GazeTheWeb - Tweet

Project Lab and Research Lab at University of Koblenz-Landau

by

Svenja Neuneier, Dennis Hahn, Caterine Ospina Ocampo, Sergei Diez, Saskia Handabura, Kim Ballmes, Wojciech Kwasnik, Benjamin Stephan, Eike Idczak, Matthias Barde, Annika Wießgügel, Philipp Weber

Supervised by: Chandan Kumar, Raphael Menges

Koblenz, WS 2015/16

Contents

Chapter 1

Introduction

Disabled people are often shut off from the world due to their disability. Since they cannot leave the house on their own their social contacts suffer and they tend to live a secluded life. Modern technology however offers the possibility to keep on track with the world, rebuild and keep social contacts and regain personal independence. Social networks in particular can be like an open window to the world for disabled people - if they are able to use such pages. Since most social network platforms are not built for other input methods than using a mouse there is a need to enable physically disabled people to access them the way their disability allows them to. This means there is a need for social network implementations that support eye tracking, brain interactions or speech input.
Project MAMEM (short for Multimedia Authoring and Management)[1] currently researches in the direction of disabled people using social networks by extending the APIs of current operating systems to accept brain or gaze input.

> "MAMEM's final objective is to assess the impact of this technology in making these people more socially integrated by, for instance, becoming more active in sharing content through social networks and communicating with their friends and family." [mam]

Being part of this project, the research lab wants to take a step into this direction as well by building an eye tracking application on top of a common social media platform so that it will be available for physically disabled people. Most of the big platform APIs have too much restrictions on what you can achieve with it as they do not provide all functionalities. To let the user experience the eye tracking application to the same extent as if he was using the regular application, the platform that has been implemented and redesigned is "Twitter"[2].

Structure of the report. The following report will have a look on the technologies that served as basis for the application in section 2.2. Chapter 3.1 will head deeper into the application and explain the structure that has been chosen and the functionalities that have been realized of the regular twitter ones. The implementation side of the application as well as some information on which obstacles had to be overcome and how the lab itself has been structured can be found in chapter 3.1. Chapter 4 covers the final evaluation. A conclusion of the report can be found in chapter 5.

[1] MAMEM - Multimedia Authoring & Management using your Eyes & Mind: `http://www.mamem.eu/` [Accessed: 2016-02-24]
[2] Twitter: `http://www.twitter.com`. [Accessed: 2016-02-24]

Chapter 2

Technologies

At the beginning it has not been clear what kind of application should be the goal of the research lab. After a short time the decision has been made to do a social network instead of a web search application. The latter would not have been enough work for a project team of this size. To find the social media platform that suits the lab's goals best, different APIs (short for *Application Programming Interfaces*), namely Facebook[1], Telegram[2], Twitter[3], Youtube[4], Flickr[5] and Google+[6], have been evaluated. While some of them are missing important functionalities (Facebook, Google+)others do not have a reasonable amount of functionalities and challenges for a research lab (Flickr, Youtube). Twitter has than been the API of choice, it provides a reasonable amount of challenges concerning eye tracking, enough functionalities to have really use it afterwards and also enough work to distribute it among twelve people.
In general the application has been written in C++11 using the following technologies. It does not depend either on Windows or Linux as it has been developed on both operating platforms.

2.1 Twitters REST APIs

A central question developing the application was how to realize the connection to Twitter to be able to read and write Twitter data. Twitter provides a large collection of well documented REST APIs for this purpose[7].
Twitter only allows access to these APIs via so called *Twitter Applications*[8]. When an application is registered it is able to authenticate against Twitter via OAuth[9] or Application-only authentication[10] to communicate with the APIs.
It is important to understand that such an application is not the same thing as a Twitter (user) account. For example, if one wants to check his timeline using Twitters REST APIs, one would have to authenticate his Twitter application at first. Then he has to grant the application access

[1] Facebook for Developers: `https://developers.facebook.com/`. [Accessed on 23.04.16]
[2] Telegram APIs: `https://core.telegram.org/api`. [Accessed on 23.04.16]
[3] twitcurl: `https://github.com/swatkat/twitcurl`. [Accessed on 23.04.16]
[4] Youtube for Developers: `https://www.youtube.com/yt/dev/`. [Accessed on 23.04.16]
[5] The App Garden: `https://www.flickr.com/services/api/`. [Accessed on 23.04.16]
[6] Google+ API: `https://developers.google.com/+/web/api/rest/`. [Accessed on 23.04.16]
[7] Rest APIs: `https://dev.twitter.com/rest/public`. [Accessed on 23.04.16]
[8] Twitter Application Management: `https://apps.twitter.com/`. [Accessed on 23.04.16]
[9] OAuth: `https://dev.twitter.com/oauth`. [Accessed on 23.04.16]
[10] Application-only authentication: `https://dev.twitter.com/oauth/application-only`. [Accessed on 23.04.16]

to his Twitter (user) account. In this phase he usually would have to log in to his account and push a button to authorize the application. However, this process can be handled automatically – as it is done in *GazeTheWeb - Tweet*– but at least the user has to provide his personal Twitter credentials. Finally the application is able to access the API endpoint which delivers the timeline for the user.
Furthermore the Twitter APIs have rate limits[11]. Depending on the authentication method these limits are calculated per-user using the application or for the whole application – which is a difference, as stated before. Because every endpoint requires an authentication, every API call is affected by this limitations and if they are exceeded it is not possible to make further calls. This means an application is restricted in the number of API calls it can perform in a certain time interval. Twitter uses 15 minute windows as intervals. The amount of calls in such a window that can be performed depends on the endpoint. For example one could access the endpoint "search/tweets2 180 times in a time window, to search for tweets. But accessing the follower list by using the endpoint "followers/list", is only possible 15 times. So the limits are not too strict if you think of an usual using behavior, but you have to keep an eye on them and be aware of the possibility of limit violations which will result in certain error responses by the API.

2.2 twitcurl

A wide range of libraries exist to access Twitters REST APIs in different programming languages. Using C++ only one library, *twitcurl*, is officially listed by Twitter[12]. *twitcurl* is a pure C++ library for accessing the REST APIs of twitter and currently supports many of the v.1.1 APIs. It uses the *cURL* library[13] to handle HTTP requests and responses.
Although not all functionalities needed for the application were already implemented in *twitcurl* the decision to base implementation on it was made because it provides a solid basis. As mentioned the handling of HTTP requests and responses is already implemented as well as the OAuth authorization methods which are required to authorize the application against the Twitter APIs to be able to use them, as mentioned above. This makes it easy to extend or update the functionality of the library.
To realize the integration of *twitcurl* the static library file of it as well as the headers have to be included to the project. In fact including the whole source code of *twitcurl* became necessary because some issues came up accessing the static library file with current versions of Microsoft Visual Studio. This setup also made it easier to apply changes to the library. This was useful because some methods were missing, outdated or did not match the requirements of the application. To encapsulate access to the library the wrapper class *twitterClient* has been created. It provides methods to get data from or send data to twitter using the methods of *twitcurl* library which connects to Twitter REST APIs using *cURL* (see figure 2.2). This way the class serves as link between Twitter and the rest of the application.
To make communication even easier a JSON parser has been added to the pipeline to parse the JSON strings achieved from API responses to JSON objects. This way the wrapper class directly returns JSON objects which can be handled very easy by the rest of the application. *RapidJSON*[14] has been selected for this task because it is a simple to integrate lightweight header-only C++ library which maintains a nice performance.

[11] `https://dev.twitter.com/rest/public/rate-limits`. [Accessed on 05.05.2016]
[12]Twitter libraries: `https://dev.twitter.com/overview/api/twitter-libraries`. [Accessed on 24.04.16]
[13]cURL: `https://curl.haxx.se/`. [Accessed on 23.04.16]
[14]RapidJSON: `https://github.com/miloyip/rapidjson`. [Accessed on 23.04.16]

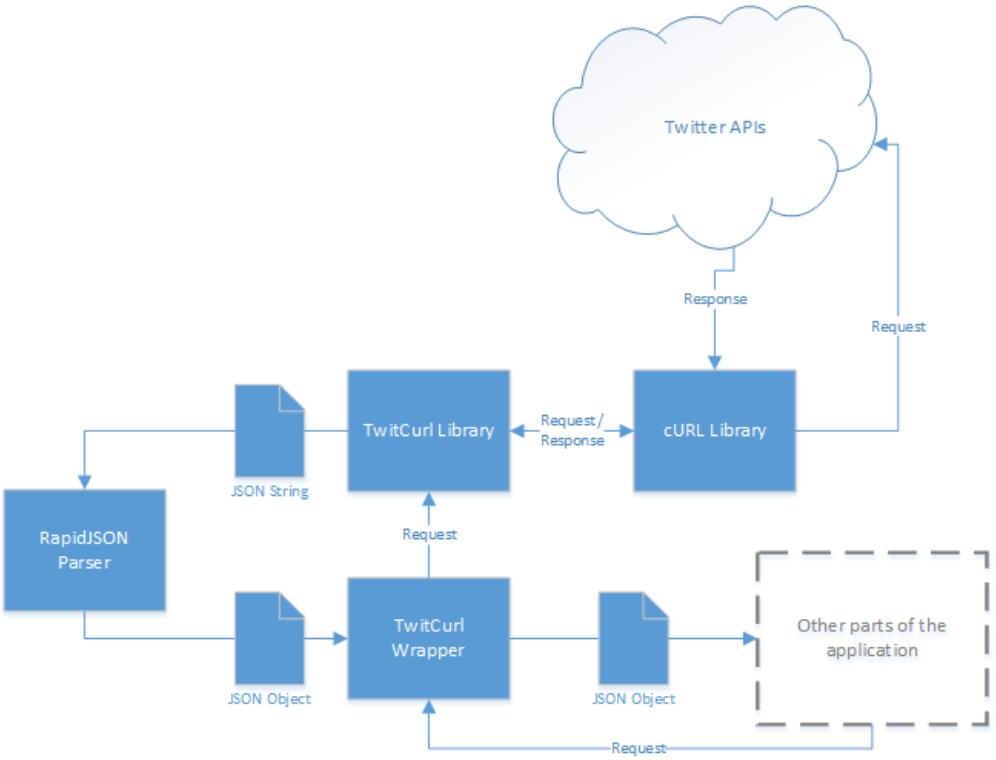

Figure 2.1: Structural integration of *twitcurl* library.

2.3 eyeGUI

The user interface of the application has been made using Raphael Philipp Menges' eyeGUI *eyeGUI*[Men], a library "to load, manipulate and render user interfaces for eye tracking input" [Men]. Using this library offers a way to be in unison with another MAMEM application, *GazeTheWeb - Browse*[15], a browser that can be controlled with an eye tracker. Both projects contributing to GazeTheWeb can be found in the official MAMEM GitHub repository [16].
eyeGUI is written in C++ 11 and based on OpenGL. It offers an easy way to build user interfaces for eye tracking by adding XML-files as layouts and manipulating elements, like buttons (see figure 2.3), within those layouts via listeners. Those listeners can later be accessed in the regular application code to give every button an own functionality and also interact with external APIs, like twitcurl.

```
<boxbutton id="mode" switch="true"></boxbutton >
```

Figure 2.2: Initialization of a simple button with *eyeGUI*(from [Men])

[15]GazeTheWeb - Browse:`https://github.com/MAMEM/GazeTheWeb/tree/master/Browse`. [Accessed on 23.04.16]
[16]GazeTheWeb: `https://github.com/MAMEM/GazeTheWeb` [Accessed on 26.04.2016]

A variety of elements, like buttons, images and texts, to build a proper interface can be used from the library. Most of them can be customized in their size, appearance or behaviour. Buttons for instance can be given an arbitrary background image that scales automatically when the overall size of the interface changes, though the ratio of height and width of the image stays the same.

All elements are designed especially for eye tracking in their size and the way the user interacts with them. Buttons for instance get activated when the gaze hits them, and shrink during the time it needs to trigger the button. A colored overlay increasing in size works as a visual representation of the remaining time until triggering.
eyeGUI also offers other eye tracking-specific features like optionally showing the gaze path during usage. Control can be switched between mouse-controlled for development purposes and eye-controlled for actual utilization.
An introductory video on how the library and its elements work, can be found in the repository of the library itself.

Chapter 3

GazeTheWeb - Tweet

As the foundation of *GazeTheWeb - Tweet* has now been described by the technologies, this chapter gives an insight into how the layout of the application has been designed. The functionality the application provides that the users can expect from it and also a short overview over the design process of the interface. There will also be details of the implementation process and the development on the application in general.

