

Faceboard: Hands-free Text-Entry based on head movement

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Mario Schwarz

Registration Number 01425172

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig

Assistance: Dipl. Ing. Richard Schlögl

Dipl. Ing. Christoph Wimmer

Vienna, 21st May, 2025

Signature Author

Signature Advisor



Faceboard: Hands-free Text-Entry based on head movement

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Mario Schwarz

Registration Number 01425172

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Advisor: Thomas Grechenig

Wien, 21st May, 2025

Erklärung zur Verfassung der Arbeit

Mario Schwarz

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 21. Mai 2025

Mario Schwarz

Acknowledgements

I would like to express my gratitude to everyone who supported me during my study.

First and foremost, I thank my family and friends with their encouragement, the discussions, the motivation and their valuable time.

I am very grateful to all participants of the study - friends, family, colleagues from work and total strangers met at the climbing gym - enabling me to write this thesis.

Last but not least I would like to thank my supervisors for their patience and support.

Abstract

The introduction of smartphones allows billions of people to enter text on the go. However, users with limited hand functionality face challenges when entering text, as current solutions usually require additional hardware such as eye-tracking devices. This study explores a novel and hands-free input method for smartphones utilizing the camera of the smartphone. By using machine learning algorithms, head movements and blinking are processed and mapped onto the screen in real-time to allow swipe based input without the need to touch the phones' screen.

Evaluated by 16 participants, the average user managed to input text with 5.98 words per minute and a keystrokes per character value of 1.67. Other text-entry methods like EyeSwipe proposed by Kurauchi et al. [KFJ⁺16] or Tiltwriter by Castellucci et al. [CMM⁺19] achieved average speeds of 11.7 and 12.1 words per minute, which indicates room for improvement.

Qualitative feedback was collected via System Usability Scale (SUS) and NASA-TLX, Faceboard showed an average SUS score of 59.1 and an average weighted TLX workload of 58. Compared to the generally accepted values of 68 for SUS and an averaged weighted workload of 42 for TLX, usability and workload were problems for participants [SotScho11] [Eu21]. Participants showed the need for improvement in reducing physical and mental strain. Possible future work aims to enhance usability by making parameters configurable and introducing dwell-time selection instead of blinking.

Keywords: *Mobile Devices, Input Techniques, Accessibility, Head-based-Pointing, Human-Centered Design*

Contents

Abstract	ix
Contents	xi
1 Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.3 Areas of Application	5
1.4 Methodology (Overview)	6
1.5 Structure	7
2 Related Work	9
2.1 Medical Background	9
2.2 Mobile Devices Accessibility	11
2.3 Typing	12
2.4 Gesture-based input	13
2.5 Gaze-based input	15
2.6 Dwell-Time input	16
2.7 Google GameFace	19
2.8 Neural Based Input	19
3 Implementation	21
3.1 Faceboard - Android Accessibility Application	21
3.2 Hyper Typer	31
3.3 Data Backend	31
3.4 Functional User Tests	32
4 Methodology	37
4.1 Research Approach	37
4.2 Data Collection	38
5 Study Design	47
5.1 Procedure	47
5.2 Preparations	56
	xi

5.3	Participants	60
6	Evaluation	63
6.1	Hypotheses	63
6.2	Quantitative Evaluation	65
6.3	Questionnaires	66
7	Results	67
7.1	Results - Faceboard	67
7.2	Results - Usual Input Method	72
8	Questionnaires	75
8.1	System Usability Scale - SUS	75
8.2	NASA TLX	78
8.3	Additional Feedback	80
9	Discussion	83
9.1	Influence of Short vs. Long Words	85
9.2	Possibilities and Struggles	85
9.3	Accessibility and Practical Constraints	85
9.4	Data quality	86
9.5	Participant Selection	86
9.6	Participant Feedback	86
10	Conclusion	87
10.1	Future Work	89
10.2	Design Optimizations	90
10.3	Extended User Studies	91
	List of Figures	93
	List of Tables	95
	List of Algorithms	97
	Data	99
	Participation Notes	99
	Moderator script	136
	Bibliography	141

CHAPTER 1

Introduction

With 1.37 billion smartphones sold in the year 2020 alone, it is an integral part of daily life [Cel]. According to penetration rates, almost 50% of the population is using smartphones [Sma]. With the rapid expansion of smartphones as a communication tool, texting quickly turned into one of the most important ways of communication [LLCP10]. The predominant ways to create input on handheld devices are the on-screen keyboard using fingers and voice-input [SC15]. Using finger-based input on small touchscreens can be cumbersome or even impossible for people with tremors or lack of fine motor skills [CSC⁺13]. Due to this bias, people with limited possibility of using a smartphone with their hands are significantly deprived when it comes to broadly available solutions to input text on their devices.

There exist several solutions for hands-free input but most of them are restricted to proprietary eye-tracking devices like Tobii EyeX eye-trackers¹ as used in works by Kurauchi et al. and Špakov et al. [KFJ⁺16] [M14]. They do offer great performance but they do need additional hardware to work.

Taking a look at other research areas like virtual reality shows different input approaches like head-based pointing [YGY⁺17]. Here, the users are using a head-mounted display with the utilization of sensors and cameras to track movement and allow precise control of the cursor to write on a virtual keyboard.

Studies have shown that for people with motor impairment, traditional input methods are not performing very well. Riviere and Thakor evaluated the accuracy and linearity of disabled or elderly people with computer mice [RT96]. They show that with age and impairment, the accuracy and linearity of input is decreasing. Wiegand and Patel evaluated the impact of motor impairment on touch input methods [WP15]. They showed that impaired users create a lot of unwanted inputs near the edges of the touchscreen.

¹<https://www.tobii.com/solutions/cognitive-and-psychological-research>

Some participants also had problems reaching the corner opposite to their used hand which is problematic when trying to reach the letters on the touchscreen keyboard. Irwin et al. showed a decrease in speed in tapping tasks for motor-impaired participants compared to non-impaired participants [IS12]. Their study also showed that impaired people apply more pressure than necessary to the touch device which can cause harm to the human musculoskeletal system leading to musculoskeletal disorders when touch systems are used regularly.

Using head-based pointing as often used in virtual reality setups without the proprietary hardware needs would allow for a new possible input method for smartphones. Utilizing the computing power of modern smartphones by using machine-learning algorithms instead of expensive external sensors could enable a new way of head-based input to broaden availability.

With that in mind, a new way to input text on Android² smartphones called 'Faceboard' was created and evaluated. This new input technique utilizes machine-learning algorithms to gather information about users' head positions by utilizing the smartphones' front-facing camera. A cursor is shown on the smartphones' screen which maps the users' head positions in the real world onto the screen of the phone. Users can move their heads left, right, up, and down, which is interpreted by Faceboard and translates the position of the cursor on the Android device accordingly in real time. Leveraging the machine-learning algorithm, blinking is recognized via the front-facing camera. When users blink, Faceboard starts to record a gesture, meaning every movement of the cursor is tracked but not yet executed. Once they blink again, the recordings stop and the recorded gestures are mapped onto the device's screens like a touch input. By doing so, the simulated touch input can generate text if the virtual gesture is on a keyboard. Using these movement paths, the device becomes more accessible for people with disabilities such as shaky fingers, multiple sclerosis, or any other condition preventing them from using their hands for input.

1.1 Motivation

The motivation to introduce new input techniques is twofold. On the one hand, expanding possible ways for motion-impaired people to input text allows more people to participate in the more and more relevant digital ecosystem that smartphones created. On the other hand, exploring how emerging technologies like machine learning can help people in need of assistive technology is an exciting field of work. Reaching accessibility by incorporating existing sensors like the front-facing camera which is built-in in nearly every modern smartphone enables more inclusive mobile device experiences.

With the proposed technique, the study tries to lower the entry barrier for finger-less text entry by just utilizing the users' front camera as a sensor. Since the penetration rate on smartphones is high it is expected to be usable for a lot of people without the

²Android is a mobile operating system designed for touchscreen devices.

needs to purchase or carry around additional hardware. Previous work like EyeSwipe proposed by Kurauchi et al. focused on gaze-based input using specialized eye-tracking devices [KFJ⁺16]. On average, input methods utilizing eye-tracking algorithms made for motor impaired persons achieve good performance with over 10 words per minute according to Polacek et al. [PSS17].

1.2 Objectives

For the study, an interactive proof of concept app for Android was created to prove the feasibility of the above mentioned algorithms. This app uses modern machine learning algorithms to recognize users eyes, faces, tilts as well as the movement of their heads. Complexity-wise, the challenge of this concept is to reliably map users head movements to input on the keyboard as a swipe gesture without jitter and noise to reduce the amount of wrong input but still allow cursor movement in real-time. The Android app was designed to work as an accessibility service in the background. A small test was conducted on a range of devices to gather information on how it works on different devices.

This thesis aims to study the possibilities of this input type and whether it is reliable enough to use daily. To prove these points, the proof of concept app was compared to other known input variants on mobile phones.

In order to thoroughly verify that Faceboard works, a user study was conducted. To prove that, Hyper Typer, a game created by Schlögl et al. [Sch20], was played by the participants of the user study. This game allows performance measurements of text input by having the goal to transcribe phrases as fast and error-free as possible while calculating performance metrics like words per minute or keystrokes per character in the background.

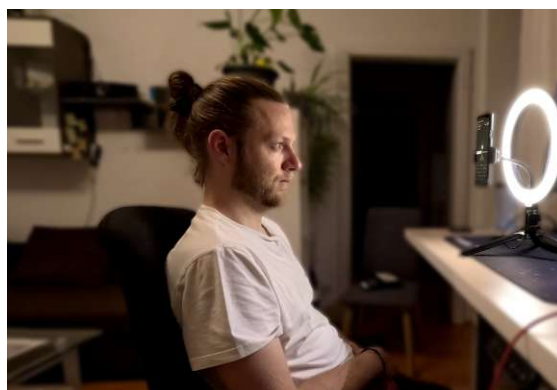


Figure 1.1: One of the participants during the experiment

With the objectives in mind, the following research goals were set:

- RG1** Develop a functional proof of concept application to demonstrate the technical feasibility of the proposed input method.
- RG2** Conduct an experiment using the developed prototype and the users' standard input method to assess the practical usability in a comparable manner.

To evaluate whether these research goals were reached and to assess the usability of the developed method, the following hypotheses were formulated:

- H1.1** Text entry speed (measured in words per minute, WPM) differs between the prototype and users' usual input method.
- H1.2** Text entry efficiency (measured in keystrokes per character, KSPC) differs between the prototype and users' usual input method.
- H1.3** Text entry error rate (measured as Corrected Error Rate) differs between the prototype and users' usual input method.

To examine the impact of a possible learning effect, additional hypotheses were proposed:

- H2.1** Text entry speed improves for users that repeated the experiment with the prototype.
- H2.2** Text entry efficiency improves for users that repeated the experiment with the prototype.
- H2.3** Text entry error rate decreases for users that repeated the experiment with the prototype.

For further analysis of usability, non-functional requirements were formulated:

- NFR1** The users are able to set up the service on their own.
- NFR2** The users are able to learn the newly proposed method by playing a tutorial

1.3 Areas of Application

The focus of this thesis lies on the control-flow of entering text via an Android smartphone. If the applications usability is feasible, the concept of this app could be extended to control the whole operating system.

The main area of application is assistive technology for people who have problems using their hands due to motor impairment. Reasons for the need for assistive technology vary based on the person's condition. Even though there are already lots of different input methods for motor-impaired persons, not every input method is suitable for every condition. In an overview by Polacek et al., they provide a comparison between 150 different text-entry methods for people with motor-impairment [PSS17]. With 16% of the world's population suffering from some kind of major disability, the variety of needs is high [WHO23]. A voice-based text-entry system might be sufficient for someone who is paralyzed but might not be suitable for people with speech impairment. Different physical ability ranges can be categorized in 7 different categories (adapted from Polacek et al. [PSS17]):

1. People with locked-in syndrome
2. People capable of using single switch interfaces
 - a) by blinking
 - b) by facial muscle contractions
 - c) by a push button
3. People able to use an eye tracker
4. People able to use a nonverbal vocal input
5. People able to use their speech
6. People able to use stationary pointing devices, such as trackballs, joysticks, or 4-way arrow keyboards
7. People with reduced skill of the upper limbs using dedicated keyboards with a reduced number of keys

Since the algorithm proposed requires people to blink and move their head, it is most suitable for people of category 7. The algorithm is device-independent and only needs a camera as hardware to be used on every possible device which allows third-party programs to be installed. So the porting on other devices should be no problem and the areas of application are broadened.

1.3.1 Beyond Text-entry

Due to being implemented as a system-wide accessibility-service, the application can be extended to overall control the whole Android operating system hands-free. By this, it would be possible to implement new gesture types to allow other control-flows to be executed via head movements like starting apps as well as using them.

With the emerging technology of augmented- and virtual reality, another possibly interesting area of application is possible. Even though gesture-based typing like Yu et al. proposed already exists [YGY⁺17], Faceboard could be an addition to be incorporated into VR and AR Headsets. Wearable computers like smart glasses could be another possible application. Head-movement could be recorded with more precision using more sensors and cameras which might allow more precise input and higher input rates. By that, the gesture recording method of Faceboard could be used with more precision and could allow faster typing for AR and VR applications.

Faceboards' head-movement and blinking framework might even be applied in *rehabilitation and therapy programs*. Using the gamification approach of Hyper Typer and Faceboards' head-movements-based input, it could be used as some form of physiotherapy for people as proposed by Baranyi et al. in 2014 [BRL⁺14]. Controlled head movements and blinking gestures embedded into a serious game could motivate and empower users to perform necessary exercises.

1.4 Methodology (Overview)

In order to check the feasibility of the proposed input method, a proof of concept Android application was created as part of this study. With prototyping and pilot tests, the usability was increased before the actual experiment. Then, a user study with 16 participants was conducted. Utilizing head movements and blinking, the participants of the study transcribed phrases in a controlled laboratory experiment. The phrases consisted of predefined words to ensure equal distribution of letters and better comparability between the participants. This setup allowed to assess the performance and usability of Faceboard. As a baseline, the participants also transcribed phrases with their usual input method. This allowed a comparison between the participants' usual input method and the newly proposed method.

By automatic data collection during the transcription phase, quantitative analysis of efficiency, accuracy, and usability was possible. To ensure the statistical significance of the calculated statistics, comparative as well as correlation and regression analysis were used.

Each participant ran through the transcription phase twice to find possible learning effects.

For a broader picture, questionnaires and a concluding brief interview were conducted after the experiment. The System Usability Scale and NASA-TLX were used as well as brief questions about the overall experience. The qualitative insight on workload and usability was then combined with the quantitative performance data to gather deeper insight.

1.5 Structure

This thesis is divided into 10 chapters with each focusing on different aspects of the study.

In **chapter 2 - 'Related Work'**, existing hands-free text-entry methods like gesture-based, gaze-based, and other alternative input techniques are discussed to provide the needed background on text-entry methods. Medical conditions making alternative input techniques necessary are introduced and explained. Afterward, a gap in the current state of the art that Faceboard can fit in is shown by comparing the different available technologies.

Chapter 3 - 'Implementation' focuses on the technical implementation of Faceboard: The head movement, as well as the blinking recognition, is discussed and the machine-learning approach is presented. Details on how the Android accessibility API works and how it is used to transfer gestures onto the display are provided. Data collection and processing are explained.

The **chapter 4 - 'Methodology'** then defines the collected data like metrics in detail and how it is analyzed after collection. The chapter provides theoretical information on why transcription tasks are used to evaluate Faceboard and the questionnaires used to gather qualitative feedback.

The **chapter 5 - 'Study Design'** takes a deep dive into the demography of the participants as well as their recruitment, the study setup and a step-by-step plan of the user-study for comprehensibility.

The **chapter 6 - 'Evaluation'** is about how the data is analyzed and methodologies are used. Statistical analysis is defined to find the first indicators of the usability of the newly proposed approach. Performance metrics like words per minute (WPM) are analyzed to measure the efficiency of the participants. Methods for statistical significance by usage of comparative as well as correlation and regression analysis are defined.

Finally, in **chapter 7 - 'Results'**, the methods from the chapter Evaluation are applied and statistics are derived from the collected data. Correlations between participants first and second sessions are concluded. Users are compared to each other and possible similarities are discussed. Hypotheses are checked against the results to see if the set research goals were met.

The feedback from the questionnaires as well as the informal interview gives deeper insight into participants' usage of Faceboard in **chapter 8 - 'Questionnaires'**. Standardized evaluation methods like the System Usability Scale (SUS) and NASA-TLX are used to provide valuable feedback from the participants regarding usability and workload.

Chapter 9 - 'Discussion' covers the findings from the chapters 'Results' and 'Questionnaires' and discusses it in regard to related work. Strengths and weaknesses of Faceboard are highlighted, reasons for observed patterns are explored.

The final **chapter 10 - 'Conclusion'** summarizes the problem statement, methodology, and results of this work. In summary, the conclusion shows that the concept is working with acceptable usability but with the need to improve the physical strain of the blinking algorithm. Lastly, a forecast on possible future work regarding the proposed proof of concept is given. This includes theoretical evaluation as well as technical improvements suggested by the participants as well as found during interpretation of the results.

Related Work

In our interconnected world, smartphones are omnipresent: From online payments via credit card, buying a ticket for public transport, or a simple message to a friend: a lot of processes nowadays rely on the usage of a smartphone. Most people might find entering text via touchscreens easy and natural, but for people with motor impairments, these processes can be cumbersome since interacting with standard input devices like touchscreens and keyboards can be difficult [SYF07] [PSS17]. According to the WHO, 16% of the world's population suffers from a significant disability [WHO23]. Physical impairment has varying degrees of severity and can influence a person's possibilities differently. To gather insight on medical conditions relevant to this work and how Facebook could be applied, a medical background is created in the upcoming sections.

2.1 Medical Background

In the context of this work, motor impairment is the most relevant physical impairment. Rosenbaum defined a Motor Impairment as a type of condition where a person loses partial or full muscle function [Ros09]. Movement is a complex system with interactions between the brain, muscles, and limbs. As Jacko et al. state in their 'review and reappraisal of technologies for individuals with disabilities', five major types of physical impairment have a strong influence on information technology usage: musculoskeletal disorders, cerebral palsy, neuromuscular diseases, seizure disorders, and injuries [JV01].

2.1.1 Musculoskeletal disorders

Musculoskeletal disorders are disorders concerning the musculoskeletal system. The most common of them are types of arthritis, which are inflammations regarding the joints of the human movement apparatus. Due to the aging of the population and increasing obesity, arthritis like osteoarthritis is expected to keep growing in count [VFN⁺12] .

Inflammations in the joints of the finger can render finger-based input for an affected person unusable. Head-based input like the one proposed with Faceboard could allow users to control the phone regardless of pain in the fingers.

2.1.2 Cerebral Palsy

Cerebral palsy describes disorders created by brain injury which mostly occur during birth or in the first 5 years of life [VDB20]. Depending on the affected part of the brain, predominant clinical features are poor balance, sensory deficits as well as disorders of movement [O'S08]. The aging process can affect our ability to move as well [WBPV10].

The limitations of Cerebral Palsy are highly variable. In consideration of the dysfunctional part of the brain and also the severity of the disorders, Faceboard could help people with slight restrictions created by cerebral palsy.

2.1.3 Neuromuscular Diseases

Neuromuscular diseases are disorders regarding nerves. Symptoms can highly vary in extent but have a lot in common: for diseases like multiple sclerosis or Parkinson's disease, it can influence people by weakening them, numbing fingers/toes, making them spastic or/and giving them cognitive defects [Lub05] [BOK21][FGRL14].

People suffering from neuromuscular diseases can benefit from Faceboard. Again, depending on the symptoms, finger tremors or stiffness of muscles can make it hard for smartphone users to write on on-screen keyboards. For Parkinson's disease, tremors are not typical, but still, 80% of all affected persons have a kind of tremor [BOK21]. For those, Faceboard could be an alternative way to input text.

2.1.4 Seizure Disorders

A seizure is a sudden, mostly uncontrollable movement, loss of consciousness, or loss of ability to move and lose awareness of the surroundings. These symptoms are caused by abnormal function of brain nerves [McI13]. People suffering from seizures are not directly the target group of Faceboard, but still can use it.

2.1.5 Injuries

Injuries can highly affect a person's motor abilities. Usually resulting from acute trauma, injuries can turn into chronic disabilities if not treated right. Depending on the grade of impairment, spinal cord injuries for example can cause different effects from sensory issues to restricted or motor function at all [AWN⁺17]. Depending on where the spinal cord is injured, people may be able to control the movement of the head, but not the limbs. Concerned parties may therefore have difficulties controlling technical systems that need touch input.

Fine motor control needed when operating a touchscreen, mouse, or keyboard, often renders a significant barrier for impaired individuals. As highlighted by Fager et al., limitation in movement precision is one of the greatest problems in accessibility work to be overcome [FFOJB19]. Tremors and lack of precision impact users with disabilities in being unable to click on small buttons. Frameworks like Nomon evaluated by Bonaker et al. which offers a flexible text-entry interface for single switch users with noisy input show improvement in input rates compared to earlier approaches [BNVB22].

With the introduction of more powerful smartphones with accurate sensing technologies, the detection of accidental or unintentional movements improved. For some specific conditions, there have been a lot of studies that focus on providing support, like Bächlin et al. propose a wearable assistant for patients with a special form of Parkinson's disease [BPR⁺10]. Technological solutions like the one proposed by Bächlin et al. emphasize the importance of assistive technologies to allow people suffering from impairment to act more independently.

People with injuries are an interesting group of possible users. Following acute injury, a person might not be able to use their usual input method for a limited period of time. Being used to utilizing a smartphone for communication, injured people might suffer from the fact that they are temporarily restricted from smartphone usage. Here, the injured party can profit from Faceboard as a temporary alternative to their usual input method.

Also, there is a possibility that Faceboard can be used in rehabilitation and therapy programs as a sort of physiotherapy. Patients are encouraged to carry out their exercises with serious games [BRL⁺14]. Faceboard in combination with Hyper Typer could be adapted and act as a serious game.

2.2 Mobile Devices Accessibility

In general, mobile devices have undergone a transition from physical buttons to touchscreen-based interfaces. With this shift, problems for individuals with disabilities regarding vision, motor, and nervous systems have been introduced. Fine motor control is required to press on the right position of the screen. With these downsides of touch-focused devices, manufacturers had to make strides on the software side to improve accessibility for an ever-growing base of users with disabilities. For example, Google created the Android Accessibility Suite¹. Part of it is 'TalkBack'², which allows people who have bad vision or are completely blind to interact with their Android devices using spoken feedback. Here, the users can use simple left and right gestures and double-tap on the screen to select different elements on the screen. Those elements then are synthesized by text-to-speech.

For people with limited motor function or no ability to move at all, smartphone producers

¹<https://www.google.com/accessibility/products-features/>

²<https://support.google.com/accessibility/android/answer/6007100>

Apple and Google created voice control systems^{3,4}. Those systems enable the possibility to completely control the device using voice commands.

Solutions like these allow users with specific needs to be included in a highly digitized world. The reason a person needs device accessibility highly varies based on the condition the person suffers from. Even though Google and Apple did important work on including Accessibility Services per default on their devices, it will not fit every condition. An input method might be suitable for one person's condition but rendered completely unusable for another. Social patterns of people change due to mobile devices and a lot of communication is done virtual instead of physical face-to-face conversations. This hardens the need for mobile device accessibility to allow people with impairment to take part in all parts of social life. For text-entry systems, a lot of systems have been created to allow people to find their best working input type. Input techniques for mobile devices can be roughly divided into the following types: Typing, Gestures-based input like Swiping, Gaze-based input, dwell-time-based input, and neural communication-based input. Some concepts combine several techniques into one approach to mitigate disadvantages.

Facebook uses swiping as the underlying technique for entering text. Incorporating swiping gestures to input text via head movement should increase input rates compared to simple dwell-time typing or other typing-based algorithms that are based on head movement.

2.3 Typing

By Typing, this study refers to pressing keys to enter text. Virtual as well as physical keyboards usually have a key for each letter and allow fast typing and high accuracy when users are used to typing. The act of typing requires hand- and finger mobility, which could be a usage barrier for users with motor impairments like hand tremors [NJ12]. On the other hand, it allows high input speeds for proficient users with high precision. The term 'typing' collects a variety of different techniques used to input text via keys such as hunt-and-peck (which refers to searching for a specific letter and pressing it with one finger), 10-finger-writing (using all ten fingers at once for faster typing), auto-correction (which allows for errors to be automatically corrected) and predictive text (using algorithms to find the most probable next words in the written text).

2.3.1 Physical to Virtual Keyboards

Translating to computer-based writing triggered by the spread of touch-screen based systems like PDA and later on smartphones, the physical keyboard got competition by virtual keyboards. Virtual keyboards are mostly based on the QWERTY-layout as their physical predecessors, still it can be challenging to type on since the size is a lot smaller compared to a physical keyboard. Researchers proposed alternatives to the QWERTY

³<https://www.android.com/accessibility/mobility/>

⁴<https://support.apple.com/en-us/111778>

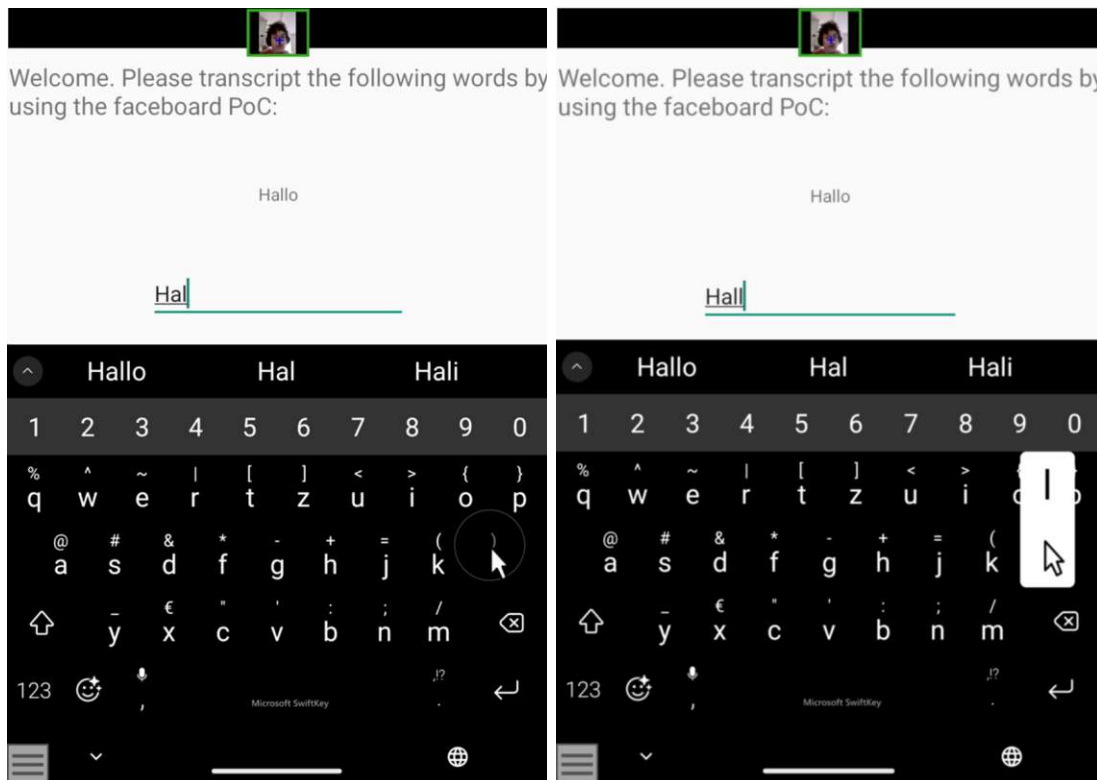


Figure 2.1: Transcribing the word 'Hallo' with EVA face mouse[Mau].

layout, the most widespread layout for on-screen keyboards as presented by MacKenzie et al. [MS02]. Still, QWERTY is the most used entry-method, even though MacKenzie et al. showed there are keyboards designed to have faster input rates compared to the QWERTY-layout.

Considering accessibility, users with fine-motor-skill limitations or tremors may struggle to press the small keys presented on the screen of the devices. Modern keyboards often incorporate features like auto-correction, spelling suggestions and predictive text.

2.4 Gesture-based input

Gesture-based input in general describes text-entry generated by movement. These technologies do not have a fixed set of symbols the users draw but rather interpret the users' input with an algorithm to match the movement to a certain character or word.

When text is entered via **virtual gestures** like swiping, the users drag their finger or a stylus across an onscreen keyboard to trace a path between each letter of the word they want to type.

Cirrin, a 'word-level unistroke keyboard for pen input' as presented by Mankoff et al. [MA98] uses gesture-based circular input to create words. Here, the users draw single strokes and hit the characters they want to write in the corresponding order of the word (See figure 2.2).

Gesture-based text entry method often use predictive algorithms to interpret the drawn gesture and by that determines the most probable word. Fitrianie et al. for example enhanced the mentioned Cirrin keyboard with predictive language-features [FR07]. It allows fast text input and users do not have to be as accurate as compared to typing. Compared to some other input methods, it can be rather complicated for some users, and swiping highly depends on dictionary data in the background.

