Zoran Prodanović, BSc

Analysis of Alternative HCI Concepts in the Field of Smart Dairy Farming

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. BSc Johanna Pirker

Co-Supervisor

Dipl.-Ing. Dr.techn. Tobias Rauter

Institute for Interactive Systems and Data Science (ISDS)
Head: Univ.-Prof. Dipl-Inf. Dr. Stefanie Lindstaedt

Graz, October 2019

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

Due to the rise of the Internet of Things, also known as the fourth industrial revolution, more and more devices and gadgets are interconnected over the internet. This evolution also takes place in the farming sector, where farming machines and devices are continuously developed, and modern technologies are used. As sensors are becoming smaller and cheaper, the range of application areas increases over time. In the farming industry machines and especially software has to fulfil certain criteria: it must be easy to handle, simple to learn and highly reliable. Otherwise, farmworkers can not be convinced to complete a purchase.

In the case of dairy farming, many companies are developing their sensor technology to keep track of various livestock data, for example, the temperature or the activity of a cow can be used as an indicator to predict different diseases. The HCI concept implemented in this thesis emerged in cooperation with smaXtec, which is a company working on sensor technology in dairy farming.

This work covers the basic workflow of the existing system, used in dairy farming, and features ideas to improve it. To be more precise, we discuss the design and implementation of a new, alternative HCI concept based on the disadvantages of the existing solution. The evaluation of the final prototype concerned its usability and usefulness in dairy farming. We used the System Usability Score as a measure to guarantee comparable results since the domain experts as well as people without reference to dairy farming tested and rated the prototype.

Acknowledgements

First of all, I would like to express my very heartfelt gratitude to my supervisor, Johanna Pirker. Thanks to her time with helpful feedback on my questions and concerns, she has contributed a great deal to making this work what it is today.

Furthermore, I would also like to express my great appreciations to my co-supervisor, Tobias Rauter, for keeping me motivated and providing technical as well as organizational experience during the whole thesis.

My gratitude also goes to my colleagues at smaXtec, who helped with brainstorming input at the beginning of the work and also served as domain expert testers at the end of the work.

Finally, I would also like to thank my girlfriend and my whole family, who have always supported me and motivated me during my Master's thesis.

Contents

1 Introduction

Dairy farming has a long history. About 9000 years ago, the domestication of cattle started. Proofs regarding the processing of milk fats trace back to the early Neolithic, which is approximately at the same time (Cattle Today Inc., 2017). Over time the human digestive system adjusted to decompose milk and milk products (Hollox, 2004).

In the early stages of farming, dairy products were essential to overcome harvest failures, which could have led to starvation otherwise (Curry, 2013). Due to the perishability of raw milk, it either had to be consumed as a fresh product, or processed into cheese, butter or yoghurt to increase the durability. Therefore, the farm sizes were small compared to nowadays because the milk had to be consumed as soon as possible and transportation over long distances was impracticable.

In the late 19th century the pasteurization process was invented. Pasteurization is the process of heating food for several seconds to kill certain bacteria resulting in an increased shelf life (Tewari & Juneja, 2007). This process was a key factor for the facilitation of the distribution of milk and thus the increasing demand in dairy farms. To keep up with demand, dairy farmers had to increase their farm sizes from a few animals to several hundred animals (Barkema et al., 2015). Due to that, the number of farms decreased over time, but the average number of animals per farm increased remarkably.

Even though the farm sizes are increasing over time, dairy farmers still have to keep an overview of their farm status. According to Morgan (2004), there are five areas where good agricultural practice should be applied to sustain profitability since they are producing food for a living:

1. animal health: the main income comes from selling milk, which has to be produced by healthy animals

2. milking hygiene: hygiene standards should be kept high for harvesting and storing milk
3. animal feeding and water: the quality of food and water have to be adjusted accordingly
4. animal welfare: in general, animal welfare is about the well-being of an animal, which is ensured by the five freedoms for animals[1]
5. environment: the local environment of the farm should not suffer through milk production

Dairy farming nowadays is a much more technologized environment than it was a few decades ago (BCDairy, 2019). Whereas the cows were milked by hand years ago, nowadays milking is done automatically and with the help of robots as well. These days it is also possible to monitor the health of the herd by using ear tags with RFID chips, pedometers, comparable with Fitbits[2] for cows, or other sensor technology. Even though modern technologies are entering the agricultural sector continuously, there are still a few difficulties which represent current issues. First of all, many solutions involve a smartphone application that includes the need for operations by hand. Moreover, the loud and dirty environment is posing new challenges to overcome as well as the smartphone requires a visual focus which is distracting from other work. The final result of the thesis should provide an alternative solution to deal with these problems.

1.1 Motivation and Goals

The motivation behind this work premises on the personal relationship to dairy farming, thanks to working for the company smaXtec animal care GmbH[3]. smaXtec produces bolus-like sensors, which measure various data such as activity, temperature as well as the pH-value, from the rumen of a dairy cow. This data is then processed, and the users can watch it directly over a web application or a mobile application. Since most of the users work

[1] https://www.animalhealthaustralia.com.au/news/feedlots-and-the-five-animal-welfare-freedoms/

[2] https://www.fitbit.com/home

[3] https://www.smaxtec.com/

with the system via mobile phones, the goal of the thesis is to come up with an alternative human-computer interaction (HCI) concept. This concept should be capable of dealing with the rough conditions in dairy farming, which we will ensure by analyzing its practicability.

1.2 Outline

Three main parts represent the basic structure of the thesis. The first part introduces the theoretical background of the work (Chapter 2). The second part addresses the problem description, the design (Chapter 3) and the practical application (Chapter 4). Finally, the third part focuses on an evaluation of the implemented system (Chapter 5), the lessons learned (Chapter 6) during research and development, future work (Chapter 7) regarding improvements of the implemented prototype as well as the final summary of the thesis (Chapter 8).

Chapter 2 gives insights into some dairy farming background, different potential devices for the implementation as well as related work based on the topic of the thesis. In the beginning, we provide background knowledge about dairy farming. Afterwards, we present various input methods and output devices, which came up during a brainstorming session. Finally, related work in smart farming and applications with similar circumstances conclude the chapter.

Chapter 3 firstly deals with the general workflow of the already existing system and pointing out issues in regards to using it. Thereupon, we identify the functional and non-functional requirements for the implementation. Based on them, we decided on the hardware and software components for the implementation.

Chapter 4 covers the implementation of the final prototype of an alternative HCI concept in smart dairy farming. Each component of the application is described in detail in this chapter. This consists of various Amazon services, the Alexa Skills Kit and an Amazon Web Service Lambda function, which is responsible for the backend logic and connection to the smaXtec API.

Additionally, we used the Alexa Voice Service to create an Alexa-enabled device.

Chapter 5 describes the whole evaluation process. This involves two questionnaires, one used to get the most relevant information about the testers, and the other focusing on the implementation. Furthermore, the specifications of the test setup, the testing procedure itself and finally, we present the analyzed results regarding the experience with the prototype.

Chapter 6 deals with the lessons learned during the research phase of appropriate hardware, the insights gained while developing the prototype as well as the gain of experience in the evaluation process.

Chapter 7 shows possible improvements to the implementation regarding the hardware and the software. These improvements are considerable if the implementation gets integrated into the running system. Finally, Chapter 8 sums up the thesis and provides a possible forecast of applying the concept for operating purposes.

2 Background and Related Work

The first section of is chapter provides an overview of the daily routines on a dairy farm and the usage of devices and technologies in the context of smart dairy farming. Furthermore, research on different approaches of input methods and output devices is covered. The final section provides related work to smart farming as well as other fields, which include the practical use of the input and output options.

2.1 Dairy Farming Background

There is no fixed or prewritten daily schedule when it comes to dairy farming. The determining factors regarding the tasks and duties on a dairy farm are the following: the size of the farm, the number of animals and workers on the farm, the use of technology, the daily weather conditions and more. "Day In the Life of a Dairy Farmer" (2015) and Williams (2018) served as an inspiration for the following fictional daily routine of a dairy farmer:

The farmer starts the day early in the morning, as cows are habitual animals and used to certain routines. Usually, a farmer milks the cows twice or thrice a day, and there has to be enough resting time between each milking process. Afterwards, the farmer has to feed the animals and also check the livestock's health condition regularly. Therefore the farmer examines the animals and adapts the food composition to their needs. Furthermore, the farmer has to be on standby for observing and assisting a cow when it is giving birth or when the veterinarian visits the farm. This daily routine is just briefly illustrating which tasks arise when being responsible for a dairy farm.

There are many additional tasks to handle besides the treatment of the animals, which leads to the increasing usage of modern technology in agriculture ("How new technology is transforming dairy farming," 2018). One of the technologies used in modern dairy farming are automatic milking devices such as a rotary milking parlour[1]. These devices allow the milking of multiple animals and simultaneously using little human resources. Milking robots[2] are even one step ahead since they do not require human interaction at all. Next, there are various kinds of systems which perform health measurement. There are either external sensors which get integrated into eartags[3], collars[4] or pedometers[5], whereby the last is comparable to a Fitbit for cows. The other option is the use of internal sensors, like the solution from smaXtec, which measures data from inside the rumen. Another widespread technology is the so-called cow brush[6], which starts rotating when the animal establishes contact and stops when it walks away. The brushes increase the blood circulation of the animals and improve their health and wellbeing (BCDairy, 2019). Furthermore, there is also facial recognition technology in trials, where the identification of individual cattle and monitoring of irregularities in behaviour are tested ("How new technology is transforming dairy farming," 2018).

Nowadays, smartphones are also widely spread in the agricultural sector. Therefore, modern technologies often provide their mobile application, or an existing system integrates smartphones, whereby the device is responsible for displaying the data to the customer (Laws, 2012).

One downside of this approach is that farmers have to focus on their mobile devices. They can not operate a machine or treat their animals at the same time because they are busy with the smartphone in their hands. The upcoming sections cover research on various input methods and output devices. This work focuses on alternatives that do not necessarily require

[1] https://www.gea.com/en/productgroups/milking-systems/milking-parlors/rotary-milking-systems/index.jsp

[2] https://www.lely.com/farming-insights/robotic-milking-concept/

[3] https://www.smartbow.com/en/home.aspx

[4] https://www.cowlar.com/store/product

[5] https://www.afimilk.com/cow-monitoring

[6] http://www.delavalcorporate.com/our-products-and-services/animal-welfare/delaval-swinging-cow-brush-scb/

navigation via typing or looking on a screen. The findings are listed and described in the following two sections in more detail.

2.2 Input Methods

As stated by Islam and Want (2014), smartphones, despite their young age, developed quickly to more than just a device that is used to make phone calls. Millions of people around the world use it to stay connected with other people and to stay informed about what is happening around the world. With the rise of smartphones, more people get familiar with this technology and therefore, the need to simplify the handling of mobile devices emerges. Although smartphones are already highly distributed all around the globe (Statista Research Department, 2016), operating a mobile device via a touch screen feels strange if one is used to mobile phones with physical buttons. In comparison, people had decades to get used to keyboard input with typewriters first and in combination with personal computers later on, before the first devices with touch screens were prevalent (Bellis, 2018)). The following sections will cover several input methods which were partly inspired by Hoober (2012) and also arose during a brainstorming session with five workmates from smaXtec. In addition to the methods, we discuss different pros and cons as well as use cases in the field of smart dairy farming.

(Virtual) Keyboard

The keyboard is a common input device for PC users to input letters and numbers into a PC. There are various layouts for the traditional keyboard, for example, the QWERTY layout which is widely used in the English speaking area, whereas the QWERTZ layout is common in Central Europe. The main difference between the layouts are the switched positions of the Y and the Z keys. Besides different layout types, there are also variations of designs, such as keyboards with or without a numeric keypad or function keys. The physical keyboard for PCs, displayed in Figure 2.1, serves as the template for the virtual keyboard on modern smartphones, Figure 2.2 visualizes an example. Compared to the PC keyboard, typing words with a virtual keyboard is usually slower due to the limited size of the keyboard and thus the smaller keys (Azenkot & Zhai, 2012) and (Kim, Aulck, Thamsuwan, Bartha, & Johnson, 2013). On the other hand, a physical keyboard is not feasible for prolonged use on a dairy farm, because it is too bulky to be carried around.

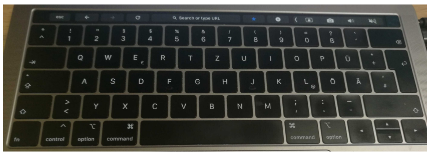

Figure 2.1: Physical MacBook Pro Keyboard

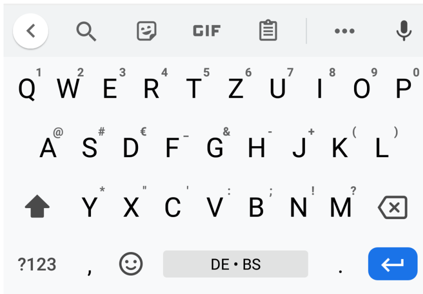

Figure 2.2: Screenshot of Virtual Keyboard on Huawei Honor 8

(Virtual) Keypad

Figure 2.3 shows an example for a keypad that can be found on a handheld calculator, on an ATM or as previously mentioned as a numeric keypad on a keyboard. Another application of keypads are feature phones, also called dumbphones, which were the predecessors of today's smartphones. Keypads, as seen in Figure 2.4, on such a mobile phone combined the input of both letters and numbers by allocating a set of letters to each digit key. The text input with a keypad differs from the input on a touch screen smartphone essentially, because repeated pressing of a single button is needed to write a certain letter. According to Silfverberg, Scott MacKenzie, and Korhonen (2000), keypad input is even slower than keyboard input, since the best keypad input method, including T9 predictions, achieved 46 words per minute whereas the best keyboard input reached 50 words per minute.

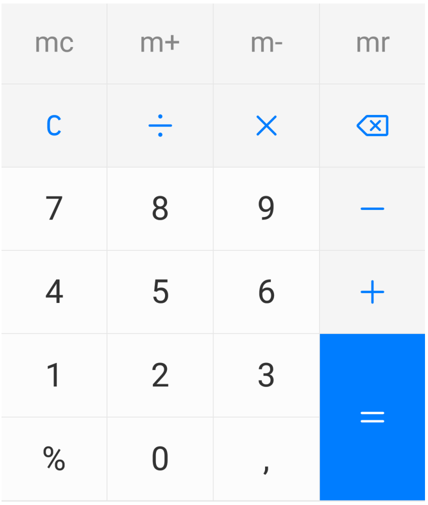

Figure 2.3: Screenshot of the Keypad of the Huawei Honor 8 Calculator

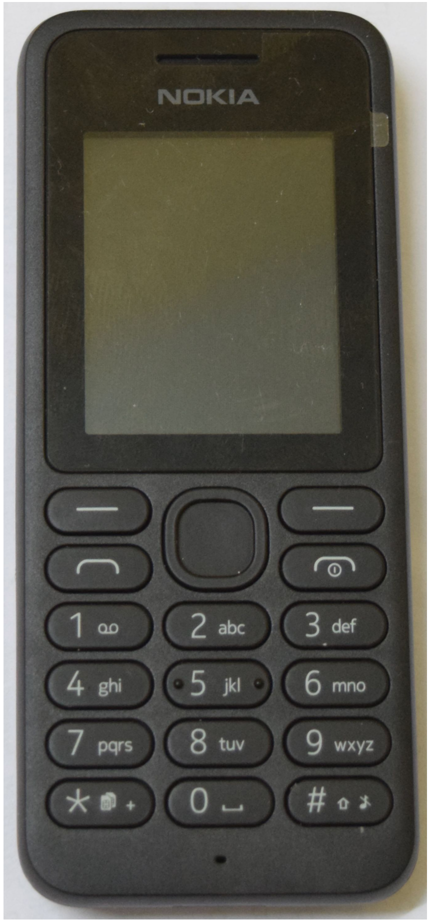

Figure 2.4: Mobile Phone Keypad (by Nikhilb239 (2018))

Constrained Data Entry

As the name partly suggests, constrained data entry is a limited version of a virtual keyboard. An example of this is a date or time picker, Figure 2.5 illustrates an example for a time picker. In that specific case, Android provides a virtual thumbwheel to pick a time, which is a simple entry option related to the input context. Constrained data entries are useful when the user is lead in the direction of providing needed data by preventing arbitrary input options at the same time (Hoober, 2012).

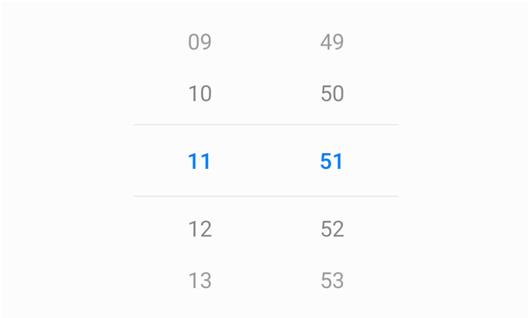

Figure 2.5: Screenshot of Timepicker on Huawei Honor 8

Stylus

A stylus, displayed in Figure 2.6, is a pen-shaped object used as an input device on touch screens such as smartphones, tablets or computers. The advantage of a stylus-based input is that it is comparable to writing with a pen on paper. Thus, it is only necessary to get used to the different feel when writing on a touch screen instead of a piece of paper. Lastly, input latency cannot be avoided completely, which also affects the user experience (Ng, Annett, Dietz, Gupta, & Bischof, 2014).

