eyeTube

Project Lab and Research Lab

University of Koblenz-Landau, WS 2016/17

Supervisors: Dr. Chandan Kumar, Korok Sengupta, Raphael Menges

Participants: Denise Dünnebier, Mariya Chkalova, Min Ke, Yessika Legat, Arsenii Smyrnov, Benjamin Becker, Daniyal Akbari, Matthias Greber, Ramin Hedeshy, Steven Timothy Schürstedt, Jannis Eisenmenger

Table of Contents

1. Introduction

Life is difficult for disabled people. If you have no control over your hands or no hands at all, it is very difficult to perform even simple tasks in your daily life. Spending your free time to watch some videos is impossible without support of another person. But not with EyeTube! Controlling a computer only with your eyes is perfectly possible in our age. But it is hard. So we (a Team of students supervised by Dr. Chandan Kumar at University Koblenz) made it our task to develop a web page for browsing videos only using your eyes. It is a video platform (e. g. YouTube) and it allows the user to browse for videos based on the YouTube API, watch, search and rate them. And make it easier to control with gaze than, for example, YouTube. Normal webpages are difficult to control with eyes only, because all the buttons are very small and difficult to hit. Some features, like fast forwarding a video, are difficult to control at all.

When someone navigates our webpage with his eyes, it provides us with additional data. To make gaze control even an advantage we used this data to improve search algorithms and recommended videos.

This report is divided into two parts. In the first part of the report we will describe our page and its functionality in great detail. In the second part we will evaluate a thesis based on an experiment we made with our page.

2. Technologies

2.1. Devices

We used SMI REDn Scientific Eyetracker1 to offer an application which can be controlled over gaze. Moreover we used the EMOTIV Epoc BCI2 (brain computer interface) to compliment the gaze-based interaction.

2.1.1. SMI REDn Scientific Eyetracker

For the eyetracking part of the study we use the REDn Scientific eyetracker by SMI. It is a lightweight (75g) monitor mounted device with good accuracy (0.4°). The operating range of 40 - 100 cm is ideal for our use case since it allows our participants to sit comfortably in front of the screen while watching videos.

2.1.2. EMOTIV Epoc BCI

The Epoc BCI by EMOTIV is our brain computer interface of choice. It is a lightweight 14-channel BCI device. Its strong point is the relatively short setup time. Having only 14 channels it lacks some of the accuracy of heavier models but it is sufficient for our purpose. The SDK separates into six emotions - Interest, Stress, Engagement, Focus and Relaxation.

2.2. Frontend

The following technologies are used in our web application to reach specific aims, which will be described in the lower sections.

2.2.1. Bootstrap

Bootstrap[3] is a free and open-source front-end web framework for developing and designing websites and web applications. It contains HTML- and CSS-based design templates for buttons, navigation and other interface components. Furthermore it contains JavaScript extensions. Because of this helpful variety we decided to use Bootstrap. The navigation bars and the layout were implemented with Bootstrap.

1 https://www.smivision.com/eye-tracking/product/redn-scientific-eye-tracker/
2 https://www.emotiv.com/epoc/
3 http://getbootstrap.com/

2.2.2. HTML5

HTML5[4] (Hypertext Markup Language) is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard. Our web application exists of three main websites. HTML5 was used for the structure and presentation of all web pages.

2.2.3. CSS

CSS[5] (Cascading Style Sheets) is a language that describes the style of an HTML document. It describes how HTML elements should be displayed. The whole design of the web applications has been done with CSS. For each website exists an own CSS-file. In addition there is another file for the upper navigation bar, which is included in every HTML-file to avoid redundant style and improve the performance.

2.2.4. JavaScript

JavaScript[6] is the programming language of HTML and the Web. Basically we used it to save important data for the backend (e. g. to access the video id). One instance in which we have used JavaScript is the star rating to enable a half star rating which can be filled dynamically in the backend.

2.2.5. Ajax

In order to send asynchronous requests to the server-side we have used XMLHttpRequest which provide us to keep a record of user actions on the web application without blocking the execution of code which creates freezing on the screen and an unresponsive user experience[7]. This technology is used in star rating section as well as in the process of collecting fixation point while users watching a video.

2.2.6. Youtube API

YouTube iframe API[8] is providing the videos data for our web application. It also allows us to query and search through the available videos base on a keyword. Communication with the YouTube API is possible using both JavaScript and Python.

As far as we have our own visualisation for the player control buttons we are using YouTube JavaScript event handler functions to change the state of the player whenever user triggers an event. An event could be playing, stopping, pausing a video, changing the volume or etc.

---

4 https://www.w3schools.com/html
5 https://www.w3schools.com/css
6 https://www.w3schools.com/js
7 https://www.w3schools.com/js/js_ajax_intro.asp
8 https://developers.google.com/youtube/

2.2.7. JQuery

jQuery is a JavaScript based frontend web application framework which greatly simplifies JavaScript programming. JQuery is used in many of important sections of eyeTube. The most significant usage of JQuery in the application is to establish connection with the player iframe.

2.3. Django/Python

The technical tools and libraries described below are used in backend of our web application.

2.3.1. Python

Python is an object-oriented programming language which is simple and has clean syntax. Python is used in our web application because the expressiveness of it is easy to learn and it has various applications from web development and large libraries, such as Django, Numpy and Pandas. And unlike Java the user is not forced to define classes in Python language but it is possible to do so when required. It is also worth mentioning that YouTube have also used Python in their development.

2.3.2. Django

Django is an open-source and high-level web framework written in Python. It follows the Model-View-Control (MVC) architectural pattern and the official project site describes Django as "a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source."[9] Their motto is “Don’t repeat yourself”.

Django has been developed by keeping in mind the front-end development; therefore it is user-friendly and easy to learn by those who intend to work on front-end development.

But the two key factors that set Django apart from other frameworks like Pyramid and Flask are:

1. Django is more emphasized to start with ready to use modules and bundles which will be really time saving and convenient to use.
2. Django compared to the other frameworks have much more documentation and support, which would be really helpful for developers in time of need.

2.4. MySQL

MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.[10]

[9] django Project (https://www.djangoproject.com/)
[10] MySQL (https://www.mysql.com/about/)

MySQL have many advantages such as being really easy to use, from setup and installation to the implementation and performing tasks. It is widely used on different platforms and because of it there are lots of available documentations and support communities for developers to refer to. Performance is also another strong point of MySQL and it is vital for some web application to be considered, especially when the data in database grows into larger scales.

2.5. Lab streaming layer

"The lab streaming layer (LSL) is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access as well as optionally the centralized collection, viewing and disk recording of the data."[11]

It is developed by the Swartz Center for Computational Neuroscience at the University of California San Diego. LSL synchronises the timestamps of the data streams acquired from different devices.

[11] David Medine (https://github.com/sccn/labstreaminglayer/wiki)

3. Challenges and Research

3.1. Team Django

3.1.1. Data Structures

For the Structure of the Data, we had many suggestions. During a meeting, after doing some individual research, we had many ideas, which data structures would be most suitable for which data.

Unfortunately, most suggestions could not be provided, because at a later point, there was an requirement added to the project to use databases. The databases of the university do not support anything but mySQL and SQLite.

For most of the data, databases were proposed anyways (see illustration below).

	Hashtable	Hashmap				ArrayList	
	Set	Dict	Trie	rel DB	Graph	List	Splay
User-user-similarity							
Tags -> Videos							
user-video-rating							
user-video-emotions							
user-video-eye							
recommendations							
playlist							
userprofile							
videoprofile							
user-session-emotions							
user-session-eye							
user-searchQueries							
history							

Table 3.1 Chart of suggested data structures.

Yellow: possible, light green: good, dark green: best

Database design and implementation

After much consideration and discussions within the group, we decided to model the database as an EER diagram, so there would be easier and more clear understanding of the database structure and its components. And later it could have be used as a base to implement the project database.

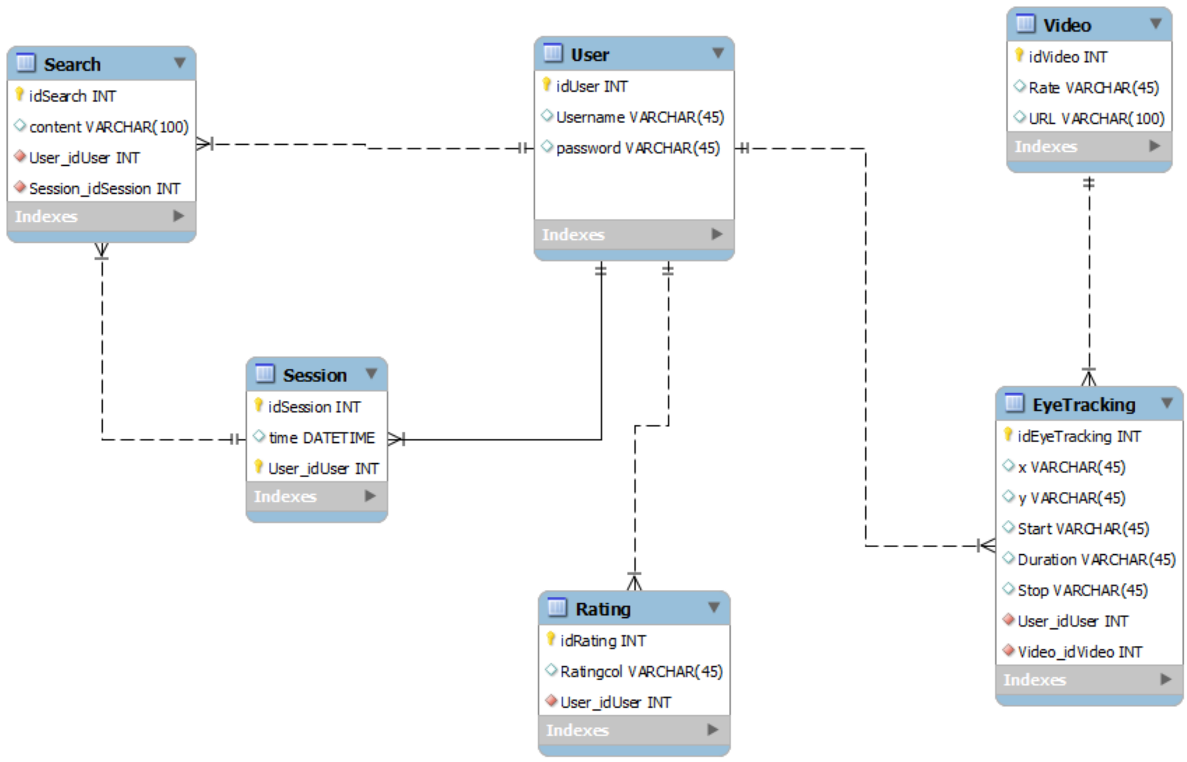

Figure 3.1Connections between tables, including primary and foreign keys

In the diagram, all the required tables are visible as well as their primary and foreign keys and how they are connected to each other.

The next step was to add some test data in our database and see if there are any inconsistencies in the structure and try to fix any problems that we encounter during this process.

The challenging part was when we needed to make a connection to the database and we needed to use different libraries to make that possible. But unfortunately there was lots of inconsistency between the version of python which we used.

Similarities between users

For the similarities between the individual users, we proposed a Graph, as Graphs can be implemented very easily in python with libraries such as NetworkX. Graphs are a very useful data-structure, when you want to connect different elements (nodes).

In the end, we skipped using NetworkX. The similarities are stored in a relational database, which is equivalent to an adjacency matrix. An adjacency matrix is one possible representation of a graph.

Tags

To store the tags for the videos, the most efficient and elegant solution would be to use a trie. Tries are one of the most space-efficient and fast data structures, to store strings or dictionaries. The idea is to make a tree out of letters of the words. The maximum height of the tree is equal to the length of the longest word. This leads to short times for search. Words with the same prefix share the same path in the tree for that prefix. This leads to little space requirements.

Another proposal was to use a splay. Splays are quite uncommon binary trees. They are special, because whenever a value is added or looked at, it gets 'splayed' to the top of the tree. This means, elements, which are frequently used are quick to access. Especially for huge amounts of data like for an online-video player with millions of tags, this would make sense. As our project is small and for research-purposes only, we skipped this idea.

The current solution is again the database. The trie-version was implemented, but not added to the project because of the project requirement.

Devices

To store the data from the devices, we proposed to use a database, where the raw data would be uploaded at runtime. This is very close to the actual implementation.

For the eye-tracker, raw data is processed first. Only the fixation points and average values are uploaded. At the same time, for safety reasons and because it is much faster than a database, the data is stored as file. To create the file, a library called pickle is used. For the BCI only the average emotion rates are uploaded to the database.

User-Actions

One proposal was to record each single click from the users and all data, which would be helpful to reconstruct the user-behaviour. It might be useful to add this functionality in future, to be able to tell, what was happening at what time. Right now, the recording starts when a video starts playing, is paused when the video is paused, and ends when a video ends. For the current experiment, recording each single user-action is not necessary. This is, because the evaluation only requires the data from the devices during the playback of the videos. So far, we only proposed a possible structure, how the databases could be implemented. For this approach, all streams from the devices would be recorded without interruptions and all user-actions would be recorded (see illustration below). This approach would work with offsets inside the databases. After each session, it would be possible to assign the recorded data from the devices to the actions of the users. Because there was no positive feedback from other participants of the project and it was not required for the experiments, we skipped the idea.

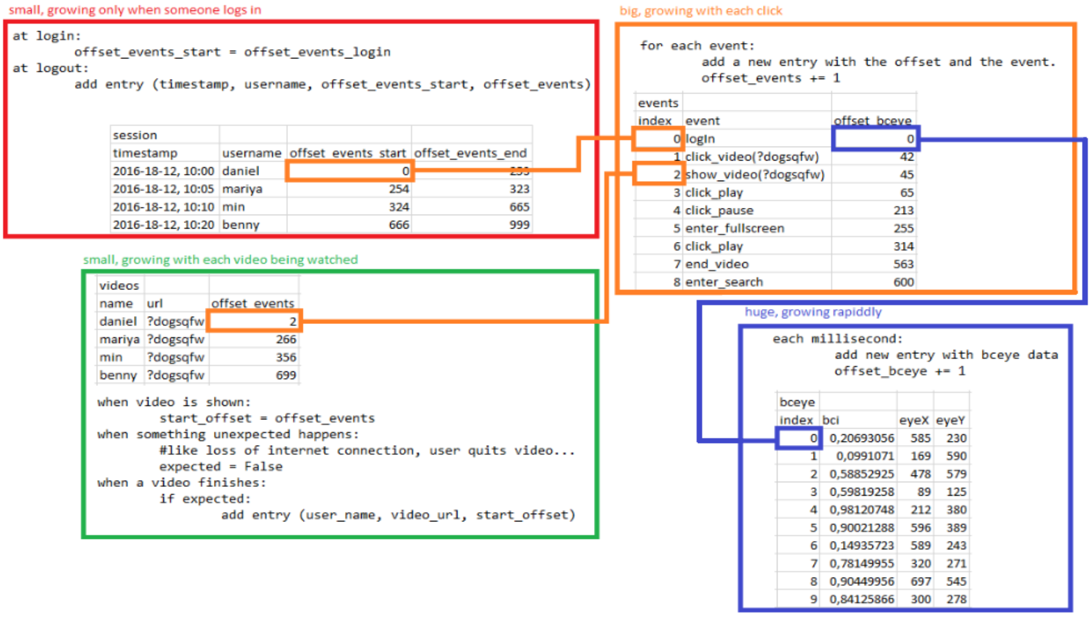

Figure 3.2 Databases to store user-actions
(orange), videos (green), bci and eye tracking data (blue), sessions (red)

3.1.2. Proposals for usage of data

Another task of team Django was to think about what we can be done with the brain- and gaze data. Many of our proposals were about how the data could be used to classify the videos. Other proposals were about comparing users. Of course it was impossible to realize all proposals in our project. However, it was an important part of the project to propose those ideas.

The final direction of the project was influenced by the proposal described in chapter 3.2.2. This proposal is also part of the evaluation of the project.

3.1.2.1. Retrieving information about videos

Classifying videos

We suggested two possible ways, which could be implemented to classify videos based on the data from the devices.

One way idea is to calculate average values for the feelings from different users. The most significant feelings would be used to classify the video.

The other idea was to calculate average feelings for the videos. But the results from those calculations would then be used to calculate similarities between the videos. Those similarities would be very useful to suggest new videos to users after watching a video.

