effectiveness and performance of brainstorming sessions are hard to measure. However, these sessions are often efficient in generating possible design solutions (Sutton & Hargadon, 1996). Some similarities can also be seen between focus groups and brainstorming sessions. Preece et. al. (2002) describes focus groups as good opportunities for stakeholders to meet and highlight different areas of disagreement. Focus groups provide mostly qualitative data, and introduces possible issues if certain participants are very dominant in the discussion (Preece, Rogers, & Sharp, 2002).

2.5.2 **Heuristic evaluation**

Created by Nielsen and Molich in 1990, heuristic evaluation is a cost-effective way of identifying usability problems in a UI. It has been shown that letting several individuals analyze an interface with consideration to a checklist of heuristics can greatly increase the number of problems identified. The heuristics are used to describe what the issue with a certain element or feature is and categorize these issues. The number of evaluators needed depends on the cost benefit analysis.

Roughly 60% of errors can be found with two evaluators and the cost to benefit ratio decrease as the number of evaluators exceed three. (Nielsen, 1995b).

Heuristic evaluation is a rather straightforward evaluation method consisting of three stages: Briefing, evaluation and debriefing. During the briefing session, the experts are told what to do during the evaluation. The evaluation is thereafter performed. It is a good idea to take two passes through the interface, the first focusing on the general feel of the system and the second pass focusing of specific elements. The evaluation is followed by a debriefing where the findings are discussed (Preece, Rogers, & Sharp, 2002).

Nielsen (1995a) presented 10 general principles for interaction design, that can be used for heuristic evaluation of a user interface. These heuristics can be found in Appendix A.

2.5.3 **Surveys**

Surveys can be used to understand preferences of a broad base of users. They can be used in any stage of the products lifecycle, but are often more impactful in early stages (Rubin & Chisnell, 2008). Likert scales are used for measuring opinions, attitudes or beliefs. It is important to keep the use of scales consistent and not switch ranges or the meaning of different ratings (Preece, Rogers, & Sharp, 2002). Using rating scales with between five and nine choices are optimal for most surveys, as adding more quickly gives diminishing returns (Spector, 1992).

When writing questions for surveys it is important that the questions are adapted to the intended audience and not biased to a specific answer. Questions might be open, where the respondent freely writes the response, or closed, where the respondent can choose between available options. As with any research method, a survey should be piloted on a member of the target population to identify issues or errors (Kelly, Clark, Brown, & Sitzia, 2003).

3 Overview of Methods

This section presents the projects different phases and gives a high-level description of the methods used in each phase.

3.1 Design process

Inspiration for the design process was taken from Arvola (2014) and his three design phases, seen in Figure 12. In the concept phase, new ideas and patterns are explored to form the foundation of the product. This foundation is then the basis used in the elaboration and detailing phases, used to refine the goals and functionality of the product. The overall motivation behind the three-step process is to reduce uncertainty about the product and combining ideas to a final product (Arvola, 2014). In practice, this is done by using an iterative process in all the design phases.

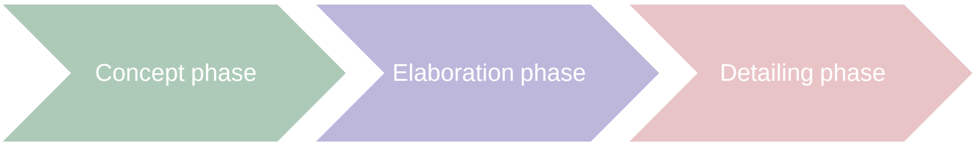


Figure 12. The different design phases suggested by (Arvola, 2014).

No product or design is perfect from the start, and each design phase consists of an iterative design cycle. The cycle consists of the design, test and evaluation steps for the current iteration of the interface, as seen in Figure 13 (Nielsen, 1993).The iterative process makes it possible to find good and bad characteristics of the product early in the design process and correct them if necessary. Users are often very good at detecting what they do not want in a product, and therefore building many prototypes in several iterations creates a better result (Preece, Rogers, & Sharp, 2002).

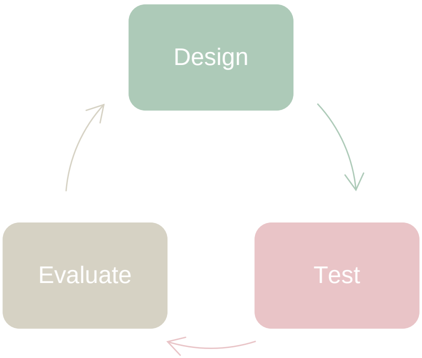


Figure 13. The iterative design process, inspired by Nielsen (Nielsen, 1993).

3.1.1 Concept phase

In the concept phase, requirements for the project will be set up and the SBR model's basic features designed. To gain an understanding of the domain of PBX systems, routing and *Flow*, the phase will start with a literature study and expert interviews. The literature study is an important part of the master thesis project and it is crucial that sources and literature used is both relevant and trustworthy. For this project, Lund University's tool LUBsearch was used to find relevant sources whenever possible. These sources are then verified using Web of Science based on article and author citations, as recommended by Lunds Tekniska Högskola Libraries (LTH Libraries, 2017).

This initial study will result in a glossary to assert the naming of different entities. Heuristic evaluation will be used to find good features and problems in the current version of *Flow*, to shape the design of the new SBR system. User stories will be developed to identify the use cases for the different user categories, described as personas based of the interviews. The queueing theory considerations described in the theory section will be used to create an effective SBR model, through brainstorming sessions.

Based on the results from these steps, low-fidelity prototypes for the different interfaces will be created. The low-fidelity prototype will consider the different design practices and guidelines described in the theory section.

3.1.2 Elaboration phase

In the elaboration phase, a medium-fidelity prototype will be created and evaluated through in-person user testing. This middle step is done to identify problems with understanding the SBR concept and UI interaction before moving on to the high-fidelity prototype.

A survey will be created and distributed to fine-tune different parameters of the SBR model to match expectations of the clients calling into the system. These model features will be described through flowcharts.

3.1.3 Detailing phase

In the detailing phase, a high-fidelity prototype will be developed. The high-fidelity prototype will be based on all previous results from test sessions, surveys, input etc. The results from this prototype will then be evaluated through unmoderated remote testing. The purpose of the high-fidelity prototype is to evaluate the participants understanding of the system, as well as move closer to a finished product. A test plan will be created to guide the remote testing and make sure the testing will be as consistent as possible.

3.2 Triangulation

There are four kinds of triangulation that can be used to verify and validate findings: methods triangulation, triangulation of sources, analyst triangulation and theory/perspective triangulation. Methods triangulation will be used in this project's early stages to ensure consistency in data collected through different methods. The point of this is to offer opportunities for deeper insight into the phenomenon that is being studied (Patton, 1999). In usability design and testing, this can be used to make sure most issues in a product are detected. This project utilizes heuristic evaluation, interviews and user testing to triangulate user behavior that can then be used as a base for the system and interface design.

4 Concept Phase

The first part of the project was the concept phase. This started with a very broad literature and industry study, moving on to skill based routing model design and user interface sketching. The results from this section would then be evaluated.

4.1 Requirements

Setting up requirements is the first and arguably most important step of a project. They are used to identify the opportunities and objectives of creating a new system and clearly defining the functionalities and constraints. They give a clear goal to work towards and are important for stakeholder communication and progress tracking (Pohl, 2010).

4.1.1 **Data gathering**

The first few weeks of the project was used to gather data and understand the users of Telavox *Flow*. All the information gathered was written down and structured. The gathering of data started very broad to not miss out any aspects of PBX systems or routing. The motivation behind this was to later prioritize the necessary functionality of the project, while filtering out less important features. For this part, it was very important to consider the aspects defined in the goals and delimitations section.

4.1.1.1 Expert presentations

An introduction was held by Henrik Thorvinger and involved a thorough presentation of Telavox and its main product, the virtual PBX system *Flow*. The introduction of *Flow* described which options and parameters an administrator of the system can configure, but also which information is and can be gathered from the system's incoming calls.

It was identified that according to Eason's (2005) definition the primary users of *Flow* are the administrators and the agents because they directly interact with the system. The clients are considered tertiary users as they call and are affected by the system, but will never interact with the UIs. No secondary users were identified.

The average company that is using *Flow* does not have any considerable queue times, as they keep their number of agents (servers) to arrival rate ratio large for business reasons. This means that clients in most cases get connected to an agent directly. However, the system should be designed to withstand heavier load and must manage incoming calls in a fair way.

After being introduced to the relevant parameters that the *Flow* system can collect from an incoming call, a better overall understanding of the system could be formed. This led to an introduction showing some thoughts of how SBR could be designed based on the incoming information. The thoughts of how the information could be used is presented in Table 2.

Table 2. Summary of the results gathered through the presentations, showing the different possible usages of the inputs to the system.

Input	*Example usage*
Caller ID (A-number)	Country code for language identification and processing. Lookup to database for finding e.g. assigned agent, agents last talked to, generic keywords or descriptions from previous interactions, segment or location lookup.
Called number (B-number)	Country code for language identification and processing. Segment or office lookup
IVR (Interactive Voice Response) input	Segment or topic lookup. Query databases with personal/customer number.

Following the parameter detection was a discussion about how the call strategy for SBR could be made. The call strategies available for queues today can't really be applied to an SBR system. The current implementation of calling all available agents is for example not applicable to the SBR scenario, as it completely negates the point of smart prioritized routing. It was identified that every incoming call would generate a unique prioritized list of relevant agents to call, and to keep the system flexible, a new type of strategy to route the calls must be implemented.

The proposed call strategy concept was called "circles of offering" and was defined as follows: Start off by calling a defined number agents on the prioritized list of agents and notify a few more agents down the list that a call might come to them. After a specified time, if no agent picks up the call, start a new phase. In this phase, add those who previously were notified to the calling list and call them as well, while notifying a few other agents down the list and so on. The circles of offering phase strategy is illustrated in Figure 14.

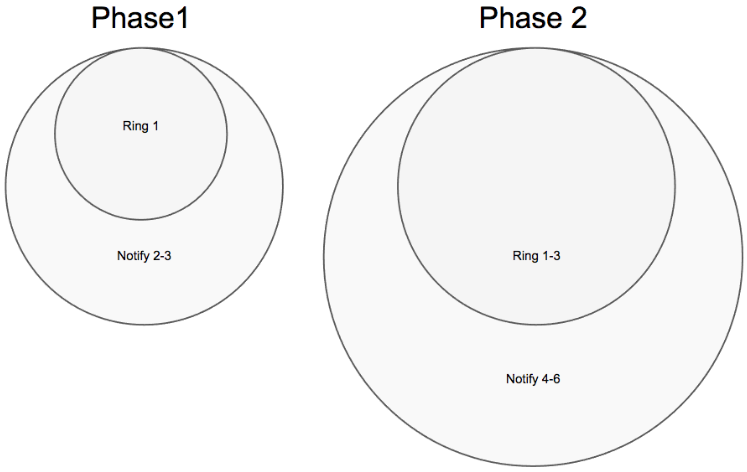


Figure 14. The circles of offering call strategy concept, with two phases. The numbers represent the ordered numbers of agents on the prioritized list.

4.1.1.2 Unstructured interviews

Three unstructured interviews with the expert Henrik Thorvinger were made throughout the first weeks, with approximately three days in between. The main points that were brought up during these interviews were:

Overflows between queues When no one answers in one queue and the call is redirected to another, it can be very confusing in *Flow*. Most companies set up overflows to ensure a client will always reach a person at the company in the end. Agents then do not know what clients are calling about or who they want to reach, as calls can come from other queues overflows. On a technical level, it also affects statistics by counting incoming calls several times.

Queue time Queue times can be seen as perceived and actual times. The perceived queue time for clients can be lowered by keeping the clients busy with IVR choices while they are in a queue.

IVR endpoints Because IVR choices can be seen in a tree structure hierarchy, the concept of skill could be defined as the endpoints on these trees. The category or skill a client calls then be the very last IVR choice has been pressed. This would then map to an agent skill.

Agents skills Three different alternatives were identified: The first one being a 0% to 100% scale that would be used as a weight parameter when calculating the relevance list for the incoming calls. The second alternative would be a predetermined number of levels such as “None”, “Beginner”, “Intermediate” and “Expert”. The third and last options was a prioritized set or list of skills, such as skill A is a primary skill, skill B is a secondary skill and so on.

Companies privacy May be of concern, and they may want to protect customer and agent data. A possible solution would be to implement a way for the company to generate and use their own prioritized agent list.

To fully understand how to operate *Flow* and to get to know it better, new accounts to the system were provided to the project members and full administrator access to the accounts was granted. Even though the main areas of *Flow* were covered in the presentation a single-handed exploration of the system gives more input. This is because using the product introduced the possibility of performing a hands-on heuristic evaluation of *Flow*.

4.1.1.3 Heuristic evaluation

The evaluation was done by both project members, focusing not only on design flaws but also good features of *Flow*. The motivation behind this is that the end result of the project will not be a replacement UI for *Flow*, but rather a separate feature and system. The heuristics were used to provide a structured way to perform the evaluation, providing some input for the design of the new SBR system. The evaluation was done with respect to the checklist for heuristic evaluation seen in Appendix A, and then summarized and categorized in Table 3 according to Donald Normans seven fundamental principles of design (Norman, 2013). The findings were discussed, and the most important points, which were agreed upon by the project members are highlighted in Table 3.

Table 3. Results from the heuristic evaluation, with the key points highlighted in bold.

Principle	***Identified strengths and weaknesses***
Discoverability	System and agent status is clear though activity profiles. Summary screen shows everything that happened during a certain period.
Feedback	**When making changes, they easily can be either undone or saved.** The system is very responsive. Buttons provided on-hover and pressed states. Loading bars appear when larger data sets are fetched.
Conceptual model	**Mobile first design with clear layers and navigation**. **The PBX hierarchy clearly represents how a call flows through the system.** Android and iPhone interfaces are different, outside of their defined guidelines.
Affordances	It is clear what is a clickable icon and what is not.
Signifiers	Clicking an object inspects it closer and opens options.
Mappings	Contact have pictures. **PBX elements have clear icons**. **IVR setup shown as a keypad.**
Constraints	Some buttons appear available but are not clickable. Navigation can only be done with the mouse. Keyboard can be used for input.

These strengths and weaknesses were later considered as a basis for the SBR system and UI generation.

4.1.2 **Epics and user stories**

Instead of creating detailed use cases for all required functionality, epics and user stories were created. The reason for this is that when developing interfaces without proper implementation, the UX should not go into too much detail, but rather be planned generally (Laubheimer & Loranger, 2017).

In an effort to consider all possible end users in the design phase, personas with different characteristics were created. Personas were created for both the primary and tertiary users, in form of agents, clients and administrators. These personas are based on the interviews performed earlier on the project, as well as desired features found through brainstorming. The purpose of the personas was both points for discussion for model creation, as well as bases for mock data creation. A list of the different personas and their basic characteristics can be seen in Table 4.

Table 4. Created personas and their characteristics.

Persona	*Type*	*Description*
Alice	Agent	Alice is 35 years old. She is very outgoing, has good social skills and is very skilled in one of the company's needed skillsets. She works in a call center and answers more than 40 calls on an average day. She uses a headset to talk on her mobile phone, but uses a computer to see information about calls and direct them to her phone.
Bob	Agent	Bob is 44 years old. He has worked in the company for many years and is equally skilled in two different but related segments. He is a sales representative at a firm that sells clothing and always carries his phone with him answering if someone calls the store. Bob never has access to a laptop.
Charlie	Agent	Charlie is 52 years old. He is multilingual, has worked in several of the company's departments and is competent in a wide variety of areas. Often gets "random" calls from overflows as he is part of many queues. He can take any call, he does not care about customer relations since it is easy for him to talk to people.
Anton	Client	Anton is 21 years old. He is interested in only one specific segment of a company's products. He does not tolerate waiting and want to reach a human as soon as possible when calling. For him it doesn't matter whether the answering agent is a knowledgeable person or not, because the questions he asks do not need any specific set of skills.
Beatrice	Client	Beatrice is 36 years old. She represents a business client and makes calls about many different issues and segments. She needs fast support, but likes to reach her assigned business contacts directly. For her, there is no time to be passed around between agents nor talk to an inexperienced agent.
Carrie	Client	Carrie is 29 years old. She calls for support about one time per year, and does not care much about waiting when doing so as long as her issue is resolved. It is easy for Carrie to have a conversation with other people and she is bilingual.
Arnold	Administrator	Arnold is 40 years old. He has close contact with everyone at his company and knows their skillsets. He is tech savvy and has years of experience managing IT systems. He has never used any PBX system before and wants to try if Flow is a good choice for the company. He is not scared of making errors, since the impact on his small company is not that great if he does.
Bianca	Administrator	Bianca is 23 years old. She is a newly graduated IT student and has a lot of experience with computers. She is newly hired and works part time with IT at a company. She has good IT knowledge but has a busy day to day schedule which is why she wants to have as much call throughput as possible.

Christof	Administrator	Christof is 38 years old. He is a skilled IT administrator, but likes to spend most of his time alone. He does not know the other people at his company, and is not interested in their skill-sets or background. It is better to have a bit higher throughput for the company he works for, instead of having long queues and waiting times.
Danny	Administrator	Danny is 61 years old. He manages his own company and has the responsibility of handling everything from billing to IT administration. He has no interest in computers, but see them as a necessary evil to get his work done. He doesn't know which call tactic the company needs, but wants it to be balanced.

From the created personas, epics, and user stories were created as tasks for each user groups. These tasks defined what the users should be able to accomplish in the system. Some of the defined tasks are already available in the current implementation of *Flow*, and were deemed important but not part of the scope. This include tasks like answering a call or redirecting it, which are important but not closely related to the SBR model. The results of the different features required for agents where summarized and the tasks can be found in Table 5.

Table 5. Summary of desired tasks found through the agent user stories.

Category	***Feature***
In call	Answer. Hang up. See the 1st and 2nd ranked agent in the unique priority list. See own rank (how good the match is) of the incoming call. See skill topic of the call.
During ringing	Transfer call to web/mobile. Assign and lock self as the 1st ranked agent in the priority list. Toggle the 1st and 2nd ranked agent in the priority list. Transfer call to other agents.
History	Toggle the 1st and 2nd ranked agent in the priority list. Assign and lock self as the 1st ranked agent in the priority list. Call the client.

For the administrators, the results were also summarized and the tasks can be found in Table 6. The features are all related to setting up and administrating the different agents in an SBR pool.

Table 6. Summary of desired tasks found through the administrator user stories.

Category	*Feature*
Initialize	Convert an existing queue to an SBR pool. Create a new SBR pool.
IVR	Add IVR choices. Label the IVR choices with skill names.
Skill setup	Add agents to the pool. Remove agents from the pool. Change the skills for agents. Filter available skills. Search agents in the pool.
Call strategy	Select a pre-made call strategy. Select a custom call strategy. Create a custom call strategy. Save the custom call strategy.
Tiebreaker / Custom routing	Select a tiebreaker system to use. Connect an external tiebreaker system.

While considered, no interface would be created for the clients as it was found through the expert interviews that no one would like to download an application or use a web service in order to call for support.

4.2 Model generation

Based on the data gathered in the previous section, a model suggestion was created that would support the desired traits.

4.2.1 **Presentation**

To summarize the requirements generation, a ten-minute meta-level presentation was held by the project members for the project supervisor. The motivation behind this was to create a platform for discussion and force prioritization of features to deem what was important. The presentation was broken down into four sections: Problems, system contribution, best usage scenarios and inner workings. The topics of each section can be seen in Table 7.

Table 7. The four sections discussed during the meta-level presentation.

	Specificities
Problems	Transfer between agents reduce efficiency. Disconnects increases irritation and introduce more transfers. Call topics are unknown to the routing system and the agents. Account managers currently need to share private/internal agent or group numbers with clients.
System contribution	Simple setup of a smart PBX. Web and mobile interfaces for both administrators and agents.
Best usage scenarios	Mixed areas of competence or relative skill between agents. Handling of recurring clients calling the same number.
Inner workings	1. Analyze incoming calls. 2. Create a unique list of matching agents. 3. Distribute calls to agent. 4. Store information about the call for future routing.

The inner workings section, seen in Figure 15, is a high-level description of how the SBR system will work. The design of the interfaces will differ slightly from this flow, as a large part of it is done in the background and not visible in any kind of front-end UI. The first step in Figure 15 is analyzing the paraments available from the call, these are then used in step two to create a prioritized list of agents that the call should be routed to. The call is then distributed to the agents on the list according to a defined call strategy. Lastly the information about the call, for instance who talked to who, about what and how long, is stored in a databased and used to create better match lists in the future. An important feature of the model is that it is always at least as good at routing as the current solution, not creating any unwanted confusion for users.

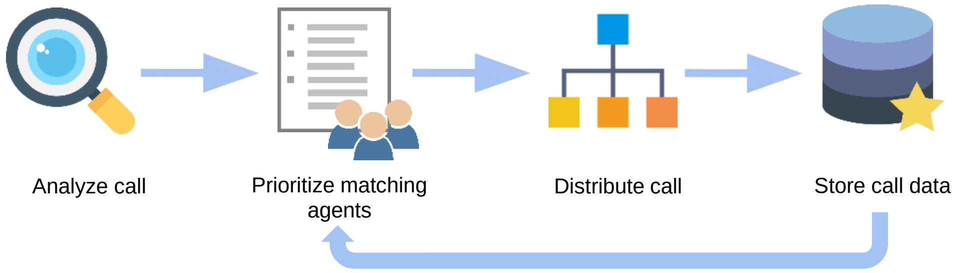


Figure 15. The four-step description of how a call is handled.

4.2.2 Model components

It was realized early in the project that to make the model creation feasible, the different components would have to be identified and analyzed. These components would then be combined into a final model. The model should be as simple as possible, while still feature rich and future proof. The different components where found as:

Agent skills How the knowledge of the agents is represented. Suggestions included percentage scales, labeled skill levels or keywords/tags. The usage for this would be to decide which agent is best suitable for a specific input. Three different ways to assign and create the skills were found: Crowdsourcing (self-assigned) from the agents, setup by team leaders or setup by a system administrator.

Queue times As described in the Theory section, the system could be static or dynamic. A dynamic system would react to how long clients stand in line. An example usage of this is to use thresholds that will call more agents or lower the needed skill requirements if queue times get too long.

Languages It was identified that languages are slightly different from regular skills, because of their nature. If an agent can give support in a specific language is a “binary” skill, either the agent can give support in that language or not. It was considered to classify the incoming calls' language using different probabilities, based on the A- and B-number. Another approach was to let the users choose their language with IVR choices.

Tiebreakers Since skills are defined as a rather broad concept, tiebreakers would be needed to separate agents in a finer gradient. This is because many agents with the same skill sets would have the same rank in the prioritized list, hence the name "tiebreaker". The tiebreakers that were found were: calls today, time in call today, a last-talked-to vector/index and number of matching hashtags. Such tiebreakers would also introduce a level of fairness for agents, changing up the routing so that one person does not receive all incoming calls. The advantages of using a hashtag based system is high customizability, but it requires more setup through defining which tags are more important and is thus inherently more complex. An option was also to allow tiebreakers from customers Customer Relationship Management (CRM) systems, to improve customizability of the model. Such a system would be implemented by the customer and integrated into the SBR routing, opening up for a new level of flexibility. An example usage of this could be to ask the client for a customer or personal number when calling, enabling the system to get a large amount of external input.

Distribution Regular call strategies such as randomization, round robin or call everyone does not make much sense in a SBR system, as the client always want to reach the best suited agent. Therefore, a new way to distribute the calls is needed. The considerations came down to a phase based call strategy, as described earlier in the Expert presentations section, see Figure 14. The strategy will have a specified number of agents that will be called for a specific duration for each phase. As time passes, the number of agents called will grow as the system moves in to the next phase. This keeps the best suited agents as potential call answerers, while it attempts to shorten the queue time through a best effort mindset. An option is also to allow override of the routing system to create VIP queues or routings depending on input or available data.

List generation The list with agent-client matches could be generated in different ways depending on how skills, languages and other parameters are defined. In the case of skill scales, 0 to 100 for example, a mathematical “match” rating for each agent would have to be calculated to sort by. In the case of a more binary tag system, sequential sorting algorithms could be used instead.

4.2.3 Model results

Several iterations where made from the model using different combinations of the described components and then discussed together with both development and UX experts at Telavox. The final model was defined with the following components:

Agent skills An agent will have a prioritized list of skills, meaning a primary, secondary etc. skill, providing a hierarchy between different skills. This enables list creation without calculating a score for each agent every time a call reaches the system.

Queue times The system was defined to be static, as *Flow* rarely has to handle long queue times. A dynamic system introduced a lot of complexity while not giving enough efficiency to be worth implementing.

Languages It was seen that languages added a layer of complexity that was hard to solve, as there are many edge cases that was hard to prepare for. To simplify the model, it was decided that division by languages would have to be done manually by the system administrator. This would be done by creating the appropriate skill, or using separate SBR systems and phone numbers for each language.

Tiebreakers Because of the added flexibility and business value for the model, support for fetching outside data was incorporated into the model. This enables administrators to use custom routing parameters and CRM systems. However, the exact implementation of this was deemed outside the scope of the project and suitable for future studies. A last-talked-to vector/index however became the default tiebreaker, meaning that the system will try to connect the client to the, for that client, most recently talked to agent.

Distribution It was decided that the phase based system should be used, where administrators could create their own custom call strategies as well as use predefined strategies. The phase system was chosen because it was seen as intuitive and explainable to the users.

List generation The early alternatives of the models relied on calculating different scores of agents, depending on for instance the matched skills, agent knowledge, availability and queue times. The benefit of this approach is that different tiebreakers do not need a fixed hierarchy, for instance that languages should always be the first tiebreaker. Instead different tiebreakers can have the same priority that results in a total agent match score. These versions however proved too complex for the small added benefit that they provided, and the result can be seen in Table 8.

Table 8. The hierarchy of the proposed tiebreaking. The skill groups are inner sorted with the parameters in 3-5.

Priority	*Description*
1	The agent's primary skill.
2	The agent's secondary skill.
3	Inner sort: How recent the agent talked to the client.
4	Inner sort: The agent's number of calls today.
5	Inner sort: Arbitrary number of skills (motivated by Wallace and Whitt (2005)).

On a conceptual level, the final model can be described through Figure 15. The call comes into the PBX system and the A-number is used to do an external database query, detecting if there is information about where the call should be routed. Alternatively, information is gathered through an IVR button press. When the call reaches the skill pool, an ordered list is created with agents according to their primary and secondary skill sets.

An example scenario could be described as follows: "*Bill calls number +123123 and is prompted to press button 1 on his phone to reach tech support, and button 2 to reach the sales team. He presses the number 2, and his call is tagged with the "Sales" skill by the system. The system then creates a list of people, ordered first by agents who have the primary skill "Sales" assigned to them, and then agents who have "Sales" as a secondary skill. These agents are then called in an expanding fashion until someone picks up the phone."* The result of the example can be seen in Table 9, where the different sorting and inner sorting are color coded.

Table 9. Example of the prioritized list when a call about Sales comes in to queue with 5 agents. The sorting is color coded. It is first sorted on the primary and secondary skills. These two groups are then inner sorted based on the tiebreaker. Finally, agent E is sorted above agent D as E has one skill and therefore is less flexible.

Priority	*Agent*	*Primary skill*	*Secondary skill*	*Tiebreaker (Last-talked-to)*
1	C	Sales	-	Today
2	B	Sales	-	Yesterday
3	A	Tech support	Sales	Two weeks ago
4	E	Tech support	Sales	-
5	D	Tech support	Sales, Computers	-

On a low level, the queue model is closely related to the "Greedy 2" policy described in the 2.4 Routing and queueing theory considerations section, but the queue structure looks different for every call. The incoming arrivals λ_x are the clients seeking specific skills, and there are servers with different capacities depending on how many agents has the specified skills as primary, secondary and so on. This makes the SBR model rather difficult to describe and simulate properly. However, some simulation trials were created using spreadsheets, with a wide variety or parameters to tweak. These spreadsheets took an incoming number, generated the incoming parameters and from these created a prioritized list depending on the mock agents' different skills. The spreadsheets highlighted the advantages and disadvantages of the different model alternatives and screenshots can be found in Appendix B. Because no queues are generally found in *Flow*, the area was not explored further and suitable for future SBR studies.

Two different flowcharts for how the client would interact with the system were also created. These two options describe different ways to ask the client for input when calling, depending on how much the system knows about the user. Due to their size, the flowcharts can be found in Appendix B. The first flowchart is a basic one where the user inputs what they want to discuss and then automatically is routed to a previously talk to agent if one is available. The second flowchart instead asks if the client want to discuss the same topic and they are then routed to the best suited agent. The second option can also include the possibility of on-hook waiting, a system where the previously talked to agent would call the client back when available. It is however not included in the flowchart since the system behaves in the same way with or without the on-hook waiting functionality. With this option enabled, the client does not have to stay on the line to reach the agent. The advantage of this option is that the concept of SBR is more exposed to the client, making it more of a feature. They do however add extra complexity. To evaluate which option is the most desirable for clients, a survey was planned for the elaboration phase.

4.3 Low-fidelity prototyping

To have a basis for the interactive medium-fidelity prototypes, sketching was performed as an early design step. The sketches were created with the epics as a functional basis for every interface page. As per the Nielsen Norman Group’s (2016) definitions in Table 1, this kind of prototype can be considered low-fidelity in terms of both visuals and interaction. Thus, it will only give a brief understanding of how the product will look like, and will not have any real functionality implemented.

A few iterations of low-fidelity prototypes had to be made to be able to achieve the results described in the sections below. Most early sketches that were made can be seen in Appendix D.

4.3.1 **Administrator interface**

The administrator interface is thought of as a three-step setup wizard used for the initial activation of an SBR, see Figure 16. The steps are also individually reachable in an intuitive way after this setup is performed, to configure specific parts of the system at a later time.

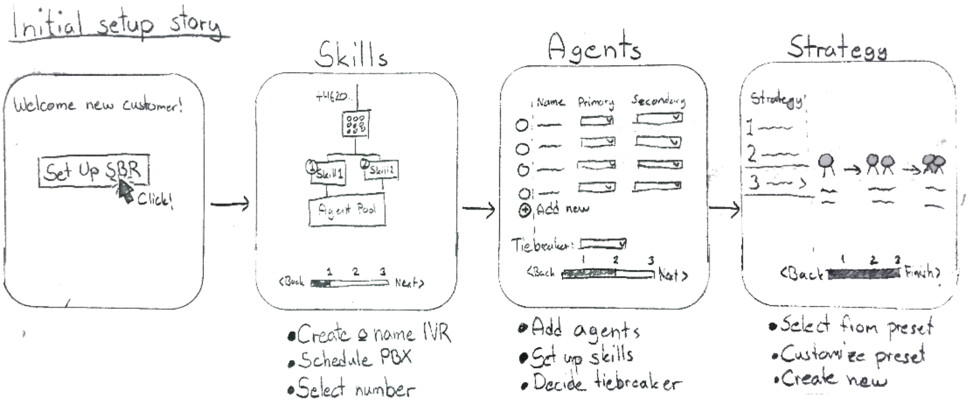


Figure 16. Storyboard over the initial setup. It shows a high-level overview of the different steps for setting up SBR and what is done in each step.

The first part of the setup is the *define page*. The administrator interacts directly with the tree, adding new IVR choices and skills as well as naming them. Adding a new node to the tree is done by clicking the numpad, seen at the top of Figure 17. Doing so will provide feedback by darkening the pressed numpad key and extending the tree to the right, as seen in the right part of the figure. The administrator then chooses to add either another numpad to provide nestled options, or a skill. A skill is then named by changing the label.

