

Stella Lukasser

# Document Exploration Based on Eye Tracking

## **Bachelor's Thesis**

to achieve the university degree of  
Bachelor of Science

submitted to

**Graz University of Technology**

## Supervisors

Univ.-Prof Dr.-Volksw. Dr.rer.nat. M.Sc. Tobias Schreck

and

Dr.techn. M.Sc. Lin Shao

Institute of Computer Graphics and Knowledge Visualisation

Vasoldsberg, December 2022

This document is set in Palatino, compiled with [pdfL<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>](#) and [Biber](#).

The L<sup>A</sup>T<sub>E</sub>X template from Karl Voit is based on [KOMA script](#) and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

---

## Affidavit

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

---

Date

---

Signature



# Abstract

In this thesis, eye tracking data is used to explore a document about type II diabetes presented on a Consumer Health Information System (CHIS) website. Using a dwell time based approach, where no additional manual input is needed, mouse, touchpad, and keyboard get replaced by an eye tracker. This tracker provides the screen coordinates the user is looking at.

The document data gets presented hierarchically in the form of a table of contents (TOC). The information in form of continuous text can be accessed by focusing on the chapters in the table for a given amount of time. In other words, instead of clicking on a chapter, the user focuses on the chapter of interest with his or her eyes for a given amount of time. The full chapter text is then shown on a side-panel on the right side of the table.

As a proof of concept, that eye tracking data can also be used to obtain information about the user, time data gets logged during user-computer interaction. On one hand, the system logs how long the user gazes at each chapter in the TOC, on the other hand, the words the user reads get logged. This individual information can be used in future projects for an adaptive and personalized presentation of information.

To evaluate the usability of this project, a user study with ten participants was conducted. The user study provided positive feedback but also concrete improvements showing numerous possibilities for future extensions of the project.



# Contents

<b>Abstract</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background and Related Work</b>	<b>5</b>
2.1 Eye Tracking in Interactive Applications . . . . .	5
2.1.1 Eye Movements . . . . .	7
2.1.2 Accuracy and Precision . . . . .	7
2.1.3 Midas Touch Effect . . . . .	8
2.1.4 Eye Tracking as Input for Target Selection . . . . .	8
<b>3 Concept and Approach</b>	<b>11</b>
3.1 Interaction Concepts . . . . .	11
3.1.1 Redo Options . . . . .	14
3.1.2 Gaze-based Scrolling . . . . .	15
3.2 Logging User Data . . . . .	15
<b>4 Implementation</b>	<b>17</b>
4.1 Framework . . . . .	17
4.2 Eye Tracking . . . . .	17
4.2.1 Eye Tracking Technology . . . . .	17
4.2.2 Eye Tracker Usage . . . . .	18
4.3 Program Pipeline . . . . .	20
4.3.1 Data Preparation . . . . .	20
4.3.2 Features . . . . .	21
4.3.2.1 Target Selection . . . . .	21
4.3.2.2 Scrolling . . . . .	22
4.3.2.3 Logging User Data . . . . .	22

## Contents

---

<b>5</b>	<b>Results and Evaluation</b>	<b>23</b>
5.1	Apparatus . . . . .	23
5.2	User Study . . . . .	24
5.2.1	Participants and Tracking Environment . . . . .	25
5.2.2	Procedure . . . . .	25
5.2.3	Feedback . . . . .	26
<b>6</b>	<b>Future Work</b>	<b>29</b>
<b>7</b>	<b>Conclusion</b>	<b>31</b>
	<b>Bibliography</b>	<b>33</b>



# List of Figures

1.1	Program pipeline simplified. . . . .	2
2.1	Adaptive legends. . . . .	6
3.1	Chapter hierarchy. . . . .	12
3.2	Areas of interest. . . . .	13
3.3	Full-text side panel. . . . .	14
4.1	The Eye Tribe tracker. . . . .	18
4.2	Eye-tracking scenario. . . . .	19
4.3	EyeTribe UI. . . . .	19
4.4	Program pipeline. . . . .	20
4.5	Dwell time click on chapters. . . . .	21
4.6	Dwell time click on sub-chapters. . . . .	22
5.1	Post-evaluation tool of EyeTribe UI. . . . .	24
5.2	Logged words with number of timeframes. . . . .	28



# 1 Introduction

Eye tracking has become more and more precise and affordable in the last decades, but in the general public the technology is not well-known yet. In principle, eye trackers provide us with data about where the user is looking at on the screen. This data can be processed and interpreted in different ways. We thus receive information about where the user's interests lie, what content is ignored and what reactions are initiated by certain actions. This information can be used for user studies, on one hand, to study the human's interest and behavior but also, on the other hand, to extend existing user studies in Human and Computer Interaction (HCI) to gain more information about what the user did or explored. In HCI, eye tracking can also be used to interact actively with the computer by using the gaze coordinates as input, like this work does.

Generally, we use our sense of sight to gain information about the outside world. However, we are not used to the fact, that this data gets tracked and for example used for navigation or individualizing the content we see. For one thing, this provides a very powerful new tool for efficient working, reading, use of applications, or receiving knowledge. Then again, the data we gain through this technology is very private as it is very difficult to control where we look. Much personal information can be implied by this data. Hence, eye tracking has played a major role in the field of psychology for several decades. In recent times, however, the quality and affordability of eye trackers have improved to such an extent that eye-tracking research has also been intensified in the area of user interaction with technical devices. Especially in the last few years, research has focused more and more on the use of user data for the adaptive presentation of information. Shao et al. [16] for example use eye tracking to recommend scatter plot data the user hasn't already looked at while exploring a scatter plot matrix. Another example

is the work of Gobel et al. [12], which helps to reduce the complexity of thematic maps by adapting the map legend based on the user's gaze.

This work is part of the A+CHIS project<sup>1</sup>, which is a cooperation of the Graz University of Technology with the Medical University of Graz and the Karl Franzens University of Graz. The goal of this project is to present health information for type II diabetes in an interactive, adaptive, and personalized way, in order to take into account the various needs of health information of individual persons. Our part of the project uses eye-tracking data to navigate one part of the website and thus facilitate the search for information. The A+CHIS website includes a table of contents (TOC) component, which gets expanded in this project to be controllable by gaze.

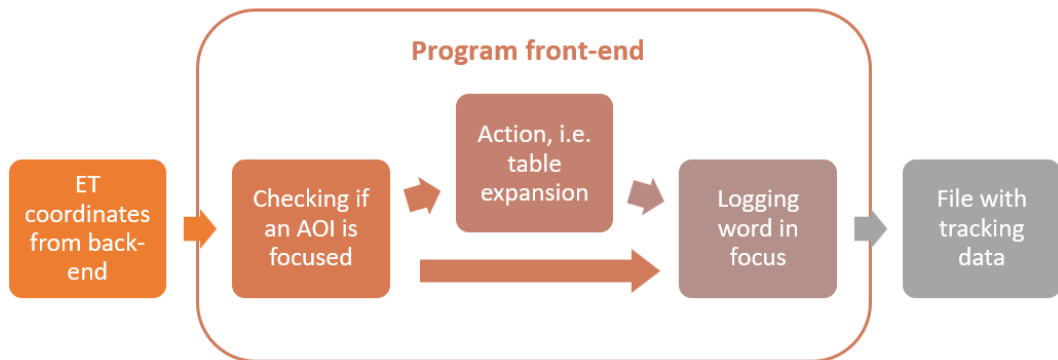


Figure 1.1: The simplified program pipeline showing what actions are performed in the front-end of the implementation.