Video Recommendation

To identify the best way for our project, we needed first to understand what are the possible methods and currently used recommendation algorithm. Therefore we decided to find and investigate as much papers as possible about this issue. And summarize them into a table so we would have a better understanding of the subjects.

Abstract - short summary, 1-2 sentences	#Citations from google scholar	relevancy1-5 "stars" how relevant for our project	relevancy - why is it (not) relevant? 1-2 sentences
Uses coviewership-Model (User Similarity Concept) for video recommendation. Two factors 1.User Type 2.Video Type	0	****	It is about video recommendation with complete examples and scenarios.
Experiment with use of eye tracking to find solutions to given questions. Evaluating the influence of ranking in the experiment.	588	*	It is not helpful in regards to video recommendation process.
It explains how the youtube recommendation works and how the related videos are calculated.	360	***	It could be helpful to understand youtube process,so that it can gives us better ideas for our algorithm
This goes more into detail about youtube process.	6	***	Same case as paper06.

In this research we focused on how much the paper is relevant to our project and number of citations of the paper based on the Google scholar.

Find important parts of a video

There are peaks in the graphs of the BCI-data. Some of those peaks are random, caused by muscle contractions or other kinds of noise. But if many users have peaks at the same time during a video, it might be interesting to find out why. Can we say which parts of a video are important, just by the Graphs of the Brain Data?

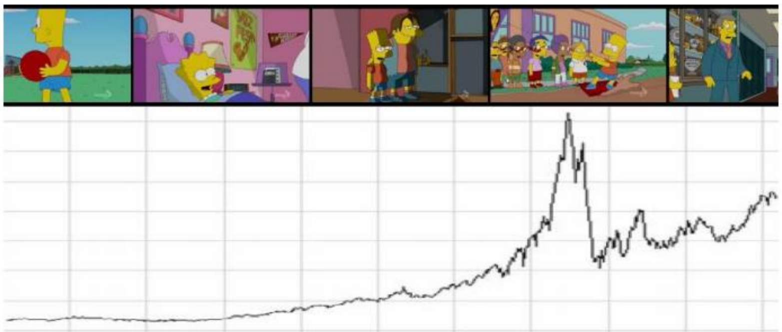

Figure 3.3Finding important parts of a video based on BCI data

We did not do any further research on this, because we decided to go for the proposal described in chapter 3.2.2.

3.1.2.2. Similarities between users

Interesting questions, when you compare the BCI- and gaze-data of two users, could be: “If two users have similar brain-activity and similar gaze-points during a video, are they similar? Do they like the same videos?” We decided to focus on those questions. Based on that assumption, we created a search algorithm, which takes into account the similarities between users.

Calculating similarities

The idea is to compare the BCI- and Gaze-Data for each pair of users, for each video. This means, that there will be one calculation for each pair of users that watched the same video. All fixation-points and also all entries for the BCI-Data are being compared. The result is a single similarity value for each user-pair for each video. An average over all videos is calculated for each pair of users. This value tells, how similar users are.

Search algorithm

As a second step, the similarity will be used inside a search algorithm. Whenever a user searches for a video, there is a list of all results with tags that match the search-request.
For each video in that list, a score is calculated. The score is based on the ratings of other users. But the score is not an average of the other ratings. Instead, the ratings are weighted. The weight is the similarity between user X and the user, who rated the video. The score is divided by the sum of all similarities between user X and everyone who rated the video.
The List of videos is then sorted by the scored, descending. The first video in the sorted list is the video at top of the user's search results.

Implementation for eyeTube

We decided to implement this approach in the project. The algorithms to calculate the similarities were implemented by the device-teams. Our team provided the idea, the database-queries and the implementation of the search algorithm.

Because our project is very small, we decided to mix the search results from our own algorithm with search results from YouTube. This not only gives more and better results to the users of our web player. It now is also possible for our database to grow, because new videos from YouTube can be viewed and entered to the database.

The idea of calculating similarities between individual users is also key part the final evaluation.

3.1.2.3. Analyzing the design of the web-page

Another one of our very early suggestions was to use the data from the eye-tracker to give the designers of the web-page a visual feedback. Fixation-points would be displayed on a screenshot of a web-page. The colours of the fixation-points would be influenced by the emotions of the users. The size of the circles would represent the duration of the fixation. This way, the designers would be able to tell, which regions of the web-page are the most important. Also, it would help them to improve the design. This suggestion was not implemented, as the focus of the project went towards analysing videos.

A similar visualization for videos was implemented by the Eye-tracker-Team. Also the idea of the gaze-points was implemented by that team to calculate similarities.

Figure 3.4 Visualization of fixation-points

3.1.3. Connection to Databases for Eye tracking-Team

This work was requested by the eye tracker-team. The queries for Databases not only include getters and setters. Many calculations, for example for averages and total amounts, are done. This chapter only describes the four most notable functions. Besides those four functions, there are of course many other functions to help the Eye tracking-Team to use the required databases (see EyeTracking/player/helpers/dbmanager.py).

missing_similarity_calculations

"missing_similarity_calculations" returns a list of all userpairs for all videos, where the similarity can be calculated from the existing data, but the similarity is missing in the current database. To generate this list, first the database UsersVideoSim is used to get a set (S) of all existing similarities. After, the database FixationPoints and all files are used to generate a set (T) with all existing video-user-pairs, for which fixations exist, is generated. Finally, the difference between (S) and (T) is the set of calculations, which are missing.

update_similarity

Another function "update_similarity" calculates the arithmetic mean of similarities over all videos for one user-pair. This similarities are stored in the database UsersSim.

store_fixations

"store_fixations" is used to store a whole list of fixation points for one user for one video. The list is stored both in the database FixationPoints and as a file. The filename consists of the user-id and the video-id. The files are created, so if there are any problems with the database during or after the experiments, the data is not lost. Also, saving and loading data from files is much faster than using a database. Also, the total amount of fixation-duration is calculated from the single values and stored in a database.

load_fixations

"load_fixations" returns a list of all fixation-points for one user for one video. first tries to load the requested fixation points from the file. If there are any problems with that, the fixations are loaded from the database.

4. Implementation

4.1. Code Architecture

4.1.1. MVC pattern

The project follows Model-View-Control architectural style, supported by the Django framework that we used for building the application.

Django framework provides the default hierarchy of the project structure. After the new Django project is initialized, the standard code files are created organized on the way, to follow framework conventions. So we extended the project code around the predefined architecture. We added required Python files and integrated them to the project in a way, so that the standard Django structure will not be broken.

In general, there are a lot of conventions provided by the framework that should be followed. Thus, as well as the code, that should be executed only once on the app start event, must be written in the particular place only; the html, cs, js files should be only organized following the specific folder paths; the way how the urls should be formed, etc. So compared with the other technologies, that also provides the project structure for the MVC pattern application (in particular ASP.NET MVC), Django required the strict adherence to the conventions and less flexibility to developers to customize the architecture.

4.1.2. Project infrastructure

First of all, after the new Django project is initialized from the command line utility, the following folders and files were created:

➔Here the folder **EyeTracking** contains the files with the general project setup:

- ➢ **settings.py** represents the overall configurations the following information:
  - registered applications within the project;
  - connection the database, account credentials used to establish connection;
  - redirect urls;
  - relative paths within the file structure to the static files with html, css, js code;
  - time zone, language preferences, etc.
- ➢ **url.py** file contains the registered global project urls, such as relative links to the login, logout pages and for each application of the project.

➔File **manage.py** is the main executable file of the Django framework.

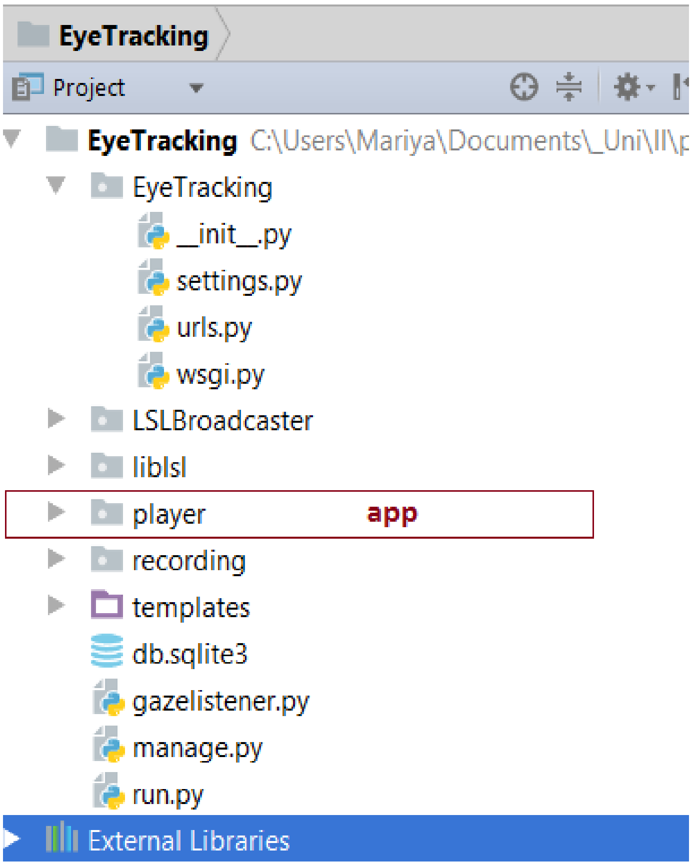

Figure 4.1 Project infrastructure

Many actions for the Django project could only be done from the command line, by executing the command "Django manage.py ...parameters…", where the parameters define the particular action. Thus, Django server could be started/stopped by referencing manage.py; admin account is also could be created only from the command line; the migration of the database changes are also executed with the reference to manage.py and could not be done from the IDE, like PyCharm.

➔Folder **templates** provides html files for the login and logout pages.

➔Folder **player** contains the files of the application "player", which was registered in the project.

In general, Django project comes with the idea of the separate applications within the project. Thus the project itself is just the infrastructure of the common settings, login, logout functionality and maybe landing page. But after that, each functional unit of the web site should be registered as separate application. In our project we have 1 application - player.

4.1.3. Application "player" structure

Now, let's take a look directly on the structure of the registered application "player" within the Django project:

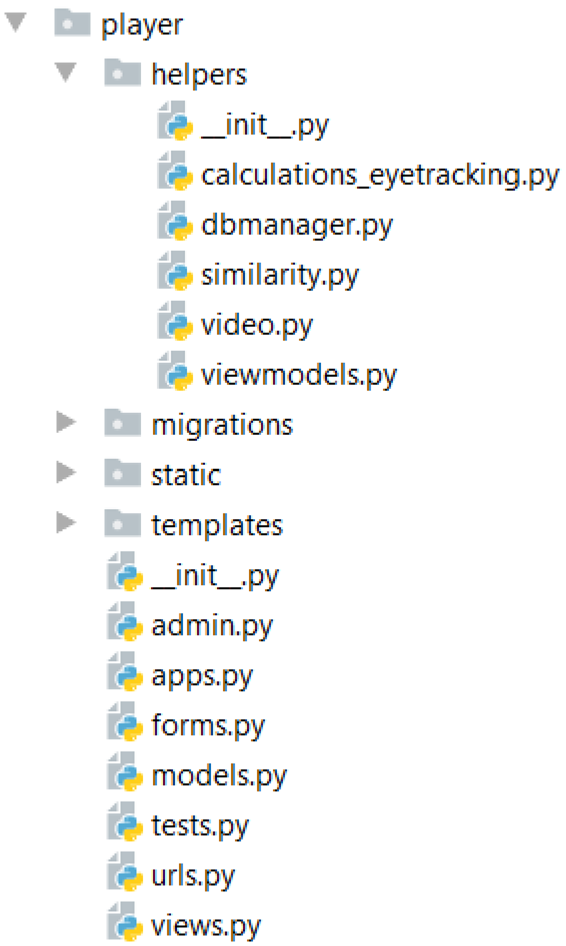

Figure 4.2 Application "player" structure

➔Python files located directly in the folder player are the standard files, required to run Django web site. Below is the explanation of the roles of some files that we customized:

- **apps.py** contains the code to initialize application configuration. It is also the right location to place the code, that should be executed only once on the application start. In our case we start listening to the eye interface here, once the application is launched.
- **models.py** is the file, that contains the classes for the custom tables, that we want to create in the database.

It is necessary to mention here, that apart of the listed classes from the models.py file, Django framework creates also the other standard tables with user credentials, authentication details, etc. The tables, created in the file models.py are treated as the application specific tables, so after applying the migration to the databases with the new table classes, they will be created using the following name convention: projectName_classname (case insensitive). In our case, we have player_usersim table for class, name UserSim.

Below is the diagram of all tables, created in the database by applying the migrations. So we used the Code First approach by working with the database, no manual changes from the database side were made.

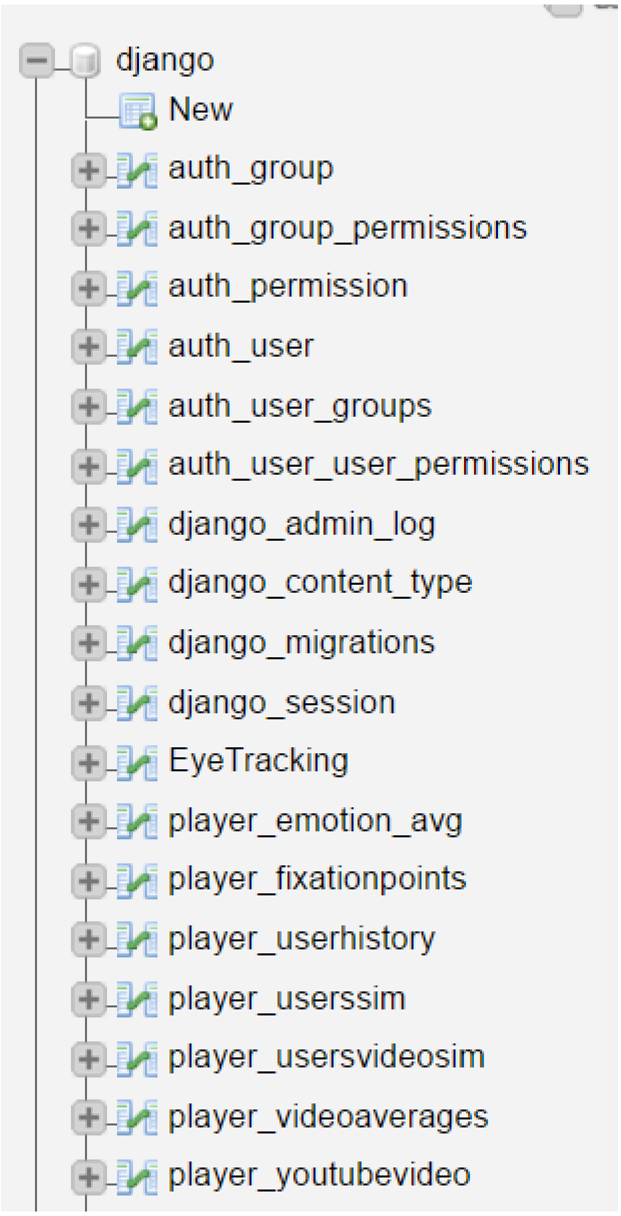

Figure 4.3 Database structure

(The table EyeTracking was created manually for the test purposes, thus it does not follow name conventions and is not used by the project).

- **views.py** is the main file with the execution logic. It is responsible for rendering webpages and the connection between user interaction with the web pages and processing on the backend. The file contains classes and methods for each view, that are used to pass data to the views and process user inputs and actions.
- **urls.py** contains the registered urls, specific for the application "player" (while the inner file urls.py (described above), provided the initialization of the global web site urls, applicable for all applications of the project). For each view class and method from the views.py file, it is necessary to register url here in order to execute method or render the view.