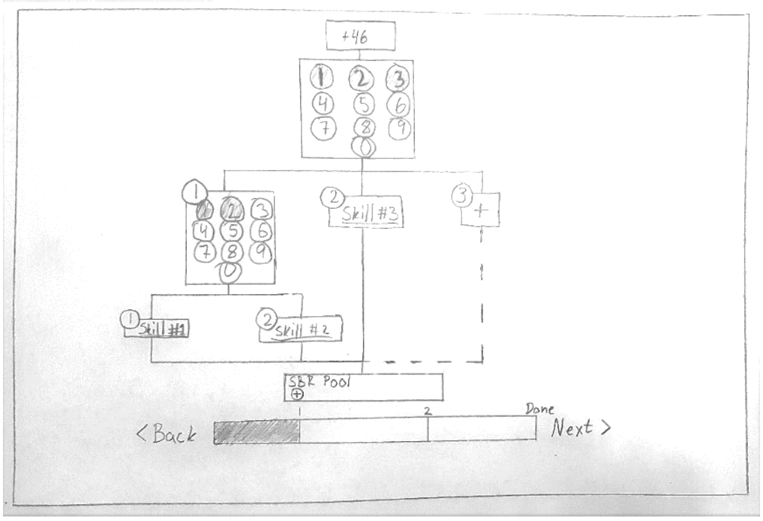


Figure 17. The define screen. The first part of the setup includes adding IVR choices and naming the different sections.

The next step of the setup is the *assign page*, as seen in Figure 18. Here the administrator chooses which agents to include in the pool, and the priority of their preferred skill sections. The interface uses the structure of a table with rows and columns, with the rows representing a specific agent and the columns their information and skill priority order. Agents are added by choosing from a drop-down menu, populated externally from the SBR setup by creating accounts in *Flow*. The skills created and named in the *define page* appear as options in the drop-down menus of the different columns. Adding a skill in the last available column, creates a new column. This ensures that unnecessary columns are hidden until needed to save screen space. The logic behind the tiebreaker drop-down menu at the bottom of Figure 18 is that the administrator can choose to use an internal or external system to decide ties between agent priorities. For instance, a company using the SBR can choose to use their existing customer relations system to decide the routing.

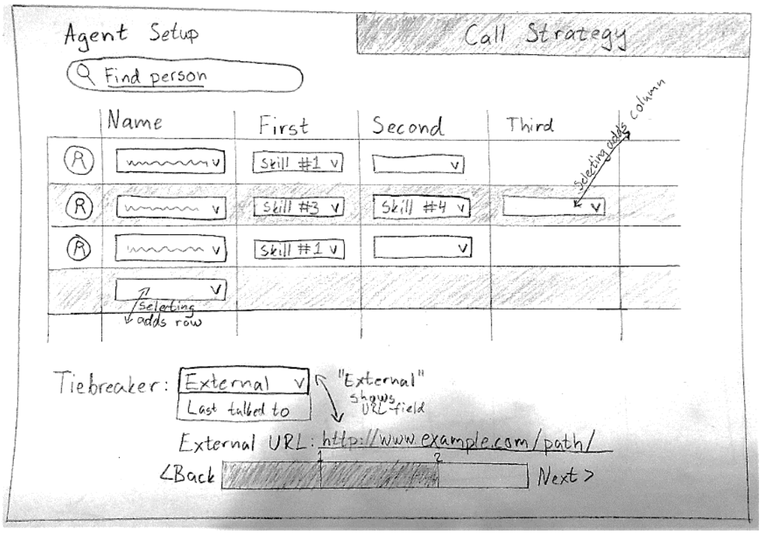


Figure 18. The assign screen. The second part of the setup included adding agents and setting up their skills.

The third and last part of the proposed setup is the *strategy page*, seen in Figure 19. This page is used to decide in which way to route the calls out to the agents, based on the prioritized list created by the model. The general concept is that a growing number of agents will be called over time, starting with the best suited agents. Each phase has a duration and number of agents, represented in Figure 19 as groups of people with a time underneath. The administrator can decide to use presets, or customize their own strategy. This gives the best suited agents a chance to answer the call, while ensuring that the queues do not get congested by introducing new servers as time passes. The notify call checkboxes under each phase is a setting so that the agents in the next phase would be notified that the call might soon be incoming.

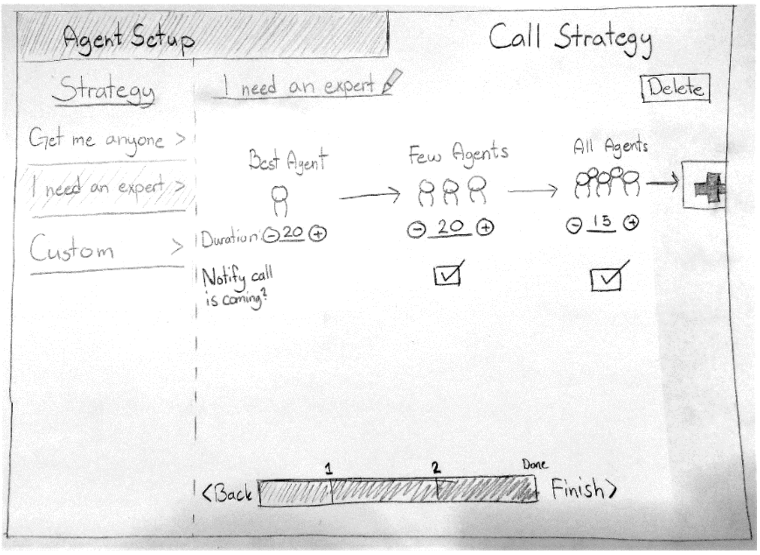


Figure 19. The call strategy page. The last part of the setup included choosing the strategy for which the calls will be distributed.

While reaching the *Agent and Call strategy pages* in the above sketches is done through the bottom navigation bar, they can also be accesses after the initial setup. This is done by clicking on the "SBR Pool" square at the bottom of Figure 17, which navigates the administrator to the page in Figure 18, however without the bottom progress tracker. This enables changes to be made after the initial setup is done.

4.3.2 **Agent interface**

The agents will have two different interfaces, one mobile and one web, which are related to each other. The designs were created with using the mobile first method as described in 2.2.2 Responsive design and mobile first.

Figure 20 shows a flowchart of how an agent can answer an incoming call as well as how to transfer the call to a mobile device. It describes on a high level how a call can be received on one device and transferred to another.

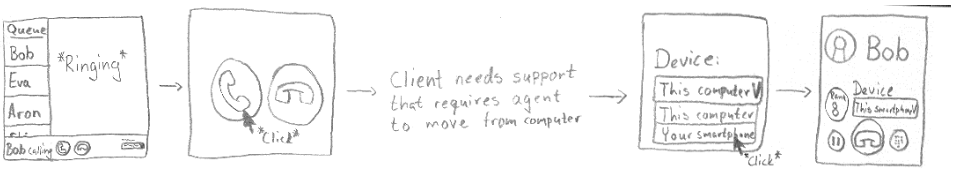


Figure 20. Flowchart for an incoming call.

The next flowchart, Figure 21 shows the main concept of how an agent can view the call history and then edit the agent priority for a certain client. This allows the agent to change how routing is done for a specific client, by swapping the order of assigned agents using a switch.

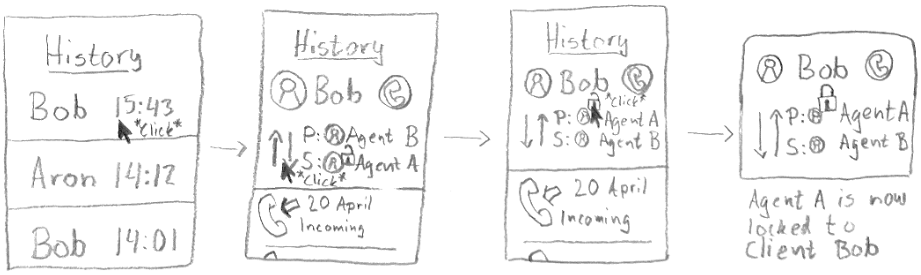


Figure 21. Flowchart for editing agent priority in the call history.

The agent web interface will display the same information as the mobile one, with screens such as history and call queue seen in Figure 20 and Figure 21. In Figure 22 the web version of the interface in Figure 23 is shown for an incoming call. In the mobile interface, some views must be hidden while on larger screens they can be shown, for example the call queue.

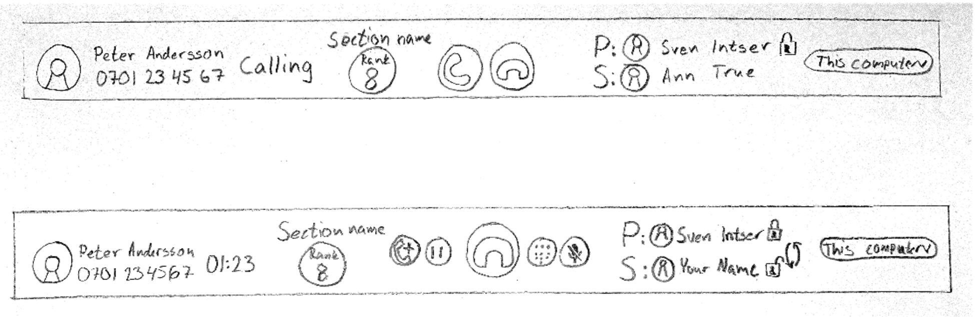


Figure 22. The bottom part of the agent web interface.

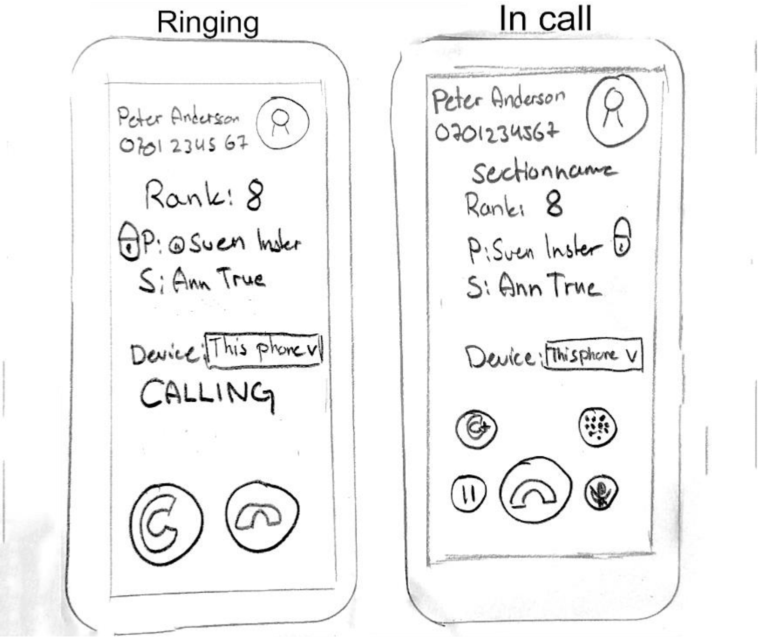


Figure 23. The mobile interfaces when the phone rings (left) and after a call is answered (right).

In Figure 23 the agent can see how good of a match they are to the caller by looking at the “Rank”. The “P” and “S” fields in the interface shows the agent which other agents are the best suited people to answer the call.

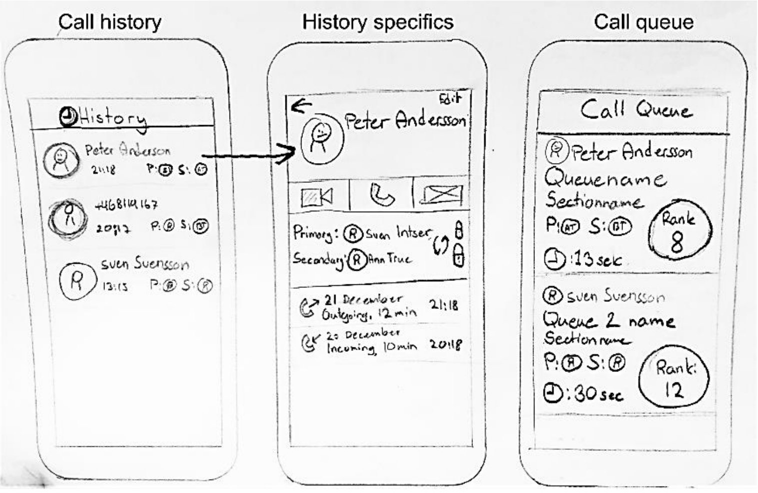


Figure 24. Agent mobile interface showing call history and live details of the different queues for the agent.

In Figure 24 the mobile interface can be seen. The call history page shows the recent incoming and outgoing calls for the agent in chronological order. To see more details about a specific caller, an entry can be pressed. This leads to the history specifics screen, showing contact information and interactions with this client, for example a list of all previous calls.

The call queue displays, in real time, who is first in line in the different queues that the agent is part of. The agent can see his or her rank to answer this call, and how long it will take before the call will propagate to them and their phone will ring. As described in the system model, a call will propagate and hunt down more and more agents if it is not answered by anyone.

4.4 Low-fidelity evaluation

The low-fidelity prototype was evaluated through expert discussions. All the sketches from the prototype were presented to one of the UX experts at Telavox and discussed. All employees at Telavox are also end users, including the UX expert, as they use *Flow* every day for internal communication. The navigation flow between each sketch was described together with the storyboards that were created.

There were iterations and propositions made before finally completing the low-fidelity prototype. Most of the iterations can be seen in Appendix D. The results have been shown in section 4.3 Low-fidelity prototyping.

The biggest hindrance identified was providing enough options for the administrator, while keeping the UI clean and understandable. It was decided that some of the feature would have to be discussed and tested through the medium-fidelity prototype, as they require more interaction to be evaluated.

One of the issues found with the *Call strategy page* was describing in a transparent manner how many agents were part of each phase. Because a list is generated uniquely for every caller, depending on the skills they require, the phases contain different total number of agents for every call. This could cause an administrator to define phases based on their conceptual model of the SBR pool, but, the skill assignments might make actual routing completely different.

The second issue with the *Call strategy page* was understanding the concept of notify. Notify was meant as a feature that would tell agents which calls would soon reach their phones, and inform them what the call is about. However, trying to convey this concept in the setup proved to be difficult.

5 Elaboration Phase

During the elaboration phase, the interfaces went through another iteration, resulting in a medium-fidelity prototype. A survey was created and distributed to gather client expectations when calling an SBR system.

5.1 Medium-fidelity prototyping

The first part of the elaboration phase was dedicated to medium-fidelity, or mid-fidelity prototyping. This expanded on the low-fidelity prototype with the feedback from the previous evaluation. This prototype was defined as a mid-fidelity prototype since it doesn't meet up with all the criteria for a high-fidelity prototype seen in Table 1. The mid-fidelity prototypes add the aspect of interaction to the low-fidelity prototype, making them more suitable for user testing. However only limited amounts of other changes were made compared to the low-fidelity prototype. The tool used to create the mid-fidelity prototype was a web application called *Ninjamock*. *Ninjamock* is an application that combines the handmade sketch feeling with interaction through click navigation on desktop.

The mid-fidelity prototype was divided into two parts, the administrator interface and the agent interface. The administrator interface was once again a three-step setup that an administrator of the system must go through to be able to set up skill based routing. When skill based routing has been set up the agents should be able to use the agent interface.

The agent interface is the view the agents are met by when interacting with the system. This interface describes what the agents are able to do before picking up an incoming call, during a call and after a call.

5.1.1 **Administrator interface**

The first view the user was presented with viewing the mid-fidelity prototype was the structure page, seen in Figure 25. The structure page described how a phone call can be directed through the system in a hierarchical way. Since this was a mid-fidelity prototype, most of the items are already added and the user could only add and remove one of the skills.

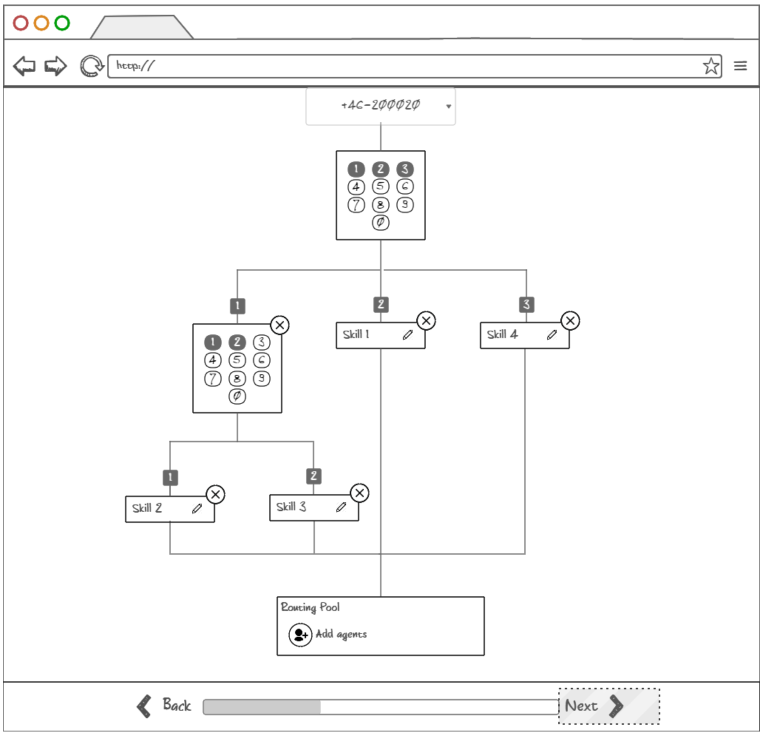


Figure 25. The mid-fidelity version of the structure page.

At the top of the structure page the B-number was presented. The B-number was connected to a keypad which represents the IVR choices a calling client can make. When the administrator presses one of the numbers the underlying tree structure expands with the number of branches the administrator has pressed, as seen in Figure 26. After expanding the tree structure with a branch, the administrator must choose what kind of item should be in that branch, for example an agent skill or another IVR keypad.

All possible paths a phone call may take in the system eventually end up at an end node which always is an agent skill. All end nodes are then connected to one final node, the routing pool.

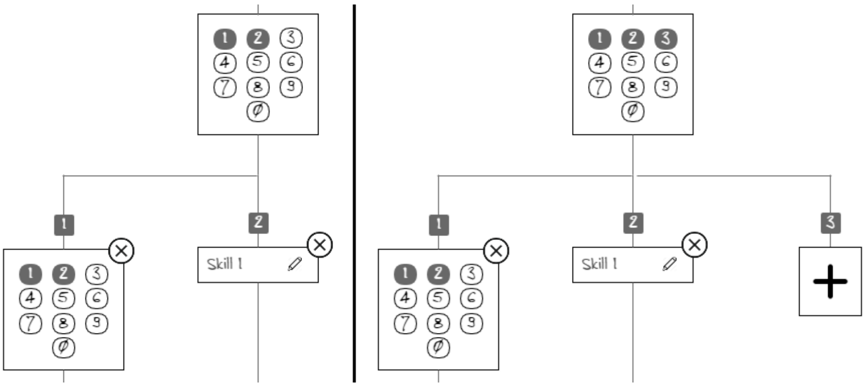
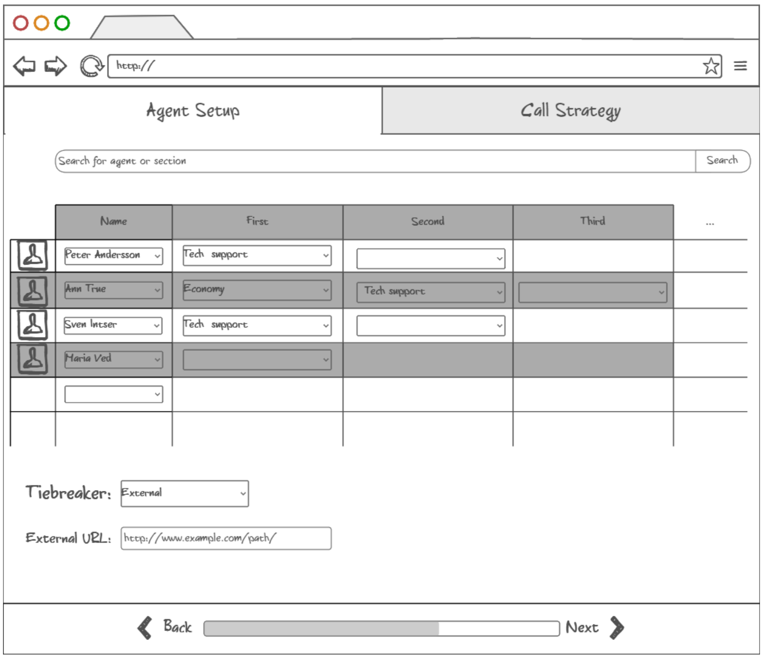


Figure 26. ***Structure page*** **- The tree expands as the user enables more IVR choices.**

When the administrator has finished creating the call routing structure of the system, the next step in the setup is the agent setup, seen in Figure 27. The setup has different columns as in the low-fidelity prototype. Drop down menus enables the administrator to add skills to the different agents. Only one skill is allowed per column, and each column represents a hierarchy level. This means that when a call reaches the SBR, the agents with the required skill in their "first" column will be called first, then agents with the skill in the "second" column and so on. At the bottom is the tiebreaker system picker, that could be set to external or to use the last talked to index as described in 4.2.3 Model results.



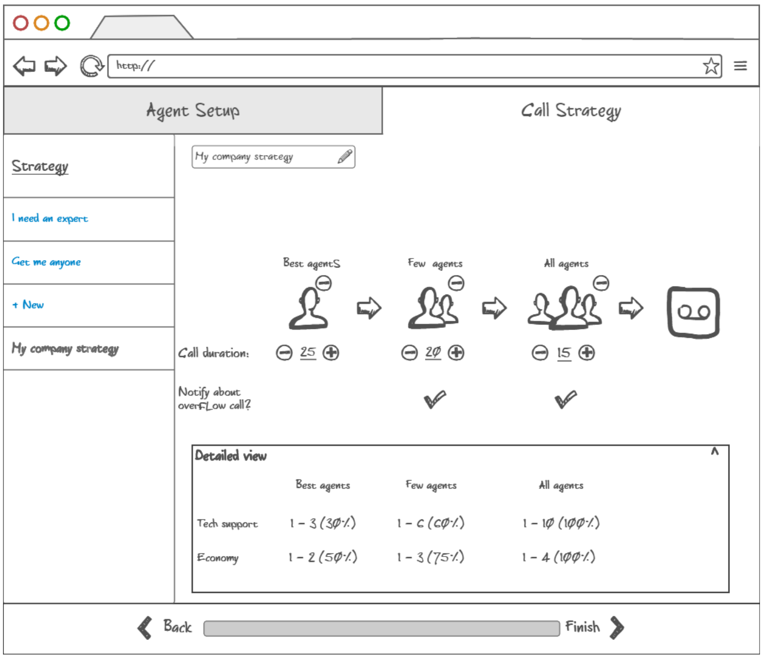
Name	First	Second	Third	...
Peter Andersson	Tech support			
Ann True	Economy	Tech support		
Sven Intser	Tech support			
Maria Ved				

Tiebreaker: External

External URL: http://www.example.com/path/

Figure 27. The mid-fidelity version of the agent page.

The last page of the setup is the call strategy page, seen in Figure 28. The administrator can select from different preset strategies, or create his own by defining phases. Each phase has a number of agents and a duration, as well as an ending phase. In the case of Figure 28 the end phase is set to voicemail. At the lower part of the call strategy page is the detailed view, attempting to show the administrator how many agents will be called when each of the different skills are required. This detailed pane was added because of the low-fidelity evaluation, where the issue with phases and number of agents in each skill was identified.



My company strategy

	Best agents	Few agents	All agents
Call duration:	25	20	15
Notify about overflow call?		✓	✓

Detailed view

	Best agents	Few agents	All agents
Tech support	1 – 3 (30%)	1 – 6 (60%)	1 – 10 (100%)
Economy	1 – 2 (50%)	1 – 3 (75%)	1 – 4 (100%)

Figure 28. The mid-fidelity version of the call strategy interface.

5.1.2 **Agent interface**

While the low-fidelity prototype provided the basic understanding of which buttons and functionalities that should be present, this mid-fidelity prototype shows how they can be interacted with.

When using the agent interface, the user is first presented to a view where a client can choose to answer a call. As seen in Figure 29, the agent has accepted the call. When comparing this view with the one created in the low-fidelity prototype, see Figure 22, there are many similarities.

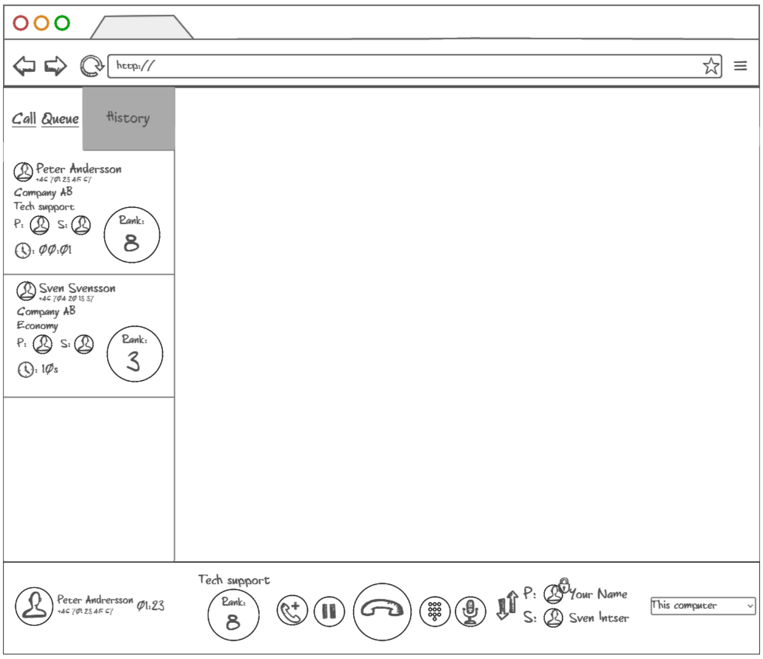


Figure 29. The mid-fidelity version of the agent in-call interface for the web.

The view placed to the left, displaying the call queue in Figure 29 can be replaced by pressing the “History” tab. This brings up a list of earlier conversations with some information about the call made. This can be seen to the left in Figure 30.

On a mobile device, it would not be possible to show the in-call interface together with the call queue or history tab since there is a limited available screen space. This is solved by only showing one view at once on mobile devices, limiting the prototype’s multitasking capabilities on mobile devices.

The History view and the History Specifics view, as seen in Figure 24 provided a solid base for developing the views seen in Figure 30. Some changes have been made to the interface, the tabs are a central change for example, but most of the functionalities are more or less the same here as well.

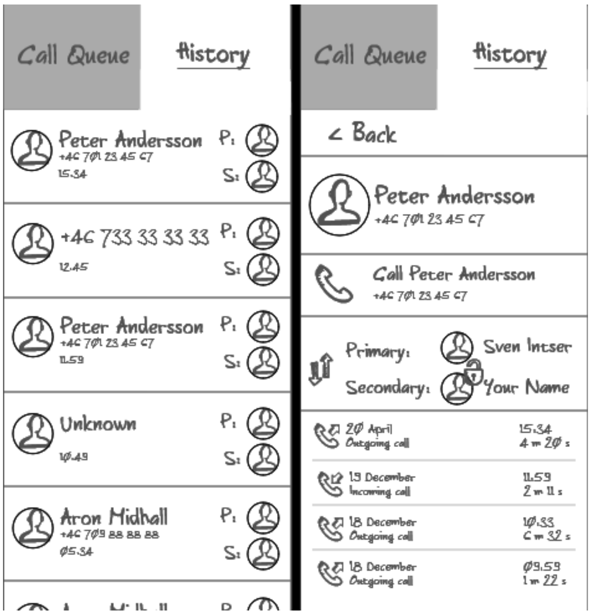


Figure 30. The mid-fidelity version of the call history views.

When looking at the mobile version of the in-call interface, in Figure 31, all the calling functionalities are present in this view as in the web interface. The difference in this view between the web and the mobile device, is that the content is aligned top to bottom in the mobile interface instead of from left to right.

The buttons related to the call, hanging up, muting and bringing up the numpad for example, are aligned at the bottom in the middle since they are rather central when interacting with the prototype.

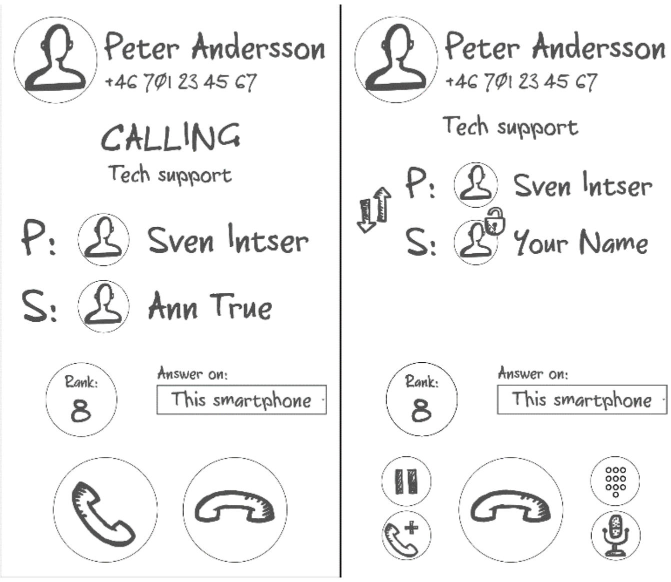


Figure 31. The mid-fidelity version of the in-call interface on a mobile device.

5.2 Mid-fidelity evaluation

When the mid-fidelity prototype was considered completed, two types of evaluations were performed, one where the system and the prototype was presented for a group of six UX specialists on Telavox and one small user test.

5.2.1 **UX designer presentation and demonstration**

The prototype was presented for a group of six UX specialists on Telavox. As Telavox uses *Flow* for internal communication, they can all be considered end users of the system. The group consisted of five male and one female participant, all between the age of 25 and 35. The male participants had been working with *Flow* for about 3 years, while the female participant one had 2 months experience with the system. The group did not have any prior knowledge of SBR, and the general concept was introduced together with the different design considerations. A lot of useful material explaining SBR was taken from the presentation that took place earlier in the process and is explained in chapter 4.2.1 Presentation.

In addition to the concept and model of how SBR should work the presentation included pictures of the mid-fidelity prototype and how it relates to the system model and concept. After the presentation, a quick demonstration of the mid-fidelity prototype took place. This demonstration showed the prototype in a more in-depth manner and complemented the model. These UX specialists were not part of any later evaluations.

The entire meeting including the demonstration and presentation was more of an open forum, for discussing every part in detail. This added a good understanding of how easy or hard it is to understand the different parts of the system for persons that have not yet been introduced to SBR. The findings of the discussion are described in the evaluation results.

5.2.2 **Mid-fidelity Testing**

The mid-fidelity prototype was tested on three participants through exploratory testing. The participants were all users outside of Telavox, that did not have any prior knowledge about *Flow*. They were all engineering students at Lund University between the age of 20 and 25. To perform an exploratory testing session, both the test leader and the participant should be active when it comes to explaining what is happening, what the participant is thinking and how to make changes to the system to make it easier and more intuitive (Rubin & Chisnell, 2008). It should be considered that the prototype was slightly limited in terms of interaction, because of the nature of the tool used to create it.

The test session consisted of two parts, the first part evaluated the administrator setup and a second part that evaluated the agent views. The setup was a desktop computer with the projects in the application Ninjamock displaying. Due to availability, the tests were conducted in home environments using laptop and desktop computers. The test leader was present during the test and had a short agenda of tasks to be explored. There were no time limitations for the tasks that were to be made. The goal of the test was to evaluate the overall understanding of the SBR concept, as well as discuss the different ways that it was realized in the prototype.

5.2.2.1 Administrator tasks

The participants were asked to set up a new SBR system through the prototype. They were not given clear instructions, but knew that it included defining skills, assigning them to agents and choosing a call strategy.

The participants were asked to complete the setup process and then look around the interface to verify that everything was saved as expected.

5.2.2.2 Agent view tasks

The agent view was more limited when it came to ways to interaction, and therefore offered less options for the participant to choose from.

The test consisted of a few tasks starting with looking at the call history. The participants were instructed to check how many times a conversation has taken place with a specific client and set the participant as the client’s primary agent.

The test simulated a call coming in to the system, where the participant could answer it and perform different actions. The actions available was to switch the agent priority, lock the agent priority, answer and hang up the call. The participant was also asked to finish by transferring the call to a smartphone and perform the same tasks again.