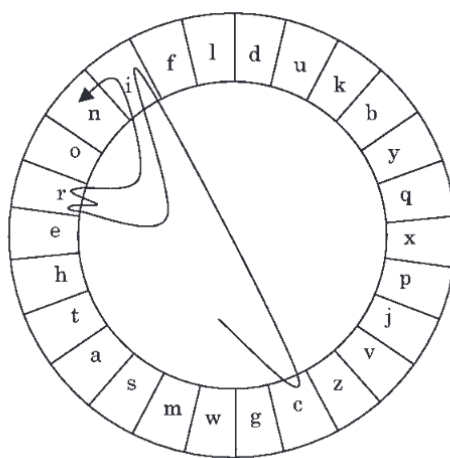


Figure 2.2: Writing the word 'cirrin' on the cirrin keyboard. Figure by MacKenzie et al. [MS02]

The most common gesture-based input for mobile devices would probably be **swiping a finger over a keyboard**. Most major on-screen keyboards like Google Keyboard⁵, SwiftKey⁶ or the standard iPhone keyboard⁷ support swiping. Here, the users place their finger on the first letter of the word, swipe their finger over the subsequent letters and finally lift off their finger at the word's end. This typically works as followed: The system records the fingers coordinates as the finger drags across the screen. With each letter passed, the keyboard temporarily stores the path. In the background, the keyboard has a dictionary with words. The keyboard processes the path created by users in real-time and calculates a probability for each letter. Now, the most probable word is calculated based on contextual cues like previous written words, spelling as well as the frequency of different words. By using probabilities, swiping has a higher error tolerance compared

⁵<https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin>

⁶<https://www.microsoft.com/en-US/swiftkey>

⁷<https://support.apple.com/de-at/guide/iphone/iph3c50f96e>

to typing and allows users to write less precisely since the path does not have to be exact. The algorithm proposed in this thesis is highly based on this method: It utilizes an Android smartphone with the SwiftKey keyboard installed and directly translate the head-movement onto the keyboard. By that, the keyboard recognizes as finger-drag and acts exactly as if users would use their fingers to swipe.

2.5 Gaze-based input

When talking about Gaze-based input, the focus lies on methods that track the users' eye movements to determine where they are looking on a screen. This gaze-point is interpreted as input which allows users to generate whatever input they need: typing, selecting, or simply interacting with the interface. To make the system work, one needs an eye-tracking device like those produced by Tobii or Irisbond. With an eye-tracking device, users can achieve fully hands-free input which makes it suitable for users with limited physical mobility like people suffering from locked-in syndrome as proposed by Bonaker et al. [BNVB22]. Gaze-based text input allows high precision with moderate input speed at very high accessibility since one only needs to use eye movement. On the downside, these systems are expensive, need additional hardware and setup to work properly and though are not usable for everyone.

2.5.1 EyeSwipe: Dwell-free Text Entry Using Gaze Paths

The algorithm proposed by Kurauchi et al. [KFJ⁺16] is a fast, gaze-based input method. Here, the users look at the keyboard to input text by swiping around with his/her gaze over the different letters. Similar to this approach, the word is created by moving the cursor over the single letters of the word he/she wants to type. The path then is computed to the nearest word (see figure 2.3). While Faceboard uses camera-based gesture paths, here the gazed path is derived from a proprietary Tobii EyeX eye-tracking device which allows more precise inputs by the users.

Kurauchi et al. compared their method with dwell-time input methods as proposed in the EVA Facial mouse. With each session held, the participants gained more confidence with the swipe-based input method.

Having a similar approach on how to enter text based on swiping gestures, made it a great candidate to compare with Faceboard regarding efficiency and usability. Regarding the words per minute metric, the last session already allowed an average typing speed of 11.7wpm for EyeSwipe compared to 9.5wpm for dwell-time input.

2.5.2 Fast Gaze Typing With An Adjustable Dwell Time

In the experiment conducted by Majaranta et al., they created a dwell-time-based input method utilizing a Tobii eye-tracking display [MAotSu09]. They allowed to adapt the dwell time during the experiment. By that, the participants of their study could increase their typing speed while typing and fit it to their speed.

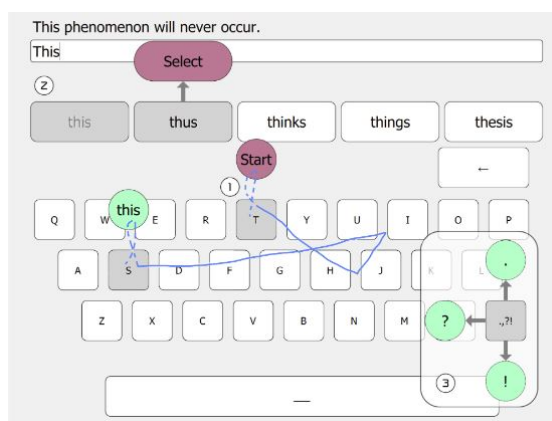


Figure 2.3: EyeSwipe as proposed by Kurauchi et al. [KFJ⁺16]. Figure by Kurauchi et al.

2.6 Dwell-Time input

Mostly used in connection with gesture-based input, dwell-time input needs the users to focus on specific points on the screen's keyboard for a short period of time (e.g. one second) until the system registers the input. This can be realized using a physical device to recognize the movement to a specific point or can utilize a movement recorded via camera. Dwell-Time input often is combined with gaze-based input.

Dwell-time input incorporates waiting to allow a specific action to start or end. This idling can influence the performance of systems like typing by lowering text-entry speed. Kurauchi et al. showed in their work that dwell-time had slower input rates compared to swiping [KFJ⁺16]. Facebook does not use dwell time to activate gestures but utilizes blinking detection to start and end a gesture.

2.6.1 EVA Facial Mouse

With over 5,000,000 installations via the Google Play Store, the EVA Face Mouse is a highly used Android app [Mau]. The app allows users to have a virtual mouse pointers similar to mouse-based operating systems which are controlled via head movements. As shown in figure 2.1, EVA Face Mouse utilizes the "hunt-and-peck". The interaction is based on point-dwell-click input, which makes it rather slow compared to this swipe-based approach for text input. However, EVA Facial Mouse is designed to allow the control of the whole operating system which could make it a good addition to swipe-based keyboard controls like the one proposed in this work.

2.6.2 Head-based Pointing On Smartphones For People With Motor Impairments

In Designing and Evaluating Head-based Pointing on Smartphones for People with Motor Impairments, Cicek et al. present a similar way to create input [CDF⁺20]. Here, they also present a hardware-free and highly available way to interact with an Android phone only utilizing the camera to create input. Since in their approach they are the same underlying machine-learning Kit 'ML Kit for Firebase', it makes it a great comparison in terms of speed and usability. The proposed algorithm uses the nose's position to be translated onto the screen.

Instead of swiping gestures as proposed in this work, the purpose of the implementation by Cicek et al. is to allow the users to point and click based on their head movements. Their functionality was limited compared complexity-wise, as the study participants only had to move the cursor to some points on the screen instead of drawing long gestures. Still, the work is notable due to using the same underlying technology and sharing some of the algorithmic work like edge-clipping.

Compared to the EVA Facial Mouse, the results regarding throughput are almost the same. As the EVA Facial mouse, the algorithm proposed by Cicek et al. focuses on clicking, instead of swiping. Again, this allows precise input, but at the cost of speed and effectiveness in terms of text input.

2.6.3 Swipe-like Text Entry By Head Movements And A Single Row Keyboard

Reducing complexity to a single-dimensional input is the proposed concept by Nowosielski et al. [Now17]. Similar to the previous proposals, the users generate input by moving their head, but instead of making use of horizontal and vertical input, only vertical input is processed. The keyboard consists of a single lexicographical sorted row of letters. The swiping is started automatically by moving the head either left or right. The users can now hover the cursor over the letters the word consists of. If a letter is selected is defined by a time-based threshold. Using their algorithm and prototype, participants of the study achieved 33.7 characters per minute on average.

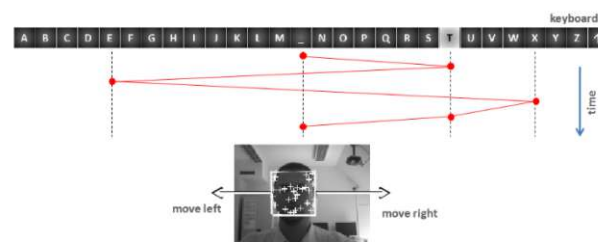


Figure 2.4: Moving the cursor to the selected letter and stopping there will accept the letter as input [Now17]. Figure by Nowosielski et al.

Having a similar approach by writing text with head movement and providing comparable performance measures, the work by Nowosielski et al. provides a good base to compare Faceboard with.

2.6.4 Tiltwriter

Tiltwriter describes another way of text input on mobile devices. Here, Castellucci et al. designed and evaluated no-touch keyboards for handheld devices based on tilting the phone [CMM⁺19]. The users can choose between a QWERTY-based layout as well as a custom ambiguous keyboard, similar to the T9 input known from feature phones. Once activated, a virtual tracking ball appears which can be moved over the keyboard. Dwelling over a specific key, users select letter after letter, the dwell time is configurable. After 10 sessions of testing the keyboard, the following performance results were gathered:

Keyboard mode	Words per minute	Error rate
QWERTY	12.1wpm	0.76%
CUSTOM	10.7wpm	0.62%

The algorithm allows rather low error rates compared with comparable efficiency. Also, the Tiltwriter keyboards' performance peaked with higher dwell times since the participants had less stress in selecting and generating less wrong input. In contrast to the proposed algorithm by Majaranta et al. [MAotSu09], Tiltwriter does not incorporate configurable dwell time. The participants achieved high performance with little frustration but rather high mental workload according to the overall NASA-TLX scores (see figure 2.5).

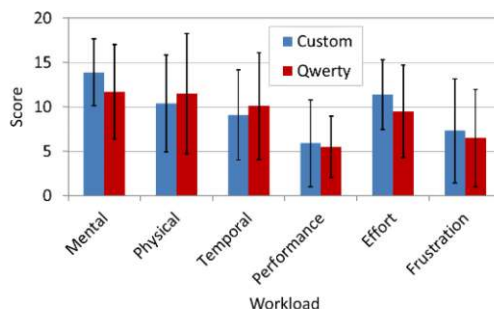


Figure 2.5: NASA-TLX scores of the evaluation of Tiltwriter. Figure by Castellucci et al.[CMM⁺19]

Tiltwriter is a touch-less input that utilizes movement like Faceboard, but has different target audiences since Tiltwriter is controlled by hand whereas Faceboard is solely controlled by face. Comparison between the two is still relevant since the approaches share cursor-based input and movement to write text.

2.7 Google GameFace

In May 2024, Google launched their GameFace project for Android [Gooa]. This application utilizes a similar approach as proposed here in this study but for a different use case. GameFace focuses on 'click' and 'drag' gestures but does not allow complicated gestures. GameFace allows face-based gestures to do simple tasks like scrolling from left to right by moving specified parts of the face. Since Google provided the source code for GameFace, it would make it a good starting point for extensions for Faceboard.

2.8 Neural Based Input

An early and experimental field of input methods is neural-based input methods. Here, text is entered via brain-computer interface as it first was proposed by Lal et al. in 2005 [LBS⁺05]. Here, the neural stimuli get measured either by using EEG electroencephalography (EEG) or magnetoencephalography (MEG) to interpret signals. With EEG being the easier method to measure the brain's stimuli, MEG allows better signal quality by measuring the brain's magnetic field. According to Lal et al., MEG allows better results for completely paralyzed persons. In the end, both technologies gather the same data with different quality [LBS⁺05].

2.8.1 A Brain Computer Interface with Online Feedback Based on Magneto-encephalography

Bacher et al. showed the potential of neural-based input with their work on a brain-computer interface. In the study, a woman with partial locked-in syndrome with an implanted microelectronic array in her motor cortex wrote text using a newly designed keyboard. Text is entered via the use of electroencephalography (EEG) and electrocorticography (ECoG) and a radial point-and-click keyboard. Using this it allows input-rates of about 10 keystrokes per minute [BJM⁺15].

Since it still is an early, highly scientific field, it is not available for most users but shows the possibilities of neural-based input for people with disabilities. The availability is what differs from Faceboard. With 10 keystrokes per minute, the brain-computer interface-based input method also is relatively slow.

Figure 2.6 shows the two possible ways to enter the text 'quick' using the brain-computer-interface proposed:

At 'A', the 'q' and the 'u' have already been selected (left screen - the 'MNOPQ' and 'RSTU' below the red star), now the user selects 'IJKL' to set the next list of predictions (middle screen) which now show 'quick' which can be selected (right screen)

At 'B', the participants selects the 'q' on the QWERTY keyboard (left screen) and selects the suggestion from the upper right corner (right screen).

2. RELATED WORK

Even though Facebook and brain-computer based input in general are only slightly related, both utilize a cursor to be controlled on-screen without touching the device. Also, algorithm-based word completion is utilized (see figure 2.6).

The review of relevant related work explored different approaches for alternative input methods. Most of the proposed methods on the one hand showcase promising performances but on the other hand, are not broadly available or require additional expensive hardware.

The analysis shows the gap Facebook aimed to address: Creating an affordable and hardware-independent but nevertheless performant text-entry method based on head-movement and gesture-based text-entry.

In the following chapter, the creation of the proof of concept app is explained by providing design principles in detail. The integration of machine learning and the algorithms behind the gesture drawing mechanism are illustrated. The difficulties with Androids' security model and accessibility services are explained.

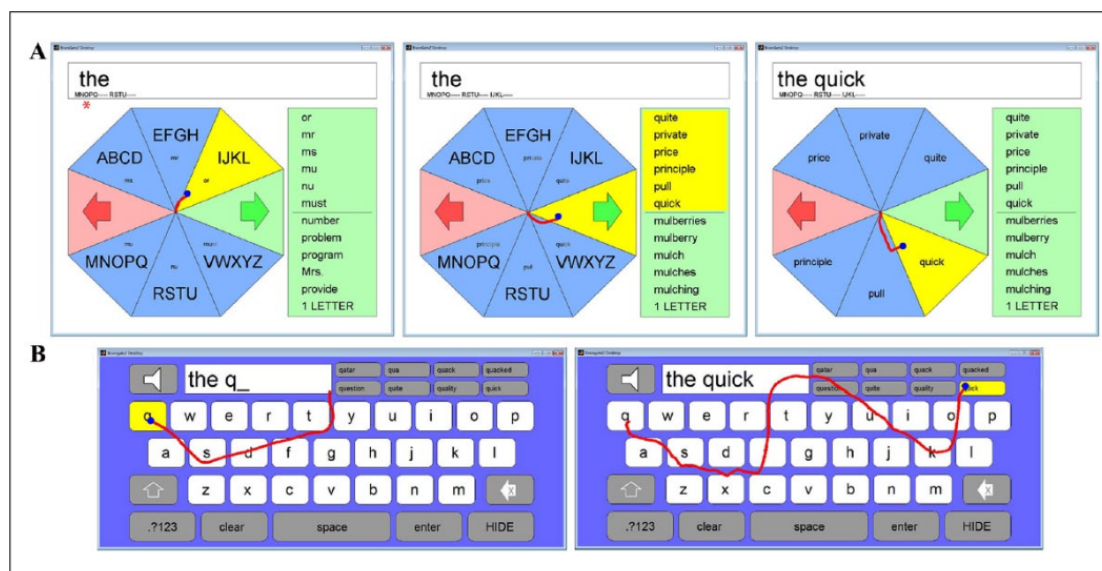


Figure 2.6: The figure shows the two possible ways to enter the text 'quick' using the brain-computer-interface proposed by Bacher et al. [BJM⁺15]. Figure by Bacher et al. [BJM⁺15].

CHAPTER 3

Implementation

The proof of concept consists of 3 separate apps: the 'Faceboard' Android app which is responsible for recording and drawing the gestures created with head movements, the 'Hyper Typer' Android app which we use for transcription- and scoring purposes and a Java-based server application, which communicates with the Hyper Typer app and is responsible for storing the gathered data for analysis.

3.1 Faceboard - Android Accessibility Application

In 2009, Android version 1.6 (Donut) introduced its first Accessibility Framework: This Framework allowed apps to interact with certain accessibility tools like the previously mentioned TalkBack screenreader. With Android version 8 released in 2017, the Android team introduced the possibility to generate complex gestures [and]. For this study, complex gestures are the main part used to insert and edit text.

The first component of the proof of concept app is the app 'Faceboard'¹. The Faceboard app is a Kotlin-based Android SDK 30 app (Android 11). This app is split into three main parts: the 'Head Movement Accessibility Service' (HMAS), the 'FaceCursorService' and the tutorial app.

¹The final implementation as well as the data gathered during the conducted studies can be found at <https://github.com/puiooo/faceboard>

3.1.1 'Head Movement Accessibility Service' (HMAS)

Android allows the creation of Accessibility Services, which are specialized components that help users to perform different actions. HMAS relies on the Accessibility API provided by the Android System. To bind to this API, Android applications have to have the permission 'BIND_ACCESSIBILITY_SERVICE'. The API allows the service to interact with the system programmatically. Using the API, the HMAS transfers gestures to the screen and deletes text if needed. Android accessibility Services like HMAS can read, alter, and delete information displayed on the phone.

Since Accessibility Services can read, alter, and delete data on the device's input fields, it is rather cumbersome to activate these services on Android. On the one hand, it is logical, as it is a security feature to make it hard to activate such a service. On the other hand, however, it should be made as easy as possible for people with disabilities to activate assistive services. Nevertheless, users have to activate services like that manually. For this, one has to enable it in the Android system settings for accessibility (3.1).

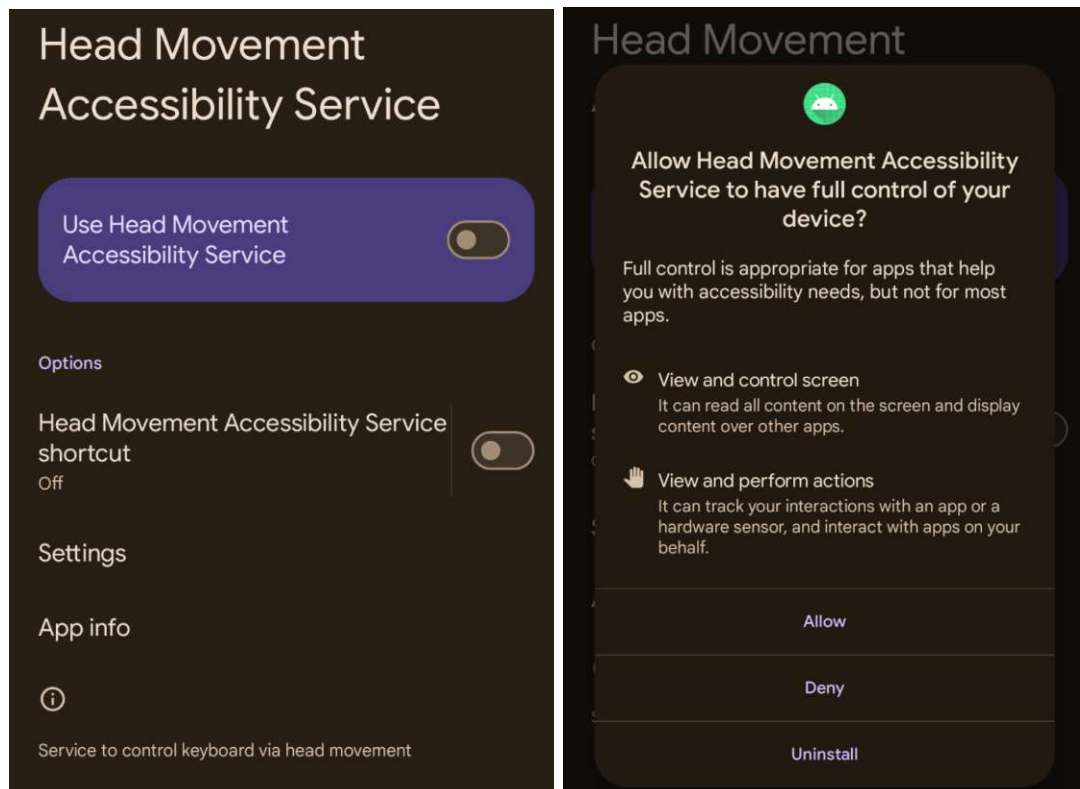


Figure 3.1: The users get security warnings when they try to enable the service

Once activated, the service starts and hooks itself to the Accessibility Event '*onAccessibilityEvent*' provided by the Android accessibility API. This event fires whenever something interesting happens like a click on a button or setting the focus on a textbox. For this proof of concept, only events for Textboxes are processed. Once users jump into a textbox, the FaceCursorService starts to do its work.

Face Cursor Service

The Face Cursor Service is highly based on the ML Kit for Firebase provided by Google [Goob]. This Framework allows on-device machine learning applications and already has built-in pattern recognition functionality. One of them allows real-time face recognition which was utilized to gain data about the head's position.

Most of the interaction between users and their devices will be done via the front camera of the smartphone. Once the Face Cursor Service is started, the cursor will be shown, and the users can interact with the smartphones via head movements. The cursor will initially be placed in the middle of the screen's lower third, where the keyboard is usually located. The cursor was implemented with 3 possible states: **WAIT**, **RECORD**, and **DRAW**.

Upon startup, the cursor rests in the WAIT state. Users can move the cursor freely in the direction they are looking. There is no touch input created and no movement recorded as long as the users are not blinking. But still the cursor can move freely around when users are moving their heads. Once the users blink, the cursor will change and record a touch gesture (RECORD). In record mode, the cursor represents the position where the touch input will be simulated. All head movements are transmitted from the Face Cursor Service after interpretation to a central **Cursor class** which calculates and stores the difference between each frame. The Cursor class is implemented with the Singleton pattern, so the state of the class is shared between all different services. As soon as the users blink for a second time, the gesture recording will be stopped, and the actual head movement is translated onto a 2d plane and onto the screen and executed by the HMAS as a virtual swipe gesture. Right now, the cursor is in DRAW state. By swiping on a keyboard that allows swipe gestures as input method like SwiftKey used in this study, the translated path is then translated to an actual word. When the gesture is finished, the cursor jumps back to WAIT state.

Initially, the horizontal position of the cursor will be the center of the screen, whereas the vertical position will be in the lower third of the screen as the onscreen keyboard is located there. Since the algorithm is calibration-free, it could occur that the users are not looking straight at the phone but the cursor position still is in the middle of the screen. To calibrate, the cursor can be moved to one edge of the screen, since the movement there is clipped as proposed in [CDF⁺20]. By that, the offset can be compensated.

Depending on the keyboard used, the input does not have to be very precise, since the swiping algorithm interpolates between the different possible words and selects the best match.

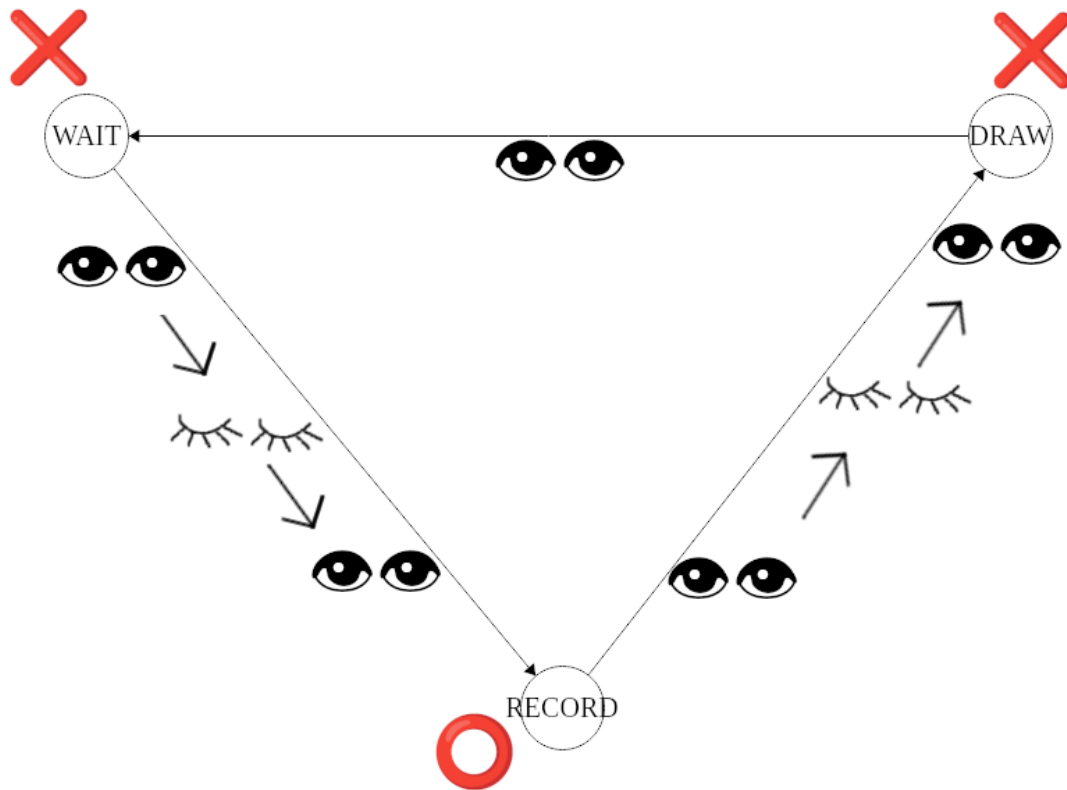


Figure 3.2: State machine for the possible states the cursor is in. Blinking triggers the state transitions. The red emojis shows the corresponding cursor presentation

To interpret the positioning of the head once a gesture-recording was started, movement data was gathered via the Accessibility API. The granularity of the mapped input highly depends on the number of pictures gathered by the phones' chip and interpreted by the machine-learning algorithm. With a rate greater or equal to 10 frames per second, the algorithm started to work as intended, otherwise, there was not enough data to be interpreted and the difference between two images was too big to be interpreted in a meaningful way. Several points all over the users' face are used to remain a functional service even if one of the points moves out of the image. For each frame, the position of the following landmarks is used

- left eyebrow
- right eyebrow
- bottom of the nose
- lower lip.

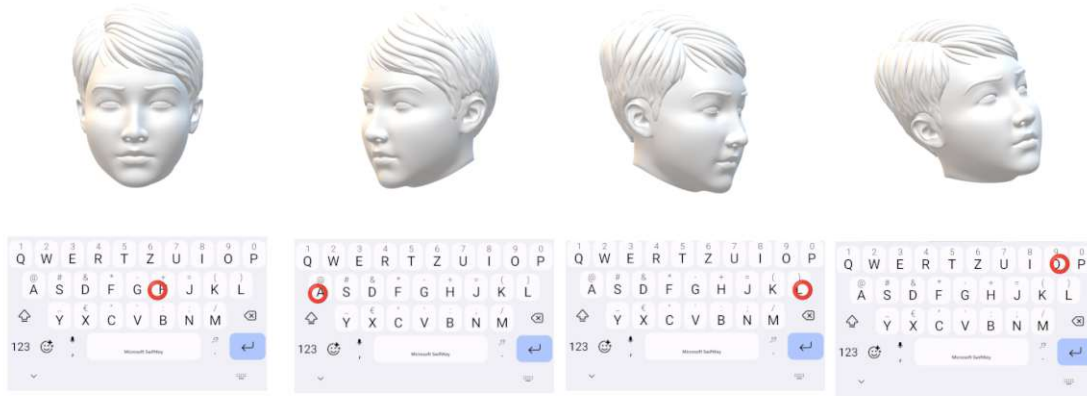


Figure 3.3: The necessary movement of the head to write the word 'Hallo'. The red circle marks the cursors' current position and visualizes the current position of the users' virtual finger.

For each of the above landmarks, the position is saved on the displays' X- and Y-axis. Finally, to receive one averaged point, $Average_{x/y}$ is specified as

$$Average_x = \frac{\sum_{n=1}^v P_{n_x}}{v}$$

and

$$Average_y = \frac{\sum_{n=1}^v P_{n_y}}{v}$$

where v is the number of visible landmarks and $P_{n_{x/y}}$ marks the landmarks' x or y coordinate.

The averaged point is then passed to another function where the delta of the current point from the previous saved point is collected, so let $Cursor_{x_n/y_n}$ be

$$Cursor_{x_n} = Cursor_{x_{n-1}} + (Average_{x_{n-1}} - Average_{x_n}) * 6$$

and

$$Cursor_{y_n} = Cursor_{y_{n-1}} - (Average_{y_{n-1}} - Average_{y_n}) * 6$$

where 6 is a scaling factor to allow bigger cursor movement with less physical effort of moving the head found most suitable during the first functional user tests. Since the actual head position is not used but rather the deltas between each frame, it allows device-independent functionality.

The ML Kit framework also allows to recognize facial expressions for example if the person's eyes are opened or closed. This data is utilized to start or stop a new input gesture. Since it's an approximation based on a trained model, a percentage is received. The probability of both eyes being closed is defined as

$$P_{\text{eyesopened}} = \frac{P_{\text{lefteyeopened}} + P_{\text{righteyeopened}}}{2}.$$

A threshold of 10% worked best to mark the eyes as closed, so

$$P_{\text{eyesclosed}} \leq 0.1$$

marks the beginning of a new gesture or the end of the current recording gesture.

Edge-Clipping

Since the display-space and the user-space are separate spaces and allow different values of movement, the users could potentially move the cursor out of the display-space. To prevent this from happening, a clipping algorithm is used in case the users go out of bounds. The algorithm works as follows:

$$Average_{x/y} = \begin{cases} 0 & \iff Average_{x/y} \leq 0 \\ Resolution_{x/y} & \iff Average_{x/y} \geq Resolution_{x/y} \\ Average_{x/y} & else \end{cases}$$

where $Resolution_{x/y}$ is the devices resolution regarding X- or Y-Axis. In words: the boundaries of the cursor are 0 and the resolution of the corresponding Axis. Because the algorithm does not include calibration and users' head starting positions can vary (e.g., they might not look straight at the display during initialization), edge-clipping allows them to recalibrate in case of deviation between the head angle and the current cursor position, similar to the approach proposed by Cicek et al. [CDF⁺20]. In figure 3.4, the red border is the clipping-space which the cursor can not exit. The blue / green border marks the barrier the cursor must pass in order for the redrawing / deletion function to be enabled.