3.1 Structure and Functionality

Being part of the MAMEM project *GazeTheWeb - Tweet* aimed at being controlled entirely with the eyes. An user should be able to lean back and navigate through the application without touching a mouse but still using his regular Twitter account. The application offers some of the most used Twitter functionalities. First of all, the user should be able to log into the account with his own credentials. An interface which is adjusted to eye tracking offers functionalities like "send a tweet", "go to the start page", "scroll through recent tweets of followers" and "view own profile". The application should also offer a way to explore new topics and acquaintances on Twitter. For disabled persons this feature is important to keeping up with current events and to stay in touch with friends . Those friends should be able to communicate via private messages (messages that can only be exchanged between two mutually following persons). In order to avoid frequent switching between different applications typing is done with a custom keyboard designed for eye tracking support. All standard actions like "Retweet", "Respond", "Like" or "Follow" should be provided for each tweet and displayed in a way that supports eye control.
In the mobile version of Twitter [1] all of those functionalities are present but they are not arranged in a way that they are easily accessible for the eyes. For the application *GazeTheWeb - Tweet* the structure of Twitter has been completely reworked and rearranged. On the one hand this means decreasing the amount of functionalities, which has also been induced by *twitcurl* but on the other hand it means a design where buttons are large enough and where application control and natural viewing behaviour do not interfere.

3.1.1 Structure

GazeTheWeb - Tweet offers such an interface by having areas that react differently to the gaze and whose positioning is consistent troughout the application. A first draft of a layout for such

[1] Twitter: `m.Twitter.com`. [Accessed on 27.04.2015]

an interface can be seen in figure 3.1. The general principle of the interface is described by Jacob and Keith:

> "Moving one's eyes is often an almost subconscious act. Unlike a mouse, it is relatively difficult to control eye position consciously and precisely at all times. The eyes continually dart from spot to spot, even when its owner thinks he or she is looking steadily at a single object, and it is not desirable for each such move to initiate a computer command."[JK03]

To achieve a desirable effect, the Content Area (2.) does not trigger any events. It is thought for reading the tweets and rest ones eyes. The Action Bar (3.) is connected to the Content Area. The tweet that is in the focus of the user is highlighted in the Content Area while all possibilities to interact with this tweet (Like, Retweet, View Profile, etc) are shown in the Action Bar. Hence all actions that a user can do with a tweet are displayed in a context sensitive menu but do not interfere with reading. The chance of clicking one of this options accidentally is reduced by this seperation of concerns.
The Navigation Bar (1.) always remains the same and offers the most important actions for Twitter (Browse, Profile, Home, Tweet, Private Messages and Search). Originally there had been also a "Back"-Button integrated into the Navigation Bar. This button has later been dismissed to have a more dynamic navigation within the application. The two Category Switch Bars (4.) on the left and right side of the application were also part of the first draft for the application design. Their functionality should have been the switching between different subcategories of one page. For the search this would have been one page showing only profiles, one page showing only Tweets and one page showing only hashtags. Therefore those two bars were planned to be enabled on some pages and disabled but still present on some pages.

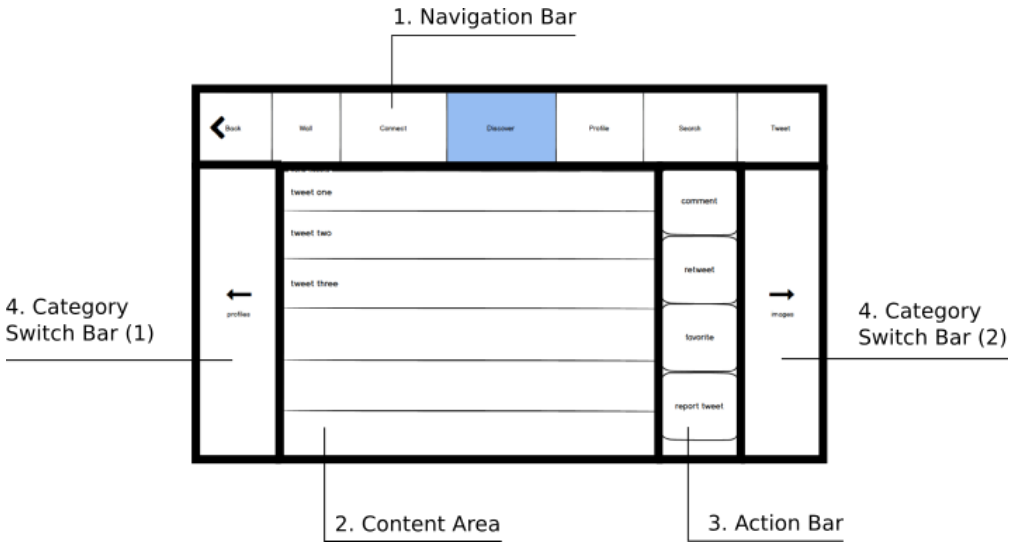

Figure 3.1: Initial page layout of the application *GazeTheWeb - Tweet*

Since the development team had not yet decided where exactly scrolling should be done, the Category Switch Bars were initially used for a kind of scrolling. Pressing one of those buttons would trigger a scrolling of four tweets at once, similar to switching a page in a book.In

the final version of the interface the categories that should fill the Category Switch Bars had all been merged into one page so that those two buttons had no functionality left and were discarded.

Various ways of scrolling had been discussed. Scrolling by looking at the upper or lower screen edge would have had the effect that if the user looks away from the screen, the application would start scrolling uncontrollably. Adding "up" and "down" buttons between the Menu Bar and the Content Area as well as below the Content Area would have added a button that could be pressed accidentally on the way to the menu bars. Also, adding "up" and "down" buttons to the action bar would have used too much space that is needed for other options. The Action Buttons then would have been too small to reach them with the eyes. Finally scrolling had been implemented implicitly. Looking at the upper or lowermost tweet in the list moves the tweet to the middle and filles its space with newer or older tweets in the timeline.

3.1.2 Design

Figure 3.2 shows the final design of the "Home"-page of *GazeTheWeb - Tweet*. It has been realised as the last step of development. All previous designs (see the following figures) had a very linear design. This one instead featured huge, flat buttons with distinct icons.

Figure 3.2: Final Design of the "Home"-page of *GazeTheWeb - Tweet*

Highlighted in grey is the tweet the user is currently looking at. If the tweet is the one on top or at the bottom it is moved one space to the middle. The context sensitivity of the Action Bar is symbolized by the visual connection of Content Area and Action Bar. The triangular button above the tweets scrolls to the top/newest tweet and the button in the lower left corner exits the program. As can be seen in table 3.3 the other pages of the application look quite similar. The only exception, the login page, allows users to log in with their own credentials. By clicking on "username" or "password" the keyboard opens.

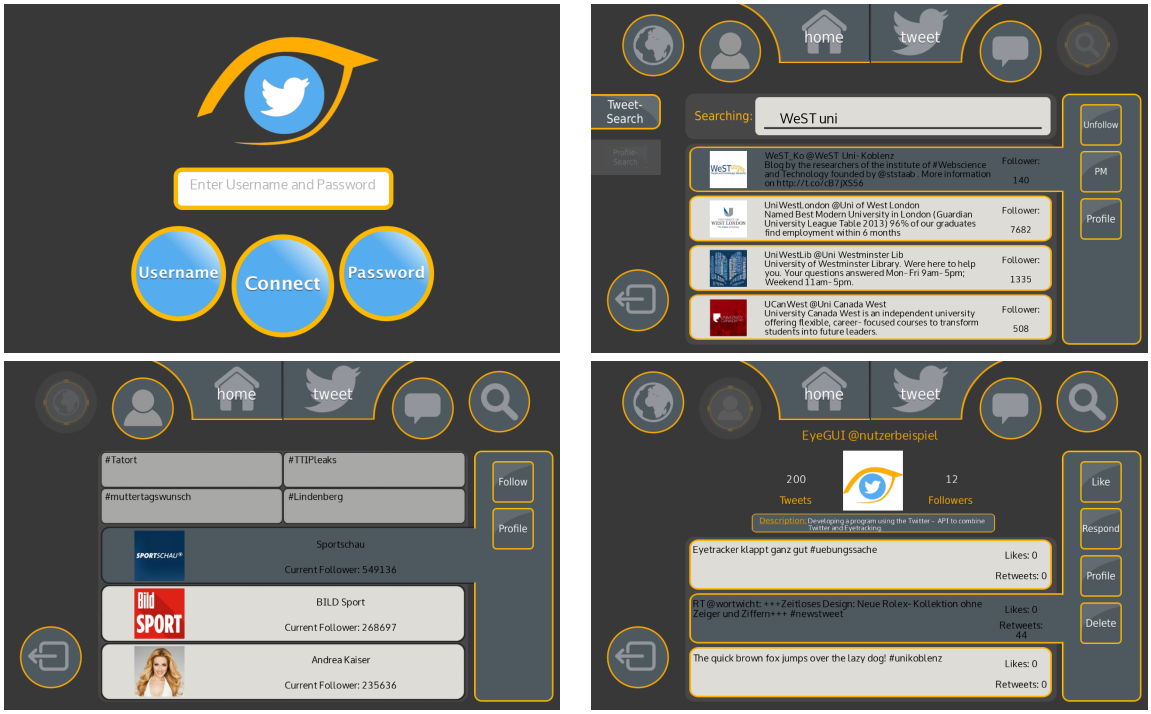

Figure 3.3: Login Screen (u.l.), Search Page (u.r.), Discover Page (l.l.) , Profile Page (l.r.)

The following images show the development of the interface during implementation. All of those designs were meant as temporary, experimental designs.
Table 3.4 shows the development of the user interface of *GazeTheWeb - Tweet*. The very first draft of a design can be seen in the upper left corner the version before the final one can be found in the lower right corner. From version 3 to version 4 the resolution of the application has changed a bit.

Typing is an important task while using Twitter. Since *eyeGUI* did not yet offer a keyboard for eyetracking there had been the need to implement an own keyboard for *GazeTheWeb - Tweet*. For the keyboard there had only been one design revision since numerous programming tasks had to be done previously. This delayed the revision of the design until the final colors and ideas for the design had come up. Figure 3.5 shows the final design while figure 3.6 shows the previous version of the keyboard. At least for letters the keyboard layout it similar to a standard QWERTZ-layout, though "Alphabetic order may be faster to learn. The keys could also be organized so that the most common letters are grouped together"[MR02] For non-disabled persons this keyboard layout is quite common and for the evaluation *GazeTheWeb - Tweet* has now been arranged equivalently to optiKey[2], the reference keyboard.(cf. chapter 4). Numbers, symbols and german umlauts are arranged on extra layers of the keyboard. They can be reached similar to the symbols on a smartphone keyboard.
The auto-completion offers the one hundred most often used words in the english language.

[2]OptiKey - Type, Click, Speak: `https://github.com/OptiKey/OptiKey/wiki`. [Accessed on 30.04.16]

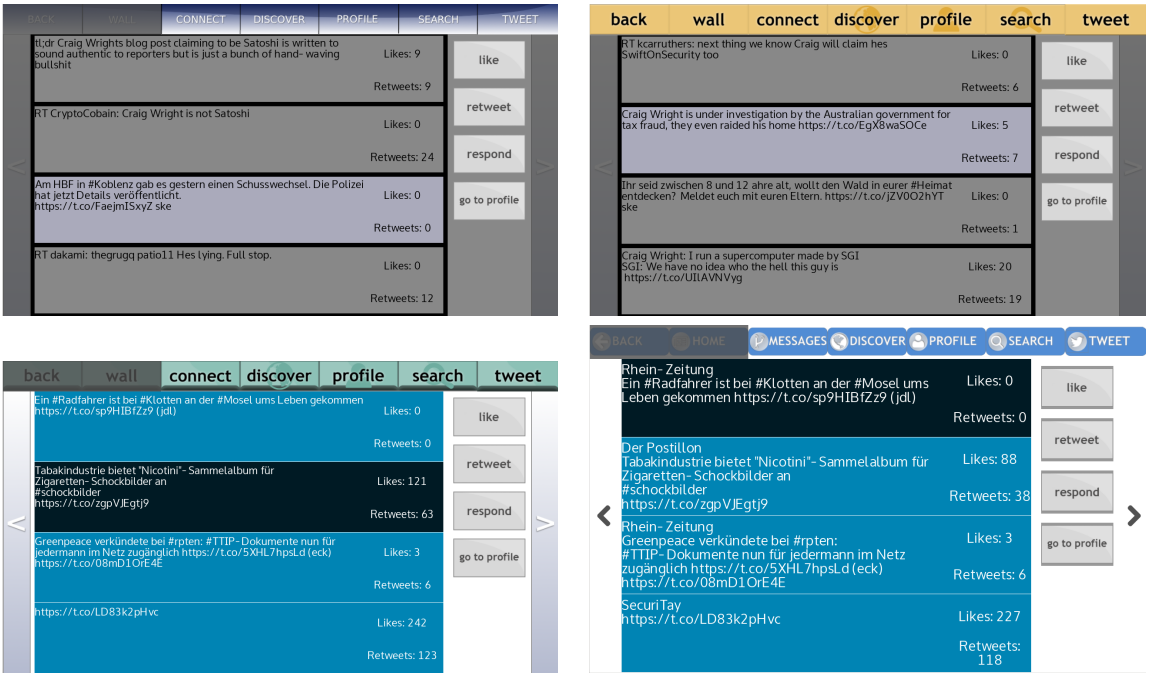

Figure 3.4: Development of the user interface of *GazeTheWeb - Tweet* over time, from oldest to newest

3.2 Implementation

The view onto the surface does not reveal the internal structures of the application. Therefore this chapter will have a deeper look at the internal structures of *GazeTheWeb - Tweet*, how it has been developed and how the code is structured.