5.2.3 **Mid-fidelity evaluation results**

5.2.3.1 UX designer presentation and demonstration

A lot of good input and thoughts were generated for the prototype. The consensus was that all the important interfaces were in place, however the flow and structure of the setup could be slightly modified. One of the major points to change in the prototype to make the design more intuitive, was to make the it clearer for the administrator that the defined skills have a big impact and by presenting it to the administrator in another way. A suggestion to do this was to change the order of the setup, so start off with defining skills. This could also save the administrator the trouble of creating and managing the tree structure, as it could be auto generated instead.

New suggestions of the structure page in the setup (Figure 25) was created right after the meeting and can be seen in Figure 32. These proposals suggest that the tree structure should be replaced with a list hierarchy. The hierarchy consists of skills to focus the attention of the administrator to the creation of them thus lifting the perceived importance of the skill concept.

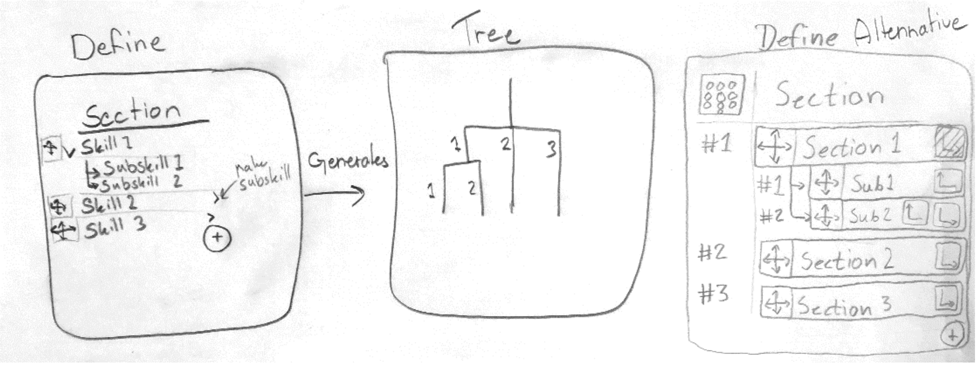


Figure 32. New suggestions for the structure page.

5.2.3.2 Mid-fidelity Testing

The exploratory testing sessions gave a lot of useful information about how to improve the system and how to change certain elements to make them more intuitive.

Worth noting is that due to the prototypes limited graphical fidelity, it made some elements that were interactive not that affordable. This could sometimes be seen on the participants as they searched for a clickable object since the mouse cursor changed shape when hovered over an object that could be interacted with.

5.2.3.2.1 Administrator setup results

The main issue for the participants considering understanding the system setup and navigating through it was that some parts of the system already had been defined from the beginning. For example, three of the four IVR choices were in the UI from the beginning and three agents were already put in the list and the call strategy was almost completed as well. This was likely confusing to the participants because a lot of unknown information directly was presented on each page. If each page would have been empty from the beginning and the participant would populate it with relevant information themselves it would be easier to understand.

To easier understand that this is a three-step setup, it would be good to have one step before seeing the structure page, that informs the user of what they are about to do.

The tiebreaker option was overall confusing, a proposition was that it should be moved to the Call Strategy page instead. Removing the setting from the setup is also an option, but undesirable because of its high business value in an SBR system.

The Call Strategy was also considered as a confusing page. It presented the user with many options and choices, which felt slightly overwhelming. An option would be to hide all the details of the preset strategies. The user could then actively choose to use a more advanced call strategy editor if so is desired.

5.2.3.2.2 Agent view results

The participants found it challenging to understand all the different concepts and options. This could be seen especially when the participants were looking at the call history.

There are two history views, one for all calls and one for each individual client call. The participants commented that this needs to be indicated more clearly through a heading text or bigger change to the interface, as they looked quite alike.

The priority switching on a specific client was intuitive and the locking function was understandable when explained. The participants understood that each client had its own priority and that agent could manage the call for them individually. However, the meaning of the letters “P” (Primary) and “S” (Secondary) letters where not completely clear. It was suggested that they should be written out or show some kind of help text. Without knowing what they stand for, it was difficult to understand what position in the interface represented to what priority.

5.3 Survey

To get input from clients, a survey was created. The goal of the survey was to evaluate what clients think about different aspects of customer support systems, resulting in a decision for the flowcharts described in 4.2.3 Model results. This was needed to make sure that the system behaves as expected from a client's point of view. The client's experience is a very important factor for the SBR system. Even if this system is designed for efficient routing, bad UX would result in a frustrating client experience, defeating the purpose of SBR.

5.3.1 **Content**

The survey brought up the following topics:

- Preferences for an agent's characteristics or profile.
- The client's usages of support systems, such as preferred support type and frequency of support contact. It also asked when telephone support was preferred and expectations while waiting in queue.
- How long the client is prepared to wait, if at all, to get better service or talk to the same agent again.
- How clients imagine an SBR system would work in terms of remembering IVR choices or agent relations.
- Frustration with phone support as well as other personal comments.

A more detailed view of the contents of the survey can be seen in Appendix E.

5.3.2 **Responses**

The survey collected 70 answers which were then summarized. Values have been rounded to whole numbers for better readability. A raw summary of the responses can be viewed in Appendix F.

5.3.2.1 Participants' profiles

The results showed that a majority of the responding participants were in between 21-25 years old. The distribution between men and women was relatively equal. The number of answering men was 57% and women 41%. 2% of participants declined to expose their gender.

5.3.2.2 Today's support systems

When asked about preferred support systems, the results were as follows:

1. 73% of respondents were positive to using an online chat support.
2. 64% of respondents were positive to using email support.
3. 54% of respondents were positive to using telephone support.
4. 49% of respondents were positive to using questions and answers online. However, a lot of respondents were indifferent to this option.
5. 36% of respondents were positive to using social media as support.
6. 36% of respondents were positive to using in-person support.

The results showed that the majority of the participants, 71%, contacted support via telephone around once per year. Very few participants, 16%, contacted the support once per month while 10% never contacted support via telephone.

When the clients were asked for which reasons they chose to contact a support via telephone, the answers showed the following results:

1. 61% answered "If my issue is complicated".
2. 57% answered "If I want quick answers".
3. 53% answered "If other options didn't solve my issue".
4. 50% answered "When other alternatives are not available".

When it comes to what kind of auditory feedback the clients prefer, the preferences varied a bit as seen in Figure 33 below.

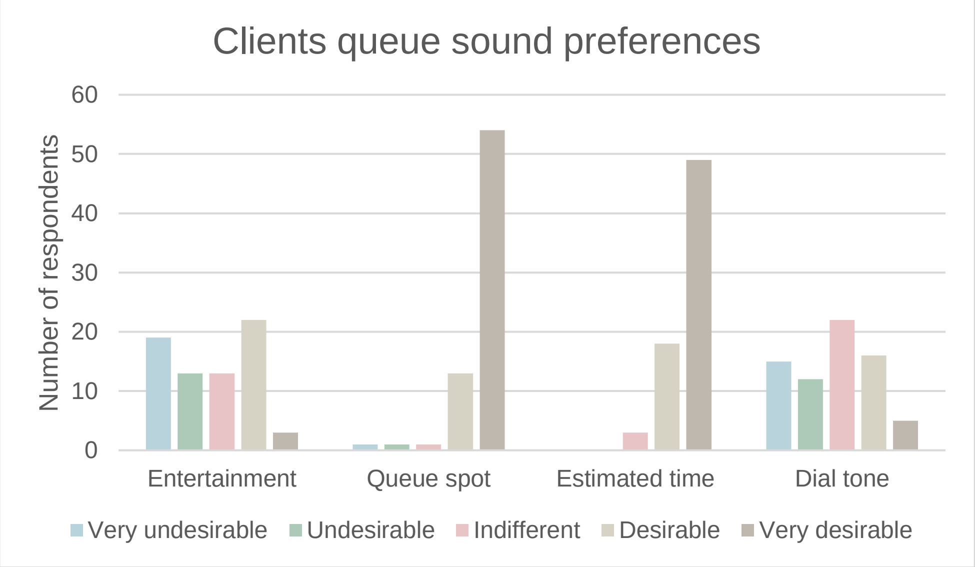


Figure 33. The survey participants desired audio feedback when waiting in a queue, shown as number of respondents.

5.3.2.3 Future PBX systems

The majority of the respondents, 73%, answered that they would rather wait for a more knowledgeable support than get support fast and talk to anyone.

Most respondents were positive to waiting a bit longer to talk to the same agents they had contact with previously about the same issue. When calling about a different topic that is not the agents main area of expertise, roughly 60% where indifferent to waiting. However, if it was about an issue outside of the agent’s expertise, 63% were negative to waiting.

The majority of the respondents answered that they would want to wait maximum ten minutes additionally to get an agent that they already had been talking to, as seen in Figure 34.

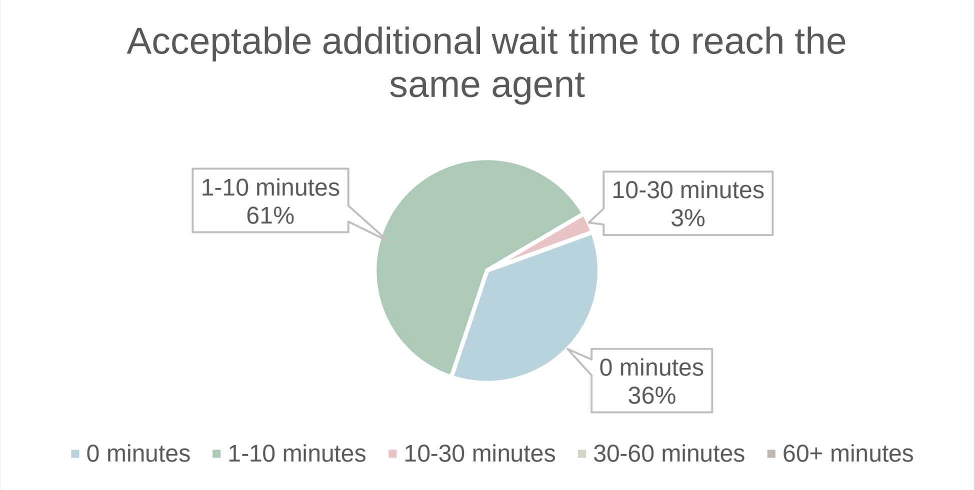


Figure 34. Acceptable additional wait time to reach the same agent, described in percentages of all 70 participants.

When on hook waiting was offered as a waiting option, the results were a bit more spread out, as seen in Figure 35. On hook waiting means that the client can wait without having an ongoing phone call open and the agent would instead call back when it is the client’s turn to get assisted.

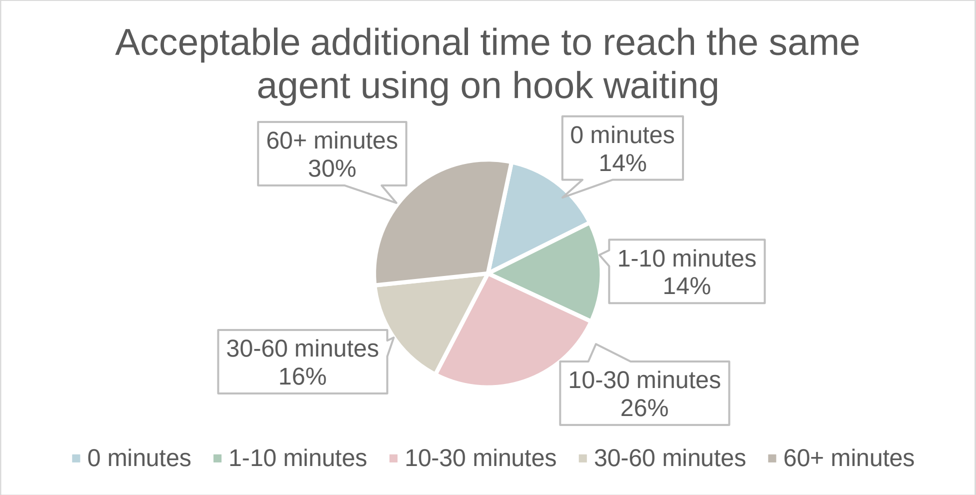


Figure 35. Desired wait time to reach the same agent if on hook waiting is allowed, described in percentages of all 70 participants.

When asked how the PBX system should act when the client calls for the second time, even though the issue may not necessarily be the same as before, the answers varied a bit. 53% answered that they would prefer if the system asks if the client wants to be put in the same queue as before. 27% answered that they would prefer that the system to read all IVR options, just like the first time they called. 21% answered that they would prefer if the system automatically put them in the same queue and prompt them to change the queue.

5.3.2.4 Thoughts about telephone support

The respondents had roughly the same opinion about the most important factors in a PBX system. First of all, the results showed that the most important thing is that the agent is pleasant to talk to and is knowledgeable. The second most important thing is to be able to solve the issue on first contact. Only a few respondents answered that it was actually important to be able to talk to the same agent as in an earlier conversation.

The following list ranks respondents' biggest annoyances when calling a support. The list is sorted from most to least annoying.

1. Not being able to resolve my issue.
2. Talking to an unknowledgeable agent.
3. Being presented to a complicated IVR system.
4. Having to contact the support regarding the same issue again.
5. Long waiting time in the queue.
6. Not being able to reach a person to talk to.
7. Being passed between agents.
8. Being kept waiting on hold.

5.3.2.5 Other comments

After filling out the form, some of the respondents wrote optional comments about the subject. The on hook waiting option brought up some discussion and many participants were positive.

Chat support solutions was also desired. With this option, the client does not need to actively wait in the queue and it is easier for the client to multitask. A hybrid chat solution was suggested, where the client would be able to start a ticket by contacting the support via chat. If the client feels the need to call the support, it should be easy to resume the issue on the phone.

One comment was about visually showing the client which IVR alternatives are available on the screen. The client then doesn’t have to wait and listen to all the different alternatives.

5.3.3 **Discussion**

This section will discuss the results from the survey presented above, identifying possible causes for the answers and some comparisons. Some improvements, and suggestions for future studies will be shown here as well.

5.3.3.1 Participants' profiles

The ages of the respondents were not evenly distributed. The majority of the respondents were young, which might have an impact on the accuracy of the results. No data was found supporting the claim that younger clients are less likely to call for customer support. However according to Microsoft (2015) the option has becoming less used in recent years, possibly caused by a generation shift. The distribution between men and women is rather even and the small difference is probably not going to affect the outcome significantly. According to Miratix, Sekhon, Theodoridis and Campos (2017) one way to combat this would be to use weights so balance the data. However, while sometimes a good option, it can also affect the result in a negative manner, especially in smaller sample sizes (Miratrix, Sekhon, Theodoridis, & Campos, 2017).

5.3.3.2 Today's support systems

The chat and email support were the most favorable support options. According to the survey, most participants do not call phone support more than once a month. This is possibly affected by the participants age, as mentioned earlier. However, this data is inconclusive as their number of issues on a monthly basis is unknown.

Telephone support seem to have some advantages. The main reason why the respondents would like to call a support would be if the issue is complicated or if they would like fast responses. Phone support is likely to be fast and questions that arise can be answered directly.

Almost every respondent answered that they would like to know which spot in the queue they had when calling a support. This gives a better overview and transparency of the system and makes it easier to estimate if it is worth waiting. This is therefore considered a desirable feature for the SBR system. The same goes for estimated waiting time. Almost every respondent wanted to know the estimated queue time. Since every company's call length is different, the waiting time is different even if a client has the same spot in the different queues. Adding audio feedback for both these options is hence preferred.

5.3.3.3 Future PBX systems

Almost three quarters of the respondents answered that they would rather wait for a more knowledgeable agent than being connected to anyone. Skill based routing will make sure that a client gets connected to a more knowledgeable agent. If a client gets connected to an agent fast, but needs to be transferred, the total time in the system will probably be longer than if the client must wait longer for a more knowledgeable agent. So, there is a trade-off between what some of the clients want and the efficiency of the system. However, the majority of the respondents are still open to waiting some additional time on the line.

Almost every respondent wanted to talk to the same agent they talked to earlier if it concerns the same issue as in an earlier interaction and would be willing to wait roughly ten minutes to do so. This is positive as it is what SBR is trying to achieve. However, if the client is calling about a new issue they are mostly indifferent to which agent they talk to. The big and important issue is therefore creating a system that effectively can identify what the call is about.

The responses considering on hook waiting was scattered, but they would in general accept much longer waiting times than without the option enabled. Many participants could imagine waiting over an extra hour to get in touch with the same agent. The reason why most people would be able to wait much longer is that the responsibility of keeping the spot in the queue is no longer theirs. The caller doesn’t have to actively wait and listen to the feedback and can instead do other things. It also removes the issue of disconnects while waiting, a growing problem as mobile phones are used over tethered phones.

When calling most PBX systems today, the system repeats the procedure to list all available IVR choices every time. However, over half of the respondents wanted the system to ask if the call is about the same issue instead, possible speeding up the process. While not certain, it could be because clients often do not call a company about many different issues.

5.3.3.4 Thoughts about telephone support

Microsoft (2015) compared the four most important aspects for a satisfying customer service experience in the United States, United Kingdom, Brazil and Japan. All countries agreed on the ranking of which was the most important down to the least important aspect, but the counties had a slightly different percentage (Microsoft, 2015). This was the result of the ranking Microsoft's survey presented:

1. Getting my issue resolved quickly.
2. Getting my issue resolved on first contact.
3. A friendly and knowledgeable customer service agent.
4. Being able to find the information I need without assistance.

This survey was performed only on Swedish participants and comparing it to Microsoft's survey it gives a slightly different ranking:

1. A friendly and knowledgeable customer service agent.
2. Getting my issue resolved on first contact.
3. Being able to find the information I need without assistance
4. Getting support from the same agents I have already talked to.

This discrepancy tells us that the results are probably not generalizable on other nationalities or client groups.

5.3.3.5 Other comments

The most commented part of this survey was the on hook waiting option. As seen on the results of the question about this topic, many respondents were positive to this option. It however puts some responsibility on the system to be able to estimate the time it will take before a call to the client can be made. If the client for some reason doesn't get any call back or if the time it takes before the call to the client is much longer than the told estimated time, the client's customer service experience drops significantly. Overestimating the call time could be way to combat this problem, but can also cause unwanted effects of clients hanging up prematurely.

The comment about displaying the IVR alternatives on the display is a very interesting suggestion. This will probably give a better overview of the IVR hierarchy, making the navigation easier for the client. There is however a limitation with this since it must likely be done through an application, which was disregarded early in the project. This is because the standard calling applications on smartphones does not support such features today.

According the survey comments, integration with CRM systems is desirable from a client's perspective. While actual implementation of such a system is deemed out of scope for the project, it is an interesting field of future studies.

Based on the results from the survey and the positive comments about on hook waiting, it was decided that the third flowchart option was the way to go for client system interaction. This includes asking the client if they are calling about the same issue, and then offering on hook waiting. However, from an administrator and business perspective, this feature needs to be optional as it is more directed to a call center customer. In the case of a PBX for dinner reservations or other bookings, the feature is not as applicable or desirable.

6 Detailing Phase

The following section describes the last phase of the project, featuring the creation and evaluation of the high-fidelity prototype. A test plan was developed for the evaluation, describing how the unmoderated remote testing should be performed.

6.1 High-fidelity prototyping

The high-fidelity prototype was created based on the results of the mid-fidelity evaluation. Since a lot of useful data had been gathered and put together from the earlier iterations, this high-fidelity prototype could be more centered around the design.

This prototype's content was created in an application called *Sketch*. In *Sketch*, all the screens' designs were created and managed. Some content, like icons for buttons, was taken and inspired from Google's material design guidelines. Some assets were also inspired or taken from Telavox's already existing prototypes, to match the existing products.

When the screens were completed, they were exported to the web application InVision. In InVision the screens were linked together with hotspots. A hotspot is an area on a screen leading another screen, adding interactivity to the static prototype screens.

6.1.1 **Administrator interface**

The administrator interface was turned into a more streamlined, three-step setup with an onboarding guide to help users understand the new features introduced with SBR. The onboarding process was added because of the results in the medium-fidelity evaluation, where it was seen that the concept of SBR was not completely clear to participants. This onboarding guide gives the administrators some context to the new features and the different options they will be presented with. The onboarding process also consists of three screens as well as a call-to-action button guiding the user to the new setup process, seen in Figure 36. It was inspired by the material design onboarding guidelines created by Google and describes the SBR system and the setup process (Google Inc., 2017).

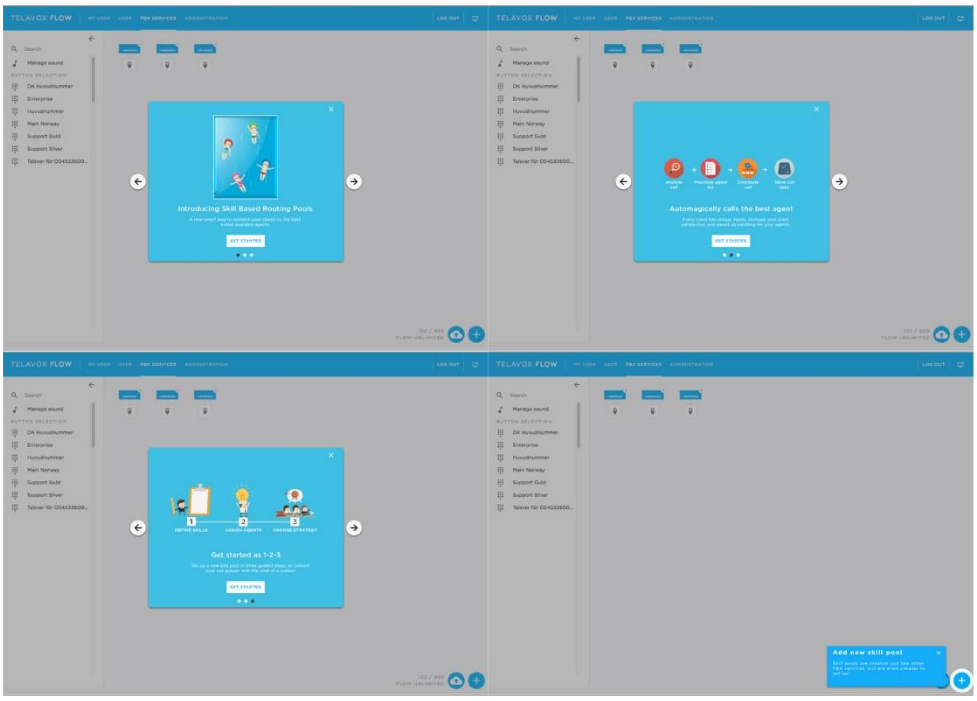


Figure 36. The onboarding screens for the administrator interface.

As inspired by the mid-fidelity testing and UX team discussions, the tree style of setting up the skills was changed to a simpler hierarchy view, seen in Figure 37. Skills are added to the list and in turn automatically generates a tree structure, that can be seen in a small preview window. When the button "Add new skill" is clicked, a small dialog appears prompting for the skill name and parent.

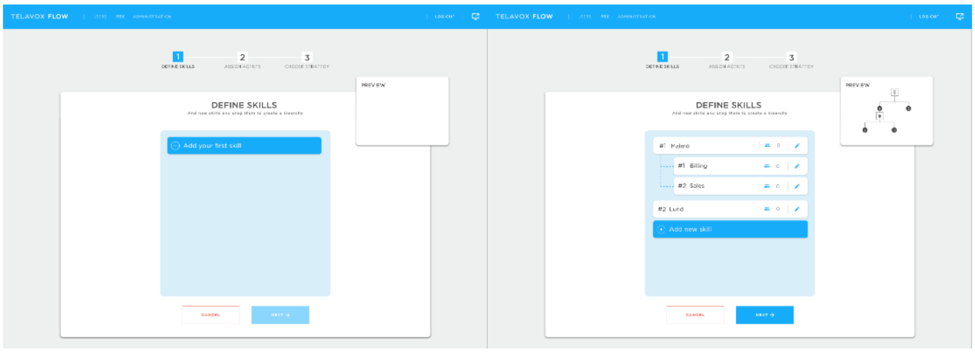


Figure 37. The skills setup page, unpopulated to the left and populated to the right.

After defining the skills, the administrator can add agents to the SBR pool and assign them skills as primary or secondary priority. While adding more levels of skills is still supported in the model, it was removed from the interface for complexity. The motivation was that three or more hierarchal levels of skills are likely unnecessary to present for most users, as found in the mid-fidelity tests. The left image in Figure 38 shows an administrator selecting and adding all the remaining available agents to the pool. The right image in Figure 38 shows the administrator assigning a skill. The skills are seen at the top of the page, and can be dragged to an agents primary and secondary cell. The available places to drop the skills are illustrated by blue dashed lines, providing assistance through closure described in 2.1.1.4 Closure.

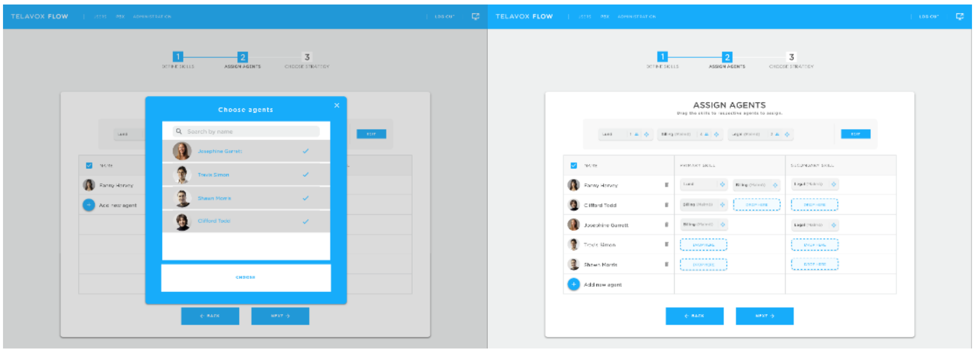
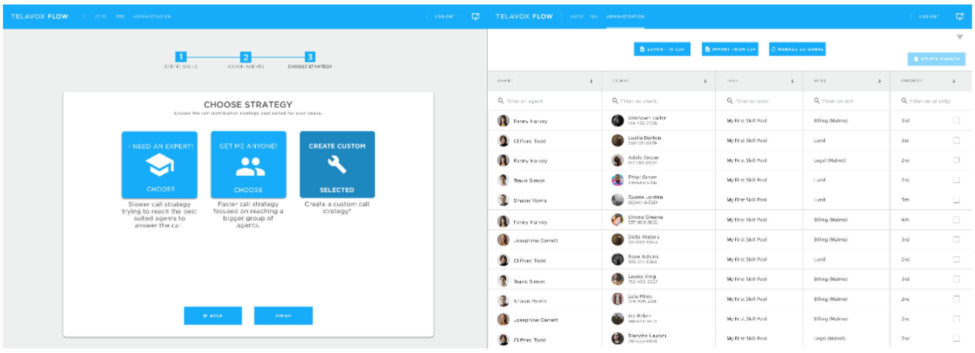


Figure 38. The skill assign page, with a prompt to select agents to the left and the drag and drop indicators on the right.

As per the mid-fidelity evaluation, the advanced call strategy setup was replaced with two presets, "I need an expert" and "Get me anyone", representing a slow and quick phases call strategy respectively. A third option "Create custom" strategy was added, but the actual design and implementation of this was left for future studies. These choices can be seen on the left in Figure 39. The right side of Figure 39 shows an overview of the SBR systems call memory. The purpose of this was to visualize the concept of each pool having a call memory table. The tiebreaker concept for external systems was removed from the interface, and only the default system of a last talked to index was used instead. This was also motivated by lower complexity of the interface.



CHOOSE STRATEGY

Choose the call distribution strategy best suited for your needs.

Slower call strategy trying to reach the best suited agents to answer the call.

Faster call strategy focused on reaching a bigger group of agents.

Create a custom call strategy"

AGENT	CLIENT	POOL	SKILL	PRIORITY
Filter on agent	Filter on client	Filter on pool	Filter on skill	Filter on priority
Fanny Harvey	Unknown caller	My First Skill Pool	Billing (Malmö)	3rd ☐
Clifford Todd	Luella Burton	My First Skill Pool	Lund	1st ☐
Fanny Harvey	Adele Green	My First Skill Pool	Legal (Malmö)	2nd ☐
Travis Simon	Ethel Green	My First Skill Pool	Lund	2nd ☐
Shawn Morris	Gussie Jordan	My First Skill Pool	Lund	5th ☐
Fanny Harvey	Elnora Greene	My First Skill Pool	Billing (Malmö)	4th ☐
Josephine Garrett	Della Waters	My First Skill Pool	Billing (Malmö)	3rd ☐
Clifford Todd	Rose Adkins	My First Skill Pool	Lund	2nd ☐
Travis Simon	Leona King	My First Skill Pool	Billing (Malmö)	3rd ☐
Shawn Morris	Lola Miles	My First Skill Pool	Billing (Malmö)	2nd ☐
Josephine Garrett	Ina Baker	My First Skill Pool	Billing (Malmö)	2nd ☐
Clifford Todd	Blanche Lawson	My First Skill Pool	Legal (Malmö)	2nd ☐

Figure 39. The call strategy screen to the left and the tiebreaker data table to the right.

When the entire setup is completed, the call routing tree is created and a hint is displayed guiding the user to the call memory page, seen in Figure 40. If the pool located at the bottom of the tree is clicked, the user is navigated to the pool settings, seen to the right in Figure 40. These settings are the same as in the three-step setup, containing skill setup, assign and call strategy picker.

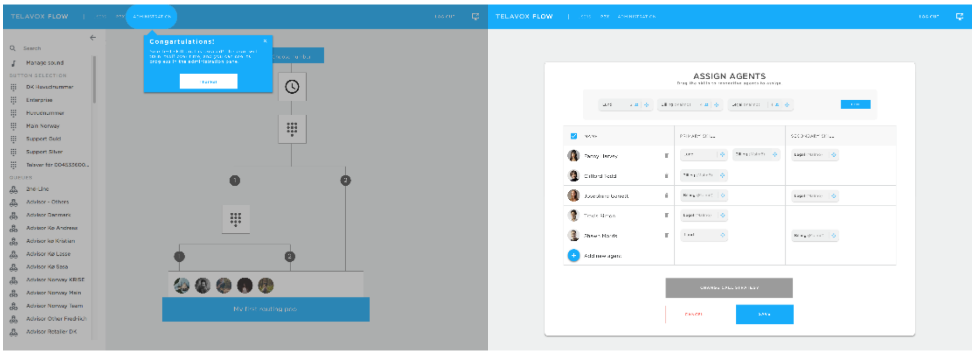


Figure 40. A hint displayed over the automatically generated tree to the left and the pool edit pane to the right.

6.1.2 Agent interface

The agent interface consists of two parts, the mobile interface and the web interface. The two are closely related, offering much of the same functionality.

6.1.2.1 Mobile interface

The high-fidelity mobile interface prototype has a lot in common with the earlier iterations. Most of the functionalities that were included in the earlier prototypes were included in this prototype as well. The results of the mobile high-fidelity prototype can be seen in Figure 41.

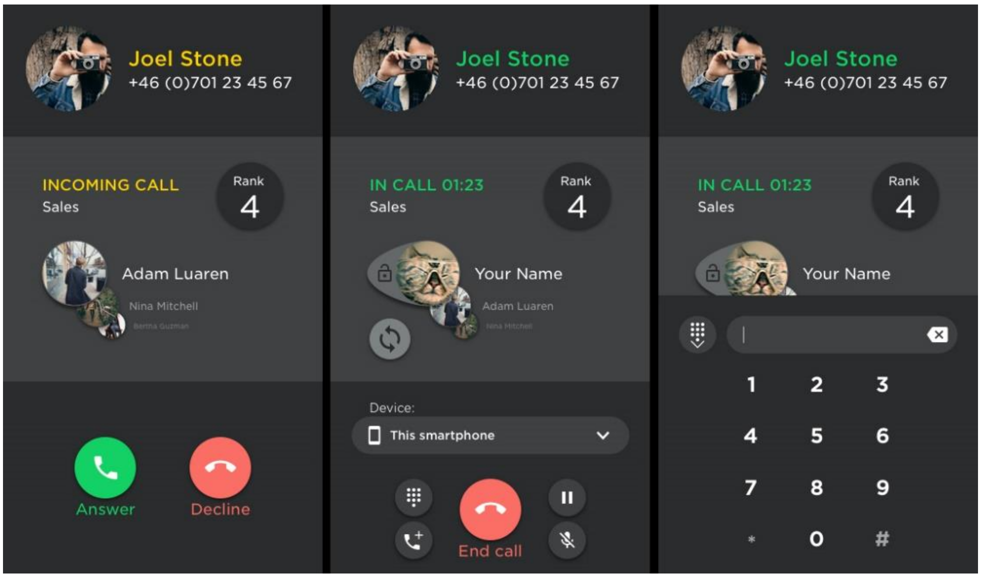


Figure 41. The mobile high-fidelity prototype for the agents. To the left is the incoming call screen, in the middle the in call screen and to the left the numpad is shown.