Figure 3.4: The clipping-space and the zones where redrawing and word deletion gets triggered.

Gesture Drawing

Once the gesture has been collected successfully and the Cursors' state switches from RECORD to DRAW, the HMAS translates the recorded points to a Path instance. Each point previously recorded represents a single position in the Path object. For long words, the gesture can be rather long if there are a lot of direction changes like for words like "Steuerzahler" as seen in Figure 3.10. When the gesture is long but the duration is fixed, SwiftKey has problems interpreting the gesture correctly. To address this issue, the duration of the gesture in milliseconds is defined as

$$Duration_{Gesture} = 500 + |Cursor_{xy}| * 5$$

So for each delta previously stored, we add 5ms to the fixed duration value of 500 milliseconds.

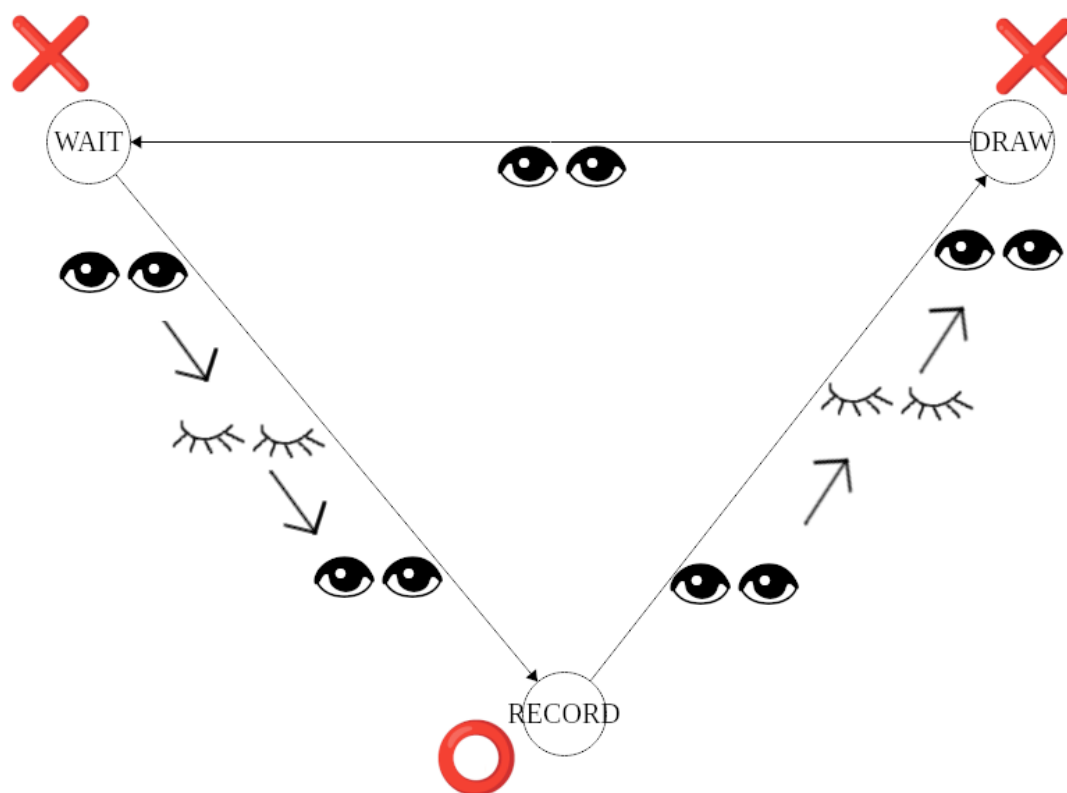


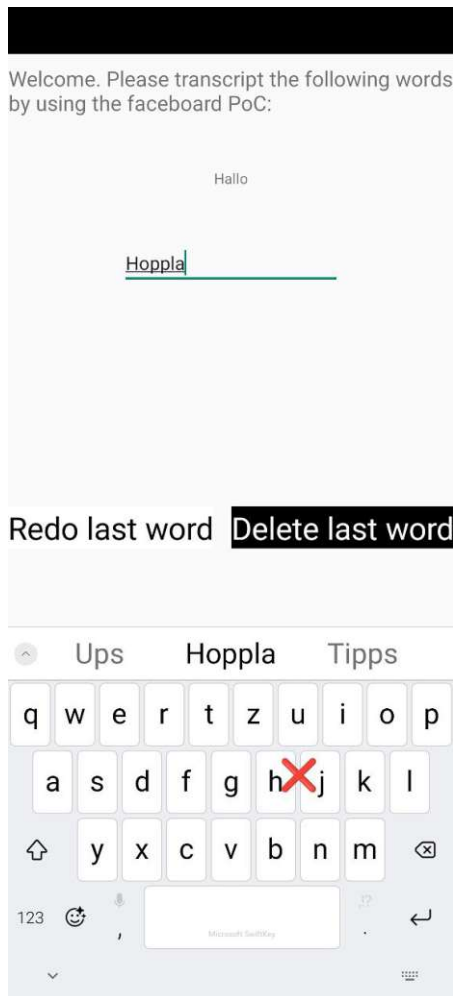
Figure 3.5: State machine for the possible states the cursor is in. Blinking triggers the state transitions. The red emojis shows the corresponding cursor presentation

Redrawing Gesture

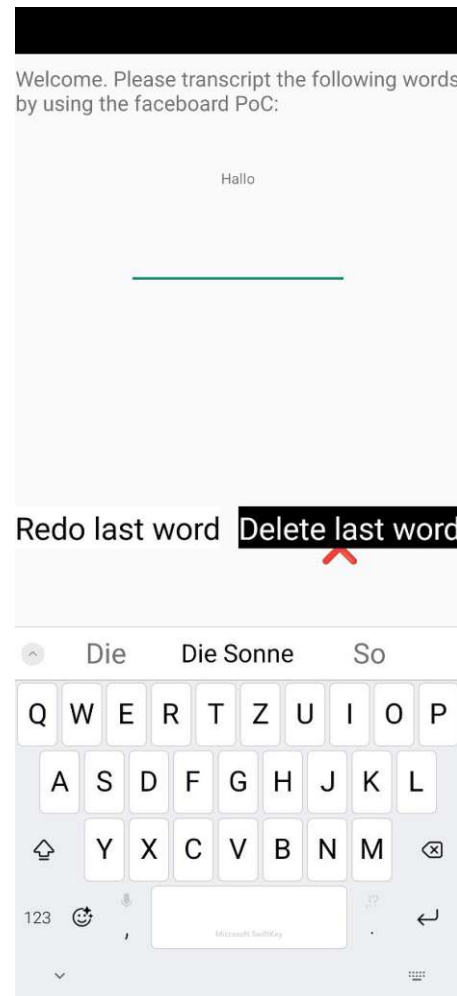
In case of accidental deletion of the last written word or if the system does not detect the last gesture, there is a “Redo last word” field above the keyboard that allows users to redraw the last gesture recorded by the service. For simplicity, the previous gesture is stored in the service and dispatched again once users enter that field.

Word Deletion

To delete the last written phrase, users move their heads upward until the cursor hits the “Delete last word” field next to the “Redo last word” field (as seen in figure 3.6b). The algorithm then deletes everything until the next whitespace is reached (for example, in “How are you doing,” it deletes “doing” and leaves “How are you ”). To delete another phrase, users must leave and re-enter the field. By doing so, the risk of accidentally deleting multiple words at once is minimized.



(a) In this scenario, a user accidentally wrote the word 'Hoppla' and wants to delete it



(b) Entering the 'delete last word' area will remove the last word from the input field

Tutorial

The tutorial part of the app handles two main tasks: permission management for the HMAS and introducing users to the new gesture-based input method. For permission management, the app presents different slides asking users to grant access to the camera (android.permission.CAMERA) for head movement recognition and to allow the app to draw over other apps (android.permission.SYSTEM_ALERT_WINDOW) for drawing the cursor as well as the deletion and redo fields (see figure 3.7). After granting these permissions, users must manually activate the HMAS service to proceed.

3. IMPLEMENTATION

Once the setup is complete, the HMAS service starts and the tutorial can be played. During the tutorial, users see a textbox along with words or sentences they can practice transcribing with the proof of concept.

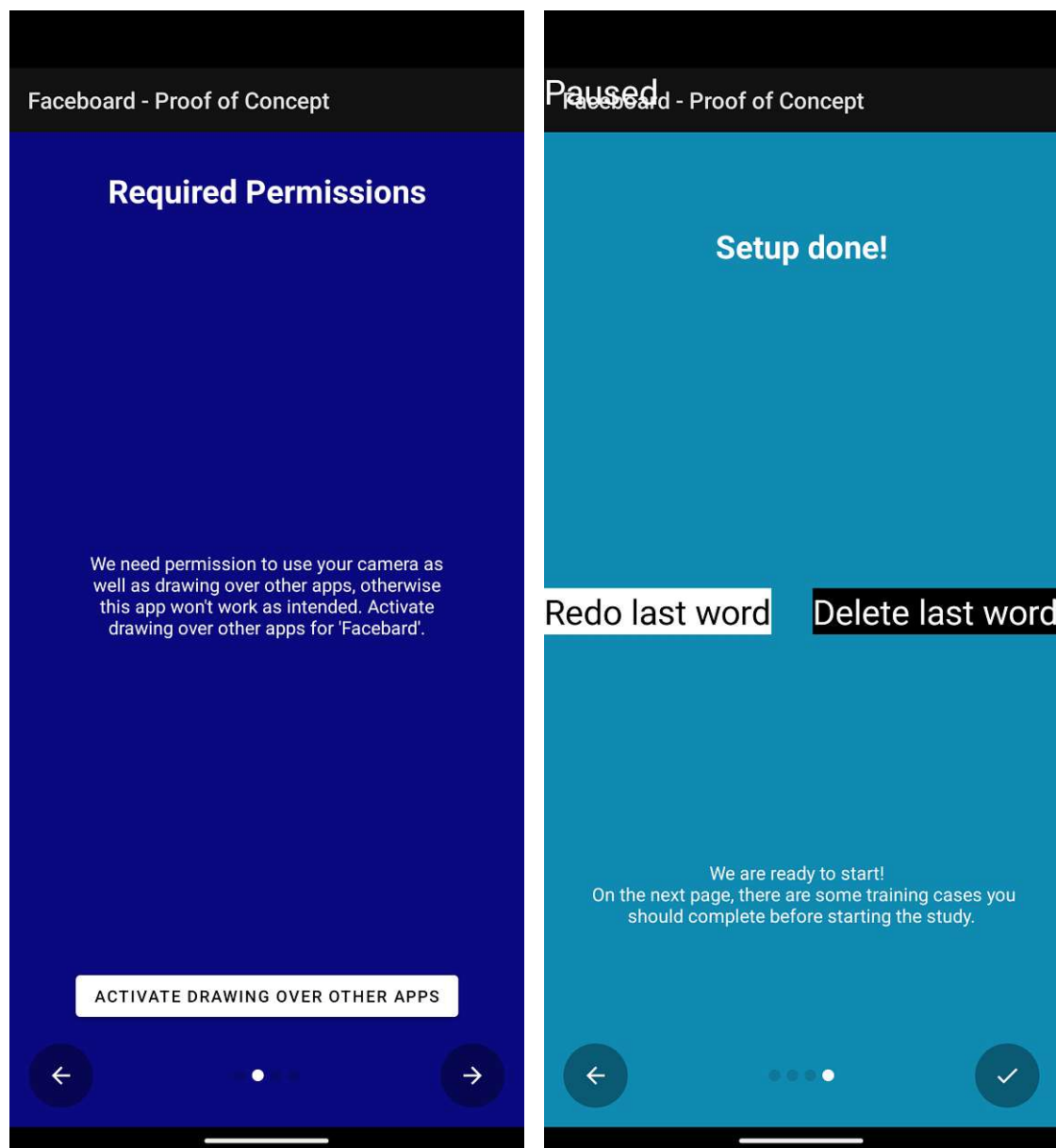


Figure 3.7: The tutorial guides users through the setup and finally activates Faceboard

3.2 Hyper Typer

To test the usability of the proof of concept, we used the serious game **Hyper Typer**. It was created in 2019 by Schlögl et al. to evaluate mobile text-entry performance in an unsupervised way [Sch20]. Hyper Typer is a native Android application based on the Unity Game engine. In this game, the users type pre-defined sentences provided randomly from a set of phrases to guide a spaceship through space (see figure 3.8). The only possibility to control the spaceship is via text entry. Now, the player tries to transcribe the phrase as fast as possible with the at least mistakes as possible. Each run is split into 5 games with 4 phrase which users have to transcribe to gather points. In the background, metrics are gathered and calculated (see chapter 'Methodology'). After finishing the last round, the game is over and the users can enter their names for high-score purposes. In the meantime, the round stats are transferred to the backend service.

Since the use case of this study was different from the one proposed by Schlögl et al., some adjustments to the app were made. To allow a hand-free finish of a round, every round automatically ended as soon as the presented text equaled the transcribed text. When one of the participants had problems transcribing one of the phrases, they still were able to do so by manually pressing enter on the phone. Also, Google Play services and with it the achievement system has been deactivated for simplicity. By the nature of the proof of concept, we had to rely on SwiftKeys' dictionary function. Some of the words in the phrase list proposed by Schlögl et al. can not be written using the standard dictionary. For that, the phrases were tested and removed if they were impossible to write.

3.3 Data Backend

The final application in the proof of concept triplet is the backend. This application is responsible for storing the data received from Hyper Typer after a game ends. The backend provides a REST-Interface for receiving the data from the game. The backend consists of a Spring Boot Java application utilizing Spring Data and a PostgreSQL database. The data received from Hyper Typer does not get modified and gets stored as received in the database.

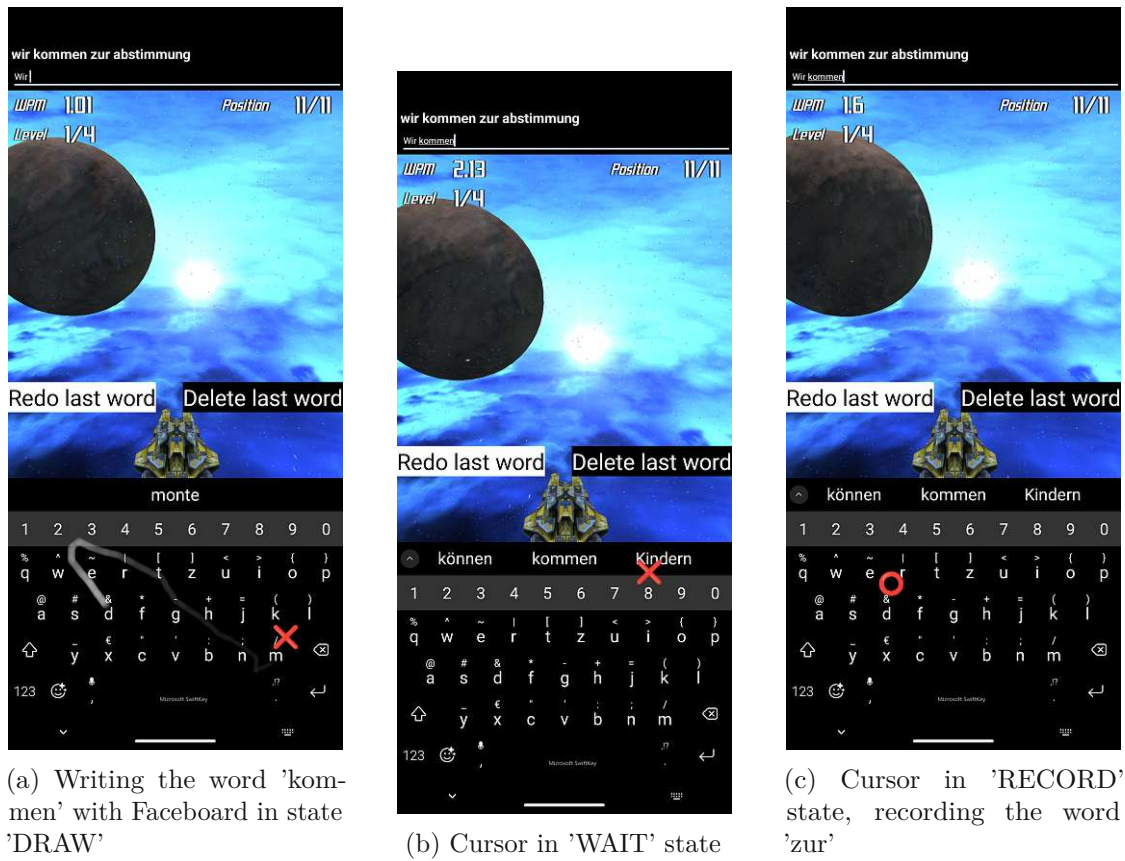


Figure 3.8: Different states in game

3.4 Functional User Tests

Functional user tests are important steps in the development of applications to ensure usability from the beginning. Testing early prototypes inefficiencies and problems with the algorithms can be identified before running the actual experiment.

3.4.1 First prototype

After finalizing the first version of the implementation, a few functional user tests with several devices and participants were conducted.

The first prototype used the position of the eyes for movement indication. Because of that, users with glasses not only had calibration problems, but also during the usage the cursor and path were not reliable. The algorithm seemed to have problems with recognizing the eye positions through the glasses.

Device	Camera	Android Version	Average Frames per Second
Xiaomi Mi 9T	20 MP	9	25
Nokia 6.1	8 MP	10	24
OnePlus 7T Pro	16 MP	8	16

Table 3.1: The different devices the first prototype was tested on

Another downside faced during prototyping was the different way users hold their phones when looking at them. For some users, the whole face was seen by the camera during the whole test whereas for others only parts of the forehead were seen.

The resolution of the camera had no significant impact on the performance since the machine learning algorithm only is fed with a small image of 320 by 240 pixels. Even though the 3 tested smartphones had similar specifications like 8 core CPU and display resolution, there were significant differences performance-wise. The framerate for the OnePlus devices was averaging at 15 frames per second whereas the Xiaomi and the Nokia device had stable framerate at about 25FPS.

Another downside noticed was differences in framerate and accuracy in different lit rooms. For example, in a low light environment with light coming from a lamp, the performance decreased drastically and the eye-detection accuracy dropped. Also, light coming from other directions than in front of the users decreased the accuracy of the machine learning algorithm. For this reason, a ring light was used for conducting the final user study to get unified lighting and prevent jitter in the data caused by bad lighting in different setups.

For some users, the recognition also tended to not recognize the eye opening immediately which resulted in flooding the app with open and close state changes and generating wrong or unintended input.

After the first user tests the following features were adapted:

- To allow better results, the algorithm for movement indication was changed. Not only the eyes' position were taken into account, but rather points all over the users' face and the average of the movement to allow people with glasses as well as people who do not fully appear on the camera's output to use the testing framework.
- a threshold of 300ms after state change from DRAW to WAIT was added to prevent accidental state changes after recording a phrase.

3.4.2 Second Prototype

Device	Camera	Android Version	Average Frames per Second
Xiaomi Mi 9T	20 MP	9	25
OnePlus 9 Pro	16 MP	11	28

Table 3.2: The different devices the second prototype was tested on

After fixing the problems with the first prototype, another round of functional user tests was conducted. During those tests, it was found that for some phones and keyboards, the accessibility service seems to ignore input gestures from time to time. To fix this, a button beside the 'delete last word' button was introduced which allowed to redo the last recorded gesture. This has a negative impact on the metrics but still allows better results than the necessity of recording the same gesture again. Also, during the studies, some users accidentally deleted written phrases by looking to the side or scratching their faces. Introducing the 'redo'-Button increased the performance of those users. Trying different swipe-based keyboards like SwiftKey, Google Keyboard and Samsung Keyboard, SwiftKey showed the best results and reliability on accepting drawn gestures compared to other keyboards tested which ultimately led to SwiftKey being the keyboard for the transcription tasks with Faceboard.

With the second prototype, the problems of non-recognized faces for people with glasses were removed by adding more landmarks. Still, the performance with glasses highly differed from those without glasses: The machine-learning framework has a hard time recognizing the eye-closing gesture depending on the head position due to glares on the glasses. To counter that, the participants all played without glasses on.

Also, the cursor was changed from a hand-symbols (see figure 3.9) to a red circle in which the users have to position the letter (when recording) and a red 'X' (when idle). This resulted in better performance and understanding since the pointing finger was confusing for the users.

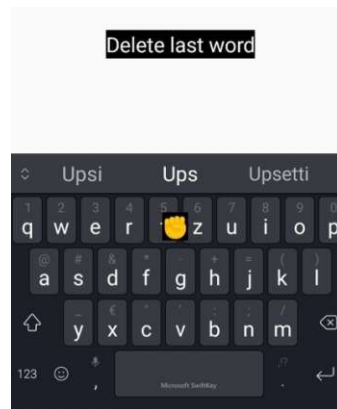


Figure 3.9: Screenshot of the first working prototype

Another point found was the problem long words generated: When the word was long and the gesture had a lot of direction changes, the fixed duration of the gesture was problematic. The different changes in direction were not interpreted correctly and wrong words were transcribed since some letters were simply left out. To fix this, the duration of the gesture was set to 500ms and for each point in the saved positions, 5ms were added. By that, even complex words like 'Steuerzahler' were possible to write without errors (see Figure 3.10, each of the 8 direction-changes is marked by a different color.).

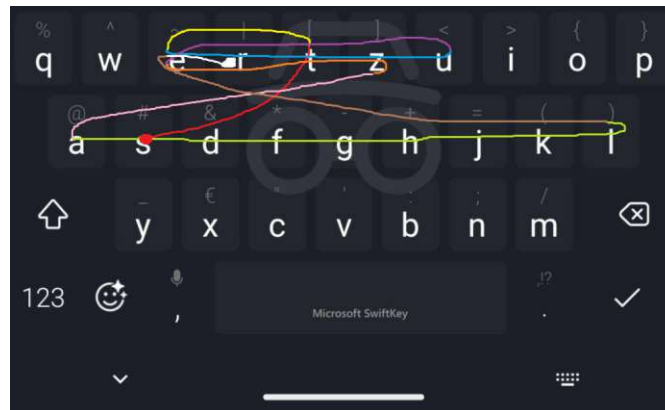


Figure 3.10: The gesture-path for the word 'Steuerzahler'.

During testing of the second prototype, a new test phone was introduced, the OnePlus 9 Pro. With an average framerate of 28 frames per second, the highest-performing phone was found. The study initially was planned to be conducted on the participants' phones, but since the performance highly varied between the phones tried during the functional user tests, it was fixed to the OnePlus 9 Pro for all participants. By this, we removed possible noisy data due to performance differences of the proof of concept.

3. IMPLEMENTATION

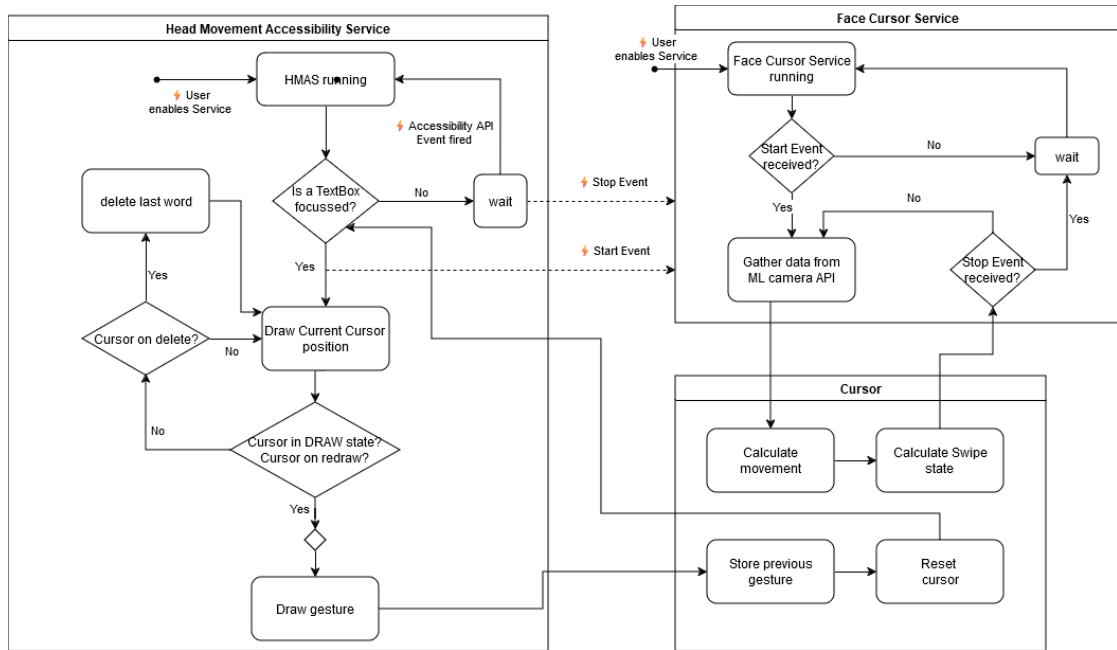


Figure 3.11: Component Interaction Diagram between the main services of the Proof of concept app.



Methodology

To prove the algorithm feasible, a **user study** was conducted. A user study is a method to evaluate how actual users are going to interact with an algorithm. Through this, insights on learnability, efficiency, memorability, errors, and overall satisfaction can be gathered. A user study allows for a better understanding of the users' behavior, challenges, and their overall subjective satisfaction with the tested concept [Nie94].

4.1 Research Approach

Designing a user study in research, there are two major types of study designs: a within-subject user study or a between-subject user study [CGK12]. Within-subject studies are experiments, where all participants are taking part in all different parts of the study. In between-subject studies, each group of participants is only exposed to part of the experiment. Using a within-subject study allows to gather more data with fewer participants and with less variability caused by differences between the participants since each of them acts as their own control. On the other hand, user studies designed as between-subject studies bear no risk of carryover effects, as participants only take care in a part of the planned experiments. Carryover effects describe possible conditions where participants are influenced between different steps of the evaluation [Gre76].

For this work, a **within-subject study** was conducted. Conducting the same study with each participant, allowed not only easier comparison between the different users but also within each individual participant. Since all volunteers played 5 matches with 4 phrases each, it can be checked if the volunteers are getting tired and creating more mistakes over time. Also, by doing the same task twice on different days, possible learning effects can be shown. Possible unwanted carryover effects like fatigue can be minimized by conducting the same experiment twice with a delay.

4.2 Data Collection

To evaluate the proposed method using head-movement-based gesture text input, data about the usage was gathered. This included statistics regarding app usage, performance metrics, questionnaires, and final interviews. The data collected was split into two groups: Experiment data and qualitative data. The experiment data describes the data that is automatically gathered by the applications during the experiment whereas the qualitative data includes everything the participants add after the technical part.

4.2.1 Experiment Data

The participants had transcript text. For the transcription task, the phrase-list from the work of Schlögl et al. was used [Sch20]. The phrases in this sets are mostly 4-grams¹ which were automatically generated using the method proposed by Leiva and Sanchis-Trilles [LST14].

The Hyper Typer application presents a phrase and the participants copy it using the newly proposed method. According to MacKenzie et al., empirical studies generally should prefer transcription tasks instead of text-creation tasks for several reasons: [Mac07]

- Participants should focus on the text-copy task instead of what to write or other tasks that are not linked to the text-input task. By eliminating the need for the participants to think about phrases to write, the text-writing performance is not compromised.
- If text-creation was used, identifying errors is the next big problem: What did the participants intend to write? Even if known beforehand, the users might forget what they were intending to write.
- Without pre-defined phrases, the distribution of the written letters and words is unknown. By that, the participants might not create a representative number of usages over all different letters.

By using pre-defined, short, and memorable phrases, participants tend to memorize the phrases which results in benefiting from the advantage a text-creation task brings: The focus of attention lies only on the input of the text. [Mac07]

The language for the transcription tasks was German. During this evaluation, the app was tested thoroughly and gathered insight into important input metrics. This evaluation was conducted with the Android App Hyper Typer. It is designed as a serious game and allows unsupervised performance analysis of different metrics (see table 4.1) established in text-input studies [Sch20]. The naming of the metrics follows the proposed naming by Soukoreff and MacKenzie [SM03].

¹A 4-gram is a sequence of 4 words that appear together in a text.

Note:

* Variables marked with an asterisk indicate session-specific configurations.

† Variables marked with a dagger indicate calculated metrics.

‡ Variables marked with a double dagger indicate metrics calculated in the database.

Symbol	Description
For each Game	
device_dpi	Android DPI class of the screen.
device_locale	Display language of the device.
device_name	Model of the device.
device_sdk	Android SDK version of the device.
end_time	End time of the game.
experiment_id*	ID of the experiment.
final_score	Final score of the game.
game_version_code	Internal version code.
game_version_name	Human-readable version.
game_was_paused	Game was paused/minimized during the experiment.
installation_id	Randomly generated UUID (128-bit) per installation.
keyboard	Name of the keyboard used.
keyboard_locale	Input language of the keyboard.
lower_case_only*	P contains only lowercase letters.
nr	Number of games completed in this session.
phrase_count*	Number of phrases.
phrase_set	Name of the phrase set used.
show_suggestions*	Auto-completion suggestions of the keyboard are active.
start_time	Start time of the game.
For each Round	
bs_count	Number of backspaces.
$c^\dagger$	Correctly transcribed characters c
$cer^\dagger$	Corrected Error Rate CER
end_time	End time of the round.
$f^\dagger$	Edit operations f
$if^\dagger$	Deleted characters IF
$inf^\dagger$	Incorrectly transcribed characters INF
$kspc^\dagger$	Keystrokes per Character $KSPC$
$p_chars^\dagger$	Length of P .
phrase_score	Score for the round.
presented	Presented phrase P .
start_time	Start time of the round.
t_chars	Total entered characters.
$ter^\dagger$	Total Error Rate TER
total_time	Total time required for this phrase.