Figure 2.6: Various PDA Styluses (by Spurrier (2007))

Voice

Concerning the progress of virtual assistants, such as Amazon's Alexa[7], the Google Assistant[8] or Siri by Apple[9], to name a few of the most famous, using the human voice as an input method is increasing in popularity (Harlalka, 2013). Besides virtual assistants, smartphones also provide speech-to-text as well as text-to-speech features. Thereby, the user either dictates a text, which translates automatically into written text, or the other way around. The benefit of such a system is the hands-free input with the help of a microphone as well as up to three times faster throughput of words in comparison to written input (Ruan, Wobbrock, Liou, Ng, & Landay, 2016). On the downside, voice input technology still struggles with understanding different accents or performing well in noisy environments (Finch, 2014).

Gestures

Gestures enable another option of hands-free input, more precisely a device is not operated in the hands. To recognize human gestures additional

[7]https://alexa.amazon.com/
[8]https://assistant.google.com/
[9]https://www.apple.com/siri/

devices are needed, wired gloves for example, which transmit gestures to a computer, shown in Figure 2.7, or cameras, like Microsoft's Kinect[10], can be used for gesture-based input. Even though the Kinect is accurate and adding new gestures is easy as mentioned by Biswas and Basu (2011), there still exists a problem with gesture recognition. A fixed spot is required to install the camera, and therefore, it is not feasible in the context of dairy farming since the animals are moving around in the barn.

Figure 2.7: Wired Glove for Gesture Recognition (by NASA (2013)

Remote Data Entry / Device Gestures

Due to the predominant sales of smart TVs compared to regular TVs (Watkins, 2019), their main input device, the remote control, has improved as well. Its functions are not limited to switching between TV programs, it also has to be capable of entering text when browsing the internet or any other application. Another type of remote controller is the Wii Remote[11]. The control is done by pointing the controller onto the screen and then

[10] https://developer.microsoft.com/en-us/windows/kinect
[11] http://wii.com/

selecting the wanted input from a keyboard which seems to be an unusual input method (Hoober, 2012).

Face / Mimic Recognition

With the publication of the iPhone X in 2017 face recognition gained popularity as a new input method. As for the previous input methods, a key benefit of face recognition is the hands-free input characteristic. Furthermore, the accuracy increased over the past years by using 3D cameras and infrared cameras (Dao, 2018). On the other side, the data required for processing needs to be stored, which takes up a lot of storage space. Additionally, changes in the camera angle or facial attributes, like facial hair or sunglasses, can also interfere with the detection accuracy (Dao, 2018). Referring to Kalaiselvi and Nithya (2013), it is also possible to handle difficult light conditions that are present on a farm, very well. Kumar, Singh, Singh, Singh, and Tiwari (2017) found out that facial recognition of cattle performs very reliable with an identification accuracy of up to 95.87% by using an incremental support vector machine. Therefore face recognition could be used in dairy farming for identification or monitoring purposes ("How new technology is transforming dairy farming," 2018).

RFID / NFC

Radiofrequency identification (RFID) and near-field communication (NFC) are both wireless technologies that use radio signals for tagging and tracking (Thrasher, 2013). On the one hand, both technologies are working reliably in their fields. RFID, for example, is used in logistics where multiple items can be scanned simultaneously, without any line-of-sight limitation. The main application of NFC is direct payments via card or phone. On the other hand, both technologies have limited range, and an additional reader device is needed to be able to work with them. Both technologies could have a usage in farming to gather contextual information based on RFID or NFC tags either located in the barn or on the animals.

Image Processing

Images contain much information, such as colour, texture, objects in the image, used regularly in content-based image retrieval (Yue, Li, Liu, & Fu, 2011). Due to the information density of an image, a textual description would be much larger compared to the actual image. Humans can give a detailed image description by just having a glance at it, referring to Karpathy and Fei-Fei (2015). Image processing can be used for example to extract information content from images (Yue et al., 2011), or to enhance image quality by reducing noise (Hambal, Pei, & Ishabailu, 2017). Its weak points are the dependency on the image quality as well as the computational intensity is depending on factors such as the image size, expected quality and more. Image processing can be used to identify the animals on the farm or for constantly documenting the herd state inside a barn automatically.

Location

A user's location is also a valid input method. GPS serves as the major example because it is a famous example and widely spread. The basic concept of GPS is the communication between a receiver and at least four GPS satellites, which provide the longitude, latitude and time information (D. Kaplan & Hegarty, 2005). Other systems can use this data as input, for instance, one could track the position of their herd in the barn to check if all animals are in an expected location. The main problem is that GPS only works if there is a direct line-of-sight between the satellites and the receiver. Therefore indoor positioning is better measured with WiFi or Bluetooth (Namiot, 2015). GPS can be used to create context awareness to both, the herd manager as well as other devices, like milking robots.

Brain Computer Interface

Brain-computer interfaces (BCI) are a popular area of research for several decades already (Shurkhay, Alexandrova, Goryaynov, & Potapov, 2015). Even though it is in constant progress, state-of-the-art technology is not ready to be used by the public yet (Krusienski et al., 2011). In theory, a positive aspect of it is that there should be no limitations when technology reaches a state where everybody can control computers and machines with nothing but their mind. On the other hand, some devices, chips or sensors will be needed inside or attached to the human's body to make BCI possible.

Barcode / QR code

The classical barcode consists of parallel lines that vary in width and spacing, as shown in Figure 2.8. Since barcodes are one-dimensional and can only store very little information, QR codes, displayed in Figure 2.9, emerged in response to surpass those limitations. QR codes are two-dimensional barcodes holding information in both, horizontal and vertical direction. Both variations are used to compactly represent machine-readable data of the item they are attached to. Since various smartphone applications enable the user to scan QR codes and barcodes for free as well as the simple QR code creation online, a farmer could create own codes and place them in the barn to quickly switch between different contexts for instance. Quick soiling on a farm is an environmental enemy to QR codes, even though they can handle some information loss (Lotlikar, Kankapurkar, Parekar, & Mohite, 2013).

Figure 2.8: Barcode by (Public domain)

Figure 2.9: QR-Code by SimonWaldherr (2011)

Input Methods Overview

Below, in Table 2.1, is a tabular summary of the previously described input methods which arose during a brainstorming session. Each table entry consists of the input method, the emerged pros and cons, as well as an exemplary use case.

Table 2.1: Input Methods

Method	Pros	Cons	Use Case
(Virtual) Keyboard	Fast typing (physical) Established technology	Slow typing (virtual) Bulky (physical)	Used in combination with smartphones
(Virtual) Keypad	Old-style way of typing (known layout)	Old-style way of typing (slow)	Used in combination with feature phones
Constrained Data Entry	Display only needed input	Limited options Input depending on context Find out context	Prevent arbitrary input Lead into right direction

Continuation of Table 2.1			
Method	**Pros**	**Cons**	**Use Case**
Stylus	Familiarity to usage of pen	Different feeling than on paper Input latency	
Voice	Hands free input Faster input	Lack of precision in loud environments Hard to interpret dialect correctly	Keep focus on work No need to handle device
Gestures	Hands free input Extendable	Fixed camera position Need to learn commands	Gestures similar to real work
Remote Data Entry Device Gestures	Easy to learn	Additional device needed	
Face / Mimic Recognition	Hands free input Currently up and coming	Difficult light conditions on the farm	Automatic identification of animals
RFID/NFC	Reliable	Range Additional reader	Context switch in barn Identification of animals
Image processing	More information than text	Image quality Training & computing power	Documentation of special conditions Identification of animals
GPS	Precise location	Hard to track inside barn WIFI or Bluetooth for inside tracking Power consumption	Create context awareness
Brain computer interface (BCI)	No limitations for I/O	Devices or sensors inside body	Context switches depending on location

Continuation of Table 2.1			
Method	**Pros**	**Cons**	**Use Case**
Barcode / QR	Simpler way of image processing Compact information Readable by smartphone	Dirty environment	Context switches through QR codes in barn

2.3 Output Devices

After getting an overview of different input methods, this section deals with output devices used for displaying the input data as well as any other processed data. The devices were also part of the same brainstorming session mentioned previously, thus identified in discussion with five co-workers.

Smartphone

Due to the increasing distribution of smartphones in the agricultural field (Chhachhar & Hassan, 2013), new ways of communication and workflows have arisen (Vate-U-Lan, Quigley, & Masouras, 2017). The big advantage of smartphones is that those powerful mini-computers fit into a pocket and are easy to transport. Having constant access to a weather application is essential for a farmer to plan the workday (Vining, 1990). Additionally, manufacturers of farming devices often provide mobile applications that are either used for monitoring or operating the devices (Laws, 2012). The small screen size of a smartphone is disadvantageous because the rough circumstances on a farm increase the need for a clear presentation of data as well as an easy application handling. Nevertheless, smartphones are good for operational work on a dairy farm, especially when the farmer wants to take quick notes while being in front of an animal.

PC / Laptop Monitor

PC or laptop monitors are another widespread output device. In both cases, the stationary PC monitor as well as the mobile laptop monitor, the bigger screen size compared to smartphones is a real plus. On the contrary, only the laptop monitor is easy to transport, but still, it is too bulky to be used effectively under the harsh conditions on a farm. In general, the bigger screen size opens up new possibilities regarding management tasks on a farm, like getting an overview of key performance indicators or checking graphs and diagrams in more detail.

TV

A PC or laptop monitor may not meet the user criteria regarding the display size. In such a case, a TV screen should provide relief to the user. The bigger screen size makes limitations in mobility unavoidable. The user can influence the TV's lifespan by determining a fixed position for the device, which reduces the environmental impacts on the device to a minimum (Silva, 2019). Therefore it is more convenient to be used as a presentation device in the barn or in a separate showroom rather than functioning as an everyday tool to work with.

Projector

If a TV still cannot satisfy the users' needs of an output device that can display data on a large screen, a projector would be the next device to be considered. Due to low lighting conditions on a farm, one cannot install the cheapest alternative, and therefore, the costs of a fitting projector which can deal with those circumstances are high. Furthermore, a projector requires enough space to ensure a good projection quality, so an installation inside the barn might prove difficult.

Speakers / Headphones

Speakers and headphones are devices which are used for sound output by converting electrical signals into sound waves (Talbot-Smith, 2012). Compared to the previously described devices, one upside of sound output is that it does not rely on any screen size, so the size of information is not limited and can be processed instantly. Additionally, it is not needed to read data from a screen nor handling with a device in hands, whereby other work can be done manually in parallel. On the downside, loud and noisy environments can make it difficult to concentrate and understand the sound output.

Tablets

Tablets are combining the advantages and disadvantages of a smartphone and a laptop (Dent, 2018). One positive aspect of a tablet is the higher portability than a laptop due to their low weight, and at the same time, they have a bigger screen size than smartphones. Furthermore, the average price is also remarkably lower than for a laptop, and the start-up time is usually shorter. On the contrary, the processor speed, and therefore, the overall performance of a tablet cannot compete with a laptop. However, when it comes to practicality, a tablet cannot be pocketed quickly like a smartphone.

Head Mounted Displays

Referring to Shibata (2002) head-mounted displays (HMD) are devices that are worn on the head with either one continuous screen for both eyes simultaneously as seen in Figure 2.10 or two separate screens, one for each eye which is displayed in Figure 2.11. HMDs can be used for simulations in medicine (Birkfellner et al., 2002), virtual reality or personal theater (Shibata, 2002). Since the display is directly in front of the eyes users do not get distracted by their surroundings, which is good if focussing on the information is desired but in the context of farming, this is handicapping the farmer. In farming the user wants to keep the overview of the surrounding, which cannot be guaranteed while operating an HMD.

Figure 2.10: HMD with one screen (by Brycearm (2012a)

Figure 2.11: HMD with two screens (by Brycearm (2012b)

BCI

In the previous section 2.2, we gave a short overview of BCI as an input method. It also can be considered as an output device (Rao, 2019). According to the Gartner Hype Cycle for Emerging Technologies from 2018 (Panetta, 2018), BCI is still considered in a development stage since it will not reach its plateau within the next 10 years. In regards to this, no accurate predictions neither regarding the input nor the output are possible at this moment. Therefore, the bottom line is that in theory "everything is possible" using BCI until further research sets the limitations.

Wearables

Wearables, also called wearable computers, are small devices that users usually wear on the body. Nowadays well-known wearables are smartwatches, like the Samsung Galaxy Watch[12] or the Apple Watch[13], and fitness

[12]https://www.samsung.com/global/galaxy/galaxy-watch/
[13]https://www.apple.com/apple-watch-series-4/

trackers from Fitbit or Garmin. Within the same category are devices like the GoPro[14], an action camera, or the Google Glasses[15], a head-mounted display. Wearables allow the user to gather more personal data with the integrated sensors which open up new possibilities, for example, an individually adapted diet and sports plan or early detection of diseases and thus early treatment. The user can get access to needed information constantly. One negative aspect of wearables is that the small display sizes are limiting the presentation of information. The big downside is the usually short battery life of wearables as well as depending on a smartphone ("Pros and Cons of Wearable Technologies," 2017).

Printer

A relatively old (Seiko Epson Corporation, 2019), but well-established technology and also widely used output device are printers. Printers usually take input from a computer, whether text or graphics and bring the information onto paper ("Merriam-Webster.com," 2019). From personal experience gathered during working at smaXtec, printers are widespread in the agricultural sector. For example, printing daily ToDo lists and distributing them to the appropriate worker. It is convenient for the user to make notes on a piece of paper rather than on a computer or similar (Poretsky, 2012). Then again, each printed page costs money, unlike electronic documents. Moreover, the user interfaces of printers are still capable of improvement to make them easier to operate.

Lights

The biggest advantage of using lights as an output device is at the same time a disadvantage. With the help of lights, one can visualize a status by either turning a light on or off. A single light can only display binary information. At the same time, the information value is clear. To represent more

[14]https://de.shop.gopro.com/EMEA/cameras/
[15]https://www.google.com/glass/start/

informative data, multiple lights must be combined, taking into account that this leads to giving up on simplicity.

Haptics

According to Robles-De-La-Torre (2006), haptic is about perceiving information by active exploration, typically with the hands. A device can provide haptic output by providing force or vibrations to the user. The device does not require a screen to process the information, rather than that direct contact to the skin, the largest organ of a human is needed. Different information can be output by varying the intensity and duration of the haptic output. Therefore it is very ineffective since every human has different skin sensitivity and light outputs may not be noticed at all by some people.

Output Devices Overview

The devices outlined above are summarized below in Table 2.2, also including pros and cons and use cases like in the previous table.

Table 2.2: Output Devices

Device	Pros	Cons	Use Case
Smartphone	Mini computer in the pocket Comfortable for operative work	Small screen size	Take notes quickly while working
PC/laptop monitor	Medium screen size Useful for management (comparing diagrams etc.)	Bulky device to carry around in the barn / on the field	Get overview of KPIs Check graphs and diagrams

Continuation of Table 2.2			
Device	**Pros**	**Cons**	**Use Case**
TV	Big screen size	Limitations in mobility Needs fix position in barn Rough & dirty environment	Presentation device in barn or external showroom
Projector	Even bigger screen size	Projection space Needs good light conditions Costly Installation in barn	Presentation device
Speakers / Headphones	Voice output not depending on screen size No manual device handling	Loud / Noisy environment	Processing information directly
Tablets	Higher portability Bigger screen than smartphone Low-cost	Processor speed / performance Bulky to pocket	
Head Mounted Displays	Embedded information display No distractions	Additional devices needed Technical know-how needed	Display critical information from close-up
BCI	"everything is possible"	Not usable yet (research)	
Wearables	Additional sensors Direct access (on body)	Really small screen Battery life / maintenance Smartphone dependency	Gathering of most relevant information quickly

Continuation of Table 2.2			
Device	**Pros**	**Cons**	**Use Case**
Printer	Established technology Already widely used	Costs UI improvable	Printing lists for everyday tasks
Lights	Simple status display	Simple status display	Different LED light depending on status of a cow
Haptic feedback	No need to look at screen	Limited options (vary intensity or duration)	Vibration Electric stimulation

2.4 Preliminary Work

This section covers related work of digital tools in the context of smart farming, mainly focussing on dairy farming, bearing in mind the already presented input methods and output devices in the previous two sections. Furthermore, related applications, which are in other fields than agriculture but still have similar circumstances to farming in certain respects, will be included in this section as well.

Smart Farming

As discussed by Chhachhar and Hassan (2013), smartphones are an essential tool with increasing usage by farmers. Farmers use smartphones to acquire weather information fast because the weather conditions determine the planning of a farmers day. Another important function is the opportunity to contact market brokers where they sell their products or gather information about the current market situation.