This prototype is aiming to highlight the main concepts of SBR. To be able to achieve this the canvas is split up into three sections showing on the screen, seen in Figure 41.

The top section is a darker area showing only the contact information, such as the profile picture, the name of the caller (if available) and the number the caller is calling from (if available as well).

When there is an incoming call, the name of the caller and the text “INCOMING CALL” is colored in yellow to notify the agent. Once the agent answers the call by pressing the green answer button, the texts changes color to green as well, making the agent aware of what impact the answer action has made.

The middle section is the important part from a skill based routing point of view. The section has a lighter color than the surrounding sections, which draws the attention of the user to this part. In this section, the agent can see how prioritized he or she is to answer the client, through the “Rank-bubble” in Figure 41.

Once the call is answered, the agent is put as the most prioritized agent to answer that client in the future. If the agent doesn’t want to be the most prioritized agent for that client, the switch-button can toggle the priority for that agent-client relation down to third or second priority as well. It was noted during the mid-fidelity testing that the letters describing priorities was not clear. In the high-fidelity prototype a more visual way of representing the priority was added, using pictures and depth instead of letters. To lift the importance of the three agents showing, different sizes on the profile pictures and the names of those agents as well as the shadows has been presented. The bigger the agent representation is, the higher priority the agent has for that agent-client relation.

The agent also has the option to lock itself as the most prioritized agent to answer that client, this option is only available if no agent already has that client locked to itself.

The bottom section is a darker part that includes a dropdown list of devices to answer the call from and some of the basic functionalities of a call. Some basic functionalities, for example choosing to listen to the call on the speakers instead, have been skipped in this prototype since this prototype is aiming to highlight the main concepts of SBR.

6.1.2.2 Web interface

The high-fidelity web interface prototype fundamentally has the same functionalities as the mobile interface. The arrangement of the buttons and the views are a bit different, but should not result in confusion since they display the same sort of information.

The main difference between the mobile interface and the web interface is that the web interface has more screen area. This means that more content can be shown to the user at the same time making the web interface more suitable for faster and more effective interactions.

Pressing the buttons as seen in Figure 43 can pause and mute the call, as well as bring up the dial pad and add additional people to an ongoing call to create a group call. Pressing the dial pad button opened up a dial pad to the right of the screen. The add people feature was not implemented in the prototype.

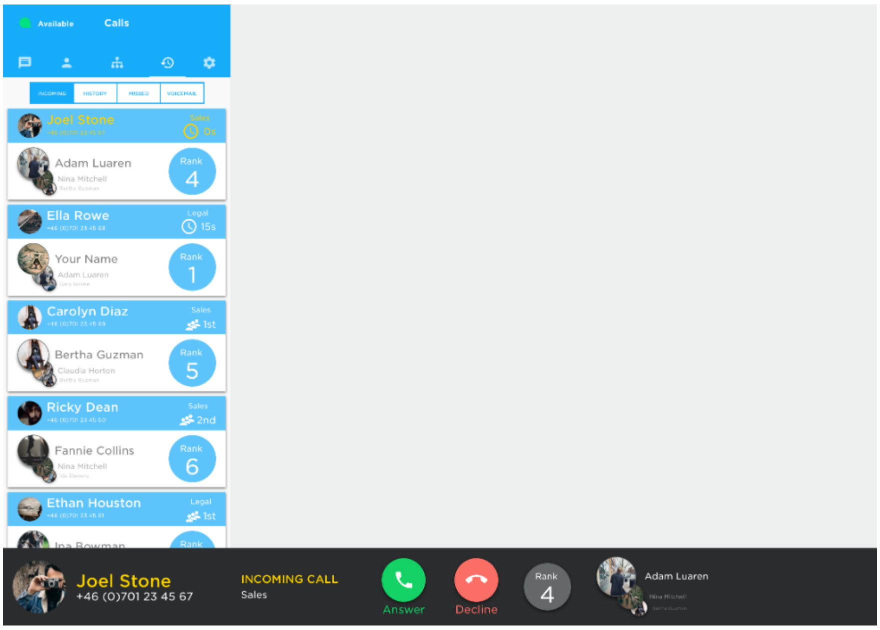


Figure 42. The web interface for an incoming call. The middle part of the screen is reserved for functionality like chats.

As seen in Figure 42 the same information for an incoming call is displayed at the bottom as in the leftmost picture in the mobile interface in Figure 41. However, the blue side panel is accessible at the same time as the call is managed in the web interface. The blue side panel's content can be switched out for the history or incoming call views seen in Figure 44.

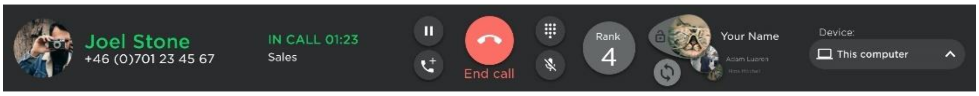


Figure 43. The web interface's bottom panel for an answered call.

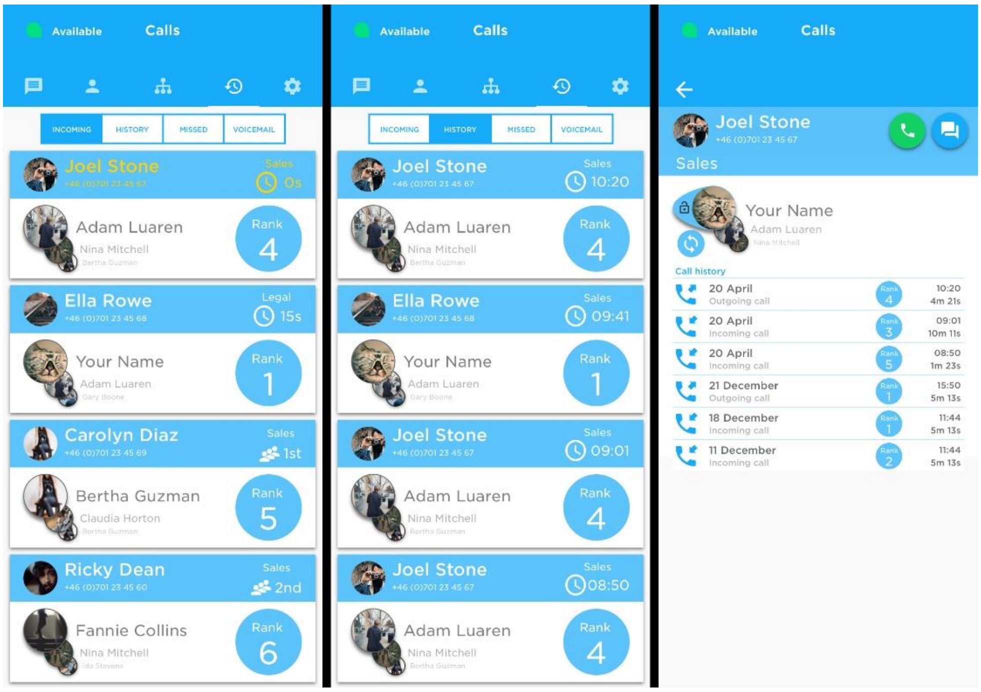


Figure 44. The side panel for managing incoming calls and call history.

6.2 High-fidelity evaluation

For the high-fidelity evaluation, a test plan was developed according to Rubin & Chrisnell's (2008) specification described in the section 2.3.1 Test plan. The full test plan can be found in Appendix G.

The prototypes were sent out to ten employees at Telavox and ten participants outside of Telavox, together with two separate forms with instructions of how to use the prototypes. There were in total 14 participants in the high-fidelity test, 8 inexperienced and 6 experienced with Telavox *Flow*. In the test plan found in Appendix G, the preferred number of participants where 10 of each category. The outcome therefore fell slightly short of the goal, with 80% of the inexperienced participants and 60% of the experienced participants performing the evaluation. To combat the transfer of learning effect, the participants were divided into two groups. Both groups evaluated both the agent and administrator prototypes. The first group had to do the administrator setup prototype first then proceed with the agent UI prototype while the second group were instructed to do the testing of the prototypes in the reversed order. The question and task forms can be viewed in detail in Appendix H.

6.2.1 High-fidelity testing

The high-fidelity prototype was divided into two separate parts for the evaluation as well, the administrator prototype and the agent prototype. For each part, a form was created with the tasks to be performed, as well as questions for feedback. For every task, the participant was asked to give it a difficulty rating from 1 to 5, as well as general comments about performing the tasks.

In the administrator interface the participants were asked to first complete the onboarding and then follow the tree steps of defining and assigning skills and choosing a call strategy. Finally, they were asked to navigate around the interface verifying their choices. This was done through nine tasks, accompanied by additional instructions:

1. Go through the introduction guide and navigate to the setup process.
2. Define the different skills of your company and continue to the next step.
3. Fanny Harvey is primarily located in the Lund office. Add her to the Pool and assign her to the Lund office.
4. Speed up the process by adding all of your remaining available agents to the pool.
5. Fanny, Clifford and Josephine all work in the Malmö Billing section. Shawn will also take some calls if the queue gets busy. Assign the agents to the Malmö Billing department.
6. Assign the rest of the skills to your agents and continue to the next step.
7. Choose the call strategy you think fits your company best and finish the setup.
8. Navigate to the administration page and look who is currently the best agent to answer Adele Green's call.
9. Go back to the PBX page and check that all the skills you defined for the Pool are saved.

For the agent interface, the general purpose of the tasks was to evaluate the priority and tiebreaker presentation to agents, as well as combining the web and mobile interfaces. These main tasks were then divided into twelve subtasks:

1. Begin by starting the prototype's countdown and answer the incoming call.
2. Stone explains his issue, you need some time to download a few files. Put Stone on hold and then resume the conversation again.
3. You are bored of Joel Stone's voice. Dial the "#" character on the numpad, then hide the numpad again.

4. After talking to Stone for a while, you decide that whenever he calls, the system should always route the call to you first. Nobody else should be able to take your position.
5. You realize that your co-worker Adam Luaren is better suited to answer Stone. Set Luaren as the first answering agent for Stone, but set yourself as the secondary answering agent.
6. You now need to move from your computer. Transfer the call to your smartphone.
7. As you walk around with your smartphone, there is a lot of noise around you. Try muting your microphone, then unmute it again.
8. End the call with Stone.
9. Check how many times you have talked to Joel Stone in total.
10. Your co-worker Adam Luaren is tired of answering Stone. Without calling, lock so that Stone's calls will be routed to you.
11. You need to talk to Stone again. Call Stone.
12. End the call with Stone

A full list of the tasks and questions can be seen in Appendix H. In the section 2.3.2.1 Unmoderated remote testing it is described that one of the problems with remote testing uncertainty if tasks are completed correctly. To try to address this slightly, one validation question was added to each form. In the agent interface the participant was asked to count how many calls a certain person made. Respectively in the administrator interface, the participant was asked to find the best suited person to answer a specific call.

6.2.2 High-fidelity results

The experienced participants were all staff members at Telavox, while the inexperienced participants were mostly engineering students not working at Telavox. On the scale ranging from one to five, the inexperienced participants answered an average of 1.1 ($\sigma = 0.5$) familiarity with *Flow* and 1.8 ($\sigma = 0.9$) with the SBR concept. The experiences participants answered 4.7 ($\sigma = 0.5$) familiarity with *Flow* and 2.8 ($\sigma = 0.8$) with the SBR concept. Both groups claimed to be good with computers in general responding 4.1 ($\sigma = 0.8$) and 4.5 ($\sigma = 0.6$) respectively for inexperience and experienced participants. In Figure 45 the age and gender distribution can be found, showing a slight overrepresentation of male participants, and most participants are between the age of 21 and 30. None of the participants claimed to have any form of color blindness.

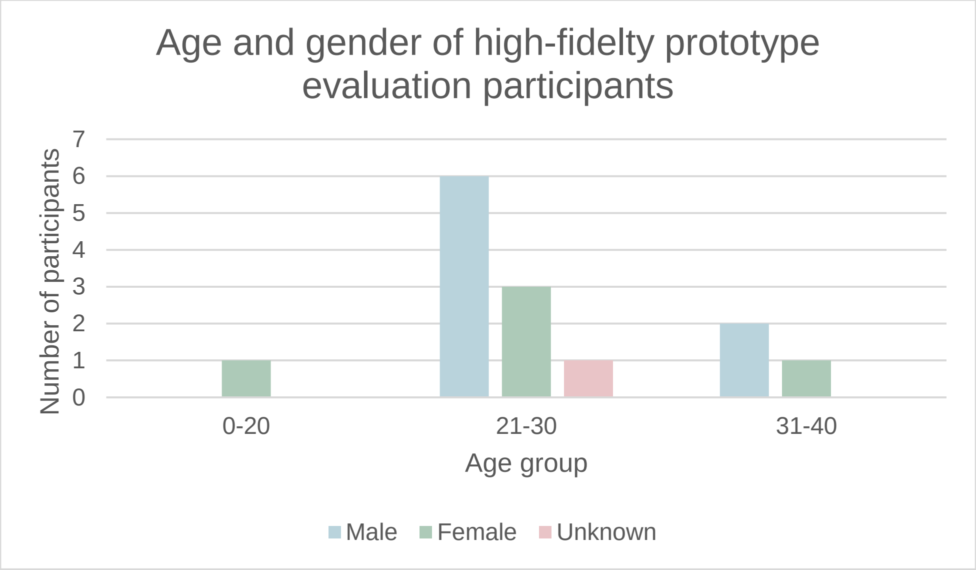


Figure 45. The age and gender distribution of the high-fidelity evaluation participants, as number of participants of each group.

6.2.2.1 Administrator interface

In Table 10 the calculated mean values and standard deviations of the evaluation tasks performed on the administrator interface. The values are divided between the experienced and inexperienced users for comparison. The tasks had an overall average difficulty of 1.8 for inexperienced users and 2.2 for experienced users. Some tasks were defined as more difficult by having a rating above these averages.

Table 10. The averages and standard deviations from the 1-5 rating scales responding to difficulty of tasks performed on the administrator interface. Difficult tasks are bolded.

Task number	*Mean (inexperienced)*	*Standard Deviation (inexperienced)*	*Mean (experienced)*	*Standard Deviation (experienced)*
1	**2.6**	**1.5**	**2.5**	**1.6**
2	**2.5**	**1.1**	**2.5**	**1.6**
3	1.4	0.5	1.8	1.0
4	1.4	0.7	1.5	0.8
5	**2.0**	**1.3**	**2.5**	**0.8**
6	**1.5**	**0.5**	**2.8**	**1.2**
7	1.1	0.4	1.7	1.2
8	**1.9**	**0.8**	**2.8**	**1.5**
9	1.4	0.5	1.8	1.3

The subjective data gathered for the nine tasks are summarized below, grouped based on the different screens the tasks were performed on.

Onboarding The onboarding screen was appreciated but a little unclear. Some participants did not understand how many steps the onboarding screen had, and dismissed it on accident. One participants complained about too many signifiers.

Skill setup The participants found it hard to know how many skills should be created as part of the first tasks. The mapping between the word skill, and the locations "Lund" and "Malmö" was not clear, instead participants saw a location and skill as different things. The participants also found an error in the prototype, with one of the skills changing names from one skill to another when added. This error was due to a erroneous link in the prototype, navigating some users to an older screen if the pressed the left side of the add button. Once the next navigation was done, the participant was back on the proper navigation track of the prototype.

Skill assign Participants had a problem understanding that a location name can be a skill and felt it was not explained enough. They found it strange that assigning an agent to a skill "Lund" was confusing. Some participants commented on the scaling on the view, highlighting that it works well in a small case like the prototype but might have problems if the number of agents increases. One participant had trouble seeing the skills at the top of the page. Responding to the question if it would be better to assign agents to different skill groups, the participants were divided. Four participants were positive to the idea, five were negative and another five said it could maybe be a better option. It was commented that it could depend on the administrators input data, that will most likely be grouped by agents working in a certain department. Therefore, having both options available was suggested as an idea.

Call strategy While straightforward UI wise, the concept of call strategies was not adequately explained, especially noted by the inexperienced participants.

Tiebreaker table Some participants did not understand what the table represented at first and suggested that naming it “call history” could make it clearer. Some positive comments were made about the various ways to filter and sort the table. Only about 65%, 9 out of 14 participants, got the correct answer to the validation question, “Travis”, while the other 35% incorrectly answered “Fanny”.

Tree page While not much feedback was given about the tree page, one participant thought that an edit icon would make it clearer how to reach the pool edit screen.

6.2.2.2 Agent interface

In Table 11 the calculated mean values and standard deviations of the ten evaluation tasks performed on the agent interface. The values are divided between the experienced and inexperienced users for comparison. The tasks had an overall average difficulty of 1.7 for inexperienced users and 1.9 for experienced users. Some tasks were defined as more difficult by having a rating above these averages.

Table 11. The averages and standard deviations from the 1-5 rating scales responding to difficulty of tasks performed on the agent interface. Difficult tasks are bolded.

Task number	*Mean (inexperienced)*	*Standard Deviation (inexperienced)*	*Mean (experienced)*	*Standard Deviation (experienced)*
1	1.0	0.0	1.0	0.0
2	**1.8**	**0.7**	**1.5**	**0.8**
3	1.5	0.8	2.2	1.6
4	**3.6**	**1.2**	**3.3**	**1.4**
5	**3.3**	**1.6**	**4.3**	**1.2**
6	1.0	0.0	1.3	0.5
7	**1.1**	**0.0**	**2.5**	**2.0**
8	1.0	0.0	1.0	0.0
9	1.5	0.8	1.8	0.8
10	**2.6**	**1.6**	**1.7**	**1.2**
11	1.0	0.0	1.0	0.0
12	1.0	0.0	1.0	0.0

The subjective data gathered for the twelve different tasks are summarized below, grouped based on the different screens the tasks were performed on.

Queue overview The participants thought answering the call was easy, but the notification and countdown could be clearer. The yellow text seen in the top left corner in Figure 42 was not easy to see.

Web call Participants though that the icons seen in Figure 43 during a call were not clear enough. Several participants asked for tags or texts to be shown on hover to explain the different icons. Comments were made about the location of the dial pad, as it would feel more natural in the middle of the screen instead of to the right. The concept of changing priority order and priority was difficult to understand and needs more explaining. Once again hover text or other information such as a quick onboarding would be needed. The lock icon sometimes appeared and sometimes not, making the interface feel inconsistent. One participant suggested adding different border colors to separate agents' images from clients. Another participant did not like the concept of adding strategic functionality to the call UI.

Mobile call Transferring the call to mobile was well received and understandable. The UIs were consistent but felt slightly crowded. However, some settings did not save when transferring between web and mobile, which was unclear to some participants.

Call history The call history tab was hard to find for some participants. It was not clear whether the history was for a specific agent or several/the entire system. Each call has a time next to it, as seen in Figure 44, but it was not clear if this represented time of call, duration or time in queue. However, all participants managed to answer the validation question regarding number of previous interactions with a specific client, which was six. Some participants though the queue overview and history looked too much alike and were hard to separate.

6.2.3 **High-fidelity discussion**

The participants in the evaluation where all rather young, however it can be argued that they are representative of the administrators of *Flow*. The gender distribution was rather even, and should not affect the result significantly. While the full goal of participants was not met, it was still within the limit of four to five users, presented in 2.3.1 Test plan. It could have been a good option to use a standardized score for outside validity. This was however not added as the duration of the test was already reaching roughly 40 minutes for the participants. Increasing the extent of the survey further could have affect the answer rate. The standardized test would therefore have had to replace some one the existing questions and tasks.

6.2.3.1 Administrator interface

All the rating averages of the administrator interface was under the middle value of 3, indicating that none of the tasks were difficult to perform. In Table 10, it can be seen that the most difficult tasks were 1, 2, 5, 6 and 8, representing the onboarding, skill definition and skill assign screens.

Even though the onboarding screens were created following the material design guidelines, some participants commented that it was too easy to dismiss prematurely. This issue is likely the explanation of the slightly higher difficulty score, as well as the large standard deviation. One way to address the problem could be increasing the size of the paginators at the bottom, showing more clearly that there are three steps in the guide. It could also be possible to lock the user to follow the guide, and disabling the possibility to exit the guide at any time. This however goes against the guidelines set up by Google.

Because of limitation in the prototype, no free text can be written to the different screens. Some users complained that it was hard to understand were the different names and locations come from, which is partly a prototype issue. The concept of a skill having the name of a location like “Lund” was not clearly conveyed and caused confusion. The scenario was chosen because it is a rather common case for a customer of Telavox to have two locations and one central phone number. There are two possible ways to clarify the concept. One way is that users would create one SBR with different skills for each location, separating the two concepts that way. Another option would be to explain through for instance more onboarding or hints, that skills are generic and can be defined any way the users want. These options would likely need some more user testing to verify.

The last issue with assigning skills in task 5 and 6 are also likely affected by the prototypes limited interaction. The prototype states that drag and drop functionality is to be used, however this is not currently supported in the prototype. It is surprising that the more experienced participants seemed to have more problems with the tasks than the inexperienced participants. A possible answer is that the experienced participants have used other methods for multi agent editing through *Flow*, having a harder time adjusting to the new concept. However, no comments about this was given by the participants. The comments about scalability are to be considered, and adding a way to mass assign skills would probably be the way to go in the future. While many of the use cases will be with few agents, there should be support for larger enterprises in the interface.

The last difficult task was understanding the tiebreaker data table, and only a rather small percentage of the participants got it right. While the purpose of the table was explained through a hint text, it is possible that the users did not understand why it was populated so quickly after creating the SBR. This could be explained through an issue with the prototype or the explanation in the task itself. It also indicates that the concept of priority was not completely clear to the participants. This could possibly be changed by explaining it more in the onboarding process.

6.2.3.2 Agent interface

The most difficult tasks in the agent interface were tasks 2, 4, 5, 7 and 10. Except task 2 and 7, these issues were all related to the priority and locking system. It was commented by several participants that icons meanings were not clear and hints or explanatory texts would make the interface easier to use. It is likely a good idea and possible to do in the web interface, however in the mobile interface this is slightly harder. Hover states are generally not used in smartphone interaction, and comments were made about the interface being crowded already. Therefore, one way to solve the issue would be to remove the strategic routing choices from the call, and only provide the option on desktop and in the history pane.

The concept of pausing the call was an unknown feature to some users and they did not completely understand it. However, this is a rather common in many of the internet based phone solutions today. Once used and established in a real scenario, it is likely that it would be understood. As it is not an integral part of the SBR solution, further development of the feature was ignored. This is also the case with task 7 of muting the microphone.

It was also commented that the lock function appearing and disappearing depending on which agent was on top of the priority stack felt inconsistent. A better option would likely be to gray out the icon instead of removing it. The motivation behind removing the icon is that an agent cannot lock another agent to a specific client, to prevent misuse or abuse.

While comments were made about the clarity of the call history page, most participants seemed to be able to use it effectively. A descriptive text should probably be added, describing that the time field on each history entry represents time of the call. This could also be achieved by combining the time with a date.

7 Discussion

This section will cover the discussion of the methods used in the project, as well as the results gathered through evaluations and surveys.

7.1 Design process and methods

The overall design process, as described in 3.1 Design process, was inspired by Arvola (2014). Structuring the process in three different steps enabled the project to be narrowed down initially, prioritizing the important aspect of SBR. The iterative design process used in the project helped the design in many ways, detecting issues and features essential to the project's success. Starting off with a thorough exploration of the subject was crucial to understand the extent of the subject that is SBR. The projects focus was in interaction design through interfaces, but a big part of the design is also contained in the system model. While the overall interface design did not change very much during the iterations, every upgrade in fidelity exposed new issues and aspects of SBR. Each prototype was evaluated in a slightly different way.

Heuristic evaluation was used early in the process to evaluate the positive and negative features of *Flow* in a structured manner. This combined with the many brainstorming sessions and presentations created a good foundation through methods triangulation when developing both the model and interfaces. Combining many different methods of evaluation and idea generation provided invaluable information for the model and interfaces.

Gathering information from the actual users of the products proved to be quite challenging, and both the low- and to some extent mid-fidelity prototypes had to be evaluated through other means. While there is no way to tell how this affected the outcome of the project, the discussions and methods used instead gave large amount of subjective qualitative feedback and limited objective data.

The high-fidelity prototype was evaluated through unmoderated remote testing on users with and without knowledge of SBR and *Flow*. This made it possible to conduct multiple test sessions and compile the results within a limited timeframe. The choice to conduct remote unmoderated testing was both done to be able to answer RQ5 from 1.4 Goals and delimitations, but also in order to finish the project within the fixed timeframe. To achieve better results and get a better user insight, in-person testing would likely be a better option. However, due to limited availability of participants, this kind of testing was one of the few feasible options providing adequate results. The high-fidelity prototype did not include any standardized test for evaluation, which limits the outside validity of the results. This was a conscious decision as there are no similar interfaces available to compare against, combined with restrictive interaction in the prototype. The test was also lengthy in its current implementation, and adding further questions would likely make the evaluation infeasible as test participants would not want to endure even longer testing sessions.

The project was finished within the required timeframe, in accordance to the project plan. All the stages of the project had even work distribution between the two students, who participated in all activities. The startup during the concept phase took the longest, in accordance to the project plan. Writing in the iterative form of the project helped making sure the reporting kept up with the development of the prototypes. It was desired to perform the final evaluation using an in-person method, however the scheduling of test participants proved very problematic. Overall the time assigned to each of the phases proved reasonable.

7.2 Research questions

Six research questions were presented in the beginning of this report. With the gained experience from this project, these research questions are answered in the following section.

RQ1. Which parameters are necessary to create a successful SBR system?

Since SBR is a rather unexplored area in practice, there is no known definite way, or must-have parameters and aspects for implementing it. One of the first tasks for starting this project was to gather as many viable aspects of data input as possible. It turned out that there are many aspects that can be considered when developing an SBR system, even though the input data from an incoming call might be limited.

The parameters used as input to the system was the ones presented in Table 2, A-number, B-number and IVR input. These parameters can be used in many different ways, however one of the most important findings was different uses of the IVR choice. IVR can be used for input of personal or customer numbers to increase the available input parameters, and create stronger routing. This was however deemed outside the project scope, and suitable for future studies.

RQ2. How can a system model for SBR be created that is both forgiving and flexible?

The complexity of the system increases dramatically as more parameters are introduced, especially if skills are defined scales or ranges. This effect comes because calculations are needed for every call and agent. It is also difficult to establish clear hierarchies between different parameters once addition or multiplication is involved instead of clear sorting algorithms. When the definitions are limited to scales that have fewer steps the flexibility for customization is limited as well. A system that is too complicated for the features it offers will not be used. Therefore, it was deemed that using binary skills with hierarchy and tiebreakers provide enough routing functionality for a full fletches SBR model. This should then be used to create a prioritized list of agents matching the certain skill, to which the call will then be distributed.

It was seen that for clients, the concept of on hook waiting was interesting and should be included in a good SBR model. While no tests were performed on the clients, as per the delimitations, the survey provided useful information in the clients' preferences. The survey could have been better suited earlier in the project during the concept phase, to shape the model earlier on. This could possibly have increased its impact on the design of the model and shaped it more in the favor of the clients.

Whether the model is forgiving when implemented could not be answered through the limited simulations and interface designs created in this project. It would require an actual implementation of the SBR system to evaluate flexibility and forgiveness. However, provided the models support for external system integration and queries, it could be argued that it is flexible in terms of routing.

RQ3. How should a UI be designed to present the setup of an SBR system?

To answer this question, the three-step setup wizard was created. It was identified early in the project that the steps of defining skills, assigning them to agents and deciding on a distribution strategy were good basic steps. The setup differed a bit from how other PBX systems usually are set up in *Flow* today. The way the PBX systems are set up in Flow is through the tree view seen in Figure 1, where nodes are placed in the tree view and customized by clicking on them. The administrator adds and customizes every node separately and is finished whenever the administrator likes. Based on the feedback, especially through the mid-fidelity evaluation, a slightly different approach with an auto generating tree structure was created instead. This is arguably a simpler cognitive process than the current implementation of creating a PBX.

Using an onboarding process to explain the concept and steps included in setting up a SBR pool is an important part of creating a good setup experience. As per the evaluation results, no task was deemed overly difficult, and it could be argued that the wizard described in 6.2.2 High-fidelity results is a good way to create a setup process.

RQ4. How should a UI be designed to present routing and calls to the agents in a transparent manner?

This was done via the agent UI described in 6.2.2 High-fidelity results, using both a mobile device and a bigger device using the web. The agent UI can display a list of incoming calls and relevant information about the client calling, personalized for the agent, as seen in Figure 44. The agent can see which rank the agent has for that client, which place the client has in the queue and the three best matching agents for that client.

The attempt to provide transparency by showing the top three suited agents as well as their own rank for the current call, falls a bit short. While no unusually high difficulty score was measured for any task, understanding the concept of rank and priority was the most challenging part of the agent interface. The interface described in 6.2.2 High-fidelity results is possibly a working solution, if accompanied with an onboarding tutorial or hint texts. However more testing and evaluation is needed.

RQ5. Is unmoderated remote testing an effective way to evaluate high-fidelity prototypes?

This type of testing showed to be useful when conducted within a limited timeframe and availability of participants. It is likely that more information could have been gathered if individual in-person test sessions would have been conducted, however this was not feasible in the project. In-person test sessions forces the test participants to answer the questions asked in a more thorough way, and enables follow up from the test leader. They might also provide more information about the prototype since a user probably won't write down every thought when performing a remote test.

The unmoderated tests were split into small tasks that had their own comment sections in the forms. This made it easier to get more information from the tests. Each task pointed out the main issues that might be present for the part of the UI that is being tested. Writing test cases and instructions can be challenging, as there is a fine line between helping to little, and helping too much. Providing too much information to the participants might hide certain issues with the interface, while providing to little might lead them to not understand the tasks properly.

While not recommended, unmoderated remote testing sessions is an option to gather information about the difficulties and preferences of an interface. Although some more work is required in creating the test, there is very little resources spent on performing the testing. It also makes scheduling easier, as participants can perform the test whenever they have time. Both qualitative and quantitative data can be gathered, however there is limited objective data gathering without solutions such as screen recording and post-test processing.

RQ6. How do clients expect an SBR system to behave?

To find an answer to this question, the survey described in section 5.3 Survey was created. In 5.3.3.3 Future PBX systems, several client expectations are described in detail.

Most of the respondents would rather wait additionally a few minutes to get support from an agent that they already have had an interaction with earlier. This is especially true when the system has implemented the on hook waiting functionality, where clients could accept waiting times. In the general case however, it can be argued that most clients want to reach a knowledgeable person quickly. Therefore, a SBR system should have a main goal of reducing waiting times as much as possible. However, this can differ on a company basis, as different companies offer different services.

When calling an SBR PBX, over half of the respondents wanted the system to ask if it is about the same issue, which would introduce an additional button press for the client. The proposition in 5.3.3.5 Other comments might be a good solution both from the efficiency aspect as well as the customer satisfaction aspect. Especially for client cases that is not resolved on first contact. Based on the survey results, it can be argued that clients want the system to offer options and be transparent about its routing. This is possibly because they do not completely trust in the technology for it to do routing completely on its own, and would rather be able to make choices themselves.

7.3 Future studies

Some aspects of the system that were found interesting through the course of the project, were disregarded either because of time limitations or because of the rather low gain for the high complexity.

The important next step for this project would be to implement the SBR model and interfaces in code, in order to evaluate the systems real usability. This would require more integration into the current working of *Flow* and how PBX systems are created currently. However, many of the features and key concepts of the design could likely be reused in such a transition.