➔**folder templates and static** contains the html and css,js files correspondingly. Below is the inner content:

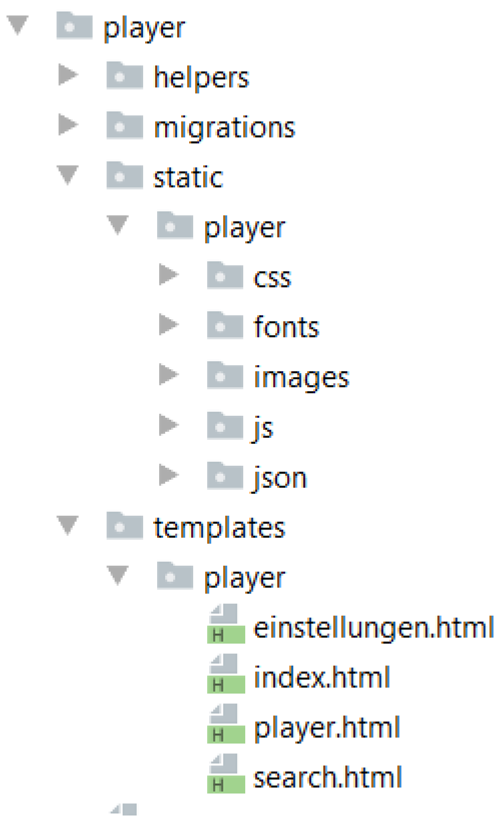

Figure 4.4 Front-end structure

Html files represent the mixture of the common html schema with the python code, integrated inside. Python code helped to process the data passed from the backend. The navigation to the pages and passing the parameters to backend methods (like the search keywords or rating values) are also done due to the Python code integrated to the views. Another example is the repeating sections with the different video categories on the Main page. This is 1 section of the common html schema and python loop, that repeats the section 4 times, passing 4 different lists of videos to each section.

➔**folder helpers** contains the main py files with the logic. This folder was not initialized by the Django framework, we created it manually in order to place their custom py files and separate them from the standard Django py files. It contains the following files:

- **dbmanager.py** file provided the methods to work with the database, store and retrieve data
- **video.py** contain the key algorithms, such as updating user similarity, searching keywords in our database and YouTube and operations of eye-tracking device. The search-algorithm described in chapter 3.2.1.2, is implemented in video.py as well
- **similarity.py** returns videos from our database, which match a search string. Similarity.py uses machine learning algorithms from Sklearn and also language processing functions from NLTK for videos retrieval from the database.

➢ In **viewmodels.py** file more objects are created for implementing the ratings of the videos

4.2. Design and Functionality

4.2.1. Logo

Figure 4.5 Logo

The design of the logo is based on the outline of an eye. It shows the striking features of the human eye. The focus of the logo is placed on the iris. The play icon is in the colour red (RGB (228, 45, 39)) and above this icon a heart rate is displayed. The curve and the outline of the iris are shown in grey (RGB(133,133,133)), which gives a clear contrast to the background and the rest of the logo. Both, the shape of the eye and the play icon, appear in the same colours as the YouTube scheme. It is used to evoke associations of YouTube.

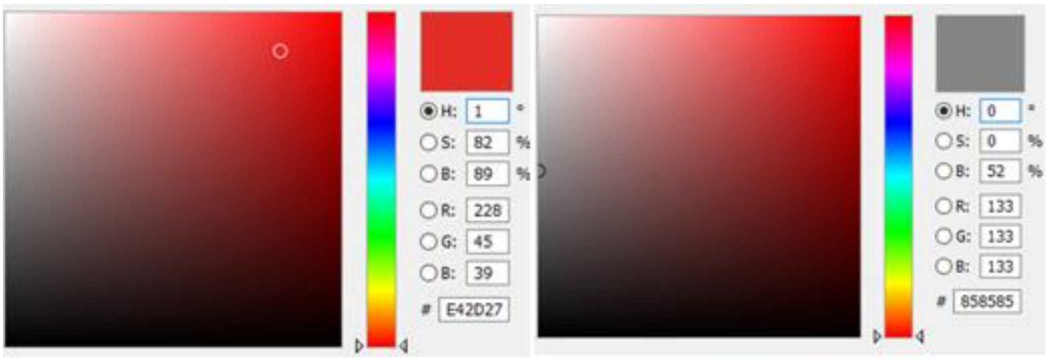

Figure 4.6 Colour schemes

4.2.2. Web Pages

The web pages are built with darker colours, going from pitch black to light grey. The aim was to create a comfortable-to-look, even if you use it for a longer period of time. The background and header/footer are in a dark grey. Buttons and intractable items are in light grey, so they pop out and are easier to see. The font colour is white to create a clear contrast to the dark background.

4.2.3. Navigation Bars

Top Navigation Bar (displayed on every webpage)

Figure 4.7 Top Navigation Bar

From left to right the top navigation bar consists of the following elements:

Element	Functionality
Logo	Go back to main page
Button “Left Arrow”	Go one step back in browser history
Search bar	Search for some videos
Button “Smiley”	Rate a video with the five star rating (only visible in player page)
Button “Star”	Select your feeling after watching a video (only visible in player page)
Button “User”	Open the drop-down menu and manage the account: ● Settings: Change either to mouse or eye tracking control; hide or show mouse cursor; Change the video quality ● Information of the web application ● Help: only placeholder ● Logout

Bottom Navigation Bar (displayed on the player page)

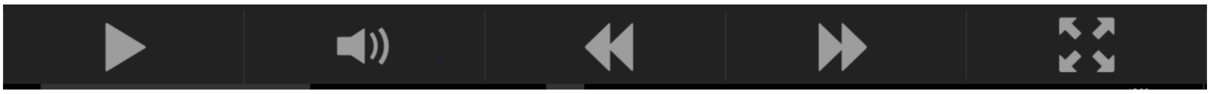

Figure 4.8 Bottom Navigation Bar

From left to right the bottom navigation bar consists of the following elements

Element	Functionality
Button “Play”	Start the video
Button “Volume”	Change the volume; Mute/Unmute
Button “Rewind”	Rewind the video 10 seconds
Button “Fast forward”	Fast forward the video 10 seconds
Button “Full screen”	Activate fullscreen mode

When a user selects the play button the video starts and a progress bar appears. This progress bar shows the current time and the whole duration of the video. It visualize how much time is still missing. Furthermore the play buttons changes to a pause button. After selecting this button once the video stops and the pause button changes to a play button again.

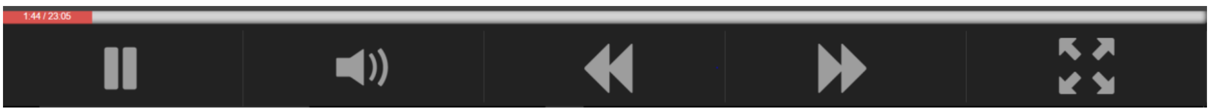

Figure 4.9 Play bar

When a user want to change the volume he can select the button for the volume. After that three different buttons appear: a first button to increase the volume, a second button to mute the volume and a third button to decrease the volume. In addition to that, a small rectangle is visible, which shows the current volume in the range from 0% to 100%.

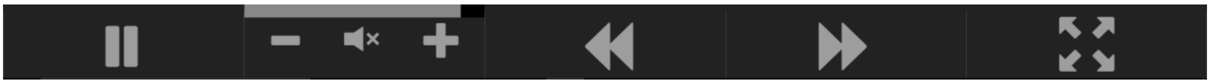

Figure 4.10 Sound bar

4.2.4. Login Page

4.2.4.1. Design

The login pages enables the user to log into his personal account. The account stores data about favourite videos of the user and enables the website to calculate what videos the user might want to see.

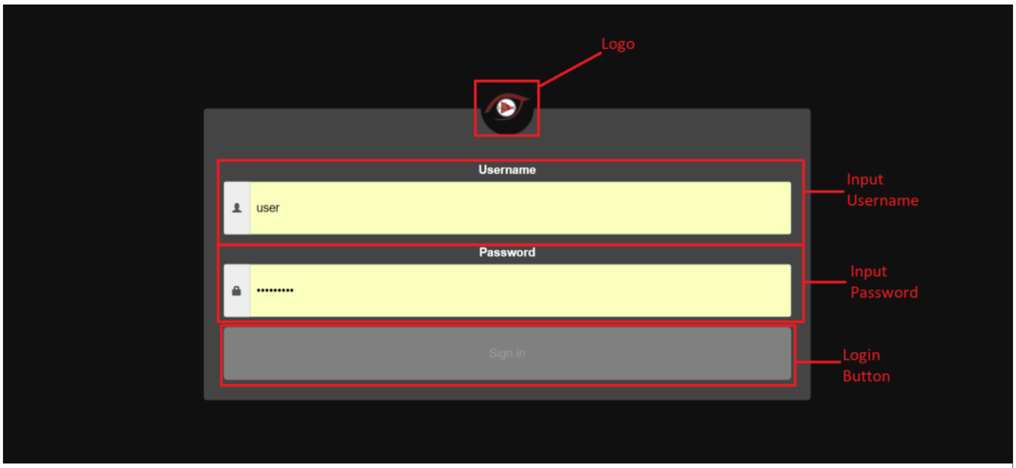

Figure 4.11 Login page

4.2.4.2. Functionality

User accounts to login into the application are created from the admin interface of the Django service.
Admin section is available via the relative path ~/admin/
In order to access admin interface, it is necessary to login with Admin credentials.

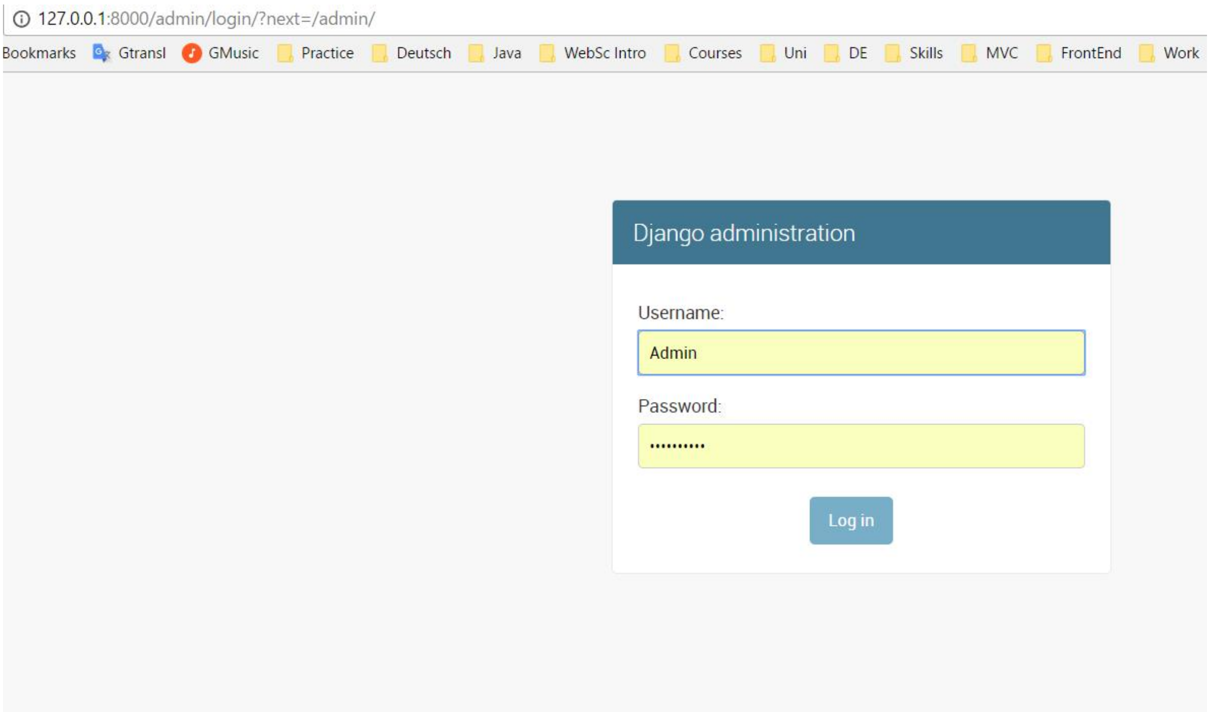

Figure 4.12 Admin login

Admin user account could be created with Power Shell scripting executing the required commands on manage.py file of the Django solution.

Admin site provides the possibility for admin user to create users and groups (roles or permissions - we did not use Groups in our project, all users have equal permissions).

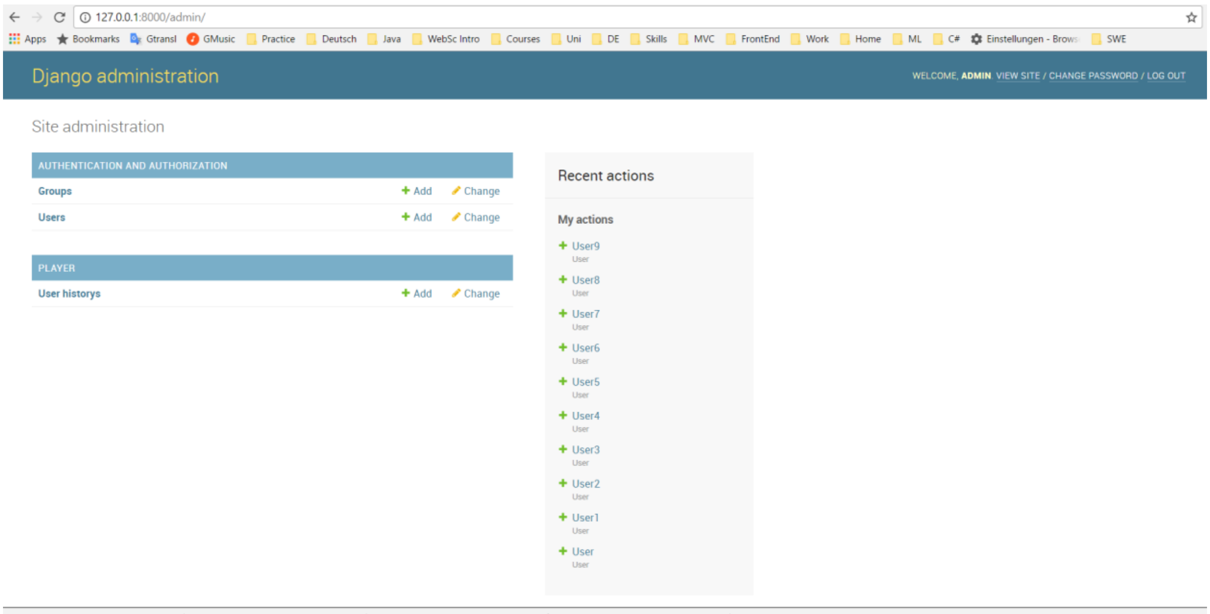

Figure 4.13 Admin page

User information is stored in the database on the table auth_user, that is one of the system tables, populated automatically by the Django framework. The table keeps the encrypted user passwords, information concerning the last login date and user profiles.

UserId are integers, generated automatically on the moment of the new user creation. In the project UserIds fields are used as foreign keys to reference the corresponding users in such tables as player_userhistory, player_usersim, player_usersvideosim.
Django server is configured to redirect site visitors to the login page if the request is not authenticated. It means that whatever link the visitor wants to navigate, login page will be displayed first if the current user is not log in yet. This setup is written on the class PlayerIndexView of the file ~/player/views.py.
In case of the successful login, when the entered username and password passed the validation towards database values, user will be automatically redirected to the index page of the web applicaton ~/player/.

4.2.5. Main Page

4.2.5.1. Design

The main page is the first page the user sees after he logged in. The interface of the main page is divided into two parts: The header (top navigation bar) and body. The main page shows some video suggestions related to the user's interests and based on recommended videos of YouTube. It is categorized in five different groups (top rated, last watched, recommended and test videos ...) and the videos are displayed as thumbnails. Each thumbnail corresponds to the initial image of a video. A hover effect enables the start of a video by looking on the thumbnail of a video in GTW browser. In a standard browser it is possible to select a video by clicking on the respective image.