4. METHODOLOGY

transcribed	Transcribed phrase T .
uer [†]	Uncorrected Error Rate UER
wpm [†]	Words per Minute WPM
wpm_adjusted [‡]	Adjusted Words per Minute $WPM_{adjusted}$
For each character entered	
auto_added	Character was added via auto-completion.
c_char	The entered character itself.
int	Character is a backspace.
time	Timestamp of the entered character.

Table 4.1: Overview of Variables and Metrics gathered, adapted from Schlögl et al., 2019 [Sch20]

Most of the technical game data collected by Hyper Typer was ignored since it is not needed in the scope of this study. The *start_time* and *end_time* were used to check how long each experiment took. The configurable settings were the same for every experiment and set as follows:

lower_case_only	true
phrase_count	4
show_suggestions	true

For the proof of concept, it was not intended to distinguish between lower- and uppercase, so the *lower_case_only* function was necessary. The *show_suggestions* configuration also had to be enabled, otherwise, SwiftKey did not allow swipe gestures, and by that the Faceboard app would not have worked.

For recognition purposes, the *installation_id* was used to match the different games of the same participants. To store it persistently and make analyzing the data easier, a table 'installation_user_correlation' was created in the database which linked the installation_ids to the participants (see 4.2).

installation_id	person	input_method
9b730168-0b8f-46ae-bbd0-d66dbfc0a33e	Noah	typing
d38204c0-0a97-41d4-b665-1c2275984297	Noah	faceboard
ec3e8828-8eea-4c71-9599-40569a9e52f5	Mia	swiping

Table 4.2: Example entries from the installation_user_correlation table

Entry Rates

The most used measure for text entry is the metric **Words per minute (WPM)**. Each word is a chop of 5 characters, including whitespaces. The WPM is defined as

$$\text{WPM} = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5}$$

where T is the transcribed string entered by the participants and therefore $|T|$ is the length of the transcribed phrase. S is the time needed for entering the string.

WPM alone is not sufficient enough to analyze a text-entry experiment. For the calculations in this study, the adjusted WPM is utilized. Here, The **uncorrected error rate UER** is taken into account. Most of the participants ended the phrases automatically by transcribing the phrase correctly, but still, some errors were left incorrect due to frustration or exhaustion. the uncorrected error rate is defined as (using the definitions from table 4.3)

$$\text{Uncorrected Error Rate UER} = \frac{\text{INF}}{\text{C} + \text{INF} + \text{IF}} \cdot 100.$$

With this, we can define the adjusted WPM with the following formula:

$$\text{WPM}_{\text{adjusted}} = \text{WPM} \cdot (1 - \text{UER})^a$$

Error Rates

Errors created during typing also have to be taken into account, and one possible quantifier is **Keystrokes per Character (KSPC)**. The *KSPC* metric measures the total written characters (Input Stream or *IS*) in relation to the actual phrase written (Transcribed phrase or *T*), so let KSPC be

$$\text{KSPC} = \frac{|IS|}{|T|}.$$

The drawback for KSPC as a metric for studies is that there are no distinctions between deleted characters, wrongly deleted characters, and initially, correct characters (when a letter in the middle of the word is incorrect and the writer deletes all correct characters until the incorrect one) [Wob07]. This means values closer to 1 mean higher efficiency. For the execution of this study, participants were only able to delete full words by using the 'delete last word' area. By this, the KSPC is a relatively good starting point in how many failed attempts a phrase caused.

To gain deeper insights into error rates, another metric looked at is the **Corrected Error Rate CER**. The CER measures errors the participants have corrected relative to the total number of keystrokes. This means the CER calculates how many errors actually were fixed, not just errors in the final text. The CER is defined as (using the definitions from table 4.3)

$$\text{CER} = \frac{\text{IF}}{\text{C} + \text{INF} + \text{IF}} \cdot 100.$$

Correct (C)	All correct characters in the transcribed text.
Incorrect-not-fixed (INF)	All incorrect characters in the transcribed text.
Incorrect-fixed (IF)	All characters backspaced during entry.
Fix (F)	All backspaces.

Table 4.3: Character classes used in computing corrected, uncorrected, and total error rates. From Mackenzie et al. [Mac07]

Now, the combination of CER and UER is the **Total Error Rate TER** with the formula

$$\text{Total Error Rate TER} = \frac{\text{INF} + \text{IF}}{\text{C} + \text{INF} + \text{IF}} \cdot 100.$$

By using these evaluation metrics, standardized results from all participants were gathered, which allows reliable comparisons between them. To remove possible noise, all participants used the SwiftKey keyboard for the newly proposed method, which supports gestures (swiping) to input text.

4.2.2 Word Extraction

Since users of Faceboard were only able to insert word by word, new data from the data mentioned in table 4.1 was derived for further analysis by an algorithm. The algorithm was made to reconstruct the words written by the participants by processing the keystroke (*c_har*) data. In the following section, the algorithm 4.1 is explained in detail for better understanding.

Symbol	Description
id	id of the word
word	the written word
deleted	if the word was deleted or not
time	timestamp of the word

Table 4.4: data structure for words

In the context of the data structure from table 4.4, a sample entrance can be found in table 4.5. Here, the participant wrote the phrase 'diese feststellung ist wichtig'. In the third row of the table, one can see that the participant accidentally wrote 'hat' instead of 'ist', so they deleted it and wrote the correct word instead. Using this structure allows efficient tracking of written words and their corresponding status after finishing the experiment. By that, word-by-word analysis was possible.

Phrase Filtering

First and foremost, the phrases get filtered by a list of installation_ids and phrases. By this, only specific installations can be looked into, for example, only the sessions that

used Faceboard as input method. Also, for Faceboard input method, the ones that were not completed correctly, have been omitted. Phrase filtering allows to focus on relevant data only.

Word Construction

Now, words are reconstructed from the actual phrases. Each phrase consists of keystrokes which are either letters, a space, or a backspace. Backspaces mark the deletion of a character whereas a space marks the end of a word. Each character now is added to a temporary word storage, until a space occurs.

Final Word Handling

If there is a backspace detected, it sure is a deleted word, so the word as a whole is marked as deleted. When a space is encountered, the word as well as metadata gets saved. The metadata includes if the word was deleted, the timestamp when it was written, and what phrase it is part of. The temporary literal is reset and the algorithm starts over with the next phrase.

4.2.3 Questionnaires

For subjective assessment regarding the usability of the newly proposed approach, questionnaires were conducted. To gain insight into how the participants liked the new approach or if they had problems, feedback in the form of the System Usability Scale (SUS) and NASA Task Load Index (NASA-TLX) were gathered [B⁺96][Har06]. For further feedback, informal, unstructured concluding interviews were held.

System Usability Scale (SUS)

The SUS is a fast way for participants to rate effectiveness, efficiency, and overall satisfaction with the proposed tool. The SUS is a quick, standardized questionnaire that is widely used in the evaluation of the usability of systems. According to a study by Lewis et al. from 2009, the questionnaire is used in 43% of industrial usability studies [LS09]. The SUS consists of 10 questions (see Fig. 5.2), and each of them is answered with a score from 1 ('Strongly disagree') to 5 ('Strongly agree'). Half of the questions contain a negative statement, the other half are positive statements. By that, response bias can be weakened. For each participant, a single usability score based on rules is created. For the score, the questions 1,3,5 and 7 with a positive statement are assigned with its scale position minus 1. The remaining questions (2,3,6 and 9) have a score of 5 minus their scale position. Then, all scores are summed up and multiplied by 2.5. This usability score provides valuable feedback about the system. Low SUS scores indicate issues with usability such as problems with the reliability of the blinking algorithm or the gesture drawing being too cumbersome. High SUS scores on the other hand would suggest overall satisfaction with the system and a high usability. In a study by Sauro et al., the acceptable score for above-average usability should be at least 68 [SotScho11].

NASA Task Load Index (NASA-TLX)

Besides using the System Usability Scale, more qualitative data was gathered by using NASA-TLX (Task Load Index) [Har06]. The NASA-TLX is a subjective workload assessment tool developed by Hart and Staveland in 1988. With its wide usage across domains including human-computer interaction, it allows comparing between assessed systems. Using TLX, further information on how the subjects felt during the tasks regarding mental workload was collected. NASA's TLX is rated in the six dimensions **Mental Demand, Physical Demand, Temporal Demand, Effort, Performance, and Frustration** (from Hart et al. [HS88]):

1. The dimension **Mental Demand** measures how much mental and perceptual activity was needed to fulfill the task. With focus on Faceboard, this can be tasks like searching for the right letters, remembering the phrase, and thinking about the path one has to draw to generate the desired word.
2. **Physical Demand** describes the amount of physical activity needed to finish a task. In the context of the proposed application, this could be the physical strains of moving the head, forcefully blinking the eyes and staying still to not move the cursor unintentionally.
3. The amount of time pressure felt during the experiment is indicated with the dimension **Temporal Demand**.
4. The **Effort** specifies how hard participants had to work to accomplish the performance delivered.
5. With **Performance**, the participants rate their subjective success in finishing the required tasks.
6. High levels of **Frustration** show that the users felt insecure, discouraged, irritated, stressed, and annoyed during their participation in the study. Low levels would mean the participants felt relaxed, gratified, content, and secure using Faceboard. Frustration is an important factor for new proposed systems as high levels of frustration can show flaws in the underlying mechanisms.

The participants rated each of the six dimensions on a scale from 0 to 100 which indicated the level of workload experienced. Then, the users rated the six dimensions against each other (pairwise comparison). The goal of rating the dimensions against each other is to create a weighting on the amount of workload each of the six dimensions contributed to the overall workload. Now, combining the score and the weighting leads to the Task Load Index. In context of this thesis, assessing the workload in the six dimensions is important to gain insights on how the participants felt during the experiment and where possible bottlenecks of the newly proposed method lie.

NASA-TLX provides a good view of the workload by splitting it up into six dimensions. Compared to the simplicity of the SUS questionnaire, the NASA-TLX is more time-consuming and also more complicated.

Overall, using NASA-TLX and SUS allows robust estimates on functionality, usability, and participants' satisfaction with Faceboard. According to Long et al., NASA-TLX and SUS do not correlate with each other [ou18]. Still, the combination of both questionnaires allowed deeper insights into the users' subjective performance and usability perception. While NASA-TLX focuses on the workload part of Faceboard, SUS allows more insights into usability. Combined they reveal an imbalance between usability and workload. For example, a high SUS score and high NASA-TLX workload scores could show that the system is generally usable, but may be physically straining.

Interview

To gather further feedback, a concluding unstructured interview was held after the second participation day after the questionnaire. The interview was of a supportive nature and intended to provide additional subjective feedback and an open feedback channel. The provided feedback was written down and can be found in section 'Participation Notes'. To minimize potential discrepancies in the response, participants are advised to answer honestly on both the questionnaire and the interview questions. Additionally, removing possible time lag and context shifts is minimized by conducting the questionnaires and interviews right after each other. Minimizing those common issues described by Harris and Brown in 2019, the triangulation of open questions and questionnaires should be more straightforward [HB19].

Using a concluding interview allows the participants to give additional input about challenges they faced during the process, possible ups and downsides they noticed, and ideas on how to make the proof of concept more usable. This also allowed to combine it with the participants' questionnaire outcomes and gain further possibilities to interpret the qualitative data gathered. Using unstructured interviews might uncover unexpected issues, suggestions for future improvement, or additional feedback that would not emerge through the two used questionnaires.

Richer feedback is expected from the concluding interview as, after finishing the practical part and the two questionnaires, the participants may still have fresh reflections on the experience. Also, it allows the users to further explain their ratings or statements made in the SUS or NASA-TLX.

Algorithm 4.1: Derive Words from Keystrokes

```

1: Initialize current_word as an empty string
2: Initialize deleted_word as false
3: for all phrase_id in phrase_stats where transcribed = presented and
   installation_id is in list of installation_ids do
4:   current_word ← ""
5:   for all keystroke_row in keystroke with current phrase_id ordered by
     id, time, phrase_id do
6:     if keystroke_row.c_char = " and keystroke_row.int = false then
7:       deleted_word ← true
8:     else if keystroke_row.c_char ≠ ' ' and not deleted_word then
9:       Append keystroke_row.c_char to current_word
10:    else
11:      if current_word ≠ "" then
12:        Insert current_word into words_usual with:
13:          deleted ← deleted_word
14:      end if
15:      current_word ← ""
16:      if deleted_word then
17:        Append keystroke_row.c_char to current_word
18:      end if
19:      deleted_word ← false
20:    end if
21:  end for
22:  if current_word ≠ "" then
23:    Insert current_word into words with:
24:      deleted ← deleted_word
25:  end if
26: end for

```

id	word	deleted	timestamp
29019	diese	false	12531
29020	feststellung	false	21666
29021	hat	true	30814
29022	ist	false	40961
29023	wichtig	false	42043

Table 4.5: Example entries from the words table

CHAPTER 5

Study Design

The study was designed as a laboratory experiment. The participants were supervised during the whole process in a controlled environment with the same setup for every user. By this, possible disturbances like noise can be controlled which results in more uniform data.

5.1 Procedure

To check the second research goal (**RG2** - Conduct an experiment using the developed prototype and the users' standard input method to assess the practical usability in a comparable manner), the participants were guided through the following steps:

1. Participants received a fresh install of the Faceboard app paired with the accessibility service. Providing an installer, it is tested if the users are able to install and start the service on their own (non functional requirement NFR1).
2. A tutorial on how to use the new input method is given in the tutorial part of the application. This should be sufficient for the participants to input text via the proposed head-based swiping (non functional requirement NFR2).
3. The participants are using Hyper Typer with the proposed approach. Each participant is using Faceboard with a freshly installed and configured SwiftKey keyboard. The users try to insert text as fast and correctly as possible with as few errors as possible while metrics are gathered in the background of the app. Analyzing the collected metrics afterwards allows to check whether research goal **RG2** is met or not.
4. A second run of typing with Hyper Typer using the new input method is done after at least one day of break to weaken possible unwanted carryover effects.

5. The participants are using Hyper Typer with their usual input method (can be swiping or single tap input via the keyboard). The extracted data will be the baseline of the quantitative evaluation. The metrics as described in table 4.1 were collected and processed.

This procedure enabled direct comparison between the participants' performance on their usual input method as well as Faceboard. Thereby, the hypotheses **H1.1** through **H1.3**, which check for differences in text entry speed, efficiency and error rate, can be tested. Additionally, it enabled to check the performance of all participants in general. Potential learning effects could be observed by running the transcription task using Faceboard for a second time. By repeating the tasks, hypotheses **H2.1** through **H2.3**, which examine whether a learning effect regarding speed, efficiency and error rate occurs, are tested.

After every round, the gathered data was sent via WiFi connection from the phone to a backend. This backend consists of a web server that collects the data and stores it in a PostgreSQL database.

For the evaluation, a desk with a ring light was used to provide equal lighting for all participants and also get more stable results since the users do not have to hold the phone with their hands during the whole experiment (see figure 5.1). The light had a built-in phone mount, which allowed to freely move the phone to the users needs. At the same time, using the phone mount, possible bad data/input due to hand tremors was reduced. It also allows similar lighting for all participants, regardless of the time of day the study was conducted. During the study, the participants sat at a height-adjustable desk and height-adjustable chair to change the perspective of the camera in relation to the face and the users needs. This enabled fine-tuning the cursors' position as well as allowed people with smaller eyes to look up at the screens' keyboard instead of down which allowed the machine-learning algorithm to better recognize the blinking.

Using this standardized setup allowed us to gather data produced by the participants to be more consistent and made the runs easier to compare between each other.



Figure 5.1: Example of the evaluation setup with one of the participants during one of the pilot studies.

For the runs, the study was conducted in a quiet room with minimal background noise and distractions to allow participants to concentrate fully on the tasks to their needs. During the runs, no help was provided, only in case the participants asked for help, finished one of the rounds, asked for a break, or wanted to continue the experiment.

So, the full procedure consisted of the following steps:

Procedure Day 1:

1. Install Hyper Typer
2. Install Faceboard
3. Install SwiftKey
4. Setup Faceboard
5. Play the tutorial
6. Break
7. Play Hyper Typer with Faceboard (If necessary, break between each of the 5 games)

Procedure Day 2:

1. Install Hyper Typer
2. Install Faceboard
3. Install Keyboard
4. Play Hyper Typer with users usual input method
5. Break
6. Setup Faceboard
7. Play the tutorial
8. Break
9. Play Hyper Typer with Faceboard (If necessary, break between each of the 5 games)
10. SUS
11. NASA-TLX
12. Concluding Unstructured Interview

5.1.1 Procedure Steps In Detail

To keep track about the progress, achieved goals by the participants were written down in the following way:

- ☐ Setup Faceboard.
 - ☐ Independently setup Accessibility Service.
 - ☐ Allow camera permissions.
- ☐ Play tutorial mode
- ☐ Play 5 rounds of Hyper Typer with Faceboard.
- ☐ Play 5 rounds of Hyper Typer with usual input method.
- ☐ Play 5 rounds of Hyper Typer with Faceboard.
- ☐ Fill out SUS questionnaire
- ☐ Fill out TLX questionnaire

The notes taken during the procedure can be found in the section 'Participation Notes'.

Setup Faceboard

First and foremost, the participants set up Faceboard as far as possible on their own. To do so, they have to finish the following steps:

- Configure the Head Accessibility Service on the smartphone. Without this crucial step, the application will not function as intended and the study can not be continued.
- All necessary permission have to be granted (Drawing over other apps, allow camera access)

If they are not able to set this up on their own, they will get assistance to make sure the study can be continued. When the users finished the setup, the 'Setup Faceboard' step is checked. If they did not finish it independently, it will be noted next to the checkbox.

Play Tutorial Mode

When the participants are ready and set up the Head Accessibility Service, they proceed by playing the tutorial mode. The step introduces them to Faceboard and provides the opportunity to get used to the head gestures as well as the blinking for starting and stopping the gestures. Also, since there is a very high chance of errors during the first tries, the users are able to learn the deletion and redoing of words. In the tutorial mode, the participants have 4 simple tasks to do: It starts with the transcription of the words 'Die', 'Sonne' and 'lacht', to get used to writing single words. Afterwards, the users are prompted with the sentence 'Die Sonne lacht'. Now, the participants try to enter each word by word. After finishing the four tasks, the participants are encouraged to keep trying as long as they want and feel confident.

During the tutorial mode of the study, the participants are already encouraged to give feedback.

Play 5 Rounds of Hyper Typer with Faceboard

Now the experiment starts with the first five rounds of Hyper Typer. Here, the users play five games with 4 phrases each to transcribe. After every round or every 4 phrases, the participants can choose to take a break. To make sure the five games are conducted, a note is taken after every round. Also, throughout the experiment, the metrics like WPM and error rates are recorded and sent to the backend to store for later analysis. Additionally, the feedback given by the participants is written down on paper and later transcribed for the theoretical evaluation.

After finishing the first five rounds of Hyper Typer with Faceboard, the first day of the study is over. Now, the participants wait at least one day to minimize unwanted carryover effects like fatigue.

Play 5 Rounds of Hyper Typer with Usual Input Method

On the second day of the study, the participants are again introduced to Hyper Typer, but this time using their usual input method like swiping or typing, depending on their preference. Another five rounds of Hyper Typer are played, this can serve as a baseline for comparison of their usual performance to the one made by Faceboard.

Play 5 Rounds of Hyper Typer with Faceboard Again

Afterwards, the users play five rounds of Hyper Typer using Faceboard once again. To introduce them to Faceboard again, the transcription part of the study is played again. When the participants feel confident to proceed to Hyper Typer, the actual transcription tasks are started.

Again, feedback is encouraged throughout the whole experiment.

Fill Out SUS Questionnaire

After finishing the practical part of the study, the participants filled out the two questionnaires. This started immediately after the task was finished to make sure participants did not forget about their experience. The first questionnaire was the System Usability Scale (SUS). The survey measures usability and satisfaction with the Faceboard entry system. The questionnaires were held via computer as proposed by Hart et al. for laboratory environments [HS88]. For this, the participants were introduced to a web form based on HTML and JavaScript, which contains 10 questions with a Likert-scale-based rating scheme from 'Strongly disagree' to 'Strongly agree'. The participants click the corresponding value for each question and when they finish, they finish it and the corresponding answers are saved, including the SUS score.

I think that I would like to use this system frequently				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I found the system unnecessarily complex				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I thought the system was easy to use				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I think that I would need the support of a technical person to be able to use this system				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I found the various functions in this system were well integrated				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I thought there was too much inconsistency in this system				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I would imagine that most people would learn to use this system very quickly				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I found the system very cumbersome to use				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I felt very confident using the system				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree
I needed to learn a lot of things before I could get going with this system				
☐	☐	☐	☐	☐
Strongly disagree				Strongly agree

Figure 5.2: System Usability Scale (SUS)

Click on each scale at the point that best indicates your experience of the task

Mental Demand Low High	How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand Low High	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand Low High	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
Performance Good Poor	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
Effort Low High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Low High	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Continue >>

Figure 5.3: NASA-TLX Step 1

Fill Out TLX Questionnaire

Next, the NASA Task Load Index (NASA-TLX) questionnaire was filled out. The NASA-TLX was done via a computer with a modified version made by Keith Vertanen [?]. Again the users got introduced to the questionnaire and how the digital version of it works. The process is completed in two separate steps. First, the participants rated the workload based on the six dimensions (see 5.3). For consistency of the study, the NASA-TLX is referenced. During the questionnaires the german version was used.

After the first step, the second step is weighing the dimensions. This is done by pairwise comparison of each dimension, reflecting their workload contribution relative to each other (as seen in figure 5.4). The resulting weights are used to calculate the overall workload with respect to the participants' subjective weight of each workload dimension. By the TLX questionnaire, data about the cognitive and physical effort required to use Facebook gets collected.

Click on the factor that represents the more important contributor to workload for the task

Physical Demand	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
OR	
Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Click on the factor that represents the more important contributor to workload for the task

Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?
OR	
Mental Demand	How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Click on the factor that represents the more important contributor to workload for the task

Performance	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
OR	
Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Click on the factor that represents the more important contributor to workload for the task

Mental Demand	How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?
OR	
Effort	How hard did you have to work (mentally and physically) to accomplish your level of performance?

Click on the factor that represents the more important contributor to workload for the task

Physical Demand	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
OR	
Temporal Demand	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
OR	
Effort	How hard did you have to work (mentally and physically) to accomplish your level of performance?

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
OR	
Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Click on the factor that represents the more important contributor to workload for the task

Performance	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
OR	
Temporal Demand	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

Click on the factor that represents the more important contributor to workload for the task

Performance	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
OR	
Physical Demand	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

Figure 5.4: NASA-TLX Step 2 - Rating on subjective workload across the six dimensions

Concluding Interview

Finally, concluding unstructured, informal interviews with the participants are held. Questions are asked to gather additional feedback besides the already collected feedback during the tasks.

- Did you find the process of moving your head exhausting after a while?
- Did something else bother you when using Faceboard?
- What did you like about the approach?
- Any additional feedback?

5.2 Preparations

To make sure there were no interferences between the different runs of the participants, some preparations before writing with Faceboard were necessary.

5.2.1 Preparing SwiftKey

Since SwiftKey learns how often words are written on keyboards and by this modifies the probabilities of words, the App Data had to be deleted between each experiment. This made sure the participants did not accidentally influence each other.

SwiftKey had to be configured in the correct way to work with Faceboard: First, auto-correction, automatic capitalization, and intelligent punctuation had to be deactivated in SwiftKeys' settings since the keyboard automatic features your input might distort the participants' real input metrics. Also, the language pack "Deutsch '(Deutschland) / Deutsch (DE)'" had to be downloaded since the phrases transcribed were in German language.

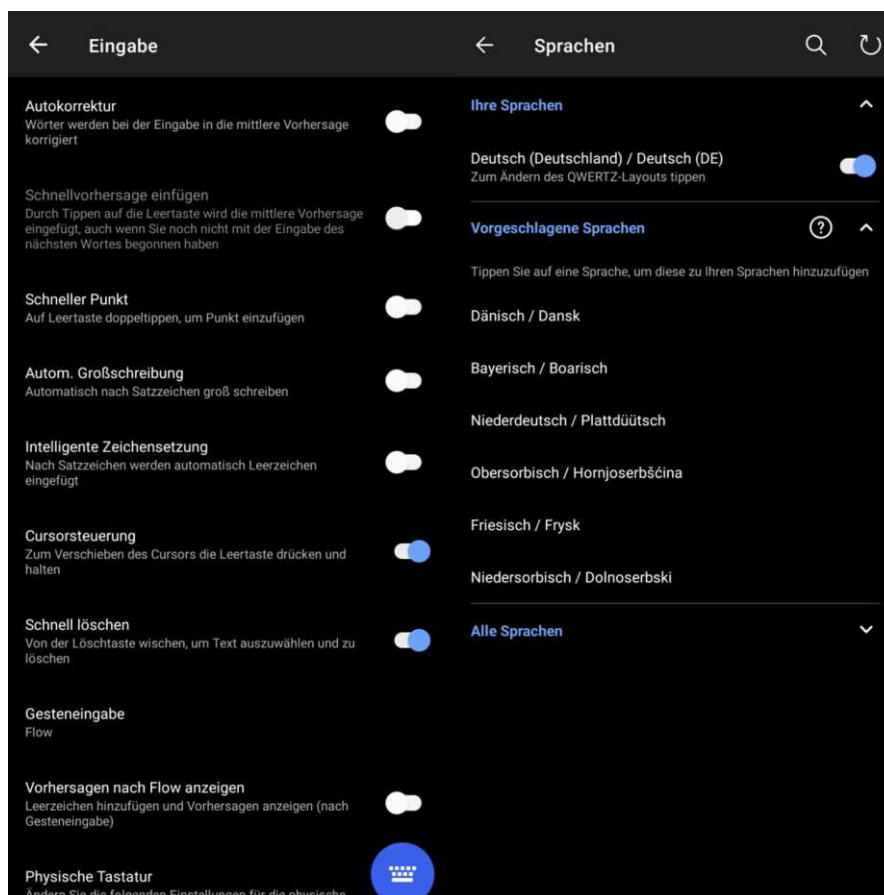


Figure 5.5: Necessary Swiftkey Configuration

5.2.2 Preparing Hyper Typer

To make sure Hyper Typer and the backend worked as intended, a test run was done to make sure the data arrived in the backends' database as planned before the participants' runs of the game.

Also, for easier distinction between the participants' runs, Hyper Typer was reinstalled. By that, each users' dataset was linked to a different Android *installation_id* which allowed it to easily research individual performances.

5.2.3 Preparing Faceboard

Since the Faceboard service was installed throughout the whole experiment, App Data had to be deleted before each run. This ensured that all participants in the study had to approve the same configurations and permissions for the app.

5.2.4 Preparing Environment

To minimize distractions, the desk where the experiment was conducted was clean and all distractions like monitors and smartphones were removed. Lights were turned on to ensure all participants had equal conditions throughout the process.

The phone was set to 'Do Not Disturb' to remove unwanted noise due to notifications.

5.2.5 Moderator Script

To weaken unwanted noise caused by inconsistent explanations, the conducted user tests followed a moderator script. This moderator script was used as guidance to unify the process of the studies as much as possible.

During each user study, notes have been taken whenever the participant had supplementary comments which could be interesting for the qualitative study.

The moderator script is divided into 8 steps each representing a different phase of the conducted experiment. Using the subdivisions, it is easier to understand and follow.

5.2.6 Moderator Script Outline

The notes, as well as the 'word-by-word', moderator script can be found in the appendix in section 'Participation Notes'.