Other SBR related aspects could also be good research topics for future fields of study:

Queue memory One feature that was discovered through the project is the queue memory feature. This feature is about the handling of clients in the queue before they are assigned to an agent. The concept is that when a calling client waits in the queue, the client should not immediately be thrown out of the queue upon being disconnected. This should also apply to whenever a client gets disconnected during a conversation with an agent.

The queue memory might be a very valuable functionality when it comes to customer satisfaction, especially if there are long queues. In case of an unfortunate disconnect, the client will be positively surprised when the smart system didn't put the client at the end of the queue upon calling again.

External database integration As mentioned in section 4.1.1.2 Unstructured interviews privacy aspects for companies must be considered. To solve this, it is possible for the system to implement an external database integration, so that no information about the clients are leaked outside the company's database. This also greatly increases the flexibility of the system.

In the case of an external database integration, the system might then instead of using the last-talked-to tiebreaker send the calling client's information to the external database. The external CRM system will then generate a list of prioritized agents based on their private data and send that back. The returned list of prioritized agents can then be used as a tiebreaker instead of the default last-talked-to method.

Call strategy creator The call strategy creator, seen in both the low-fidelity and the mid-fidelity prototypes, was not implemented in the high-fidelity prototype due to its potential for high complexity. When choosing call strategy in the high-fidelity prototype the user can pick between two presets and a custom-built strategy. The button for a custom strategy is however not implemented.

There are a lot of factors to be considered when creating a custom call strategy. The call strategy creator was a bit difficult to understand according to the test participants from the mid-fidelity prototype and might be something that only the more advanced administrator is willing to do.

Some options that should be included in a call strategy are: how many phases it should have, how long it should last and how many agents that should be called in each phase. The number of agents that should be called is a complicated issue, since the queues in one SBR pool can have a different number of agents assigned. For example, calling 50% of the agents in one queue can mean only calling one agent in queue A (two agents in total), but can also mean calling 50 agents in queue B (100 agents in total).

8 Conclusions

This section will finalize this report by covering the conclusions that can be drawn from this project.

The purpose of this thesis was to create a powerful system model for an SBR system and visualize its set up and use through interface prototypes. The concept of SBR can be very complex, but even a less complex system can provide numerous benefits over regular routing.

An iterative, user-centered design process was applied to this project to make sure that the product matches expectations of both primary and tertiary users. A system model considering different parameter usage and routing options was presented. Two interfaces were designed and evaluated in three phases. One administrator interface for setting up the system, and an agent interface for interacting with clients and the newly added SBR functionality.

Each prototype was evaluated in a different way, utilizing experts, group evaluations and in-person testing for the low- and medium-fidelity prototypes. The high-fidelity prototype was evaluated through unmoderated remote testing sessions, as a solution to low end user availability. While not recommended as a first evaluation option, it served as an acceptable test method given the projects requirements. The evaluation was performed both by experienced PBX users as well as inexperienced participants. None of the evaluation tasks proved to be over medium difficulty on average, confirming that the interfaces created in the project could work as a guideline for a future implementation.

SBR is an interesting way to add intelligence to the concept of telephone routing, making it an integral part for future PBX solutions. There is much to be gained from further research in the subject, for business and clients alike, and it is likely that SBR will improve the experience of customer service interactions all over the world.

References

Arvola, M. (2014). *Interaktionsdesign och UX.* Lund: Studentlitteratur.

Chaffey, D. (2017). *Mobile Marketing Statistics compilation.* Retrieved from Smart Insights: http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

Eason, K. (2005). *Information Technology And Organisational Change.* CRC Press.

Flaherty, K. (2015). *How Much Time Does It Take to Create Personas?* Retrieved March 6, 2017, from Nielsen Norman Group: https://www.nngroup.com/articles/persona-budgets/

Garnett, O., & Mandelbaum, A. (2000). An Introduction to Skill-Based Routing and its Operational Complexities. Technion.

Google Inc. (2017). *Growth & Communications - Onboarding*. Retrieved from Material design: https://material.io/guidelines/growth-communications/onboarding.html#onboarding-top-user-benefits

Graham, L. (2008). Gestalt Theory in Interactive Media Design. *Journal of Humanities & Social Sciences*, 1-12.

Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). *Fundamentals of Queueing Theory, Fourth Edition.* John Wiley & Jons, Inc.

Harley, A. (2015). *Personas Make Users Memorable for Product Team Members*. Retrieved March 6, 2017, from Nielsen Norman Group: https://www.nngroup.com/articles/persona/

International Organization for Standardization [ISO]. (2010). *Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems (ISO 9241-210:2010).*

Johnson, J. (2010). *Designing with the mind in mind.* Elsvier Inc.

Kelly, K., Clark, B., Brown, V., & Sitzia, J. (2003). Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, 261-266.

Laubheimer, P., & Loranger, H. (2017). *Accounting for UX Work with User Stories in Agile Projects.* Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/ux-user-stories/

LTH Libraries. (2017). *Where to start searching for information for master students LTH.* Retrieved February 14, 2017, from Metodkurs Examensarbete: http://libguides.lub.lu.se/ld.php?content_id=12013745

Microsoft. (2015). *Global State of Multichannel Customer Service Report.* Retrieved March 8, 2017, from Parature Resource Library: http://paratureprod.blob.core.windows.net/wp-uploads/Global-State-of-Multichannel-Customer-Service-Report.pdf

Miratrix, L., Sekhon, J., Theodoridis, A., & Campos, L. (2017). *Worth Weighting? How to Think About and Use Sample Weights in Survey Experiments.*

Nielsen Norman Group. (2016). UX Prototypes: Low Fidelity vs. High Fidelity.

Nielsen, J. (1993). *Iterative User-Interface Design.* Morristown: IEEE.

Nielsen, J. (1995a). *10 Usability Heuristics for User Interface Design.* Retrieved February 22, 2017, from Nielsen Norman Group: https://www.nngroup.com/articles/ten-usability-heuristics/

Nielsen, J. (1995b). *How to Conduct a Heuristic Evaluation.* Retrieved February 27, 2017, from Nielsen Norman Group: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Norman, D. (2013). *Design of Everyday Things: Revised and expanded edition.* New York: Basic Books.

Patton, M. Q. (1999). Enhancing the Quality and Credibility of Qualitative Analysis. *Health services research,* 1193-1194.

Pichler, R. (2014). *From Personas to User Stories*. Retrieved March 6, 2017, from romanpickler: http://www.romanpichler.com/blog/personas-epics-user-stories/

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques.* Springer-Verlag Berlin Heidelberg.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design : beyond human-computer interaction.* New York: John Wiley & Sons Inc.

Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing.* Indianapolis: Whiley Publishing, Inc.

Schade, A. (2013). *Remote Usability Tests.* Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/remote-usability-tests/

Schade, A. (2014). *Responsive Web Design (RWD) and User Experience*. Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/responsive-web-design-definition/

Soucy, K. (2010). *Unmoderated, Remote Usability Testing: Good or Evil?* Retrieved from UX matters:

http://www.uxmatters.com/mt/archives/2010/01/unmoderated-remote-usability-testing-good-or-evil.php#comments

Spector, E. P. (1992). *Summated Rating Scale Construction.* Newbury Park: SAGE Publications Inc.

Sutton, R., & Hargadon, A. (1996). Brainstorming Groups in Context: Effectiveness in a Product Design Firm. *Administrative Science Quarterly*, 685-718.

Wallace, R. B., & Whitt, W. (2005). *A Staffing Algorithm for Call Centers with.* Retrieved January 27, 2017, from Manufacturing & Service Operations Management: http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&db=inh&AN=8839994&site=eds-live&scope=site

Wroblewski, L. (2009). *Mobile First*. Retrieved from LukeW: https://www.lukew.com/ff/entry.asp?933

Wroblewski, L. (2012). *Mobile First, live at Event Apart.* Retrieved from Vimeo: https://vimeo.com/38187066

Zeaaraoui, A., Bougroun, Z., & Belkasmi, M. a. (2013). User stories template for object-oriented applications. *Third International Conference on Innovative Computing Technology.* London. Retrieved February 28, 2017, from http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=inh&AN=13881818&site=eds-live&scope=site

Appendix A Checklist for Heuristic Evaluation

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

(Nielsen, 1995a)

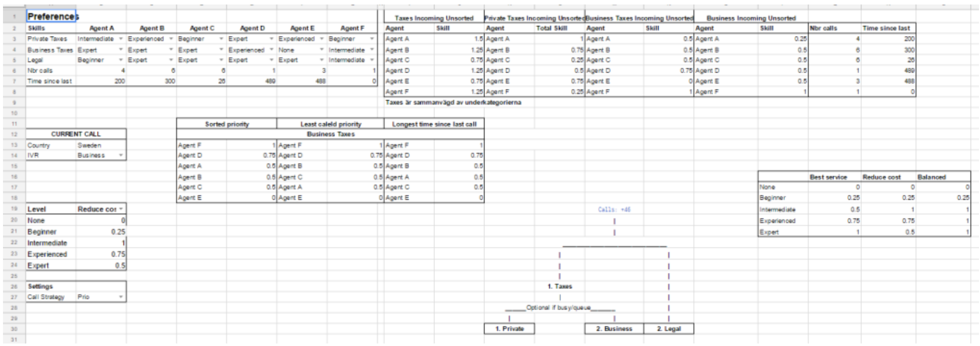
Appendix B Spreadsheets for system model simulation

The following spreadsheets were created in the early stage of the project in order to try and simulate the inner working of the skill based routing model. They proved difficult to both describe and implement and thus were not iterated on further.

The first spreadsheet used five levels for each agent skill: None, Beginner, Intermediate, Experience and Expert. These had different weight parameters attached to them. Depending on which strategy was in place: “Best service” “Reduced cost” or “Balanced”, different levels would be prioritized according to the rightmost table. The thought behind this was to in some cases save the experts for last, always route the call to the best agent, or try to balance it between the best agents. The agents’ suitability would be calculated with the following formula:

$$\mathit{Rating} = \mathit{requiredSkill}\ \cdot \mathit{skillLevel} \cdot \mathit{strategyWeight}$$

Where *requiredSkill* is the skill that comes in to the system, *skillLevel* is the level of each agent (none, beginner and so on) and *strategyWeight* is the corresponding weight value to the skill level of that agent. The agents would then be inner sorted (as a tiebreaker) on the number of calls today, time since last call or nothing, resulting in three different lists.



Preferences

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Private Taxes	Intermediate	Experienced	Beginner	Expert	Experienced	Beginner
Business Taxes	Expert	Expert	Expert	Experienced	None	Intermediate
Legal	Beginner	Expert	Expert	Expert	Expert	Intermediate
Nbr calls	4	6	6	1	3	1
Time since last	200	300	26	489	488	0

Taxes Incoming Unsorted		Private Taxes Incoming Unsorted		Business Taxes Incoming Unsorted		Business Incoming Unsorted			
Agent	Skill	Agent	Total Skill	Agent	Skill	Agent	Skill	Nbr calls	Time since last
Agent A	1.5	Agent A	1	Agent A	0.5	Agent A	0.25	4	200
Agent B	1.25	Agent B	0.75	Agent B	0.5	Agent B	0.5	6	300
Agent C	0.75	Agent C	0.25	Agent C	0.5	Agent C	0.5	6	26
Agent D	1.25	Agent D	0.5	Agent D	0.75	Agent D	0.5	1	489
Agent E	0.75	Agent E	0.75	Agent E	0	Agent E	0.5	3	488
Agent F	1.25	Agent F	0.25	Agent F	1	Agent F	1	1	0

Taxes är sammanvägd av underkategorierna

CURRENT CALL	
Country	Sweden
IVR	Business

Sorted priority		Least caleld priority		Longest time since last call	
Business Taxes					
Agent F	1	Agent F	1	Agent F	1
Agent D	0.75	Agent D	0.75	Agent D	0.75
Agent A	0.5	Agent B	0.5	Agent B	0.5
Agent B	0.5	Agent C	0.5	Agent A	0.5
Agent C	0.5	Agent A	0.5	Agent C	0.5
Agent E	0	Agent E	0	Agent E	0

Level	Reduce cos
None	0
Beginner	0.25
Intermediate	1
Experienced	0.75
Expert	0.5

Settings	
Call Strategy	Prio

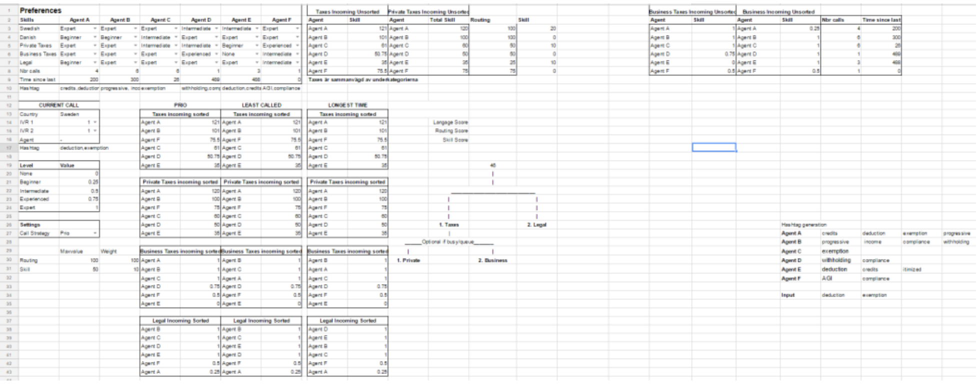
	Best service	Reduce cost	Balanced
None	0	0	0
Beginner	0.25	0.25	0.25
Intermediate	0.5	1	1
Experienced	0.75	0.75	1
Expert	1	0.5	1

Figure B 1. The first spreadsheet, describing agents with different skills, calls today and time since last call. Using weight parameters, a score was calculated, resulting in different prioritized list outputs.

The second spreadsheet worked roughly the same way as the first one, however it also considered general hashtags and counted the matches of these tags to provide more fine grain routing. The thought being that these tags would be populated from an external system and be for instance certain products or “low level skills”.

$$
\begin{aligned}
Rating = \big((&anbrLanguage \cdot languagePreference + \ bnbrLanguage \\
&\cdot languagePreference) \cdot languageWeight \\
&+ (requiredSkill \cdot skillPreference) \cdot skillWeight \\
&+ (nbrMatchingTags \cdot tagWeight)\big)
\end{aligned}
$$

Where *anbrLanguage* is the language corresponding to the a-numbers country code, *bnbrLanguage* is language corresponding to the b-numbers country code, *requiredSkill* is the skill needed to answer the call and *skillPreference* is how good an agent is at the certain skill. The number of matching tags is seen in *nbrMatchiungTags*. The issue with this model is that several matching tags can outperform for instance a language tag depending on the weights, creating very unclear routing. Once more than two parameters with weight values are used, it gets difficult to balance against this outperformance problem, without putting limits on the values. Adding limits to the values further increase the complexity.



Preferences

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	Expert	Expert	Expert	Intermediate	Intermediate	Expert
Danish	Beginner	Beginner	Intermediate	Expert	Expert	Expert
Private Taxes	Expert	Expert	Intermediate	Intermediate	Beginner	Experienced
Business Taxes	Expert	Expert	Expert	Experienced	None	Intermediate
Legal	Beginner	Expert	Expert	Expert	Expert	Intermediate
Nbr calls	4	6	6	1	3	1
Time since last	200	300	26	469	468	0
Hashtag	credits, deductior	progressive, inco	exemption	withholding,comp	deduction,credits	AGI,compliance

Taxes Incoming Unsorted		Private Taxes Incoming Unsorted			
Agent	Skill	Agent	Total Skill	Routing	Skill
Agent A	121	Agent A	120	100	20
Agent B	101	Agent B	100	100	0
Agent C	61	Agent C	60	50	10
Agent D	50.75	Agent D	50	50	0
Agent E	35	Agent E	35	25	10
Agent F	75.5	Agent F	75	75	0

Taxes är sammanvägd av underkategorierna

Business Taxes Incoming Unsorted		Business Incoming Unsorted			
Agent	Skill	Agent	Skill	Nbr calls	Time since last
Agent A	1	Agent A	0.25	4	200
Agent B	1	Agent B	1	6	300
Agent C	1	Agent C	1	6	26
Agent D	0.75	Agent D	1	1	469
Agent E	0	Agent E	1	3	468
Agent F	0.5	Agent F	0.5	1	0

CURRENT CALL	
Country	Sweden
IVR 1	1
IVR 2	1
Agent	-
Hashtag	deduction,exemption

Level	Value
None	0
Beginner	0.25
Intermediate	0.5
Experienced	0.75
Expert	1

Settings	
Call Strategy	Prio

	Maxvalue	Weight
Routing	100	100
Skill	50	10

PRIO		LEAST CALLED		LONGEST TIME	
Taxes incoming sorted		Taxes incoming sorted		Taxes incoming sorted	
Agent A	121	Agent A	121	Agent A	121
Agent B	101	Agent B	101	Agent B	101
Agent F	75.5	Agent F	75.5	Agent F	75.5
Agent C	61	Agent C	61	Agent C	61
Agent D	50.75	Agent D	50.75	Agent D	50.75
Agent E	35	Agent E	35	Agent E	35
Private Taxes incoming sorted		Private Taxes incoming sorted		Private Taxes incoming sorted	
Agent A	120	Agent A	120	Agent A	120
Agent B	100	Agent B	100	Agent B	100
Agent F	75	Agent F	75	Agent F	75
Agent C	60	Agent C	60	Agent C	60
Agent D	50	Agent D	50	Agent D	50
Agent E	35	Agent E	35	Agent E	35
Business Taxes incoming sorted		Business Taxes incoming sorted		Business Taxes incoming sorted	
Agent A	1	Agent B	1	Agent B	1
Agent B	1	Agent C	1	Agent A	1
Agent C	1	Agent A	1	Agent C	1
Agent D	0.75	Agent D	0.75	Agent D	0.75
Agent F	0.5	Agent F	0.5	Agent F	0.5
Agent E	0	Agent E	0	Agent E	0
Legal Incoming Sorted		Legal Incoming Sorted		Legal Incoming Sorted	
Agent B	1	Agent B	1	Agent D	1
Agent C	1	Agent C	1	Agent E	1
Agent D	1	Agent E	1	Agent B	1
Agent E	1	Agent D	1	Agent C	1
Agent F	0.5	Agent F	0.5	Agent F	0.5
Agent A	0.25	Agent A	0.25	Agent A	0.25

Langage Score
Routing Score
Skill Score

Hashtag generation				
Agent A	credits	deduction	exemption	progressive
Agent B	progressive	income	compliance	withholding
Agent C	exemption			
Agent D	withholding	compliance		
Agent E	deduction	credits	itimized	
Agent F	AGI	compliance		
Input	deduction	exemption		

Figure B 2. The second spreadsheet, describing agents with different languages, skills, calls today and time since last call. Different weight parameters as well as different tiebreakers systems resulted in several different output lists.

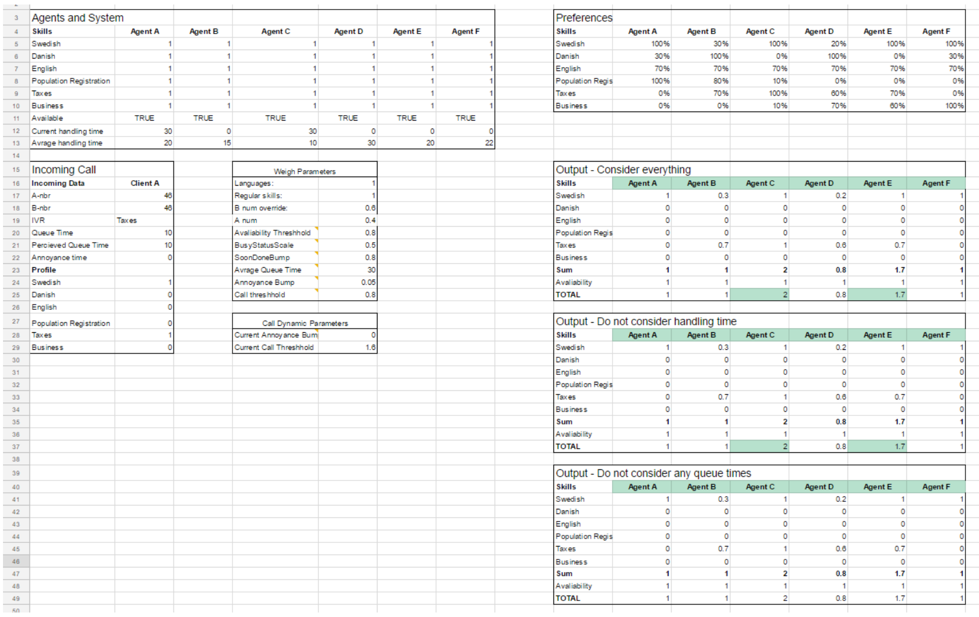
The third spreadsheets used a rather complex formula to calculate a total skill rating for the agents. The calculation of each agent in the spreadsheet can be summarized by the following formulas in combination:

$$
\begin{aligned}
Rating = \Big(\big((&anbrLanguage \cdot languagePreference + \ bnbrLanguage \\
&\cdot languagePreference) \cdot languageWeight \\
&+ (requiredSkill \cdot skillPreference) \cdot skillWeight\big) + annoyanceBump\Big) \\
&\cdot \ avaliability
\end{aligned}
$$

$$annoyanceBump = annoyanceTime \cdot bumpAmount$$

$$\text{annoyanceTime} = \begin{cases} \text{queueTime} - \text{averageQueueTime} & \text{if queueTime} > \text{averageQueueTime} \\ 0 & \textit{otherwise} \end{cases}$$

This model is very similar to model two and used the same primary way of calculating the skill. However, it also added comparison between queues through *annoyanceTime* and availability of agents. In words, this means that if the agent is not available, they will have a rating of 0 because of the availability value that is either 0 or 1. If the client has been in the queue for longer than the average queue time, their *annoyanceBump* increases. This was though of a way to bump up clients that had been waiting for a long time in a specific queue, to be prioritized over clients in other queues. However, cross-considering different queues quickly proved very advanced and the concept was scrapped.



Agents and System

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	1	1	1	1	1	1
Danish	1	1	1	1	1	1
English	1	1	1	1	1	1
Population Registration	1	1	1	1	1	1
Taxes	1	1	1	1	1	1
Business	1	1	1	1	1	1
Available	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
Current handling time	30	0	30	0	0	0
Avrage handling time	20	15	10	30	20	22

Incoming Call

Incoming Data	Client A
A-nbr	46
B-nbr	46
IVR	Taxes
Queue Time	10
Percieved Queue Time	10
Annoyance time	0
Profile	
Swedish	1
Danish	0
English	0
Population Registration	0
Taxes	1
Business	0

Weigh Parameters	
Languages:	1
Regular skills:	1
B num override:	0.6
A num	0.4
Avaliability Threshhold	0.8
BusyStatusScale	0.5
SoonDoneBump	0.8
Avrage Queue Time	30
Annoyance Bump	0.05
Call threshhold	0.8

Call Dynamic Parameters	
Current Annoyance Bum	0
Current Call Threshhold	1.6

Preferences

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	100%	30%	100%	20%	100%	100%
Danish	30%	100%	0%	100%	0%	30%
English	70%	70%	70%	70%	70%	70%
Population Regis	100%	80%	10%	0%	0%	0%
Taxes	0%	70%	100%	60%	70%	0%
Business	0%	0%	10%	70%	60%	100%

Output - Consider everything

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	1	0.3	1	0.2	1	1
Danish	0	0	0	0	0	0
English	0	0	0	0	0	0
Population Regis	0	0	0	0	0	0
Taxes	0	0.7	1	0.6	0.7	0
Business	0	0	0	0	0	0
Sum	**1**	**1**	**2**	**0.8**	**1.7**	**1**
Avaliability	1	1	1	1	1	1
TOTAL	1	1	2	0.8	1.7	1

Output - Do not consider handling time

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	1	0.3	1	0.2	1	1
Danish	0	0	0	0	0	0
English	0	0	0	0	0	0
Population Regis	0	0	0	0	0	0
Taxes	0	0.7	1	0.6	0.7	0
Business	0	0	0	0	0	0
Sum	**1**	**1**	**2**	**0.8**	**1.7**	**1**
Avaliability	1	1	1	1	1	1
TOTAL	1	1	2	0.8	1.7	1

Output - Do not consider any queue times

Skills	Agent A	Agent B	Agent C	Agent D	Agent E	Agent F
Swedish	1	0.3	1	0.2	1	1
Danish	0	0	0	0	0	0
English	0	0	0	0	0	0
Population Regis	0	0	0	0	0	0
Taxes	0	0.7	1	0.6	0.7	0
Business	0	0	0	0	0	0
Sum	**1**	**1**	**2**	**0.8**	**1.7**	**1**
Avaliability	1	1	1	1	1	1
TOTAL	1	1	2	0.8	1.7	1

Figure B 3. The third spreadsheet, showing a very advanced skill based routing system model. It uses several parameters and statistics for each agent, as well as dynamic parameters. These dynamic parameters changed depending on system load, client wait time and average times in the system. The model proved very complex and was soon scrapped.

Appendix C Client system flowcharts

Below are high resolution images of the flowchart alternatives created for clients calling the SBR system.

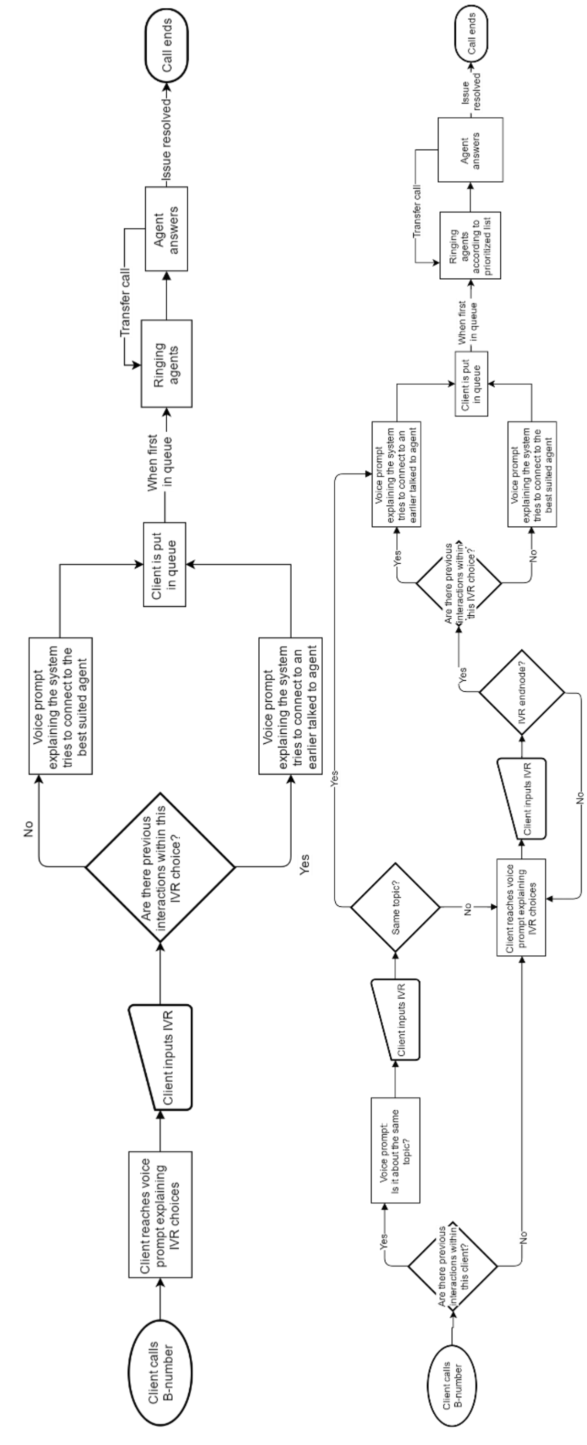

Client calls B-number
Client reaches voice prompt explaining IVR choices
Client inputs IVR
Are there previous interactions within this IVR choice?
No
Yes
Voice prompt explaining the system tries to connect to the best suited agent
Voice prompt explaining the system tries to connect to an earlier talked to agent
Client is put in queue
When first in queue
Ringing agents
Transfer call
Agent answers
Issue resolved
Call ends
Client calls B-number
Are there previous interactions within this client?
Yes
No
Voice prompt: Is it about the same topic?
Client inputs IVR
Same topic?
Yes
No
Client reaches voice prompt explaining IVR choices
Client inputs IVR
IVR endnode?
Yes
No
Are there previous interactions within this IVR choice?
Yes
No
Voice prompt explaining the system tries to connect to an earlier talked to agent
Voice prompt explaining the system tries to connect to the best suited agent
Client is put in queue
When first in queue
Ringing agents according to prioritized list
Transfer call
Agent answers
Issue resolved
Call ends

Figure C 1. The two different flowchart alternatives developed for clients' interaction when calling the system.

Appendix D Early prototyping sketches

There were several early sketches for the interfaces at an early stage. These sketches were created to visualize different ideas and iterated on to improve then going into the low-fidelity prototyping stage.

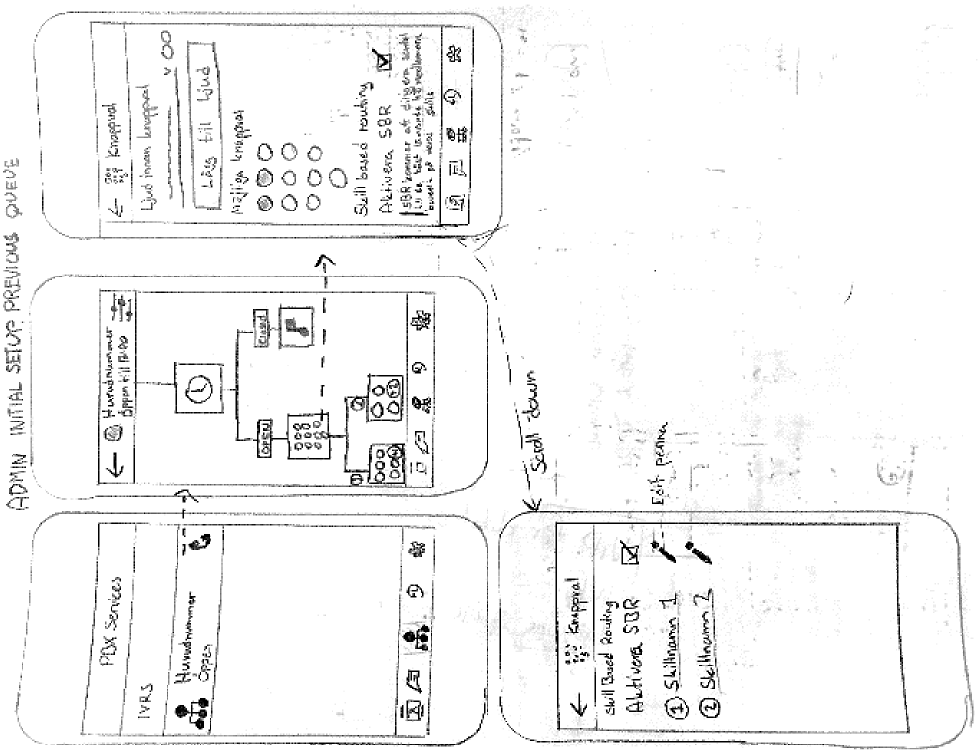


Figure D 1. Admin initial setup from a previously available queue.