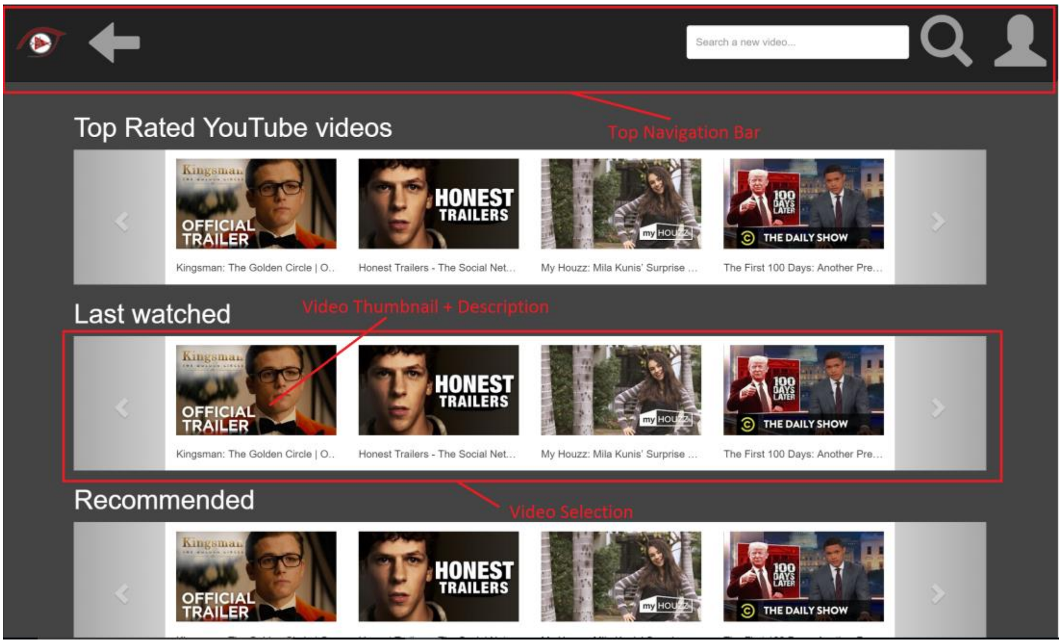

Figure 4.14 Main page

4.2.5.2. Functionality

Class PlayerIndexView of the file player/views.py provides method to pass to the Main page lists of videos for each section.
2 lists of videos are populated and returned to the view:

- top-rated YouTube videos
- last-watched videos of the current authenticated user.

Top-rated videos are retrieved directly from the YouTube API. We retrieve 10 top-rated videos, however the number could be adjusted, as it is one of the parameter of the web request to YouTube web API:
https://www.googleapis.com/youtube/v3/videos?chart=mostPopular&part=snippet&type=video&maxResults=10&key=... 10 last watched videos are populated in 2 steps:

1) from the database table player_userhistory define the records of the last 10 videoIds, that the current user watched
2) for each videoId from the previous step define video information for the view from the YouTube API:

https://www.googleapis.com/youtube/v3/videos?part=snippet,contentDetails&id=" + str(videoid) + "&key=...

4.2.6. Search Page

4.2.6.1. Design

The interface of the search page is divided into two parts: The header (top navigation bar) and body. It enables the user to search for videos via the search text field. When entering a set of keywords into the search field, they are sent to the youtube database, which returns a list of corresponding search results. This list consists of eight videos, which are sorted descending by views and popularity. This means the videos with the most views is played on top. Each list entry consists of a thumbnail, the title, description, views, average rating, duration and channel of a video. By looking on the play button or the thumbnail the player page opens.

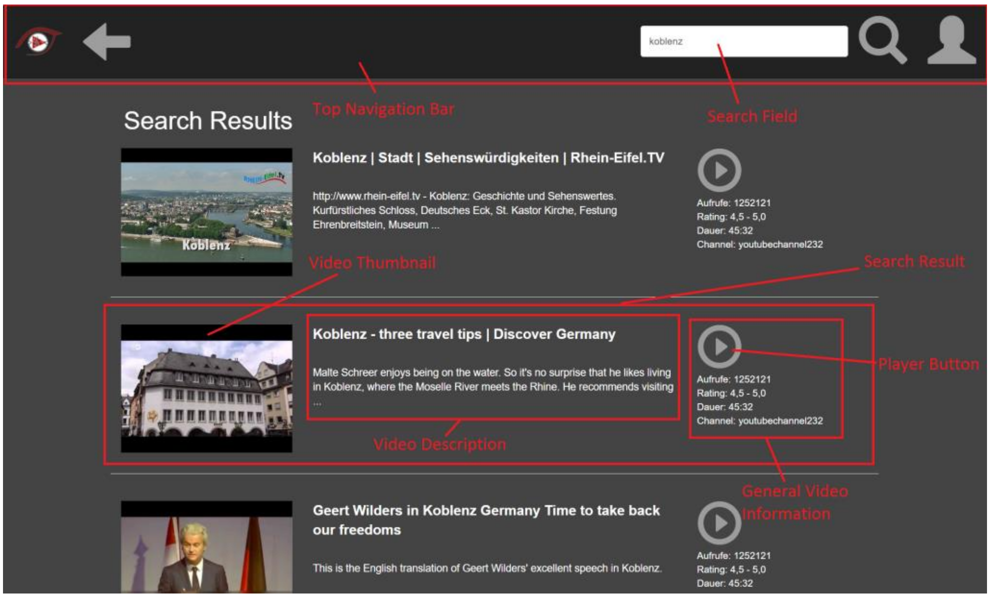

Figure 4.15 Search page

4.2.6.2. Functionality

The class SearchView (player/views.py) provides methods that retrieve result set of videos that corresponds to the search keywords entered by the user.

The search result set of videos constitutes of the mixture of 2 videos sets:

1) 25 videos retrieved from the YouTube API by calling API function, that returns similar videos to the input keyword, that is one of the parameters of the function: **https://www.googleapis.com/youtube/v3/search?part=snippet&&maxResults=25&type=video&q="** + input + **"&key=...**

2) YouTube results are mixed with the videos that are found from the database based on the user similarities. The idea of the approach is to define similarity of the current users with the stored in the database users and add to the search results videos that similar users already watched and rated them high. Besides these videos should also corresponds to the search keyword.

The steps of mixing videos from Youtube and our database are described following:

I. Relevant users who have similarity scores with the current user in the user-similarity table UsersSim are retrieved ascendingly.

II. Video histories of relevant users are retrieved from history-user table UserHistory, as well as the maximum ratings each user gave to the videos .

III. Rated-similarity scores are ratings of watched videos multiplied with users' similarity scores.

IV. From the watched video history table History_videos, we use similarity methods to get the relevant videos by the search keywords. It returns top 10 relevant watched videos. The similarity methods will be described later in this part.

V. Calculate the total rated-similarity scores of each top 10 videos by summing up rated-similarity scores of the video all users watched.

VI. Return these videos by descending order as the search-history result.

VII. The Youtube search result is valued from 5 descendingly. If there are search-history result in the Youtube search videos, those videos are weighted more.

VIII. The descending ordered result of last step is presented as the final result of our method.

Similarity algorithm in search session is cosine TF-IDF similarity. We use nltk (Natural Language Toolkit) to process the text, sklearn to transform language to statistical form for further analysis. The steps are described following:

I. Handle the text of both video tags and search input: lowercase and delete all punctuation and stopwords such as “the”,”a” and “are”.

II. Tokenize and stem the text to separate sentences into words and remove morphological affixes from words.

III. Vectorize and transform the words into matrix using TF-IDF algorithmus TfidfVectorizer and fit_transform.

IV. Calculate the cosine similarity between tags of videos and search input.

V. Return the 10 videos from highest to lowest scores.

4.2.7. Player Page

4.2.7.1. Design

The player page is the core of the web application. Every time when a user selects a video this page appears and the video can be played. The interface of the player page is divided into three parts: The header (top navigation bar), body and footer (bottom navigation bar). Below the header, a video and a playlist beside it can be seen. If a user chooses another video with mouse or gaze, the web page refreshes and the video is displayed on the right side. Under the main video you can find some additional information to it (e. g. rating and the current emotional state of the viewer). The footer serves a user to control the video. Therefore it consists of various buttons, which have been described in section Bottom Navigation Bar (displayed on the player page). If a user will activate the full screen mode the video size will maximize and the footer is still visible. So the user still has the control over the video. In addition to the bottom navigation bar a small grey rectangle with an arrow inside appears in the right side over the footer. By looking at this rectangle the bottom navigation bar will be closed so the user can fully enjoy the video.

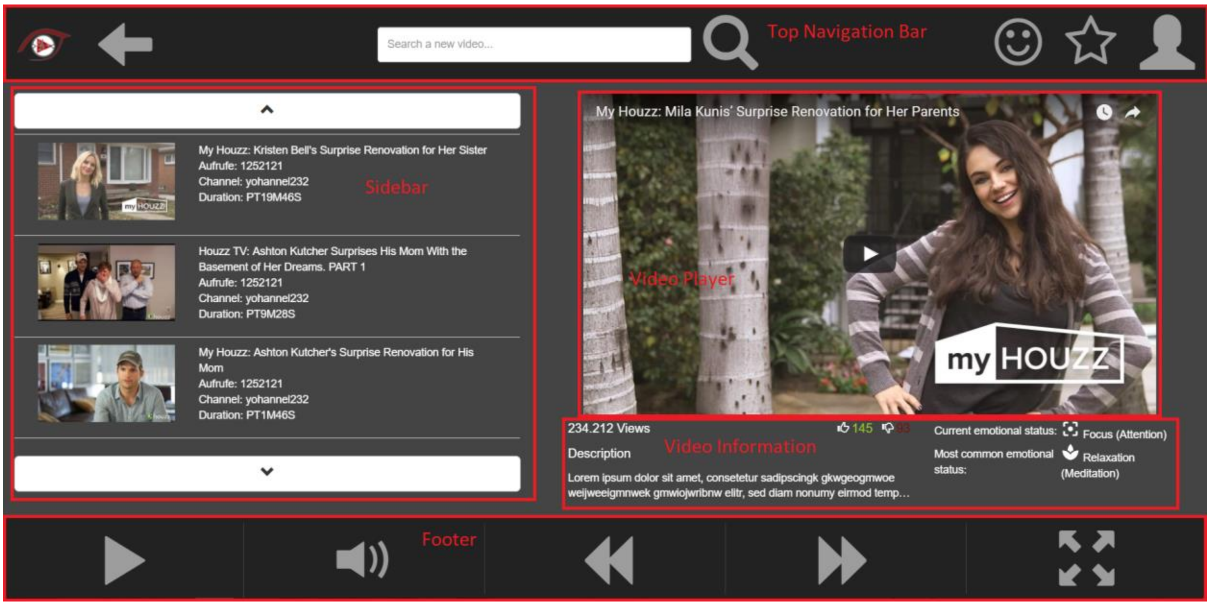

Figure 4.16 Player Page

4.2.7.2. Functionality

Whenever a Video starts playing, the recording of the gaze-data automatically starts. The gaze-data is pre-processed. The result of the pre-processing is a list of fixation-points. After each video, the fixation-points are uploaded to a database and also stored as a pickle-dump in a file. The primary keys for the database are the user-ID and the video-ID. If there are entries for this user-ID and video-ID in the database already, the fixations are not uploaded to the database. User-ID and video-ID are also used as filename for the pickle-dumps. The pickle-dumps are being created as a backup, in case there are problems with the database.

4.2.8. Special Functions

4.2.8.1. Browser

The web application can be used in a standard browser (e. g. Chrome Web Browser) or in GTW browser. In the standard browser the user can use the mouse for clicking on elements and the mouse wheel for scrolling up and down. In GTW browser the scrolling function is already given. That means that a user can scroll up and down only with his eyes and the scrolling with the mouse wheel is not necessary in GTW browser. Furthermore the mouse click is replaced by another technique - the hover effect. This effect is described in Hover-Effect section.

4.2.8.2. Hover-Effect

As far as there is no actual click possibility with the eye tracking devices, we had to simulate the click function with hover effect. Therefore, we used the approach of Greek partner who has developed an interactive Web page, which is directly controllable with gaze[12]. We utilized standard mouse over events for triggering dwell time based elements on the Web page. Therefore, we have implemented a class function which allows us to simulate the click process on any element when a user look at an element for a specific amount of time. This time is defined as a constant variable with the value of 1.5 seconds in JavaScript which can be easily changed to any other desirable value. When the mouse is hovering over a triggerable button, the dwell time is started and a visualization queues the remaining time to the user.

In order to indicate this functionality to user so that users know what elements are clickable through only staring at them, we have also added some visualised effect to the mentioned elements. These effects could be in different forms. For instance, a transparent overlay effect will appear whenever an user stare at the buttons on the toolbars in player page or a progress bar will appear under the image thumbnails in case of selecting items in the index or the search page.

Using mouse over let one control the page in standard Web browsers with mouse and in GTW browser with gaze, as long as "mouse follows gaze" is activated.

4.2.8.3. Star Rating

The used star rating enables each user the opportunity to rate each video once. The collected information will be stored in a database. With this information it is possible to calculate an average of popularity. A user can either give minimum one star to symbolize a poor video or maximum five stars to symbolize an excellent video. The following image shows a bootstrap modal which opens when a user selects the star in the top navigation bar. This popup shows the five-star-rating, the calculated average of the previous rates and an overview of how many people give one, two, three, four or five stars.

Figure 4.17 Rating menu

[12] https://github.com/MAMEM/prototype-interfaces-training

4.2.8.4. Functionality

In order to make the rating functionality work "on the fly", Ajax technology was used.
So once the user clicks on any star, the Ajax call triggers the backend method from the views.py file and passes to it the rating value, and videoId that was rated. For this purpose, the corresponding backend method was first registered as the url in the urls.py file, and as a result, it could be launched from the external Ajax calls. After parameters rating value and videoId were received from the user interaction with the popup frame, they are stored in the database.
The rating popup contains also the statistical overview of the rating values for this particular video. So after storing the new rating value in the database, all statistical values are recalculated and returned to the frontend as the json data. So the frontend js function could parse the returned json and update the popup window with the new statistics just after user clicked on the star.
Backend methods that implement processing of the rating data and calculating statistics are located on the file player/helpers/viewmodels.py. The method rating() from the views.py is registered as the separate url to trigger.

4.2.8.5. Full Screen Mode

As mentioned above we used the YouTube API. Though this API includes an own method for the full screen mode, it could not be used for our web application, because the YouTube API presumed the usage of the established full screen mode which YouTube used also. This mode includes tiny buttons which were too small to select them with gaze only. Therefore it was not possible to use it. Consequently it was necessary to implement an own full screen mode with a second navigation bar which consists of the bigger buttons for the navigation of a video instead of the smaller ones of the popular full screen mode of YouTube.

4.3. Calculation of User Similarity based on Eye Tracking data

In this research project, an approach is presented on how similarities of users can be determined by gaze data. A method has been developed to convert captured gaze data into fixations. Afterwards, these fixations of the individual users are compared with each other, in order to make statements about their similarity.
So far, heat maps are mostly used for research in the literature to sum up the viewpoints of individuals and to figure out which objects were particularly watched in pictures or web pages. A common use case is the Usability of web pages and to track user's attentions.[13] [14] [15] [16] [17] [18]

---

13 Granka, Laura A., Thorsten Joachims, and Geri Gay. "Eye-tracking analysis of user behavior in WWW search."
14 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.

Our approach deals with the similarity of users based eye tracking data, while watching video trailer. A method consisting of 6 steps is applied and will be described individually hereinafter. The complete graphic can be found in Figure 4.31.

Figure 4.18 Overview of the general approach: Calculate user similarity based on eye tracking data

Figure 4.18 Overview of the general approach: Calculate user similarity based on eye tracking dataFigure 4.18 shows the general idea of this method. The idea is to figure out whether persons who have similarities in gaze data also show similarities in film preferences. If there is a connection, this information can be useful, for example, to get a better individual recommendation list for videos.

<u>Step 1:</u> Collect Gaze data of User

For our analysis we record the gaze points from the participant while watching a video to detect the raw data of gaze points. While a Video is played, the Gaze-Stream of the user will be recorded. As soon as the user has finished the full Video the Gaze stream will be locally stored for post processing. The Gaze stream contains every gaze position with each timestamp.

Figure 4.19 Step 1 collecting the gaze data for different Videos

[15] Xu, Songhua, Hao Jiang, and Francis Lau. "Personalized online document, image and video recommendation via commodity eye-tracking." *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 2008.
[16] Al-Rahayfeh, A. M. E. R., and M. I. A. D. Faezipour. "Eye tracking and head movement detection: A state-of-art survey." *IEEE journal of translational engineering in health and medicine* 1 (2013): 2100212-2100212.
[17] Jacob, R. J., and Keith S. Karn. "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises." *Mind* 2.3 (2003): 4.
[18] Jacob, Robert JK. "Eye tracking in advanced interface design." *Virtual environments and advanced interface design* (1995): 258-288.

Step 2: Calculate Fixations

To get more significant information about the gaze data we calculate fixations. '**Fixations** are eye movements that stabilize the retina over a stationary object of interest.'

The goal of eye movement signal analysis is to **characterize the signal** in terms of salient eye movements, i.e., **saccades and fixations** (and possibly smooth pursuits). Figure 4.20 shows an occurring fixation and displays the meaning of a saccade.