In Figure 1.1 a simplified program pipeline of the front-end part of the project is shown. The eye-tracking coordinates, which are sent over a Web-Socket to the front-end, and the component coordinates get compared. If the eye-tracking coordinates overlap for two seconds with a specified area, an action like a table expansion happens, see in section [Interaction Concepts](#). Additionally, user data, like the words the user focuses on while exploring the website, gets collected and saved in a file ready to download. This data can be used in the future for an adaptive presentation of information. To make the search for information easier, a scrolling feature controlled by gaze was also implemented.

---

<sup>1</sup><https://apchis.cg.v.tugraz.at/>

---

At the end of this project, the implementation got evaluated by conducting a small user study. Ten participants explored the tool just using their eyes and gave qualitative as well as quantitative feedback. They were first asked to simply explore the website. Afterward, they were given tasks to search for specific words in the text, where the time between the start of the task and the moment the user focused on the searched word got recorded. In general, the participants were widely satisfied with the tool. There were no problems with calibrating the eye tracker. Problems arose because of inaccuracies of the eye tracker, which especially concerned the y-axis, and because of unintentional actions partly due to missing visual feedback.



## 2 Background and Related Work

Due to eye tracking becoming easily available and cheap in the last decades, there has been done more and more research. Papers were published in many different research areas including HCI or information visualization but also in psychological research or medical diagnostic. In HCI there exists much work on using eye-tracking data to test the usability of websites. In the last two decades, research focused mainly on the use of eye-gaze data as an additional input for interaction with technologies. This research area of gaze-based interactive applications is where the main part of this work falls into.

### 2.1 Eye Tracking in Interactive Applications

Majoranta and Bulling [15] divide applications using gaze interaction into four categories, in order to help understanding the differences in the use of eye-tracking data. The first category includes applications using explicit eye input. In these applications, eye-tracking data is used to navigate the computer. Blinking, eye gestures, gaze-controlled cursors, and head movement are some techniques that are used for control and communication. This group of eye-tracking applications is often used for people with motor impairments like the work of Istance et al. [13], who designed gaze gestures for gaze control in online games.

The second group of applications includes those using attentive user interfaces. The user should use his or her eyes naturally without any unnatural movements of the eye. Gaze information gets collected in the background. The work of Shao et al. [16] for example uses eye-tracking data to simplify the evaluation of large amounts of high-dimensional visualized data. To

## 2 Background and Related Work

be exact, while the user explores large scatter plot matrices (SPLOMs), the patterns the user views get logged and potentially new patterns get recommended to the user. This should provide the user with a higher pattern recall. Gobel et al. [12] use gaze data to adapt the legends of thematic maps. In Figure 2.1 the functionality of their work is shown. Depending on the user's fixation the legend highlights the relevant parts while fading out the rest. This approach reduces the time the user has to look at the legend.

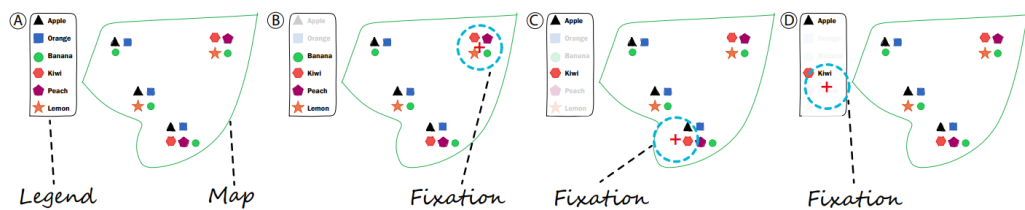


Figure 2.1: [A] shows the basic legend with no gaze-based adaptations. [B], [C] and [D] show the different legends adapted to the user's gaze fixation.

Applications in the third group use eye-tracking data for gaze-based user modeling. This group of applications has a long history because psychological research using eye tracking also falls into this group. Cognitive abilities, the users' degree of expertise, or their level of focus can be derived from eye-tracking data [8, 18]. These works aim to use the gained user information to provide dynamically adapting visualization systems. Bulling et al. [4, 5, 6] use the obtained information to predict user behavior or to recognize human activities. Based on gaze information it is therefore possible to differentiate between tasks the user is currently performing, like reading or office activities.

The last group collects user data in the background to process the data afterward. There is no interactive interface reacting to the user's behavior. Most times these works study long-term user behavior [7]. Here, research shows that it is possible to derive contextual cues about what tasks humans perform on a daily basis from long-term eye movement data.

The majority of this work falls into the first and second categories, as gaze data is used to trigger actions on the website. This should happen in the best possible natural way. However, it is necessary for the user to learn certain



actions for navigation. As data gets stored for later processing there are also parts of this work falling into the fourth category.

### 2.1.1 Eye Movements

To understand the challenges of using eye-tracking data for interactive applications, we need to know how the human eye perceives the environment. The human eye provides high visual acuity for an about 2-degree visual angle. Compared to the length of an arm, this is about the size of a thumbnail. Outside of this small area, called the fovea, the acuity drops rapidly. Therefore, we have to move our eyes to cover the whole visual field and change the position of the fovea [9]. Moving the fovea ballistically and abruptly is called a saccade (about  $1^{\circ}$ - $45^{\circ}$  visual angle). However, most of the time we fixate on a specific object. This is called fixation. Hereby we focus on an object for about 200-600 ms, whereby our gaze is not completely stable. Our eye performs small tremulous movements during these fixations [14]. In this work, only fixations are used.

### 2.1.2 Accuracy and Precision

Accuracy, the offset from the actual gaze point, and precision, the spread of the gaze points, are two values measuring the quality of eye-tracking data [11]. Eye-tracking technologies have improved a lot through progression and research in the field of computer graphics or higher resolution cameras, but the flaws of the collected data are still an issue. Eye trackers are nowadays able to provide an accuracy of about 0.5 to 1 degree. This accords to an accuracy of 0.87 - 1 cm, if the distance between the user and the screen is 50 cm [17]. These values, however, only apply in suitable environments. At the latest, when it comes to the everyday use of eye trackers the values deviate. Different eye-tracking technologies are sensitive to other factors. This work uses pupil and corneal reflection tracking. In this case mascara, reflections through light conditions, or glasses as well as contact lenses are examples of influencing factors. In general, eye physiology, light intensity, the form of light, or calibration are important to consider. Artificial light

is usually preferable [11]. Resulting inaccuracy be it due to environmental influences or the trembling of the eyes during fixations entails a major challenge.