3.2.1 Organisation

The project has been organised over GitLab[3] a web-based Git repository. Initially the team has been divided into two groups. The first one had been responsible for the setup of *eyeGUI* and for developing a first design that should be realisable using the *eyeGUI*. As part of their work many design mockups have been created, one of which you can see in chapter 3.1.1, but also basic outlines of the interface design with the eyegui. The second group had been assigned to the API related work. First, a demo application has been created with the purpose of accessing the Twitter-API using the *TwitterClient* (cf. 2.2). Working on the demo application there had already been parts of *twitcurl* that needed revision. This had become an accompanying process during the project. The core of the demo application, the *TwitterClient*-class, then covered all important functions and was also able to manage all API related connections that were necessary for testing.
After bringing the two smaller projects together, the teams had to be restructured to smaller

[3]GitLab Community Edition: `https://gitlab.uni-koblenz.de/users/sign_in`. [Accessed on 30.04.16].

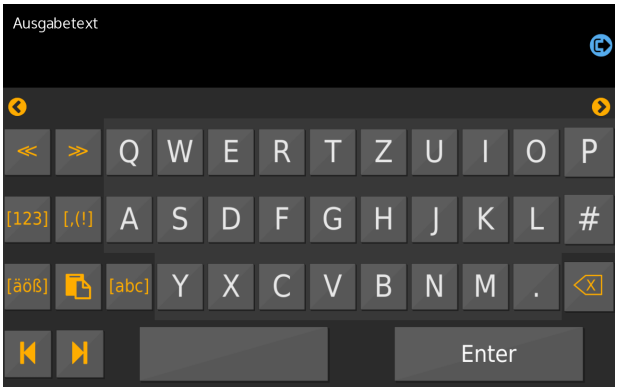

Figure 3.5: Final design of the keyboard of *GazeTheWeb - Tweet*

Figure 3.6: First design of the keyboard of *GazeTheWeb - Tweet*

teams to do different tasks. One team consisting of two members, one from the interface group and one from the API-group, merged both projects by accessing the Twitter API to fill the previously prepared designs with content and giving most of the buttons their functionality. Still a very important feature has been missing at that point: Until then the command line had to be used to enter content, so another group took care of implementing the keyboard in a separate branch on GitLab.

3.2.2 Structure of the Code

The main code documentation for the application is the doxygen documentation that accompanies this documentation. There, all classes and their connections can be found.
In general the application is managed by a class called “TwitterApplication”. It is realised as a singleton and instantiates the various interface elements. Navigation and scrolling are also done by this class. All interface elements inherit from the class “Element” and can be stacked together in an arbitrary way. ActionBar and MenuBar are always a part of the interface but inherit from “Element” as well. For the usage of the *eyeGUI* this means that layout are not built statically but dynamically based on single elements that will be fit into the pages in any order.
The ActionBar-class handles the options that get activated or deactivated dependant on the selected tweets. The menu buttons are handled by the MenuBar-class. They only become active

or inactive.
The application gets initialised by the main-class. It takes care of starting the *eyeGUI* and all OpenGL-functions. It also starts the login that always appears before the actual start of the application. The OAuth authentification is started by the login-class as well as the keyboard that is needed for entering the credentials.
Other classes that use the keyboard are "Search", "Tweet" and "Connect" (= Private Messages). Basically the keyboard can be called from everywhere in the code. It has different functionalities: A cursor to move within the text, a basic word completion that features the one hundred most used words in the english language. The dictionary can also be switched to german, french or dutch. Single words or sentences can be added to the dictionaries and removed from the same. Different layers of the keyboard are letters, capital letters, german umlauts, numbers, symbols.

3.2.3 Connection to the eye tracker

Currently *GazeTheWeb - Tweet* supports two different eye tracker: The *Tobii EyeX Controller* developed by a swedish company called *Tobii* [4], as well as the *REDn Scientific* by *SensoMotoric Instruments*, which is a german company from Berlin[5].
The producers of both eye tracker also provide software development kits for their devices which include examples how to access the input stream of the devices. Based on this examples *GazeTheWeb - Tweet* contains classes to handle the connection to the desired eye tracker. These parts of the code are only included to the build of the application if certain variables in the *CMakeLists.txt* file are set. So it is not required to install these SDKs if you want to run *GazeTheWeb - Tweet* on a machine without eye tracker connection, for example for testing purposes. In this case you are able to control the application via mouse.
To enable eye tracker support you have to set the variable `USEEYETRACKER` to `ON`. If the variable `USETOBII` is set to `ON` as well, the parts of the code are included to the build which are relevant for the connection to *Tobii EyeX Controller*. If you want to use the eye tracker of *SensoMotoric Instruments* you have to set mentioned variable to `OFF`.
If eye tracker support is enabled the required SDK files and libraries will be included to the build and relevant parts of the code, which are blocked by preprocessor macros (namely `USEEYETRACKER_TOBII` and `USEEYETRACKER_IVIEW`), will be enabled.

3.2.4 Implementation problems

Some disadvantages are due to third party technologies. Twitcurl for example does not cover all functions that Twitter provides and is outdated partly, but it was possible to modify parts of twitcurl to update functions or add new ones. Still, there has been yet another problem which prevented a full coverage of all API functions: Due to limited space and the relatively huge buttons used for eye tracking, the amount of interactive elements has been restricted. Adding more functions to the page would have lead to a breach with the design idea and would have been implemented at cost of application usability. Therefore only the most important functions have been implemented to make sure, that the usability is as good as possible for using an eyetracker.

The application also has some disadvantages concerning the perfomance. Each time some sort of content is required the application downloads the requested content, resulting disruptive laggs caused by API calls and image downloads when switching between pages which can not be

[4] Tobii: `http://www.tobii.com/`. [Accessed on 05.05.16].
[5] About SensoMotoric Instruments: `http://www.smivision.com/en/gaze-and-eye-tracking-systems/about-smi/company.html`. [Accessed on 05.05.16].

directly avoided but maybe bypassed by threading specific parts of the application. Moreover the password entering process is still difficult because the password is displayed during entering and therefore visible for potential audience. Note that this problem can't be easily fixed by cencoring the password which would usually be a appropriate solution in most cases, but in our case the entering process itself, displayed on the monitor, is slow enough for every viewer to read the password from directly the keyboard buttons. Another eyetracking fitting solution is required, one approach could be a picture bound password, where the user has to look at prior defined areas in the image to gain access.

Chapter 4

Evaluation

The evaluation aims to assess whether *GazeTheWeb - Tweet* offers a better handling with an eye tracker as comparable applications. Moreover it is found out if *GazeTheWeb - Tweet* provides modern design and functionality to the user or if the user rejects the application developed for the eye tracker. The main question is therefore why should people use *GazeTheWeb - Tweet*? Is it better to use applications that are especially designed for the eye tracker? Or is it better to use a mouse and keyboard emulator for eye tracking in combination with already existing applications? To answer these questions *GazeTheWeb - Tweet* is compared to the *Mobile twitter application with OptiKey*. The aim is to determine the usability of both *Mobile twitter application with OptiKey* and *GazeTheWeb - Tweet*. Our definition of usability corresponds to the international standard ISO DIN 9241-110 where usability is achieved through three properties: effectiveness, efficiency and satisfaction. Effectiveness means that a user can achieve certain goals completely in a particular environment. The focus of efficiency lies on the resources spent to reach those certain goals. The third component of usability is the user's satisfaction which is in general the comfort when using the system and his acceptability towards the system.

4.1 Hypothesis

The questions of the previous part form the main hypothesis of this evaluation. The main hypothesis is confirmed or dropped depending on evaluation results. To compare both applications it is important to determine their usability. To measure usability the measurements functionality, handling, design and satisfaction are used. Moreover several sub hypotheses are set up to support the main hypothesis and to serve as guidelines through the evaluation results.

Main Hypothesis
It is better to have one application which is adapted to eye tracking than using an emulator with web pages that are not adapted to it. Therefore an adapted application offers better usability.

Satisfaction

H1 *GazeTheWeb - Tweet* has less negative impact on the mood of the participants. When using the *Mobile twitter application with OptiKey* the level of frustration is higher.

Functionality

H2 *GazeTheWeb - Tweet* offers the main twitter functions.

H3 The system (*GazeTheWeb - Tweet*) is consistent.

Handling

H4 The application *GazeTheWeb - Tweet* is intuitively usable.

H5 People with computer skills need minimal help to use *GazeTheWeb - Tweet*.

Design

H6 *GazeTheWeb - Tweet* offers a design adapted to the eye tracker.

H7 The user experiences the design of *GazeTheWeb - Tweet* in general as adequate and pleasant.

H8 *GazeTheWeb - Tweet* fulfils the requirements of self-descriptiveness and the conformity with user expectations as described in ISO 9241-110.

4.2 Experimental set-up

This section describes the set-up of the test case, the evaluation method and the choice of the participants. To proceed the evaluation the discount usability test is chosen as test method. In comparison to simply sending questionnaires to the target group the usability test finds out about real user behaviour. The users which are representatives of the target group have to perform representative tasks with the system while being observed by supervisors. This enables the supervisors to find out about problems and positive aspects of the system. A supervisor of the usability test represents a key role in the process. He has to be an expert in testing and he has to be familiar with the tested software. A usability test can be performed several times during the software development. One can measure usability of prototypes, fully developed products and of all steps in between.
Additionally the subjective perception of a user can be obtained using questionnaires. Comparing this aspect to the real user behaviour leads to observations about the user's self-perception and reality that can be surprising. Based on the idea of discount usability testing by Jacob Nielsen the evaluation study of *GazeTheWeb - Tweet*is designed on a low budget basis. Instead of videotaping the participants are invited to use the "thinking aloud" method. While doing the given tasks the participants have to say out loud what they are thinking. This method delivers insight into the thoughts of the participants and replaces an expensive acquisition of equipment. After conducting the usability test the supervisors discuss their observations. Additionally key data can be measured like duration of executing a task. They analyse positive aspects of the software and often made mistakes of the participants. The collected results are summarized in a report where user quotations or user reactions support important facts. During this evaluation method there is no use of cameras or expensive software to record the computer screen or cursor movements.[Nie] [too]

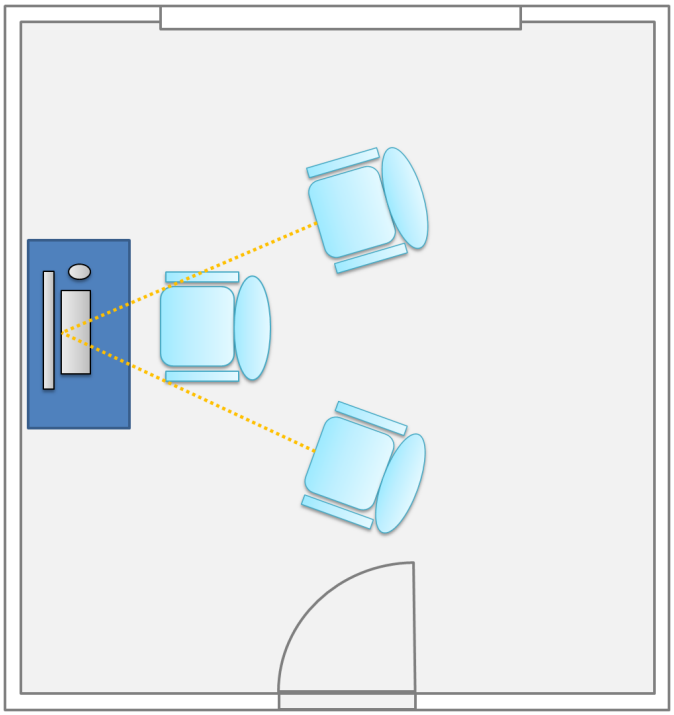

Figure 4.1: Room set-up to perform the evaluation method.

Instead only thirteen participants are tested in a simplified environment. Moreover two supervisors are present, where one introduces the project and leads the participant through the tasks. The other one observes the participant's reactions and measures the needed time per task. Figure 4.1 shows the set-up of the evaluation where the participant sits in front of the computer screen with the eye tracker while the supervisors are sitting behind him.

Before testing *GazeTheWeb - Tweet* and *Mobile twitter application with OptiKey* have to provide the same properties to be comparable to each other thus adjustments have to be made. Table 4.2 gives an overview of the adapted properties.

value	default	current
Dock thickness	50	90
Speech Volume	100	0
Sound in general	enabled	disabled
Autospace between words	enabled	disabled
Auto capitalize words	enabled	disabled
Multikey captures	enabled	disabled
Maximum number of dictionary matches	20	1000
Source	Mouse	Tobii
Selection Progress Indicator	Pie	Shrink
Key fixation time to lock-on (ms)	250	1
Key fixation time to complete (ms)	?	1000
Point fixation time to complete (ms)	1250	1000
Selection Progress indicator end size	20%	30%

Figure 4.2: Adjustment of OptiKey properties to be comparable to *GazeTheWeb - Tweet*.