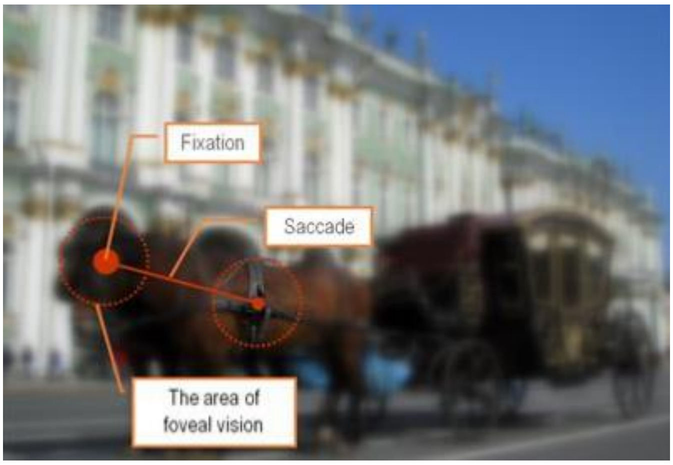

Figure 4.20 Pictures shows difference between fixation and saccade[19]

Typically, the analysis task is to **locate regions** where the signal average changes abruptly indicating the end of a fixation and the onset of a saccade and then again assumes a stationary characteristic indicating the beginning of a new fixation. [20]

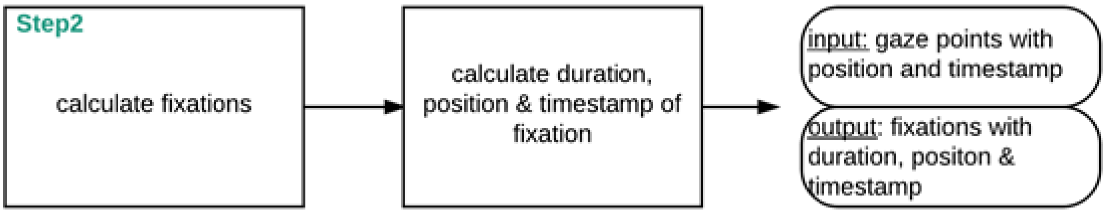

Figure 4.21 Step 2 Transformation of raw data to fixation

To identify fixation is a common method to reduce the complexity of the eye tracking data. While retaining only the most significant characteristics for the purpose of understanding. [21]

We implemented a simplified approach, which is shown in pseudocode in Figure 4.22, where *max_dist* is the maximum allowed Euclidean distance between two gaze points in a fixation it was set

[19] http://eyetracking.ch/wissen/was-ist-eye-tracking/ (november 2016)

[20] Duchowski, Andrew. *Eye tracking methodology: Theory and practice*. Vol. 373. Springer Science & Business Media, 2007.

[21] Salvucci, Dario D., and Joseph H. Goldberg. "Identifying fixations and saccades in eye-tracking protocols." *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, 2000.

to 50px and where *min_dur* is the minimum duration a set of gaze points needs to count as fixation, it was set to 200ms. 200ms duration for fixations was suggested by several papers whereas the maximum distance of 50 pixels was determined by our own experiments. [22] [23] [24]

```
//Pseudo code:

For all points:
If (dist( avg(gazeList), new_point) <= max_dist) {
    add point to gazeList }
Else {
    If (len(gazeList) >= min_dur){
            add fixation(gazeList)  to fixList}
    clear(gazeList);
}
return fixList;
```

Figure 4.22 Pseudo Code for transformation of raw data to fixation

With this information we defined a number of Areas of Interests (AoI) .[25]

In our approach the Input data will be deleted and only the output data is stored. This reduces the needed storage for the data base.

```
var fixList = [{
    'x' : 546.5245,
    'y' : 325.4351,
    'start' : 0.3245,
    'end': 0.2497
    },
    {...}
];
```

Figure 4.23 Representation of the fixations in a list of json objects

Fixations are stored in a List with several JSON objects, comparable to Figure 4.23), including the following information:

- x- and y-position of fixation
- start and end time in milliseconds
- Thus the duration of the fixation can be calculated with (end time - start time).

[22] Blascheck, Tanja, et al. "State-of-the-art of visualization for eye tracking data." *Proceedings of EuroVis*. Vol. 2014. 2014.

[23] Nyström, Marcus, and Kenneth Holmqvist. "An adaptive algorithm for fixation, saccade, and glissade detection in *Eye tracking* data." *Behavior research methods* 42.1 (2010): 188-204.

[24] Duchowski, Andrew. *Eye tracking methodology: Theory and practice*. Vol. 373. Springer Science & Business Media, 2007.

[25] Duchowski, Andrew. *Eye tracking methodology: Theory and practice*. Vol. 373. Springer Science & Business Media, 2007.

<u>Step 3:</u> Calculate common fixations

Based on the stored fixation data of the users we can calculate, if users have common fixations. Location and time will be considered.

The fixation lists of two users act as input to calculate common fixations. A **common fixation** occurs when two users simultaneously look at a similar position in the video. A similar position is given, if the centres of the fixation are not more than *50 px* apart. Comparable Figure 4.24, which depicts the definition.

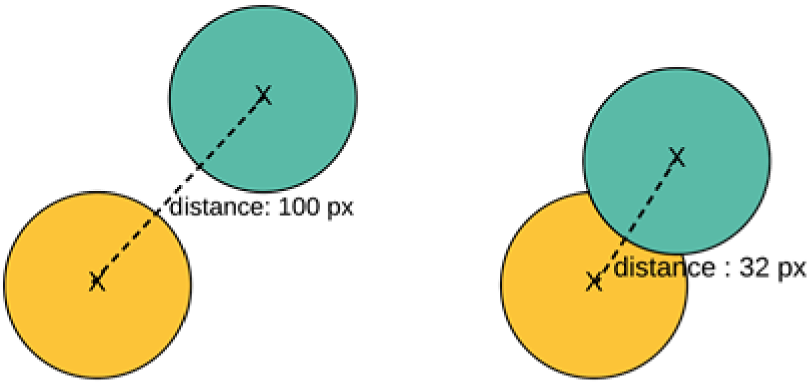

Figure 4.24 Common fixation

Common fixation is detected, if the distance between the two fixation center points (X) is less than *50px*. On the left side two fixations appear at the same timestamp, but the distance is more than the threshold of *50 px*. Instead on the right side a real common fixation is shown.

It is possible and common that only a part of a fixation of user A overlaps with the corresponding fixation of user B. The grey bars indicate the time period of the fixation, the first letter indicates the user (***U1*** and ***U2***), and the second letter indicates whether it is a start- ***(s)*** or end end-point ***(e)***. The blue colour visualizes the overlapping fixation time.

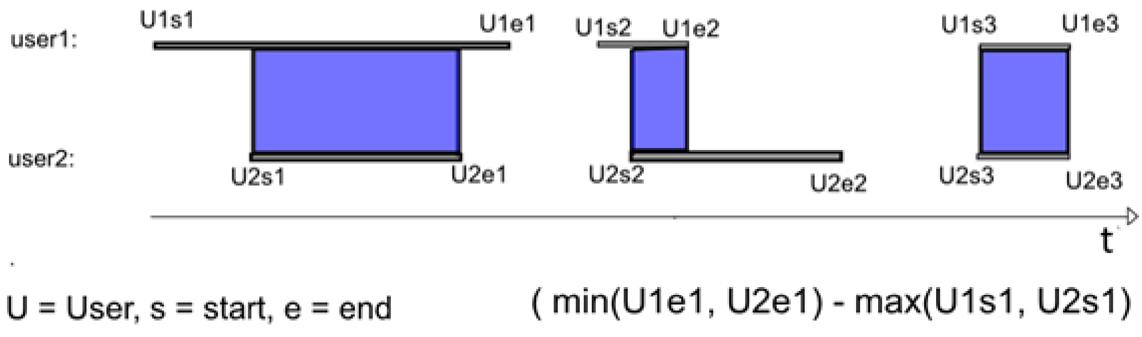

Figure 4.25 Common Fixation of two user (U1,U2) over time (t).
Only time is considered.

The formula:

$$\min(U1e1, U2e1) - \max(U1s1, U2s1)$$

calculates whether there is a temporal overlap of the fixation, afterwards the locality of the fixation is considered. If the distance between the two fixation centre points is less than *50px,* this indicated a common fixation.

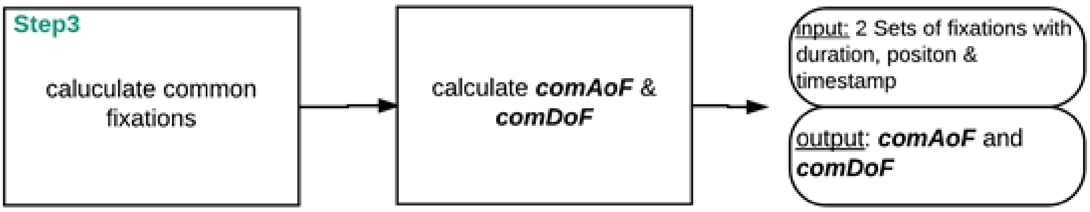

Figure 4.26 Step 3 Calculation of common Fixations for one Video

If all fixations are calculated, the amount is stored in the ***comAoF*** (**com**mon **A**mount **o**f **F**ixations) variable. The duration of all overlaps is saved in the variable ***comDoF*** (**com**mon **D**uration **o**f **F**ixations). Both variables are needed for the further calculation of the similarity of two users by the fixation and will be stored for every user pair that watched the same video. An overview of this step is given by Figure 4.26.

<u>Step 4:</u> Calculate arithmetic mean of accruing fixations for each Video

Each video is viewed individually in this step. All existing data from *N*-different users are needed to calculate the average fixation number ***avAoF*** ***(av**erage**A**mount**o**f**F**ixation) and fixation time ***avDoF*** ***(av**erage**D**uration**o**f**F**ixation). Figure 4.27 visualises this method.

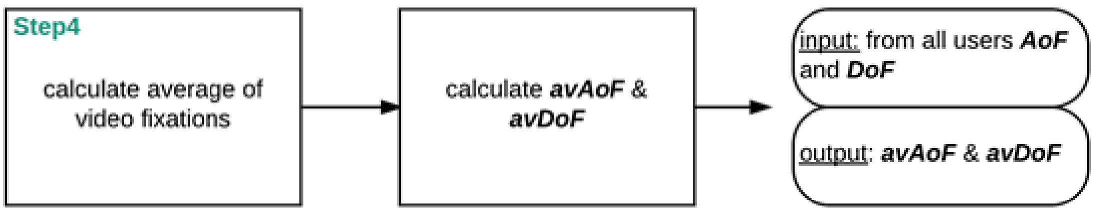

Figure 4.27 Step 4 Calculation of arithmetic mean of Amount and Duration of Fixations

The arithmetic mean of the amount and duration of the fixations is calculated by iterating over all data and adding the resulting fixations to ***totalAoF*** and the associated duration to ***totalDoF***. Afterwards each value is divided by the amount of users, who have watched this video. The results are further processed in step 5. They determine how the similarity of the users in the individual videos is weighted. To understand step 4 more precisely, Figure 4.28 should be considered.

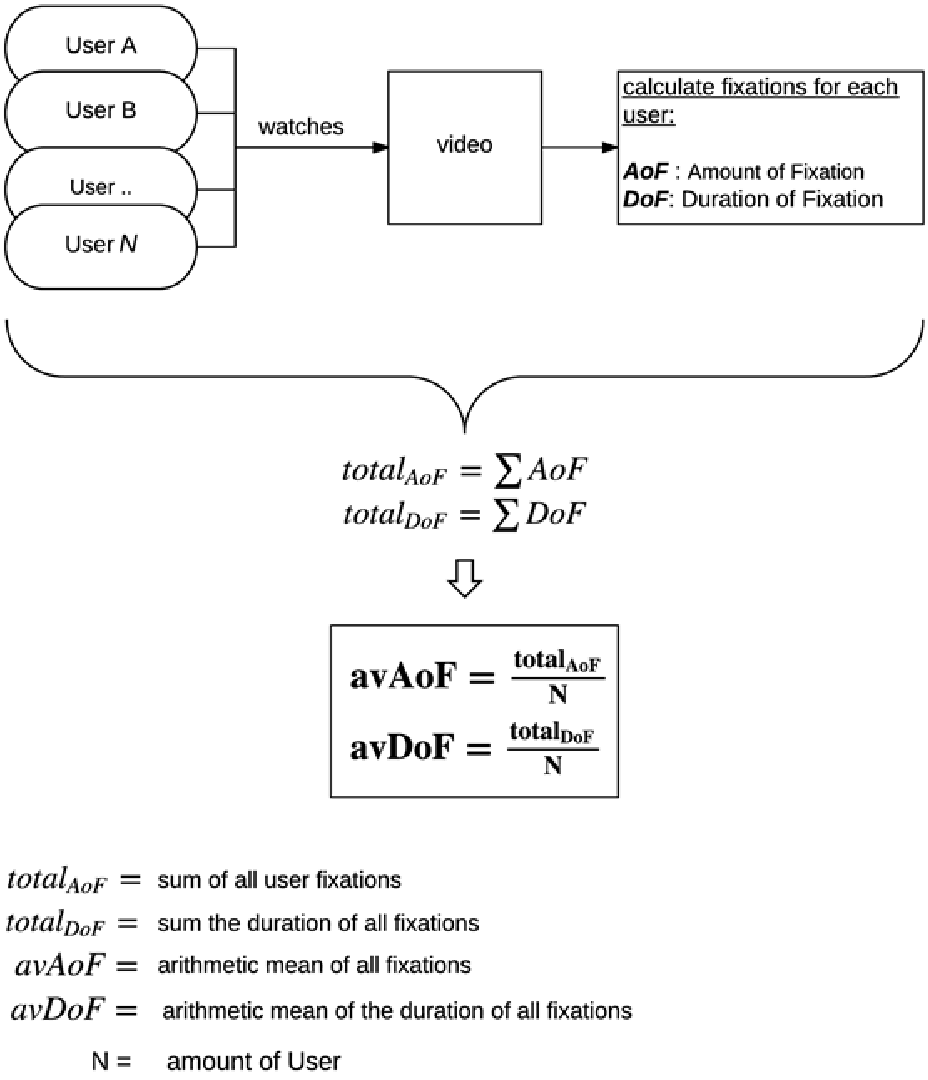

Figure 4.28 Step 4 Calculate the average of fixation per person and the duration time of fixations

<u>Step 5</u>: Calculation of a Video similarity

To calculate the similarity of two users we apply a scale to the common fixations for each video the both of them watched.

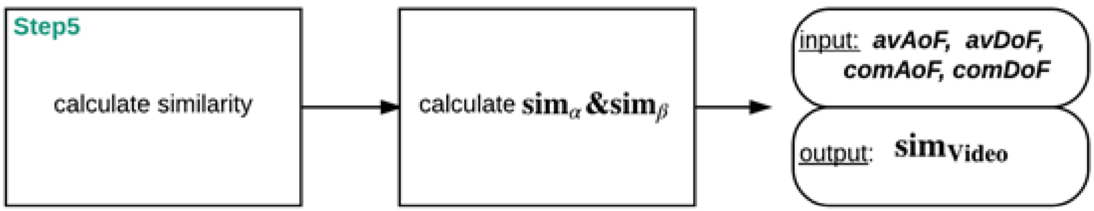

Figure 4.29 Step 5 Calculaton of Video similarity for two users

The scale maps the common fixations to the values 0, 0.25, 0.5, 0.75 and 1, whereas 0 is the least and 1 the highest possible similarity.
This is applied to the common fixation count and the common fixation duration separately,
by calculating the average of the two resulting value. It is based on the average fixation duration (**avFoD**) and amount (**avAoF**) respectively for the video in question and goes as follows:

sim_{α} | sim_{β}	0%	1 - 24 %	25 - 49 %	50 - 74 %	75 - 100 %
sim_{Video}	0	0.25	0.5	0.75	1

Table 4.1 Division of the weight of similarity

$\boldsymbol{sim_{\alpha}}$ = Describes the similarity value for the number of fixations that occur. This value is calculated with the average amount of fixation (*avAoF*) of the Video.
100% correspond to the average fixation of individual users (*avAoF*)
50% are accordingly = $(\frac{\boldsymbol{avAoF}}{\boldsymbol{2}})$, 25% = $(\frac{\boldsymbol{avAoF}}{\boldsymbol{4}})$ ..