### 2.1.3 Midas Touch Effect

Another major challenge for applications using eye tracking, is the so-called Midas Touch Effect. People are not used to using their gaze as an output, to let actions happen, rather than an input, to just receive information. More precisely, there has to be a difference between just exploring the scene with the eyes and triggering actions through gazing. If the eye gaze itself is used to issue a command without additional inputs or restrictions, the resulting problem is that actions are triggered unintentionally [14].

### 2.1.4 Eye Tracking as Input for Target Selection

This work's main part uses eye-tracking data as an input to select defined targets. This type of application is quite controversial among researchers, as several aspects speak against using gaze input for target selection.

However, Vertegaal [19] names four reasons why using eye-tracking data for target selection might seem to make sense. One situation where this is the case would be, if the hands are occupied. Hereby, the eyes can be used as an additional input. The second reason to prefer eye gaze as an input for target selection is that the eyes are the fastest muscle in our body. This generally implies that this form of input should be faster than other manual inputs. Thirdly, we use our eyes for hours without them getting tired. This suggests, on the basis that eye tracking works by using the eyes naturally, this form of input action does not affect our body. Further, humans use their gaze natively, to indicate what they are referring to. The gaze therefore provides the context.

However, the author also indicates that the problem with the use of eye tracking in a navigation or selection context is, that humans are not used

to using their eyes to provide output. Pointing with our eyes on exact coordinates is not something we are used to, see [Midas Touch Effect](#).

Based on this view, the work of Vertegaal evaluates the selection of visual targets with eye-gaze input and manual input as well as combinations. His work states that eye tracking using dwell time, meaning a click corresponds to focusing on a target for a specific time, is 46% faster than using the mouse for target selection. The combination of mouse clicks and eye tracking is 16% faster. He, however, also experienced a 43% higher error rate for dwell time click, and an 11.7% higher rate for the combination method. This early work therefore already outlines the problem that high error rates make the use of eye tracking very difficult.

It is a major but essential part of this work to address this problem of high error rates. While many papers indicate that the usability of applications that use eye tracking for target selection is highly dependent on the precision and accuracy of the eye trackers, the work of Feit and colleagues [11] in particular provides data on how to choose target sizes and design recommendations for these very applications. The target size gets derived from accuracy and precision values assuming that the gaze points are normally distributed. This means that appropriate target sizes vary with changes in accuracy and precision. As they, among other works, also experienced noticeable differences in tracking quality in different environmental conditions and with different users, this also means the optimal target size varies.



## 3 Concept and Approach

This work's project was built into the A+CHIS framework, specifically the TOC component of the website. Here the user normally - before eye tracking was integrated - used the mouse to open chapters he or she was interested in. The chapters are shown in [Figure 3.1](#). The mouse click expands the table and sub-chapters of the chapter are shown. We expanded the functionality of this component by adding the possibility to open full-text information if a sub-chapter is clicked again, see [Figure 3.3](#). This full text information corresponds to the health information brochure of a German health insurance company called Allgemeine Ortskrankenkasse (AOK).

To explore the document on the website this work uses eye-tracking coordinates in combination with dwell time instead of mouse clicks. The dwell time is the time the user has to focus on one Area of Interest (AOI) with his or her eyes to simulate a mouse click and therefore initiate an action. This concept should provide a way to get information faster and easier, than with the usual interaction involving mouse, touchpad, and keyboard.

### 3.1 Interaction Concepts

To use eye tracking for target selection, we have to consider several issues, of which some were already covered in [Section 2](#). The problem, that the accuracy of eye trackers especially in natural environments varies, is one of the most significant issues. The obvious way to overcome this problem, is to increase the size of the AOIs. This in turn leads to the fact that fewer objects can be placed on the screen. Some works solved this by gaze-based zooming features or fish eye lenses [2, 3]. Our work resolves this by hierarchical organizing, see [Figure 3.2](#). The chapters are collapsed first, after focusing on

### 3 Concept and Approach

---

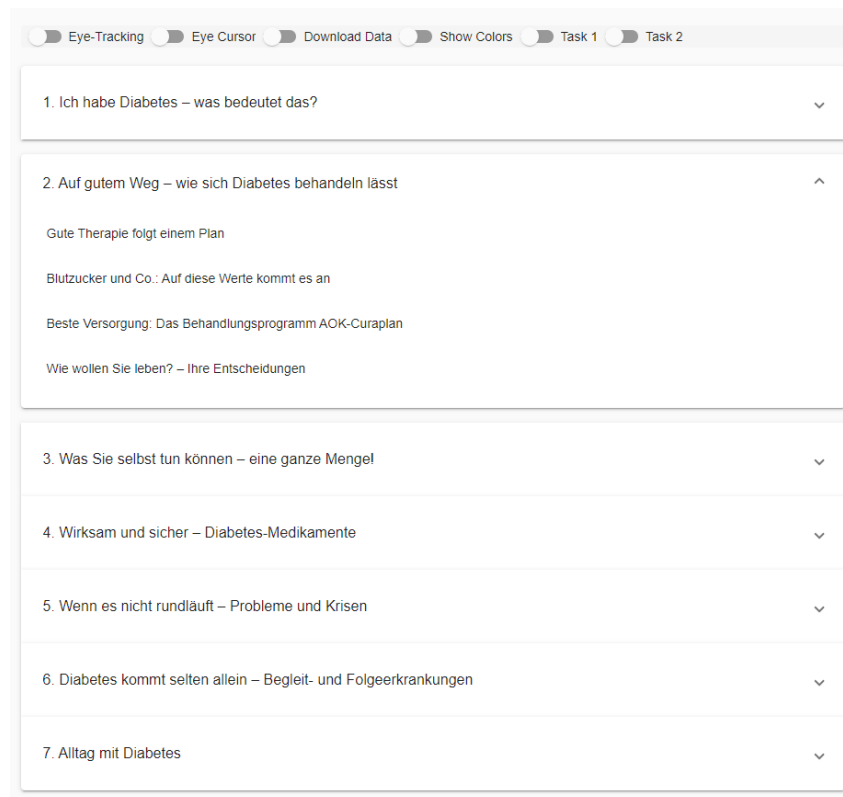


Figure 3.1: Chapter hierarchy. The page shows a TOC with 7 chapters, each consisting of several initially hidden sub-chapters.

a chapter, by looking into the AOI, it expands and more information as well as actions are accessible. Examples for AOIs are marked in red in [Figure 3.2](#).

Placing the important information or object, which catches the user's interest, in the middle of the AOI is another approach to overcome imprecise data. By looking in the middle, inaccurate data is with higher probability still located in the range of the AOI. Smoothing the data, which for this work is mostly done by the provided eye-tracking software itself, also improves usability, since, as stated in [Section 2](#), the eye never stands still. This leads, especially without further smoothing, to data being imprecise.