Step 1: Welcoming the Participant

- *“Hello _____ and thanks for helping me out by evaluating my app for my master’s thesis. Can I get you a drink? I also brought some snacks since this will probably take a while for both of us. Feel free to grab them whenever you’re hungry.”*
- *Proceed once the participant is comfortable and ready.*

Step 2: Explain What We’re Going to Do and Why

- *Explain what Faceboard is about, how it works, why it is needed*

Step 3: Preparations for the Experiment

- **Check Preconditions for experiment**
 - *Ask the participant if they know how to use a swipe-based keyboard.*
 - *If not, explain swiping with an example word such as “Hello”: press and hold ‘H’, then drag across the letters ‘e’, ‘l’, ‘l’, ‘o’, and finally lift your finger.*
- **Reinstall SwiftKey / Delete App Data of SwiftKey**
 - *Configure SwiftKey to work with Faceboard*
 - *Download the language pack “Deutsch (Deutschland) / Deutsch (DE)”.*
 - *Install Faceboard if necessary*
 - *Re-Install Hyper Typer*

Step 4: The Experiment and its Process

- **Faceboard Setup and Usage**
 - *Participants setup Faceboard*
 - *Tell the participant to try to set up independently (provide help only if needed)*
 - *Participant plays tutorial mode*
 - *Provide Tips on how to use Faceboard in a more efficient way*
- **Provide Tips on Using Faceboard Efficiently**
 - *Tell the participant how the calibration works*
 - *Provide tips on where to put the cursor on the letters of the keyboard*
 - *Stop gesture on the way to the last letter for better performance*

Step 5: Play 5 rounds of Hyper Typer with Faceboard

- *Explain the game “Hyper Typer”: How does it work with Faceboard, what is the main goal, whats happening in the background*
- *Explain procedure: 5 rounds played with Faceboard with focus on correctness of the transcribed phrases*

Step 6: Hyper Typer with Normal Input Method (only on second run)

- *Explain the game “Hyper Typer” again if needed: How does it work with usual input method, what is the main goal, whats happening in the background*
- *Explain procedure again: 5 rounds played with usual input method with focus on correctness of the transcribed phrases*

Step 7: Filling Out the Questionnaires (only on second run)

- *“The practical part is over. Thanks for participating so far. Now you will fill out two forms: the System Usability Scale (SUS) and the NASA-TLX.”*
- **System Usability Scale (SUS):**
 - *Explain what the System Usability Scale is, why we use it and how it works.*
- **NASA-TLX:**
 - *Explain what the NASA-TLX is and the different Dimensions (Mental Demand, Physical Demand, Temporal Demand, Effort, Performance, Frustration), why we use it and how it works.*
 - *Each dimension is scored from 0 to 100, then you weigh them by comparing each dimension to the others.*
- *Tell the participants to answer honest since it is crucial for the outcome of the study*

Step 8: Final Questions (every run)

- *Did you find the process of moving your head exhausting after a while?*
- *Did something else bother you while using Faceboard?*
- *What did you like about the approach?*
- *Any additional feedback you would like to share?*

Participant	Gender	Age (years)	Occupation	Usual Input	Restrictions
Ethan	Male	34	Self-employed	Swipe	No
Emma	Female	23	Student	Typing	No
Noah	Male	33	Software developer	Typing	Glasses*
Oliver	Male	56	Railway worker	Typing	Glasses†, herniated discs (C4 / C5)
Sophia	Female	56	Retired waitress	Typing	Glasses*, herniated discs (C2 / C3)
Benjamin	Male	30	Project manager	Typing	No
Liam	Male	30	Software developer	Typing	No
William	Male	32	Software developer	Typing	No
Daniel	Male	32	Self-employed	Typing	No
Olivia	Female	33	Kindergarten teacher	Swipe	Glasses†
Grace	Female	31	Public service	Typing	No
Henry	Male	45	Software developer	Swipe	Glasses*
Samuel	Male	31	Software developer	Swipe	No
Mia	Female	22	Court clerk	Swipe	No
Michael	Male	30	Project acquisition	Typing	No
Charlotte	Female	30	Student	Typing	Glasses*, Rheumatoid Arthritis

Note: * Glasses removed during game rounds. † Wore contact lenses. ‡ Glasses not used during the experiment.

Table 5.1: Demographic Characteristics of Participants

5.3 Participants

The study was conducted with 16 volunteers of different ages, genders, ethnicity, and fields of work. The 16 participants were aged between 20 and 57 years, of which 10 of the participants were male and 6 were female. All of them speak German, 13 of the participants are native speakers, and the remaining three speak native English, Turkish, or Spanish. For privacy reasons, the names of the participants were changed.

The field of work represents a wide spectrum from tech-savvy users like software developers to retired individuals and kindergarten teachers with little to no technical background. The diversity in technology knowledge allowed to capture feedback regarding usability from different areas of employment and technical understanding during the study.

13 of the participants are healthy or not limited in their head movement or finger mobility, whereas for the remaining participants, one has limited head mobility combined with pain in the fingers after disc surgery, one participant suffers from discopathy in the cervical spine resulting in pain on movement and one has rheumatoid arthritis. These three participants represent a critical subgroup for evaluation of the proposed system in regard to accessibility. Feedback of the three participants with medical challenges allowed to assess the systems' usability across different user groups.

7 of the 16 participants require corrective vision aids like glasses or contact lenses. During prototyping, problems with glasses and the blinking detection have been noticed. For this reason, participants were advised to take off their glasses during the tutorial if the performance of the blinking detection was poor.

The 16 participants were recruited using a convenience sampling approach, meaning the people were selected based on their availability and their willingness to take part in the study [Dor07]. No selection based on age, ethnicity, gender or other social variables happened, the participant selection was random. Still, the random sample is a good representation of the population: According to WHO, 16% of the world's population is suffering from some kind of major disability [WHO23]. In 2017, the Austrian Ministry of Social Affairs reported 18,4% of Austria suffer from a permanent impairment [soz]. With about 19% of the participants being impaired to some extent, the convenience sample is a good representation of the population.

All participants had to fulfill the same steps of the study independently from each other. This ensured the comparability between the studies. To eliminate the participants' phones' performance influencing the result in any way, the phone used for the head-assisted typing was fixed to the OnePlus 9 Pro, since it showed the best overall performance during the prototyping phase.

Split between their usual input methods, 5 participants use swiping as their usual input method whereas the remaining 11 users use typing as their preferred input method. Since Facebook input method uses similar mechanics as swiping, possible advantages were looked into as the transition to head-assisted swiping might be easier for swiping participants. Conversely, typing participants who never use swiping as their usual input method might require more practice to master the newly proposed input method.

To analyze their usual input method, the participants used their own phones as well as their preferred keyboard. This allowed users to type in a familiar environment and eliminate any unwanted effects of typing on a different phone with different configurations regarding keyboard type, keyboard/screen size, touchscreen sensitivity, etc.

5.3.1 Participants With Relevant Restrictions

The focus of this work lies on accessibility for motor impaired persons. Even though 13 participants are healthy, 3 participants with relevant restrictions represent the critical subgroup most interesting to evaluate the application on.

Sophia

Sophia, a 56-year-old retired waitress, is faced with herniation in the C2 and C3 of the cervical spine which results in spinal stenosis and cervical radiculopathy. Due to a delayed operation, nerves have been permanently damaged resulting in numb fingers. Occasional dizziness caused by the spinal stenosis further complicates her daily life. On days when her condition is worse, Sophia experiences problems with fine motor control when typing on her phone caused by the numbness and pain of her fingers.

To allow more insights into the process as well as the feelings during the experimental usage of Faceboard, Sophia's process was recorded and transcribed.

Oliver

Oliver, a 56-year-old railway worker, has two herniated discs in the cervical spine (C4 and C5). Because of this condition, Oliver has problems staying in a static position for a prolonged period of time or doing repetitive movements. Additionally, numbness in his right hand when sitting in the same stance for a long time occurs. On days when his condition is worse, occasional cramps in the fingers of the right hand can happen.

Charlotte

Charlotte has rheumatoid arthritis which is a condition characterized by joint inflammation and pain. The degree of symptoms fluctuates with medication availability, stress levels, and other factors. Joint stiffness, especially in the morning hours, can influence their performance with usual input methods.

Participants with Vision Impairment

7 participants of the study do wear glasses or contact lenses. Since the blinking algorithm relies on recognizing the participant's eyes, having participants with vision impairment is particularly relevant to the study. The ability of the camera to clearly capture the eye positioning and blinking state might be affected by lens reflections. Also, people with vision impairment might have problems reading the phrases that had to be transcribed.

CHAPTER 6

Evaluation

This chapter focuses on how the data was analyzed and methodologies were used. Statistical analysis was defined to find the first indicators of the usability of the newly proposed approach. Performance metrics like words per minute (WPM) were analyzed to measure the efficiency of the participants. Methods for statistical significance by usage of comparative as well as correlation and regression analysis are defined.

To analyze the outcomes of this thesis, the tested hypotheses were evaluated regarding the usability of Faceboard. The previously introduced hypotheses are described in the upcoming section and success criteria and benchmarks for usability are introduced.

6.1 Hypotheses

Each hypothesis validated a different aspect of the proof of concept.

6.1.1 H1 Series - Text-Entry Metrics Comparisons

The first hypothesis-series aimed to validate the core functionality of the app - text input. It was verified by checking if the users were able to produce meaningful text input using head movements and gestures as described. The hypothesis was validated by analyzing the transcribed text and the corresponding input metrics and checking if they actually corresponded with the shown phrases the participants were supposed to transcribe. For comparison, the previously proposed metrics (see Table 4.1) were checked against established input methods and other works.

H1.1: Text entry speed differs between the prototype and users' usual input method.

Here, it was examined if the prototype and users' usual input method differ in comparison to entry speed. It served as a primary indicator of the input method's speed.

H1.2: Text entry efficiency differs between the prototype and users' usual input method.

The keystrokes per character metric (KSPC) reflects how many keystrokes are required to create correct input. This metric helped to evaluate whether Faceboard is efficient or not and how efficient it is compared to the usual input method.

H1.3: Text entry error rate differs between the prototype and users' usual input method.

The Corrected Error Rate (CER) reflects the amount of errors fixed relative to the total number of keystrokes. With the corrected error rate, it can be calculated how many errors were fixed, not just the remaining, unfixed mistakes in the final transcription. It allowed to identify whether typing on the prototype leads to fewer or more mistakes compared to the usual input method.

6.1.2 H2 Series - Learning effects

The second hypothesis-series aimed to find possible learning effects. Since each participant did the experiment twice, comparing the metrics of each experiment revealed changes in performance.

H2.1: Text entry speed improves for users that repeated the experiment with the prototype.

With hypothesis H2.1, a possible speed improvement for each participant after gaining experience with Faceboard was checked.

H2.2: Text entry efficiency improves for users who repeated the experiment with the prototype.

The keystrokes per character metric (KSPC) might improve with subsequent tries of the experiment which would show a positive learning curve due to familiarity with the system.

H2.3: Text entry error rate decreases for users that repeated the experiment with the prototype.

This hypothesis assessed whether the users had to correct fewer errors on their second runs compared to the first runs. Again, this would show a learning effect for subsequent tries.

Together, the hypotheses formed the base evaluation of the performance and usability of Faceboard.

6.2 Quantitative Evaluation

After finishing the studies with the participants, the data provided by the users and collected by the app was used to answer the hypotheses as well as answer other questions regarding the usability, effectiveness and performance of the app and its users. By combining different data collected during the process, a better understanding of the systems' potential for practical usage was derived. To make sure that the study outcomes are relevant, we checked them for their statistical significance using comparative as well as correlation and regression analysis.

6.2.1 Data analysis and combination

For data analysis, a multi-modal approach was used. Qualitative and quantitative data collected were used for evaluation. Combining both measuring types gained a comprehensive view of the participants' interactions with Facebook and set it in relation to their usual behavior as well as showed strengths and limitations.

Performance Metrics To measure the efficiency of the participants, performance metrics were used. The quantitative data described in section 'Methodology' like words per minute for text entry speed, keystrokes per character and corrected error rate allow statistical analysis. This is a good first indicator of the usability of the newly proposed approach. Performance metrics also allowed to check and compare related work to prove that the proposed method is feasible.

The performance metrics first were analyzed for each participant separately by checking averages, maximum, and minimum. Then, a comparative analysis between the initial and subsequent rounds of playing was conducted. Finally, the participants were analyzed in relation to each other. By this, possible outliers were identified and removed.

System data The gathered system data like the `installation_id` was used to enable matching of participants' different rounds. By this, statistics about the runtime of each experiment could be derived. It allowed to further look into learning effects between the rounds each participant took.

Transcription data Simple analysis on the transcribed phrases was conducted. This included the number of phrases transcribed in total. Word-by-word analysis was done. For this, the different words were set into relation with each other for word length as well as how often they got deleted. By this quantitative measurement, interesting findings about relations between word length and deletion were analyzed.

To simplify the process, Jamovi, the 'open statistical software for the desktop and cloud' which allows simple and fast processing of the collected data, was used [Jam] [rpr].

6.3 Questionnaires

Questionnaires are an important aspect of the evaluation of user-centered applications since it allows to capture users' perceptions on the usability of the Faceboard system.

6.3.1 System Usability Scale - SUS

For this study, the outcomes of the SUS were analyzed in two ways: The overall system usability and question by question. For the SUS score, measures such as mean and median were calculated. To check variability, the standard deviation, minimum and maximum were calculated to check the amount of scattering in the qualitative rating of Faceboard.

For deeper insight, the 10 standardized questions were analyzed on their own. Again, the mean and median were calculated to identify the overall trend of how the participants felt about the proposed system. Additionally, each question was interpreted together with the participants' feedback gathered during the final interview to contextualize the different scores from the SUS statements.

6.3.2 NASA-TLX

The outcomes of NASA's TLX questionnaire allow workload ratings across six different dimensions. In this study, the outcomes of the participants' studies were again first analyzed on dimension level. For this, the rating of each (non-weighted) dimension was analyzed for tendencies by calculating mean and median as well as variability.

Then, descriptive analysis was conducted over the weightings of the different dimensions. This allowed to check for possible tendencies on dimensions that had a major influence on the overall workload. For example, high weights in the Frustration dimension would mean that, when frustration did occur, it had a significant impact on the workload.

The weights and the scores of the dimensions are then calculated to the weighted workload score. This score then gets compared to related literature to benchmark the newly proposed solution with already existing ones.

Correlation analysis between the different dimensions was conducted to find potential significant relationships. For example, a strong correlation between Physical Demand and Frustration could indicate that participants, who found the head movement physically demanding also felt more frustrated using Faceboard.

6.3.3 Concluding Interview

The concluding interview enabled an additional depth of information to the feedback gathered by the questionnaires. To allow a holistic understanding of the users' experiences, the additional feedback gathered by the unstructured interview allowed further understanding and support of the outcomes of the SUS and NASA-TLX.

CHAPTER 7

Results

Each of the 16 participants successfully completed the study, but the outcome differed greatly between the test subjects. First and foremost, Android is very restrictive in the way apps can interact with the system, it can be a rather complicated process to enable accessibility services for users without a technical background. However, all participants were able to set up the Faceboard accessibility service without any help just using the on-screen help and tutorials (Non-functional requirement **NFR1**). Some mentioned that it is 'unnecessarily complicated' to set up the accessibility service.

Where some users had trouble with the blinking algorithm and could not reliably finish words as they wanted to, others had no problems at all and rushed through the steps of the study as planned. To allow more accurate analysis, two outliers have been removed: one person had persistent problems with face- as well as blinking recognition which resulted in rather bad metrics (less than 3 words per minute and more than 4 keystrokes per character since words did never come out as planned) and one outlier which had very great performance (more than 11 words per minute and less than 1.1 keystrokes per character).

7.1 Results - Faceboard

In total, the users have written 584 phrases in total. 36 of the phrases have had errors in them. Since the experiment finishes by itself, this means, that out of 584 times, the participants gave up on finishing a phrase 36 times. Since it is a small number of phrases for each participant, they did not get looked into. The average experiment with the new proof of concept input method took 32 minutes, with the shortest being done in 15 minutes and 47 seconds and the longest experiment taking 1 hour and 1 minute.

7. RESULTS

Correlation Matrix			
		wordlength	frequency_deleted
wordlength	Pearson's r	—	-0.851
	df	—	16
	p-Wert	—	<0.001
frequency_deleted	Pearson's r	-0.851	—
	df	16	—
	p-Wert	<0.001	—

Note. Ha is negative correlation

Table 7.1: correlation between word length and frequency deleted

The first thing discovered during the studies, the participants seemed to have more problems writing short words than long words. This resulted in a lot more deletions for short words than for long words, as seen in 7.1.

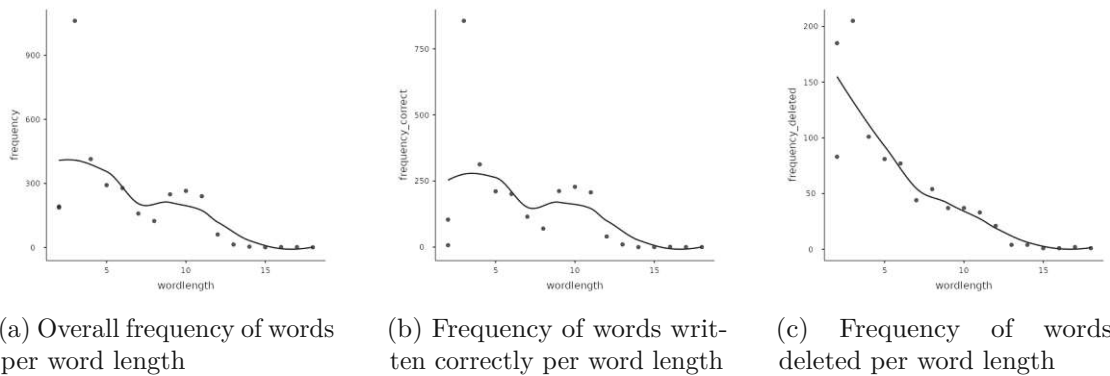


Figure 7.1: Comparison of word frequencies for Faceboard input method - short words seem to be more error-prone

As expected, there is a negative correlation between the word length and the frequency with which the words were deleted (see table 7.1). Although short words appear more often in the written text, the frequency of short words being deleted exceeds the overall word frequency.

The participants reached an **average of 5.98 words per minute**. The fastest user reached 8.52 wpm, whereas the slowest user could only type text with 3.66wpm. The **average KSPC of 1.67** reports a rather high value for keystrokes per character, which means that participants had to delete and rewrite words rather often. The average **corrected error rate of 18.9%** shows that a lot of input was wrongly created and had to be corrected to correctly transcribe the text.

Faceboard descriptive statistics			
	wpm_avg	kspc_avg	cer_avg
N	28	28	28
Median	5.98	1.67	18.9
Standard deviation	1.41	0.514	6.91
Varianz	1.98	0.264	47.7
Minimum	3.66	1.04	5.03
Maximum	8.52	3.56	29.3

Table 7.2: Different metrics averaged over all runs

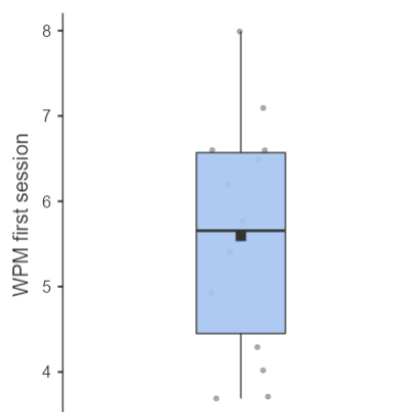
Descriptives	WPM 1st	WPM 2nd	KSPC	KSPC 2nd	CER 1st	CER 2nd
N	14	14	14	14	14	14
Mean	5.59	6.31	1.87	1.54	20.8	17.0
Standard deviation	1.33	1.44	0.606	0.351	6.51	7.00
Minimum	3.69	3.66	1.17	1.04	9.24	5.03
Maximum	7.99	8.52	3.56	2.17	29.3	26.5

Table 7.3: Participants get better result on second round compared to first round

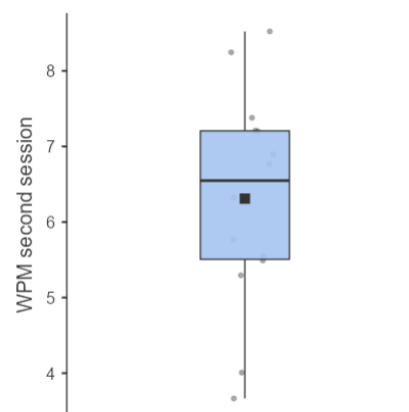
As expected, the figure 7.3 shows that on average, users performed better on their second attempt playing Hyper Typer using Faceboard. Also, the box plots of the two sessions show a slight shift of distribution which means general improvement across participants (see figure 7.2). The average words per minute increased by 12.8%, whereas the keystrokes per character decreased by 21%. The participants tended to make fewer mistakes on their second run paired with faster input rates. The paired samples T-Test proves the results as significant ($p=0.044$ for WPM and $p=0.039$ for KSPC) (see table 7.4). This also overlaps with some of the participants' feedback describing the second round of Faceboard as 'easier' and 'less frustrating' than the first round. Fewer extreme outliers and data shift downwards relative to the first session in terms of KSPC emphasizes increased efficiency (see figure 7.3). Furthermore, this confirms the initial assumption that there is a noticeable learning effect when there is more than one try.

This proves the hypotheses **H2.1** through **H2.3** of the **H2** series: On average, there was a positive learning effect for the participants regarding entry speed, error rate and keystrokes per character.

7. RESULTS



(a) average word per minute of first sessions



(b) average word per minute of second sessions

Figure 7.2: Box plots of words per minute for each of the two sessions

Paired Samples T-Test

			statistic	df	p
WPM 2nd	WPM 1st	Student's t	1.84	13.0	0.044
KSPC 1st	KSPC 2nd	Student's t	1.92	13.0	0.039
CER 1st	CER 2nd	Student's t	1.87	13.0	0.042

Note. $H_{\text{Measure 1 - Measure 2} > 0}$

Table 7.4: Participants tended to get better result on second rounds

7.1.1 Interesting Players

How much the results varied, can be best described with the two outlying participants.

Best performing player

Reaching a speed of 11 words per minute and just 1.1 keystrokes per character, the fastest and most reliable participant had almost no problem writing with the new approach. Compared to his standard input method and a speed of 54 words per minute (and an average KSPC of 1.21), it still is rather slow but shows the possibilities of the proof of concept app.

Player with problems

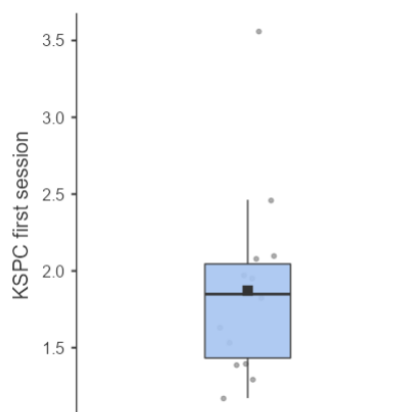
One player - the participant with rheumatoid arthritis - had problems with face recognition which made it hard for them to input text at all. Using the same height-adjustable desk as well as the same ring light on the same desk, the app just could not recognize the blinking. There seems to be a bias in the training data for the face recognition which rendered the app basically useless for this participant. It resulted in input rates of less than 3 words per minute and a KSPC over 4. With 36 words per minute on her usual input method and a KSPC of 1.2, the difference between her typing and her using Faceboard was tremendous. The problems that occurred were not connected to the rheumatoid arthritis.

Players with Relevant Restrictions

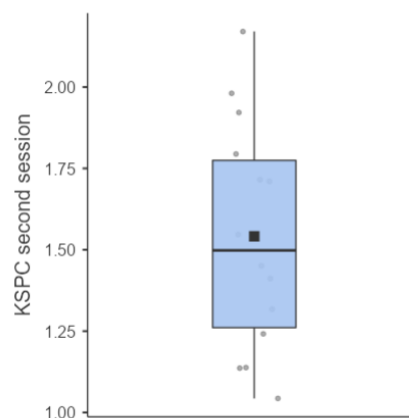
The two remaining participants with relevant restrictions did finish the experiment as intended without any major problems. For both players, there was **no significant improvement** in terms of typing speed between the two runs. There were slight improvements in terms of KSPC (2.2 vs 2.05), but still no significant change. The corrected error rate increased (23.8 vs 25.5%). Also, the two restricted participants were at the lower end of the performance range.

Descriptives - Players with Relevant Restrictions

	WPM 1st	WPM 2nd	KSPC 1st	KSPC 2nd	CER 1st	CER 2nd
N	2	2	2	2	2	2
Mean	3.86	3.84	2.20	2.05	23.8	25.5



(a) average keystrokes per character of first sessions



(b) average keystrokes per character of second sessions

Figure 7.3: Box plots of keystrokes per character for each of the two sessions

7.2 Results - Usual Input Method

For the usual input method, 9 of the 14 relevant participants choose the more common typing on the keyboard, whereas 5 of the participants prefer swiping on their keyboard instead of typing (7.7).

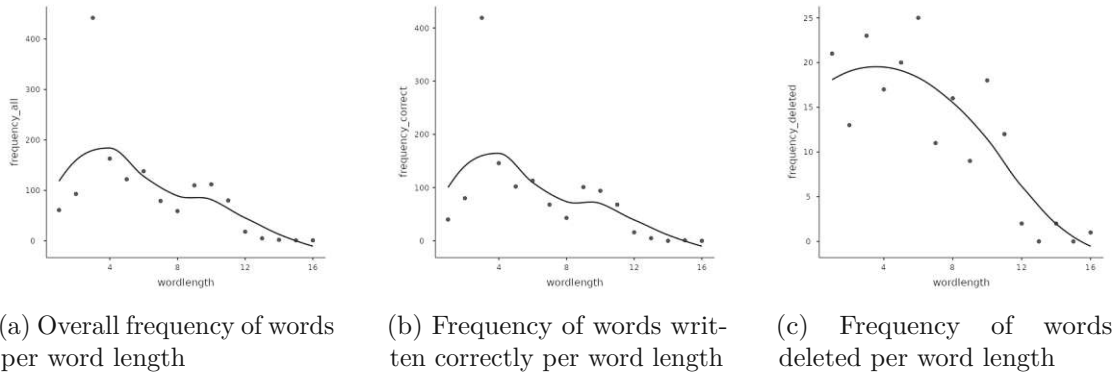


Figure 7.4: Comparison of word frequencies for usual input method

The metrics for the users' usual input method highly differ from the results generated with Faceboard as expected. With input speeds of up to 54.2wpm and a maximum of just 1.62kspc, the users not only wrote a lot faster, it was also less error-prone. By this, the hypotheses **H1.1** through **H1.3** of the **H1** series are met negatively: On average, the participants wrote a lot slower and more error-prone on Faceboard compared to their usual input methods.

Descriptives - usual input			
	wpm_avg	kspc_avg	cer_avg
N	14	14	14
Average	36.6	1.29	10.5
Median	39.6	1.29	11.3
Standard deviation	11.2	0.186	4.97
Minimum	18.3	1.02	1.78
Maximum	54.2	1.62	17.5

Table 7.5: Averages over all participants - usual input method

As already noted during the Faceboard runs of the experiment, there seems to be a correlation between the deletion of words and their respective length. There is also a significant negative correlation (see figure 7.4) for the played games with usual input methods but with a little weaker correlation (see table 7.6) compared to the runs the users played with Faceboard.

Correlation Matrix		wordlength	frequency_deleted
wordlength	Pearson's r	—	
	df	—	
	p-value	—	
frequency_deleted	Pearson's r	-0.830	—
	df	14	—
	p-value	<0.001	—

Note. Ha is negative correlation

Table 7.6: correlation between wordlength and frequency deleted for usual input method

Descriptives - usual input methods split by method				
	input_method	wpm_avg	kspc_avg	cer_avg
N	swiping	5	5	5
	typing	9	9	9
Average	swiping	37.7	1.34	12.1
	typing	36.0	1.26	9.56
Median	swiping	41.1	1.37	13.4
	typing	38.2	1.21	10.4
Standard deviation	swiping	13.2	0.221	6.04
	typing	10.8	0.172	4.38
Minimum	swiping	19.4	1.04	3.14
	typing	18.3	1.02	1.78
Maximum	swiping	54.2	1.62	17.5
	typing	48.4	1.50	14.4

Table 7.7: Averages over all participants - usual input method split by swiping and typing

Correlation Matrix		wpm_avg_faceboard	wpm_avg_usual
wpm_avg_faceboard	Pearson's r	—	
	df	—	
	p-value	—	
wpm_avg_usual	Pearson's r	0.536	—
	df	12	—
	p-value	0.024	—

Note. H is positive correlation

Table 7.8: Correlation between WPM averages between Faceboard and usual input methods

Descriptives - Faceboard split by swiping and typing				
	input_method	wpm_avg	kspc_avg	cer_avg
N	swipe	4	4	4
	type	10	10	10
Mean	swipe	6.29	1.48	15.6
	type	6.32	1.57	17.5
Median	swipe	6.33	1.48	15.8
	type	6.55	1.58	18.9
Standard deviation	swipe	0.904	0.201	4.20
	type	1.64	0.402	7.98
Minimum	swipe	5.30	1.24	10.7
	type	3.66	1.04	5.03
Maximum	swipe	7.20	1.72	20.2
	type	8.52	2.17	26.5

Table 7.9: Averages over all participants - Faceboard split by swiping and typing

As figure 7.8 suggests, there is a statistically significant positive correlation between the words per minute generated with Faceboard and the usual input methods. This means, participants, which were faster using Faceboard, are also faster using their usual input method. The statistical significance of $p = 0.024$ supports the correctness of this correlation.

As argued before, the statement that participants, which use swiping as usual input method would perform better than people using typing, could not be supported ($p = 0.511$). There is no significant difference between the two classes (see figure 7.9). Notable, the fastest Faceboard users (besides the outlier) are all using typing as their usual input method.

Questionnaires

Since every participant filled out the System Usability Scale as well as the NASA-TLX questionnaire, deeper insight into the usability of the proof of concept has been gathered and how the participants experienced it. It also increases comparability with existing solutions discussed in the related work section.

8.1 System Usability Scale - SUS

System Usability Scale	
	Total
Mean	59.1
Median	57.5
Standard deviation	20.0
Minimum	27.5
Maximum	97.5

Table 8.1: Participants' SUS score

The average score the system received was 59.1 points. According to Bangor et al., this suggests a moderate level of usability[BKM09]. In Sauro et al. study, they find that the generally acceptable usability threshold is 68 for SUS scores [SotScoh11]. With a standard deviation of 20, a high variation between the users' qualitative rating of the usability of the proof of concept can be seen.