$\boldsymbol{sim_{\beta}}$ = Describes the similarity value for the duration of fixations that occur
This value is calculated with the average duration of fixation (*avDoF*) of the Video.
100% correspond to the average duration of fixations of individual users (*avDoF*)
50% are accordingly = $(\frac{\boldsymbol{avDoF}}{\boldsymbol{2}})$, 25% = $(\frac{\boldsymbol{avDoF}}{\boldsymbol{4}})$ ..

The values were chosen because each video can have a different number of fixations and duration. Therefore, each video is viewed individually, in order to make a vague statement about the similarity of two users. If there are common fixations with two users as often as the average value for a person, then an enormous similarity is to be inferred.

By adding $\boldsymbol{sim_{\alpha}}$ and $\boldsymbol{sim_{\beta}}$ and dividing these value through 2 we get the user similarity regarding the individual video ($\boldsymbol{sim_{Video}}$).

$$sim_{Video} = \frac{sim_{\alpha} + sim_{\beta}}{2}$$

This method is applied to all shared videos and stores a value for each video that contains the similarity to that video.

<u>Step 6:</u> Calculate the overall similarity

Subsequently, the total similarity of the users can be determined by means of gaze data. To this end, all existing video similarity values are added and divided by the number of viewed videos. Figure 4.30 visualize this process. The value is between 0 and 1 and indicates the similarity, whereas 0 is the least and 1 the highest possible similarity.

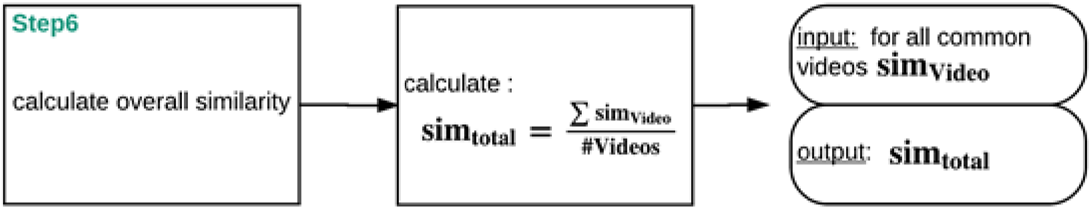

Figure 4.30 Step 6 Calculaton of the overall similarity for two users, based on each Video similarity

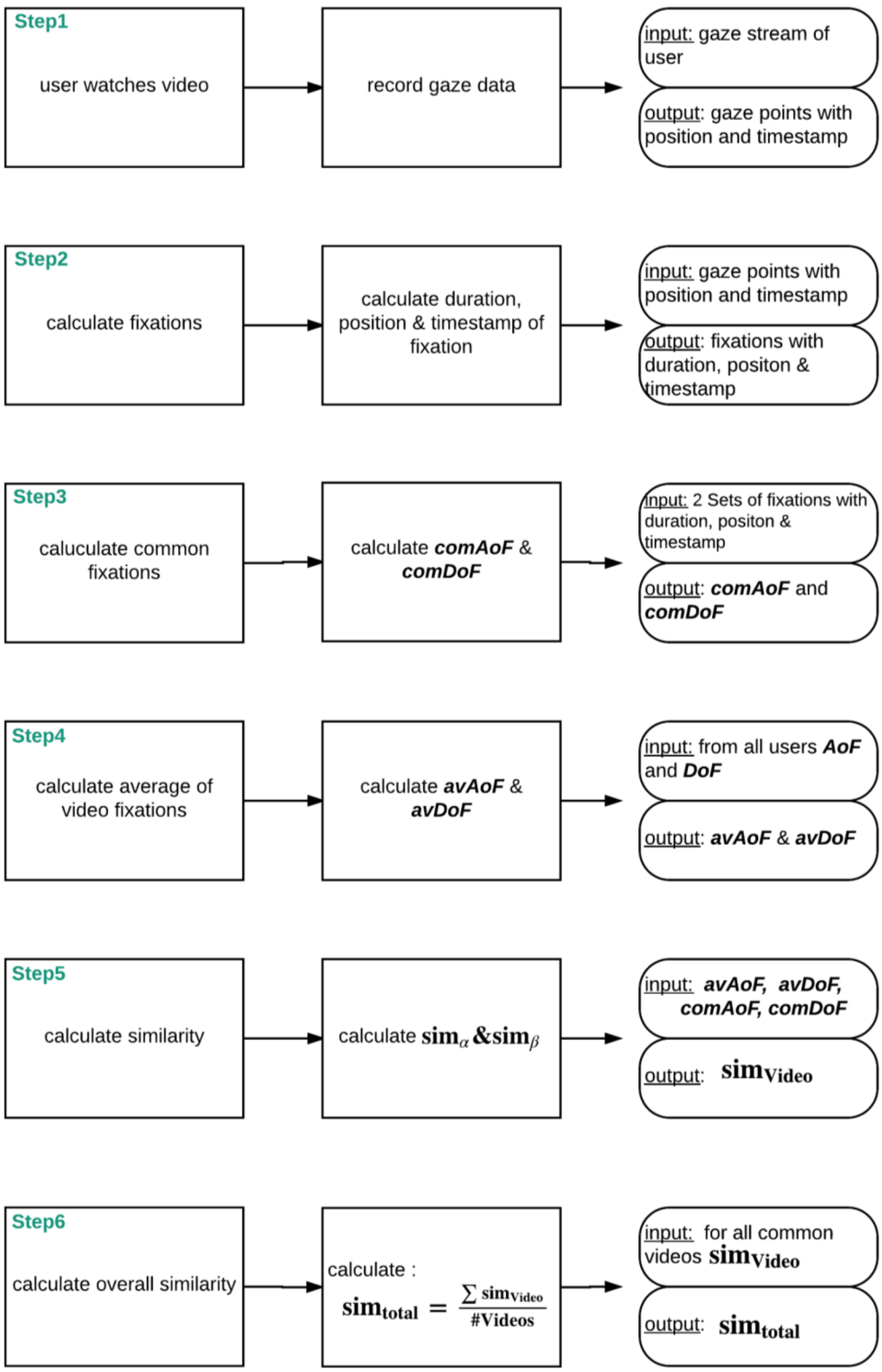

Figure 4.31 Summary of all 6 individual steps for calculation of user similarity based on gaze data

Legend:

comAoF	= **com**mon **A**mount **o**f **F**ixation	$\mathbf{sim}_{\alpha}$	= similarity value for Amount of Fixations
comDoF	= **com**mon **D**uration **o**f **F**ixation	$\mathbf{sim}_{\beta}$	= similarity value for Duraton of Fixations
avAoF	= **av**eraged **A**mount **o**f **F**ixation	$\mathbf{sim}_{\mathbf{Video}}$	= averaged similarity value for <u>one</u> Video
avDoF	= **av**eraged **D**uration **o**f **F**ixation	$\mathbf{sim}_{\mathbf{total}}$	= total similarity value for two user

4.4. Visualisation of Fixation Points

The visualization of the individual fixation points of the user is another feature in '**EyeTube**'. It allows the visual comparison of the fixation points. The user can display the data of the last 5 persons with their different fixation points for each video, which has already stored data. An individual color is assigned to each person. For example: User A is represented by the colour blue, user B by the color red. Every time user A has a Fixation a blue circle will appear in the appropriate region of the video, the same applies to user B with a red circle.

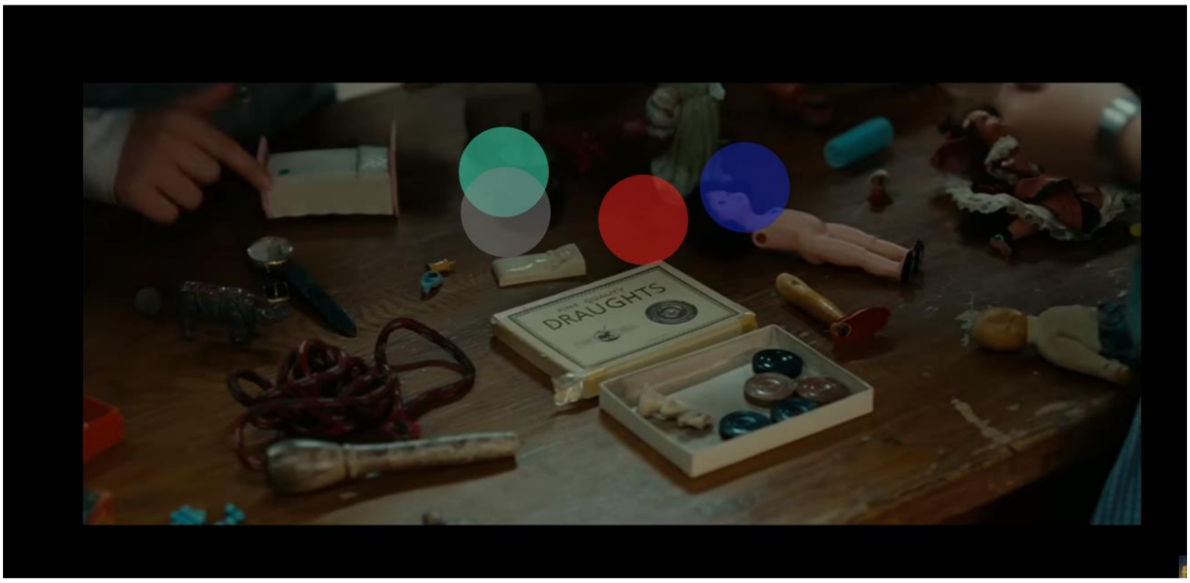

Figure 4.32 Visualization tool 1

Background = 'Trailer: Never let me go' [26]

Is the 'Development Mode' activated, it is possible to start the visualisation tool. By activating this option you can see the visualized fixations of the last 5 Users. Used for analysis purpose and to clarify what is meant by an ***overlapping fixation*** of two users, the feature is realized with HTML and JavaScript.

The Visualisation is realised by different layers for the JavaScript canvas.

In the background-layer the original video is displayed. For each user a new layer with a canvas is added. The fixations are visualised as points around the central point of the fixation with a 50px radius. Users are represented by a unique colour.

For each user a fixation-list exists, comparable to Figure 4.32. Every fixation is represented in a Json-Object inside this list. A fixation has a position (x- and y-coordinate) and a start- and end-time in milliseconds. With these timestamps the duration of the fixation can be calculated (end time - start

[26] Video:: https://www.youtube.com/watch?v=sXiRZhDEo8A

time). The JavaScript intern Interval-Method is used to automatically call the drawing method for each fixation. This method draws the coloured circle as soon as a fixation appears in the correct canvas and calls the *clear()*-method to erase the circle as soon, as the duration of the fixation is over.

The visualisation tool underlines the meaning of the overlapping fixation of users. Comparable to Figure 4.32. It displays a screenshot of a full screen visualisation of the video ‘Never let me go’ with the fixations of 5 users. The screenshot displays four different colours, because only these persons had a fixation at this time. The grey and the light blue circles have a common fixation at this timestamp. Recognizable by the overlapping coloured circles.

Figure 4.33 Visualization tool 2

In Figure 4.33 all five User have a fixation at this time of the video. Each User has a common fixation to another. But the red user only has a common fixation with the blue one. Whereas the blue user has a common fixation with the red and the green user.

4.5. BCI and emotion detection

4.5.1. Technology overview

For subjective and complex products such as movies, music, news, user emotion plays surprising critical roles in the decision process. One of the modalities that can observe the brain activities and emotional reactions is electroencephalogram (EEG). EEG signals have characteristics to change each time (non-stationary) and random because of a very dynamic activity in the brain. Although the EEG signals are dynamic, but they can be classified based on certain characteristics, one of them through the power spectral analysis of several brain waves frequency range. The use of EEG data to recognize user emotion may contribute to building reliable recommender systems.

Specific emotional states, like mental stress, concentration, relaxation, fatigue, and cognitive increase activation in Delta (0.5-4 Hz), Theta (4-7 Hz), Alpha 1 and 2 (8-12 Hz), Beta 1 and 2 (12-30 Hz), Gamma (30-70 Hz) frequencies. The increase of frontal Beta-1 spectral power is associated with cognitive tasks demands and the decrease of Beta-1 power values reflects relaxation. Alpha is the dominant frequency in the human EEG and is generated in widespread areas of the cortex through corticocortical and thalamo-cortical interactions reflecting emotions.

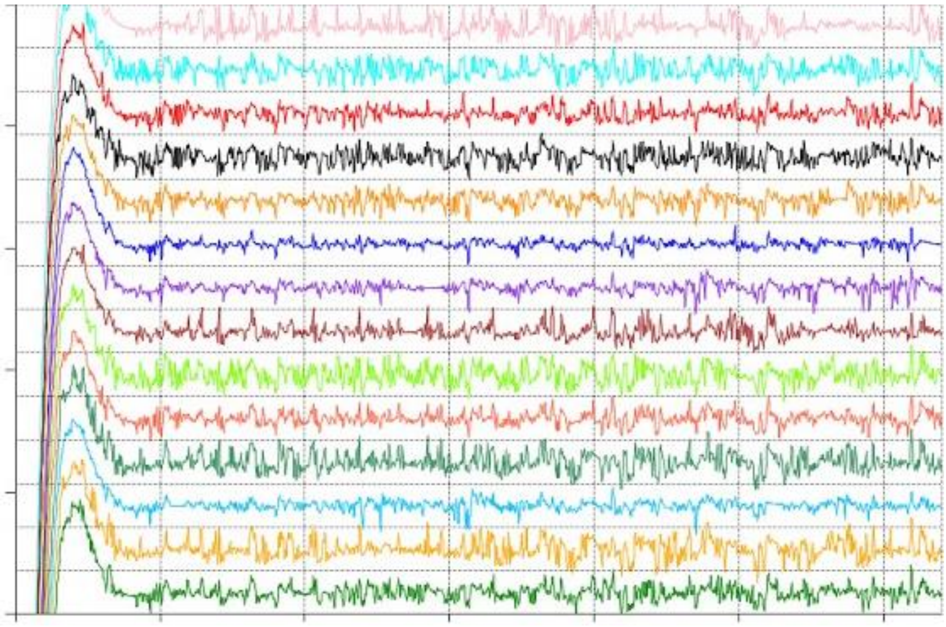

Figure 4.34 EEG visualization

As mentioned earlier, the EEG signals will be collected using a wireless EEG headset in real time, specifically the Emotiv EPOC wireless headset with a sampling frequency 128Hz. The headset has fourteen data collecting electrodes and two reference electrodes. Electrodes are placed approximately at the 10-20 locations AF3/4, F3/4, FC5/6, F7/8, T7/8, P7/8, and O1/2 as shown in Fig. X to interface with the Emotiv EPOC wireless headset. The system computes the relative power in two non-overlapping frequency bands (10-20Hz, and 20-30Hz) and generates rates from the computed values. Relative power is a simple measure that can readily be computed in real time. The EEG spectrum is known to depend on the mental state (e.g., relaxation, sleep). Alpha waves are typical for an alert, but relaxed mental state, thus high activity is associated with meditative state, visualization, and idleness. In contrast, beta activity is related to an active state of mind during intense focused mental activity, and high beta activity is associated with fear and anxiety.

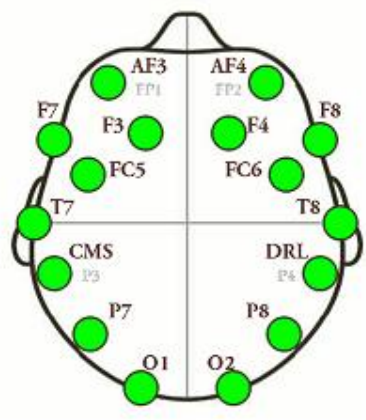

Figure 4.35 The EPOC electrode positions on the head approximate the international 10-20 system.

There are different emotion classifications proposed by researchers. In this research was used the model developed by Emotiv. Manufacturer provides five basic measures of mental performance, derived directly from the mental activity. Each measure is automatically scaled to suit the normal range and base level of each condition – the system learns the usual state and capabilities and provides an adjusted value showing the relative performance on each occasion, compared to the overall behavior.

The measures include:

Engagement: the level of immersion in the moment. A mixture of attention and concentration.

Focus: a measure of the fixed attention to one specific task. Focus measures the depth of attention as well as the frequency that the attention switches between tasks. A high level of task switching is an indication of poor focus and distraction. Focus is closely related to the Flow state.

Interest: the degree of attraction or aversion to the current activity. Low interest scores indicate a strong aversion to the task, high interest indicates a strong affinity with the task while mid-range scores indicate user neither likes nor dislikes the activity. Interest is related to the enjoyment of the current task.