A very important parameter for the interaction on the website is the dwell

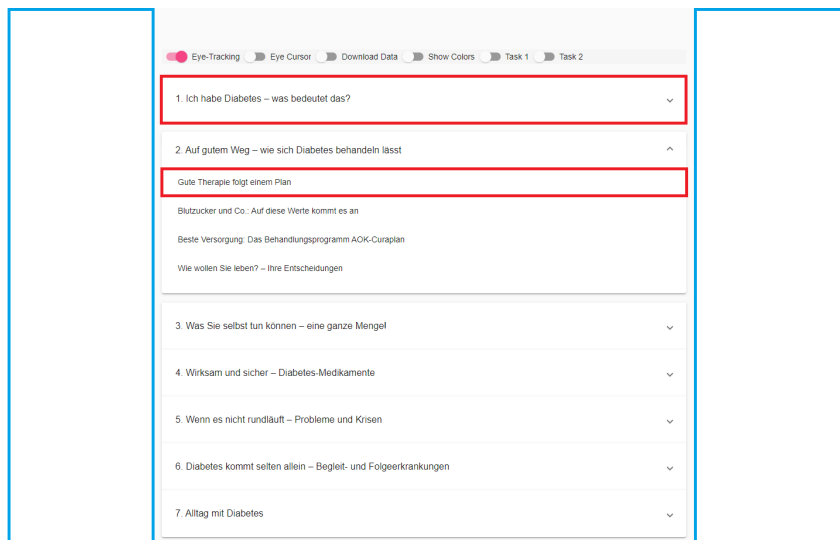


Figure 3.2: Areas of Interest in the chapter hierarchy. Two examples of AOIs to open chapters are highlighted in red. The blue rectangles show the AOIs for the redaction option to collapse all chapters.

time. In general, the approach should speed up the search for information. If the dwell time is chosen too long the time gained by using the eye for navigation gets eaten up by the dwell time. A long dwell time also feels unnatural and annoying for the user. A too short dwell time leads to unintended actions. The user just wants to explore the content and triggers actions with his or her gaze without wanting to, see [Midas Touch Effect](#). In this project, the dwell time was, with 1.5 seconds for the outer chapters and 2 seconds for the inner chapters, chosen to be longer than in other comparable works. This is due to the text the user has to read in the AOI, see [Figure 3.2](#). The values were chosen based on the authors perception and evaluated in the conducted user study, see [Section 5.2](#).

Another thing that has to be taken into account when using eye tracking as a control input is, that humans get distracted by movement in peripheral vision. For the dwell time, this entails that the focus can be pulled away by movement on other parts of the screen. As a result, it is important to avoid movement in the peripheral vision.

### 3 Concept and Approach

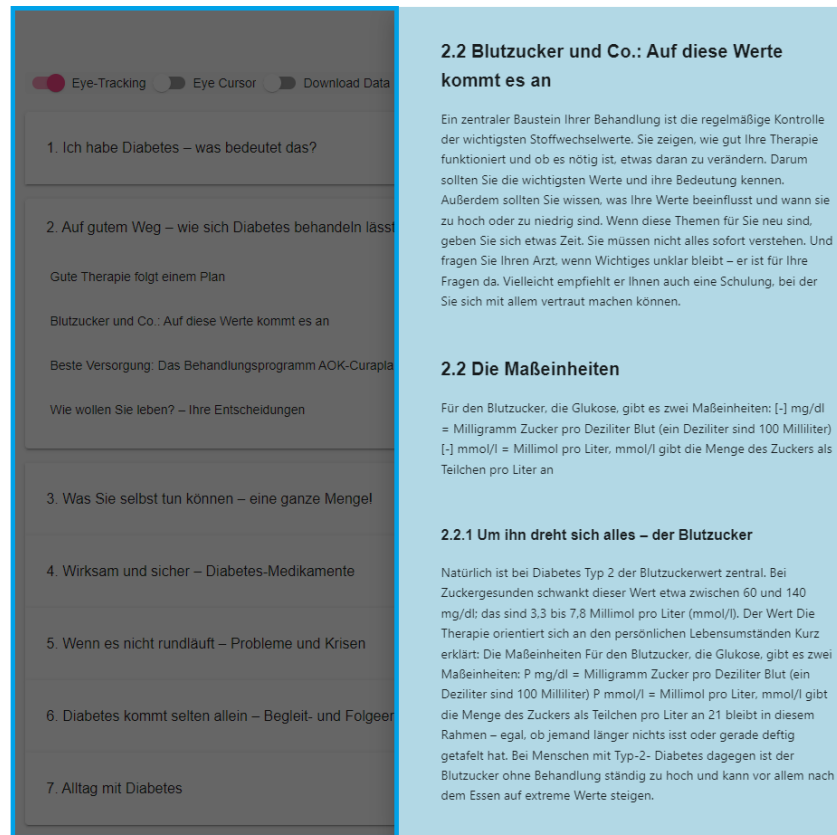


Figure 3.3: After focusing on an inner chapter for 2 seconds this blue side panel opens, showing the full-text information for the according chapter. The blue rectangle on the left illustrates the AOI to close the side panel.

#### 3.1.1 Redo Options

As gaze-based controlling and navigation are, in general, more error-prone than using a mouse or keyboard [19], it is very important to provide fast redo options. Up to a certain limit, this makes errors less invasive and leads to a higher acceptance of imperfections. To redo the expansion of chapters, the user has to look outside the TOC, see blue rectangles in Figure 3.3, for one second. To close the full-text information the user has to look outside the side panel, see the blue rectangle in Figure 3.2, for two seconds. These values were chosen based on the authors' perception and evaluated in the



conducted user study, see [Section 5.2](#).

### 3.1.2 Gaze-based Scrolling

To avoid using the mouse in addition to the gaze-based target selection, gaze-based scrolling was implemented. For each coordinate frame, the coordinates are checked if they appear to be below or above a fixed threshold near the lower respectively the upper edge of the screen. If this is the case, the page scrolls down respectively up. To be exact, the threshold is set 50 browser pixels below/above the lower/upper edge of the screen. If the side panel is closed and only the TOC is shown, the page scrolls 15 browser pixels up respectively down, if the side panel is opened it moves 10 pixels. These values were also chosen on the authors' perception and evaluated in the user study.

## 3.2 Logging User Data

As proof of concept that the tracked coordinates can be used for an adaptive and personalized presentation of data, our tool logs the time the user looks at a chapter in the TOC, in addition to the gaze-based interaction concepts. This information can be used later to recommend unseen chapters, similar chapters, or other supplementary content. In addition to this, our tool also logs the words the user reads. This functionality provides us with even more valuable data. Not only the content the user reads gets logged, but also the reading type or level of focus could be inferred, see [Related Work](#).



# 4 Implementation

## 4.1 Framework

The A+CHIS project consists of a web application, where the back-end API is based on Django<sup>1</sup>. Angular<sup>2</sup> is used for the front-end. This work got implemented into the existing project where the eye-tracking interface is built into the back-end. The obtained data gets sent via WebSocket to the front end where the coordinates get processed and used for interaction.