SUS question analysis		
	Mean	Median
I think that I would like to use this system frequently	0.857	0.500
I found the system unnecessarily complex	1.071	1.000
I thought the system was easy to use	2.214	2.000
I think that I would need the support of a technical person to	0.857	1.000
I found the various functions in this system were well integrat	2.643	2.000
I thought there was too much inconsistency in this system	2.071	2.000
I would imagine that most people would learn to use this system	3.143	3.000
I found the system very cumbersome to use	2.571	3.000
I felt very confident using the system	2.071	2.500
I needed to learn a lot of things before I could get going with	0.714	0.500

Table 8.2: Participants SUS question per question analysis

8.1.1 Question by Question Analysis

I think that I would like to use this system frequently

The users were pretty clear about using the system frequently, with a mean of just 0.857 indicating that they strongly agree to disagree. As some of the participants stated, it has to do with the fact that most are healthy ('probably would not use it as long as I am healthy', 'Still would not use it as long as my hands are healthy') and do not have problems with typing on their phones or the performance was not good enough to think about using it('It would be good if the performance were better.'). One of the motion-impaired users mentioned she would use the app if she had 'really bad days when my fingers really hurt [...]').

I found the system unnecessarily complex

The study participants did not find the system complex.

I thought the system was easy to use

The users thought the system was relatively easy to use, especially on the second try ('This time it was a lot easier and also I was a lot less frustrated', 'For the use case, the app itself is not overly complicated. It actually is good'). Still, there is room to improve as people suggested the calibration was to manage ('I think it is hard to reset the cursor to the middle' and 'I find the calibration rather cumbersome to use. A simple button or gestures would be much easier').

I think that I would need the support of a technical person to use this system

The participants think, that the app is rather self-explaining.

I found the various functions in this system were well integrated

In general, the test subjects agreed that the functions were well integrated, but they did have some recommendations on how they could be better ('It would probably be a good idea if you would not have to move the cursor up all the way to recalibrate but to have a visible barrier instead', 'Calibration is important for this input method, so I would rethink how it works').

I thought there was too much inconsistency in this system

With a mean of 2.071, the users agreed that there was a moderate amount of inconsistency in the proof of concept. In the additional feedback, there is a lot of complaining about short words being hard to write ('I had the most problems with really short words'), and for some faces, the cursor moves down when the eyes are closed, which makes it hard to write words correctly for some users ('The cursor moves down when I close my eyes, that was annoying'), which seems to be a problem of the facial recognition since it did not happen for all faces and the head itself clearly did not move. Users also tended to accidentally delete words by stretching their necks or resting their eyes during tries ('I accidentally deleted the correct words. Very frustrating').

I would imagine that most people would learn to use this system very quickly

The participants tend to think that one could learn to use this system very quickly. As we've proven in 7.4, a learning effect was there after just two sessions of 5 rounds ('I was a lot more frustrated on the first try, the second try was a lot better and also I felt like I performed a lot better.', 'Already a lot easier on second try').

I found the system very cumbersome to use

Again, the participants tended to agree with the question. Mostly, the calibration was stated as cumbersome ('I find the calibration rather cumbersome to use') and users mentioned eye strain a lot ('It was exhausting for the eyes', 'Sometimes, I increased the pressure to close my eyes for no reason, even if it was working with less force', 'After a long day at work this is really exhausting to use'). The two participants with motion impairment also stated problems with the neck having to look up too far ('I might feel dizzy by using it too much, '[...] it was a little bit exhausting and also bad on the neck').

I felt very confident using the system

The users had mixed feelings about their confidence in using the Faceboard system. On the first try, the users mostly did not feel confident but increased on the second try ('[...] I felt like I performed a lot better. I think it mostly was because I knew what I was doing on the second try', 'This time it was a lot easier and also I was a lot less frustrated'). Some of the participants forgot what word they had to write during a gesture since it takes longer than writing it by hand ('I sometimes forgot which letters I currently was swiping and which would come next since the gestures take so long', 'I had to search for the letters and sometimes couldn't find them, even though I type on a keyboard every day.').

I needed to learn a lot of things before I could get going with this system

The users disagreed mostly on this one and some even stated, that you could get better by trying more often ('I think, with a lot of practice, it could work out pretty well').

8.2 NASA TLX

The average weighted workload score of the proof of concept app is 58. Hertzum et al. reported a mean weighted workload of 42 with a standard deviation of 13 in a meta-study about NASA-TLX [Eu21]. With an average weighted workload of 58 and a standard deviation of 22.5, the participants rate the workload higher and with more deviation (see table 8.4). Also, the dimension which was weighted the highest, was highly variable between the participants. There are also significant correlations between different dimensions. High physical demand strongly correlates with temporal demand and with increased effort and frustration.

8.2.1 Dimension Analysis

As mentioned before, the participants' subjective weighting of the dimensions is almost spread evenly with a slightly higher rating for the dimension **Effort** (see table 8.3).

NASA TLX - Average weighting from 0 to 5						
	MD	PD	TD	P	EFF	FR
Mean	2.86	2.36	2.36	2.43	2.71	2.29
Median	3.00	2.00	2.50	2.50	3.00	2.00
Standard deviation	1.56	1.78	1.34	1.70	1.54	2.05

Table 8.3: Participants TLX dimension weighting

Mental Demand (MD)

Compared to the average mental demand of Hertzum et al., the proof of concept got a higher-than-average rating (49 vs 52.1). Some participants found it hard to focus on the word they had to write, since recording the gesture with head movements can take longer than writing it by hand ('I sometimes forgot which letters I currently was swiping and which would come next since the gestures take so long').

Physical Demand (PD)

With an average of 53.2 points, the physical demand was the second highest weighted dimension as well as the dimension with the highest deviation from the mean TLX-score (32) which suggests that the participants found the system physically demanding. The Feedback gathered during and after the experiments hardened this value, since a lot of users reported eye strain caused by the blinking algorithm ('I closed my eyes too hard [...] This resulted in a lot of tears and eye pain', 'After a long day at work this is really exhausting to use'). Some users also tended to close their eyes harder than needed during the experiment which made it even more exhausting to use ('Sometimes, I increased the pressure to close my eyes for no reason, even if it was working with less force').

For the two motion-impaired users, the physical demand was rated low, compared to the mean value, even though they reported a bad impact on their cervical spine ('[...] it was a little bit exhausting and also bad on the neck.' and '[...] maybe the head movement is a bit too much and I might feel dizzy by using it too much.'). One reason could be that people with chronic pain tend to rate pain/discomfort differently than healthy individuals [Tur99].

Temporal Demand (TD)

The Temporal Demand was again, higher rated than the average system. Since participation in the study involved playing a game that primarily shows the users' words per minute as a score, the users might tend to input text as fast as possible. Also, the in-game music of Hyper Typer was stressing some of the participants, where others were motivated by it ('[...] the music in the Hyper Typer App made the temporal demand on the study higher, once we turned it off, it was a lot easier on the mind to not rush into writing the words.' and '[...] also the music motivated me').

There is also a strong correlation between Temporal Demand and Physical Demand which suggests that participants, who think that the study was physically demanding also tend to think the temporal demand was high (see table 8.5).

Performance (P)

Participants of the experiment rate their performance equal to the TLX average (45) with a higher deviation (29.4 vs 19).

Effort (EF)

Being significantly above the average, participants found the effort to transcribe sentences with the Facebook service. This also aligns with the strong correlation to physical and mental demand.

Frustration (FR)

Frustration level was rated above TLX average (36 vs 43.2). Considering the strong correlation with temporal demand dimension as well as moderate correlations to physical and mental demand, one can argue that people tended to get frustrated over time by increasing stress on their eyes or neck. This indicates that longer sessions may lead to discomfort for some users.

Weighted Work Load (WWL)

With the overall Weighted Work Load Index being 58 and above the average of 42, there is room for improvement. The combination of physical demand caused by the strain on the neck and eyes in combination with temporal demand and high correlation with frustration could explain the high workload with the amount of times inputs with the proof of concept took.

8.3 Additional Feedback

Each participant was asked to provide additional feedback throughout the process to gain more insights and unique perspectives on how the users coped with the newly proposed input method.

8.3.1 Initial Impressions and Learning Curve

Since a blinking and head-movement-based entry method was a novelty for each participant, the initial impressions and learning curve came out quite similar. On the first interaction, most participants were skeptical, and first interactions were stated with *'It was very exhausting and it works really bad'* or *'I like the concept but it is not working that good'*. Despite the first feedback, many of the users noted on the second try, it felt significantly easier. One individual summarized it as *'I was a lot more frustrated on the first try, the second try was a lot better, and also I felt like I performed a lot better.'* The participants showed a learning curve which is backed by quantitative data.

8.3.2 Physical and Cognitive Demands

A recurring theme was the physical strain associated with the head movement and blinking. Users reported that they had to move their heads extensively or had to close their eyes too hard to make the blinking recognition work.

One participant reported that looking upward to recalibrate the cursor was 'a little painful'. Also, the participant mentioned that *'The distance I have to move my head is too far, I would prefer it to be lot less than the current version'*.

Several users reported eye strain caused by the blinking-recognition. Tearing, eye pain, and discomfort were a recurring problem for the participants. Users reported *'I closed my eyes too hard in the beginning because I thought otherwise the camera would not recognize it. This resulted in a lot of tears and eye pain.'* and *'Sometimes, I increased the pressure to close my eyes for no reason, even if it was working with less force.'* Again, a learning effect could be seen as the user independently found out that using force to shut their eyes is not necessary.

Participants also noted that for extended periods of time, using Faceboard was tiring. One user reported Faceboard to be exhausting *'after a long day at work'* and another one reported *'It hurts my eyes to play for this long'*

Some users mentioned high memory load like problems with forgetting letters in the middle of a gesture or the phrase they were about to write (*'I kept forgetting the word I was writing during the process of a gesture which was exhausting.'*)

8.3.3 Interaction and Calibration

Interacting with Faceboard was cumbersome for some of the participants. Fear of accidental input was a recurring issue and mentioned: *'During gestures, I sometimes was afraid to blink [...] I was scared to blink and have to do the whole gesture again'*.

The distance that was necessary to move the cursor to write words was too big for some users (*'The distance I have to move my head is too far.'*). This also applied to recalibrating the cursor in case it got out of sync with the head movements. Users found it cumbersome to move the cursor all the way up to the top of the display to vertically recalibrate cursor (*'Calibration is important for this input method, so I would rethink how it works'* and *'It would probably be a good idea, if you wouldn't have to move the cursor up all the way to recalibrate but to have a visible barrier instead.'*).

8.3.4 Usability and Performance

The overall qualitative rating of usability and performance was mixed: One user highlighted the positivity on the swiping algorithm's capability to correct small errors (*'It is good that you do not have to be as precise as possible since the swipe algorithm fixes your little mistakes.'*), where others found it hard to find letters on the keyboard.

The participants generally stated they would rather not use Faceboard as their go-to input method, however, they acknowledged the potential benefits for users with motion impairments.

NASA TLX							
	MD	PD	TD	P	EFF	FR	WWL
Mean	52.1	53.2	52.9	45.0	62.1	43.2	58.0
Median	52.5	60.0	57.5	37.5	75.0	45.0	64.5
Sum	730	745	740	630	870	605	812
Standard deviation	28.9	26.9	28.5	29.4	30.3	27.7	22.5
Minimum	5	15	10	5	5	5	17.3
Maximum	100	85	95	85	95	80	84.7

Table 8.4: Participants TLX dimension statistics

NASA TLX Dimension correlations							
		MD	PD	TD	P	EFF	FR
MD	Pearson's r	—					
	df	—					
	p-value	—					
PD	Pearson's r	-0.027	—				
	df	12	—				
	p-value	0.927	—				
TD	Pearson's r	0.210	0.777	—			
	df	12	12	—			
	p-value	0.472	0.001	—			
P	Pearson's r	-0.215	0.444	0.285	—		
	df	12	12	12	—		
	p-value	0.460	0.112	0.324	—		
EFF	Pearson's r	0.533	0.538	0.541	0.168	—	
	df	12	12	12	12	—	
	p-value	0.050	0.047	0.046	0.565	—	
FR	Pearson's r	0.474	0.539	0.570	0.330	0.337	—
	df	12	12	12	12	12	—
	p-value	0.087	0.047	0.033	0.249	0.239	—

Table 8.5: TLX dimension correlations

CHAPTER 9

Discussion

Looking at the data extracted from the studies, the feedback and the questionnaires and comparing it to related works, several key aspects of usability can be derived and allow to make claims on the validity of the conducted experiments. The two research goals are recalled:

- RG1** Develop a functional proof of concept application to demonstrate the technical feasibility of the proposed input method.
- RG2** Conduct an experiment using the developed prototype and the users' standard input method to assess the practical usability in a comparable manner.

To check whether the research goals were reached, following hypotheses were drawn up:

- H1.1** Text entry speed (measured in words per minute, WPM) differs between the prototype and users' usual input method.
- H1.2** Text entry efficiency (measured in keystrokes per character, KSPC) differs between the prototype and users' usual input method.
- H1.3** Text entry error rate (measured as Corrected Error Rate) differs between the prototype and users' usual input method.
- H2.1** Text entry speed improves for users that repeated the experiment with the prototype.
- H2.2** Text entry efficiency improves for users that repeated the experiment with the prototype.
- H2.3** Text entry error rate decreases for users that repeated the experiment with the prototype.

Since all hypotheses were answered positively during this thesis, both goals were reached within the scope of this thesis: With the first user-tests, a functional proof of concept application was created and the technical feasibility of the proposed input method was shown (**RG1**). The second research goal, to conducting an experiment to assess the practical usability in a comparable manner was also reached (**RG2**).

The most similar approach to Faceboard would be EyeSwipe: They also utilize swiping but use eye-tracking devices instead of head movement [KFJ⁺16]. In terms of input speed using swipe, EyeSwipe averaged at 11.7wpm, where the proof of concept reached about 6wpm on average. One could justify the gap with the difference in accuracy from the specialized eye-tracking device compared to calculating the heads' position from camera inputs as well as faster cursor movement by gaze compared to head movements. In terms of error rate, where Faceboard averaged at 18.9%, EyeSwipe had a similar rate of 13.25%. Just like using Faceboard, users of EyeSwipe tend to get better on second sessions, for correction rate as well as words per minute.

Similar results have been collected during Majaranta et al., a study on a dwell-time eye-tracking solution [MAotSu09]. In their experiment, they did 10 sessions of text input and also saw a significant increase in speed, but at the same time increase in errors made. With a KSPC value of 1.18 and WPM values of 19.9, their method performed better and less error-prone, but in a comparable scope.

One could speculate that with more than two sessions, users might reach higher speeds and lower error rates using Faceboard. Since Majaranta et al. saw decelerated learning after the fourth session and Yu et al. similarly saw decreasing effects after the 5th or 6th session [YGY⁺17], there might be a plateau after a certain number of sessions. Additionally, the skill of controlling a cursor via head movement and blinking is not something people have tried before, which could result in higher learning effects in subsequent sessions compared to gaze-based input methods.

Tiltwriter, using the approach to move the cursor by tilting the device, created input rates at 12.1wpm [CMM⁺19]. The error rate, with only 0.76% for QWERTY-based keyboards, is extremely low compared to other approaches and also a KSPC value of only 1.0298 means almost negligible false input. With Faceboard reaching about half the input rates at higher error rates, there is room for improvement.

Even though having higher error rates and lower input speed, people who cannot use their hands due to tremors or rheumatoid arthritis might find Faceboard more accessible.

The *Swipe-like text entry by head movement and a single row keyboard* concept proposed by Nowosielski et al. shows a speed of 33.7 characters per minute (CPM) on average [Now16]. Translated to the common definition of words per minute (WPM) as 5CPM, about 6.74WPM on average for this approach was calculated. Interestingly, the performance of both head-based input methods is on par, even though Nowosielski wrote about acquainted users, whereas Faceboards' is an average value-based on users who were not familiar with the system. Based on this knowledge, with further improvements, Faceboard and other proof of concepts could push the upper limits of head-movement-based keyboards higher.

9.1 Influence of Short vs. Long Words

It was observed that shorter words led to significantly more deletions than longer words.

This could be interpreted in several ways: There is a possibility that shorter swipe paths are easily 'overdone' with head movements since small gestures are harder to control and keep error-free than long paths. Another possibility is that predictive text algorithms have more problems recognizing short words than long words since for short words there is less data for correct guessing.

9.2 Possibilities and Struggles

The two outliers show the possibilities as well as the struggles of the proposed system. The best-performing participant did not struggle at all and could reach 11 words per minute (WPM) and 1.1 keystrokes per character (KSPC). This suggests the potential of Faceboard with further refinement or if users received enough practice.

On the other hand, the participant struggling with the system could barely input text as the blinking was not recognized. Since the lighting conditions were equal for all participants, possible environmental problems can be excluded. This clearly shows the limitation of a blink-detection-based algorithm: There is a need to include a robust calibration phase for the users' face or better a training set for the machine-learning module without biased data.

9.3 Accessibility and Practical Constraints

All participants were able to enable and start the accessibility service on their own. Still, some users reported it to be 'unnecessarily complicated'. This reflects the real-world problems with assistive technologies on Android, it can be cumbersome to enable some of them for people without help.

On the practical side, the proof of concept is usable in a laboratory environment. For real-world usage, it is not applicable due to the application relying on a certain amount of stability and lighting to work as intended.

Also, the machine-learning modules' performance highly varies between different phones. This is a possible trade-off: More sophisticated AI-based recognition could highly improve accuracy but at the same time requires more processing power which means the entry barrier gets raised again. In the landscape of assistive input methods, Faceboards' advantage is to solely rely on commonly available front-facing cameras which makes Faceboard more cost-effective and portable compared to specialized eye-tracking hardware.

9.4 Data quality

In a study by Palin et al., they collected and analyzed the input data of 37,000 volunteers and came to a result of about 1.12kspc and 36.17wpm for the participants preferred input method [PZK⁺19]. Compared to the analysis of figure 7.5 and the results of the experiment carried out by Schlögl et al. (33.6wpm and 1.11kspc), the studies' rather small dataset and number of participants are similar in the case of metrics [Sch20]. For the gathered data, identification and removal of outliers with unusually high performance on Faceboard and also their usual input method and one with persistent face-tracking problems made the data more representative and helped ensure that the averages reported reflect typical user experience with less deviation caused by anomalies. It can be concluded that a rather small size of participants still can generate significant and relevant data with acceptable quality.

9.5 Participant Selection

Since the study was designed as a laboratory experiment, the recruitment for the participants was hand-selected. The demography of the participants shows a broad range of different backgrounds and ages spanning from 22 to 56 years. Varying occupations outside the technical space show that a technical background is not necessary to configure and use the accessibility service on Android, even though the process is more complicated than the usual installation of an application. It can be seen that the implementation is suitable independently of demographic characteristics.

The three participants with health problems had no major problems with the proof of concept. Two of them reported some strain on the neck due to the head movement, but still, it was usable for them. The person with rheumatoid arthritis had no problems in connection with their condition when using Faceboard.

The 5 participants wearing glasses all had problems when using their glasses, especially with the blinking-recognition. Without their glasses, the blinking algorithm was working without major problems.

9.6 Participant Feedback

In general, most participants saw the potential and noted that users with motor impairments might profit from the new approach. However, healthy users would not adopt Faceboard into their daily lives as it does not bring significant improvements in comfort and speed. One participant got to the heart of the matter: *'If I wouldn't have two healthy arms, I would probably give it a try as my main type of input, but right now, I would rather use my fingers.'* Fatigue and exhaustion after using Faceboard were common problems for most of the participants in the study, especially when the session lasted longer. The participants' feedback provided inspiration for extensions and improvements, which are part of the discussion in the next chapter '**Conclusion**'.

CHAPTER 10

Conclusion

With Faceboard, a new input method using gesture-based swiping was proposed. For this, the users had to utilize head movements and blinking to interact with a cursor on the screen. By mapping the head movement to the display of the mobile phone using a virtual cursor, the users can draw swipe-like gestures to input text. For this, users start recording a gesture by blinking. Then, the swipe gesture can be drawn. After blinking a second time, the gesture is completed and will get mapped on the device's screen, writing a word on the mobile device. In case of an error, the last written word can be deleted using a separate area reachable with the virtual cursor.

The major motivations behind Faceboard were accessibility and availability. Research has shown that usual input methods are often not suitable for people with disabilities. There are a lot of solutions available that often rely on third-party accessories like eye-tracking devices or, even harder to get, a brain-computer interface. By utilizing the camera of the smartphone (which most likely is part of every newer phone), the entry barrier for users is as low as it can get.

To prove the feasibility of this new approach, a within-subject study with 16 participants was executed. All 16 persons were able to complete the tasks, but high variation in the results was shown. Some users had no problems at all using the proposed method, others could not get along with the blinking mechanics to start and stop gestures, which led to unreliable wording. One participant even had problems with their face not being recognized which seems to be a bias in the training data of the machine-learning library. The app works moderately as a text input method. The two outliers show highly varying data with input rates from 8.52 words per minute (WPM) and 1.1 keystrokes per character (KSPC) to only 3WPM and more than 4KSPC. With an average typing speed of 5.98WPM there is room for improvement. Nevertheless, there was a learning effect on the second run with an average increase in speed of 12.8%, while the KSPC decreased by 21%. The learning effect was significant for both metrics ($p = 0.044$ for WPM and $p = 0.039$ for KSPC). There is a significant correlation between usual input rates with the

newly proposed input rate (Pearson's $r = 0.536$, $p = 0.024$), so faster typers and swipers are also faster using Faceboard.

Participants' demographic characteristics show that technical background is not necessary to configure the accessibility service on Android, even though the process is more complicated than the usual installation of an application by restriction of the Android system regarding security. It can be seen that the implementation is suitable independently of demographic characteristics. Notably, participants' age had no influence on ease of installation. Participants had no problem learning and setting up the newly proposed technique on their own using the provided tutorial (**NFR1** and **NFR2**).

Users tended to give critical feedback about the software based on SUS, NASA-TLX, and additional feedback gathered after and during the experiment. The current implementation was rated particularly high in workload by the respondents. Using NASA-TLX, Faceboard received a higher-than-average weighted workload (58 vs. 48) with a significant standard deviation. Participants rated the application as mentally and physically demanding, especially on the eyes, neck and head; some of them also found it frustrating to use. Interestingly, participants with chronic pain rated the workload lower than healthy participants. Still, most of the users showed learning effects and less physical and mental demand on the second try of the proposed method.

For daily usage, Faceboard needs improvements regarding configurability, consistency, reliability and speed, as suggested in the chapter 'Future Work'. Even though it ran on all Android Smartphones tried during user tests, the speed and stability varied highly between the different phones and manufacturers and could render the application unusable depending on the framerate of the face recognition framework.

Compared to related studies around input methods for people with disabilities, the approach seems to be less usable than other proposed concepts regarding performance and usability.

Even though one participant had problems with the face-recognition, the machine-learning part seems to be sufficient to recognize head movements and blinking gestures. There is potential to enhance the proof of concept as proposed in the section Future Work.

10.1 Future Work

The previous chapters discussed the usability and performance of Faceboard. The physical and cognitive demands were rated particularly high by participants of the user study. Some of the users became frustrated over time because the blink detection did not work as intended. The performance is usable but was rated below average by participants on the SUS questionnaire, indicating that there is room for improvement. Some of the participants even came up with their own ideas for improvement, which helped to shape possible future work for more usability and performance. In this chapter possible improvements, extensions and changes are presented and discussed.

10.1.1 System-wide Implementation

Since the head-tracking already is implemented as a system-wide service, it is obvious that one could integrate it seamlessly across different parts of the Android operating system. Allowing users to freely navigate through menus and allow clicking either by dwelling on an item or blinking twice would allow impaired users to have one solution to control the whole operating system.

During prototyping, it already was possible to control parts of the operating system. For the sake of simplicity, the head movement service was disabled when the focus was not on a text input field.

10.1.2 Implement for Different Operating Systems

Participants would like to use the service across different platforms ('I would also prefer to have this solution on the laptop [...]'). Incorporating the computing speed of a full-sized computer could allow higher frame rates and much better usability.

10.1.3 Dwell-time Instead of Blinking

Since a lot of the participants of the study found the blinking algorithm cumbersome to use and exhausting ('It was exhausting for the eyes', 'My eyes were full of tears and hurt [...]'), it would make sense to introduce dwell-time state change instead of blinking. Dwell-time means selecting an item by hovering the cursor over the item for a predefined or configurable amount of time. By this, physical strain associated with the frequent blinking and eye pain as well as the need to take off the glasses probably could be reduced.

10.1.4 Smoothing Algorithm

To allow jitter-free cursor movement, a smoothing algorithm would allow higher cursor movement precision, especially for devices with lower computing power and respective lower frame rates. Unintended movements could get filtered out and might allow more precise controlling of the cursor.

10.1.5 Configurable Parameters

Since some of the users complained about the movement of the cursor being too slow and the distance to move the head being too far, it makes sense to introduce configurable parameters and allow users to customize the system to their specific needs and preferences, which again, improves the overall usability of the app. Example parameters would be:

Scaling Factor for Movement Speed

Allow users to customize the speed of the cursor for faster or more precise typing.

Positions of the Buttons

It would make sense, especially combined with the movement speed scaling factor, to allow the free position of the redo and delete word areas.

Closed Eye Threshold

Some users might prefer to keep their eyes closed longer until a gesture starts or ends.

Also, a **feedback loop** which fine-tunes detection settings for each user automatically could help with the users' frustration and success rates.

10.2 Design Optimizations

10.2.1 Add feedback on gesture

Some users stated it would be good to have other feedback combined with the changing cursor like a sound playing on starting and ending a gesture ('I would love to have an acoustic signal when switching from "X" to "O" mode. I am looking at the change, but still I have no idea if it worked.'). This might enhance the users' experience and may be beneficial for users with visual impairments or in some situations where the visual feedback is not sufficient. Also, vibration would be a possible feedback for a gesture. In Majaranta et al. 'Auditory and visual feedback during eye typing', the users benefited from auditory feedback with gains in words per minute [MM3].

10.2.2 Improve Stability By Using Sensor Data

Since most Android phones include sensors like accelerometers and gyroscopes, it would be possible to eliminate the jitter created by holding the phone when not using the phone on a stand. This would make the service more usable on the go.

10.2.3 Improved Edge-Clipping

Some users stated that it would make sense to add a visible barrier to the cursors' position ('It would probably be a good idea if you would not have to move the cursor up all the

way to recalibrate but to have a visible barrier instead'). By that, to calibrate the cursor position, the users would not have to move their head all the way up to the edge of the phone but, for example, there is a visible barrier right above the buttons to redo and delete words.

Rebase on Google's GameFace / MediaPipe Framework

One year after the start of the thesis, Google launched the project Google GameFace [Gooa]. Since it is open source and utilizes the new MediaPipe Framework by Google, it might allow better performance, and results created on top of the new Framework could allow better-performing proof-of-concept apps. Also, the base app already allows to add different actions to be bound to different gestures like smiling, mouth opening or raising eyebrows which would make it a good starter for another proof of concept.

10.2.4 Alternative Keyboard-Layout

It is possible that an alternative layout as proposed by Nowosielski et al. could perform better than QWERTZ-based layouts [Now16].

10.3 Extended User Studies

With 16 participants, the sample size is decent for a pilot study and shows similar results compared to studies with bigger sample sizes. With a larger and more diverse participant group focused on users with different types of motion impairment, deeper insights might be gained and more feedback could be gathered.

Another possibility for an extended user study would be a longitudinal study. An experiment over a longer time span could capture long-term learning effects. Possible performance plateaus could be identified.

Also, a real-world deployment study could show how quickly new users would adopt Faceboard and if they would keep using it.