Relaxation: a measure of the ability to switch off and allow to rest and recover from intense concentration.

Stress: a measure of the level of comfort with the current challenge. High stress can result from an inability to complete a difficult task, feeling overwhelmed and fearing negative consequences for failing to satisfy the task requirements. Generally a low to moderate level of Stress can improve productivity, whereas a higher level tends to be destructive and can have long term consequences for your health and wellbeing.

4.5.2. EmoEngine and Emotiv SDK

The Emotiv API is exposed as an ANSI C interface that is declared in 3 header files (edk.h, EmoStateDLL.h, edkErrorCode.h) and implemented in 2 Windows DLLs (edk.dll and edk_utils.dll). C or C++ applications that use the Emotiv API simply include edk.h and link with edk.dll.

There are three main categories of EmoEngine events that the application may handle:

Hardware-related events: Events that communicate when users connect or disconnect Emotiv input devices to the computer (e.g. EE_UserAdded).

New EmoState events: Events that communicate changes in the user’s facial, cognitive and emotional state. User can retrieve the updated EmoState by calling EE_EmoEngineEventGetEmoState(). (e.g. EE_EmoStateUpdated).

Suite-specific events: Events related to training and configuring the Mental Commands and Performance Metrics detection suites (e.g. EE_CognitivEvent).

Solution used in the research is based on provided by the EMOTIV emostate_logger.py code and data resolved from the Perfomance metrics interfaces.

4.5.3. Accuracy issues

According the Emotiv statements internal measures have accuracy between 65% and 100% depending on the emotion and the subject.

The output for each emotion is a floating point number between zero and one. Ideally they should be scaled based on historical patterns of each individual user, but in this case during each experiment would take a few hours for the system to settle down for a given subject. Self-scaling provides a useful within-subject scale but makes it very difficult to compare subjects.

Performance Metrics detections are the most heavily filtered and the most likely to be shut down temporarily by excess noise. There is no connection between facial expression information into the emotional detections implemented in this case. Lots of frowns would be associated with higher frustration levels.

4.5.4. Calculation of User Similarity based on emotions

Step 1: Detecting and logging emotions

For further user similarity measurement 6 emotional states are recorded during each session of video watching. While a video is played, the stream of the EEG data will be recorded. After the video is finished the stream will be locally stored for post processing. The final record contains the average of each emotion state. Storing and analysing raw EEG data proved to be inefficient in measuring user-similarity as well as hardware-consuming,

Step 2: Calculation of a Video similarity

Based on the stored emotion averages the similarity will be calculated. For calculations would be used Pearson correlation optimised for 6 dimensional items-sets (6 emotions.)

a; b - users

$r_{a,p}$ - emotion of user *a* for item *p*

p : set of items, rated both by *a* and *b*

Possible similarity values between 0 and 1

$$\boldsymbol{sim(a,b)} = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

This approach is usually used in collaborative filtering. Apart from that it was chosen for the following reasons:

- distribution of correlation between random vectors becomes narrowly focused around zero as the dimensionality is increased;
- the significance of small correlations increases with growing dimensionality;
- approach is effective for capturing the similarity of patterns of feature changes.

5. Evaluation

5.1. Introduction

A recommendation system based on similarity measurements has become one of the most used approaches to provide personalized services for users. The key of this approach is to find similar users or items using user-item rating matrix so that the system can show recommendations for users. However, most approaches related to this approach are based on similarity algorithms, such as cosine, Pearson correlation coefficient, and mean squared difference. These methods still need improvement, especially in the cold user conditions. The video player for YouTube, developed in this research, has functionality to collect additional data which can be used for a new user similarity models which respectively may be used to improve the recommendation performance comparing to case when only the ratings are available to calculate the similarities for each user. The proposed model considers the captured fixation points and emotions based on EEG. The results of this evaluation aim to explore how close the new similarity model will be to the rating-based model.

5.1.1. Goal of the evaluation

The goal of the evaluation process is to understand and assess the effectiveness of developed system, test the accuracy of used emotion recognition algorithm and compare three similarity measurements based on different types of data obtained from the users: rating feedback, fixations and emotions. Similarity measures based on the rating given by users will be used as a control data since for now this is the major method of similarity measurement used by online video-sharing/streaming services. The Evaluation methods that will be used are: questionnaires, observations and individual experiments.

5.1.2. Relevant points

Currently, most advanced methods of similarity measurements use not only the rating but also compare users by the videos they had or had not watched. This technology helps to make more efficient predictions but requires more users and bigger sets of watched videos. Considering the resource and time limitations, in this research lab it is impossible to implement this feature for new models. For this reason, to make the comparison fair, rating collected directly from YouTube cannot be used during the evaluation. Instead, users will rate all watched videos using alternative rating scale created inside the project and based on ratings given only by test subjects.

5.2. Formulation of the Hypotheses

Main Hypothesis

Similarity measurements based on the fixation points and emotions provide relevant similarity rates with accuracy close to similarity measurements based on rating. Rating-based similarity in this case is considered to be the control data.

Reasoning

If it proves effective to calculate user similarity based on gaze data and emotions it would be a base for a better video recommendation system.
Eye tracking is already being used to check user similarity.
Gaze data can give hints about the places, which are interesting for the people.
More or longer fixations show more interest of the user.

5.3. Experimental Set-Up

The experiment took place from 19.05.2017 to 26.05.2017 in the research laboratory of the working group WEST at the University of Koblenz-Landau, Campus Koblenz. A total number of 20 subjects were invited one after the other. Consequently it was not a group task. The duration of the experiment was approximately 30 minutes. At the beginning of the experiment the participants should sign an "Informed Consent Form", which was prepared before the experiment. A study coordinator (participant of the research lab/project lab) was in the same room all the time. This supervisor was about one meter away from the subject and explained the experiment and the tasks to be accomplished. In addition, he/she was always available to answer questions. The subject has sat in front of a second monitor. The eye tracker was attached to the screen and the BCI was placed on the head of the test subject.
All manipulations with the video player including switching between videos were done by the supervisor.

Overview

Goal of test studies is to record gaze- and EEG- data of users while they are watching videos. Further (not during the test) recorded data will be used for evaluate similarity of users and accuracy of recorded emotions caused by videos. During this evaluation all test subjects will also have to answer a set of questions using prepared Google Forms. Data from questionnaire will be used to analyse and compare the recorded data.

Average time for full setup completed by backend team member	ca. 10 minutes
Estimated time	ca. 20-30 minutes
Participants	supervisor and test subject
Supervisor	research lab participant, responsible for preparation of the test environment, assisting and leading the test subject and recording of the results.
Test subject	volunteer, participating in the experiment.

Table 5.1 Experiment Parameters

Computer

Computer used for the experiment is a laptop and the monitor connected to it. The first monitor displays the player interface, the second monitor displays the additional software. This would allow supervisor to control all software, devices' connection states and the record processes, not disturbing the test subject.

Devices

To insure better recognition:

- the curtains on the windows must be drawn;
- test subject must seat on a static chair and limit her/his movements/gesticulation;
- test subject should not talk while watching videos.

Software

Software includes:

- Python 3 environment
- the server of the player
- programs connected to the devices iViewRED and Emotiv ControlPanel
- LSL connections

Server

Console output of recording. It must be monitored during the whole experiment to ensure the correct recording.

iViewRED

iViewRED – software used to connect the eye-tracker to computer and do the calibration.

Figure 5.1 iViewRED

The distance and visibility of user's eyes can be controlled. Optimal distance should be 50-60 cm. Movement of test subject should be limited during the watch.

Calibration

Calibration must be done before starting the experiment and before showing each new video. During the calibration the test subject must look at the red dot, moving to five different parts of the screen. Calibration should be re-done if:

1. one eye was not detected in any of five cases or
2. the distance between two eye-traces or eyes-traces and red dot is two big in all five cases.

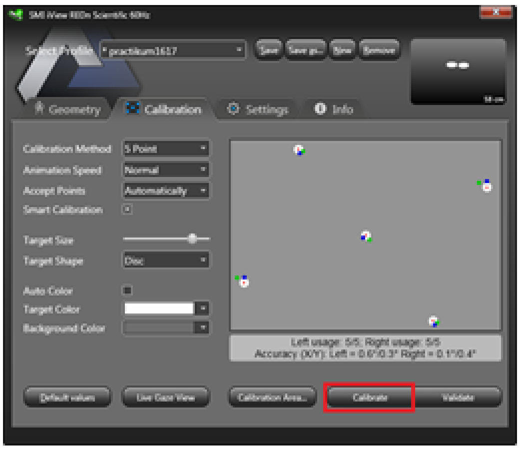

Figure 5.2 Example of good calibration

EmotivXavierControlpanel

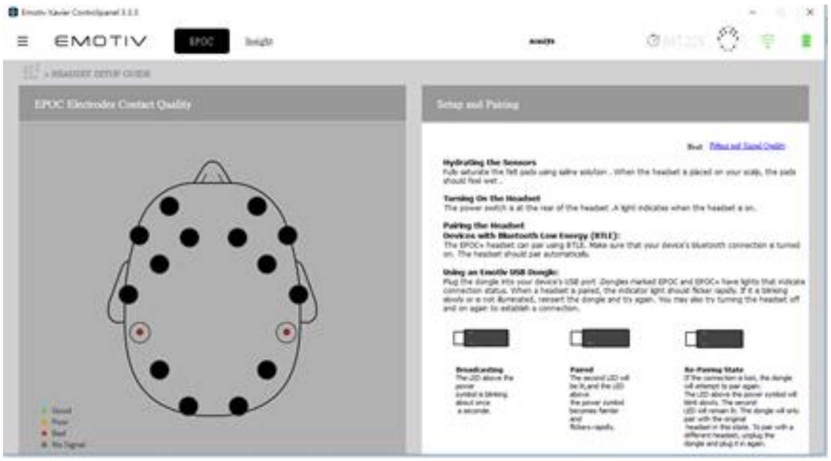

Figure 5.3 EmotivXavierControlpanel

EmotivXavierControlpanel – software used to connect the BCI to computer and do check the input signal quality. Considering that experiment was conducted using damaged connectors, up to 5 of them might not work or have worse signal quality.

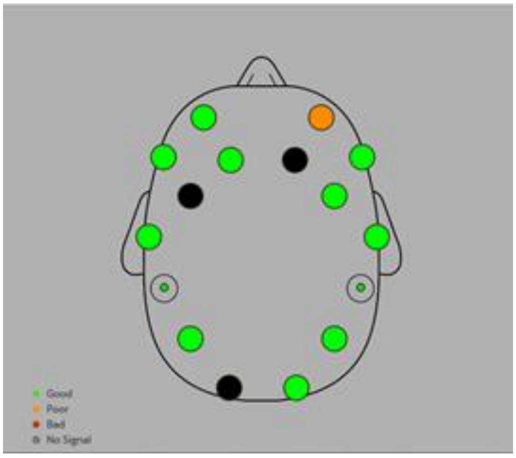

Figure 5.4 Acceptable input

Videos and Questionnaires

In the Bookmarks Bar of Chrome browser were created two folders containing links to videos (already opened in the player) and the questionnaires. Links to videos are named the following way:

- 1-Going in Style
- 2-Suicide Squad
- 3-Love is all you need?
- 4-Wolves
- 5-Lights out
- 6-Ouija

This enumeration should be used when defining the order of showing videos according to the Latin Square.

Recording control and check

In order to start the recording of gaze- and EEG- data only one page (one window, one tab) should be opened at the same time.

Video must be started only by pressing custom made buttons. Usage of built-in youtube buttons will not start the recording and will not finish it.

Video must be played from beginning to the end to insure the record starting and stopping correctly.

In order to insure successful record before starting next video supervisor must check the database.

5.3.1. Setup steps

1. Turn on the computer.
2. Switch to extended screens.
3. Draw the curtains on the windows.
4. Attach and connect the eye-tracker.
5. Launch the software (setup.bat or list).
6. Turn on the BCI and charge it if the battery is low.
7. Assemble BCI by fastening the connectors.
8. Clean the connectors.
9. Hydrate the connectors.
10. Define Test Subject’s number and the order of videos.
11. Open first video.
12. Open the questionnaires and enter the Test Subject’s number.
13. Open the database.

5.4. Realization

Pretest

Before running the experiment with the test subjects the setup was pretested by project participants. During the first iteration of pretest, detailed experiment steps and limitations caused by specific recording procedure were defined. Also several data storing errors were discovered.

During the second iteration of pretest participants insured that the data storing error was fixed and the overall environment functioned correctly.

Experiment steps

1	Put the BCI on a Test Subject. Explain rules (movement/talking while watching)
2	Do calibration for the Test Subject.
3	Test Subject answers Questionnaire 1.
4	Start 1st video.
5	Stop 1st video.
6	Test Subject answers Questionnaire 2 for 1st video.
7	Check the state of record.
8	Do calibration for the Test Subject.
9	Start 2nd video.
10	Stop 2nd video.
11	Test Subject answers Questionnaire 2 for 2nd video.
12	Check the state of record.
13	Do calibration for the Test Subject.
14	Start 3rd video.
15	Stop 3rd video.
16	Test Subject answers Questionnaire 2 for 3rd video.
17	Check the state of record.
18	Do calibration for the Test Subject.
19	Start 4th video.
20	Stop 4th video.
21	Test Subject answers Questionnaire 2 for 4th video.
22	Check the state of record.
23	Do calibration for the Test Subject.
24	Start 5th video.
25	Stop 5th video.
26	Test Subject answers Questionnaire 2 for 5th video.
27	Check the state of record.
28	Do calibration for the Test Subject.
29	Start 6th video.
30	Stop 6th video.
31	Test Subject answers Questionnaire 2 for 6th video.
32	Check the state of record.

Post-experiment steps

1	Clean and hydrate the connectors.
2	Close all software; turn off devices and the computer.
3	Hide devices to the cupboard, close the cupboard and the lab.

5.5. Latin Square Design

The Latin Square design is a counterbalanced measures design. The order in which treatments are given can actually affect the behaviour of the subjects or elicit a false response, due to fatigue or outside factors changing the behaviour of the subjects. To counteract this, we decided to use a counterbalanced design, which reduces the chances of the order of treatment or other factors adversely influencing the results. This design was also used to control the variation in the experiment.

In experimental design a Latin Square is an n x n array filled with n different symbols. For the Latin Square design there are two factors, that are divided into a tabular grid with the property that each row and each column receive each treatment exactly once. This design allows experiments with a relatively small number of runs. So it was possible to use it in our scenario, because our sample consisted of 20 subjects. One disadvantage was that the 6 x 6 Latin Square was only practicable for six people. That's why we used the Latin Square three times completely and started again with a new Latin Square at the last run.

Since we showed each participant (first factor) all six trailers (second factor), we have used a 6 x 6 Latin Square:

Number for the Latin Square	Name of the trailer
1	Going in Style
2	Suicide squad
3	Love is all you need?
4	Wolves
5	Lights out
6	Ouija

Table 5.3 Latin Square movie numbers

	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Subject 1	1	2	6	3	5	4
Subject 2	2	3	1	4	6	5
Subject 3	3	4	2	5	1	6
Subject 4	4	5	3	6	2	1
Subject 5	5	6	4	1	3	2
Subject 6	6	1	5	2	4	3

Table 5.4 Latin Square

In this example, subject 1 received the treatments levels in order 1 (Going in Style), 2 (Suicide squad), 3 (Love is all you need?), 4 (Wolves), 5 (Lights out) and 6 (Ouija) and subject 2 received the treatments levels in order 2 (Suicide squad), 3 (Love is all you need?), 1 (Going in Style), 4 (Wolves), 6 (Ouija) and 5 (Lights out).

5.5.1. Questionnaires

We used two different questionnaires - a demographic[27] and a post-task questionnaire[28] - which will be detailedly described in the next subsections. For the creation of these forms we used Google Forms[29]. The reason of this was that the group had one collaboration point where every member can access the files and results simultaneously. Another feature of this solution was that all responses to our questions were collected automatically and clearly in real-time. Both questionnaires contain, among other things, a text field for the user id. This was necessary in order to be able to assign the individual questionnaires to the recorded data and for the data evaluation.