Vate-U-Lan et al. (2017) published a case study that covers the Internet of Things (IoT) in agriculture in the context of smart dairy farming on a farm in Ontario. One case presented a sensor for cows, particularly a pedometer, which is used for identification as well as for measuring the amount of produced milk, the cows' activity and further data. Concluding, all cows with such a pedometer can be monitored by additional software and furthermore, the profitability on the farm can be improved. Jayaraman, Yavari, Georgakopoulos, Morshed, and Zaslavsky (2016) highlighted another IoT application in smart farming. They presented the SmartFarmNet, which is a platform for scalable sensor data acquisition, analysis as well as visualisation. Besides presenting the design of the SmartFarmNet's architecture, they emphasized its novel statistical analysis approach, which validates user queries almost in real-time, even if they consist of high-velocity data streams. Jayaraman et al. (2016) also evaluated the scalability of the platform by using real farming data.

Next, Vate-U-Lan et al. (2017) list a sensing technology that is connected to a computer or mobile phone and alerts the farmer when the cow is about to give birth to a calf as well. This is, in fact, crucial because so-called dystocia, distress during the birthing phase, can be fatal for both the cow and the calf and in further consequence the farmer's profitability as well. Fadul et al. (2017) covered a similar topic of predicting the calving time with another technology, and they discovered that restless behaviour emphasizes the last two hours of calving. Additionally, Vate-U-Lan et al. (2017) touches the topic of GPS-driven crop harvesting equipment, since dairy farms often grow their feed to reduce operating costs. Using the software, in combination with GPS, opens the possibility of mapping their fields and process data regarding cost-effective crop management. Neményi, Mesterházi, Pecze, and Stépán (2003) illustrate the use of GPS in precision farming in more detail, for instance, covering the field mapping as described before but including the visualization of data on a real map as well.

Driessen and Heutinck (2015) consider the ethical aspects of the wider distribution of milking robots and automated milking systems on dairy farms, especially in the Netherlands. They compare the issues between the traditional milking machine, which has to be operated by a worker, and the automated milking systems and milking robots, which do not require human interaction at all. In conclusion, both approaches have their right to exist, regarding the ethical view, and in the end, it depends on the animal's needs, if human assistance is necessary or not while milking. Besides milking robots, there are also many other fields which develop agricultural robots.

Shamshiri et al. (2018) present various agricultural robots, which can be used in the fields of harvesting, weed control and targeted spraying, as well as for field scouting and data collecting. The main reason that such robots are not as widespread as milking robots is that in the current state of development, those robots cannot compete with humans regarding speed and efficiency.

Another aspect of smart farming is plant growing. Therefore, it is important to start monitoring plants from their growth stage until the time of harvesting. Jhuria, Kumar, and Borse (2013) demonstrates algorithms for detecting diseases, evaluates the spread of diseases as well as grades mangos based on their size and calculated weight. The image processing algorithms were developed with the help of neural networks in MATLAB and have shown to work well in the experiments of the paper. Chetan Dwarkani, Ganesh Ram, Jagannathan, and Priyatharshini (2015) discusses the general automation of duties in the agricultural field. Since a high percentage of India's population is employed in agriculture, and the additional lack of automation, most work is handled manually. Many of those manual steps can be automated nowadays. The authors of the paper presented a smart sensing system in combination with an irrigator system, which is used to automatically dispenses the right nutrients based on the needs of the crops. With IoT being a rising factor in the field of smart farming, the number of sensors in farming is increasing drastically, and therefore much data is collected as well. Wolfert, Ge, Verdouw, and Bogaardt (2017) cover this topic of Big Data in smart farming which also includes the already mentioned increase of data. Since different devices and objects are connected wirelessly, the scope of Big Data does not stop at farming alone; on the contrary, it goes far beyond and includes the supply chain as well. Due to the enormous amounts of data, the decision-making advanced to a higher level than ever before.

Related Applications

Big Data is not exclusively relevant to smart farming. Due to the increasing amount of smartphones as well as wearable devices, more and more users manage and track their health themselves (Dimitrov, 2016). The devices alert the user about changes in blood pressure or upcoming appointments. With the help of IoT, one can also be monitored and advised remotely, which enables real-time communication with doctors. Furthermore, doctors can enter the patient's data electronically, which would ease the access and the readability compared to paper records. According to Birkfellner et al. (2002), computer-aided surgery is one of the most promising fields to apply augmented reality. They deal with the problems of AR in a surgical environment regarding the needed common focal plane between the real world and the computer-generated image, which uses an HMD as a see-through device. By manipulating a commercial HMD for AR visualization, they established a system where the error between the visual position and the position displayed in the HMD did not exceed 2 mm. Smartphones are also beneficial devices in the medical context states Baumgart (2011). At that time, 64% of US physicians owned a smartphone which increased over the following years to over 80% (Statista Research Department, 2015). Smartphones enable the professionals to acquire the most up to date information within seconds which the specialist can provide to the patients. There are also various applications, for example, enabling to access and edit electronic medical records, and additionally, monitoring patients via a smartphone. Further research applications exist as well, like attaching a "laboratory on chip" to perform sample preparation automatically. Becker et al. (2009) illustrated another assistive system which they called the SmartDrawer. They used RFID tags to track medication usage and also remind users to take their prescriptions. They came up with a prototype of the SmartDrawer that can be further expanded, for instance, having sensors which update the drawer state or even using a scale to make sure that the patient took a certain medication.

Fuller, Ding, and Sattineni (2003) describe the usage of wearable computers in context with the construction industry. They compared the performance in a punch-list experiment using three different methods, by hand, by Palm PC and by wearable computer. Furthermore, they analyzed the pros and cons of each approach. Even though they came to the decision that wearable computers are more applicable in construction than a Palm PC, they emphasized on many negative aspects of wearable computers back then. Additionally, they stressed that one should not ignore the future development of Palm PCs. As stated by Nevogt (2018), digital tools, more specific virtual reality, are rising in the construction industry. VR enables users to view construction sites in 3D without the need of being on-site. Furthermore, simulating training situations can be used to save money as well as ensure the safety of employees. Additionally, virtual meetings with clients can help to ease communication. Both parties can access a virtual realization of the specifications rather than being in need to read a confusing textual description. Based on Kondratova et al. (2004), smartphones have two major disadvantages for construction field users. First, the small screen size and subsequently also the small keyboard are challenging tasks regarding usability. Therefore, voice control can provide relief since it has no graphical user interface which requires focus. Moreover, giving audio input is quicker than typing in information via keyboard. Kondratova et al. (2004) further suggest investing in performance and usability studies in the field.

One more way to input data also stated in the previous sections, is to use the human's voice. Lv, Zhang, and Li (2008) show one application where they used their spoken voice commands to control a robot. They used simple commands like, for instance, the commands "go forward" or "go backward". Even in a noisy environment, the results of the command recognition were over 70%, which proved to be efficient enough for real-time operation. Smart home devices are another quite popular use case for voice control (Mittal et al., 2015). The concept of a smart home includes automation of various appliances and gadgets like lighting, air-conditioners, computers, audio systems, and many more. Usually, the users can control their smart home remotely, since the devices establish a connection to the internet over Wi-Fi or similar technologies. The bottom line of the implemented system by Mittal et al. (2015) is that the voice recognition module requires placement at a common location. Furthermore, they suggest a training process for every home resident as well as further testing under difficult conditions.

2.5 Summary

This chapter described an exemplary overview of a farmer's day on a dairy farm, including the early wake-up of a farmer and the habitual characteristic of cows. To create a basic understanding a few commonly used modern technologies, like rotary milking parlours or sensors used to track the cows' behaviour, were serving as an example for smart farming additionally to the basic daily routine. The commonalities of those technologies are that they either require or provide an application on a smartphone, which is used to control or monitor the devices. This reliance on a smartphone brings up new challenges since the smartphone requires the user to focus on it as well as operate it manually. Therefore, various input methods and output devices were listed and described in further detail to overcome those challenges. For example, (virtual) keyboards picture an input method that is known by PC users as well as smartphone users, whereas a projector served as an example for an output device. Additionally, advantages like voice input being a hands-free technology, and disadvantages such as the bulkiness of a physical keyboard, were described for each method and device. An input table and an output table summarized the findings, including the pros, cons, and use cases in the context of smart farming. The last section of this chapter illustrated the preliminary work of digital tools in the specific context of smart dairy farming, like using smartphones to acquire weather information quickly. Moreover, a variety of other practically implemented and used digital tools that were mentioned in the sections before were further described, such as using wearable computers as well as augmented reality in the construction industry.

3 Problem and Application Design

The major goal of this thesis is to implement and evaluate an alternative HCI concept which should make life easier for workers in smart dairy farming. Achieving this objective involves working out the issues of the current workflow with the smaXtec system. Based on those findings, the next step is to identify the functional and non-functional requirements for future improvements. Combining these requirements with the input methods and output devices from Chapter 2, narrow down the options for feasible devices that come into consideration for the implementation. The following sections describe these steps in more detail, ending in the final decision on which device to implement the new HCI concept.

3.1 Current Workflow with smaXtec

Customers of smaXtec have two options to work with the system. The first way is the web application, which is accessible by visiting the web site messenger.smaxtec.com. On the first visit, they get to the login screen. There they can directly login if they already have an account. Otherwise, they have to register if they are new to the system. After logging in, the user gets redirected to the dashboard, which is the core of the web application, illustrated in Figure 3.1. At this screen, the user gets an overview of the current overall state on the farm and can check further data of specific animals.

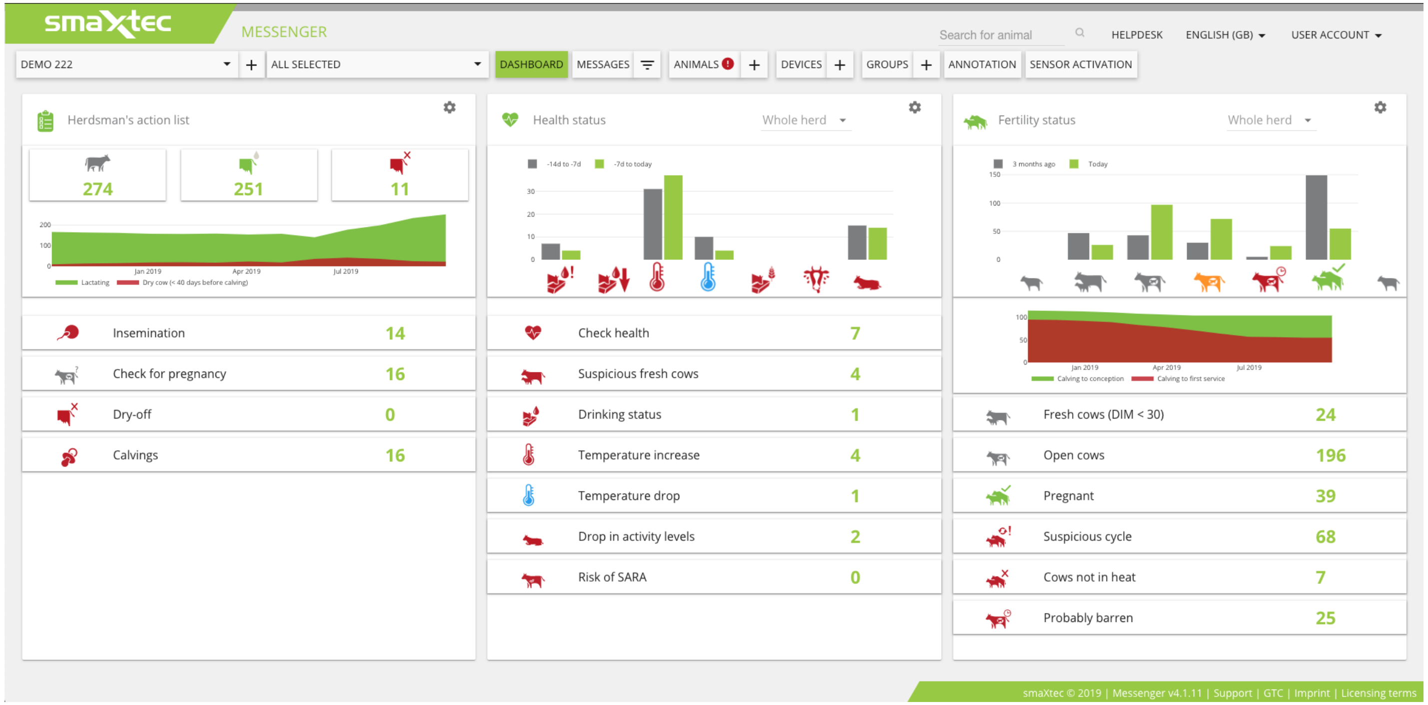

Figure 3.1: Screenshot of the Dashboard of the smaXtec Web Application

The second approach is to use the smaXtec mobile application, which is available for Android and iOS in their respective app stores. Again, the first entry leads to the login screen, shown in Figure 3.2. After the login, the user gets forwarded to the tasks dashboard, illustrated in Figure 3.3. The most recent tasks of the last days are listed there, which is one of the most significant parts of the app.

Figure 3.2: Screenshot of the Login Screen of the smaXtec Mobile Application

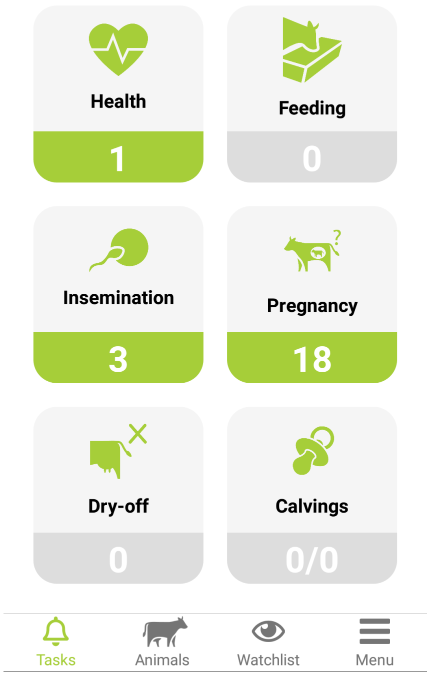

Figure 3.3: Screenshot of the Tasks Dashboard of the smaXtec Mobile Application

Since the majority of the smaXtec customers use their smartphones to operate the system, the analysis will focus on the workflow of the smartphone. The whole system is controllable by touch input. Therefore, the device screen is essential for selecting options, providing data input as well as navigating through the application. While doing so, the farmer needs at least one hand for operating the smartphone. Additionally, the focus is on the smartphone rather than on the animals. The problem with this is that the farmer needs his hands to either check the animals or for a different type of work like feeding.

There are two primary use cases when working with the existing system. One is when the farmer manually views the application to check the overall state on the farm or some specific animal. A push notification or email notification is usually the starting point of the second use case. In this scenario, the user is alarmed that irregularities have occurred on the farm and that actions should be carried out. 3.4 visualizes both use cases.

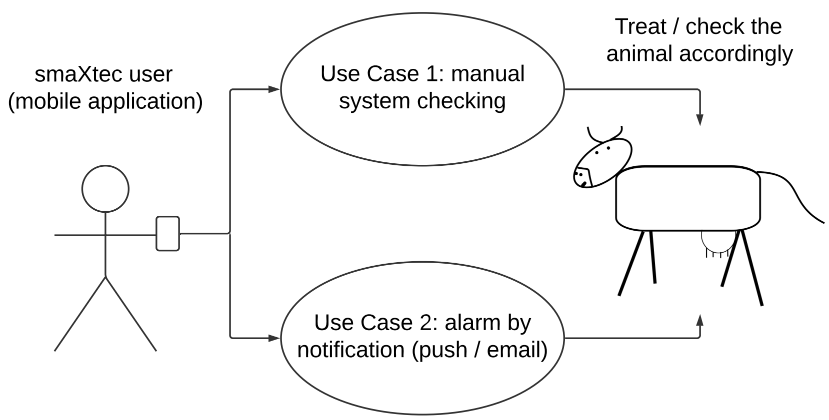

Figure 3.4: Use Case Diagram of the smaXtec system

3.2 Requirements Analysis

We will achieve the goals of this thesis by tackling the problems mentioned in the previous section, operating the system by hand and additionally requiring a visual focus on a smartphone screen, and developing a suitable solution. For this reason, before diving directly into the practical part, requirements have to be identified. The following two sections will cover the non-functional as well as the functional requirements, which determine the hardware as well as the software which we will use for the practical implementation.

Non-Functional Requirements

The key stakeholders of the mobile application are the farmers who use the smaXtec system regularly. Because they are already familiar with the existing system, the implementation should be as simply usable as the current solution. According to the first issue mentioned earlier, operating the application by using at least one hand, another requirement is to work with the system hands-free. The second issue, requiring a visual focus on a screen, will be considered as well so that the solution does not distract the user while working on the farm. Additionally, the application is supposed to deal with difficult circumstances, like a loud and dirty environment. Lastly, we will apply a feasible technology for the implementation, and additionally, develop a testable system within the time frame of this master thesis. In summary, the most important non-functional requirements for practical implementation are the following:

- Easy handling (EH)
- Hands-free operation (HF)
- No need for visual focus on a screen (VF)
- Capable of rough environmental influences (EI)
- Feasible technology (FT)

Taking account of the input methods and output devices discussed in the previous Chapter 2 eased to approach the decision-making for the hardware. Considering the requirements, only a subset of the input methods and output devices remains suitable for the application. Table 3.1 lists the subset of input methods and Table 3.2 shows the filtered output devices.