List of Figures

1.1	One of the participants during the experiment	3
2.1	Transcribing the word 'Hallo' with EVA face mouse[Mau].	13
2.2	Writing the word 'cirrin' on the cirrin keyboard. Figure by MacKenzie et al. [MS02]	14
2.3	EyeSwipe as proposed by Kurauchi et al. [KFJ ⁺ 16]. Figure by Kurauchi et al.	16
2.4	Moving the cursor to the selected letter and stopping there will accept the letter as input [Now17]. Figure by Nowosielski et al.	17
2.5	NASA-TLX scores of the evaluation of Tiltwriter. Figure by Castellucci et al.[CMM ⁺ 19]	18
2.6	The figure shows the two possible ways to enter the text 'quick' using the brain-computer-interface proposed by Bacher et al. [BJM ⁺ 15]. Figure by Bacher et al. [BJM ⁺ 15].	20
3.1	The users get security warnings when they try to enable the service . . .	22
3.2	State machine for the possible states the cursor is in. Blinking triggers the state transitions. The red emojis shows the corresponding cursor presentation	24
3.3	The necessary movement of the head to write the word 'Hallo'. The red circle marks the cursors' current position and visualizes the current position of the users' virtual finger.	25
3.4	The clipping-space and the zones where redrawing and word deletion gets triggered.	27
3.5	State machine for the possible states the cursor is in. Blinking triggers the state transitions. The red emojis shows the corresponding cursor presentation	28
3.7	The tutorial guides users through the setup and finally activates Faceboard	30
3.8	Different states in game	32
3.9	Screenshot of the first working prototype	35
3.10	The gesture-path for the word 'Steuerzahler'.	35
3.11	Component Interaction Diagram between the main services of the Proof of concept app.	36
5.1	Example of the evaluation setup with one of the participants during one of the pilot studies.	49
		93

5.2	System Usability Scale (SUS)	53
5.3	NASA-TLX Step 1	54
5.4	NASA-TLX Step 2 - Rating on subjective workload across the six dimensions	55
5.5	Necessary Swiftkey Configuration	56
7.1	Comparison of word frequencies for Faceboard input method - short words seem to be more error-prone	68
7.2	Box plots of words per minute for each of the two sessions	70
7.3	Box plots of keystrokes per character for each of the two sessions	71
7.4	Comparison of word frequencies for usual input method	72

List of Tables

3.1	The different devices the first prototype was tested on	33
3.2	The different devices the second prototype was tested on	34
4.1	Overview of Variables and Metrics gathered, adapted from Schlögl et al., 2019 [Sch20]	40
4.2	Example entries from the <code>installation_user_correlation</code> table .	40
4.3	Character classes used in computing corrected, uncorrected, and total error rates. From Mackenzie et al. [Mac07]	42
4.4	data structure for words	42
4.5	Example entries from the <code>words</code> table	46
5.1	Demographic Characteristics of Participants	60
7.1	correlation between word length and frequency deleted	68
7.2	Different metrics averaged over all runs	69
7.3	Participants get better result on second round compared to first round . .	69
7.4	Participants tended to get better result on second rounds	70
7.5	Averages over all participants - usual input method	72
7.6	correlation between wordlength and frequency deleted for usual input method	73
7.7	Averages over all participants - usual input method split by swiping and typing	73
7.8	Correlation between WPM averages between Faceboard and usual input methods	73
7.9	Averages over all participants - Faceboard split by swiping and typing . .	74
8.1	Participants' SUS score	75
8.2	Participants SUS question per question analysis	76
8.3	Participants TLX dimension weighting	78
8.4	Participants TLX dimension statistics	82
8.5	TLX dimension correlations	82

List of Algorithms

4.1	Derive Words from Keystrokes	46
-----	--	----

Participation Notes

Data

Participant - Ethan

- 34 years old
- self employed
- usual input: swipe
- restrictions: no

Achieved Goals

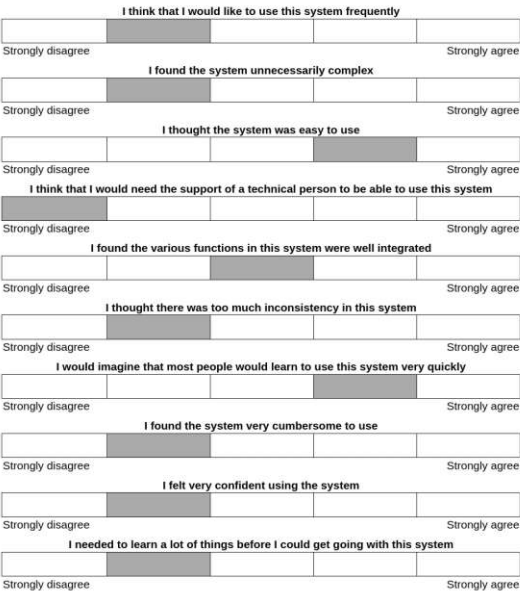
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

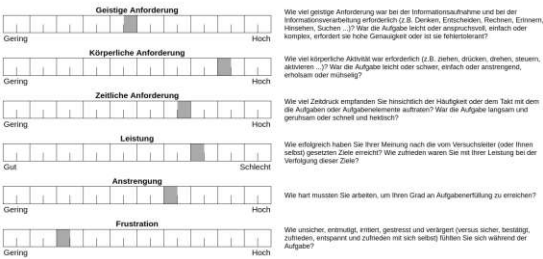
Additional Feedback

First try "I like the concept but it is not working that good"

Second Try "It is not bad, but also not very good. I probably would not use it as long as I am healthy"



(a) SUS Score 65



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	50	4	0.26666666666666666
Körperliche Anforderung	85	2	0.13333333333333333
Zeitliche Anforderung	70	2	0.13333333333333333
Leistung	75	0	0
Anstrengung	65	2	0.13333333333333333
Frustration	25	5	0.33333333333333333

Gesamtbeanspruchung = 51

(c) TLX Part 2

Participant - Emma

- 23 years old
- Student
- usual input: typing
- restrictions: no

Achieved Goals

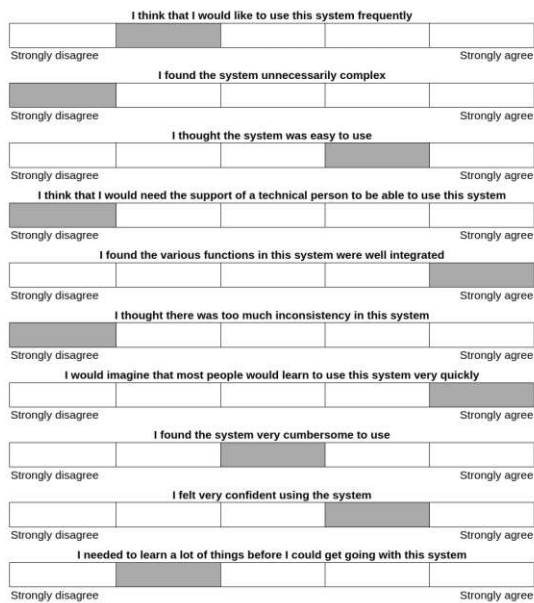
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "In my opinion, the music in the HyperTyper App made the temporal demand on the study higher, once we turned it off, it was a lot easier on the mind to not rush into writing the words." "During gestures, I sometimes was afraid to blink, even though it would work since I would have to blink for a long time to stop the gesture. But still, I was scared to blink and have to do the whole gesture again" "It gave me a bit of neck pain over the time since I had to really concentrate on not accidentally moving the cursor around and have to reposition it"

Second Try "I was a lot more frustrated on the first try, the second try was a lot better and also I felt like I performed a lot better. I think it mostly was because I knew what I was doing on the second try"



(a) SUS Score 80



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	35	2	0.13333333333333333
Körperliche Anforderung	80	5	0.3333333333333333
Zeitliche Anforderung	80	3	0.2
Leistung	85	2	0.13333333333333333
Anstrengung	45	0	0
Frustration	65	3	0.2

Gesamtbeanspruchung = 71.66666666666666

(c) TLX Part 2

Participant - Noah

- 33 years old
- Software developer
- usual input: typing
- restrictions: glasses (tried during tutorial, removed during game rounds)

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "I kept forgetting the word I was writing during the process of a gesture which was exhausting. I also had problems with spelling of some words, so I missed some letters which resulted in wrong words which was frustrating a bit. I would never use it as long as I have the ability to use my hands properly, maybe after improvement but right now, it is not an option at all"

Second try "This time it was a lot easier and also I was a lot less frustrated." "Still would not use it as long as my hands are healthy" "The music did not bother me at all"



(a) SUS Score 47.5



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	80	4	0.26666666666666666
Körperliche Anforderung	40	0	0
Zeitliche Anforderung	80	3	0.2
Leistung	25	2	0.13333333333333333
Anstrengung	75	3	0.2
Frustration	25	3	0.2

Gesamtbeanspruchung = 60.666666666666664

(c) TLX Part 2

Participant - Oliver

- 56 years old
- Herniated discs C4 and C5 in the cervical spine. Repetitive movement or staying in a position for a long time hurts Oliver after a while. The herniated discs make itself felt by a numb right hand when staying in the same position for some time, especially when sat down. Sometimes, unwanted cramps in the fingers of the right hand happen.
- Railway worker
- usual input: typing
- restrictions: glasses (not used during experiment)

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

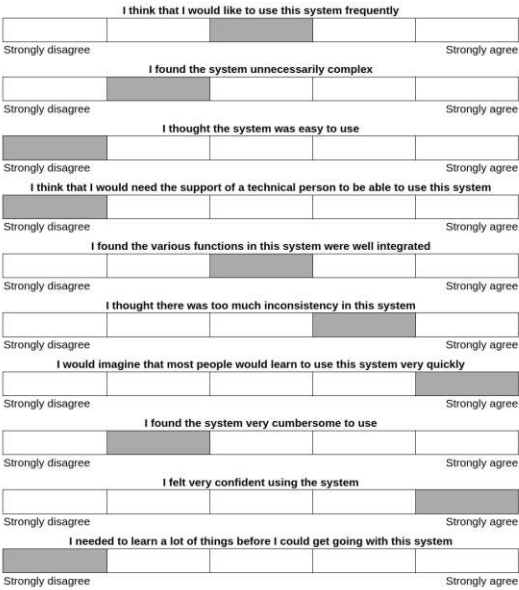
Questionnaires

Additional Feedback

Since Oliver has problems with the cervical spine, he mentioned problems with looking up to calibrate the cursor as well as turning his head that much was a little painful.

First try "The distance I have to move my head is to far, I would prefer it to be lot less than the current version" "I sometimes had problems with spelling words and forgot how to write it, which resulted in a lot of mistakes as I read the sentence once and then did the gesture"

Second try "It got easier when I knew that I didn't have to be exact to write the words." "If I wouldn't have two healthy arms, I would probably give it a try as my main type of input, but right now, I would rather use my fingers since it was a little bit exhausting and also bad on the neck."



(a) SUS Score 67.5



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	5	2	0.13333333333333333
Körperliche Anforderung	30	1	0.06666666666666667
Zeitliche Anforderung	10	4	0.26666666666666666
Leistung	75	2	0.13333333333333333
Anstrengung	5	1	0.06666666666666667
Frustration	5	5	0.3333333333333333

Gesamtbeanspruchung = 17.333333333333332

(c) TLX Part 2

Participant - Sophia

- 56 years old
- Waitress - Currently retired due to herniated discs C2 and C3 in the cervical spine. The herniated discs pinched some nerves which caused permanent damage to the nerves - Spinal stenosis (<https://orthoinfo.aaos.org/en/diseases-conditions/cervical-radiculopathy-pinched-nerve> + <https://www.niams.nih.gov/health-topics/spinal-stenosis-symptoms>). Covid-crisis caused delay for necessary operations to fixate the damaged discs. After operation, the nerves never fully recovered.
- This resulted in numb fingers caused by the damaged nerves in the cervical spine. The numb fingers cause Problems with finger movement as well as problems for usual typing methods on smartphones. The typing is exhausting as well as painful after some time, also accuracy is lower than before the nerve problems occurred.
- Also there are days, where the participant feels dizzy caused by the damaged nerves which is caused by the spinal stenosis. This can result in problems looking on screens
- usual input: typing
- restrictions: glasses (tried during tutorial, removed during game rounds), Spinal stenosis, cervical radiculopathy

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Participation Transcript Day 1

M: Hello Sophia, thanks for taking part in the study. I hope you are doing well.

S: Hello, thanks. I am doing good. I am currently a little dizzy and my fingers are numb, but I think that's a good thing for participating in the study (laughs).

M: I hope you get well soon. I will now tell you what we're going to do and what exactly you have to do.

Annot. - explanation/installation as given in the moderation script -

M: So I think we are ready to go. Are you ready?

S: Yes, I am.

Annot. - Participant had no problems in setting up the Faceboard App as well as initiating the Head movement accessibility service. Sophia just mentioned during the setup that she finds that 'it is unnecessary complicated to set up the app (Annot: The Accessibility Service)' -

M: Great, now we play a little tutorial to find the optimal position of your head. At the top, you see the word you have to write down, on the bottom the keyboard. Also, you can see the cursor - which is a little 'X' - move around when you move your head. This is the position in which your head currently aims. If you move your head, you will also see the cursor move with your movement.

Annot. - Sophia moves her head and sees the cursor move on screen -

S: Okay, I understand. So when I move my head, it also moves the little 'X' on the screen and this is the position I will write at?

M: Exactly, so when you want to write 'Hallo' for example, you start at 'H'. So let's try it and position the cursor over the 'H'.

Annot. - Sophia positions the cursor with her head over the letter 'H' -

M: So far so good, now we have our starting point. Now, when we want to initiate a gesture or, easier said, a word we want to write, we need to blink. To allow normal blinking, we have to keep our eyes closed a little bit longer than usual, so that you can blink normally during the whole participation. When you now blink, you will see the cursor change from an 'X' to a circle. This means we have started to record the word we want to write.

Annot. - Sophia blinks and it changes to a circle -

S: Okay, I get it.

M: Great, now if we want to write the word 'Hallo' we go letter by letter with our little cursor. We start at 'H', go to 'A', then to 'L' and finally to 'O'. When we are at our last letter, we can stop the recording by blinking again. Can you try that now?

S: Okay.

Annot. - Sophia tries to write the word 'Hallo' and ends up with the word 'Bali'. -

S: Why does it write Bali now?

M: This can happen if the cursor is too far down on the screen and near the letter 'B' instead of in the middle of the letter 'H'. But don't worry, this might happen a lot. For this, we have the 'Delete last word' surface, which allows you to delete the last transcribed word. To do that, you just have to move your cursor above the surface and it will delete the last written word. You can try that.

Annot. - Sophia moves the cursor above the 'delete last word' button and the word gets removed -

S: It is a bit hard to go up to the 'delete last word' surface since I have to look up far and my mobility is limited, but it is okay. I understand, should I try again now?

M: Yes, please.

Annot. - Sophia tries again and now gets the word 'Hallo' -

S: Okay, I get it. Should I now transcribe the next word?

M: Exactly.

Annot. - Sophia keeps going and transcribes the exercise words 'Danke', 'die', 'Sonne', and 'lacht' without major problems. -

M: Great, now we have our first sentence 'die Sonne lacht'. For this, you do exactly the same, you stop the gesture after every word.

Annot. - Sophia tries to write the sentence in one go and one gesture instead of 3 gestures, which results in the word 'Diagonal'. -

M: Okay, maybe I explained it badly. What I meant was that you have to blink, write the word 'die', blink again. Then you blink, write the word 'Sonne', blink again, and so on.

S: Okay, I get it now, sorry.

M: No problem, it was my bad explanation. You are not the first one trying this.

Annot. - Sophia writes the 3 words with only one error, writing 'langt' instead of 'lacht', which she fixed on the second try. -

M: Great, that basically was the tutorial for our study. What do you think so far?

S: It is okay, I think one can get used to this. I am sure it will be exhausting after a while, but so far it is okay.

M: Okay, now we basically do the same thing, just packed into the game HyperTyper I told you recently about. At the top of the game, you will see the sentence you have to transcribe, at the bottom there will be your keyboard and your cursor just as before. Are you ready?

S: Yes, I am.

Annot. - We started HyperTyper -

S: Can we turn off the music?

M: Of course.

Annot. - We started the first round of Hyper Typer -

M: As you see, the round automatically stops and starts the next round, as long as the sentence is correct. If you get exhausted, we can take a break whenever you need to. There will be automatic breaks after every 4 sentences.

Annot. - First 4 rounds done -

M: How do you feel? Do you need a break? We have 4 more rounds to go today.

S: It's not bad, it doesn't work that well but maybe I'll get into it. We can keep going.

Annot. - Second 4 rounds done -

Annot. - Sophia did a break here and when we got back, she had to re-calibrate the cursor to sync it with her intended face movement -

S: It is hard to look far up to get the cursor back to the center of the keyboard. I cannot look up that far since my spine is blocking due to the fixation of C2 and C3 with screws.

Annot. - Sophia was referring to re-calibrating after looking away, which can end in a badly positioned cursor and problems on how to reposition it to the center of the keyboard. For this, you have to hit the edges of the display with the cursor and move it back down so the cursor and the head once again are in sync. -

Annot. - Third 4 rounds done -

Annot. - Last 4 rounds done -

M: What do you think of today's tries?

S: I think it is hard to reset the cursor to the middle, but once you have it back in the middle, it is working not bad. The blinking detection is not that good and it sometimes said I was blinking even if I was not. On bad days with big pain in my fingers, I would probably give it a try. But usually, when my fingers are in pain, also my head is dizzy so it would probably not be good to move my head around that much. I would also prefer to have this solution on the laptop, not on the phone since on the phone it is easy to switch from right to left hand.

Participation Transcript Day 2

M: Good morning Sophia, how are you feeling today?

S: I am tired but at least I am not dizzy today (laughs).

M: Okay, that's great to hear. Today we start off with an easy task. First, you will play HyperTyper with your own mobile phone and your usual input method. Do you usually type key by key or use swipe?

S: I do not use swipe, I type letter by letter.

M: Okay, good, let's set up the HyperTyper App on your phone.

Annot. - Setting up HyperTyper on Sophia's phone -

S: Okay and now I do the same as yesterday, just with my fingers?

M: Exactly. Are you ready?

S: Yes, let's start.

Annot. - Sophia finishes the first of 4 rounds -

S: Just keep going?

M: Yes, 4 rounds, just like yesterday with the head movement.

Annot. - Sophia finishes the other 3 rounds -

M: Okay, thanks. Now we do the same as yesterday. You will have to play 4 rounds of HyperTyper with the faceboard keyboard. Are you ready?

S: Yes, can we try to play with my glasses on?

M: Of course, let's try it and see if it works.

Annot. - Sophia plays the first of 4 rounds, having problems with eye recognition using her glasses. She takes them off after the second phrase -

S: I felt more confident yesterday.

M: Don't worry, maybe you will get into it. Most of the participants struggled with the start.

Annot. - Second round done -

S: You are right, it's feeling better.

M: Should we keep going or do you need a break?

S: No, let's keep going.

Annot. - Third round done -

M: Okay, final round.

S: Great.

Annot. - Fourth round finished -

M: And we're done. How did you feel this time?

S: I still have the feeling it was better yesterday. But still, it was okay.

M: Thanks, we now have some questionnaires to answer.

S: Okay, I am ready.

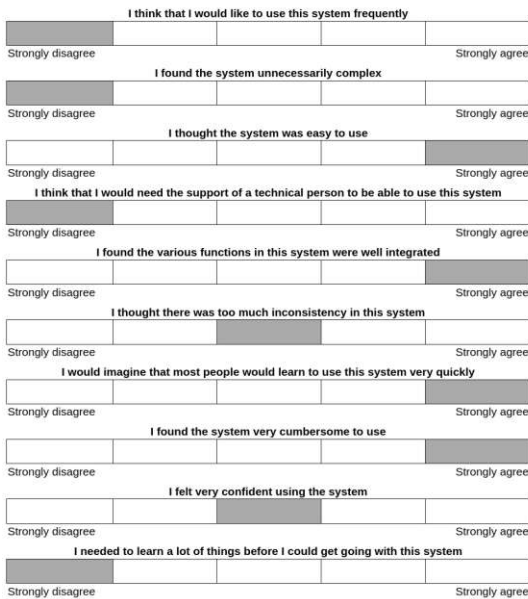
Annot. - Sophia fills out questionnaires -

M: Thanks, do you have any additional feedback on the app and the usage of the app?

S: Well, it works okay-ish, but I still prefer typing with my fingers, even when they are numb like yesterday. It does hurt a bit, but as long as I don't have to write an essay on the phone, it is okay. Also, it would be good if it worked better with glasses, since it is hard to read the sentences without glasses. But still, it works well without my glasses, so it is fine. I also noticed that I had the most problems with really short words; the longer the word, the better it was on the first try. On really bad days when my fingers really hurt, I can imagine using this app. But as I said yesterday, maybe the head movement is a bit too much and I might feel dizzy by using it too much.

M: Did you get dizzy using it now?
S: Well, I was already dizzy using it yesterday, but it didn't get worse from it. Today it was fine.
M: Thank you.

Questionnaires



(a) SUS Score 70



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	10	5	0.3333333333333333
Körperliche Anforderung	25	2	0.1333333333333333
Zeitliche Anforderung	20	3	0.2
Leistung	20	4	0.2666666666666666
Anstrengung	25	1	0.0666666666666667
Frustration	20	0	0

Gesamtbeanspruchung = 17.666666666666668

(c) TLX Part 2

Participant - Benjamin

- 30 years old
- Project manager
- usual input: typing
- restrictions: no

Achieved Goals

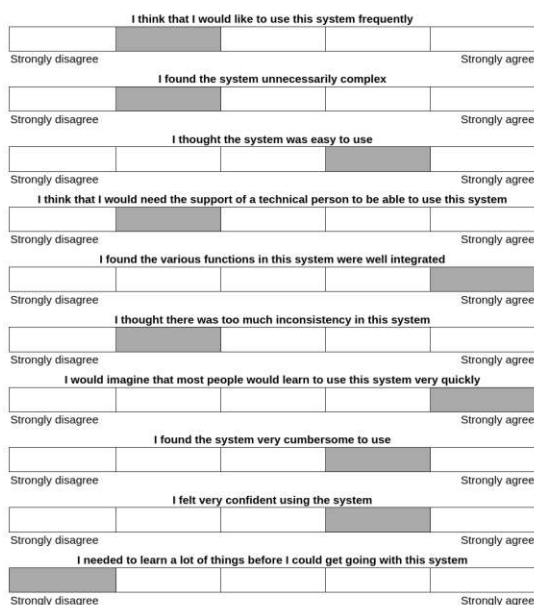
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try Benjamin had problems with short words, especially 'Er' and 'Wie' as well as 'Wir'.

Second try "It would be good, if the performance were better." "I kept forgetting where the letters are on the keyboard, which makes it rather hard to write words"



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	30	0	0
Körperliche Anforderung	75	1	0.06666666666666667
Zeitliche Anforderung	90	5	0.3333333333333333
Leistung	85	3	0.2
Anstrengung	90	4	0.26666666666666666
Frustration	65	2	0.13333333333333333

Gesamtbeanspruchung = 84.66666666666667

(c) TLX Part 2

Participant - Liam

- 30 years old
- Software developer
- usual input: typing
- restrictions: no

Achieved Goals

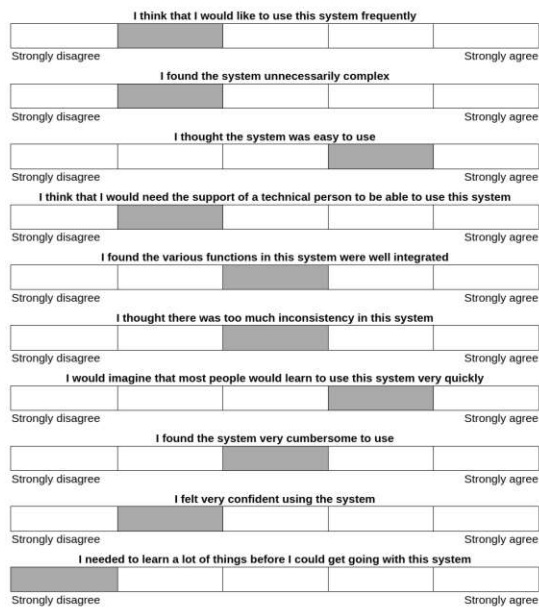
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "It was very exhausting and it works really bad. I would not use it at all"

Second try "A lot better this time, but still I would not use it. There are better options like voice input"



(a) SUS Score 60



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	55	3	0.2
Körperliche Anforderung	75	5	0.3333333333333333
Zeitliche Anforderung	40	1	0.06666666666666667
Leistung	55	0	0
Anstrengung	80	4	0.26666666666666666
Frustration	65	2	0.13333333333333333

Gesamtbeanspruchung = 68.66666666666667

(c) TLX Part 2

Participant - William

- 32 years old
- Software Developer
- usual input: typing
- restrictions: no

Achieved Goals

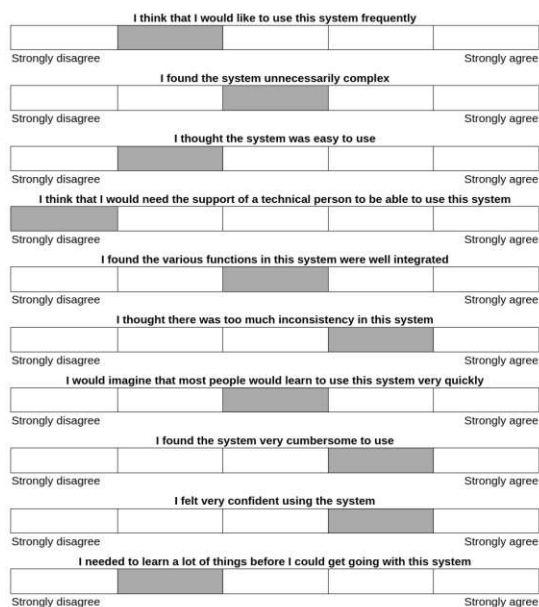
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "Shorter words are a lot worse to write than longer words"

Second Try "I like the competitive approach of HyperTyper which motivated me to get better with every try. It actually was fun to do!"



(a) SUS Score 50



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	75	1	0.06666666666666667
Körperliche Anforderung	65	4	0.26666666666666666
Zeitliche Anforderung	65	3	0.2
Leistung	5	0	0
Anstrengung	70	2	0.13333333333333333
Frustration	65	5	0.3333333333333333

Gesamtbeanspruchung = 66.33333333333333

(c) TLX Part 2

Participant - Daniel

- 32 years old
- Self-employed
- usual input: typing
- restrictions: no

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "I closed my eyes too hard during the beginning because I thought otherwise the camera would not recognize if I had closed my eyes. This resulted in a lot of tears and eye pain. Afterwards, it has been better."

Second try "A lot better this time." "I would love to have an acoustic signal when switching from "X" to "O" mode. I am looking at the change, but still I have no idea if it worked."



(a) SUS Score 30



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	55	1	0.06666666666666667
Körperliche Anforderung	85	2	0.13333333333333333
Zeitliche Anforderung	95	0	0
Leistung	70	3	0.2
Anstrengung	80	4	0.26666666666666666
Frustration	70	5	0.3333333333333333

Gesamtbeanspruchung = 73.66666666666666

(c) TLX Part 2

Participant - Olivia

- 33 years old
- Kindergartner
- usual input: swiping
- restrictions: glasses, used contact lenses

Achieved Goals

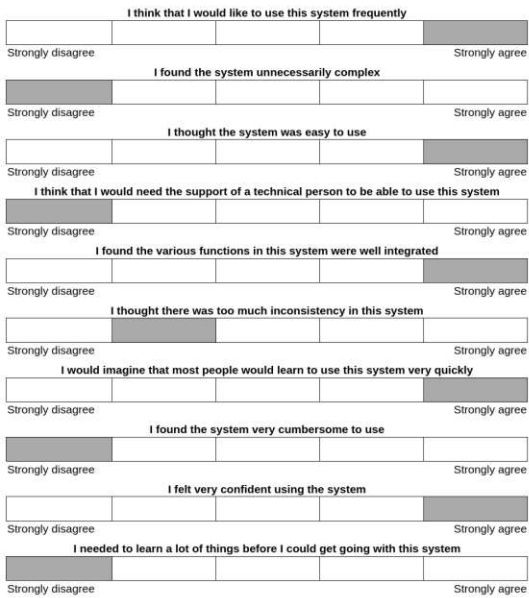
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "The cursor moves down when I close my eyes, that was annoying. Otherwise it was fun"

Second try "I think it is weird that the blanks are added automatically. Also, how would I be able to insert special characters or even capitalize a word?" "Besides that it was cool, the game was fun and also the music motivated me"



(a) SUS Score 97.5



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	35	4	0.26666666666666666
Körperliche Anforderung	20	0	0
Zeitliche Anforderung	25	3	0.2
Leistung	20	5	0.3333333333333333
Anstrengung	15	2	0.13333333333333333
Frustration	20	1	0.06666666666666667

Gesamtbeanspruchung = 24.333333333333332

(c) TLX Part 2

Participant - Grace

- 31 years old
- Public Service
- usual input: typing
- restrictions: no

Achieved Goals

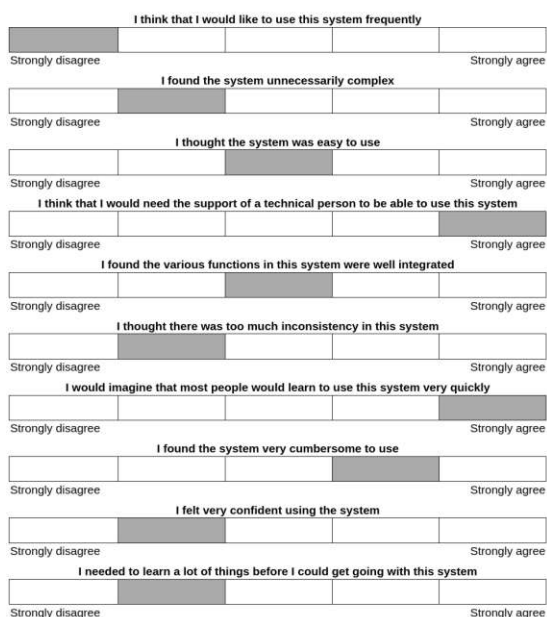
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

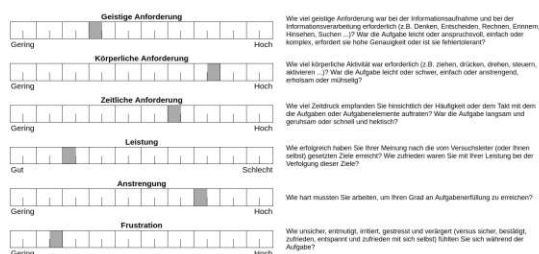
Additional Feedback

First try "I find the calibration rather cumbersome to use. A simple button or gestures would be much easier"

Second try "After a long day at work this is really exhausting to use. I was tired and already looking at screens the whole day." "It would probably be a good idea, if you wouldn't have to move the cursor up all the way to recalibrate but to have a visible barrier instead." "With some small tricks like 'blink, before reaching the last letter of the word' it got a lot better. I think, with a lot of practice, it could work out pretty well"



(a) SUS Score 47.5



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	35	3	0.2
Körperliche Anforderung	80	5	0.3333333333333333
Zeitliche Anforderung	65	2	0.1333333333333333
Leistung	25	0	0
Anstrengung	75	4	0.2666666666666666
Frustration	20	1	0.0666666666666666

Gesamtbeanspruchung = 63.666666666666664

(c) TLX Part 2

Participant - Henry

- 45 years old
- Software developer
- usual input: swiping
- restrictions: glasses (tried during tutorial, removed during game rounds)

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "Calibration is important for this input method, so I would rethink how it works" "It is good that you do not have to be as precise as possible since the swipe algorithm kinda fixes your little mistakes. "Sometimes, I increased the push to close my eyes for no reason, even if it was working with less force."