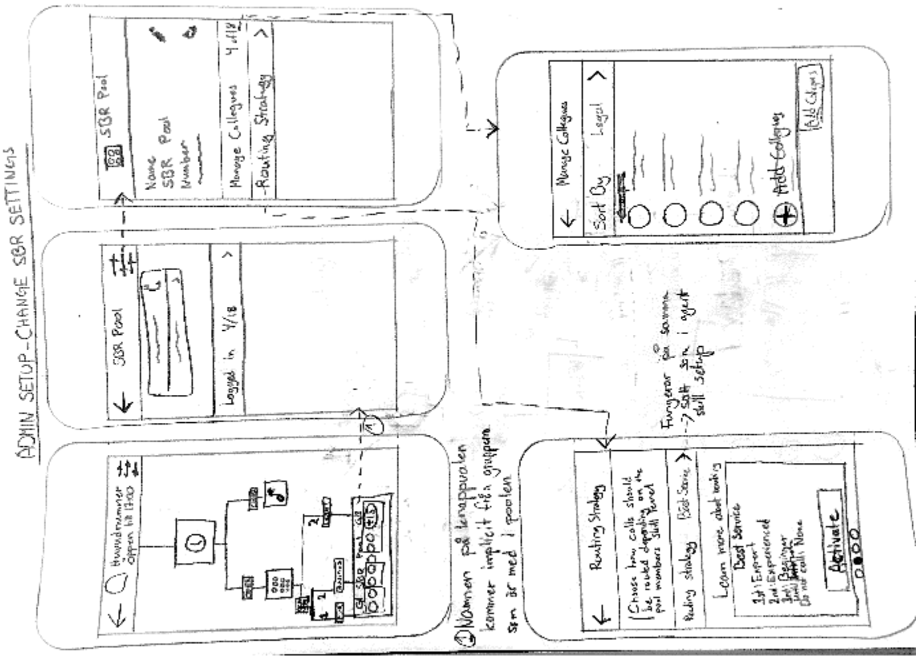


Figure D 2. Admin setup – Change SBR settings like agents and call strategies.

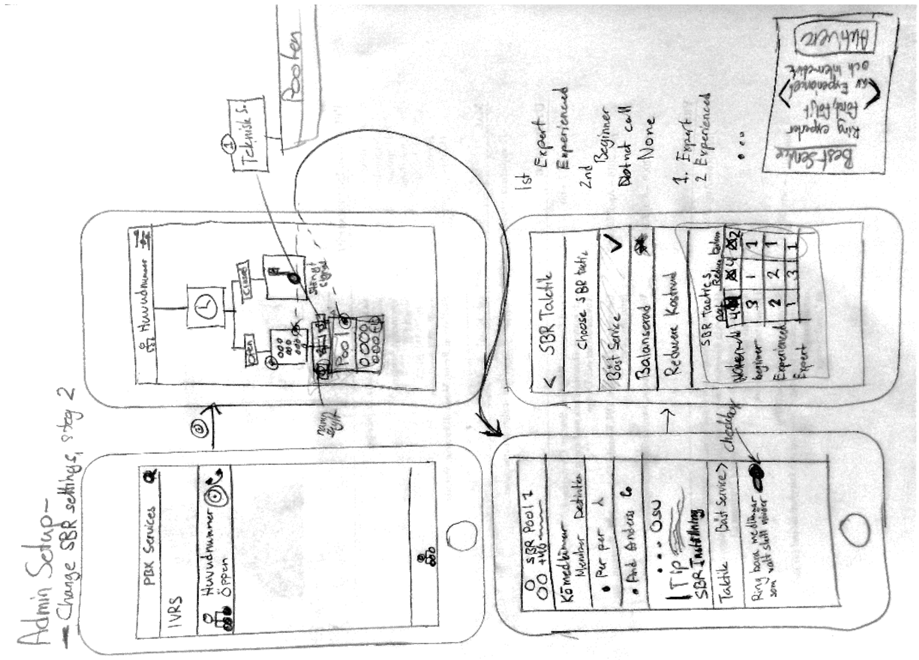


Figure D 3. Admin setup, change SBR settings like call strategy.

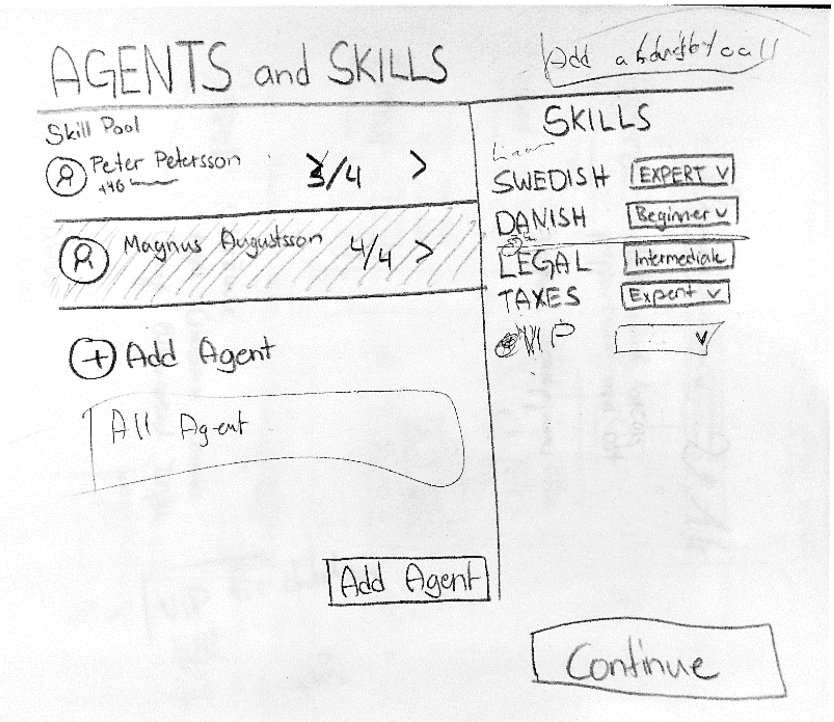


Figure D 4. Early sketch of how agents and skills should be added to the pool.

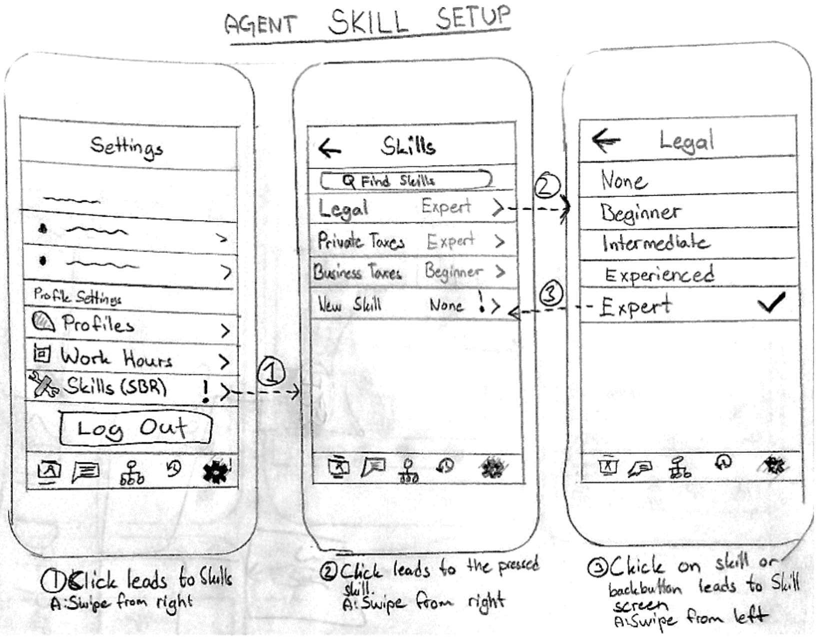


Figure D 5. Early sketches of how new skills could be assigned to agents

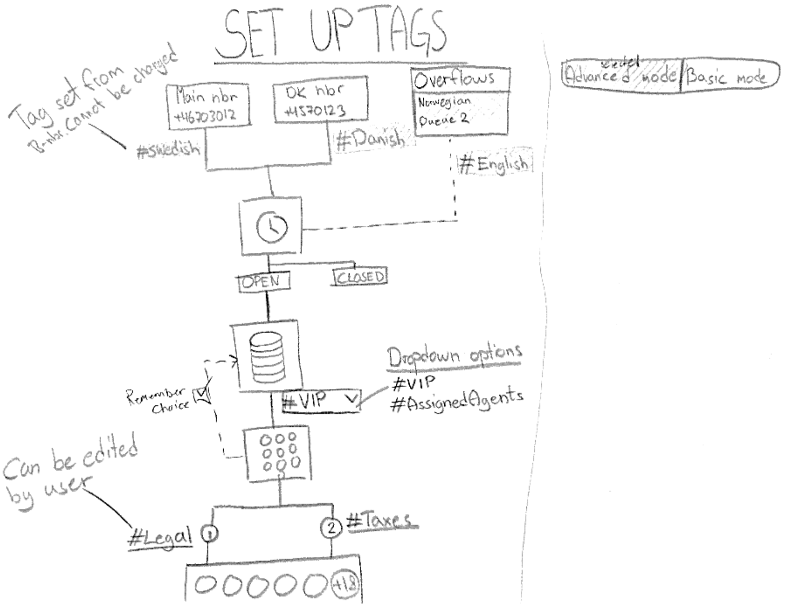
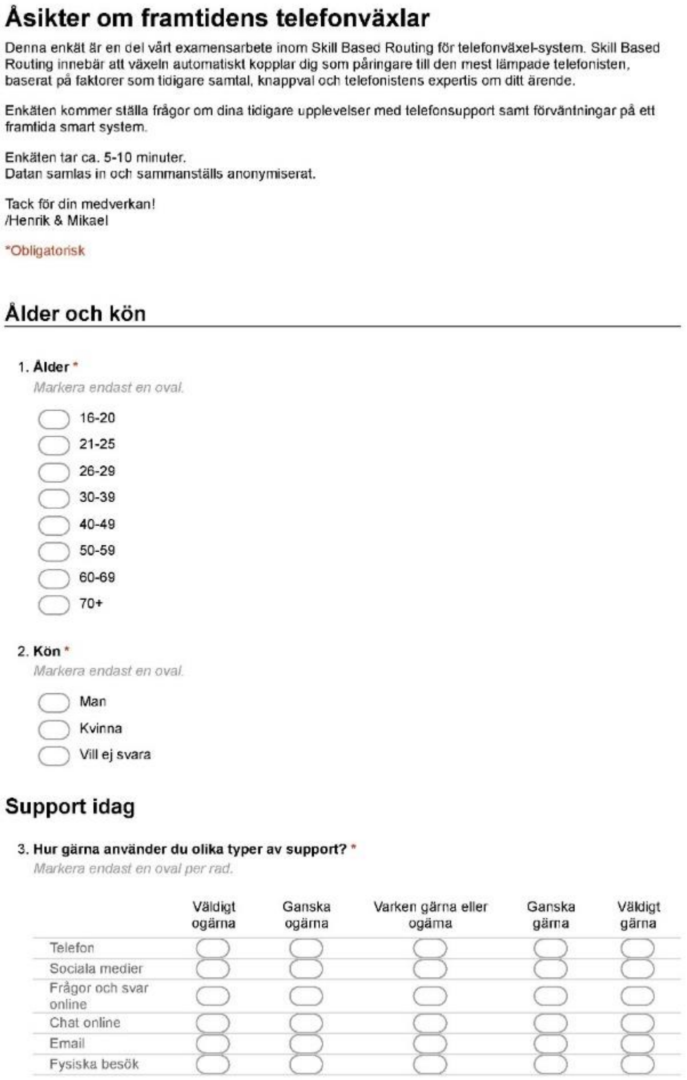


Figure D 6. Early version of setting up skills, here named as tags. The image shows which numbers lead to the PBX, including overflowing queues that end up in it.

Appendix E Caller preference survey

The survey was created to evaluate how the system should perform when clients call a smart PBX system and is presented in detail here.



Åsikter om framtidens telefonväxlar

Denna enkät är en del vårt examensarbete inom Skill Based Routing för telefonväxel-system. Skill Based Routing innebär att växeln automatiskt kopplar dig som påringare till den mest lämpade telefonisten, baserat på faktorer som tidigare samtal, knappval och telefonistens expertis om ditt ärende.

Enkäten kommer ställa frågor om dina tidigare upplevelser med telefonsupport samt förväntningar på ett framtida smart system.

Enkäten tar ca. 5-10 minuter.
Datan samlas in och sammanställs anonymiserat.

Tack för din medverkan!
/Henrik & Mikael

*Obligatorisk

Ålder och kön

1. **Ålder** *

Markera endast en oval.

○ 16-20
○ 21-25
○ 26-29
○ 30-39
○ 40-49
○ 50-59
○ 60-69
○ 70+

2. **Kön** *

Markera endast en oval.

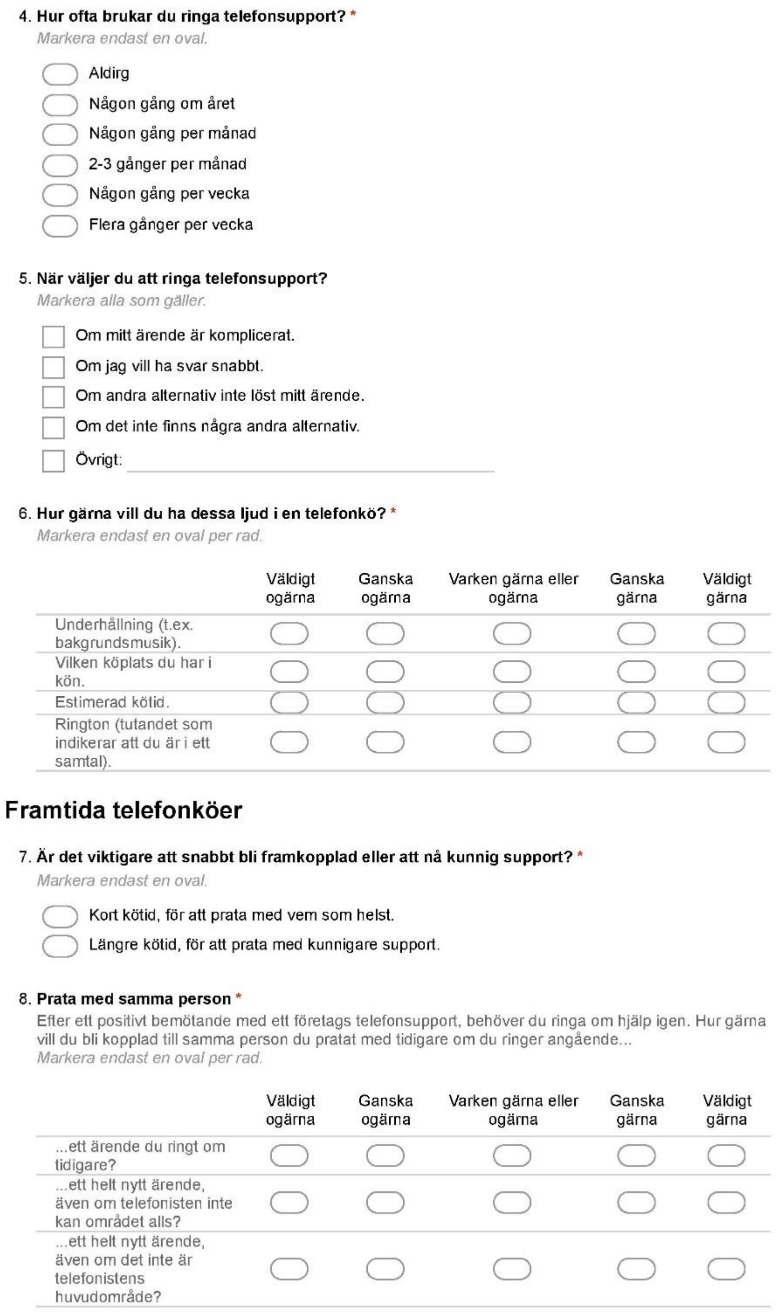
○ Man
○ Kvinna
○ Vill ej svara

Support idag

3. **Hur gärna använder du olika typer av support?** *

Markera endast en oval per rad.

	Väldigt ogärna	Ganska ogärna	Varken gärna eller ogärna	Ganska gärna	Väldigt gärna
Telefon	○	○	○	○	○
Sociala medier	○	○	○	○	○
Frågor och svar online	○	○	○	○	○
Chat online	○	○	○	○	○
Email	○	○	○	○	○
Fysiska besök	○	○	○	○	○



4. **Hur ofta brukar du ringa telefonsupport?** *

Markera endast en oval.

○ Aldirg
○ Någon gång om året
○ Någon gång per månad
○ 2-3 gånger per månad
○ Någon gång per vecka
○ Flera gånger per vecka

5. **När väljer du att ringa telefonsupport?**

Markera alla som gäller.

□ Om mitt ärende är komplicerat.
□ Om jag vill ha svar snabbt.
□ Om andra alternativ inte löst mitt ärende.
□ Om det inte finns några andra alternativ.
□ Övrigt: ____________________

6. **Hur gärna vill du ha dessa ljud i en telefonkö?** *

Markera endast en oval per rad.

	Väldigt ogärna	Ganska ogärna	Varken gärna eller ogärna	Ganska gärna	Väldigt gärna
Underhållning (t.ex. bakgrundsmusik).	○	○	○	○	○
Vilken köplats du har i kön.	○	○	○	○	○
Estimerad kötid.	○	○	○	○	○
Rington (tutandet som indikerar att du är i ett samtal).	○	○	○	○	○

Framtida telefonköer

7. **Är det viktigare att snabbt bli framkopplad eller att nå kunnig support?** *

Markera endast en oval.

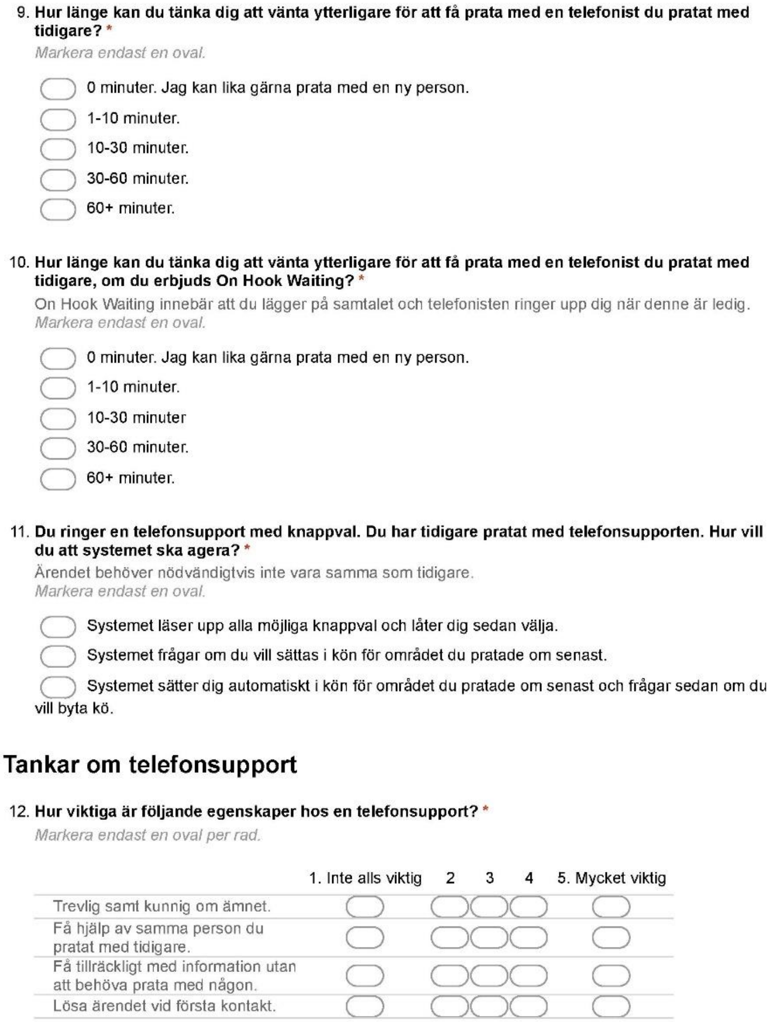
○ Kort kötid, för att prata med vem som helst.
○ Längre kötid, för att prata med kunnigare support.

8. **Prata med samma person** *

Efter ett positivt bemötande med ett företags telefonsupport, behöver du ringa om hjälp igen. Hur gärna vill du bli kopplad till samma person du pratat med tidigare om du ringer angående...

Markera endast en oval per rad.

	Väldigt ogärna	Ganska ogärna	Varken gärna eller ogärna	Ganska gärna	Väldigt gärna
...ett ärende du ringt om tidigare?	○	○	○	○	○
...ett helt nytt ärende, även om telefonisten inte kan området alls?	○	○	○	○	○
...ett helt nytt ärende, även om det inte är telefonistens huvudområde?	○	○	○	○	○



9. **Hur länge kan du tänka dig att vänta ytterligare för att få prata med en telefonist du pratat med tidigare?** *

Markera endast en oval.

○ 0 minuter. Jag kan lika gärna prata med en ny person.
○ 1-10 minuter.
○ 10-30 minuter.
○ 30-60 minuter.
○ 60+ minuter.

10. **Hur länge kan du tänka dig att vänta ytterligare för att få prata med en telefonist du pratat med tidigare, om du erbjuds On Hook Waiting?** *

On Hook Waiting innebär att du lägger på samtalet och telefonisten ringer upp dig när denne är ledig.
Markera endast en oval.

○ 0 minuter. Jag kan lika gärna prata med en ny person.
○ 1-10 minuter.
○ 10-30 minuter
○ 30-60 minuter.
○ 60+ minuter.

11. **Du ringer en telefonsupport med knappval. Du har tidigare pratat med telefonsupporten. Hur vill du att systemet ska agera?** *

Ärendet behöver nödvändigtvis inte vara samma som tidigare.
Markera endast en oval.

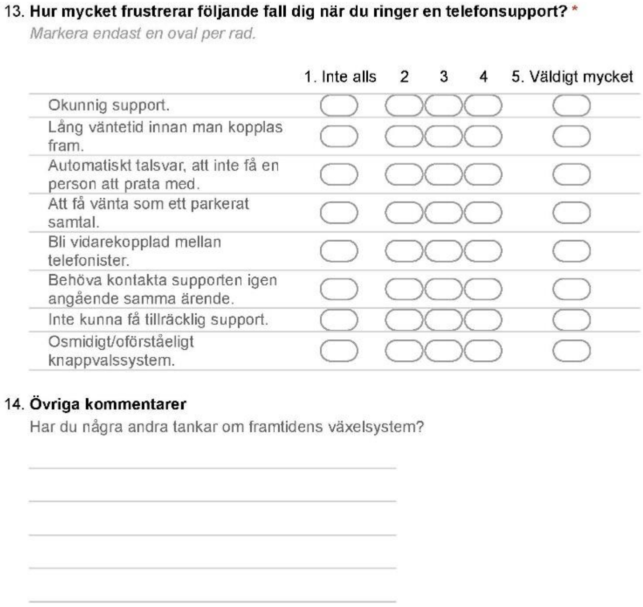
○ Systemet läser upp alla möjliga knappval och låter dig sedan välja.
○ Systemet frågar om du vill sättas i kön för området du pratade om senast.
○ Systemet sätter dig automatiskt i kön för området du pratade om senast och frågar sedan om du vill byta kö.

Tankar om telefonsupport

12. **Hur viktiga är följande egenskaper hos en telefonsupport?** *

Markera endast en oval per rad.

	1. Inte alls viktig	2	3	4	5. Mycket viktig
Trevlig samt kunnig om ämnet.	○	○	○	○	○
Få hjälp av samma person du pratat med tidigare.	○	○	○	○	○
Få tillräckligt med information utan att behöva prata med någon.	○	○	○	○	○
Lösa ärendet vid första kontakt.	○	○	○	○	○



13. **Hur mycket frustrerar följande fall dig när du ringer en telefonsupport?** *

Markera endast en oval per rad.

	1. Inte alls	2	3	4	5. Väldigt mycket
Okunnig support.	○	○	○	○	○
Lång väntetid innan man kopplas fram.	○	○	○	○	○
Automatiskt talsvar, att inte få en person att prata med.	○	○	○	○	○
Att få vänta som ett parkerat samtal.	○	○	○	○	○
Bli vidarekopplad mellan telefonister.	○	○	○	○	○
Behöva kontakta supporten igen angående samma ärende.	○	○	○	○	○
Inte kunna få tillräcklig support.	○	○	○	○	○
Osmidigt/oförståeligt knappvalssystem.	○	○	○	○	○

14. **Övriga kommentarer**

Har du några andra tankar om framtidens växelsystem?

Appendix F Caller preference survey results

The data from the surveys was used to evaluate the client flowcharts and decide on the most suitable way to set up the backend.

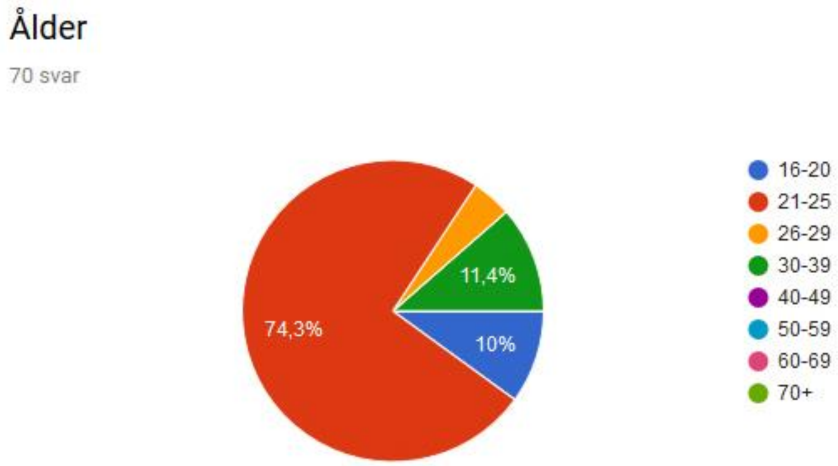


Figure F 1. The age distribution of the respondents.

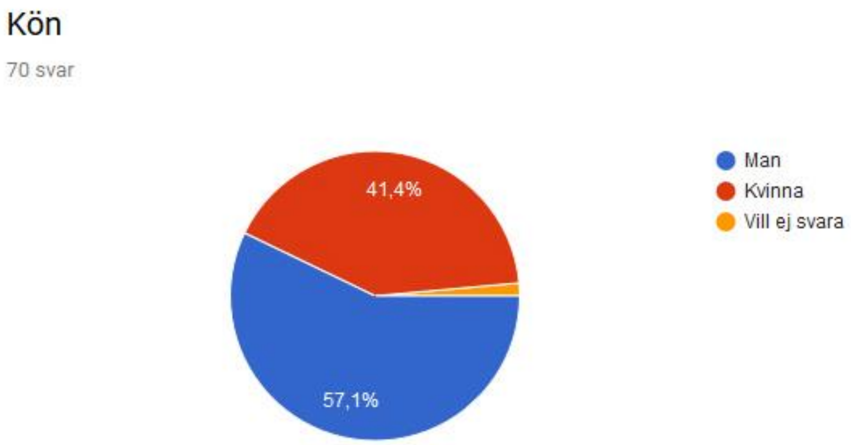


Figure F 2. The distribution between men and women.

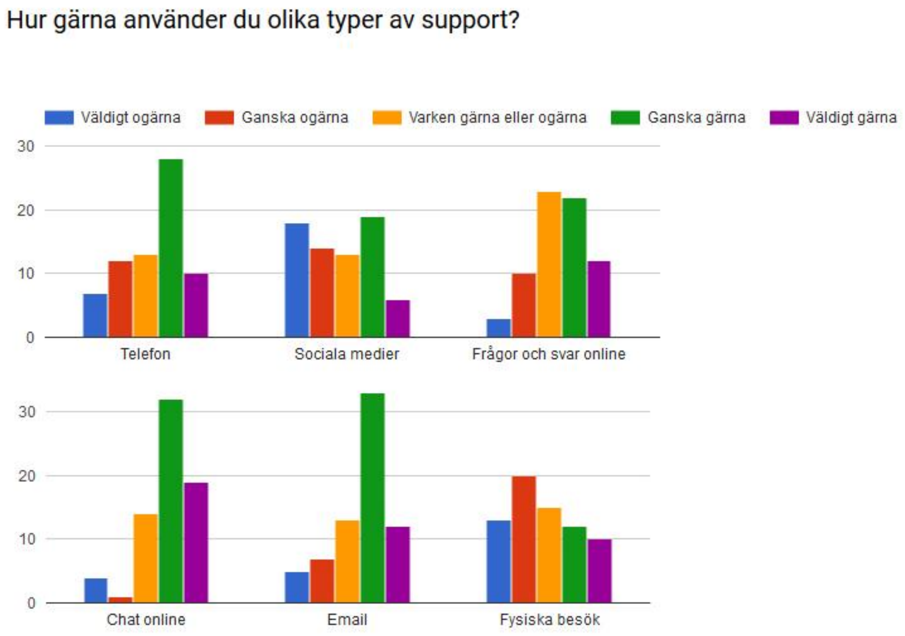


Figure F 3. What kind of support the respondents prefer.

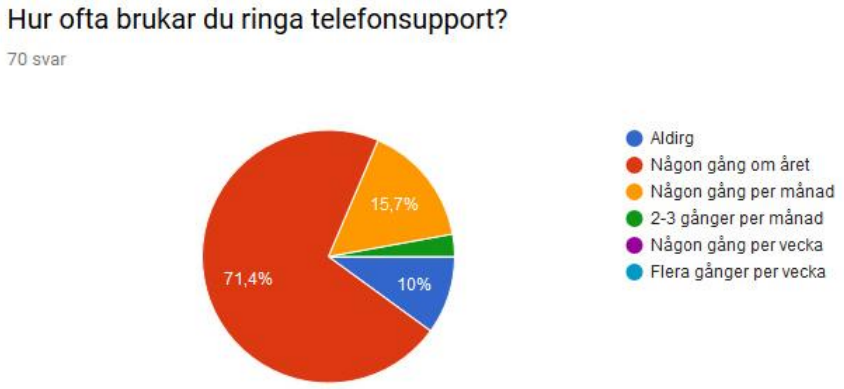


Figure F 4. How often the respondents contact telephone support.

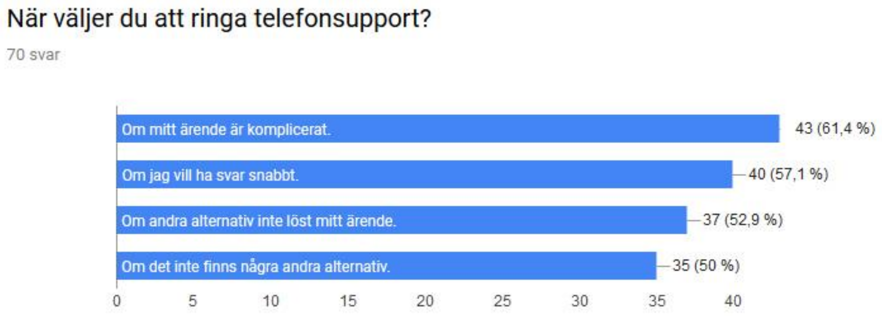


Figure F 5. When the respondents choose to call telephone support.

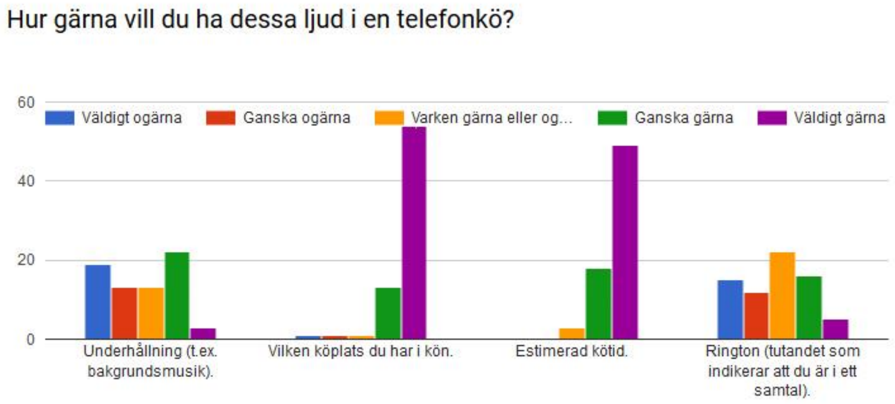


Figure F 6. What kind of auditory feedback the clients prefer.

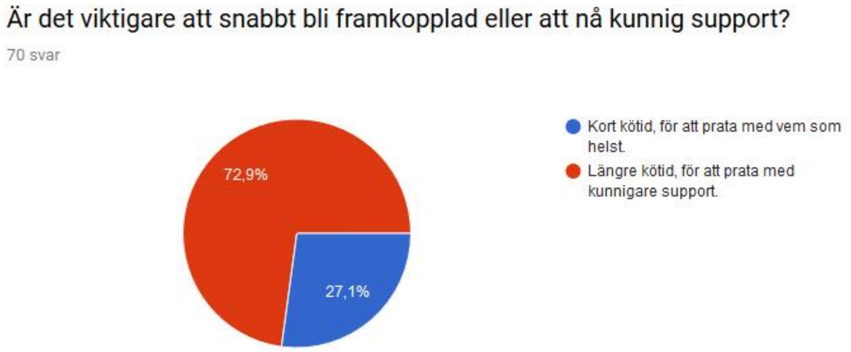


Figure F 7. Whether it is worth waiting for a more knowledgeable support.