## 4.2 Eye Tracking

The hardware which was used for eye tracking is called The Eye Tribe. The eye tracker comes with tracker software, which, besides the so-called EyeTribe Server, includes a user interface for calibration and other settings. This interface is called EyeTribe UI. On their website [1] The Eye Tribe Company also provides Software Development Kits (SDKs) either in C#, C++ or Java. The eye-tracking interface for this project, however, is a Python interface taken from GitHub<sup>3</sup> and was built into the back-end.

### 4.2.1 Eye Tracking Technology

The eye-tracking hardware consists of infrared sensors and a camera built into a small device, see Figure 4.1. The eye-tracking coordinates get cal-

---

<sup>1</sup><https://www.djangoproject.com/>

<sup>2</sup><https://angular.io/>

<sup>3</sup><https://github.com/baekgaard/peyetribe>



Figure 4.1: The Eye Tribe tracker.

culated based on the images via computer vision algorithms. The use of infrared light, aimed at the eye, provides better results with head movements as a so-called "corneal reflection" is used as a reference point [15].

### 4.2.2 Eye Tracker Usage

The eye tracker device gets placed on a mini tripod below the screen centered relative to the monitor, see [Figure 4.2](#). The maximum screen size is limited to 24". The tracker is connected to the device via USB 3. For good tracking results, calibration is needed. This can be done via the EyeTribe UI software, see [Figure 4.3](#). The software also provides a post-evaluation feature to check the accuracy of the calibration process.

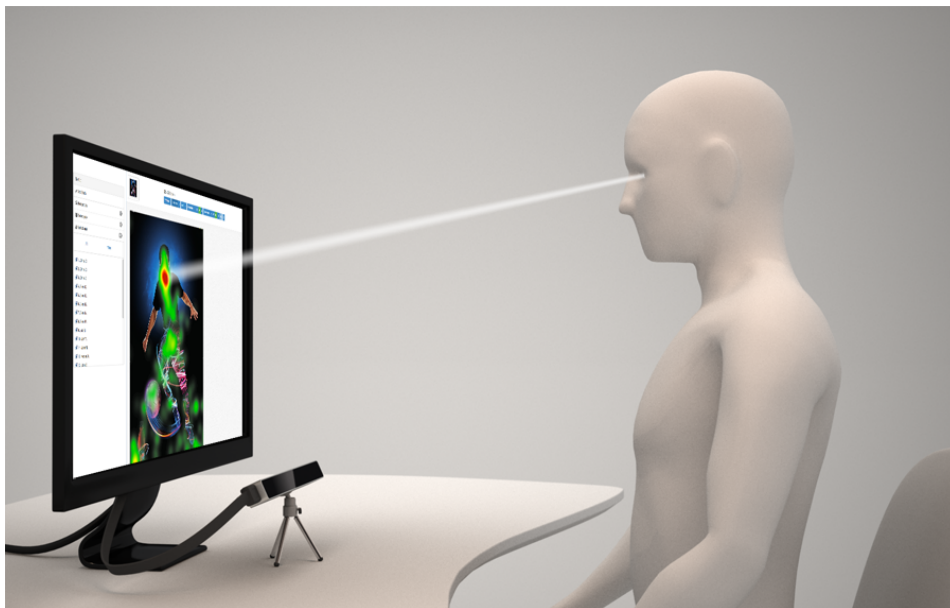


Figure 4.2: Eye tracking scenario. The Eye Tribe Tracker gets placed on a mini tripod centered relative to the monitor. Image taken from [1].

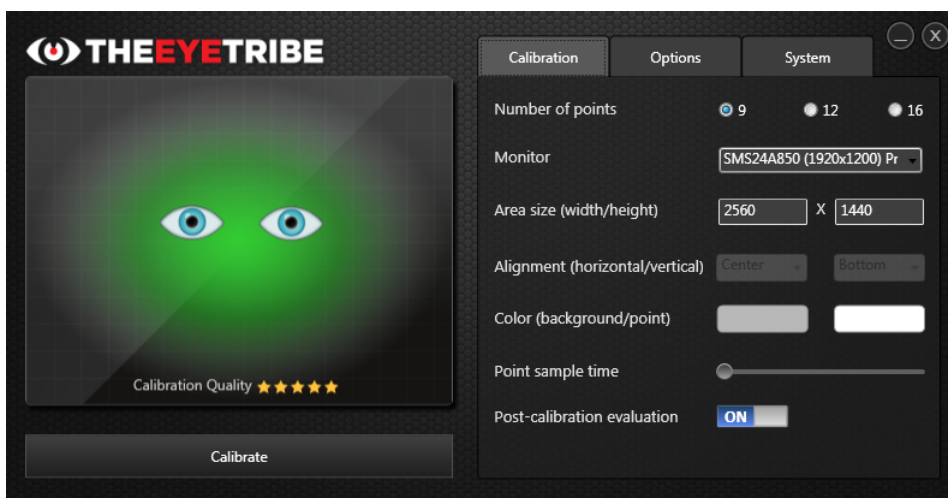


Figure 4.3: Screenshot of the provided user interface called EyeTribe UI. Image taken from [1].

### 4.3 Program Pipeline

Figure 4.4 visually demonstrates the program pipeline. From the Eye Tribe Server via the back-end of the A+CHIS project, the data gets sent to the front-end, where the main part of this work is implemented. The following sub-chapters cover the shown pipeline in more detail.

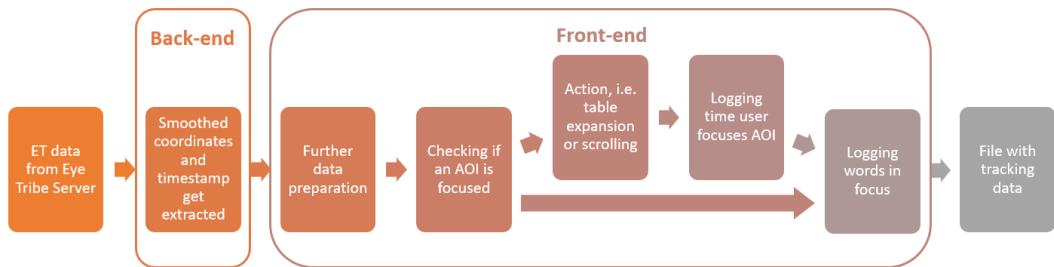


Figure 4.4: Here the program pipeline is shown.

#### 4.3.1 Data Preparation

The tracking software provides different kinds of data. Per frame with the according timestamp, we acquire the raw gaze coordinates but also smoothed coordinates, pupil size, and pupil coordinates. Each of these values is also available per eye. As soon as eye tracking gets activated in the front-end a message is sent to the back-end via a WebSocket to start pulling data from the tracking server. Since the trembling of the eye during fixations would negatively influence the usability, only the smoothed coordinates with the according timestamp get sent from the back-end to the front-end. As the coordinate data is given in pixels based on the screen resolution it has to be processed further by converting it into browser pixels. The implementation also provides the possibility to activate an eye cursor. This cursor is a small blue circle, showing the user, comparable to the normal mouse cursor, the currently tracked eye coordinates on the screen. This is useful to discover inaccuracies. This feature, however, is optional, because we experienced that the cursor distracts the user, who is often following

the cursor with the eyes to handle inaccuracies instead of looking at the information he or she is interested in.