First of all, the input methods, which are not hands-free, like a virtual keyboard or stylus, disqualified for further consideration. Next, technologies that are still in the development phase, such as BCI, were sorted out. Lastly, output devices that distract the user's focus as TVs, Projectors or Smartphones were removed from the filtered lists as well.

Table 3.1: Subset of Input Methods from Chapter 2

Method	Pros	Cons	Use Case	EH	HF	VF	EI	FT
Voice	Hands free input Faster input	Lack of precision in loud environments Hard to interpret dialect correctly	Keep focus on work No need to handle device	X	X	X	X	X
Gestures	Hands free input Extendable	Fixed camera position Need to learn commands	Gestures similar to real work	X	X	X	X	
Remote Data Entry	Easy to learn	Additional device needed		X		X	X	X
Face / Mimic Recognition	Hands free input Currently up and coming	Difficult light conditions on the farm	Automatic identification of animals	X	X			X
Image processing	More information than text	Image quality Training & computing power	Documentation of special conditions Identification of animals				X	X
Barcode / QR	Simpler way of image processing Compact information Readable by smartphone	Dirty environment	Context switches through QR codes in barn	X			X	X

Table 3.2: Subset of Output Devices from Chapter 2

Device	Pros	Cons	Use Case	EH	HF	VF	EI	FT
Speakers / Headphones	Voice output not depending on screen size No manual device handling	Loud / Noisy environment	Processing information directly	X	X	X	X	X
Head Mounted Displays	Embedded information display No distractions	Additional devices needed Technical know-how needed	Medicine / Virtual Reality		X		X	X
Wearables	Additional sensors Direct access (on body)	Really small screen Battery life / maintenance Smartphone dependency	Gathering of most relevant information quickly	X			X	X
Lights	Simple status display	Simple status display	Different LED light depending on status of a cow	X	X		X	
Haptic feedback	No need to look at screen	Limited options (vary intensity or duration)	Vibration Electric stimulation	X	X	X	X	

Functional Requirements

Fulton and Vandermolen (2017) describe a functional requirement as the behaviour between inputs and outputs. Unlike a non-functional requirement, which describes the characteristics a system shall achieve, a functional requirement describes what a system must do to achieve certain results (Clarkson & Eckert, 2010).

The following list describes the expected functionality of the practical implementation, which is covered in the upcoming Chapter 4 in more detail:

1. The user configures his smaXtec credentials once.
2. The user activates the system by using a simple activation catchword.
3. The user can execute a set of predefined actions which are already used in the existing system:
    - Add a heat to an animal
    - Add an insemination to an animal
    - Add a pregnancy examination to an animal
    - Add a dry off[1] to an animal
    - Add a calving to an animal
    - Add an abort to an animal
4. The user can reference to an animal by its name.
5. The user can reference to an animal by an unique ear tag number.
6. The system provides feedback and interaction with the user, covering three scenarios:
    - Providing a confirmation message to the user when the command was completed successfully.
    - Asking the user for additional information or agreement if the command was incomplete or ambiguous.
    - Asking to repeat the command if it could not be processed.
7. After a command is fully processed the information is transmitted to the API of smaXtec.

[1]https://www.lely.com/farming-insights/drying-dairy-cow/

3.3 Decision-making for Hardware

After sorting out the input and output options which did not meet the requirements, the search concentrated on the leftover possibilities. Technical feasibility was ensured by picking up-to-date technologies that are already in a commercial stage. Focusing on the input and output options which cover most of the non-functional requirements helped to make a step towards the hardware decision. Since voice input covered all of the requirements, we chose it as the desired method for realizing the practical implementation. As the distribution of wearables is increasing according to Lamkin (2018), we decided to use them for the final application, for example, a smartwatch. The primary focus is on wearables which also provide access to speakers or headphones, which were the best fitting option from all output devices. Besides, it also goes well with the chosen voice input method. Table 3.1 and Table 3.2 portray the selected options.

Due to the wide variety of smartwatches further investigation considering smartwatch features which meet the desired requirements had to be carried out. This resulted in a closer selection including the Samsung Gear 2[2] and the Apple Watch Series 0[3], which was the first version of the Apple Watch released in 2015. The Samsung Gear runs on the Tizen operating system, which is a Linux-based open-source operating system (Saxena & Kwon, 2012), whereas Apple is known for its closed ecosystem (Lin & Ye, 2009).

[2]https://www.samsung.com/uk/wearables/gear-2-r380/SM-R3800VSABTU/
[3]https://support.apple.com/kb/SP735?viewlocale=en_US

Samsung Gear 2

Due to the open-source characteristics of its OS, the initial decision was made to use the Samsung Gear 2 for the implementation. The smartwatch offers an integrated microphone as well as a camera. Unfortunately, the first trials with the device already pointed out difficulties. The Samsung Gear 2 is only able to connect with specific Samsung Galaxy smartphones and tablets easily, referring to (bhphotovideo.com, 2019). Unluckily, none of these devices was accessible at that time, and thus a workaround was needed. According to online research, there is a way to deal with this issue: Android devices that are rooted can connect and work with the Samsung Gear 2. Having a Motorola G4 available, the first step was to root the device. After various issues with the rooting process on the Motorola Moto G4 emerged, the device could be rooted successfully, whereafter the next obstacle occurred. Again, the connection between the smartphone and the smartwatch could not be established even though following the troubleshooting instructions on an online forum (stuntdouble, 2014). After several failing attempts to bypass the problem, the Samsung Gear 2 disqualified for the approach. The following Figure 3.5 shows how the Samsung Gear 2 looks like.

Figure 3.5: Samsung Gear 2(by Dambrāns (2017))

DZ09

The next step, before switching to the Apple ecosystem, was to look for an alternative smartwatch that is not locking the user into its system like the Samsung Gear 2. The search led to the DZ09, a rather cheap smartwatch that establishes a connection to any smartphone via Bluetooth, according to (smartwatchspex.com, 2019). Despite the low price, the watch offers a built-in camera as well as a microphone. Two different versions of the DZ09 were ordered online for testing purposes, whereas one looked like the watch displayed in Figure 3.6. The connection tests succeeded immediately on the first try. Unfortunately, further trials and research revealed the major weak point of the DZ09. The initial plan of developing software for the smartwatch to provide a new HCI concept to dairy farmers failed because of missing possibilities to code and install applications on the smartwatch easily within the given time frame of the master's thesis.

Figure 3.6: One version of the DZ09 (by Tomanek (2018))

Apple Watch

By now the point was reached where all the attempts with easily accessible, open-source smartwatches failed. Therefore, the decision was taken to give the Apple Watch a try. Due to the lock-in of the Apple system, further Apple devices are required to develop software on the smartwatch. Luckily, smaXtec already used Apple devices, more precisely an iPhone and a MacBook Pro, for their mobile application development. Therefore we have overcome the first obstacles without more ado. Afterwards, familiarization with the Apple Watch started by using online tutorials for Swift development, which resulted in the first smartwatch application. In further research, we tested different approaches regarding libraries and the general functioning of the Apple Watch and evaluated them to obtain the best results which comply with the requirements. The incorporation with the Apple Watch looked promising, especially because Apple provided SiriKit and Shortcuts in their latest iOS versions. Both are used to ease the communication between self-developed applications and Siri, Apple's virtual assistant. Apple introduced SiriKit 2016 together with iOS 10, and it defines so-called intents, which are a certain type of requests the user can make (Apple Inc., 2019a). The shortcoming of SiriKit is the limitation to seven predefined request types. However, Shortcuts (Apple Inc., 2019b), which came out with iOS 12 in 2018, can be used to define custom task sequences that can be triggered by user-defined commands which are recognized by Siri. First experiments with Shortcuts worked well on the tested iPhone, but testing the same Shortcuts on the Apple Watch revealed a huge drawback: The Apple Watch Series 0, shown in Figure 3.7, does not support Siri's voice feedback feature thus this option dropped out as well.

Alexa-based System

A decision for the final system components had to be taken to not prolong the implementation further. Since the voice feedback feature with Siri was enabled for the Apple Watch Series 3, one option was to acquire this model for approximately 300 €. Further research yielded another option by using Amazon's virtual assistant Alexa and developing a Custom Skill that fulfils

Figure 3.7: Apple Watch Series 0 (by Hook (2015))

the set requirements. Looking ahead, a realization with an Alexa-based system would be more affordable than a solution which requires at least a certain version of an Apple Watch and additional Apple devices. Furthermore, we cannot ensure that the voice feedback feature with Siri on the Apple Watch will be available on future versions as well, whereby the whole system would become unusable. One option would be to use an Echo Dot, illustrated in Figure 3.8, as the hardware device for an Alexa Skill. Since the Alexa API is open source, it is not limited to Amazon devices. There are libraries for other devices such as the Raspberry Pi Zero W, shown in Figure 3.9, which allow communication with the Alexa service. This is considered in the following section as well to finalize the decision regarding the hardware choice for the implementation.

Figure 3.8: Amazon's Echo Dot (by Varnum (2018))

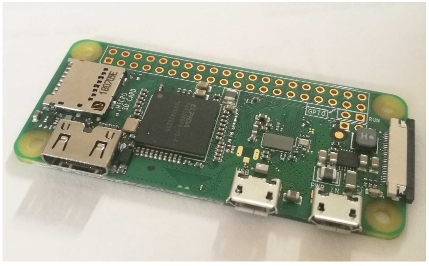

Figure 3.9: Raspberry Pi Zero W

Final Decision

Based on the findings described in the previous sections, each device has its advantages and disadvantages. A summarizing overview of the previously described devices, including their pros and cons and further information, is featured in the following Table 3.3.

Table 3.3: Overview of Tested Devices

Device	OS	Released	Price	Pros	Cons
Samsung Gear 2	Tizen	April 2014	ca. 250 €	Camera Microphone	Device limitation Expensive
DZ09	Customized Android	2015	10-20 €	Cheap Camera Microphone iOS + Android	Not programmable
Apple Watch (Series 0)	WatchOS	April 2015	ca. 300 €	Microphone Bright display	Expensive iOS only iPhone needed No voice output
Echo Dot (3rd gen.)	Fire OS	Sept. 2018	60 €	Microphone Speakers Alexa based	Not a wearable
Raspberry Pi Zero W	Raspbian	Febr. 2017	10 €	Open source Existing libraries Adaptable	Not a wearable

Amazon's Alexa will serve as the software foundation for the implementation by developing a Custom Skill. In terms of hardware, we will transform a Raspberry Pi Zero W into a wearable device, including a microphone for input and speakers or headphones as the output device. The focus here is on a low-cost wearable that meets the above requirements.

3.4 Summary

The beginning of this chapter covered the central aspects of the current workflow with the existing applications provided by smaXtec. The users can decide if they want to use the web application or the mobile phone application, which is available for Android and iOS. Since the majority of the smaXtec users is accessing the system via a smartphone, the current solution requires at least one hand for operating and navigating through the system. Based on that, the functional, as well as the non-functional requirements for the technical implementation of the alternative HCI approach in smart dairy farming, were identified. We first acquired the non-functional requirements, which meant that the application required ease of use, hands-free operation, no mandatory user focus on a screen, handling the harsh environment on a farm, and finally the use of a workable technology. The functional requirements cover the expected functionalities of the implementation, such as a simple activation phrase, simple commands for the users and interaction based on the input provided. To achieve this, the input methods and output devices from Chapter 2 were considered and roughly filtered according to the requirements. Narrowing down those lists further resulted in using voice as the final input method and wearables in combination with headphones or speakers as the output device. After several Smartwatches were available, the Samsung Gear 2, the Apple Watch Series 0 and the DZ09, cheap alternatives ordered online, several devices were tested for feasibility. Since the results of the experiments were unsatisfactory, the final decision for the hardware led to using an inexpensive Raspberry Pi Zero W with additional peripheral such as a microphone and headphones or speakers for realizing the handling using voice. We will implement a Custom Skill for Amazon's Alexa which serves as the software for interpreting and responding to the voice commands. We cover the implementation details in the following chapter.

4 Implementation

This chapter discusses the technical implementation by utilizing the capabilities of Amazon's Alexa service in combination with a Raspberry Pi Zero W. As described in the previous Chapter 3 the Raspberry Pi Zero W is inexpensive, readily programmable and extensible with various peripheral equipment and therefore functions as the hardware foundation. Due to existing Alexa libraries for the Raspberry Pi Zero W, the realization of the software bases on Amazon's Alexa services, such as the Alexa Skills Kit (ASK)[1] and the Alexa Voice Service (AVS)[2]. Figure 4.1 illustrates the basic system architecture of the prototype and the following sections cover a more detailed description of the individual component implementation.

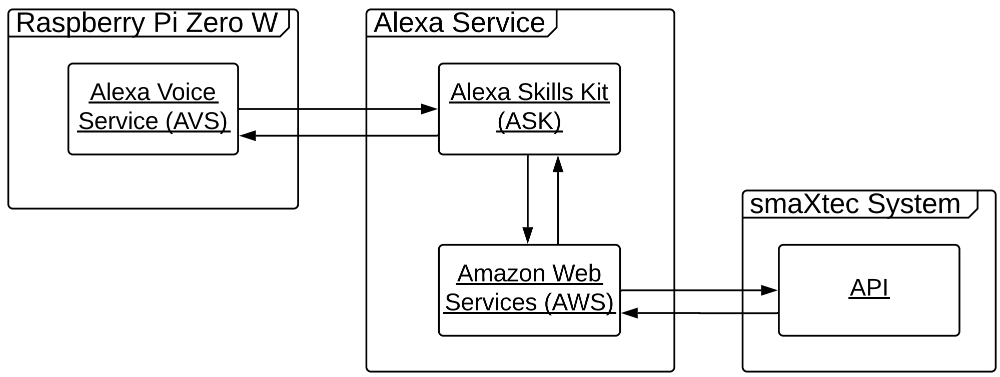

Figure 4.1: System Architecture of the Prototype

[1]https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html

[2]https://developer.amazon.com/docs/alexa-voice-service/api-overview.html

4.1 Alexa Skills Kit (ASK)

The Alexa Skills Kit (ASK) is a software development kit that enables the development of so-called Alexa Skills. Those skills include Alexa's built-in capabilities, which are used to have a conversation with Amazon's virtual assistant Alexa. Some examples of skills are starting a music stream from different providers, asking for the weather or asking definitions on Wikipedia. With the help of the ASK, one can develop new skills for Alexa. ASK includes different skill types, which are the following:

- **Smart Home Skill API:** this skill type enables the control of cloud-enabled smart home devices, such as lights, cameras, smart TVs, and so on.
- **Video Skill API:** as the name suggests this skill type allows the user to control cloud-enabled video services, for example playing a movie or finding a TV show.
- **Flash Briefing Skill API:** this skill type provides content for a customer's flash briefing, for instance providing the user with a news overview.
- **Music Skill API:** with the help of this skill type users are able to select, listen to and control audio content which is streamed on an Alexa-enabled device.
- **Custom Skill:** the custom skill type can be adapted individually, therefore applications are not limited like the previous ones. For example, one could use a custom skill to interact with an external web service or even setup games with Alexa.

Since there is no predefined skill type which covers the realization of an alternative HCI concept in the field of smart dairy farming, only the custom skill type is suitable for the implementation. The following sequence diagram, illustrated in Figure 4.2, describes the fundamental workflow of an Alexa skill.

At first, the user says an utterance to an Alexa device. The device then interprets this utterance locally with the help of speech recognition, machine learning, natural language processing (NLP), text to speech, or similar techniques. Afterwards, the skill sends the resulting intent, formatted as a

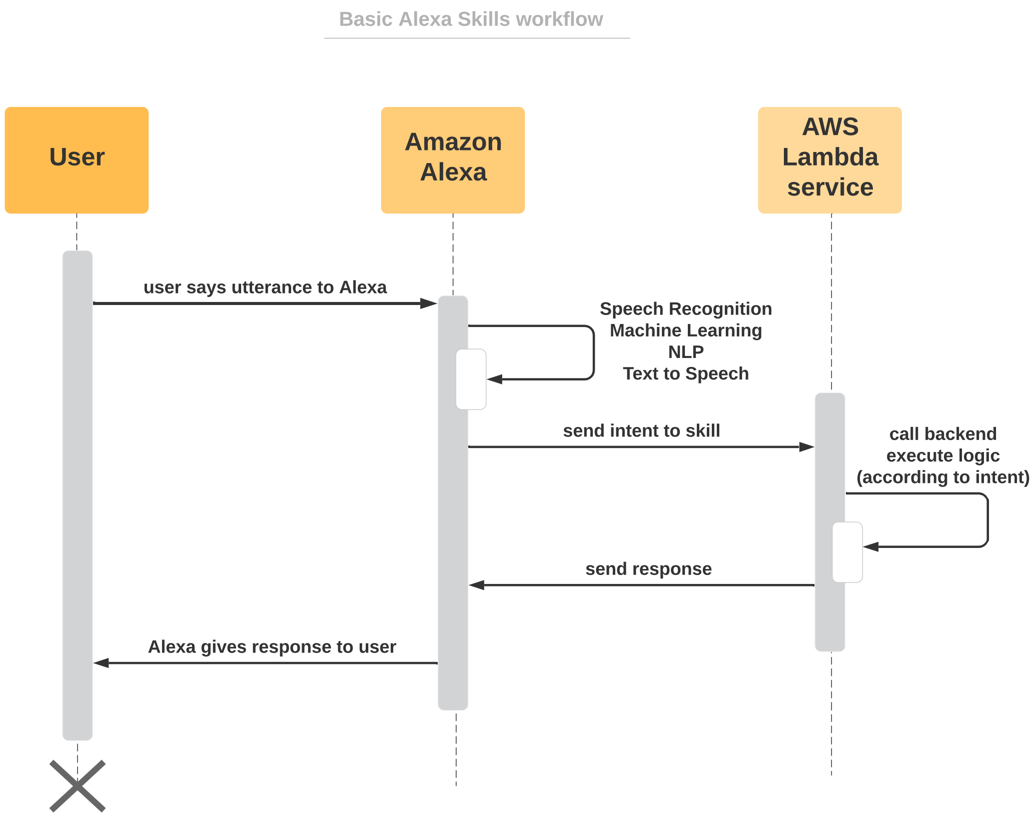

Figure 4.2: Sequence Diagram of the basic Alexa workflow

JSON document, and forwards it to the backend service. The corresponding logic is then executed depending on the received intent, which processes all needed steps and creates an output, again in JSON format. After the Alexa device receives the response, it provides the appropriate output to the user, which is either audio, textual or visual. Since this workflow describes the Alexa skill system very general, the following sections discuss the details of the prototype implementation using a custom skill in combination with an AWS Lambda service, which handles the requests.