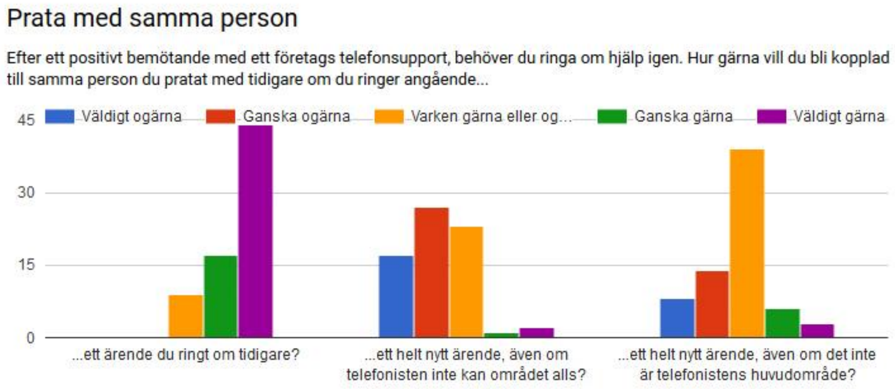
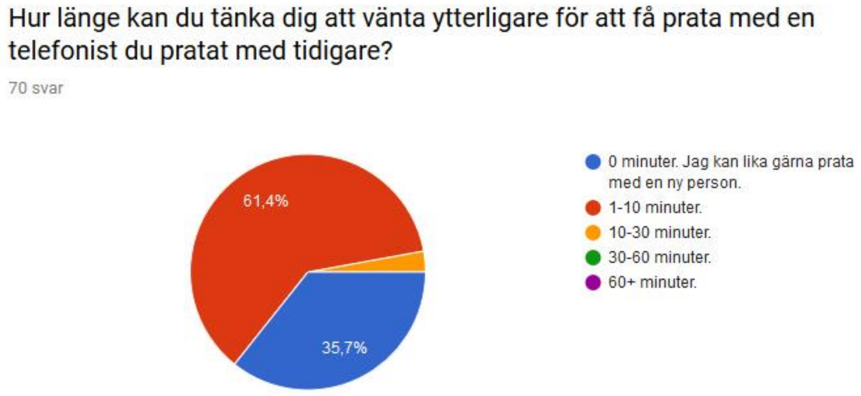


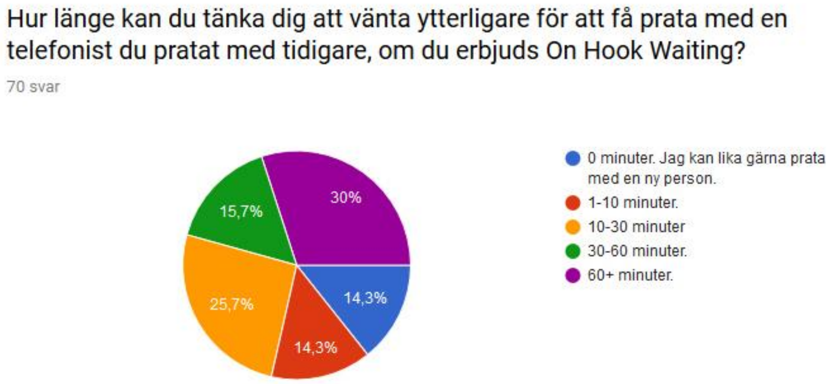
Figure F 8. When the respondents would like to talk to the same agent as in an earlier conversation.



Hur länge kan du tänka dig att vänta ytterligare för att få prata med en telefonist du pratat med tidigare?

70 svar

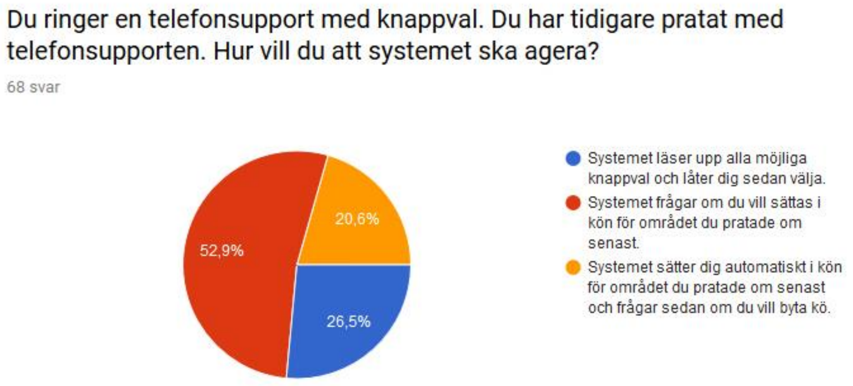
Figure F 9. How long the respondents are willing to wait additionally to talk to the same agent as in an earlier conversation.



Hur länge kan du tänka dig att vänta ytterligare för att få prata med en telefonist du pratat med tidigare, om du erbjuds On Hook Waiting?

70 svar

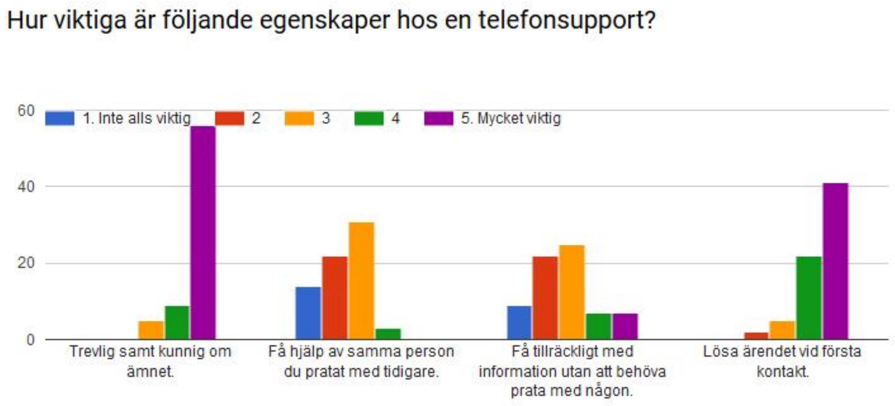
Figure F 10. How long the respondents are willing to wait additionally to talk to the same agent as in an earlier conversation if they are offered on hook waiting.



Du ringer en telefonsupport med knappval. Du har tidigare pratat med telefonsupporten. Hur vill du att systemet ska agera?

68 svar

Figure F 11. How the respondents want the system to act when they call again.



Hur viktiga är följande egenskaper hos en telefonsupport?

Figure F 12. The most important factors in a support system.

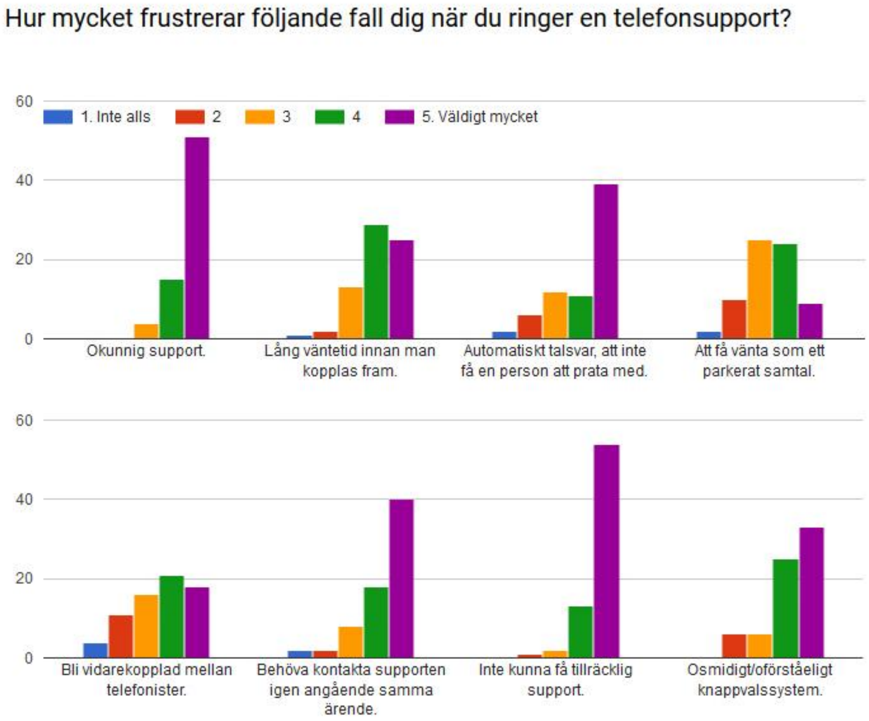
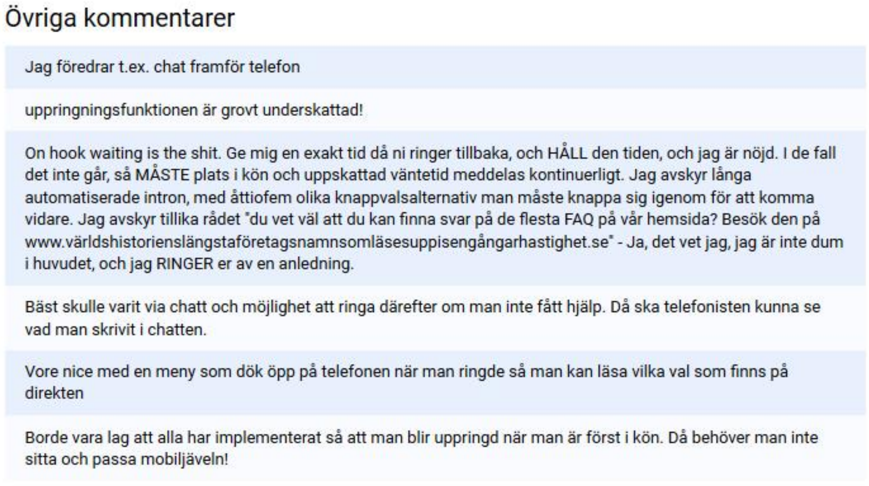


Figure F 13. What frustrates the respondents the most on a support system.



Övriga kommentarer

Jag föredrar t.ex. chat framför telefon

uppringningsfunktionen är grovt underskattad!

On hook waiting is the shit. Ge mig en exakt tid då ni ringer tillbaka, och HÅLL den tiden, och jag är nöjd. I de fall det inte går, så MÅSTE plats i kön och uppskattad väntetid meddelas kontinuerligt. Jag avskyr långa automatiserade intron, med åttiofem olika knappvalsalternativ man måste knappa sig igenom för att komma vidare. Jag avskyr tillika rådet "du vet väl att du kan finna svar på de flesta FAQ på vår hemsida? Besök den på www.världshistorienslängstaföretagsnamnsomläsesuppisengångarhastighet.se" - Ja, det vet jag, jag är inte dum i huvudet, och jag RINGER er av en anledning.

Bäst skulle varit via chatt och möjlighet att ringa därefter om man inte fått hjälp. Då ska telefonisten kunna se vad man skrivit i chatten.

Vore nice med en meny som dök öpp på telefonen när man ringde så man kan läsa vilka val som finns på direkten

Borde vara lag att alla har implementerat så att man blir uppringd när man är först i kön. Då behöver man inte sitta och passa mobiljäveln!

Figure F 14. Other comments from the respondents.

Appendix G Test plan for high-fidelity evaluation

Here is the test plan that was followed before, during and after the testing of the high-fidelity presented

Test plan - SBR

Henrik Edlund, Mikael Nilsson

1. Purpose

The purpose of conducting these tests is to evaluate the high-fidelity prototype of the administrator and agent user interface. The results from the tests will also answer whether unmoderated remote testing is an effective way of evaluating prototypes.

2. Questions

Is the three-screen introduction to the administrator interface enough to explain the features?

- Do users understand the concept of a skill?
- Can users understand the connection to the preview tree?
- Is drag and drop a good way to assign skills to agents?
- Is assigning skills to agents better than assigning agents to skills?
- Is the strategy concept clear to the users?
- How can we explain the tiebreaker priority system overview?
- Is the priority system and swap functionality intuitive?
- Do the users understand what the lock mechanism does?
- Do the users understand the concept of Rank?

3. Data collection

The evaluation will gather the following data:

Objective/ quantitative	**Objective/ qualitative**	**Subjective/ quantitative**	**Subjective/ qualitative**
-	Observation through in-person sessions	Graded scale in survey	Comments in survey

4. Test assignments

Agent UI - Incoming call

Task	**Subtask**	**Completed when...**
1. Answer the call from Joel Stone.	1.1 Press the "Answer button".	A call with Stone is connected, the first screen is now showing.
2. Put Joel Stone on hold, then resume the conversation.	2.1 Press the gray "Pause" button. 2.2 Press the now blue "Pause" button.	The first screen is showing again after the pause screen.
3. Dial "#" and hide the numpad.	3.1 Press the numpad icon. 3.2 Press "#" on the numpad. 3.3 Press any of the showing numpad icons.	The character "#" has been shown on the dialer screen and the numpad has been hidden.
4. Lock yourself as first agent to answer Stone.	4.1 Press the padlock icon or your user's profile thumbnail.	The padlock icon is locked and has a blue color.
5. Reprioritize answering agents.	5.1 Press the switch button twice.	Adam Luaren is the first agent. Your Name is the second agent.

6. Transfer call.	6.1 Press “This computer” under “Device”. 6.2 Press “Your smartphone”. 6.3 Press the transfer button.	The screen showing is the answered call screen on the smartphone.
7. Mute microphone.	7.1 Press the gray microphone button. 7.2 Press the now blue microphone button.	The answer screen is showing again.
8. End call	8.1 Press the red end call button.	The call queue screen is showing.

Agent UI - Call history

Task	**Subtask**	**Completed when...**
1. Joel Stone call check	1.1 Press the History tab. 1.2 Press the Joel Stone card. 1.3 See how many conversations you have had with Joel Stone.	There are six conversations showing.
2. Lock yourself to Stone.	2.1 Press the switch button. 2.2 Press the icon to the left of Your Name.	Your Name has been set as the primary answering agent for Stone.
3. Call Stone	3.1 Press the green call icon.	A call with Stone has been established.
4. End the call	4.1 Press the end call button	The call queue can be seen on the screen.

Administrator UI

Task	**Subtask**	**Correctly completed when..**
1. Go through the introduction guide and navigate to the setup process.	1.1 Read the text and press the “Next” arrow, repeat three times. 1.2 Click the “+”-button to open the menu. 1.3 Click the “skill pool” button.	The first step in the setup is reached.
2. Define the different skills of your company and continue.	1.1 Click on Add a new skill. 1.2 Click add. 1.3 Click on add a new skill. 1.4 Click on add. 1.5 Click on add a new skill. 1.6 Click on choose a parent. 1.7 Choose Malmö as parent. 1.8 Click on add. 1.5 Click on add a new skill. 1.6 Click on choose a parent. 1.7 Choose Malmö as parent. 1.8 Click on add. 1.9 Click on Next.	The list contains 4 skills and 2 of them are child skills to Malmö. The user navigates to the next screen.
3. Fanny Harvey is primarily located in the Lund office. Add her to the Pool and assign her to the Lund office.	3.1 Click on add new agent. 3.2 Click on fanny and then “Add”. 3.3 Click on the skill “Lund”. 3.4 Click on the “Drop Here” in the primary skill column.	Fanny is added and assigned the skill.
4. Speed up the process by adding all of your remaining available agents to the pool.	4.1 Click on add new agent. 4.2 Click on all the agents to select them. 4.3 Click on “Add agents”.	All five agents appear in the agent column.
5. Fanny, Clifford and Josephine all work in the Malmö Billing section. Shawn will also take	5.1 Assign the skills in the corresponding manner to task 3.	Fanny, Clifford and Josephine has the billing skill as primary and Shawn has the

some calls if the queue gets busy. Assign the agents to the Malmö Billing department.		billing skill as secondary.
6. Assign the rest of the skills to your agents and continue.	6.1 Assign the skills in the corresponding manner to task 3.	The call strategy screen is reached.
7. Choose the call strategy you think fits your company best.	7.1 Click on any of the call strategies to select them. 7.2 Click Finish.	The user clicked finish and is presented with the tree structure overview.
8. Look who is the best agent to answer Adele Green's call.	8.1 Go into the administration pane. 8.2 Search for Adele Green by clicking the search bar.	The user sees who is primary agent to answer Adele Green.
9. Take a look to check that all the skills you defined for the Pool are saved.	9.1 Navigate back to the PBX pane. 9.2 Click on the skill pool. 9.3 Click on the edit skills pane.	

5. Execution

The test will be done in an unmoderated remote manner. The test consists of two parts, an administrator interface and an agent interface. An email will be send out to the test participants containing a link to the InVision project and a survey link. There will be two different emails, one starting with the administrator setup and one starting with the agent interface. This is to combat the transfer of learning effects of the different systems.

6. Test participants

The test participants will be experienced users in-house at Telavox as well as people outside Telavox fitting certain personas. There will be in total ten experienced plus ten less experienced test participants.

	Less Experienced	Experienced
Home	10	10
Flow	10	10

Examples of interesting personas for the test:

- Experienced in both Flow and Home.
- Experienced in Flow.
- Experienced in Home, the older version of Flow.
- Knows about the skill based routing concept.
- Does not know anything about skill based routing.

A list of in-house participants will be provided by Henrik Thorvinger, consisting of people from the UX, Sales, Development and Advisor teams at Telavox.

7. Test environment and tools

The tests will be performed by users on their own computers, using their preferred browser and the links provided to the prototypes.

8. Roles

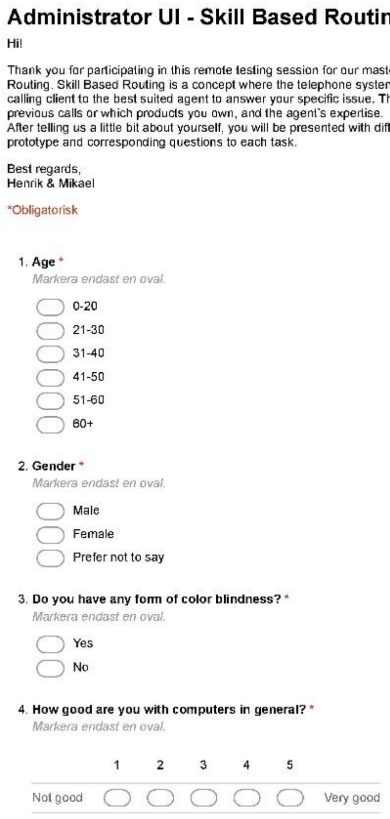
No roles are needed for the remote testing.

9. Result reporting

The results from the remote testing will be compiled by Google forms. The data that will be gathered can be represented in diagrams, tables or matrices and will be included in the report.

Appendix H High-fidelity evaluation forms

The forms in this section, Administrator UI and Agent UI, were created to evaluate the high-fidelity prototype.



Administrator UI - Skill Based Routing

Hi!

Thank you for participating in this remote testing session for our master thesis project in Skill Based Routing. Skill Based Routing is a concept where the telephone system automatically will connect you as a calling client to the best suited agent to answer your specific issue. This routing is done based on e.g. previous calls or which products you own, and the agent`s expertise.
After telling us a little bit about yourself, you will be presented with different tasks to be performed in the prototype and corresponding questions to each task.

Best regards,
Henrik & Mikael

*Obligatorisk

1. **Age** *

Markera endast en oval.

○ 0-20
○ 21-30
○ 31-40
○ 41-50
○ 51-60
○ 60+

2. **Gender** *

Markera endast en oval.

○ Male
○ Female
○ Prefer not to say

3. **Do you have any form of color blindness?** *

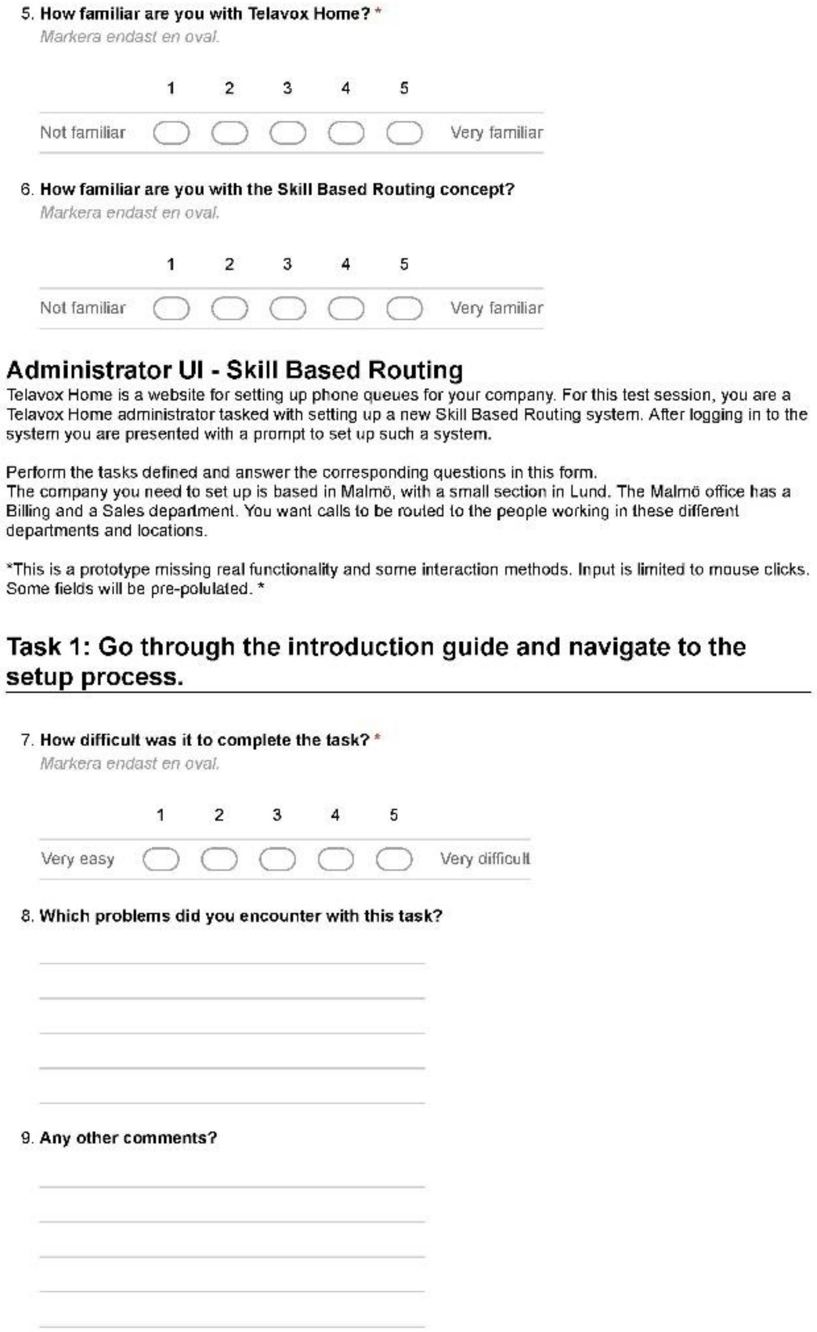
Markera endast en oval.

○ Yes
○ No

4. **How good are you with computers in general?** *

Markera endast en oval.

	1	2	3	4	5	
Not good	○	○	○	○	○	Very good



5. **How familiar are you with Telavox Home?** *

Markera endast en oval.

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

6. **How familiar are you with the Skill Based Routing concept?**

Markera endast en oval.

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

Administrator UI - Skill Based Routing

Telavox Home is a website for setting up phone queues for your company. For this test session, you are a Telavox Home administrator tasked with setting up a new Skill Based Routing system. After logging in to the system you are presented with a prompt to set up such a system.

Perform the tasks defined and answer the corresponding questions in this form.
The company you need to set up is based in Malmö, with a small section in Lund. The Malmö office has a Billing and a Sales department. You want calls to be routed to the people working in these different departments and locations.

*This is a prototype missing real functionality and some interaction methods. Input is limited to mouse clicks. Some fields will be pre-polulated. *

Task 1: Go through the introduction guide and navigate to the setup process.

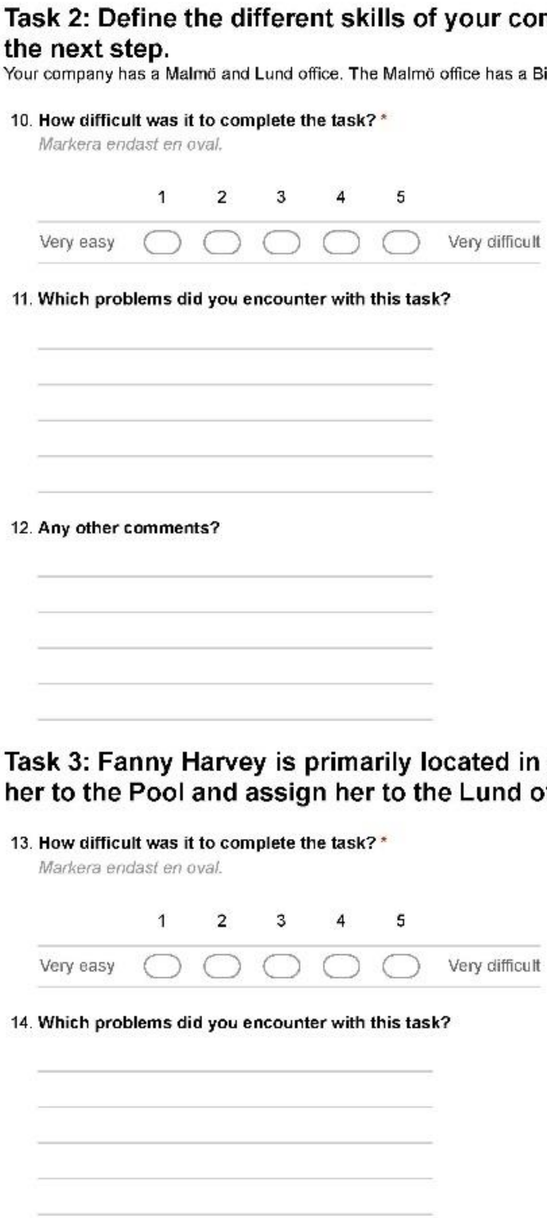
7. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

8. **Which problems did you encounter with this task?**

9. **Any other comments?**



Task 2: Define the different skills of your company and continue to the next step.

Your company has a Malmö and Lund office. The Malmö office has a Billing and a Sales department.

10. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

11. **Which problems did you encounter with this task?**

12. **Any other comments?**

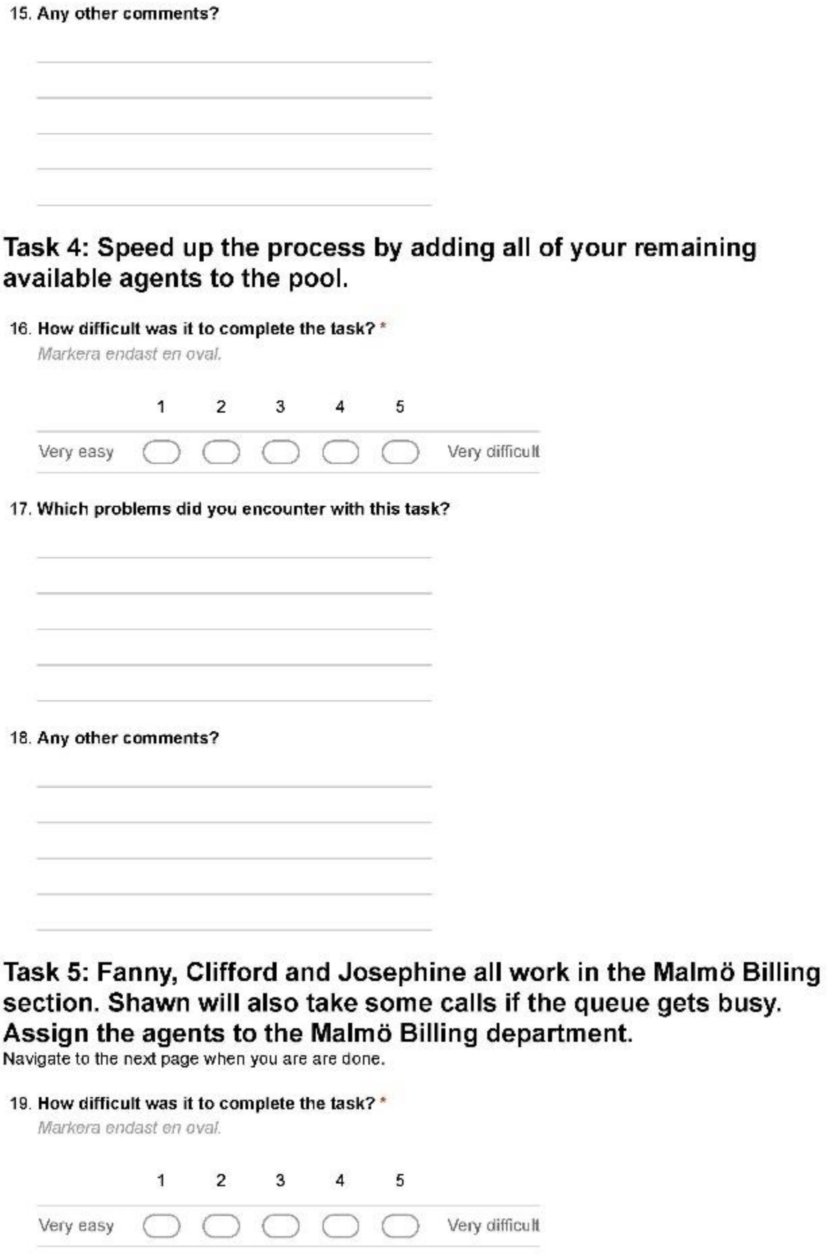
Task 3: Fanny Harvey is primarily located in the Lund office. Add her to the Pool and assign her to the Lund office.

13. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

14. **Which problems did you encounter with this task?**



15. **Any other comments?**

Task 4: Speed up the process by adding all of your remaining available agents to the pool.

16. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

17. **Which problems did you encounter with this task?**

18. **Any other comments?**

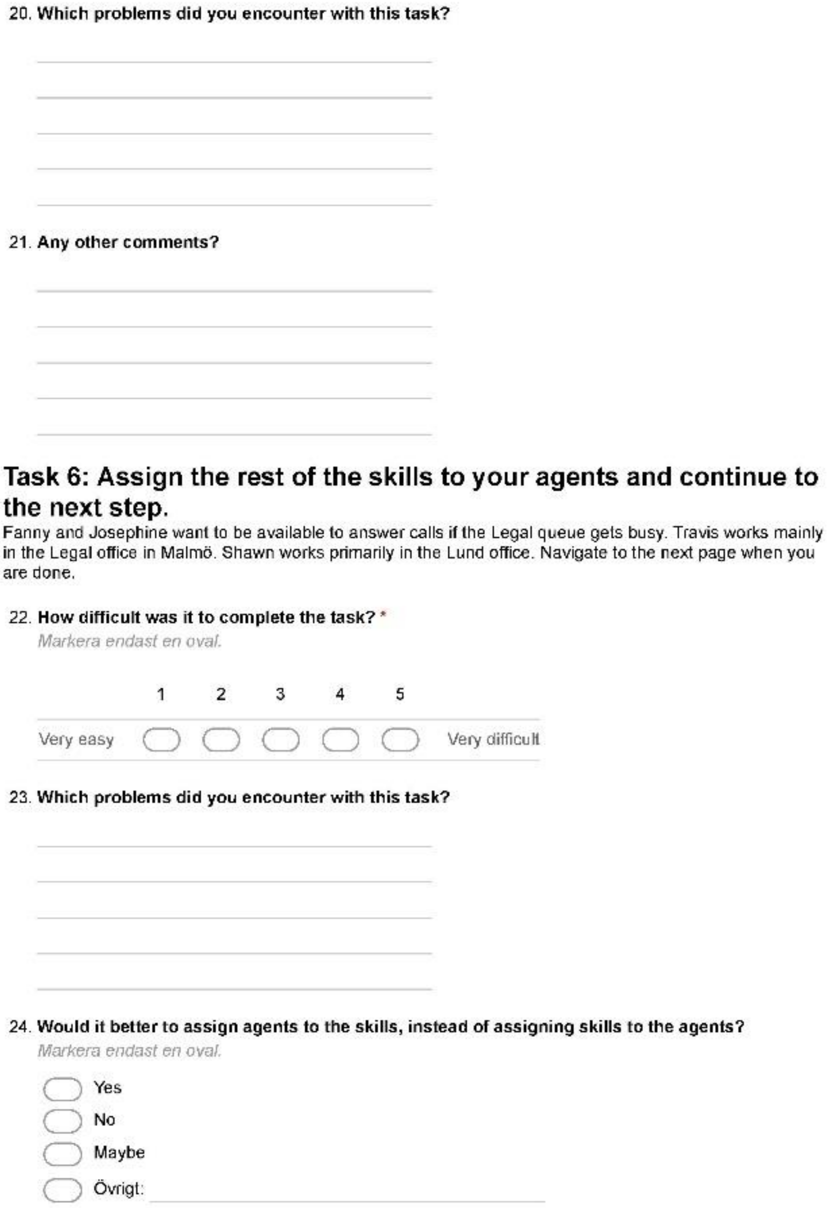
Task 5: Fanny, Clifford and Josephine all work in the Malmö Billing section. Shawn will also take some calls if the queue gets busy. Assign the agents to the Malmö Billing department.