The chosen participants are people without handicaps even though *GazeTheWeb - Tweet* aims to enable handicapped people the usage of twitter. The application was tested at university so unfortunately there was no access to this target group. The tested group consists of students and staff of the university aged between twenty and thirty-nine. Most of them are male which is shown in figure 4.3. In the evaluation the focus lies on twitter users because they already own basic knowledge about the main features in twitter. They can give detailed information on missing features and suggestions to improve the implemented features in *GazeTheWeb - Tweet*.

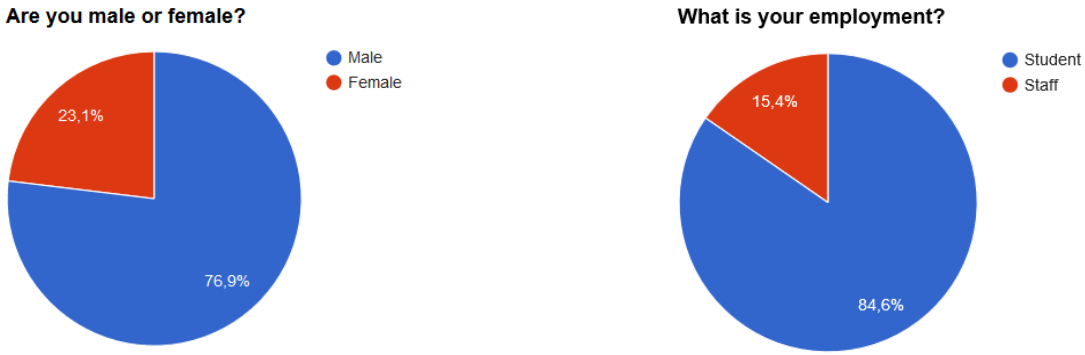

Figure 4.3: The participants' personal information

4.3 Experimental procedure

This section gives a comprehensive overview of the evaluation procedure. After introducing the project and calibration of the eye tracker the participant gets used to the eye tracker by a given

short tutorial. Afterwards the participant gives information about personal data and his usage behaviour of twitter. To draw conclusions of the participant's mood some participants start doing tasks with *GazeTheWeb - Tweet* and the others start with the *Mobile twitter application with OptiKey*. Figure 4.4 shows different tasks the participants have to complete with both applications. The tasks pose challenges to the participants in using the keyboard, searching for profiles and using the follow feature and the like feature. Immediately after doing the tasks the mental workload of the participants is measured by the NASA-TLX which is a standardized questionnaire. A description of the NASA-TLX and results can be found in 4.5.2.

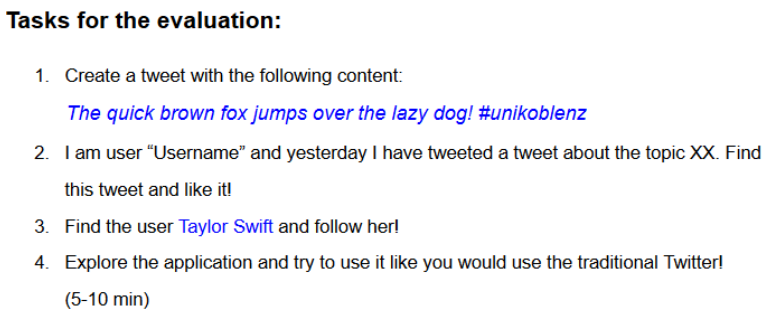

Figure 4.4: Tasks the participants have to do during the evaluation.

Then the participant has to fill out a questionnaire regarding the tested application. This questionnaire contains questions to functionality, handling and design of the corresponding application. Afterwards the participant takes the system usability scale (SUS) questionnaire to measure the usability of application. The SUS is described in 4.5.1. Then the whole procedure is repeated with the not yet tested application.

4.4 Observation

During the experimental procedure the participants show different behaviour patterns which are observed by the supervisors. It is noticeable that some participants get physical complaints during the usage like neck pain and dry eyes. Additionally it is hard for them to stay in the same position when using the eye tracker. Therefore the eye tracker has to be recalibrated several times. It is not clear if these observations of physical problems can be applied to disabled people as well. Since this is a project in the sense of MAMEM, therefore another testing with disabled people is required.
Also the participants feel stressed if the user interface reacts constantly to the user actions. If the participant finds a place in the application to rest his eyes a stress reduction can be observed. Furthermore the joy of a participant increases when he has the chance to personalize the application's functions for example the typing speed of the keyboard. Also the participants are really concentrated during the testing and overlook helpful functions like auto complete.

4.5 Results

This section present the results of the experimental procedure and shows the results of the different questionnaires and test methods like SUS and TLX.

4.5.1 System usability scale

The System Usability scale (SUS) is a standardized questionnaire which is used to measure products and services. It is used because it is a very easy scale to understand for participants and it only includes a small amount of questions so it does not take a lot of time. Also its result states reliably if the application is usable or not. That's why it has become a standard in industry and is referenced in over 1300 articles and publications. In this evaluation each participant has to do the SUS questionnaire twice to evaluate *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey*. The SUS contains 10 questions with five responding options from "strongly disagree" to "strongly agree" which is shown in figure 4.5. The result of SUS is a score which indicates the usability of a product. The score is a number between 0-100 where 68 is considered average. If the product gets a score higher than 80.3 the users are more likely to recommend the product to a friend. The score of SUS is calculated as follows: Each question contributes a value from 0 to 4 to the score. The scale position correlates to the answer of the participant. For questions 1,3,5,7 and 9 the scale position is reduced by 1. For the other questions the value 5 is reduced by the scale position. The results of each question are added together and are multiplied by 2,5 to get the final result.[sus] [Bro]

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

Figure 4.5: SUS questionnaire.

To validate that the difference of the outcomes of both systems was statistically significant a paired student's t-test was performed on the results of the SUS questionnaire which you can see in figure 4.6. A *paired* t-test is used when both datasets are collected in the same way, but the first one was collected before a certain treatment and the second one afterwards [Zha06, page 62]. This applies to the described experimental setup because both datasets were derived from the same questionnaire and in between a testing run was performed.
The paired t-test resulted in a P-value of `0.0044`. The mean of group one minus group two equaled 22.115 and the 95% confidence interval of this difference ranged from 8.340 to 35.891. The intermediate values used in calculations were `t=3.4980`, `df=12` and standard error of difference of `6.322`. With this values the difference can be considered as statistically significant [Zha06].

GazeTheWeb - Tweet achieves a score of almost 72 which makes it close to the average value while the *Mobile twitter application with OptiKey* only achieves a result of almost 50. SUS shows on the one hand that the usability of *GazeTheWeb - Tweet* is better than the one of the *Mobile twitter application with OptiKey* which supports the main hypothesis. On the other hand it states clearly that *GazeTheWeb - Tweet* still needs improvements to achieve a good usability. Figure 4.6 shows the evaluation for *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey*. F1 to F10 describe the ten questions of SUS and P1 to P13 describe the participants. The orange numbers refer to the results of each participant.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Ges	*2,5
P1	0	1	2	4	3	4	3	3	3	4	27	67,5
P2	0	4	4	4	3	4	3	3	3	4	32	80
P3	0	4	3	4	4	4	4	3	3	4	33	82,5
P4	3	3	3	3	3	4	0	2	3	3	27	67,5
P5	3	4	3	4	4	4	4	3	2	4	35	87,5
P6	0	2	1	4	4	4	3	0	3	3	24	60
P7	3	3	4	4	4	3	4	3	3	4	35	87,5
P8	1	3	2	4	3	3	3	2	2	4	27	67,5
P9	0	3	2	4	3	4	2	2	3	2	25	62,5
P10	1	1	2	3	1	3	2	2	2	1	18	45
P11	0	4	3	4	3	3	3	1	1	3	25	62,5
P12	0	4	4	4	4	4	3	3	3	4	33	82,5
P13	0	4	4	4	4	4	3	3	3	4	33	82,5

SUS average result GazeTheWeb-Tweet: 71,923

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Ges	*2,5
P1	0	1	1	4	3	2	1	0	3	4	19	47,5
P2	0	0	0	4	0	0	3	0	1	4	12	30
P3	0	3	2	3	3	2	3	1	0	3	20	50
P4	0	1	1	1	1	4	0	0	1	1	10	25
P5	0	0	0	4	3	4	3	0	0	2	16	40
P6	0	1	1	4	2	3	2	0	2	2	17	42,5
P7	0	1	2	4	2	2	1	3	2	3	20	50
P8	1	2	2	4	3	3	4	2	2	3	26	65
P9	0	3	2	4	3	4	2	1	2	2	23	57,5
P10	2	3	4	4	2	4	1	3	3	1	27	67,5
P11	0	3	3	4	3	3	3	3	3	4	29	72,5
P12	0	0	1	4	3	4	2	0	1	2	17	42,5
P13	0	3	3	3	2	1	2	3	2	4	23	57,5

SUS average result Optikey: 49,808

Figure 4.6: Evaluation of SUS

4.5.2 NASA task load index

The NASA Task Load Index (NASA TLX) contains six components: mental, physical, temporal demands, frustration, effort and performance.

The mental demand is an assessment of the effort when acquiring and processing information like thinking, deciding, calculating, remembering, looking, searching etc. The physical demand describes the amount of physical activity that is required to perform a task like pushing, pulling, turning, activating etc. Temporal demand means the time pressure that a user feels when task elements occur. It refers to the hecticness of a task. Effort means how hard the user has to work

to reach a certain aim. Performance describes how successful a user reaches the aim of each task. It also states how satisfied the user feels with his performance. The frustration consists of how insecure, irritated, stressed and angry the user feels during the task.

It is assumed that all components together represent the mental workload of a human. The mental workload is the cost a human has to spend to accomplish a task. When developing new products it is important to know the mental workload of the users to make the products usable and not to challenging for them. Each component is assessed by the participant once for *GazeTheWeb - Tweet* and once for *Mobile twitter application with OptiKey*. On a scale from 1 (low) to 100 (high) in 5-point steps for each component the participant can decide the most applicable to his mood. The following figure 4.7 shows the first part of the NASA TLX.

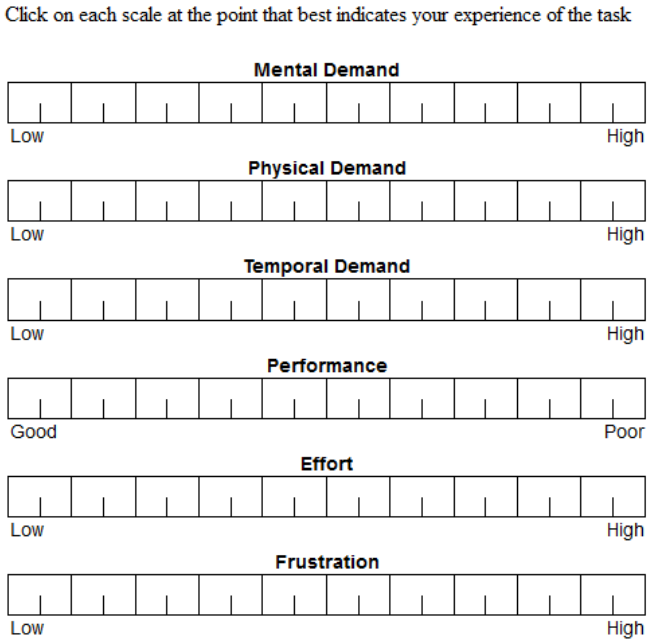

Figure 4.7: Task load index questionnaire.

Afterwards the participant has to put weights on each scale by comparing all components pairwise. The benefit of weighting is to increase the meaning to more relevant components. The participant has to decide then which component has a bigger influence on the result than another. The result is a number between 0 and 100. The higher the resulting number the higher is the mental workload for the participant. As you can see in table 4.8 *Mobile twitter application with OptiKey* (value = 63,8) causes a higher mental workload for the participants than *GazeTheWeb - Tweet* (value= 46,7).[Ver] [Gro]

	GazeTheWeb-Tweet	Optikey
P1	48,3	75
P2	19,7	52,3
P3	17,3	64,7
P4	43	79,3
P5	76,7	92
P6	52,3	50,9
P7	43,7	80,3
P8	47	66
P9	59,3	36,6
P10	62,3	43,6
P11	61	45,7
P12	52,3	88,6
P13	24,3	54,3
Summe	607,2	829,3
D.-Schnitt	46,70769231	63,79230769

Figure 4.8: The results of task load index questionnaire

As described in chapter 4.5.1 we also performed a paired t-test on the results of the task load index questionnaire which you can see in figure 4.8.
The paired t-test resulted in a P-value of `0.0238`. The mean of group one minus group two equaled -17.085, the 95% confidence interval of this difference ranged from `-31.469` to `-2.701`. The intermediate values used in calculations were `t=2.5879`, `df=12` and standard error of difference of `6.602`. With this values the difference can be considered as statistically significant [Zha06].