Custom Interaction Model

A custom skill consists of many different components. First of all, intents specify the core functionality. More precisely, these are actions a user can execute when using an Alexa skill. Next, there are sample utterances, which are words and phrases the user says to invoke the intents. We mapped the utterances to the intents, thus forming the interaction model of the skill. Furthermore, an invocation name serves as identification for the skill. That name is essential for the user to start a conversation with the skill. As described previously, an additional service that handles the intents as requests and responds accordingly is needed to generate output to the user. The AWS Lambda service, used for the prototype implementation, covers these functionalities and the next section covers a more detailed description. Finally, a configuration uniting all components is needed so that Alexa routes the requests correctly to the service and vice versa.

Developing a new Alexa skill requires an Amazon developer account. Such an account gives access to the Amazon Developer Services, which include software development kits to various Amazon applications, such as Fire tablets, Alexa-enabled devices, and many more. Furthermore, the user acquires the possibility of building, testing, and distributing applications in the Amazon Appstore. The first step in creating a custom skill is to come up with an invocation name. As previously described, this name is used to start the interaction with an Alexa skill. The chosen skill invocation name is **smax tec**. First, it fulfils the requirements of Amazon by consisting of two or more words in lower case characters and spaces between the words. It also covers the requirement, defined in Chapter 3, to use a simple catchword for activating the skill, since there is a direct connection to the smaXtec company. The second step is to define the intents, more precisely the sample utterances which the user uses to interact with the skill. Besides the built-in intents such as, for example, Amazon's HelpIntent, which is activated when the user asks for help, the skill can also handle a custom intent. Figure 4.3 illustrates all intents used for the prototype, whereas primarily the focus lies on the self-defined voiceCommands intent.

NAME	UTTERANCES	SLOTS	TYPE	ACTIONS
AMAZON.FallbackIntent	-	-	Built-In	Edit | Delete
AMAZON.CancelIntent	-	-	Required	Edit
AMAZON.HelpIntent	-	-	Required	Edit
AMAZON.StopIntent	-	-	Required	Edit
AMAZON.NavigateHomeIntent	-	-	Required	Edit
voiceCommands	4	2	Custom	Edit | Delete
AMAZON.NoIntent	-	-	Built-In	Edit | Delete

Figure 4.3: Intents of the implemented prototype

Figure 4.4 displays the sample utterances of our custom skill. Since the main user group, as well as the selected testers for the prototype, are from the German-speaking area the prototype works in German. According to the functional requirements described in Chapter 3, this intent has utterances that include one of smaXtec's predefined actions as well as a reference to an animal. Since every utterance needs an action in combination with an animal, we defined those values as variables in the utterances, so-called intent slots.

The intent slots are written within curly brackets and highlighted with a coloured background in the utterances.

Figure 4.4: Utterances of the implemented prototype

Each intent slot consists of a slot name assigned to a slot type, displayed in Figure 4.5. The slot types, shown in Figure 4.6, define the recognition and handling of the incoming data as well which values to expect in the slot.

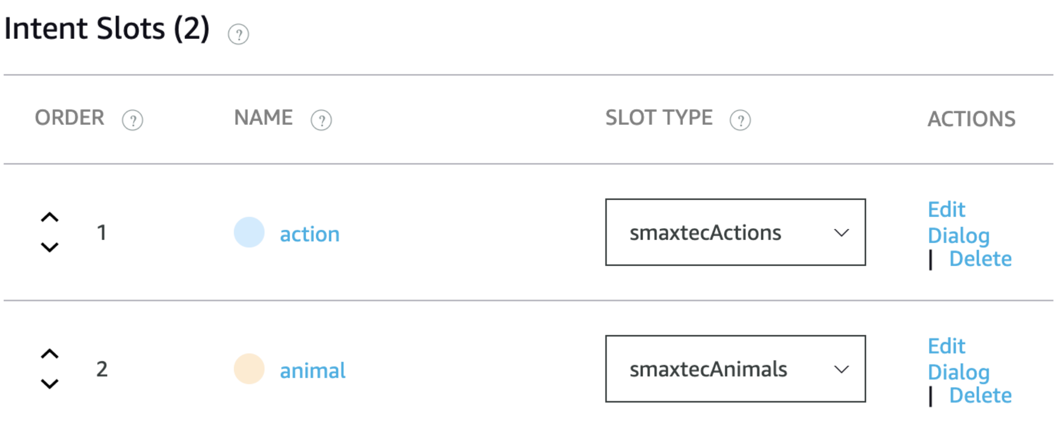

Figure 4.5: Intent slots of the implemented prototype

NAME	SLOT VALUES	TYPE	ACTIONS
smaxtecActions	6	Custom	Edit | Delete
smaxtecAnimals	1	Custom	Edit | Delete

Figure 4.6: Slot types of the implemented prototype

Figure 4.7 shows the smaxtecAnimals slot type, which is dynamic because the animals are unique to each user and therefore loaded on startup. Nonetheless, the slot type requires at least one static value because otherwise, the skill does not build. The next section covers the description of the exact behaviour regarding the loading of the animals on startup of the skill.

VALUE ?	ID (OPTIONAL) ?	SYNONYMS (OPTIONAL) ?
Musterkuh	Enter ID	Add synonym

Figure 4.7: Animals slot type of the implemented prototype

The smaxtecActions slot type, illustrated in Figure 4.8, is static and consists of the six smaXtec actions heat, insemination, pregnancy, dry-off, calving and abort, that already exist in the current system. In addition to the fixed values, some actions have interchangeable synonyms.

After discussing the details of the skill's behaviour in this section, including its invocation name, intents, utterances, and slot types, the skill itself is all set. One additional step to make the skill usable is missing, which is the handling of incoming requests and providing the appropriate responses to

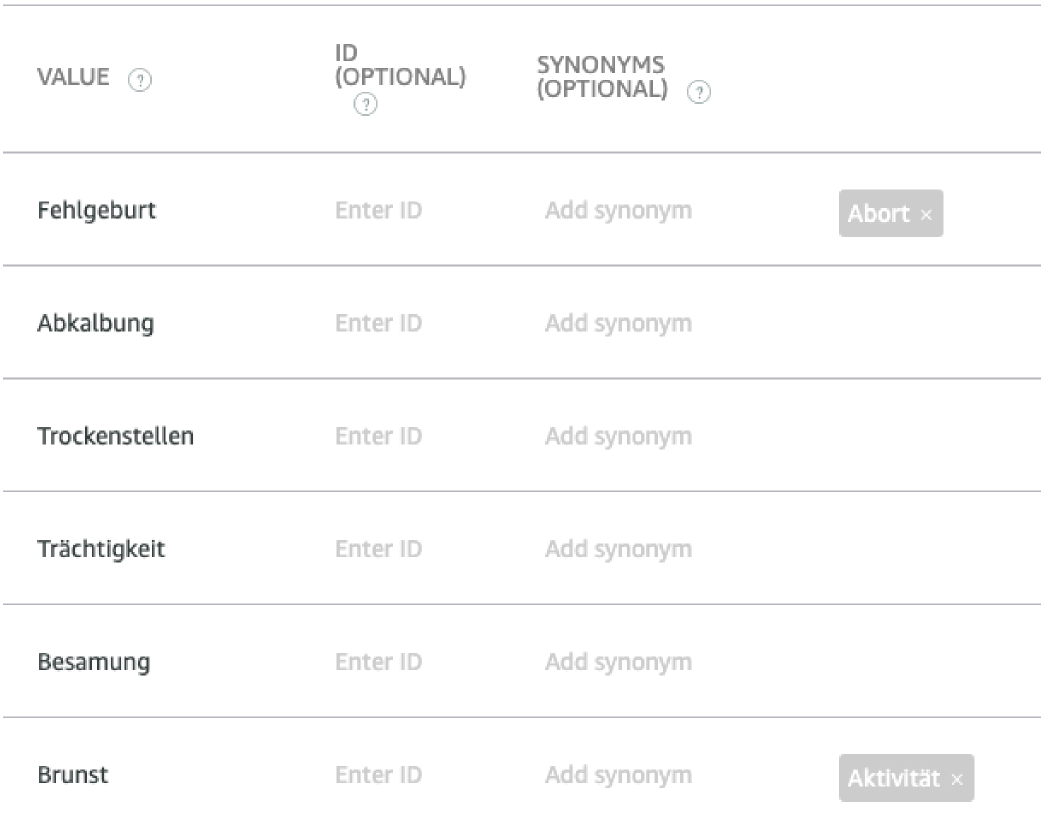

Figure 4.8: Actions slot type of the implemented prototype

the user. Therefore we used the AWS Lambda service where the next section covers the implementation details.

AWS Lambda

Building a custom skill requires access to an endpoint that has a connection to the internet. There are two available options for providing such a service to ensure Alexa's functionality with a custom skill. The first option is an AWS Lambda function, which is one of Amazon's Web Services [3]. Here, the code runs in the cloud, and Amazon manages the servers. Lambda functions can be written in Node.js, Java, Python, C#, or Go. The alternative requires writing a web service and hosting it with a cloud hosting provider by oneself. Therefore the service can be written in any programming language. In both cases, Alexa forwards the user request to the service, which then executes the necessary actions, builds an appropriate response and sends it to the user. The decision fell in favour of the AWS Lambda function since it does not require any server management. Since having the most experience with Python rather than any of the other options regarding the programming language, it serves as the programming language for the Lambda function.

The Lambda function consists of three main parts. The first part covers the handling of different request types. Next comes the initialization of available commands and animals and the intent handling, which represents the core functionality. Finally, the last part consists of the response generation in a JSON format, needed for all request types. The following sections deal with each part of the Lambda function in more detail, including code snippets as well as diagrams to clarify how it functions.

[3]https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

Request Handling

The communication between Alexa and the AWS Lambda works via requests. There are four standard request types[4] and various ones associated with certain interfaces. As the skill does not use any special interface, we will have a brief overview of the four standard types:

- **LaunchRequest:** user invokes the skill without intent.
- **IntentRequest:** user invokes the skill with a defined intent.
- **SessionEndedRequest:** when the session ends unexpectedly.
- **CanFulfillIntentRequest:** when Alexa queries the skill to check if the skill is able to process an intent request.

Since we implemented the first three types, the Lambda function needs a function for the correct request handling. This function, named lambda handler, is displayed in Figure 4.9. In the lambda_handler, we first check whether the session is newly created and if not, which request type the Alexa service has sent. Based on the input, the appropriate function, which handles each type of request, is then called. Each of the request handler functions, displayed in Figure 4.10, processes the given input and sends a message in JSON format to the Alexa service. The next section covers the initAnimals function, which is initiated by a LaunchRequest.

```
def lambda_handler(event, context):
    if event['session']['new']:
        on_start()
    if event['request']['type'] == "LaunchRequest":
        return on_launch(event)
    elif event['request']['type'] == "IntentRequest":
        return intent_scheme(event)
    elif event['request']['type'] == "SessionEndedRequest":
        return on_end()
```

Figure 4.9: AWS Lambda: lambda handler

[4] https://developer.amazon.com/docs/custom-skills/request-types-reference.html

```
def on_start():
    return output_json_builder_with_reprompt("Session Started.", "",
        False, [{'type': 'Dialog.UpdateDynamicEntities', 'updateBehavior': 'CLEAR'}])

def on_launch(event):
    initAnimals()
    onlaunch_MSG = "Hi und willkommen bei der smaXtec Sprachsteuerung." +
        "Folgende Kommandos stehen zur Auswahl: " + ', '.join(map(str, ACTIONS_LIST_DE)) +
        ". Sagen sie ein Kommando in Verbindung mit dem entsprechenden Tier."
    reprompt_MSG = "Kann ich sonst behilflich sein?"
    return output_json_builder_with_reprompt(onlaunch_MSG, reprompt_MSG,
        False, [{'type': 'Dialog.UpdateDynamicEntities', 'updateBehavior': 'REPLACE',
        'types': [{'name': 'smaxtecAnimals', 'values': animalValues }]}])

def on_end():
    return output_json_builder_with_reprompt("Session stopped", "",
        True, [{'type': 'Dialog.UpdateDynamicEntities', 'updateBehavior': 'CLEAR'}])
```

Figure 4.10: AWS Lambda: request handlers

Animal Initialisation

Figure 4.11 represents the code of the initialization step of the animals in the AWS Lambda function, which is particularly interesting as the animals are not predefined statically like the actions. Therefore, the user must provide his smaxtec credentials to obtain the animals which are in interest for the user. First, the service gathers a session token from the smaXtec API with the help of the provided credentials. Then it uses the session token to authorize the user because otherwise, operations with the API will not work. The next step is to gather all animal objects of the last selected organization, which is usually the user's farm. These objects contain a lot of data irrelevant for the application. For example, there is no need to store the birthday or the race of an animal. Therefore, we extract the name and id of the animal objects and add them into a separate list exclusive of those animal values. This list serves the purpose of checking if the user input contains a valid animal name that is associated with the farm. Following, Figure 4.12 visualizes the just described code as a sequence diagram.

```python
def initAnimals():
    email = 'john.doe@mail.com'
    pw = 'password'

    token_resp = req.get(apiv1+"user/get_token?email="+email+"&password="+pw)
    auth_token = token_resp.json()['token']
    hdr = {'Authorization': 'Bearer ' + auth_token}
    user_resp = req.get(apiv1+"user", headers=hdr)
    orga_id = user_resp.json()['metadata']['messenger']['lastSelectedOrganisationId']
    animals_resp = req.get(apiv2+"organisations/"+orga_id+
        "/animals?archived=false", headers=hdr)

    for obj in json.loads(animals_resp.text):
        synonyms = []
        if obj['official_id'] is not None:
            synonyms.append(obj['official_id'])
        if obj['display_name'] is not None:
            idx = 0
            if obj['display_name'].find(' - ') is not -1:
                idx = obj['display_name'].find(' - ') + 3
            value = obj['display_name'][idx:]
            Animals_LIST.append(value)
        animal_id = obj['_id']
        animalValues.append({'id': animal_id, 'name': {'value': value}})
```