### 4.3.2 Features

The main features of this work are target selection to navigate through the TOC and the according full-text information, the scrolling feature to replace the mouse wheel as well as the logging of gaze-based user data.

#### 4.3.2.1 Target Selection

The main part of this work is based on dwell time target selection. Starting from a TOC that only contains the main chapters, the user can open further subchapters by focusing on the chapter he is interested in, see [Figure 4.5](#). If the user then focuses on a subchapter for a given amount of time, the full text appears on the right-hand side of the table, see [Figure 4.6](#). To redo this option, more specifically to close the blue text field, the user has to look back to the left, outside of the blue field. If the side-panel is closed and the user looks outside the table, the chapters collapse and the table returns to its initial state, see [Section 3.1](#) for visualizations of the possible actions.

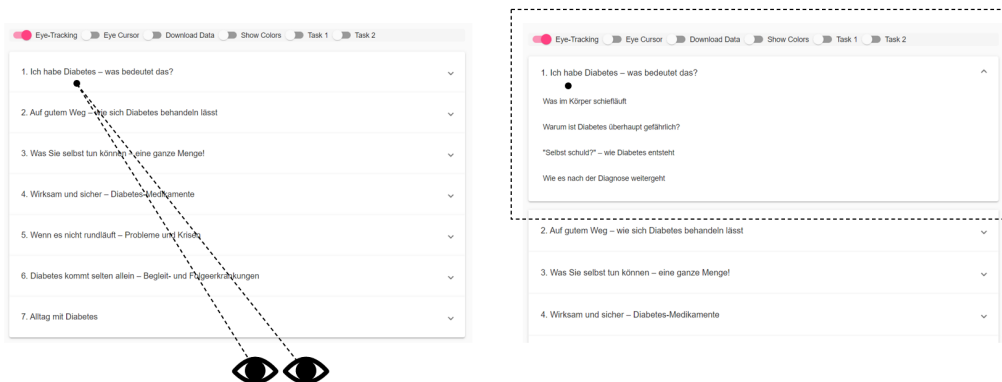


Figure 4.5: Dwell time click to show subchapters. If the user focuses on a certain chapter, e.g. chapter one, for 1.5 seconds the TOC expands.

## 4 Implementation

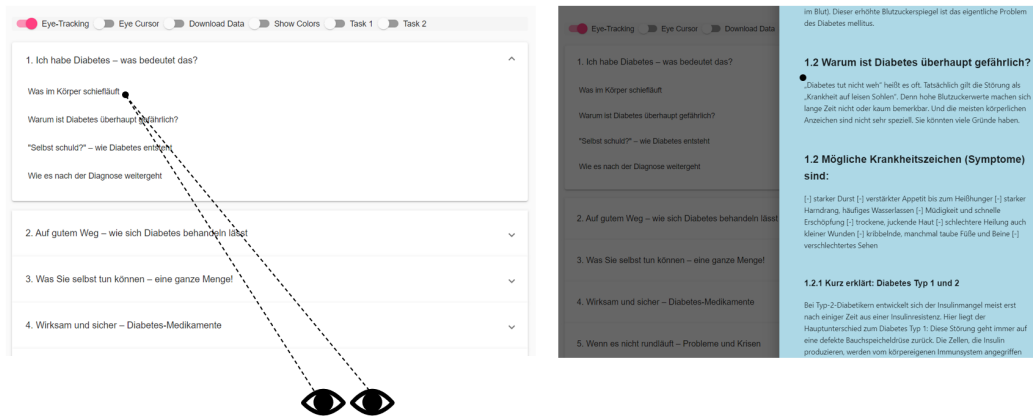


Figure 4.6: Dwell time click to open the side panel showing the full-text information. If the user focuses on a subchapter, in this case, chapter 1.1, for 2 seconds the blue side panel appears on the right-hand side.

### 4.3.2.2 Scrolling

Additionally, if the user looks at the top respectively the bottom edge of the screen, the page scrolls. This works in the TOC as well as the side panel. More precisely, there is a y-axis threshold of 50 pixels at the bottom and the top so that if the user's gaze falls below respectively exceeds this limit the page scrolls. This threshold could be customizable in the future, see [Future Work](#).

### 4.3.2.3 Logging User Data

The second part of this work is about logging the information which can be retrieved from the gaze coordinates. On one hand, the program logs how long the user looks at the individual chapter headings in the table of contents. On the other hand, it saves the word the user looked at for each timestamp. The logged data gets saved in a .json file to be downloaded from the browser after eye tracking gets stopped. For each chapter and sub-chapter in the TOC the user looked at once, there is an entry with the time in milliseconds. For each word the user looked at, the number of timeframes, the location of the word, and the timestamps are saved.



# 5 Results and Evaluation

The evaluation of this project consists of feedback obtained from ten participants of a user study. Additionally, demo videos have been provided to the institute.

## 5.1 Apparatus

Table 5.1: Technical specifications. Values taken from [1].

Frame rate	30 Hz / 60 Hz
Accuracy	0.5° - 1°
Calibration	9, 12 or 16 points
Calibration sample time	500 - 2000 ms
Calibration transition time	1 - 2000 ms
Screen size	max 24"
Distance to screen	45 - 75cm
Connection	USB 3.0 SuperSpeed
Calibration software	EyeTribe UI

Table 5.1 shows the technical specifications of The Eye Tribe tracker. The table contains specifications listed on the manufacturers' website [1] as well as the software tool EyeTribe UI, which provides different settings for calibration and the tracking itself, as mentioned in Section 4. Besides the frame rate, there are many adjustments possible regarding the calibration of the eye tracker. For this project, we used twelve calibration points with a sample time of about 1 second and a transition time of about 1.1 seconds as we experienced satisfying calibration results with these settings. The screen size was 15" and the distance to the user was usually between 50 and 60 cm. The frame rate was set to 30 Hz.

After calibration, the calibration software provides a post-evaluation feature consisting of a rating between 1 and 5 stars, where 5 stars correspond to a perfect calibration attempt, see Figure 5.1.