Figures 4.9 and 4.10 provide a more detailed view on the results of the NASA TLX. As you can see, in average the mental demand and the effort were weighted the strongest by the participants, which also corresponds to the average score values. In Figure 4.10 the values of this aspects are almost the same for both systems, which indicates that controlling a system with an eye tracker is a mentally demanding and exhausting task in general. Additionally Figure 4.9 shows that in average *Mobile twitter application with OptiKey* was perceived even more mentally demanding and exhaustive than *GazeTheWeb - Tweet*. A bit surprising is the low weighting of the physical demand, because a lot of participants complained about physical troubles during the tests, like a pain in the neck or dry eyes. But obviously they perceived the mental demand as even stronger, which underlines the importance of this aspect.

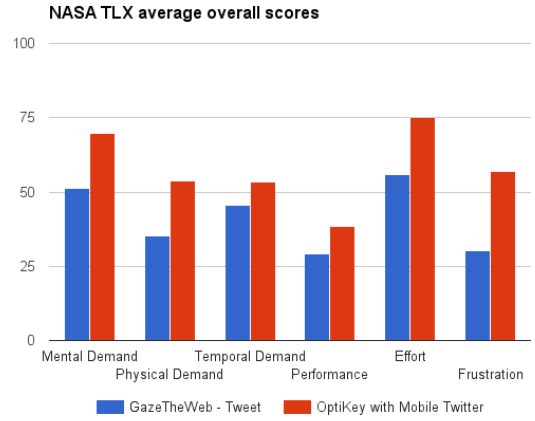

Figure 4.9: NASA TLX average scores

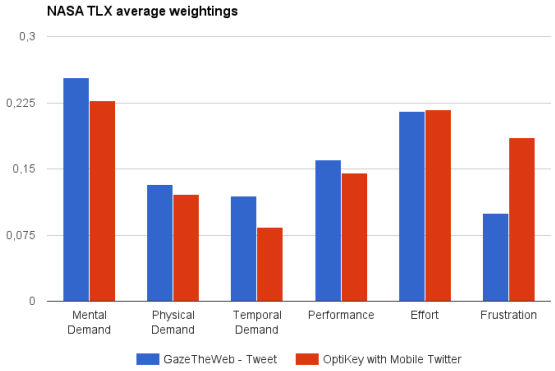

Figure 4.10: NASA TLX average weightings

A big difference between the systems can be observed at the point frustration. As you can see this aspect is weighted and scored a lot higher for *Mobile twitter application with OptiKey*. This result fully supports hypothesis one, which says: "*GazeTheWeb - Tweet* has less negative impact on the mood of the participant. When using the *Mobile twitter application with OptiKey* the level of frustration is higher". The different average scores of the aspect of mental demand and effort support this hypothesis as well.

4.5.3 Further results

The first measurement that should be looked at is the functionality of *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey*. So the participants were asked about their most used functions at twitter to compare this information to the implemented features in *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey*. Are all main features present or is something missing? As can be seen in figure 4.11 the most frequently used features are reading tweets, liking and retweeting.

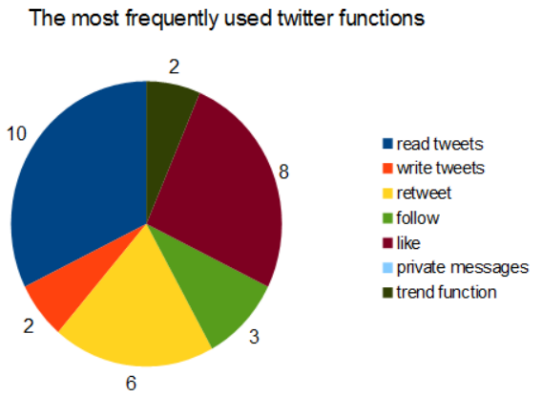

Figure 4.11: The participants' twitter usage

Figure 4.12 shows the detailed view and reveals that private messages and the trend functions are the least used features.

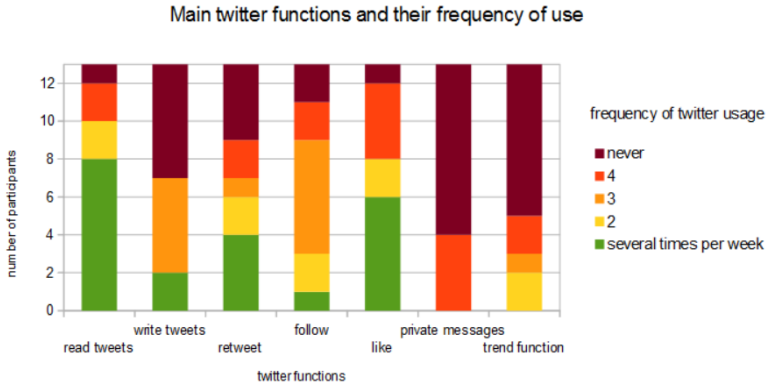

Figure 4.12: The participants' twitter usage

The most frequently used features will be allocated by *GazeTheWeb - Tweet* to the user. But are these functions fully usable and satisfactory implemented? To find it out the participants are asked whether *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey* satisfies them in this point. Here it is particularly noticeable that participants which often use twitter assess that *GazeTheWeb - Tweet* does not provide all needed functions. Participants which rarely used twitter and its features are more satisfied. So the evaluation of this point is dependent upon the use frequency and therefore on the individual twitter skills. Figure 4.13 illustrates this fact.

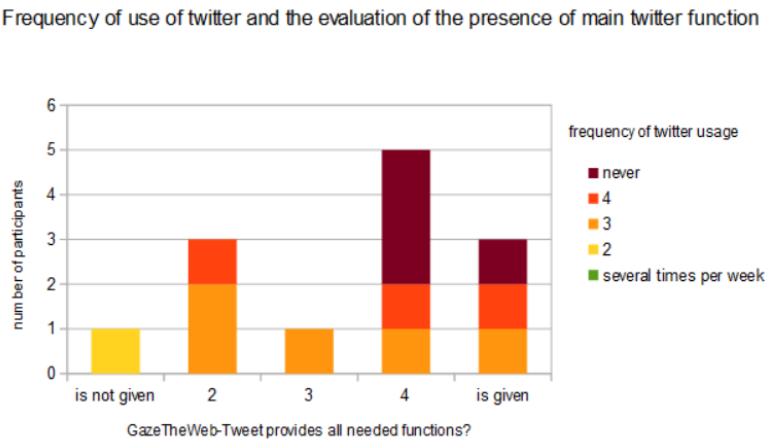

Figure 4.13: Frequency of use of twitter and the evaluation of the presence of main twitter functions.

To return to the question if *GazeTheWeb - Tweet* provides all main functions, a look is taken at the participants' answers which are presented in figure 4.14. In addition the answers to the question regarding the *Mobile twitter application with OptiKey* are also shown in this figure to compare both simultaneously.

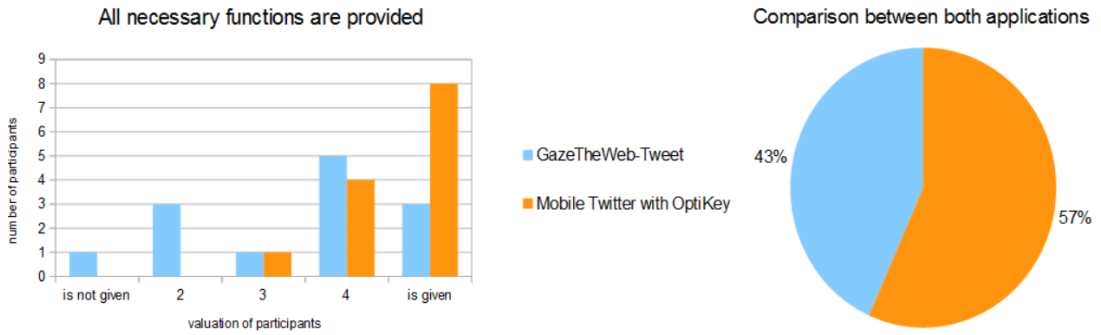

Figure 4.14: Answers if all main functions are given by the two applications.

Here it is noticeable that *GazeTheWeb - Tweet* displays some deficits in this fact whereas the *Mobile twitter application with OptiKey* provides more functions according to the participants. So hypothesis two which says that all functions are provided seems to be not fully confirmed. The main functions are given but not completely implemented. For example a user is able to follow someone but can not display his own followers. The following list shows some missing twitter features at *GazeTheWeb - Tweet* mentioned by the participants.

- List which shows own followers.
- List with people which one follows.

- Linking hash tags and "@username" in tweets.
- Navigation to external links.

Moreover the participants are asked if the available functions are well integrated in both applications. The interaction between functions and the whole system is an important aspect regarding the user. If functions are well integrated the user is able to understand the functionality of the whole system easier and the usability will increase. In this context *GazeTheWeb - Tweet* is rated on average with 4,3 and the *Mobile twitter application with OptiKey* with 3,3. Figure 4.15 shows that it is obvious that *GazeTheWeb - Tweet* convince the user in a better way than the *Mobile twitter application with OptiKey*.

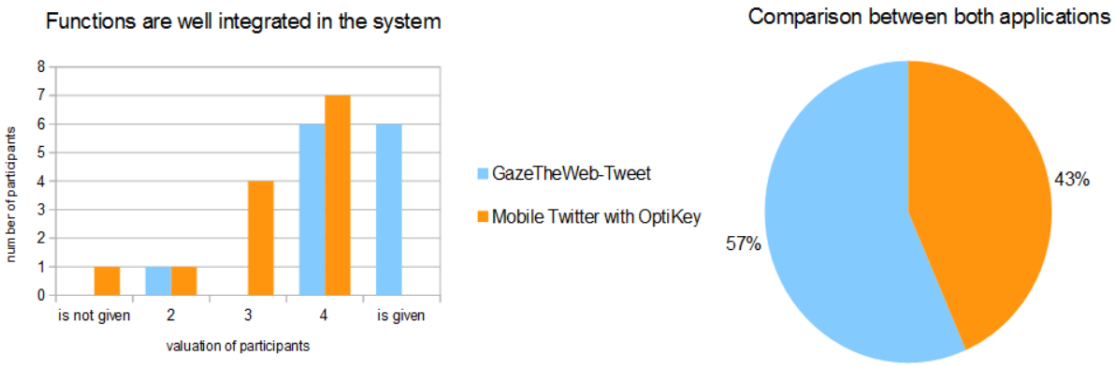

Figure 4.15: Answers if functions are well integrated in the system.

In the previous part it is established that the *Mobile twitter application with OptiKey* offers all functions however the participants assess them as "difficult to find" and "difficult to use". Therefore it should be found out if these functions are even conform with the user expectations. Is the functionality of each feature completely transparent to the user or does it arise misunderstandings during usage? Moreover it should be found out if the functions react correctly and consistent.

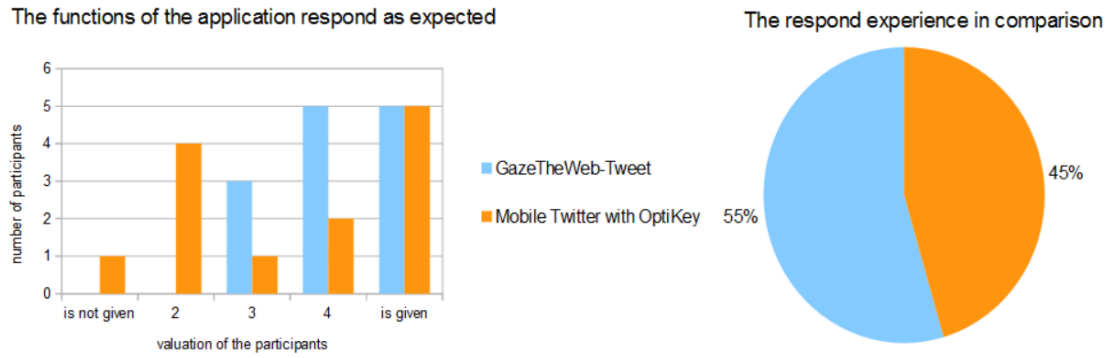

Figure 4.16: Answers if all features respond as the user expected.