Figure 4.11: AWS Lambda: initAnimals function

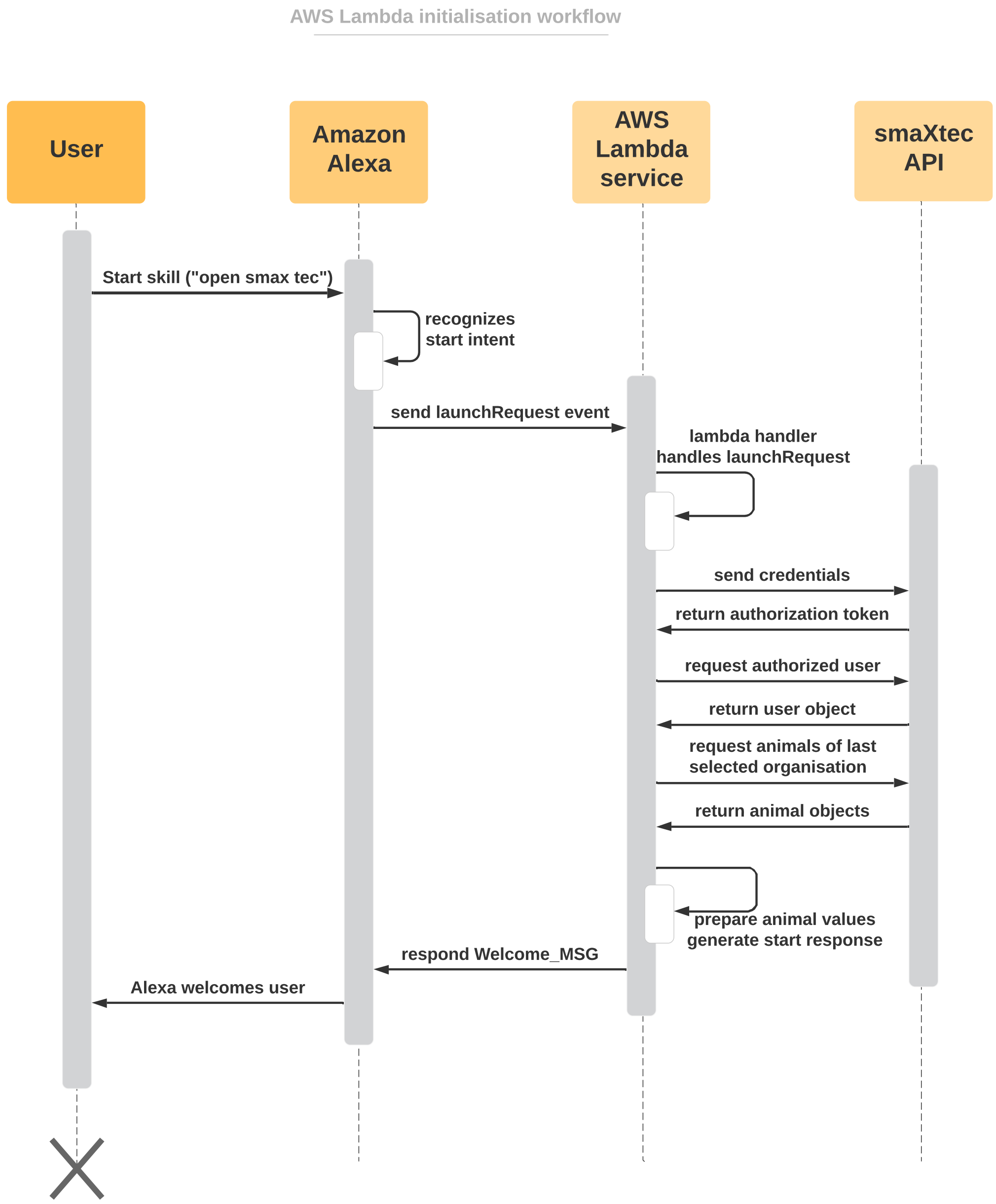

Figure 4.12: Sequence Diagram of the AWS Lambda initialization step

IntentRequest Handler

As mentioned earlier, the skill invokes the IntentRequest when the user says a defined intent of the custom skill. First of all, since there are multiple intents available, as seen in Figure 4.3, there is a need for correctly handling them as well. Therefore, the intent_scheme function comes into play, displayed in Figure 4.13, called in the lambda_handler (Figure 4.9).

```
def intent_scheme(event):
    intent_name = event['request']['intent']['name']

    if intent_name == "voiceCommands":
        return interpret_command(event)
    elif intent_name in ["AMAZON.NoIntent", "AMAZON.StopIntent", "AMAZON.CancelIntent"]:
        return stop_the_skill(event)
    elif intent_name == "AMAZON.HelpIntent":
        return assistance(event)
    elif intent_name == "AMAZON.FallbackIntent":
        return fallback_call(event)
```

Figure 4.13: AWS Lambda: intent scheme function

The predefined Amazon intents cover different scenarios. The NoIntent, the StopIntent, and the CancelIntent stop the skill by choice and hereafter Alexa says goodbye to the user. The HelpIntent provides help by repeating the instructions on how to correctly interact with the skill. Lastly, the FallbackIntent becomes effective when processing the intent fails, like the user says an unknown utterance. The user invokes the core functionality by saying one of the defined utterances, shown previously in Figure 4.4. In that case, we extract the action and animal name from the incoming intent. Thereupon, comparing the extracted values to the initialised animals as well as the predefined actions takes place. If they do not match, the service generates a response asking the user to repeat the input. If they match, a response reports the action and the animal name to the user.

Output Generation

The last section regarding the AWS Lambda covers the output generation briefly. As mentioned earlier, the generated output gets written into a

JSON format. The main part of the response, which is in the interest of the user, is the actual message provided to the user. It either consists of the welcome message after a LaunchRequest or a message after an IntentRequest. Depending on the intent, this can be a helping message, a goodbye message, a fallback message, or positive or negative output. The Alexa service receives the output of the AWS Lambda and provides an audible output to the user.

4.2 Alexa Voice Service (AVS)

Besides using the Alexa Skills Kit for the new custom skill, which is a pure software solution, Alexa requires a hardware device for input and output as well. As worked out in the previous chapter, a Raspberry Pi Zero W will serve as the hardware basis for this thesis. Since the Raspberry Pi Zero W does not come with Alexa built-in by default, the Alexa Voice Service (AVS) comes into play. The AVS is used to integrate Alexa with speech recognition, natural language understanding, and many other tools on a chosen hardware. The following sections describe the used hardware components as well as the software utilization.

Hardware Components

The Raspberry Pi Zero W forms the hardware foundation for the Alexa device. Research indicated that the Raspberry Pi Zero W is capable of working as an Alexa device. Additionally, the costs are approximately 10 €, which is much cheaper than an Amazon Echo (77 €) or an Echo Dot (30 €). Due to its small size, about half the size of a credit card, it is also easier to convert it into a wearable device than an Echo device. The Raspberry Pi Zero W comes with the following features:

- 1GHz, single-core CPU
- 512MB RAM
- onboard Wireless LAN - 2.4 GHz 802.11 b/g/n
- onboard Bluetooth 4.1 + HS Low-energy (BLE)
- Mini HDMI and micro USB ports

- Micro USB power supply connector

In addition to the Raspberry Pi Zero W itself, which is just the processing unit, the final version of the prototype needs further peripherals like a micro USB to USB adapter, a USB to audio connector, headphones and a microphone with an audio jack to work as an Alexa device. For the prototype, parts of the peripherals were ordered online as well as using already owned hardware as well. Figure 4.14 images the fully assembled prototype. After assembling the prototype, we carried out the setup of the software on the Raspberry Pi Zero W to connect the device with the custom skill.

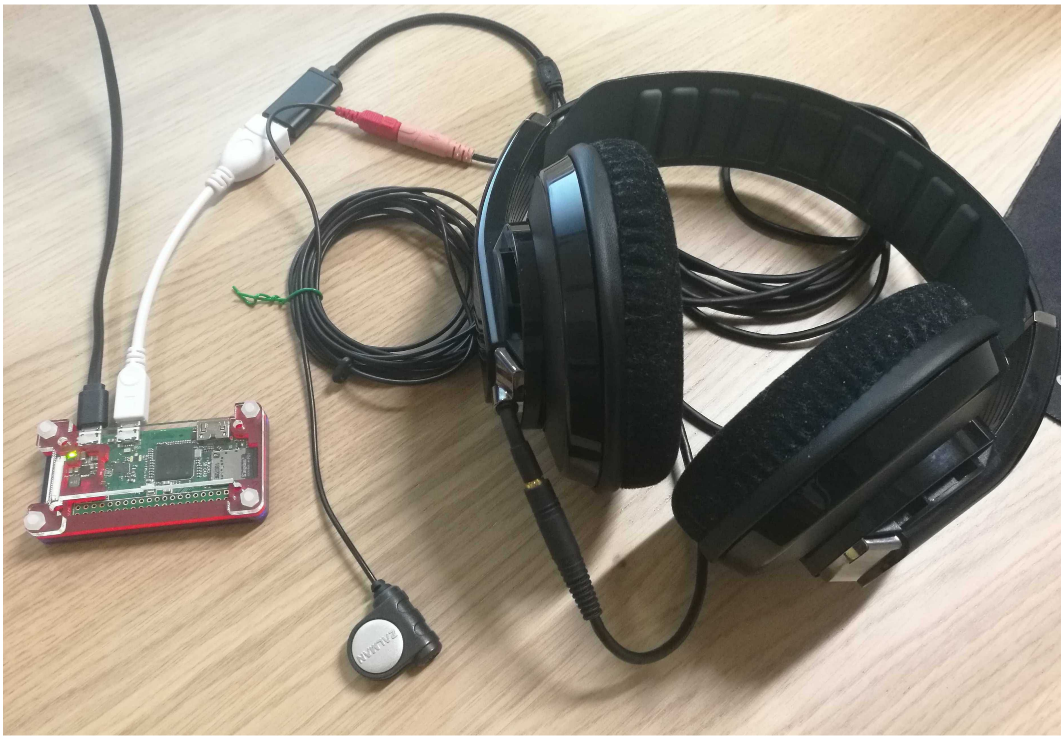

Figure 4.14: Assembled prototype including input and output peripherals

Software Components

Before interacting with Alexa, it is mandatory to take preparation measures by executing them with the help of online documentation. The first step requires flashing a microSD card with the Raspbian operating system using the application balenaEtcher[5]. Right after the flashing process, we could finally insert the SD card into the Raspberry Pi Zero W. On first boot we configured the general data like the device name, password, Wi-Fi and made sure that the peripherals are working properly as well. Using the Raspberry Pi Zero W as an Alexa device requires the registration of an AVS product, which takes place in the AVS developer console, located in the Amazon developer account. The registration includes various inputs like the product name, the application purpose, and more. Most importantly, we had to enter the IP address of the Raspberry Pi Zero W for successfully sending requests and responses over the right connection. Thereupon, when booting up the Raspberry Pi Zero W, the Raspian OS needs to be updated. Afterwards, the installation of the actual Alexa software, namely the avs-device-sdk[6], downloaded from a Git repository and installed via their instruction, which required the data of the AVS product at a later time. The Raspberry Pi Zero W is usable as an Alexa device after finishing the installation and additional troubleshooting.

4.3 Summary

This chapter summarizes the implementation details regarding the pure software Alexa skill and the wearable prototype, which consists of hardware as well as software. The first part describes the specifics of the Alexa Skills Kit. That includes use case examples and the different types of Alexa skill as well. Since the custom skill type is modifiable according to ones' needs, it serves as the basis for the Alexa interaction. Further to the theoretical background of an Alexa skill, the chapter covers the implemented intents, utterances, and slot types needed for an Alexa skill. After explaining the

[5]https://www.balena.io/etcher/
[6]https://github.com/alexa/avs-device-sdk

details regarding the Alexa service, the AWS Lambda function, which is responsible for handling incoming requests and return responses accordingly, was described. More specifically, it covers the general request handling, the initialisation of animals, the intent handling as well as the output generation. In addition to the purely software-based skill, there is also a need for hardware to interact with Alexa. The chapter includes the technical specifications of the Raspberry Pi Zero W and additional peripheral equipment used to enable communication with Amazon's Alexa. That includes a microphone as an input device and headphones as the output device. The last part of this chapter deals with the circumstance of registering a device, which does not come with Alexa built-in, at the Alexa Voice Service. This step is needed to enable access to Alexa services and tools to external devices like the Raspberry Pi Zero W. The upcoming chapter deals with the testing and evaluation of the realized prototype.

5 Evaluation

In this chapter, we scrutinize the performance of the implemented prototype with the help of a usability test. Therefore, two groups of volunteers executed a live test under specific conditions. This chapter describes the used methods, the test setup, as well as the final results.

5.1 Methodology

The most important factors about the evaluation process are the participants, the test setup, including the testing procedure itself, and finally, the reporting of the testing results. Firstly, we split the participants into two groups, an expert group with domain knowledge as well as a non-expert group with general computer usage skills. Both groups had to execute the same tasks so that usability problems could be distinguished. Domain experts focussed more on the functionality and the theoretical embedding of the voice input prototype, whereas we expected the group of non-expert to highlight general design problems.

The evaluation consisted of three parts: a pre-questionnaire, the execution of predefined tasks and a post-questionnaire. The pre-questionnaire asked the user about general information, such as the age, the field of study or job title, and about their usage behaviour regarding computers and smart devices. The evaluation tasks regarding the voice control prototype focussed on general interaction that is anticipated in real life as well. After the execution of the given tasks, the users had to fill out the post-questionnaire. The questionnaire consisted of the System Usability Scale (SUS), used for rating the usability of the prototype, as well as a couple of open questions.

This questions included possible improvements regarding the voice control prototype as well as a comparison to the existing system.

System Usability Scale

According to Brooke et al. (1996), SUS proved to be a robust and reliable evaluation tool. It is used right after the testing of the implementation before asking the open questions. The SUS consists of ten Likert scale (Likert, 1932) questions, which have five to seven options ranging from strongly disagree to strongly agree. After the question rating, we can calculate a SUS score between 0 and 100. According to Ghosh, Foong, Zhang, and Zhao (2018), the SUS applies to voice-based user interfaces like our implemented prototype as well.

5.2 Participants

The usability testing took place with a total of ten testers, divided equally in five male and five female testers. The participants had an age range from 26 to 37, which resulted in an average mean AM=28.6, with a standard deviation of SD=3.64 years. All participants were familiar with using computers and smartphones daily, mainly due to their profession or field of study. On the contrary, the participants do not use voice commands regularly. Only three of them have a voice-controlled device. As mentioned earlier, we divided the participants into two groups, six domain experts and four non-experts. Half of each group has already participated in a usability test before this evaluation. The only requirement for the selection of the participants was that they understand German and can speak it. That is because we programmed the prototype, to be precise the custom skill for Alexa, with German utterances and action slot values, since the headquarter of smaXtec is located in Graz, Austria and a large number of customers is from the German-speaking area.

Experts

The six experts consisted solely of employees from the smaXtec company. Since real end-users were not available in such a short time frame, the employees were chosen on the facts that they either have a farming background, to be precise four out of six grew up on a dairy farm, or are a lot in contact with the smaXtec customers due to their profession in the customer support of smaXtec. Furthermore, the experts are using the existing system regularly, if not daily, and therefore did not need any background about the system.

Non-Experts

The non-experts consisted of four people who never used the existing smaXtec system at all. Therefore, we invested some additional time to give them a brief background knowledge about smaXtec as well as presented them the usage of the existing system. None of the non-experts owns a voice-controlled device, and therefore the experience with such devices was nonexistent.

5.3 Prototype Testing

Each survey participant had to evaluate the prototype individually. Thus we made sure that the testers do not influence each other, which would have led to falsified test results. As described in the previous section, each tester performed the same flow, starting with the pre-questionnaire, going over to the tasks on testing the prototype, and concluding with the post-questionnaire. The only difference was that the participants belonging to the non-expert group got an introduction to the existing smaXtec system, whereas the experts could start immediately after the pre-questionnaire.

Test Setup

The testing environment was set up in a small meeting room at the smaXtec headquarter. The testing equipment consisted of the Raspberry Pi Zero W, functioning as the Alexa device, including a microphone as input and headphones as output peripherals. Figure 5.1 displays the testing environment, Figure 4.14 in Chapter 4 features a close-up view of the prototype. Additionally, a recording of a dairy farm played on a TV to simulate a realistic background noise. We ensured that all participants have the same conditions and therefore measured the noise with the help of a sound level meter on a smartphone application[1].

Figure 5.1: Testing Environment for the Evaluation of the Prototype

[1]https://play.google.com/store/apps/details?id=com.gamebasic.decibel

Tasks

For the prototype evaluation, we defined a couple of tasks that represent a partial amount of the daily chores of a worker on a dairy farm as good as possible. In addition to the tasks which should be solvable with the prototype, we introduced some unsolvable tasks to test some corner cases as well. After a short introduction into the background of the tested prototype, all participants got the same set of tasks to fulfil, which were the following:

- Start the "smax tec" skill
- Add an action[2] of your choice to an animal[3] of your choice
- Add an unknown action to an existing animal of your choice
- Add an existing action of your choice to an unknown animal
- End the skill

After the participants executed all the tasks and got an impression on how the system works, we also performed a stress test to define the boundaries regarding the background noise. Therefore, the participants needed to execute the second task from the previous list repeatedly, and after each positively processed try, we increased the sound of the TV until the prototype did not work reliably anymore. In the upcoming section, we present the results of the testing as well as the outcome data of the questionnaires.

5.4 Results

We split the results into three sections. The first one focuses on statistics regarding the general data, gathered with the questionnaire which the participants filled out before the testing. The second section covers findings according to the testing phase and the stress test, as well as the SUS results. Lastly, we sum up the answers to the final open questions, which include various information regarding the prototype. The data related to the past

[2][Heat, Insemination, Pregnancy examination, Dry Off, Abort]
[3][Musterkuh (needed for publishing the skill), Giraffe, Afrika, Gisela, Angiecurtis, Goldie]

two sections come from the post-questionnaire, evaluated after the prototype testing.

General Results

After gathering the general information about the participants, they had to fill out a survey about their experience with computers and smart devices. The ratings were based on a Likert scale from 1 (not at all) to 5 (fully agree /daily). Regarding their computer expertise, four participants rated themselves the maximum score, two picked the second-best option and the rest the medium value, so in total an AM=4.0 and SD=0.94. All testers stated that they use computers and mobile phones daily. On the contrary, they do not use smart devices, such as smartwatches or smart TVs, and voice-controlled devices much. Smart devices reached an AM=2.6, SD=1.58, and only two participants use them daily. Figure 5.2 shows the exact distribution.