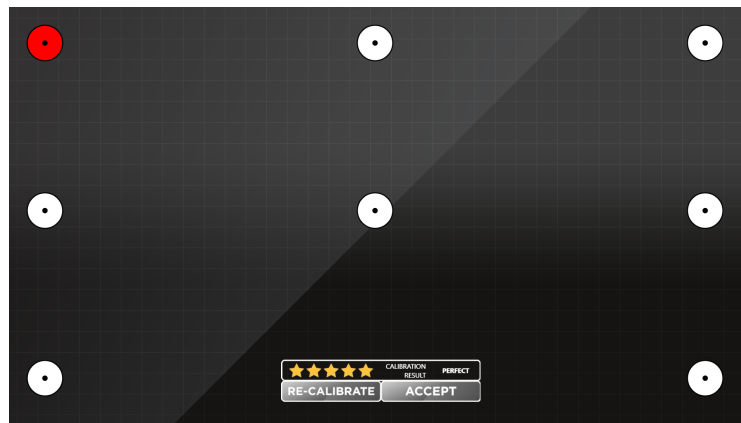


Figure 5.1: Post-evaluation tool of EyeTribe UI showing a perfect calibration attempt. The red dot marks the area of the currently tracked gaze coordinates for the user to check the calibration quality.

## 5.2 User Study

To evaluate the usability of the implementation, we conducted a user study with ten participants. The user study consisted of exploring the tool to gain

qualitative feedback as well as given tasks to gather quantitative feedback from each participant.

### 5.2.1 Participants and Tracking Environment

To find participants an email request for proposals was sent out to employees of the institute as well as to fellow students. Most of the chosen attendees were between the ages of 18 and 25, but overall the participants were between 18 and 45 years old. Two participants wore glasses, three wore contact lenses. However, there were no noticeable differences in tracking quality. Only two of the ten participants have used an eye tracker before. No attendee had any problems with calibrating the eye tracker as everybody received a five-star rating from the EyeTribe UI software. The user study was conducted in a room with little indirect sunlight, as we experienced the best tracking quality with indirect, low light.

### 5.2.2 Procedure

First, each participant filled out a short questionnaire regarding personal information like age, reading habits, and their previous eye-tracking experience as well as if they wear glasses or contact lenses. Then, each participant was asked to calibrate the eye tracker. The calibration was repeated until 5 stars were reached. Usually, they already obtained this result after the first calibration attempt, on average they needed 1.6 attempts. Some participants had to re-calibrate the eye tracker in between the tasks due to inaccuracy or head movement. After calibration, the attendees were asked to explore the website for about two minutes without activation of the eye cursor feedback. We chose this approach because we experienced that, if the user sees the tracked gaze point, he follows this point with his or her eyes trying to compensate for inaccuracies instead of focusing on the information and actions.

Afterward, each participant was asked to write down ten words he or she remembered from exploring the website. This data was then compared to the most logged words to determine if and how well this data can be used

to for example derive where the user's interests lie. At this point, the users were asked to give verbal feedback on the usability of the tool. Afterward, the participants were given two tasks each consisting of a given word and the according chapter. They were asked to search and focus on this word in as short a time as possible. This data was intended to compare users based on the time they needed for this task and their reading habits. However, the results of these tasks were mostly eliminated from the study mainly because the number of participants was too low to make conclusions and results were affected by inaccuracies. Also some participants found the terms randomly, which made the data unusable.

### 5.2.3 Feedback

The general experience was good, but some participants saw room for improvement. Half of the attendees were, to some extent, dissatisfied with the length of the dwell time. It was either too long or too short. This argues for a user-specific dwell time, which we cover in [Section 6](#) further. The verbal feedback of most participants included small inaccuracies, especially for the y-axis. The tracked gaze point often appeared a little bit (about 0.5cm) below the actual gaze point. This was especially inconvenient for opening the side panel with full-text information. As the AOI for this action is smaller concerning the y-axis, more errors respectively unintentional actions occurred here. Another thing participants reported was the missing visual feedback. As the user does not know how far the dwell time has progressed, he or she has no chance to prevent a potentially unintentional action. The use of visual feedback is therefore another part to be covered in the [Future Work](#) section. Also, the problem that blinking sometimes triggers the scroll-down feature was reported twice. This unintentional scrolling distracts the user from reading.

The participants were asked to write down ten words they remembered from exploring the tool. Most participants, however, were focused on exploring the eye-tracking functionality and were not able to remember a full ten words. Of the words the participants wrote down, nevertheless, for most participants more than half were logged frequently which suggests that the word logging works as intended, see [Figure 5.2](#). This feature depends

very much on the accuracy of the eye tracker, especially in continuous text. Here it would help to increase the line spacing or use other information visualizations.

## 5 Results and Evaluation

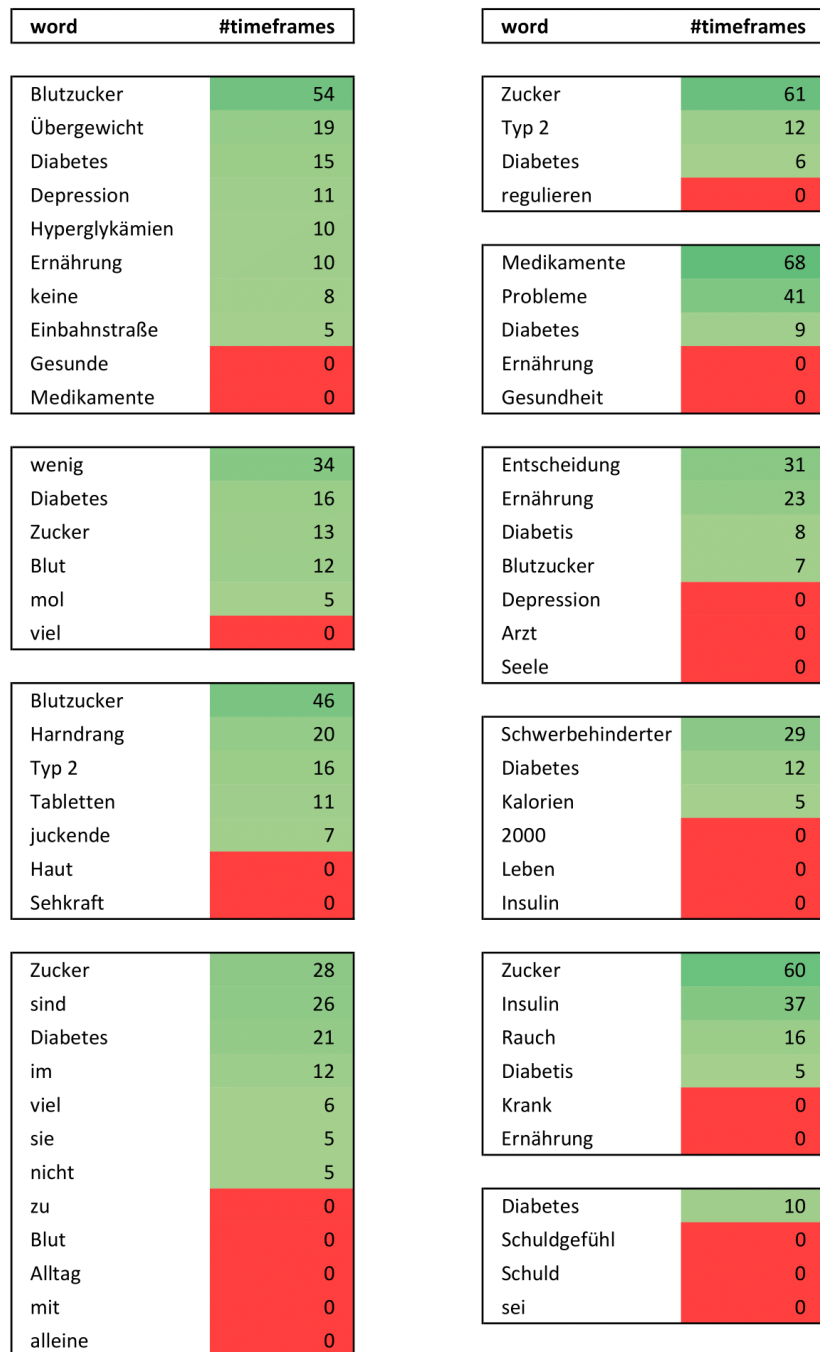


Figure 5.2: This figure shows the words each participant listed during the user study with the number of timeframes the word got logged in.