Navigate to the next page when you are are done.

19. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult



20. **Which problems did you encounter with this task?**

21. **Any other comments?**

Task 6: Assign the rest of the skills to your agents and continue to the next step.

Fanny and Josephine want to be available to answer calls if the Legal queue gets busy. Travis works mainly in the Legal office in Malmö. Shawn works primarily in the Lund office. Navigate to the next page when you are done.

22. **How difficult was it to complete the task?** *

Markera endast en oval.

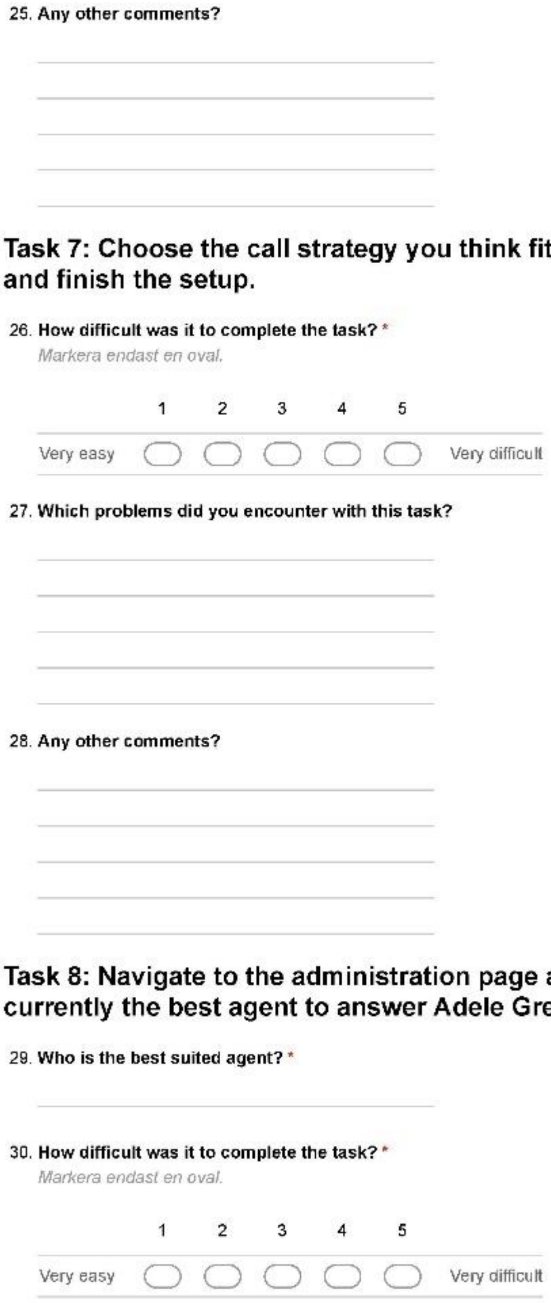
	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

23. **Which problems did you encounter with this task?**

24. **Would it better to assign agents to the skills, instead of assigning skills to the agents?**

Markera endast en oval.

○ Yes
○ No
○ Maybe
○ Övrigt: ____________



25. **Any other comments?**

Task 7: Choose the call strategy you think fits your company best and finish the setup.

26. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

27. **Which problems did you encounter with this task?**

28. **Any other comments?**

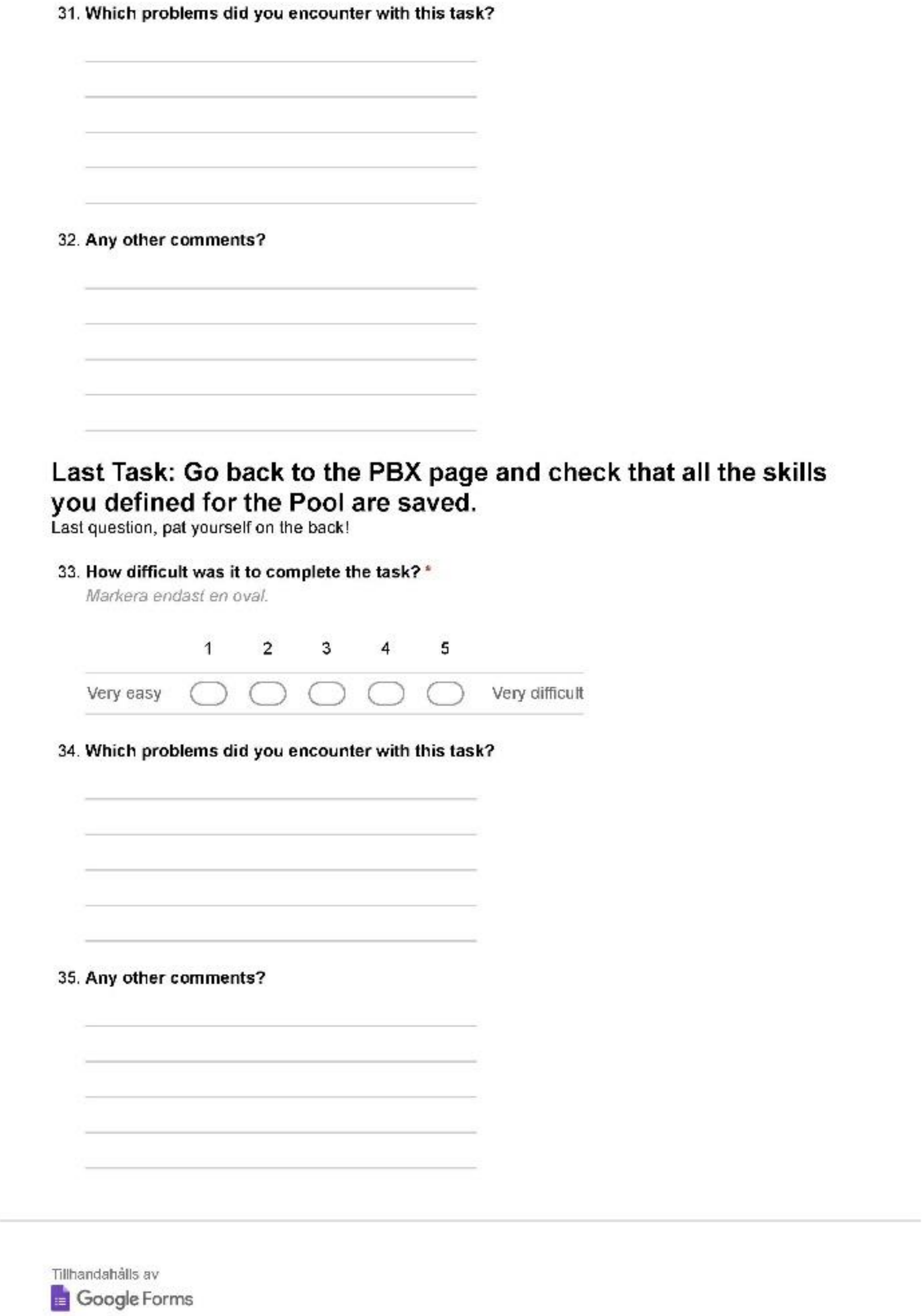
Task 8: Navigate to the administration page and look who is currently the best agent to answer Adele Green's call.

29. **Who is the best suited agent?** *

30. **How difficult was it to complete the task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult



31. **Which problems did you encounter with this task?**

32. **Any other comments?**

Last Task: Go back to the PBX page and check that all the skills you defined for the Pool are saved.

Last question, pat yourself on the back!

33. **How difficult was it to complete the task?** *

Markera endast en oval.

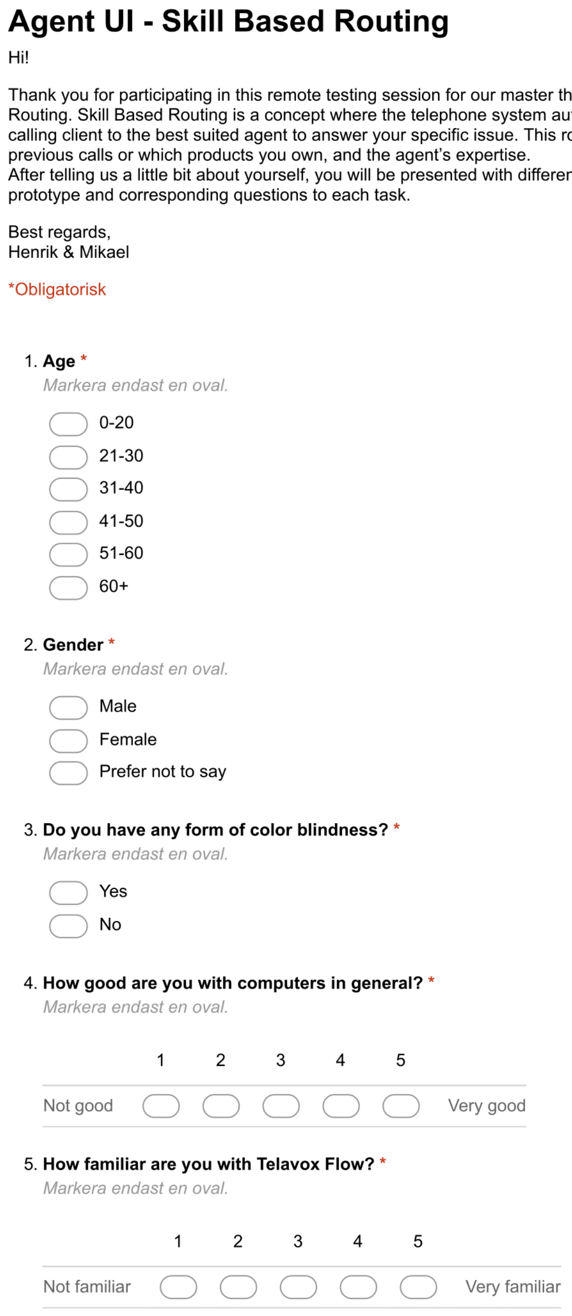
	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

34. **Which problems did you encounter with this task?**

35. **Any other comments?**

Tillhandahålls av

Google Forms



Agent UI - Skill Based Routing

Hi!

Thank you for participating in this remote testing session for our master thesis project in Skill Based Routing. Skill Based Routing is a concept where the telephone system automatically will connect you as a calling client to the best suited agent to answer your specific issue. This routing is done based on e.g. previous calls or which products you own, and the agent's expertise.
After telling us a little bit about yourself, you will be presented with different tasks to be performed in the prototype and corresponding questions to each task.

Best regards,
Henrik & Mikael

*Obligatorisk

1. **Age** *

Markera endast en oval.

○ 0-20
○ 21-30
○ 31-40
○ 41-50
○ 51-60
○ 60+

2. **Gender** *

Markera endast en oval.

○ Male
○ Female
○ Prefer not to say

3. **Do you have any form of color blindness?** *

Markera endast en oval.

○ Yes
○ No

4. **How good are you with computers in general?** *

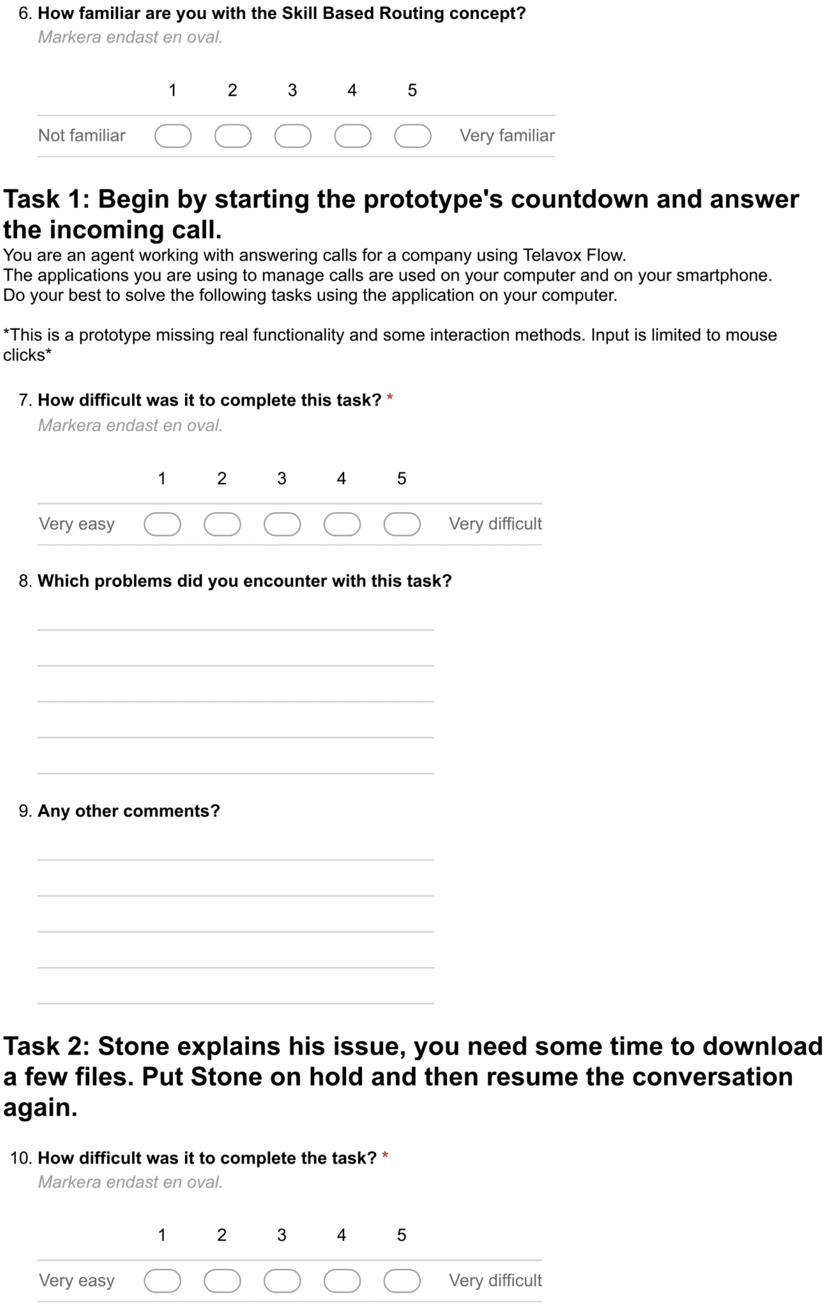
Markera endast en oval.

	1	2	3	4	5	
Not good	○	○	○	○	○	Very good

5. **How familiar are you with Telavox Flow?** *

Markera endast en oval.

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar



6. **How familiar are you with the Skill Based Routing concept?**
Markera endast en oval.

	1	2	3	4	5	
Not familiar	○	○	○	○	○	Very familiar

Task 1: Begin by starting the prototype's countdown and answer the incoming call.

You are an agent working with answering calls for a company using Telavox Flow.
The applications you are using to manage calls are used on your computer and on your smartphone.
Do your best to solve the following tasks using the application on your computer.

This is a prototype missing real functionality and some interaction methods. Input is limited to mouse clicks

7. **How difficult was it to complete this task?** *
Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

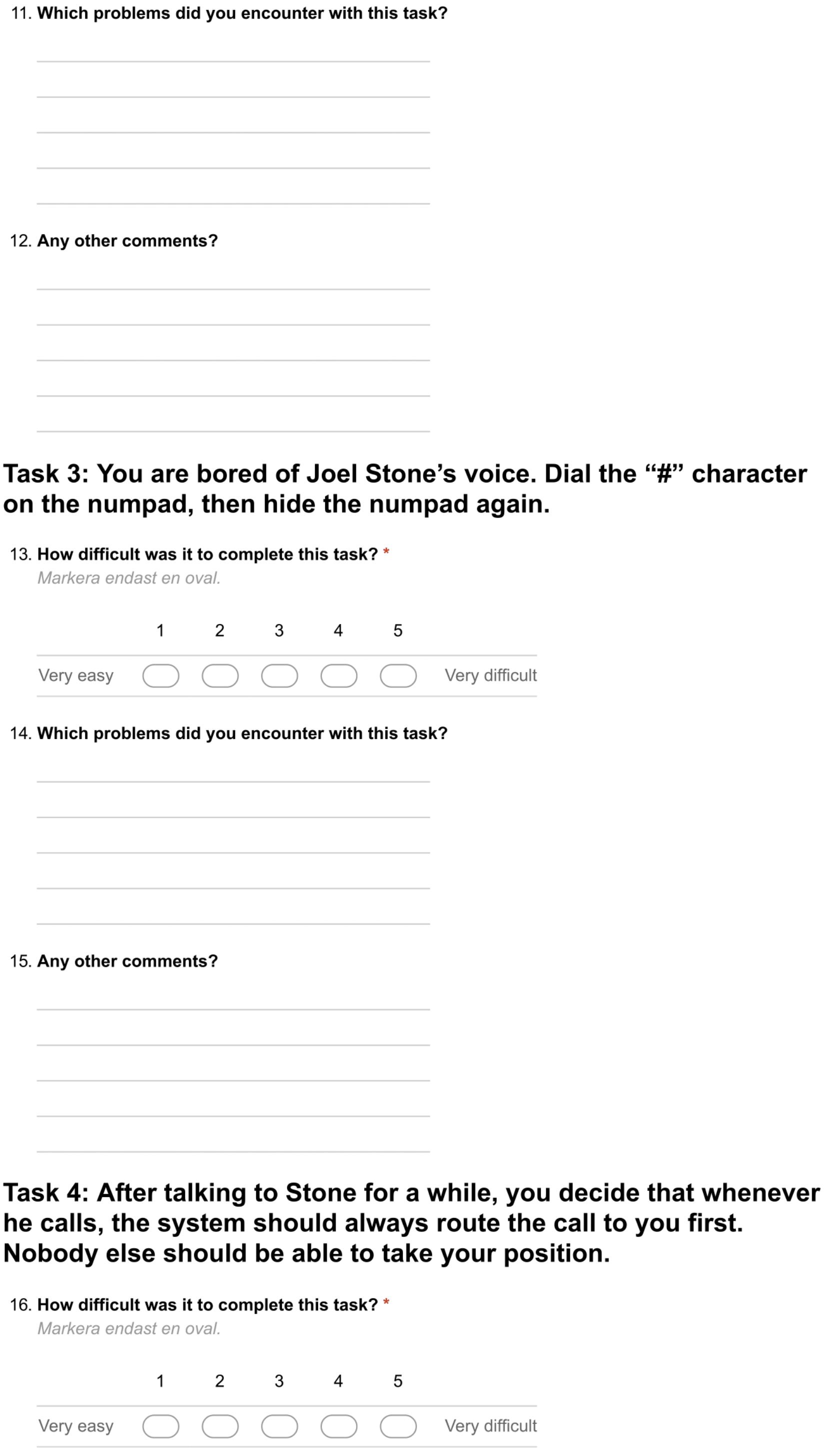
8. **Which problems did you encounter with this task?**

9. **Any other comments?**

Task 2: Stone explains his issue, you need some time to download a few files. Put Stone on hold and then resume the conversation again.

10. **How difficult was it to complete the task?** *
Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult



11. **Which problems did you encounter with this task?**

12. **Any other comments?**

Task 3: You are bored of Joel Stone's voice. Dial the "#" character on the numpad, then hide the numpad again.

13. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

14. **Which problems did you encounter with this task?**

15. **Any other comments?**

Task 4: After talking to Stone for a while, you decide that whenever he calls, the system should always route the call to you first. Nobody else should be able to take your position.

16. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

17. **Which problems did you encounter with this task?**

18. **Any other comments?**

Task 5: You realise that your co-worker Adam Luaren is better suited to answer Stone. Set Luaren as the first answering agent for Stone, but set yourself as the secondary answering agent.

19. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

20. **Which problems did you encounter with this task?**

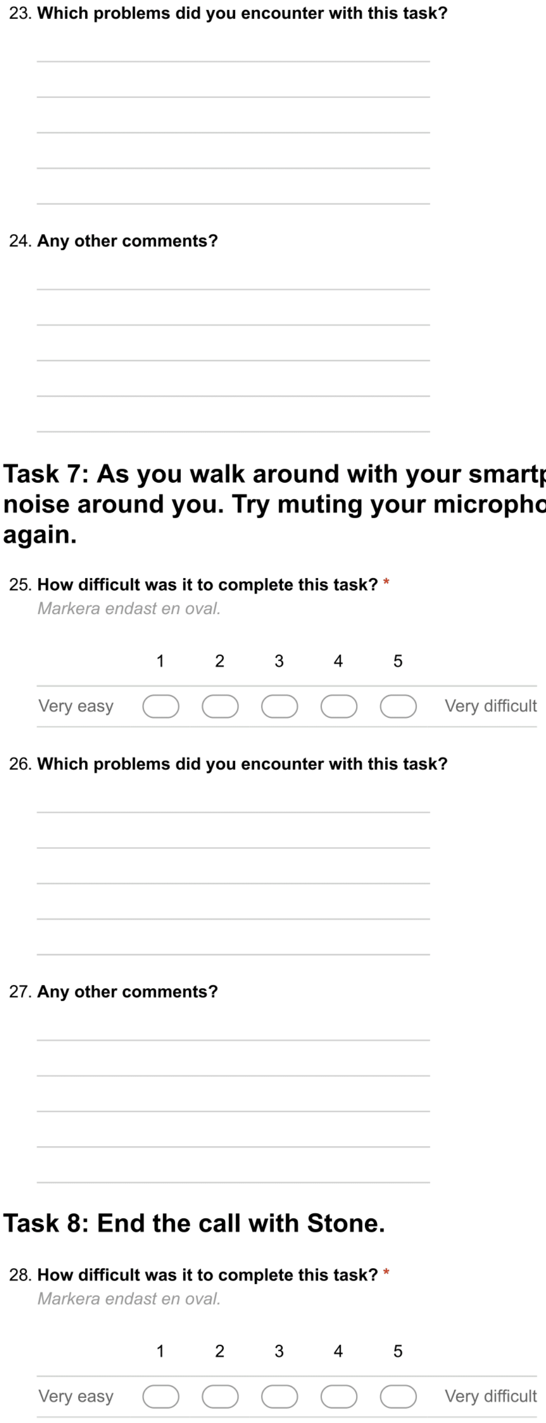
21. **Any other comments?**

Task 6: You now need to move from your computer. Transfer the call to your smartphone.

22. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult



23. **Which problems did you encounter with this task?**

24. **Any other comments?**

Task 7: As you walk around with your smartphone, there is a lot of noise around you. Try muting your microphone, then unmute it again.

25. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

26. **Which problems did you encounter with this task?**

27. **Any other comments?**

Task 8: End the call with Stone.

28. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

29. **Which problems did you encounter with this task?**

30. **Any other comments?**

Task 9: Check how many times you have talked to Joel Stone in total.

31. **How many conversations have taken place with Stone?** *

Markera endast en oval.

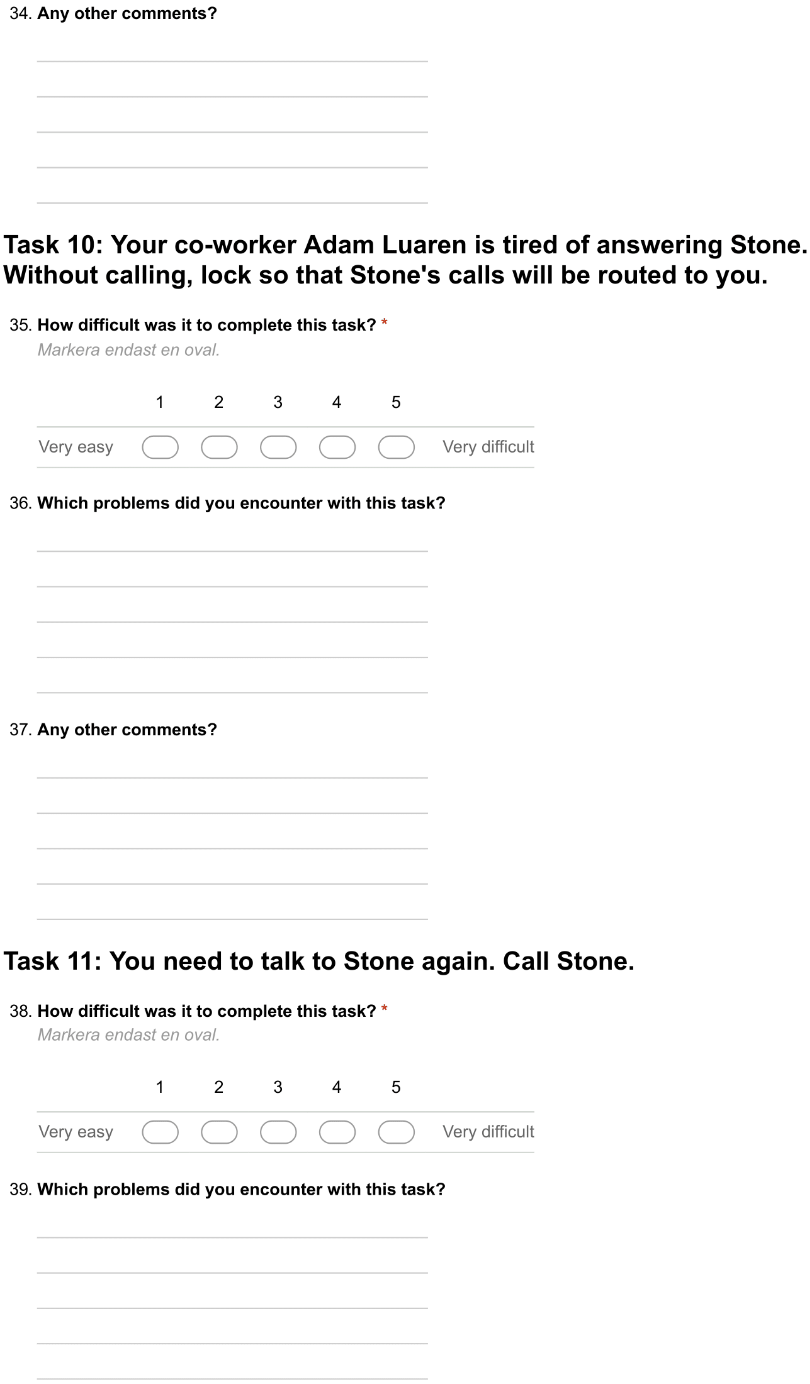
- ○ 0
- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ 6
- ○ 7
- ○ 8
- ○ 9
- ○ 10+

32. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

33. **Which problems did you encounter with this task?**



34. **Any other comments?**

Task 10: Your co-worker Adam Luaren is tired of answering Stone. Without calling, lock so that Stone's calls will be routed to you.

35. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

36. **Which problems did you encounter with this task?**

37. **Any other comments?**

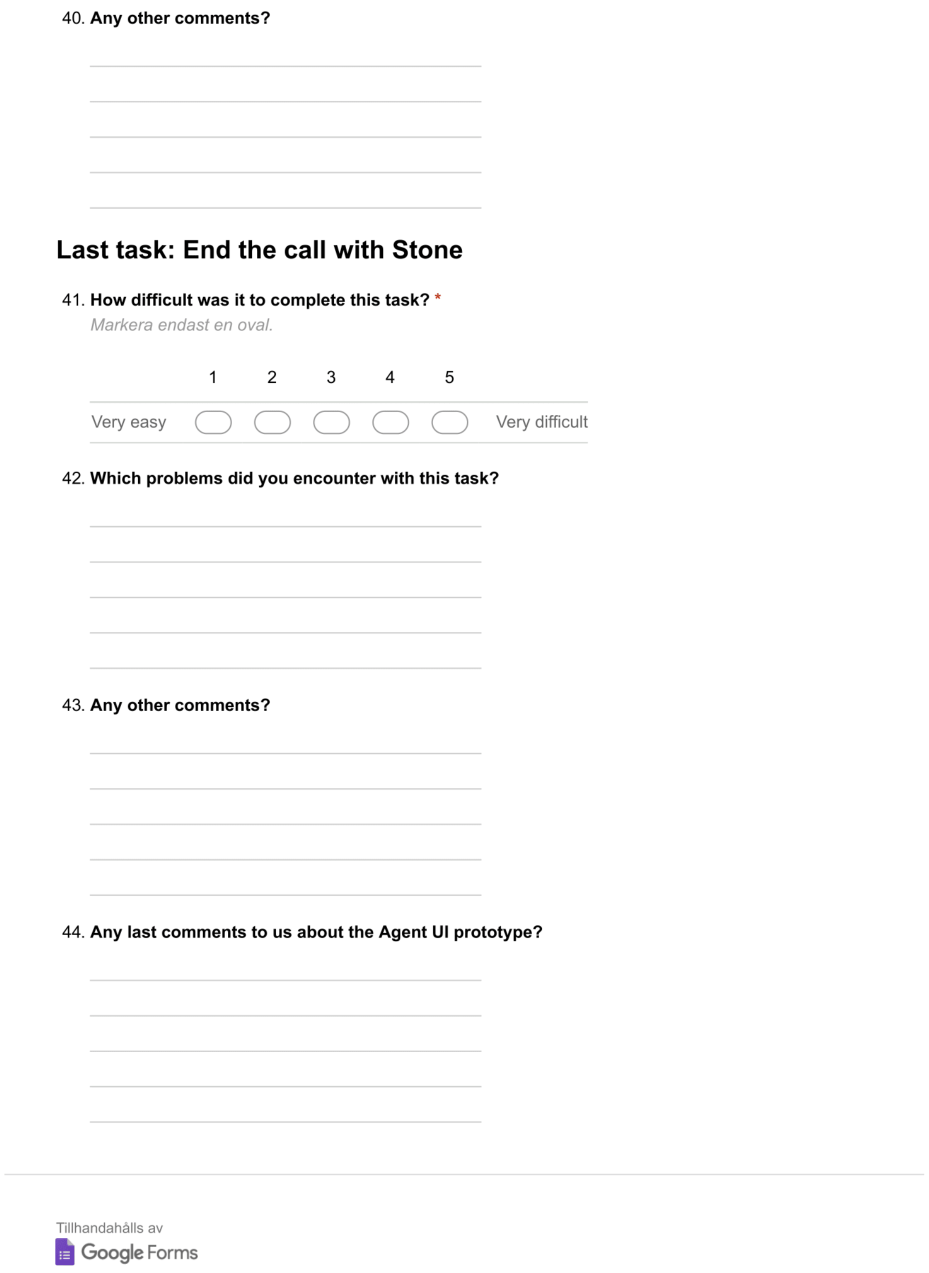
Task 11: You need to talk to Stone again. Call Stone.

38. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

39. **Which problems did you encounter with this task?**



40. **Any other comments?**

Last task: End the call with Stone

41. **How difficult was it to complete this task?** *

Markera endast en oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very difficult

42. **Which problems did you encounter with this task?**

43. **Any other comments?**

44. **Any last comments to us about the Agent UI prototype?**

Tillhandahålls av

Google Forms