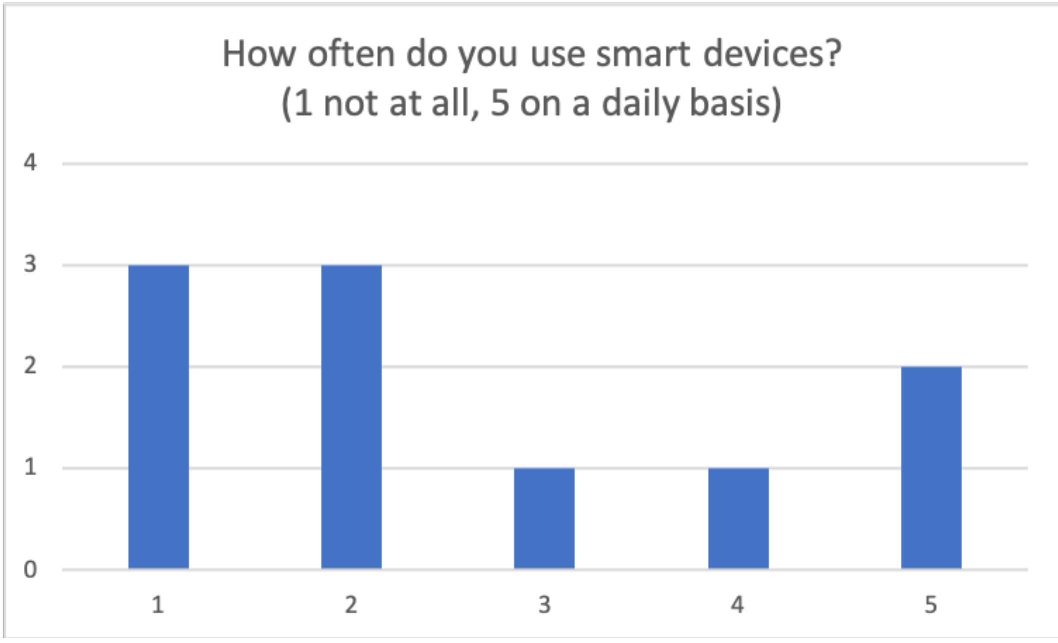

Figure 5.2: Distribution of smart device usage

Voice-controlled devices performed even worse, with an AM=1.6 and SD=0.7, whereas a single three was the highest rating, displayed in Figure 5.3.

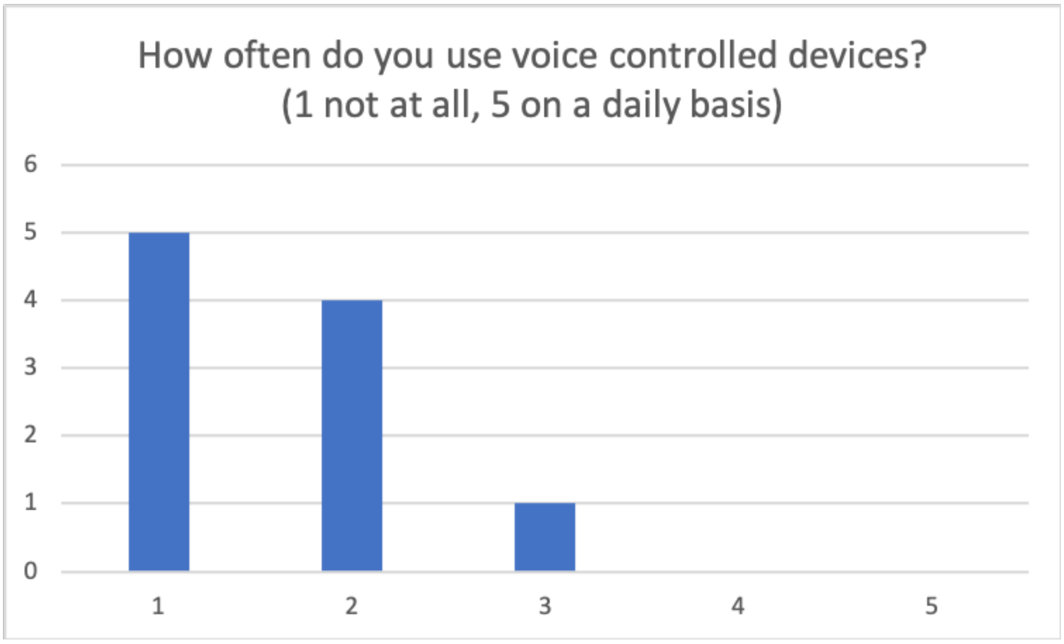

Figure 5.3: Distribution of voice controlled device usage

This is also reflected in the low distribution of voice-controlled devices since only three users, all of them from the group of experts, own such a device. As expected, the usage of the smaXtec system highlighted the difference between the two groups. The experts use it daily, whereas the non-experts do not use it at all. On top of that, the expert group also rated their satisfaction with the existing system, which resulted in a high satisfaction level with an average value of 4.17.

Usability Results

As described in the test setup section, we used a sound level meter application on a smartphone to simulate similar conditions for each tester. Therefore, we set the volume of the background video to a fixed level and took measurements. The average sound level during the normal task execution was 42dB, ranging from 38dB to 48dB. That sound level is comparable to average home noise, according to Healthwise Staff (2018). Afterwards, in the stress test phase, the volume was increased sequentially after the user confirmed that the Raspberry-Alexa recognized a command correctly. A maximum sound level between 80 and 85 dB sufficed to interfere with the Alexa device so that it could not execute the user's commands. Partially the prototype recognized commands even with the loud background noise, but due to technical limitations, we did not increase the sound further. According to Healthwise Staff (2018), noises above 80dB are comparable to a noisy restaurant or a power lawnmower, whereas they consider sounds above 85dB harmful.

As mentioned earlier, we used the SUS by Brooke et al. (1996) to evaluate the usability of the implemented prototype. The score ranges from 67.5 up to 100, whereas nine out of ten participants rated the usability higher than 70. In total, the average score resulted in an AM=81.75 with an SD=10.41. The expert group rated the prototype below the total average, with an AM=76.25 and SD=7.20. The non-experts rated the prototype less critical, emphasized with the reached AM=90.0 with SD=9.35. Overall, based on the achieved high SUS ratings, the prototype performed astonishingly good.

Open Questions Results

This final section covers the overall impressions of the tested prototype, based on the answers to the open questions of the post-questionnaire. The questions included personal likes and dislikes, suggestions for improvements, missing functionalities, preference about input and output and finally a comparison to the existing smaXtec system.

The majority of the participants mentioned that the prototype was easy to use, and they liked the simple way to start the skill as well. Furthermore, they highlighted the easily memorizable utterances and phrases for interacting with the skill and the positive feedback when Alexa processed a correct utterance. Unfortunately, Alexa sometimes stopped the skill without being asked to do so, which left a slightly negative impression. Additionally, some false animal names and actions were processed as if they were defined, which was confusing. Of course, the testers suggested a few improvements. First, since the prototype only recognized five utterances, the users emphasized the expansion of additional ways to phrase a command. Since the testers had to restart the skill after each unwanted system crash, the participants asked for a way to skip the introduction. Additionally, there was a desire for better feedback after an unsuccessful utterance or when the user asks the device for help.

Since the implemented prototype contained just a subset of the functions of the existing smaXtec system, some testers wished to include these functionalities as well. Besides that, another missed functionality was to ask for available animals and also for multiple ways to refer to an animal. Regarding the used input and output devices, the users liked the combination of headphones and a microphone. One tester commented that wireless headphones would be more practical. Some testers also suggested using an additional visual output to look up recent requests and responses, but overall, they accepted the audio output well. Lastly, comparing the prototype to the existing system in use, the users responded with mainly positive feedback. The testers loved the easy way as well as the increase of speed to input data to the system. Furthermore, they highlighted the reliability even with high background noise. Some testers even stated to be excited and that it is an innovative solution. Moreover, hands-free operating has the potential to be relevant in the future since the traditional system requires multiple clicks to achieve the same results.

5.5 Summary

The evaluation of the prototype resulted in very positive feedback by both sides, the domain experts, who use the already existing system, as well as the non-experts, who did not interact with smaXtec before the usability test. Furthermore, the testers emphasized the faster processing of input via a voice-controlled system instead of the manual clicking of buttons and entering data with a keyboard, as is currently the case. Based on their entered information, the testers were not wholly familiar with using voice command devices for data input. The short testing times showed that the prototype was easy to understand. Although there was no time limit, the testing process lasted between 15 and 20 minutes at the most. The final average SUS score of 81.75 also shows a high acceptance regarding the prototype's usability.

On the other hand, the prototype only covers a subset of smaXtec's functionalities, and as a result, the domain experts were missing some of these features during the testing. Nevertheless, we could easily extend these functions if commercial use is requested. Finally, the testers were partially surprised that the prototype worked well even when with a high level of background noise, which suggests the feasibility in rough environments like dairy farms. To sum up, the overall good response of the testers indicates many benefits of a voice-controlled system in addition to the existing system. Therefore, an introduction of the implemented prototype to real-life users could achieve a new, innovative and easy to use way of interacting with the smaXtec system.

6 Lessons Learned

This chapter gives insights into new experiences acquired while looking out for the right hardware, the development as well as the evaluation of the implemented prototype. The following sections cover the knowledge gained while figuring out which hardware to use for developing and analyzing an alternative HCI concept in smart dairy farming. Moreover, they include new understandings of developing an Amazon device prototype. The final section discusses the testing of the prototype as well as evaluating it.

6.1 Hardware

The initial step was to find hardware for an HCI concept which is not widespread in smart dairy farming yet. First, all roads led to developing a prototype with the help of a smartwatch, due to their increasing popularity. Initially, the preferred option was the Samsung Gear 2 over an Apple Watch because of the open-source characteristics of Android. While testing the Samsung Gear 2, it emerged to be improper for the application. More precisely, the Samsung smartwatch pairs only with a defined set of Samsung devices but not with any Android device as expected. Further research revealed the inconvenient truth, that branded smartwatches mainly work exclusively within their brand-ecosystem. Thereupon, before switching to the Apple system, testing two cheap Android smartwatches without a known brand resulted in the reveal of another flaw of being hardly programmable. The last tested smartwatch, the Apple Watch Series 0, looked promising due to simple pairing with an iPhone as well as the option to extend it programmatically. After getting familiar with the Apple Watch, and especially with Siri, a weak point of the smartwatch appeared which made it infeasible for the thesis. The talkback feature with Siri, required for interaction, is only available for a newer version of the Apple Watch. Since the voice interaction feature was the major part of the planned implementation, the Apple Watch disqualified. Fortunately, the Raspberry Zero W and the needed peripherals proved to be easily customizable as well as affordable.

In conclusion, the lessons learned concerning the hardware, are the following: many Android smartwatches are either locked-in to their brand or hardly programmable. On the other side, older versions of Apple Watches are missing certain features, which would have been useful for the implementation.

6.2 Development

The development of the prototype resulted in obtaining much knowledge about creating an Amazon Skill, using the Amazon Web Services properly as well as working with the Alexa Voice service. First of all, the basic functioning and different types of Amazon Skills were not known at all beforehand. Furthermore, the needed cooperation between an Amazon Skill and a web service, either self-hosted or hosting it via AWS, turned out to work sustainable as well as being easy to setup. On the other hand, preparing the Raspberry Pi Zero W to work as an Alexa-enabled device with the help of AVS turned out to be somewhat complicated but still an overall great experience. The main knowledge gain according to the prototype development, which also consumed much time, came from enabling the Custom Skill to be accessible by Alexa on the Raspberry Pi Zero W. We learned it the hard way by finding out that the initially setup framework on the Raspberry Pi Zero W did not work with the latest API of Alexa. Fortunately, we found a working framework quickly, and the gathered knowledge of the first setup could be used to get the Raspberry Pi Zero W working as an Alexa device faster.

The first lesson learned is that accessing an Amazon Skill for testing via an Alexa-enabled device requires the skill to be published in the Amazon Skills Store. Therefore, the Custom Skill must go through a certification process that tests if the skill functions in general. Our Custom Skill did not pass the process at the beginning. Due to ambiguous error reports, it took a while to find the actual cause of the problem. It turned out that the dynamic entities, used for dynamically loading the animals based on the user credentials, are limited to 100 entries. On top of that, updating and clearing already existing entities did not work as expected. The idea was to remove all entries on the skill startup and replace existing ones if the limit exceeds. At first, we intended to access the animals via name and a unique ID which would use two entity slots per animal, limiting the approach to 50 animals. Since the updating did not work as anticipated, we rejected this approach and focussed on the animal names. Since the deletion on startup did also not work as needed, multiple starts of the skill resulted in exceeding the dynamic entities limit and thus led to failing the certification process. Therefore to pass the certification, the final published prototype has access to a small farm with only five animals.

6.3 Evaluation

The evaluation of the implemented prototype revealed the high value of using a combination of standardized evaluation methods like the SUS with open-ended questions. The SUS provides a structured procedure of evaluating the usability resulting in a total score for each tester which we then might use for statistical comparisons. The open-ended questions are more unstructured compared to the SUS, but they allow the users to express their thoughts and emotions openly. This method is only useful when the users are willing to answer the questions, which was the case. The answers revealed the difference in focus between the different testing groups. On the one side, the experts focused primarily on the functionalities of the prototype, whereas the non-experts were more attentive to the basic usability and feedback of the device. If we wish to develop the prototype further, it is sufficient to use the tester's feedback as input for the first steps.

7 Future Work

In this chapter, we will discuss which potential changes are useful to enhance the usability of the current prototype. The base of those changes are the findings and the lessons learned from the previous section, whereas others emerged during the prototype deployment or reference to the not realized functional requirements. In the long run, the changes are useful to build a voice command based device to sell it commercially.

7.1 Hardware Improvements

First of all, the current prototype is not a wearable device as planned. At the moment, it depends on a connection to a power outlet which reduces the portability drastically. Therefore a need for two changes comes into play. One is to power the Raspberry Pi Zero W with the help of a battery, and the other would be to provide a casing that is attached to the human body. The current prototype already has a housing that can easily be attached to the human body. Another helpful adjustment regarding the Raspberry Pi Zero W itself is to reduce the tangled mess of cables. First, there is the power supply cable, next to it is a micro USB to a USB adapter cable. Attached to that is a USB to Audio adapter which connects to a corded microphone and corded headphones. The previously mentioned change, using a battery as a power supply, helps to get rid of the power cable. Next, Bluetooth headphones with an additional USB Microphone or, even better, using a Bluetooth headset would decrease the messy wires and thus increase the wearing comfort.

7.2 Software Improvements

Concerning the current implementation of the Custom Skill, there are also a few potential enhancements to increase its usability. First and foremost, the skill must be accessible in English, additionally to the current German version. Other languages such as French, Hindi, Italian, Japanese, Portuguese, or Spanish, which are currently supported by Amazon's Alexa, may be added as well, depending on their priority. Besides the additional languages, we should extend the variety of utterances further. This change is crucial to minimize errors and avoid annoying repetitive requests, which helps to increase the acceptance by the customer. The highest potential for improvement lies in the AWS Lambda function since it handles the initialization, the incoming requests, and creates all responses. On the one hand, the initialization has to be improved so that the user can enter his credentials in a separate user interface or a credential file for uploading it to the AWS. On the other hand, the most critical issue in the current prototype is the limitation of the dynamic entities which cannot handle more than 100 objects. In addition to that, updating and deleting the entity objects need to be reworked, since they did not work out as initially intended. After fixing this behaviour, the planned usage of animal names and unique IDs, which was part of the functional requirements, is realizable as well. This step is mainly required to support voice command interaction on big farms with several hundred to thousand animals.

8 Summary and Outlook

The distribution of technology in the farming sector increased over the past years. Since sensors are becoming smaller and at the same time, their computing power increases, various possibilities to increase productivity and economic efficiency emerge. Still, the state of the art technologies, especially in dairy farming, often lack behind because the use of new technology requires training time as well as time to establish trust, and since farm work is usually very time consuming there is no time left for the additional workload.

This thesis dealt with the design, implementation, and evaluation of an alternative HCI concept tailored to the needs of technology in smart dairy farming. First of all, we gave insights into the history of dairy farming, our relation to the dairy sector, and also addressed presently used technologies and emphasized some of their weaknesses. We then presented different input methods and output devices. In addition to that, we talked about previous work in the field of smart farming and related applications with similar circumstances. We then moved over to highlight the current workflow with the smaXtec technology and discussed general issues. For example, it requires to use the hands to input data into the smaXtec system, by operating a mouse or a smartphone. Based on that, we identified the functional and non-functional requirements for the implementation and narrowed the previously described input methods and output devices down to viable device options. To be more specific, we focussed on wearables, and after testing various smartwatches unsuccessfully, we decided to convert a Raspberry Pi Zero W into an Alexa-based wearable device. Once we fixed the decision on the device, we started the implementation of Alexa on the Raspberry Pi Zero W. Additionally, we designed and programmed an appropriate Alexa Custom Skill, which serves as the connection to the smaXtec system. We mainly explained the technical details concerning an

Alexa skill and further Amazon services, which are needed to establish a working Alexa system.