Here *GazeTheWeb - Tweet* is slightly better than *Mobile twitter application with OptiKey*. Figure 4.16 shows the detailed valuation of the participants. *GazeTheWeb - Tweet* is rated on

average with 4,15 which is close to "is given". In contrast *Mobile twitter application with OptiKey* is only assessed on average with 3,46. Furthermore the participants named some certain functions from *GazeTheWeb - Tweet* which don't react as expected.
So if the user looks at the inserted text in the search section to consider its accuracy the keyboard appears without any feedback. Another point is a missing back button to make switching between pages more comfortable. Also the "world button", the "speech bubble button" and the "exit button" confuse the user. A possible explanation is that the world symbol is often used as internet symbol. But in *GazeTheWeb - Tweet* it shows trends and popular pages. The button represented by the speech bubble is also named as an unexpected function. The user expected to tweet but it symbolized the direct messages function. Also the participants mentioned that some functions contain errors which are listed in the following.

- Tweets of large size are not displayed completely.
- The display of the like number is not updated immediately.
- Some pictures are not displayed correctly.
- Special characters are displayed in HTML format after tweeting.

Regarding the *Mobile twitter application with OptiKey* the participants named some more things. Even one participant mentioned that all functions do not react as expected. Others stated problems with scrolling and the magnifying glass. At this point hypothesis eight can be seen as confirmed because *GazeTheWeb - Tweet* fulfils the requirements of conformity with user expectations almost completely.

Next it will be take a look at the consistency. Consistency is an important property of developed applications. Function should react always in the same way to support the usability and do not confuse the user.

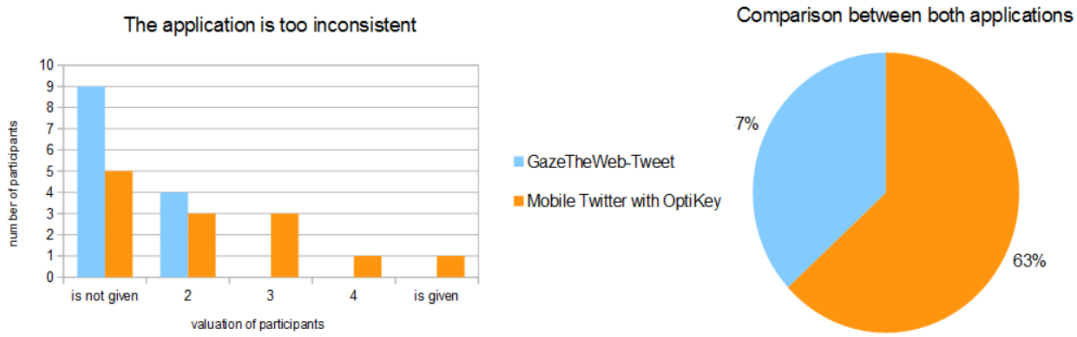

Figure 4.17: Answers if the participants experience the application as inconsistent.

The participants are asked how they experience the consistency of *GazeTheWeb - Tweet* and the *Mobile twitter application with OptiKey*.Figure 4.17 present the valuation of inconsistency. *GazeTheWeb - Tweet* is rated on average with 1,3 whereas the *Mobile twitter application with OptiKey*is rated with 2,23. This result shows that *GazeTheWeb - Tweet* achieves a good evaluation in this case. Therefore hypothesis three can be seen as confirmed. *GazeTheWeb - Tweet*has a good consistency.

Another important point regarding the functionality is the feedback given by the application. However the participants named some issues regarding missing feedback. This fact is presented in figure 4.18.

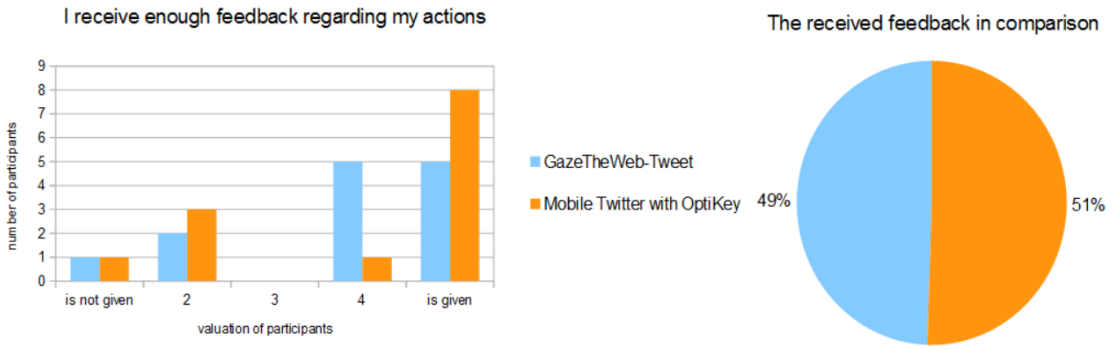

Figure 4.18: Answers if the user receives enough feedback by the user interface.

Here it should be noted that both applications need to be improved in their feedback behaviour whereas the *Mobile twitter application with OptiKey* is slightly better assessed. On average *GazeTheWeb - Tweet* is rated 3,84 at the scale from one to five and the *Mobile twitter application with OptiKey* is rated 3,92. So participants mentioned missing feedback at *GazeTheWeb - Tweet* if they look at the inserted text in the search section to consider its accuracy because the keyboard appear to be without any feedback.
This sections aims to find out if the developed application offers an attractive design which is modern and harmonious and keeps up with the compared system. Design is an important issue for applications. Features can be highlighted to the user, the design can support the handling and is important to satisfy the user during usage. If applications offer a harmonious design users perceive the application as more attractive which leads to a better usability. The participants are asked about their point of view regarding the design experience.

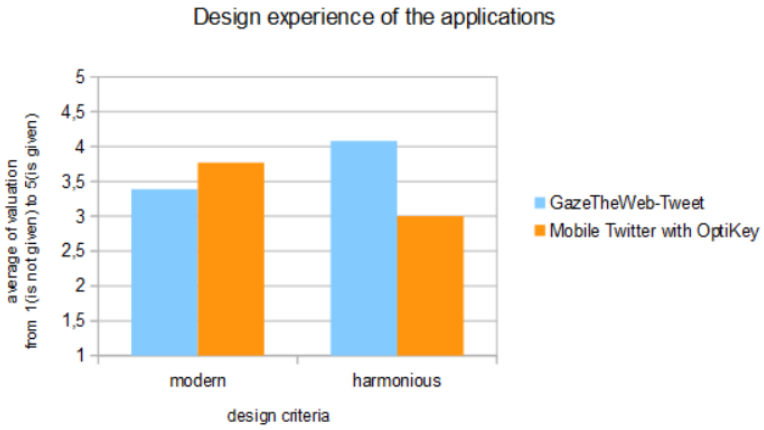

Figure 4.19: Design experience of the users.

Figure 4.19 shows the valuation of the participants. On average they assess the *Mobile twitter application with OptiKey* as more modern than *GazeTheWeb - Tweet* whereas *GazeTheWeb - Tweet* is rated as more harmonious than the *Mobile twitter application with OptiKey* which is due to the fact that *GazeTheWeb - Tweet* is a whole application developed for the eye tracking and not a joined system. Therefore this valuation is expected. Furthermore the participants are

asked if the design is pleasing to their eyes. In this point the participants are more persuaded by *GazeTheWeb - Tweet* than the *Mobile twitter application with OptiKey*. Figure 4.20 represents the evaluation of the participants regarding this topic. The comfort to the eye is rated with an average of 4,53 at the scale from one to five by working with *GazeTheWeb - Tweet*. Whereas this point is evaluated with 3,38 by working with the *Mobile twitter application with OptiKey*. This important fact supports the assumption that a complete developed system adapted to the eye tracker is more user friendly. If applications are pleasing and adequate the desire to a further usage will increase. Therefore the following deals with additional design aspects.

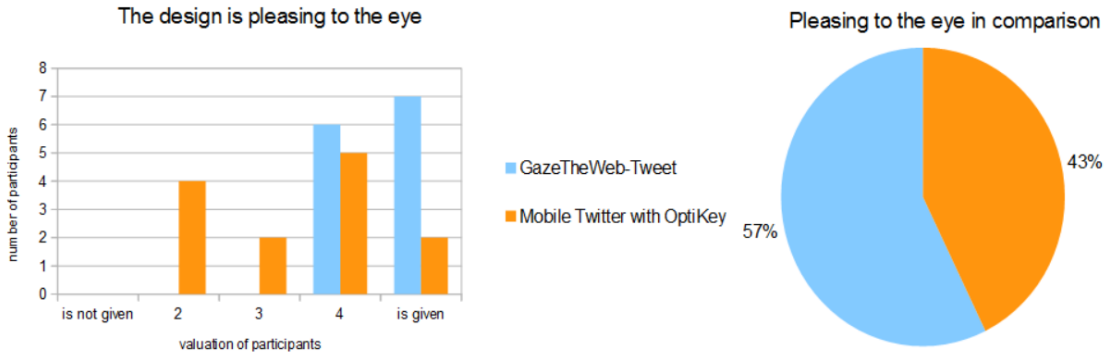

Figure 4.20: Answers if the design of the user interface is pleasing to the eye.

The scripture is a significant design component. In combination with the eye tracker it should be presented in big letters and clear font. Ornate fonts can be too much because of the high feedback rate of the application. So it should be kept simple. Figure 4.21 represents the answers of the participants on how they experience the scripture of *GazeTheWeb - Tweet*. On average *GazeTheWeb - Tweet* is rated with 4,3 at the scale from one to five and the *Mobile twitter application with OptiKey* is rated with 3,92. Therefore *GazeTheWeb - Tweet* is evaluated as very good. The second diagram represents the summarized evaluation in percent. It is noticeable that *GazeTheWeb - Tweet* is slightly better in this aspect.

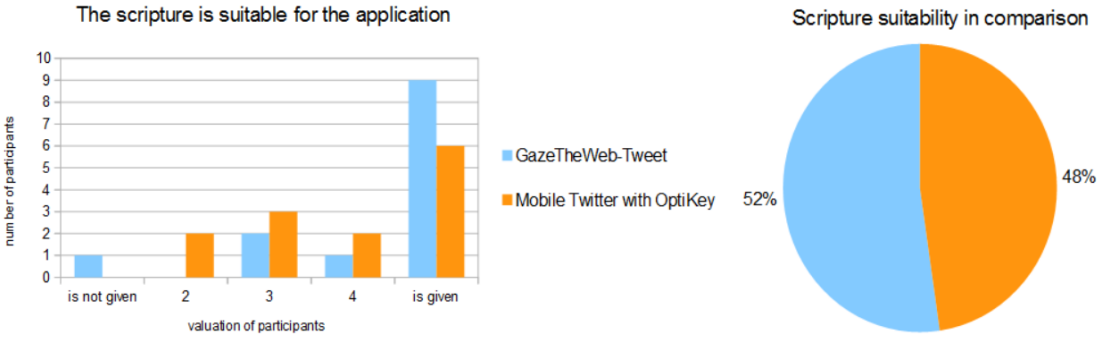

Figure 4.21: Answers if the user experience the scripture as suitable.

Moreover the participants are asked about the buttons size. Are they large enough or have the participants problems with the size? A lot of attention was given to the navigation buttons, the action buttons and the keypad buttons. On average the participants rated the size of the

navigation buttons in *GazeTheWeb - Tweet* with 4,92 which is nearly the optimum whereas the navigation buttons in the *Mobile twitter application with OptiKey* are rated with 2,69. The evaluation of the actions buttons delivers similar results. At this point the button size in *GazeTheWeb - Tweet* is evaluated with 4,84 and in the *Mobile twitter application with OptiKey* with only 2,09. This is due to the fact that the *Mobile twitter application with OptiKey* does not provide a button size adapted to the eye tracker. In the joined system a magnifier is necessary. Only the keypad buttons are similar evaluated by the participants. In *GazeTheWeb - Tweet* they are rated on average with 4,53 and in the *Mobile twitter application with OptiKey* with 4,46. At this point *Mobile twitter application with OptiKey* supports a suitable keypad button size regarding the eye tracker. Summarized hypothesis six can be seen as confirmed because of the high evaluation of the participants. Therefore *GazeTheWeb - Tweet* offers a design adapted to the eye tracker. Also hypothesis seven is fulfilled partially. Deficits can be noted in modernity. Here *GazeTheWeb - Tweet* is rated on average with 3,39.

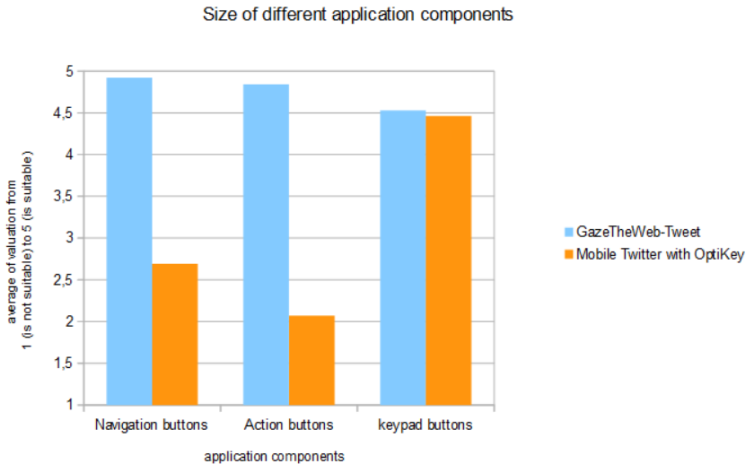

Figure 4.22: Valuation of different button size.