## 6 Future Work

Based on the user study feedback, there are several possible expansions of this project. On one hand, visual feedback could be added to test if this improves usability. For example, a progress bar could tell the user how far the dwell time has progressed. Instead of a progress bar coloring could also be used to indicate the time progress. With visual feedback, errors could be prevented as the user could stop an unintended action before it even happens. Another additional visualization that would be useful for the user is to visualize what he or she already looked at. This way the user would have a good overview of what has been seen already. A reduced version of such a visualization has been implemented, but could be extended.

On the other hand, customizing values, like the scrolling threshold or dwell time, based on the user's preferences, could also improve the individual user experience. This customizing could be done manually by each user. It might even be possible to set the values based on the gained user data. Indicators to change the current threshold respectively dwell time could be how frequently the user uses a redo option just after the associated action. It might also be possible to infer a suitable dwell time, based on reading style or other user data.

Additionally, the logged data can be used in future projects for the adaptive presentation of information. There could be separate recommendations of a topic the user has not seen yet or information that is similar to already explored topics. Also, the information could be presented in different kinds of visualizations, not only full-text information. Visualization examples would be graphs, charts, pictures, or videos. The logging functionality could also be extended to the whole project to gather more data about the user to be able to refine individual user profiles and improve the individual presentation of information.





## 7 Conclusion

This thesis aimed to study the use cases of eye tracking in document exploration. Based on feedback from a small user study, it can be concluded that the use of eye tracking for interaction with the computer is already feasible with the implementation done in this work, but that there are several extensions that might optimize the usability of this use case. The second part of this work also proved that useful data about the users' interaction activities can be collected that might be useful in future works.

Our approach was to use dwell time for navigation as this approach does not include other additional manual input. The search for information in the document should be comfortable, easy and fast while the gathered user data should provide conclusive information about the user to implement in future works. Although the results suffered from slight inaccuracies, altogether, they indicate a promising outcome. In the conducted user study, participants were able to navigate the website just using their eyes, while the data which was collected also matched the information the user remembered from exploring the document.

Further usability improvements to circumvent problems with inaccuracies, like status visualizations or personalized adaption of interaction concepts, are promising future project extensions. Evaluating the value as well as using the data collected in this project would also be a very interesting continuation.

Solely using an eye tracker for document exploration has proven to be not only feasible but could open new ways for personalizing the search for information. Especially in a project like A+CHIS, affected persons can thereby find what they need easily and effectively, as the various needs of health information of individual persons are taken into account.



## Bibliography

- [1] TheEyeTribe: Developer Site. <https://theeyetribe.com/dev.theeyetribe.com/dev.theeyetribe.com/general/index.html>, 2014. Online; accessed November 14, 2022.
- [2] Michael Ashmore, Andrew T Duchowski, and Garth Shoemaker. Efficient Eye Pointing with a Fisheye Lens. In *Proceedings of Graphics Interface*. Citeseer, 2005.
- [3] Richard Bates and Howell Istance. Zooming Interfaces! Enhancing the Performance of Eye Controlled Pointing Devices. In *Proceedings of the International ACM Conference on Assistive Technologies*, 2002.
- [4] Andreas Bulling, Jamie A Ward, and Hans Gellersen. Multimodal Recognition of Reading Activity in Transit Using Body-Worn Sensors. *ACM Transactions on Applied Perception (TAP)*, 9(1):1–21, 2012.
- [5] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Tröster. Robust Recognition of Reading Activity in Transit Using Wearable Electrooculography. In *Proceedings of the International Conference on Pervasive Computing*. Springer, 2008.
- [6] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Tröster. Eye Movement Analysis for Activity Recognition. In *Proceedings of the International Conference on Ubiquitous Computing*, 2009.
- [7] Andreas Bulling, Christian Weichel, and Hans Gellersen. EyeContext: Recognition of High-Level Contextual Cues from Human Visual Behaviour. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.

- [8] Cristina Conati, Giuseppe Carenini, Dereck Toker, and Sébastien Lallé. Towards User-Adaptive Information Visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.
- [9] Andrew T Duchowski. Gaze-Based Interaction: A 30 Year Retrospective. *Computers & Graphics*, 73:59–69, 2018.
- [10] Andrew T Duchowski, Nathan Cournia, and Hunter Murphy. Gaze-Contingent Displays: A Review. *Cyberpsychology & Behavior*, 7(6):621–634, 2004.
- [11] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2017.
- [12] Fabian Göbel, Peter Kiefer, Ioannis Giannopoulos, Andrew T Duchowski, and Martin Raubal. Improving Map Reading with Gaze-Adaptive Legends. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2018.
- [13] Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. Designing Gaze Gestures for Gaming: An Investigation of Performance. In *Proceedings of the Symposium on Eye-Tracking Research & Applications*, 2010.
- [14] Robert J. K. Jacob. The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. *ACM Transactions on Information Systems (TOIS)*, 9(2):152–169, 1991.
- [15] Päivi Majaranta and Andreas Bulling. Eye Tracking and Eye-Based Human–Computer Interaction. In *Advances in Physiological Computing*, pages 39–65. Springer, 2014.
- [16] Lin Shao, Nelson Silva, Eva Eggeling, and Tobias Schreck. Visual Exploration of Large Scatter Plot Matrices by Pattern Recommendation Based on Eye Tracking. In *Proceedings of the ACM Workshop on Exploratory Search and Interactive Data Analytics*, 2017.

- [17] Zohreh Sharafi, Bonita Sharif, Yann-Gaël Guéhéneuc, Andrew Begel, Roman Bednarik, and Martha Crosby. A Practical Guide on Conducting Eye Tracking Studies in Software Engineering. *Empirical Software Engineering*, 25(5):3128–3174, 2020.
- [18] Dereck Toker, Cristina Conati, Ben Steichen, and Giuseppe Carenini. Individual User Characteristics and Information Visualization: Connecting the Dots through Eye Tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.
- [19] Roel Vertegaal. A Fitts Law Comparison of Eye Tracking and Manual Input in the Selection of Visual Targets. In *Proceedings of the International Conference on Multimodal Interfaces*, 2008.