Second try "The manual input was a lot easier, since you can freely move your head and can take a look back at the phrase which I was afraid to do with faceboard." "I had to search for the letters and sometimes couldn't find them, even though I type on a keyboard everyday"



(a) SUS Score 80



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	80	5	0.3333333333333333
Körperliche Anforderung	15	1	0.06666666666666667
Zeitliche Anforderung	15	2	0.1333333333333333
Leistung	50	4	0.26666666666666666
Anstrengung	95	3	0.2
Frustration	15	0	0

Gesamtbeanspruchung = 62

(c) TLX Part 2

Participant - Samuel

- 31 years old
- Software developer
- usual input: swiping
- restrictions: no

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "In the middle of the third round, I noticed I could just look up with my eyes without moving my head. I kinda forgot about it since I was so focused on the cursor"

Second try "Already a lot easier on second try" "I feel like it is working better when my face is illuminated more even" "It actually is fun when it's working and I get why something like this should be a thesis to write about"

I think that I would like to use this system frequently

Strongly disagree Strongly agree

I found the system unnecessarily complex

Strongly disagree Strongly agree

I thought the system was easy to use

Strongly disagree Strongly agree

I think that I would need the support of a technical person to be able to use this system

Strongly disagree Strongly agree

I found the various functions in this system were well integrated

Strongly disagree Strongly agree

I thought there was too much inconsistency in this system

Strongly disagree Strongly agree

I would imagine that most people would learn to use this system very quickly

Strongly disagree Strongly agree

I found the system very cumbersome to use

Strongly disagree Strongly agree

I felt very confident using the system

Strongly disagree Strongly agree

I needed to learn a lot of things before I could get going with this system

Strongly disagree Strongly agree

(a) SUS Score 42.5

Geistige Anforderung Wie viel geistige Anforderung war bei der Informationsaufnahme und bei der Informationsverarbeitung erforderlich (z.B. Denken, Entscheiden, Rechnen, Erinnern, Vorstellen, Suchen...)? War die Aufgabe leicht oder anspruchsvoll, einfach oder komplex, erfordert sie hohe Genauigkeit oder ist sie fehler tolerant?

Gering Hoch

Körperliche Anforderung Wie viel körperliche Aktivität war erforderlich (z.B. Ziehen, Drücken, Drehen, Steuern, Aktivieren...)? War die Aufgabe leicht oder schwer, einfach oder anstrengend, erholsam oder mühselig?

Gering Hoch

Zeitliche Anforderung Wie viel Zeitdruck empfanden Sie hinsichtlich der Häufigkeit oder dem Takt mit dem die Aufgaben oder Aufgabenreihenfolge auftreten? Wie die Aufgabe langsam und geräusam oder schnell und hektisch?

Gering Hoch

Leistung Wie erfolgreich haben Sie Ihrer Meinung nach die vom Versuchsleiter (oder Ihnen selbst) gestellten Ziele erreicht? Wie zufrieden waren Sie mit Ihrer Leistung bei der Verfolgung dieser Ziele?

Gut Schlecht

Anstrengung Wie hart mussten Sie arbeiten, um Ihren Grad an Aufgabenerfüllung zu erreichen?

Gering Hoch

Frustration Wie unruhig, entnervt, irritiert, gereizt und verärgert (versus sicher, beständig, zufrieden, entspannt und zufrieden mit sich selbst) fühlen Sie sich während der Aufgabe?

Gering Hoch

(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	100	4	0.26666666666666666
Körperliche Anforderung	20	2	0.13333333333333333
Zeitliche Anforderung	40	2	0.13333333333333333
Leistung	20	1	0.06666666666666667
Anstrengung	40	1	0.06666666666666667
Frustration	80	5	0.3333333333333333

Gesamtbeanspruchung = 65.33333333333333

(c) TLX Part 2

Participant - Mia

- 22 years old
- Clerk at court
- usual input: swiping
- restrictions: no

Achieved Goals

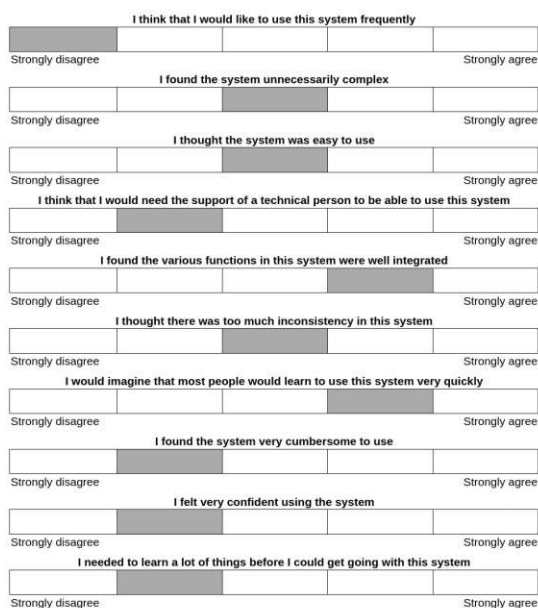
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "It was exhausting for the eyes"

Second try -



(a) SUS Score 55



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	50	2	0.13333333333333333
Körperliche Anforderung	55	3	0.2
Zeitliche Anforderung	50	1	0.06666666666666667
Leistung	15	4	0.26666666666666666
Anstrengung	80	5	0.3333333333333333
Frustration	15	0	0

Gesamtbeanspruchung = 51.666666666666664

(c) TLX Part 2

Participant - Michael

- 30 years old
- Project acquisition
- usual input: typing
- restrictions: no

Achieved Goals

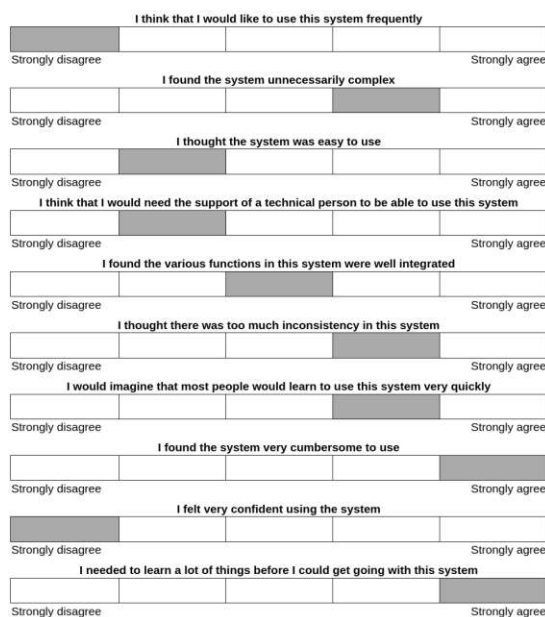
- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

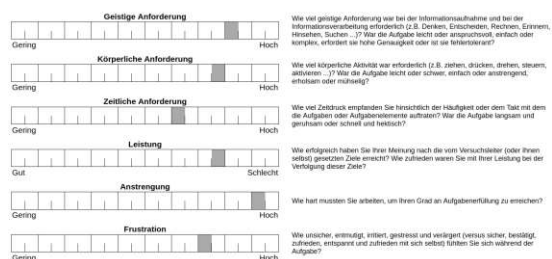
Additional Feedback

First try "It hurts my eyes to play for this long"

Second try "I sometimes forgot which letters I currently was swiping and which would come next since the gestures take so long. With my fingers, this would not be a problem"
"For the use case, the app itself is not overly complicated. It actually is good"



(a) SUS Score 27.5



(b) TLX Part 1

Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	85	4	0.2666666666666666
Körperliche Anforderung	80	2	0.1333333333333333
Zeitliche Anforderung	65	1	0.0666666666666667
Leistung	80	4	0.2666666666666666
Anstrengung	95	4	0.2666666666666666
Frustration	75	0	0

Gesamtbeanspruchung = 84.33333333333333

(c) TLX Part 2

Participant - Charlotte

- 30 years old
- Student
- usual input: typing
- restrictions: glasses (tried during tutorial, removed during game rounds), rheumatoid arthritis

Achieved Goals

- ✓ Setup Faceboard.
 - ✓ Independently setup Accessibility Service.
 - ✓ Allow camera permissions.
- ✓ Play tutorial mode
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Play 5 rounds of HyperTyper with usual input method.
- ✓ Play 5 rounds of HyperTyper with Faceboard.
- ✓ Fill out SUS questionnaire
- ✓ Fill out TLX questionnaire

Questionnaires

Additional Feedback

First try "Only the eye with astigmatism seems to work, so I only 'look' with that one. If I look through my eye without astigmatism, it just doesn't work." "Setting up the app and finding the right position took a really long time" "I sometimes had problems with how to write words correctly"

Second try "It was really frustrating this time, since it did not work at all." "My eyes were full of tears and hurt and when I wanted to relax my eyes, I accidentally deleted the correct words. Very frustrating"



Dimension	Bewertung	Klicks	Wichtung
Geistige Anforderung	10	0	0
Körperliche Anforderung	70	4	0.26666666666666666
Zeitliche Anforderung	90	2	0.13333333333333333
Leistung	15	4	0.26666666666666666
Anstrengung	100	4	0.26666666666666666
Frustration	100	1	0.06666666666666667

Gesamtbeanspruchung = 68.00000000000001

(c) TLX Part 2

Moderator script

1. **Welcoming the participant:** 'Hello _____ and thanks for helping me out on evaluating my app for my master thesis. Can I get you a drink? I also brought some snacks since this will probably take a while for both of us. Feel free to grab them when you are hungry.

2. **Explain what we're going to do and why:** Before we start, feel free to ask questions anytime as soon as they pop up. We're going to evaluate the app I've created last year. It basically is a Proof of Concept that allows you to input text via head movements and blinking with your eyes. By the use of this concept app, you and me are trying to evaluate the method regarding how well this will work compared to similar algorithms developed like eye-tracking devices, swiping keyboards as well as your usual input method.

3. **Preparations for the experiment:** For the evaluation, we have to do some preparations. For the first part of the experiment, we will be using my phone since we found it worked the best with it. Afterwards, we will need to connect your phone to my WiFi so it can connect to my local server and install an Android App that allows evaluation of your typing speed as well as the errors you make. Do you know how to use a swipe-based keyboard? It is crucial to know since you are going to do the same movement with your head instead of your fingers once the evaluation starts.
If not: Swiping on your keyboard works the following way: First you think about the word you want to write, e.g. 'Hello'. Instead of pressing on each of the letters (e.g. first you press 'H', then you press 'e', then two times 'l' and so forth and so on), you press and hold once at the letter 'H' and start to swipe your finger across the different letters of the word 'Hello'. Once you reach the 'O', you just stop there and lift your finger up. The keyboard will now calculate which word you mean and will automatically write it into the input field for you. If the word is wrong, you can just delete it and try it again.
Reinstall SwiftKey / Delete App Data of SwiftKey

4. **The experiment and its process** Okay, we're now fully prepared. I will now describe the process of today's evaluation and the involved applications and methods. First of all, you will play the game HyperTyper with your usual input method, meaning the way you input text on your phone every day. In this game, you will get points for writing text and move forward in a spaceship flying through space. In the background, the app will evaluate how many errors you made, how many words you can write in a minute, etc. You will play 5 rounds of HyperTyper with your usual input method. Each round consists of 4 levels where you have to transcribe different sentences as fast as you can with as few errors as possible. When you make a mistake, you have to correct your error. Once you get the sentence right, it will automatically progress the game.

Once finished, we will proceed with the new input method I've created for the purpose of this study. For that, I want you to start the Faceboard service, configure it as described in the setup wizard and then play the tutorial to see if you are understanding the way it is meant to be used. One goal of the study is that you can set up the service on your own, that's why I will not assist you during setup, only if you really can not make it or feel lost during the process.

After setting up the service, there will be a little tutorial on the screen. Feel free to play around and try as long as you wish and tell me once you feel confident to start the second rounds of HyperTyper with the usage of the Faceboard service. Again, we're going to play 5 rounds consisting of 4 Levels.

It might take some time to find the right head positioning as well as the way and timing to open/close your eyes. Since Google's blinking recognition is not 100% perfect and its recognition can highly vary between different persons, it could be possible that it feels quite exhausting and frustrating at first. Don't worry about it. Play around and look at the cursor: When does it recognize that your eyes seem closed (this is highly different for different persons since it is a trained algorithm)? How long do my eyes need to be closed until it starts recording? Also, try to find a good height between you and your camera. Whenever I tried, I found it the best when I was slightly higher than my smartphones' camera.

5. Further tips on how to use faceboard efficiently - also giving them again during tutorial depending on how confident the participant is

- a) By running the cursor to one of the edges of the screen, you can re-calibrate in case you're having problems with your head position and the cursors' position having an offset.
- b) Touch the 'Delete last word' field with the cursor, to delete the latest written word.
- c) When nothing happens after stopping the gesture, make sure to hit the 'Redo gesture' field with the cursor so you don't have to type the word again. I will tell you if it happens.
- d) If you accidentally delete the last word you have written, you can also use the 'Redo gesture' field to make the app draw your last gesture again.
- e) Position the cursor in the center part of the letters' button you want to swipe since the actual path is right on the cursors' center. So positioning the selected character right inside the red circle should be perfect. Although you do not have to be perfectly precise, but just experiment during the tutorial.
- f) When you got used to the swiping and blinking you can try to already stop the gesture on the way to the last letter of the word, which allows you to make fewer mistakes in case of short words by dwelling on the last letter or accidentally hover around letters around your last letter.

Once we finish that, we're going to use the computer to fill out 2 forms: two questionnaires with questions like 'I think that I would like to use this system frequently' and fixed answers from 'strongly disagree' to 'strongly agree'.

As mentioned beforehand, we will do another session with the same procedure a second time.

Last but not least, I will ask you some questions about the overall experience like 'Did you find the process of moving your head exhausting after a while?'. From your answers, I will take notes on a textbook to later transcribe it into my thesis. Are you okay with that?

I think we're ready to start the experiment. As mentioned before, please don't hesitate to ask questions but try to take the experiment as independently as possible.

6. Before filling out the Questionnaires

The practical part is over, thanks for participating. Now you're going to fill out two questionnaires and then have a little chat about your experience.

Most importantly be honest with your answers: I am not going to be mad at you if you give the system a bad rating, I ask you to answer the questions in an honest way so the results are correct. Also take your time reading the questions and think back how you felt during the experiment.

The first form you're going to fill out is the so called System Usability Scale. It's a simple 10 statement questionnaire that you are going to read and respond to. Each statement you can answer on a 5-point scale from 'Strongly disagree' to 'Strongly agree'. Notice that half of the questions have negative phrasing, the other half has positive phrasing. By that we can make sure to have a balanced rating. Using the SUS, we will capture your subjective usability of Faceboard.

The second form you're going to fill out is the so called NASA-TLX questionnaire which consists of two steps: first you're going to rate the workload in six different dimensions like mental demand, physical demand and so on. For each of the dimension you'll give a rating somewhere between 0 to 100 on a clickable scale. A score of 0 for the dimension 'mental demand' would mean you did not have to think, remember and search a lot during the experiment, whereas 100 would mean it was mentally exhausting. The six dimensions are as follows: *Mental Demand* measures how much mental and perceptual activity was needed to do the transcription with Faceboard. This can be tasks like searching for the right letters, remembering the phrase and thinking about the path one has to draw to generate the desired word. *Physical Demand* describes the amount of physical activity you needed to finish a task. In context of Faceboard this could be the physical strains of moving the head, forcefully blinking of the eyes and staying still to not move the cursor unintentionally.

Temporal Demand is the time pressure felt during the experiment for example 'Did you feel like you have to hurry transcribing the phrases to increase your score?'. *Effort* specifies how hard you had to work to accomplish the performance delivered.

Performance Here, the participants rate their subjective success on finishing the required tasks.

Performance rates your subjective success on finishing the required tasks. How good did you think you did the transcription tasks?

Frustration Last but not least, the dimension Frustration. Did you feel frustrated transcribing the tasks? Was it frustrating when you had to delete wrongly written words?

Now, to understand how important each of these dimensions were for you in regard to workload, you have to give them a weighting. You do that by comparing each of them with each other. You basically always have to choose the one dimension which had a greater impact on your overall workload. So if you think Frustration had more impact on you than the physical demand, you would click on Frustration. There is no wrong or right filling out these questions, it's simply your personal experience.

As a reminder, there is also a German description for each of the six dimensions so you can re-read when you fill out the questionnaire. Also you can always ask me during the questionnaire if you forgot how it works or you need the explanation again.

Finally, I have some open ended questions for you. Just tell me what comes to mind. It's also no problem if you have nothing more to add:

- Did you find the process of moving your head exhausting after a while?
- Did something else bother you when using Facebook?
- What did you like about the approach?
- Any additional feedback?

Bibliography

- [and] Create your own accessibility service | Android Developers — developer.android.com. <https://developer.android.com/guide/topics/ui/accessibility/service>. [Accessed 02-12-2024].
- [AWN⁺17] Christopher S. Ahuja, Jefferson R. Wilson, Satoshi Nori, Mark R. N. Kotter, Claudia Druschel, Armin Curt, and Michael G. Fehlings. Traumatic spinal cord injury. *Nature Reviews Disease Primers*, 3:17018, 4 2017.
- [B⁺96] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [BJM⁺15] Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, Sergey D. Stavisky, John D. Simeral, Katherine Newell, Erin M. Oakley, Sydney S. Cash, Gerhard Friehs, and Leigh R. Hochberg. Neural point-and-click communication by a person with incomplete locked-in syndrome. *Neurorehabilitation and Neural Repair*, 29:462–471, 6 2015.
- [BKM09] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
- [BNVB22] Nicholas Ryan Bonaker, Emli-Mari Nel, Keith Vertanen, and Tamara Broderick. A performance evaluation of nomon: A flexible interface for noisy single-switch users. In *CHI Conference on Human Factors in Computing Systems*, pages 1–17. ACM, 4 2022.
- [BOK21] Bastiaan R Bloem, Michael S Okun, and Christine Klein. Parkinson’s disease. *The Lancet*, 397:2284–2303, 6 2021.
- [BPR⁺10] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J.M. Hausdorff, N. Giladi, and G. Troster. Wearable assistant for parkinson’s disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14:436–446, 3 2010.
- [BRL⁺14] René Baranyi, Florian Reisecker, Nadja Lederer, Martin Gobber, and Thomas Grechenig. Wristdroid-a serious game to support and motivate

patients throughout their wrist rehabilitation. In *2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, pages 786–791. IEEE, 2014.

- [CDF⁺20] Muratcan Cicek, Ankit Dave, Wenxin Feng, Michael Xuelin Huang, Julia Katherine Haines, and Jeffry Nichols. Designing and evaluating head-based pointing on smartphones for people with motor impairments. pages 1–12. Association for Computing Machinery, Inc, 10 2020.
- [Cel] Smartphone sales worldwide 2007-2023| Statista — statista.com. <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>. [Accessed 23-05-2025].
- [CGK12] Gary Charness, Uri Gneezy, and Michael A Kuhn. Experimental methods: Between-subject and within-subject design. *Journal of economic behavior & organization*, 81(1):1–8, 2012.
- [CMM⁺19] Steven J. Castellucci, I. Scott MacKenzie, Mudit Misra, Laxmi Pandey, and Ahmed Sabbir Arif. Tiltwriter: Design and evaluation of a no-touch tilt-based text entry method for handheld devices. pages 1–8. Association for Computing Machinery, 11 2019.
- [CSC⁺13] Karen B Chen, Anne B Savage, Amrith O Chourasia, Douglas A Wiegmann, and Mary E Sesto. Touch screen performance by individuals with and without motor control disabilities. *Applied ergonomics*, 44(2):297–302, 2013.
- [Dor07] Zoltan Dornyei. *Research methods in applied linguistics*. Oxford university press, 2007.
- [Eu21] M Hertzum Ergonomics and undefined 2021. Reference values and subscale patterns for the task load index (tlx): a meta-analytic review. *Taylor FrancisM HertzumErgonomics*, 2021 • *Taylor Francis*, 64:869–878, 2021.
- [FFOJB19] Susan Koch Fager, Melanie Fried-Oken, Tom Jakobs, and David R. Beukelman. New and emerging access technologies for adults with complex communication needs and severe motor impairments: State of the science. *Augmentative and Alternative Communication*, 35:13–25, 1 2019.
- [FGRL14] Eva L. Feldman, Wolfgang Grisold, James W. Russell, and Wolfgang N. Löscher. *Atlas of Neuromuscular Diseases*. Springer Vienna, 2014.
- [FR07] Siska Fitrianie and Leon JM Rothkrantz. An adaptive keyboard with personalized language-based features. In *International Conference on Text, Speech and Dialogue*, pages 131–138. Springer, 2007.
- [Gooa] Google. GitHub - google/project-gameface — github.com. <https://github.com/google/project-gameface>. [Accessed 23-05-2025].

- [Goob] Google. ML Kit Google for Developers — developers.google.com. <https://developers.google.com/ml-kit/>. [Accessed 23-05-2025].
- [Gre76] Anthony G. Greenwald. Within-subjects designs: To use or not to use? *Psychological Bulletin*, 83:314–320, 3 1976.
- [Har06] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.
- [HB19] Lois R Harris and Gavin TL Brown. Mixing interview and questionnaire methods: Practical problems in aligning data. *Practical assessment, research, and evaluation*, 15(1):1, 2019.
- [HS88] Sandra G. Hart and Lowell E. Staveland. *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*. 1988.
- [IS12] Curt B Irwin and Mary E Sesto. Performance and touch characteristics of disabled and non-disabled participants during a reciprocal tapping task using touch screen technology. *Applied ergonomics*, 43(6):1038–1043, 2012.
- [Jam] Jamovi. jamovi - open statistical software for the desktop and cloud — jamovi.org. <https://www.jamovi.org/>. [Accessed 23-05-2025].
- [JV01] Julie A. Jacko and Holly S. Vitense. A review and reappraisal of information technologies within a conceptual framework for individuals with disabilities. *Universal Access in the Information Society*, 1:56–76, 6 2001.
- [KFJ⁺16] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. Eyeswipe: Dwell-free text entry using gaze paths. pages 1952–1956. Association for Computing Machinery, 5 2016.
- [LBS⁺05] Thomas Navin Lal, Niels Birbaumer, Bernhard Schölkopf, Michael Schröder, N. Jeremy Hill, Hubert Preissl, Thilo Hinterberger, Jürgen Mellinger, Martin Bogdan, Wolfgang Rosenstiel, and Thomas Hofmann. A brain computer interface with online feedback based on magnetoencephalography. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 465–472. ACM Press, 2005.
- [LLCP10] Amanda Lenhart, Rich Ling, Scott Campbell, and Kristen Purcell. Teens and mobile phones: Text messaging explodes as teens embrace it as the centerpiece of their communication strategies with friends. *Pew internet & American life project*, 2010.
- [LS09] James R Lewis and Jeff Sauro. The factor structure of the system usability scale. In *Human Centered Design: First International Conference, HCD 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009 Proceedings 1*, pages 94–103. Springer, 2009.

- [LST14] Luis A Leiva and Germán Sanchis-Trilles. Representatively memorable: sampling the right phrase set to get the text entry experiment right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1709–1712, 2014.
- [Lub05] Fred D. Lublin. Clinical features and diagnosis of multiple sclerosis. *Neurologic Clinics*, 23:1–15, 2 2005.
- [MA98] Jennifer Mankoff and Gregory D Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 213–214, 1998.
- [Mac07] I Scott MacKenzie. Evaluation of text entry techniques. *Text entry systems: Mobility, accessibility, universality*, pages 75–101, 2007.
- [Mau] Cesar Mauri. GitHub - cmauri/eva_facial_mouse: Camera based mouse emulator for Android — github.com. https://github.com/cmauri/eva_facial_mouse. [Accessed 23-05-2025].
- [MAotSu09] P Majaranta, UK Ahola, O Špakov Proceedings of the SIGCHI, and undefined 2009. Fast gaze typing with an adjustable dwell time. *dl.acm.orgP Majaranta, UK Ahola, O ŠpakovProceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2009 • dl.acm.org*, page 357, 2009.
- [McI13] Hugh B McIntyre. *The primary care of seizure disorders: a practical guide to the evaluation and comprehensive management of seizure disorders*. Butterworth-Heinemann, 2013.
- [MM3] P Majaranta, IS MacKenzie, A Aula CHI’03 Extended Abstracts . . . , and undefined 2003. Auditory and visual feedback during eye typing. *dl.acm.orgP Majaranta, IS MacKenzie, A Aula, KJ RäihäCHI’03 Extended Abstracts on Human Factors in Computing Systems, 2003 • dl.acm.org*, pages 766–767, 2003.
- [MS02] I Scott MacKenzie and R William Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction*, 17(2-3):147–198, 2002.
- [Nie94] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.
- [NJ12] Hugo Nicolau and Joaquim Jorge. Elderly text-entry performance on touchscreens. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pages 127–134. ACM, 10 2012.
- [Now16] Adam Nowosielski. Minimal interaction touchless text input with head movements and stereo vision. volume 9972 LNCS, pages 233–243. Springer Verlag, 2016.

- [Now17] Adam Nowosielski. Swipe-like text entry by head movements and a single row keyboard, 2017.
- [O'S08] THOMAS MICHAEL O'SHEA. Diagnosis, treatment, and prevention of cerebral palsy. *Clinical obstetrics and gynecology*, 51(4):816–828, 2008.
- [ou18] L Longo PloS one and undefined 2018. Experienced mental workload, perception of usability, their interaction and impact on task performance. *journals.plos.org*, 13, 8 2018.
- [PSS17] Ondrej Polacek, Adam J Sporka, and Pavel Slavik. Text input for motor-impaired people. *Universal Access in the Information Society*, 16:51–72, 2017.
- [PZK⁺19] Kseniia Palin, Anna Maria Feit ETH Zurich, Switzerland Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. How do people type on mobile de-vices? observations from a study with 37,000 volunteers. 2019.
- [Ros09] David A Rosenbaum. *Human motor control*. Academic press, 2009.
- [rpr] R core team (2022). r: A language and environment for statistical computing. (version 4.1) [computer software]. retrieved from <https://cran.r-project.org>. (r packages retrieved from cran snapshot 2023-04-07).
- [RT96] Cameron N Riviere and Nitish V Thakor. Effects of age and disability on tracking tasks with a computer mouse: Accuracy and linearity. *Journal of Rehabilitation Research and Development*, 33:6–15, 1996.
- [SC15] Amanda L Smith and Barbara S Chaparro. Smartphone text input method performance, usability, and preference with younger and older adults. *Human factors*, 57(6):1015–1028, 2015.
- [Sch20] Richard Schlögl. *Hyper Typer : Design und Entwicklung eines Serious Game zur Evaluierung von Texteingabemethoden auf mobilen Geräten*. Wien, 2020.
- [SM03] R. William Soukoreff and I. Scott MacKenzie. Metrics for text entry research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 113–120. ACM, 4 2003.
- [Sma] Global smartphone penetration 2016-2024| Statista — [statista.com](https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/). <https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/>. [Accessed 23-05-2025].
- [SotScho11] J Sauro, JR Lewis Proceedings of the SIGCHI conference on human, and undefined 2011. When designing usability questionnaires, does it hurt to be positive? *dl.acm.orgJ Sauro, JR LewisProceedings of the SIGCHI conference on human factors in computing systems, 2011 • dl.acm.org*, 2011.

- [soz] Report of the Federal Government on the situation of people with disabilities — sozialministerium.gv.at. <https://www.sozialministerium.gv.at/en/Topics/Social-Affairs/People-with-Disabilities/Report-of-the-Federal-Government-on-the-situation-of-people-with-disabilities.html>. [Accessed 23-05-2025].
- [SYF07] Andrew Sears, Mark Young, and Jinjuan Feng. Physical disabilities and computing technologies: an analysis of impairments. In *The human-computer interaction handbook*, pages 855–878. CRC Press, 2007.
- [Tur99] Dennis C. Turk. The role of psychological factors in chronic pain. *Acta Anaesthesiologica Scandinavica*, 43:885–888, 1999.
- [VDB20] Kirsten Vitrikas, Heather Dalton, and Dakota Breish. Cerebral palsy: an overview. *American family physician*, 101(4):213–220, 2020.
- [VFN⁺12] Theo Vos, Abraham D Flaxman, Mohsen Naghavi, Rafael Lozano, Catherine Michaud, Majid Ezzati, Kenji Shibuya, Joshua A Salomon, Safa Abdalla, Victor Aboyans, et al. Years lived with disability (ylds) for 1160 sequelae of 289 diseases and injuries 1990–2010: a systematic analysis for the global burden of disease study 2010. *The lancet*, 380(9859):2163–2196, 2012.
- [WBPV10] David M. Wert, Jennifer Brach, Subashan Perera, and Jessie M. VanSwearingen. Gait biomechanics, spatial and temporal characteristics, and the energy cost of walking in older adults with impaired mobility. *Physical Therapy*, 90:977–985, 7 2010.
- [WHO23] World Health Organization WHO. Disability - global report on health equity for persons with disabilities. <https://www.who.int/news-room/fact-sheets/detail/disability-and-health>, 2023. [Accessed 23-05-2025].
- [Wob07] Jacob O Wobbrock. Measures of text entry performance. *Text entry systems: Mobility, accessibility, universality*, pages 47–74, 2007.
- [WP15] Karl Wiegand and Rupal Patel. Impact of motor impairment on full-screen touch interaction. *J Technol Pers wirh Disabil*, 3(22):58–76, 2015.
- [YGY⁺17] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4479–4488, 2017.
- [M14] Oleg Špakov, Poika Isokoski, and Päivi Majaranta. Look and lean: Accurate head-assisted eye pointing. pages 35–42. Association for Computing Machinery, 3 2014.