In the previous part only the design has been considered without any conclusions regarding the handling. Arrangement of design components is an important fact in this issue. In consideration of design principles functions can be prioritized to the user. Furthermore this serves to better orientate the user in the application. Now the question is if the developed application supports this property. The participants are asked if both applications support the handling regarding the orientation. On average *GazeTheWeb - Tweet* is rated with 4,15 whereas the *Mobile twitter application with OptiKey* is rated with 3,07. This shows that *GazeTheWeb - Tweet* supports the orientation with the design elements and their arrangements in a better way than the *Mobile twitter application with OptiKey.*

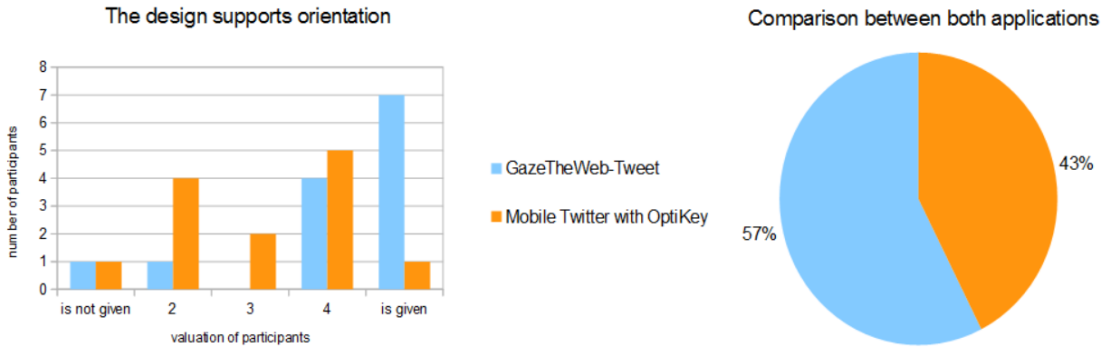

Figure 4.23: Answers if the design supports the orientation of the user.

This leads to a better handling during usage and therefore to an intuitive usage. A further aspect regarding intuitive usage are control elements like scrolling. It is important to look at the handling of the scroll function and if it is easily identifiable by the user. The participants are asked about these two properties. In *GazeTheWeb - Tweet* the scroll function is rated on an average with 3,69 regarding the handling. Some user experience difficulties during usage and need a short acclimatisation period to handle scrolling. That could be due to the fact that the scrolling function is not obvious to the user. But just after a short period of introduction they are able to use the function intuitively. The participants evaluated this function in the aspect of easy identification on average with 3,96. At this point the *Mobile twitter application with OptiKey* is still slightly better with an average of 4. But if the participants find this function in the *Mobile twitter application with OptiKey* they experience more difficulties with the intuitively usage.

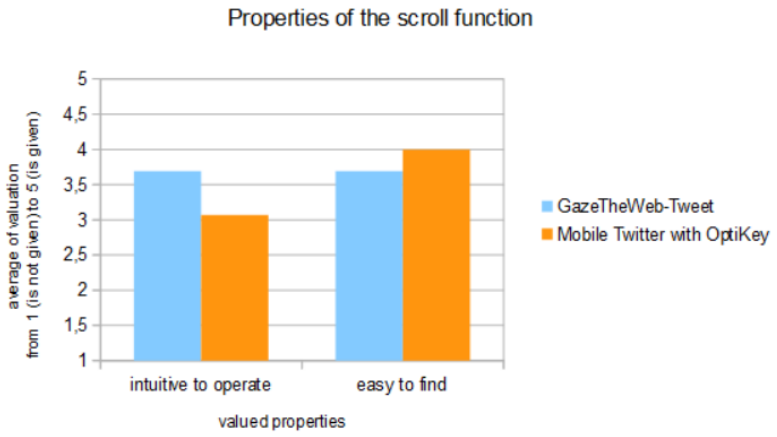

Figure 4.24: Answers if the scroll function is easy to find and intuitive to operate.

The intuitive handling depends also on the self-descriptiveness and the conformity of the user interface which is mentioned in the previous part. It is found out that hypothesis eight is confirmed almost completely because of the high evaluation by the participants. Still the application need improvements regarding feedback and some design components. Furthermore now it is noticeable that a main control element has deficits in identification by the user and

in the intuitive handling. Therefore hypothesis four can not be fully confirmed because users are not able to use some functions of *GazeTheWeb - Tweet* intuitively on the first usage. On the other hand it should be mentioned that *GazeTheWeb - Tweet* is still better than the *Mobile twitter application with OptiKey* by its clear structured user interface. So participants can handle *GazeTheWeb - Tweet* after a short familiarization period intuitively. Moreover the participants mentioned that *GazeTheWeb - Tweet* offers a clear and unambiguous design in comparison to the joined system.

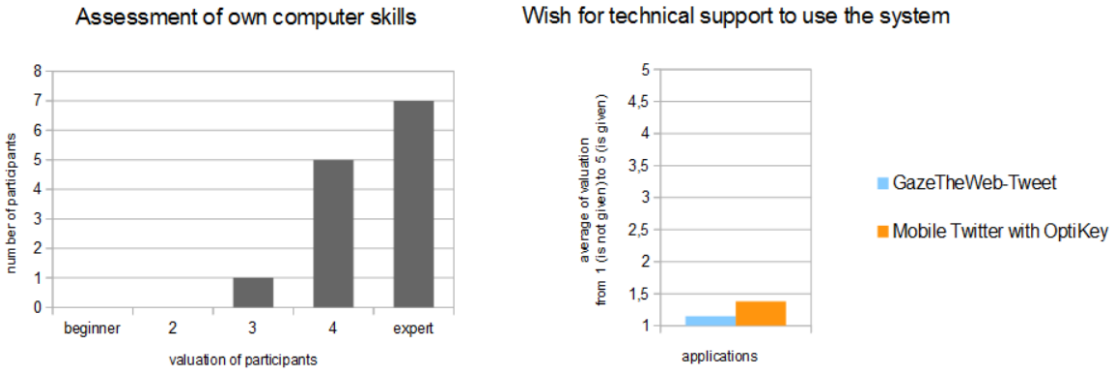

Figure 4.25: Answers how participants assess their own computer skills.

To work with *GazeTheWeb - Tweet* or the *Mobile twitter application with OptiKey* can also be difficult for users with less computer skills. Taking a look at this fact the participants are asked about their own computer skills and if they need help in usage of the applications. The valuation is shown by figure 4.25. It becomes clear that the participants mainly assess their own computer skills as very good and thus their need of help as minimal. Therefore hypothesis five is confirmed. People with good computer skills need minimal help to use *GazeTheWeb - Tweet*. Because of participants with exclusivity good computer skills it can not be said the same for people with less computer skills. But here it is also noticeable that *GazeTheWeb - Tweet* is easier in usage than the *Mobile twitter application with OptiKey* which is also shown in figure 4.26. On average the participants rated *GazeTheWeb - Tweet* with 3,84 and the *Mobile twitter application with OptiKey* with only 2,69.

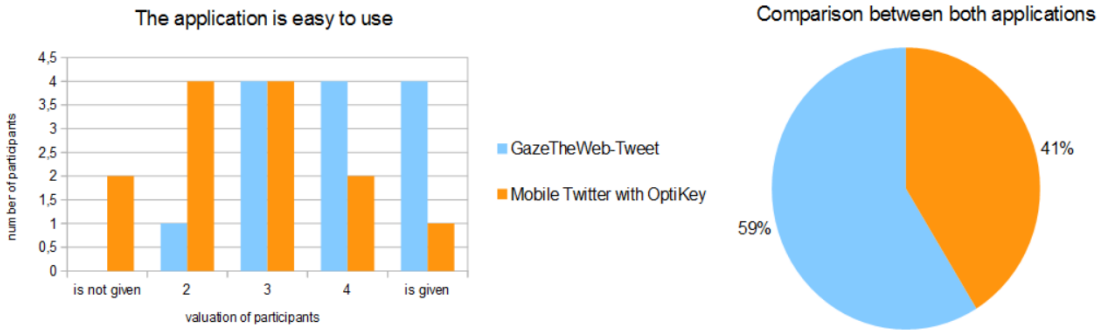

Figure 4.26: Answers if the application is easy to use.

This result is also reflected in the observations during the experimental procedure. While

testing the *Mobile twitter application with OptiKey* the participants have asked more questions and have shown more confusion. This could be due to the fact that the interaction between the emulator OptiKey and the web page is not clear enough.

Another case in this matter is the preparation before usage. *GazeTheWeb - Tweet* is kept simple and users should not need a lot of preparation before using this application. Figure 4.27 shows the evaluation of the participants. *GazeTheWeb - Tweet* is rated on average with 1,61 and the *Mobile twitter application with OptiKey* with 2,3. For this issue *GazeTheWeb - Tweet* requires less preparation than the *Mobile twitter application with OptiKey* before usage.

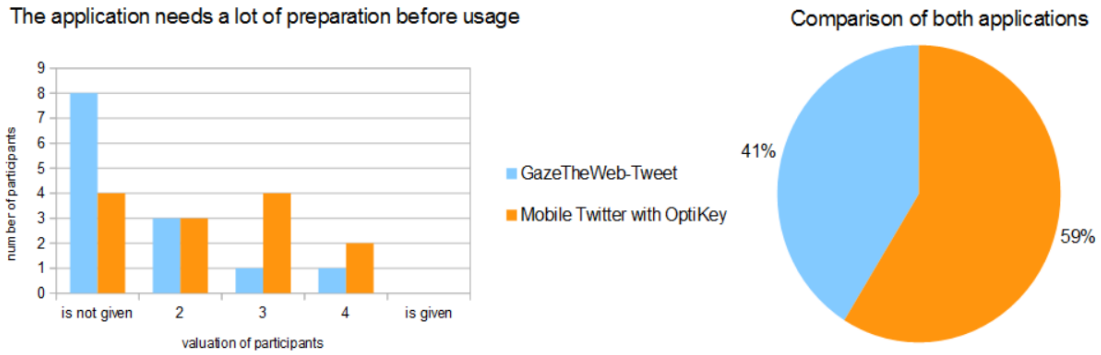

Figure 4.27: Answers if participants needed some preparation before usage.

Chapter 5

Conclusion

This chapter covers the improvement part to get a look where *GazeTheWeb - Tweet* still has problems and how these could be solved. Moreover it shows a section for future development where the new findings during evaluation are presented which is relevant for future eye tracking development. The last and final section sums up all results of the evaluation part regarding the main hypothesis if *GazeTheWeb - Tweet* is better than using an emulator with web pages that is not adapted to it.

5.1 Improvements

During the evaluation some difficulties appeared. These problems and possible improvements will be discussed in the following part. First there are several buttons which are experienced as confusing by the users. User experience is an important fact at this point. Every day people are confronted with specific symbols by public media and form associations with them. The first unclear button mentioned by the participants is the trend button represented by a world symbol. It shows trends and pages to a specific topic. This could be due to the fact that the world icon often symbolizes the internet or a network. The issue could be solved by replacing it with another symbol like a statistic image or by displaying the word "trends" when the user focus the button. Another button which is misunderstood during the experimental procedure is the direct messages function covered by a speech bubble. The participants assumed that this button offers the tweet function. This could be a result of the button arrangement and also the choice of the speech bubble icon. The speech bubble does not symbolized the necessary privacy. So the user associates its functionality with a public communication and therefore with tweeting. A letter as button symbol could be more suitable because it is well known as a private message which is only available to the recipient. Moreover the highlighting of the correct twitter button can be improved by some design changes. The button could be set to a bigger size or get a special colour. Also a size adjustment of the direct messages button could be helpful to set it more in the background. Changing the arrangement is an additional consideration. The distance of the twitter button and the direct message button could be increased to prevent clicking mistakes. In addition it was noticed that the exit button is misleading. So it can be observed that the participants exit the application a few times accidentally to get to the previous view of the application. Changing the position of the button can be useful. In most applications the main operation elements are arranged close together. Main elements meaning elements which are necessary to navigate through the application like show profile or close the application. Also the sign out button is often placed in the upper right corner. Therefore the users are familiar to this

arrangement of the exit button. Moreover some participants wish for an additional return button which leads them to the previous screen. This one could improve the usability and should be placed at the current exit button position. To decrease misunderstandings a help button could be also thinkable. This button supports the user in getting familiar with *GazeTheWeb - Tweet*. A short explanation of the scroll function and some button functionalities could be displayed to make the application more transparent to the user. Another important point are areas where the user can rest his eyes. *GazeTheWeb - Tweet* offers some of this areas but it was noticed that a few more are desirable to reduce stress due to the high rate of feedback which is essential in eye tracking applications. The mentioned functions in the previous part should be revised so the user is fully satisfied by the functionality. The following list contains them once again.