Due to different technical limitations, we did not fulfil all identified requirements. Nevertheless, the prototype was up and running with a sufficient scope of functions, which were tested adequately with ten chosen testers. The positive feedback of the prototype evaluation highlighted the possibility of a high acceptance rate so that voice control in dairy farming is no longer just a future vision. Rather than that, it could already be implemented and used for actual operations. The implementation of an Alexa-based device proved to be straight forward since we developed a testable prototype in a short time. Therefore, a voice-controlled system would be an innovative step to stand out of the mass and take a leading role in convenient smart dairy farming.

List of Figures

Appendix

Bibliography

Apple Inc., (2019a). Sirikit documentation. Retrieved July 12, 2019, from https://developer.apple.com/documentation/sirikit

Apple Inc., (2019b). Sirikit documentation. Retrieved July 12, 2019, from https://support.apple.com/guide/shortcuts/welcome/ios

Azenkot, S. & Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. (pp. 251–260). doi:10.1145/2371574.2371612

Barkema, H., von Keyserlingk, M., Kastelic, J., Lam, T., Luby, C., Roy, J.-P., ... Kelton, D. (2015). Invited review: Changes in the dairy industry affecting dairy cattle health and welfare. *Journal of Dairy Science*, *98*(11), 7426–7445. doi:https://doi.org/10.3168/jds.2015-9377

Baumgart, D. C. (2011). Smartphones in clinical practice, medical education, and research. *JAMA Internal Medicine*, *171*(14), 1294–1296. doi:10.1001/archinternmed.2011.320. eprint: https://jamanetwork.com/journals/jamainternalmedicine/articlepdf/1105844/ilt0725_1294_1296.pdf

BCDairy. (2019). Technology in the dairy industry. Retrieved June 24, 2019, from https://bcdairy.ca/milk/articles/technology-in-the-dairy-industry

Becker, E., Metsis, V., Arora, R., Vinjumur, J., Xu, Y., & Makedon, F. (2009). Smartdrawer: Rfid-based smart medicine drawer for assistive environments. In *Proceedings of the 2nd international conference on pervasive technologies related to assistive environments* (p. 49). ACM.

Bellis, M. (2018). The history of the computer keyboard. Retrieved June 24, 2019, from https://www.thoughtco.com/history-of-the-computer-keyboard-1991402

bhphotovideo.com. (2019). Samsung gear 2 smartwatch (charcoal black) specifications. Retrieved June 28, 2019, from https://www.bhphotovideo.com/c/product/1046234-REG/samsung_sm_r380ovsaxar_gear_2_watch_titan.html

Birkfellner, W., Figl, M., Huber, K., Watzinger, F., Wanschitz, F., Hummel, J., ... Ewers, R., et al. (2002). A head-mounted operating binocular for augmented reality visualization in medicine-design and initial evaluation. *IEEE Transactions on Medical Imaging*, *21*(8), 991–997.

Biswas, K. K. & Basu, S. K. (2011). Gesture recognition using microsoft kinect®. In *The 5th international conference on automation, robotics and applications* (pp. 100–103). doi:10.1109/ICARA.2011.6144864

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4–7.

Brycearm. (2012a). Nvis nvisor st50.png. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=22611442

Brycearm. (2012b). Nvis nvisor sx111.png. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=22611441

Cattle Today Inc., (2017). Retrieved June 24, 2019, from https://cattletoday.com/

Chetan Dwarkani, M., Ganesh Ram, R., Jagannathan, S., & Priyatharshini, R. (2015). Smart farming system using sensors for agricultural task automation. In *2015 ieee technological innovation in ict for agriculture and rural development (tiar)* (pp. 49–53). IEEE.

Chhachhar, A. R. & Hassan, M. S. (2013). The use of mobile phone among farmers for agriculture development. *International Journal of Scientific Research*, *2*, 95–98. doi:10.15373/22778179/JUNE2013/31

Clarkson, J. & Eckert, C. (2010). *Design process improvement: A review of current practice*. Springer Science & Business Media.

Curry, A. (2013). Archaeology: The milk revolution. Retrieved June 24, 2019, from https://www.nature.com/news/archaeology-the-milk-revolution-1.13471

D. Kaplan, E. & Hegarty, C. (2005). *Understanding gps: Principles and application*.

Dambrāns, K. (2017). Samsung gear 2. Retrieved September 16, 2019, from https://commons.wikimedia.org/wiki/File:Samsung_Gear_2_(14391298545).jpg

Dao, D. Q. (2018). 5 pros and cons of face recognition technology. Retrieved June 24, 2019, from https://www.techfunnel.com/information-technology/5-pros-and-cons-of-face-recognition-technology/

Day In the Life of a Dairy Farmer. (2015). Retrieved June 24, 2019, from https://winnersdrinkmilk.com/2015/03/a-day-in-the-life-of-a-dairy-farmer/

Dent, J. (2018). Why get a tablet? Retrieved June 24, 2019, from https://www.whistleout.com.au/Tablets/Guides/why-get-a-tablet

Dimitrov, D. V. (2016). Medical internet of things and big data in healthcare. *Healthcare informatics research*, *22*(3), 156–163.

Driessen, C. & Heutinck, L. F. (2015). Cows desiring to be milked? milking robots and the co-evolution of ethics and technology on dutch dairy farms. *Agriculture and Human Values*, *32*(1), 3–20. doi:10.1007/s10460-014-9515-5

Fadul, M., Christopher, B., Alsaaod, M., Hüsler, J., Alexander, S., Adrian, S., & Hirsbrunner, G. (2017). Prediction of calving time in dairy cattle. *Animal Reproduction Science*, *187*. doi:10.1016/j.anireprosci.2017.10.003

Finch, C. (2014). The disadvantages of voice recognition software. Retrieved June 24, 2019, from https://www.techwalla.com/articles/the-disadvantages-of-voice-recognition-software

Fuller, S., Ding, Z., & Sattineni, A. (2003). Case study: Using the wearable computer in the construction industry.

Fulton, R. & Vandermolen, R. (2017). *Airborne electronic hardware design assurance: A practitioner's guide to rtca/do-254*. CRC Press.

Ghosh, D., Foong, P. S., Zhang, S., & Zhao, S. (2018). Assessing the utility of the system usability scale for evaluating voice-based user interfaces. In *Proceedings of the sixth international symposium of chinese chi* (pp. 11–15). ACM.

Hambal, A. M., Pei, D. Z., & Ishabailu, F. L. (2017). Image noise reduction and filtering techniques. *International Journal of Science and Research (IJSR)*, *6*(3), 2033–2038. doi:10.21275/25031706

Harlalka, R. (2013). Give voice to your apps: Why speech will replace touch as smart devices' primary input. Retrieved June 24, 2019, from https://thenextweb.com/dd/2013/12/18/give-voice-apps-speech-will-replace-touch-smart-devices-primary-input/

Healthwise Staff, (2018). Harmful noise levels. Retrieved September 24, 2019, from https://www.healthlinkbc.ca/health-topics/tf4173

Hollox, E. (2004). Evolutionary genetics: Genetics of lactase persistence–fresh lessons in the history of milk drinking. Nature Publishing Group.

Hoober, S. (2012). Mobile input methods. Retrieved June 24, 2019, from https://www.uxmatters.com/mt/archives/2012/11/mobile-input-methods.php

Hook, W. (2015). Apple watch. Retrieved September 16, 2019, from https://commons.wikimedia.org/wiki/File:Apple_watch.jpg

How new technology is transforming dairy farming. (2018). Retrieved June 24, 2019, from http://www.megalac.com/about/news/151-how-new-technology-is-transforming-dairy-farming

Islam, N. & Want, R. (2014). Smartphones: Past, present, and future. *IEEE Pervasive Computing*, *13*(4), 89–92. doi:10.1109/MPRV.2014.74

Jayaraman, P., Yavari, A., Georgakopoulos, D., Morshed, A., & Zaslavsky, A. (2016). Internet of things platform for smart farming: Experiences and lessons learnt. *Sensors*, *16*(11), 1884.

Jhuria, M., Kumar, A., & Borse, R. (2013). Image processing for smart farming: Detection of disease and fruit grading. In *2013 ieee second international conference on image information processing (iciip-2013)* (pp. 521–526). IEEE.

Kalaiselvi, P. & Nithya, S. (2013). Face recognition system under varying lighting conditions. *IOSR Journal of Computer Engineering (IOSRJCE)*, *14*(3), 79–88.

Karpathy, A. & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *The ieee conference on computer vision and pattern recognition (cvpr)*.

Kim, J. H., Aulck, L., Thamsuwan, O., Bartha, M. C., & Johnson, P. W. (2013). The effects of virtual keyboard key sizes on typing productivity and physical exposures. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *57*(1), 887–891. doi:10.1177/1541931213571193. eprint: https://doi.org/10.1177/1541931213571193

Kondratova, I. et al. (2004). Voice and multimodal technology for the mobile worker. *ITcon*, *9*, 345–353.

Krusienski, D. J., Grosse-Wentrup, M., Galán, F., Coyle, D., Miller, K. J., Forney, E., & Anderson, C. W. (2011). Critical issues in state-of-the-art brain–computer interface signal processing. *Journal of Neural Engineering*, *8*(2), 025002. doi:10.1088/1741-2560/8/2/025002

Kumar, S., Singh, S. K., Singh, R. S., Singh, A. K., & Tiwari, S. (2017). Real-time recognition of cattle using animal biometrics. *Journal of Real-Time Image Processing*, *13*(3), 505–526. doi:10.1007/s11554-016-0645-4

Lamkin, P. (2018). Smartwatches to dominate wearable tech - double digit growth forecast for industry. Retrieved July 1, 2019, from https://www.forbes.com/sites/paullamkin/2018/12/19/smartwatches-to-dominate-wearable-tech-double-digit-growth-forecast-for-industry/#7cc11b911a4b

Laws, F. (2012). Smartphones and apps taking agriculture by storm. Retrieved June 24, 2019, from https://www.farmprogress.com/equipment/smartphones-and-apps-taking-agriculture-storm

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lin, F. & Ye, W. (2009). Operating system battle in the ecosystem of smartphone industry. In *2009 international symposium on information engineering and electronic commerce* (pp. 617–621). IEEE.

Lotlikar, T., Kankapurkar, R., Parekar, A., & Mohite, A. (2013). Comparative study of barcode, qr-code and rfid system. *Int. J. Comput. Technol. Appl. 4*(5), 817–821.

Lv, X., Zhang, M., & Li, H. (2008). Robot control based on voice command. In *2008 ieee international conference on automation and logistics* (pp. 2490–2494). IEEE.

Merriam-Webster.com. (2019). Retrieved June 24, 2019, from https://www.merriam-webster.com/

Mittal, Y., Toshniwal, P., Sharma, S., Singhal, D., Gupta, R., & Mittal, V. K. (2015). A voice-controlled multi-functional smart home automation system. In *2015 annual ieee india conference (indicon)* (pp. 1–6). IEEE.

Morgan, T. (2004). *Guide to good dairy farming practice*. International Dairy Federation, Food, and Agriculture Organization of the United Nations. Retrieved from http://www.fao.org/3/Y5224E/y5224e00.htm

Namiot, D. (2015). On indoor positioning. *International Journal of Open Information Technologies*, *3*, 23–26.

NASA. (2013). Head-mounted display and wired gloves, ames research center.jpg. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=752240

Neményi, M., Mesterházi, P., Pecze, Z., & Stépán, Z. (2003). The role of gis and gps in precision farming. *Computers and Electronics in Agriculture*, *40*(1-3), 45–55.

Nevogt, D. (2018). 4 ways digital tools are revolutionizing construction. Retrieved July 8, 2019, from https://www.constructconnect.com/

blog/construction-technology/4-ways-digital-tools-revolutionizing-construction/

Ng, A., Annett, M., Dietz, P., Gupta, A., & Bischof, W. F. (2014). In the blink of an eye: Investigating latency perception during stylus interaction. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1103–1112). ACM.

Nikhilb239. (2018). Nokia 130.jpg. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=36427362

Panetta, K. (2018). 5 trends emerge in the gartner hype cycle for emerging technologies, 2018. Retrieved June 24, 2019, from https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/

Poretsky, S. (2012). Advantages & disadvantages of printers. Retrieved June 24, 2019, from https://yourbusiness.azcentral.com/advantages-disadvantages-printers-4798.html

Pros and Cons of Wearable Technologies. (2017). Retrieved June 24, 2019, from https://www.clodoc.com/blog/pros-and-cons-of-wearable-technologies/

Rao, R. P. (2019). Towards neural co-processors for the brain: Combining decoding and encoding in brain–computer interfaces. *Current opinion in neurobiology*, *55*, 142–151.

Robles-De-La-Torre, G. (2006). The importance of the sense of touch in virtual and real environments. *IEEE MultiMedia*, *13*(3), 24–30. doi:10.1109/MMUL.2006.69

Ruan, S., Wobbrock, J., Liou, K., Ng, A., & Landay, J. (2016). Speech is 3x faster than typing for english and mandarin text entry on mobile devices.

Saxena, S. & Kwon, H. (2012). Tizen architecture. In *Tizen developer conference, san francisco, california*.

Seiko Epson Corporation, (2019). Ep-101. Retrieved June 24, 2019, from https://global.epson.com/company/corporate_history/milestone_products/pdf/04_ep-101.pdf

Shamshiri, R. R., Weltzien, C., Hameed, I. A., Yule, I. J., E Grift, T., Balasundram, S. K., … Chowdhary, G. (2018). Research and development in agricultural robotics: A perspective of digital farming.

Shibata, T. (2002). Head mounted display. *Displays*, *23*(1), 57–64. doi:https://doi.org/10.1016/S0141-9382(02)00010-0

Shurkhay, V., Alexandrova, E., Goryaynov, S., & Potapov, A. (2015). The current state of the brain–computer interface problem.

Silfverberg, M., Scott MacKenzie, I., & Korhonen, P. (2000). Predicting text entry speed on mobile phones. (pp. 9–16). doi:10.1145/332040.332044

Silva, D. (2019). Plasma tv basics. Retrieved June 24, 2019, from https://www.lifewire.com/guide-to-plasma-tvs-1847857

SimonWaldherr. (2011). Wikipedia qr-code.svg. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=16531652

smartwatchspex.com. (2019). Dz09 smartwatch. Retrieved June 28, 2019, from https://www.smartwatchspex.com/dz09-smartwatch-full-specs/

Spurrier, C. (2007). Styluses.jpg. Retrieved June 24, 2019, from https://commons.wikimedia.org/w/index.php?curid=2000599

Statista Research Department, (2015). Physicians' usage of smartphones for professional purposes in the u.s. from 2012 to 2015. Retrieved June 24, 2019, from https://www.statista.com/statistics/416951/smartphone-use-for-professional-purposes-among-us-physicians/

Statista Research Department, (2016). Number of smartphone users worldwide from 2014 to 2020 (in billions). Retrieved June 24, 2019, from https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

stuntdouble. (2014). [guide] galaxy gear working with motorola moto g. Retrieved June 28, 2019, from https://forum.xda-developers.com/smartwatch/samsung-galaxy-gear/galaxy-gear-motorola-moto-g-t2844391

Talbot-Smith, M. (2012). *Audio engineer's reference book*. Focal Press.

Tewari, G. & Juneja, V. K. (2007). Advances in thermal and non-thermal food preservation. (First edition, pp. 3, 96, 116). Blackwell Publishing.

Thrasher, J. (2013). Rfid vs. nfc: What's the difference? Retrieved June 24, 2019, from https://blog.atlasrfidstore.com/rfid-vs-nfc

Tomanek, A. (2018). The dz09 smartwatch with the initial screen open. Retrieved September 16, 2019, from https://commons.wikimedia.org/wiki/File:DZ09_smartwatch_with_home_screen_open.jpg

Varnum, G. (2018). Amazon echo dot june 2018. Retrieved September 17, 2019, from https://commons.wikimedia.org/wiki/File:Amazon_Echo_Dot_-_June_2018_(1952).jpg

Vate-U-Lan, P., Quigley, D., & Masouras, P. (2017). Smart dairy farming through internet of things (iot). *Asian International Journal of Social Sciences*, *17*, 23–36. doi:10.29139/aijss.20170302

Vining, K. C. (1990). Effects of weather on agricultural crops and livestock: An overview. *International Journal of Environmental Studies*, *36*(1-2), 27–39. doi:10.1080/00207239008710581. eprint: https://doi.org/10.1080/00207239008710581

Watkins, D. (2019). Strategy analytics: Samsung's tizen os leads global smart tv market despite challenge of non-proprietary alternatives. Retrieved June 24, 2019, from https://news.strategyanalytics.com/press-release/intelligent-home/strategy-analytics-samsungs-tizen-os-leads-global-smart-tv-market

Williams, E. (2018). A day in the life of a dairy cow. Retrieved June 24, 2019, from https://www.drink-milk.com/a-day-in-the-life-of-a-dairy-cow/

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big data in smart farming–a review. *Agricultural Systems*, *153*, 69–80.

Yue, J., Li, Z., Liu, L., & Fu, Z. (2011). Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling*, *54*(3), 1121–1127. Mathematical and Computer Modeling in agriculture (CCTA 2010). doi:https://doi.org/10.1016/j.mcm.2010.11.044