- List which shows own followers.
- List with people which one follows.
- Linking hash tags and "@username" in tweets.
- Navigation to external links.

Additionally the discovered errors should be fixed. Tweets of large size must be displayed completely and not just partially. After liking a tweet this fact should be updated and displayed immediately at the liking number. Moreover all pictures should be displayed correctly without distortion of shape and colour. Finally the correct presentation of special characters in tweets should be implemented.

5.2 Future development

To ensure that the implemented improvements raise the usability of *GazeTheWeb - Tweet* another usability test is required. Moreover some important aspects can be extracted by the evaluation results. These aspects have to be considered for further eye tracking development.

- The distance between buttons needs to be large enough so users can easily focus on objects. If two objects are too close to each other the user is more likely to miss the correct object which raises frustration.
- There has to be an area to rest the eyes. Especially when using the system over a long time, the eyes need to rest so the user doesn't become overstrained.
- Since future applications should be adapted to eye tracking all buttons need to have a good size to make them easier selectable.
- To provide a harmonious and pleasant design clear fonts, thought-out arrangements of all elements and colours are important.
- If the user interface reacts constantly to the user's actions the usage of the system becomes more stressful.
- Joy is increased as soon as the user gets the chance to personalize the application's functions.

5.3 Final conclusion

The evaluation part aims to show that a system like *GazeTheWeb - Tweet* is preferable for eye tracking than using an emulator with web pages that is not adapted to it. Therefore the assumption is that an adopted system offers more usability. The results of the evaluation part show that an adapted application to the eye tracker effects a significant stress reduction which increases the user's satisfaction. Moreover it offers a suitable component size for the eye tracker whereby a direct selection is possible. Working with emulators and web pages the user experiences the usage as cumbersome and confusing which involves stress during usage. On the other hand *GazeTheWeb - Tweet* offers a harmonious design because of the adapted structure and size of all application components. This fact also increases the usability regarding the orientation and navigation.

In conclusion it can be noted that an application which is adapted to the eye tracker is more user friendly and more suitable for regular usage than a joint system.

Glossary

Action Bar

Context sensitive area whose content depends on the tweet the user is looking on. Can contain the options “view”, “like”, “respond”, “retweet” and/or “profile”.

Category Switch Bars

Two bars at both sides of the interface that were initially meant for switching between the subcategories of one page. Later they were used to provide scrolling for multiple tweets. For the final design, those two bars have been discarded..

Content Area

Area where the content, usually tweets or profiles, are shown. Tweets will be highlighted when the gaze hits them and scrolled up if they are the uppermost/lowermost tweets. No other events are triggered while looking on the Content Area.

Navigation Bar

Menu that is shown at all times and that does never change. Contains the menu entries “Browse”, “Profile”, “Home”, “Tweet”, “Private Messages” and “Search”.

List of Figures

Bibliography

[Bro] Brooke, John: *SUS - A quick and dirty usability scale.* https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&ved=0ahUKEwjx9-PlhtzMAhVMaRQKHRssAksQFghHMAU&url=http%3A%2F%2Fwww.uxforthemasses.com%2Fblog%2Fwp-content%2Fuploads%2F2011%2F04%2FSUS-System-Usability-Scale.doc&usg=AFQjCNEavVWtzR69wFHSPQBdnlwmXX3UdQ&sig2=_GytxOHLBt-awj1vxd4LrQ&bvm=bv.122129774,d.d24&cad=rja, . – Accessed: 2016-05-02

[Gro] Group, Human Performance R.: *Nasa Task Load Index (TLX): Paper and Pencil Package.* http://humansystems.arc.nasa.gov/groups/tlx/downloads/TLX_pappen_manual.pdf, . – Accessed: 2016-05-02

[JK03] Jacob, Robert J. ; Karn, Keith S.: *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research.* Oxford : Elsevier Science BV, 2003. – ISBN 0–444–510206

[mam] *MAMEM - Multimedia Authoring & Management using your Eyes & Mind.* http://www.mamem.eu/, . – Accessed: 2016-02-24

[Men] Menges, Raphael P.: *eyeGUI.* https://github.com/raphaelmenges/eyeGUI, . – Accessed: 2016-04-23

[MR02] Majaranta, Päivi ; Räihä, Kari-Jouko: Twenty Years of Eye Typing: Systems and Design Issues. In: *Proceedings of Eye Tracking Research and Applications.* New Orleans LA, 2002, S. 15–22

[Nie] Nielsen, Jakob: *Usability 101: Introduction to Usability.* https://www.nngroup.com/articles/usability-101-introduction-to-usability/, . – Accessed: 2016-04-29

[sus] *System Usability Scale (SUS).* http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, . – Accessed: 2016-05-02

[too] *Usability-Test.* http://usability-toolkit.de/usability-methoden/usability-test/, . – Accessed: 2016-04-29

[Ver] Vertanen, Keith: *NASA-TLX in HTML and JavaScript.* https://www.keithv.com/software/nasatlx/nasatlx.html, . – Accessed: 2016-05-02

[Zha06] Zhang, A.: *Advanced Analysis of Gene Expression Microarray Data.* World Scientific, 2006 https://books.google.de/books?id=6QLCJQ25VUwC. – ISBN 9789812566454

Questionnaires

Evaluation GazeTheWeb-Tweet

Körperlich eingeschränkte Menschen werden durch ihre Behinderung oft von der Außenwelt abgeschottet. Da sie nicht in der Lage sind, selbstständig das Haus zu verlassen, verlieren sie oft ihre sozialen Kontakte und leben in Isolierung. Moderne Technologien ermöglichen es ihnen, ihre Kontakte wiederherzustellen, sie zu pflegen und am Leben teilzunehmen. Besonders soziale Netzwerke spielen eine wichtige Rolle, um ihnen das Fenster zur Welt zu öffnen. Leider bieten die meisten sozialen Netzwerke, außer Maus und Toucheingabe, keine alternativen Eingabemöglichkeiten um sie zu nutzen, z.B. Eye-Tracker.
Das Projekt “Multimedia Authoring and Management” (MAMEM) beschäftigt sich zurzeit mit der Weiterentwicklung von Social Network APIs, um Gedanken und Augenbewegungen als Eingabe zu ermöglichen.
Das Forschungsprojekt des WeST-Instituts unterstützt das Projekt durch die Erstellung der Eye-Tracking Anwendung “GazeTheWeb-Tweet” für die Nutzung von Twitter.
Vielen Dank für Ihre Teilnahme als Testkandidat in unserer Nutzungsstudie!

Fragen zur Person

1. **Sind Sie männlich oder weiblich?**
   Markieren Sie nur ein Oval.

   - ◯ Männlich
   - ◯ Weiblich

2. **Wie alt sind Sie?**

   _____

3. **Als was sind Sie tätig?**
   Markieren Sie nur ein Oval.

   - ◯ Student
   - ◯ Mitarbeiter
   - ◯ Sonstiges: _____

4. **Kennen Sie Twitter?**
   Markieren Sie nur ein Oval.

   - ◯ Ja
   - ◯ Nein
   - ◯ Was ist Twitter?

5. **Wie schätzen Sie Ihre Fähigkeiten im Umgang mit dem Computer ein?**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
Experte	◯	◯	◯	◯	◯	Anfänger

Nutzungsverhalten

6. **Wie oft lesen Sie Tweets?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

7. **Wie oft schreiben Sie Tweets?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

8. **Wie oft retweeten Sie etwas?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

9. **Wie oft nutzen Sie die Folgen-Funktion?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

10. **Wie oft liken Sie etwas?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

11. **Wie oft schreiben Sie Privatnachrichten?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

12. **Wie oft nutzen Sie die Trend-Funktion?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
mehrmals pro Woche	◯	◯	◯	◯	◯	nie

Bevor es los geht...

13. **Ich fühle mich angestrengt.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

14. **Ich fühle mich müde.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

15. **Ich fühle mich gestresst.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Bereitgestellt von

Evaluation GazeTheWeb - Tweet

1. **Ich fühle mich angestrengt.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

2. **Ich fühle mich müde.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

3. **Ich fühle mich gestresst.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Funktionsbezogene Fragen

4. **GazeTheWeb-Tweet stellt alle Funktionen, die ich brauche, bereit.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

5. **Welche Funktionen fehlen Ihnen?**

.....

.....

.....

.....

.....

6. **Die bereitgestellten Funktionen reagieren wie erwartet.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

7. **Welche Funktionen reagieren nicht wie erwartet?**

8. **Ich erhalte genug Feedback/ Rückmeldungen bezüglich meiner Aktionen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

9. **Welche Funktionen geben nicht genügend Feedback/ Rückmeldung?**

Fragen zum Design

10. **Ich empfinde das Design als modern.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

11. **Ich empfinde das Design als harmonisch.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

12. **Das Design trägt zur Orientierung bei.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

13. **Ich empfinde das Design angenehm für das Auge.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

14. **Ich empfinde die Schrift als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

15. **Begründung/Anmerkung (optional):**

.....

.....

.....

.....

.....

Navigationsleiste

16. **Ich empfinde die Größe der Buttons in der Navigationsleiste als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Actionbuttons

17. **Ich empfinde die Größe der Actionbuttons als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

18. **Ich empfinde die Größe der Tastaturtasten als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Fragen zur Bedienung

19. **Die Scrollfunktion ist intuitiv zu bedienen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

20. **Die Scrollfunktion ist leicht zu finden.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Schlussteil

21. **Ich würde die Anwendung regelmäßig benutzen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

22. **Ich empfinde die Anwendung als unnötig schwierig.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

23. **Ich empfinde die Anwendung als leicht nutzbar.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

24. **Ich denke ich bräuchte die Unterstützung einer technisch versierten Person um die Anwendung nutzen zu können.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

25. **Ich denke die einzelnen Funktionen sind gut in das System integriert.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

26. **Ich empfand das System als zu inkonsistent.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

27. **Ich denke, dass die meisten Personen sich schnell in das System einarbeiten könnten.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

28. **Ich finde die Nutzung der Anwendung sehr mühsam.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

29. **Ich habe mich bei der Benutzung des Systems stets sicher gefühlt.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

30. **Ich musste mir viele Dinge aneignen bevor ich das System nutzen konnte.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

31. **Haben Sie noch weitere Anmerkungen oder Verbesserungsvorschläge zu GazeTheWeb-Tweet?**

.....

.....

.....

.....

.....

Bereitgestellt von

Evaluation MobileApp mit OptiKey

1. **Ich fühle mich angestrengt.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

2. **Ich fühle mich müde.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

3. **Ich fühle mich gestresst.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Funktionsbezogene Fragen

4. **Die MobileApp stellt alle Funktionen, die ich brauche, bereit.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

5. **Welche Funktionen fehlen Ihnen?**

.....

.....

.....

.....

.....

6. **Die bereitgestellten Funktionen reagieren wie erwartet.**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

7. **Welche Funktionen reagieren nicht wie erwartet?**

8. **Ich erhalte genug Feedback/ Rückmeldungen bezüglich meiner Aktionen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

9. **Welche Funktionen geben nicht genügend Feedback/ Rückmeldung?**

Fragen zum Design

10. **Ich empfinde das Design als modern.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

11. **Ich empfinde das Design als harmonisch.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

12. **Das Design trägt zur Orientierung bei.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

13. **Ich empfinde das Design angenehm für das Auge.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

14. **Ich empfinde die Schrift als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

15. **Begründung/Anmerkung (optional):**

.....

.....

.....

.....

.....

Navigationsleiste

16. **Ich empfinde die Größe der Buttons in der Navigationsleiste als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Actionbuttons

17. **Ich empfinde die Größe der Actionbuttons als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

18. **Ich empfinde die Größe der Tastaturtasten als geeignet.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Fragen zur Bedienung

19. **Die Scrollfunktion ist intuitiv zu bedienen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

20. **Die Scrollfunktion ist leicht zu finden.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

Schlussteil

21. **Ich würde die Anwendung regelmäßig benutzen.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

22. **Ich empfinde die Anwendung als unnötig schwierig.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

23. **Ich empfinde die Anwendung als leicht nutzbar.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

24. **Ich denke ich bräuchte die Unterstützung einer technisch versierten Person um die Anwendung nutzen zu können.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

25. **Ich denke die einzelnen Funktionen sind gut in das System integriert.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

26. **Ich empfand das System als zu inkonsistent.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

27. **Ich denke, dass die meisten Personen sich schnell in das System einarbeiten könnten.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

28. **Ich finde die Nutzung der Anwendung sehr mühsam.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

29. **Ich habe mich bei der Benutzung des Systems stets sicher gefühlt.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

30. **Ich musste mir viele Dinge aneignen bevor ich das System nutzen konnte.**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft nicht zu	◯	◯	◯	◯	◯	trifft voll zu

31. **Haben Sie noch weitere Anmerkungen oder Verbesserungsvorschläge zur mobilen Twitter-Applikation in Kombination mit OptiKey?**

.....

.....

.....

.....

.....

Bereitgestellt von