5.5.1.1. Demographic Questionnaire

In general a demographic questionnaire contains questions about the sample. In this way, a meaningful data aggregate on the demographic characteristics of all participants could be created. In detail, the gender, age, level of education and the nature of the professional of the 20 respondents were asked. In addition to that, we also asked them whether they have eye diseases or whether they are wearing glasses or contact lenses.
Normally the first questions should respond to the topic and have a clear thematic reference. Consequently you put the demographic data to the end of the survey to raise interest and not to bother the participant. Though, we decided to start with this questionnaire. Because of the BCI and the calibration of the eye tracking, it was important to know whether the participant was wearing glasses or contact lenses or whether he had eye diseases.

5.5.1.2. Post-Task Questionnaire

We decided to show each subject six different videos. The main tasks for the subjects were the rating of each video with subsequent assessment of their feeling. For answering both tasks we used a Likert-scale, which will be described in the next subsection. Furthermore we asked them, if they have already watched the trailers/movies before the experiment. This information was necessary, because if a person has seen a trailer or the movie before the experiment, it is possible that he reacts differently to the video than if he had seen it for the first time. For example, if a subject already knows the movie "Ouija", than he also knows the content and the exciting scenes of the movie. Thus, it may be that he is not as excited as watching it for the first time. Additionally we asked them if they would watch the

[27] https://docs.google.com/forms/d/e/1FAIpQLScbZscuNahRAjouYL7SAob5r58oye644yx2mFVNeHu5xttgJw/viewform
[28] https://docs.google.com/forms/d/e/1FAIpQLSdBvuvViW0uJxsxLl9bklV-EuFLEENzEwsDszx-LFlDlhOmLg/viewform
[29] https://www.google.com/forms/about/

whole movie after watching the trailer, because the answer is another indication of popularity. It can be used for the correlation between rating and feelings.

5.5.1.3. Likert-Scale

In our case we used the 5-stage likert-scale to ask subjects about their feelings (interest, stress, engagement, focus, relaxation) and rating after watching each video. We have not used the 7-stage likert-scale because most persons cannot differentiate their answers so finely especially their feelings. This scale is the most frequently used scaling method in social science and has the advantage of an easy evaluation of all answers. Like any method, this also has disadvantages like the danger of counselling rather than real answer. In our case the scale for the feelings and rating were classified as follows:

Classification of the likert-scale for the feelings	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Classification of the likert-scale for the rating	Bad	Poor	Neutral	Good	Excellent

Table 5.5 Likert-scale

5.6. Data Evaluation / Results

The evaluation study was conducted with 20 subjects. There were 10 male and 10 female participants. Three data sets could not be considered in the analysis. Test subjects 10, 11 and 13 have false values, probably caused by susceptibility errors during the evaluation. A Table with the gaze-based similarity of the users can be found in Table 5.7.

The evaluation is mainly concerned with the question of whether statements on similarities can be made solely by means of the visual data of the subjects during a video.

The study is divided into three sections:

First, the observed similarities of the Eye tracking data are viewed individually. Subsequently, statements about similarity of the users are considered more closely. On the one hand, the very similar subjects are compared by their subjective opinions and preferences. Is there a connection between the visual data and the personal preferences?

On the other hand, user pairs are considered which should be different according to their gaze data.

Subsequently, the obtained data are weighed with the statements about the sensations and the measured data of the EPOC-device.

Based on the rating, given by each test subject to each movie similarity was calculated – output is in Table 5.6. For each user three most similar users were defined. The similarities based on fixations and emotions would be represented in the similar manner to assess the overlapping of similar users.

To make the visualization more clear the values are color-coded. Red colour indicates lower similarity scores while blue – is a high similarity. White colour represents average or close to average values.

User	1	2	3	4	5	6	7	8	9	10
1		0,817	0,927	0,877	0,916	0,845	0,904	0,869	0,925	0,943
2	0,817		0,961	0,801	0,951	0,916	0,85	0,974	0,901	0,778
3	0,927	0,961		0,866	0,967	0,926	0,927	0,977	0,943	0,882
4	0,877	0,801	0,866		0,928	0,907	0,811	0,867	0,837	0,98
5	0,916	0,951	0,967	0,928		0,965	0,916	0,967	0,925	0,922
6	0,845	0,916	0,926	0,907	0,965		0,895	0,967	0,914	0,884
7	0,904	0,85	0,927	0,811	0,916	0,895		0,881	0,843	0,86
8	0,869	0,974	0,977	0,867	0,967	0,967	0,881		0,956	0,853
9	0,925	0,901	0,943	0,837	0,925	0,914	0,843	0,956		0,863
10	0,943	0,778	0,882	0,98	0,922	0,884	0,86	0,853	0,863	
11	0,93	0,939	0,981	0,912	0,989	0,965	0,959	0,967	0,925	0,922
12	0,869	0,9	0,929	0,901	0,944	0,981	0,853	0,972	0,961	0,889
13	0,967	0,879	0,932	0,872	0,935	0,853	0,832	0,904	0,954	0,908
14	0,912	0,93	0,942	0,907	0,99	0,971	0,912	0,957	0,938	0,908
15	0,901	0,927	0,944	0,955	0,98	0,953	0,838	0,962	0,939	0,934
16	0,784	0,867	0,907	0,76	0,86	0,93	0,883	0,935	0,888	0,764
17	0,971	0,743	0,861	0,886	0,895	0,837	0,913	0,804	0,851	0,952
18	0,983	0,864	0,947	0,908	0,961	0,92	0,949	0,914	0,934	0,953
19	0,874	0,955	0,947	0,925	0,991	0,941	0,874	0,95	0,883	0,9
20	0,981	0,802	0,909	0,818	0,899	0,846	0,944	0,853	0,907	0,898
average	0,905	0,881	0,930	0,879	0,942	0,916	0,886	0,922	0,910	0,894
Max sim.	0,983	0,974	0,981	0,98	0,991	0,981	0,959	0,977	0,961	0,98
Most sim users	18; 20; 13	12; 3; 8	11; 8; 5	10; 15; 5	14; 19; 15	12; 14; 8;	11; 18; 20	2; 3; 12	12; 13; 8	4; 17; 18

User	11	12	13	14	15	16	17	18	19	20
1	0,93	0,869	0,967	0,912	0,901	0,784	0,971	0,983	0,874	0,981
2	0,939	0,9	0,879	0,93	0,927	0,867	0,743	0,864	0,955	0,802
3	0,981	0,929	0,932	0,942	0,944	0,907	0,861	0,947	0,947	0,909
4	0,912	0,901	0,872	0,907	0,955	0,76	0,886	0,908	0,925	0,818
5	0,989	0,944	0,935	0,99	0,98	0,86	0,895	0,961	0,991	0,899
6	0,965	0,981	0,853	0,971	0,953	0,93	0,837	0,92	0,941	0,846
7	0,959	0,853	0,832	0,912	0,838	0,883	0,913	0,949	0,874	0,944
8	0,967	0,972	0,904	0,957	0,962	0,935	0,804	0,914	0,95	0,853
9	0,925	0,961	0,954	0,938	0,939	0,888	0,851	0,934	0,883	0,907
10	0,922	0,889	0,908	0,908	0,934	0,764	0,952	0,953	0,9	0,898
11		0,944	0,918	0,977	0,956	0,904	0,91	0,973	0,967	0,927
12	0,944		0,887	0,953	0,961	0,935	0,83	0,919	0,91	0,853
13	0,918	0,887		0,931	0,943	0,757	0,909	0,951	0,913	0,927
14	0,977	0,953	0,931		0,967	0,862	0,904	0,964	0,968	0,911
15	0,956	0,961	0,943	0,967		0,838	0,862	0,933	0,974	0,853
16	0,904	0,935	0,757	0,862	0,838		0,733	0,847	0,811	0,808
17	0,91	0,83	0,909	0,904	0,862	0,733		0,975	0,851	0,972
18	0,973	0,919	0,951	0,964	0,933	0,847	0,975		0,921	0,981
19	0,967	0,91	0,913	0,968	0,974	0,811	0,851	0,921		0,842
20	0,927	0,853	0,927	0,911	0,853	0,808	0,972	0,981	0,842	
average	0	0,915	0,904	0,937	0,927	0,846	0,877	0,937	0,916	0,891
Max sim.	0,989	0,981	0,967	0,99	0,98	0,935	0,975	0,983	0,991	0,981
Most sim users	5; 3; 14	6; 8; 9; 15	1; 9; 18	5; 11; 6	5; 19; 14	8; 12; 6	18; 20; 1	1; 20; 17	5; 15; 14	1; 18; 17

Table 5.6 Rating-based similarity

5.6.1. Evaluation of the Eye Tracking Data

The test subjects have seen six actual movie trailers from three different emotional categories (horror, emotion and comedy/action). Their gaze data were recorded and only the ascertained fixings saved. Afterwards, the user's common fixations were determined and the resulting similarity was determined. Our algorithms calculate the similarities of two users according to each video.

Afterwards the arithmetic mean of all video similarities is calculated and gives information about the overall similarity of two users.

The detailed description of the procedure for determining the similarity in gauze data can be found in chapter 4.3: ‘Calculation of User Similarity based on Eye-Tracking Data’.

Ошибка! Источник ссылки не найден. and **Ошибка! Источник ссылки не найден.** show the overall similarities of all participants.

For example, User 8 has a similarity value of 0.44 with User 6. User 20 and User 2 only 0.21. The lowest reachable value is 0 and the highest 1.

User	1	2	3	4	5	6	7	8	9	10
1	0	0,27	0,25	0,21	0,27	0,25	0,25	0,25	0,25	0,25
2	0,27	0	0,27	0,27	0,4	0,33	0,27	0,31	0,35	0,25
3	0,25	0,27	0	0,25	0,29	0,35	0,29	0,33	0,33	0,25
4	0,21	0,27	0,25	0	0,27	0,27	0,27	0,21	0,25	0,25
5	0,27	0,4	0,29	0,27	0	0,38	0,31	0,29	0,33	0,25
6	0,25	0,33	0,35	0,27	0,38	0	0,38	0,44	0,46	0,25
7	0,25	0,27	0,29	0,27	0,31	0,38	0	0,31	0,27	0,25
8	0,25	0,31	0,33	0,21	0,29	0,44	0,31	0	0,38	0,25
9	0,25	0,35	0,33	0,25	0,33	0,46	0,27	0,38	0	0,25
10	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0
11	0	0	0	0	0	0	0	0	0	0
12	0,29	0,48	0,38	0,21	0,42	0,42	0,33	0,35	0,42	0,25
13	0,21	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25
14	0,31	0,44	0,33	0,25	0,38	0,35	0,38	0,33	0,33	0,25
15	0,27	0,29	0,31	0,17	0,29	0,33	0,29	0,31	0,27	0,25
16	0,29	0,42	0,4	0,21	0,46	0,5	0,35	0,38	0,44	0,25
17	0,29	0,46	0,35	0,27	0,44	0,46	0,31	0,31	0,42	0,25
18	0,29	0,42	0,33	0,25	0,42	0,4	0,29	0,35	0,44	0,25
19	0,33	0,35	0,25	0,25	0,33	0,27	0,31	0,27	0,25	0,25
20	0,25	0,21	0,21	0,21	0,25	0,21	0,21	0,21	0,21	0,21
average	0,239	0,302	0,271	0,216	0,302	0,315	0,266	0,277	0,295	0,223
highest similarity	0,33	0,48	0,4	0,27	0,46	0,5	0,38	0,44	0,46	/
users with highest similarity	19; 12; 6; 17; 18; 14	12; 14; 17	16; 12; 6; 17	17; 2; 5; 6; 7	16; 17; 18; 12	16; 9; 17	6; 14; 16	6; 9; 16	6; 16; 18	/

User	11	12	13	14	15	16	17	18	19	20
1	0	0,29	0,21	0,31	0,27	0,29	0,29	0,29	0,33	0,25
2	0	0,48	0,25	0,44	0,29	0,42	0,46	0,42	0,35	0,21
3	0	0,38	0,25	0,33	0,31	0,4	0,35	0,33	0,25	0,21
4	0	0,21	0,25	0,25	0,17	0,21	0,27	0,25	0,25	0,21
5	0	0,42	0,25	0,38	0,29	0,46	0,44	0,42	0,33	0,25
6	0	0,42	0,25	0,35	0,33	0,5	0,46	0,4	0,27	0,21
7	0	0,33	0,25	0,38	0,29	0,35	0,31	0,29	0,31	0,21
8	0	0,35	0,25	0,33	0,31	0,38	0,31	0,35	0,27	0,21
9	0	0,42	0,25	0,33	0,27	0,44	0,42	0,44	0,25	0,21
10	0	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,21
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0,25	0,5	0,33	0,5	0,48	0,44	0,44	0,21
13	0	0,25	0	0,25	0,25	0,25	0,25	0,25	0,25	0,17
14	0	0,5	0,25	0	0,31	0,46	0,48	0,44	0,48	0,25
15	0	0,33	0,25	0,31	0	0,35	0,33	0,27	0,27	0,21
16	0	0,5	0,25	0,46	0,35	0	0,5	0,5	0,31	0,21
17	0	0,48	0,25	0,48	0,33	0,5	0	0,48	0,33	0,25
18	0	0,44	0,25	0,44	0,27	0,5	0,48	0	0,33	0,21
19	0	0,44	0,25	0,48	0,27	0,31	0,33	0,33	0	0,25
20	0	0,21	0,17	0,25	0,21	0,21	0,25	0,21	0,25	0
average	0	0,335	0,219	0,326	0,255	0,339	0,333	0,318	0,276	0,197
higest similarity	/	0,5	/	0,5	0,35	0,5	0,5	0,5	0,48	0,25
user with highest similarity	/	14; 16; 17; 2	/	12; 17; 19	16; 17; 12	12, 17, 18	16;12; 14; 18	16; 17; 14; 12; 9	14; 12; 2	1; 5; 14; 17;18

Table 5.7 Gaze-based overall similarity

Mainly in red are values 0. Normally a user should have a really high value (1) compared with himself. Obviously this calculation would not generate any further knowledge, which is why the value 0 is also to be found there. (For example user 10 compared with user 10.) Since no values for the calculation of these similarities exist, these cases are evaluated as if there are no or only incorrect data. All other 0 values are followed by missing or incorrect data. A demolition of the transfer of the gauze stream during the recording may have been a possible cause. As mentioned above, 11 was not considered, the table shows clearly why.

Tests 10 and 13 were not considered, since all users were assigned the same similarity value (0,25).

The values are color-coded. See also the legend under the table. Red colour means that the numbers are very low and blue very high. The highest similarity between two users is 0.5.

This high value appears for 5 user pairs: user 6 & 16, user 12 & 14, user 12 & 16, user 16 & 17, user 16 & 18.

If you consider user 3 and user 4 with user 20, then you realize, that they have the same similarity value. While user 4 and user 20 have low similarity values continuously, the data at User 3 and User 20 are full-grown.

Since user 16 has the highest similarity values in common, this is considered more closely. **Ошибка! Источник ссылки не найден.** shows the individual data for each of the 6 videos for user 16. The similarities of the gaze data in the individual videos is shown in the table and the resulting overall similarity in the overall column at the right side.

test16:							
Users	video 1	video 2	video 3	video 4	video 5	video 6	overall
test1	0,25	0,25	0,38	0,25	0,25	0,38	0,29
test2	0,38	0,5	0,38	0,38	0,5	0,38	0,42
test3	0,38	0,38	0,38	0,62	0,25	0,38	0,4
test4	0,25	0,25	0	0,25	0,25	0,25	0,21
test5	0,25	0,62	0,25	0,38	0,62	0,62	0,46
test6	0,62	0,62	0,38	0,5	0,38	0,5	0,5
test7	0,62	0,5	0,25	0,25	0,25	0,25	0,35
test8	0,62	0,5	0,25	0,25	0,25	0,38	0,38
test9	0,5	0,25	0,38	0,5	0,38	0,62	0,44
test10	0,25	0,25	0,25	0,25	0,25	0,25	0,25
test11	0	0	0	0	0	0	0
test12	0,5	0,5	0,5	0,5	0,5	0,5	0,5
test13	0,25	0,25	0,25	0,25	0,25	0,25	0,25
test14	0,62	0,62	0,25	0,38	0,38	0,5	0,46
test15	0,38	0,38	0,38	0,38	0,38	0,25	0,35
test16	0	0	0	0	0	0	0
test17	0,25	0,62	0,38	0,62	0,62	0,5	0,5
test18	0,5	0,62	0,25	0,62	0,38	0,62	0,5
test19	0,38	0,38	0,25	0,25	0,25	0,38	0,31
test20	0,25	0,25	0,25	0	0,25	0,25	0,21

legend	0	0,1	0,2	0,3	0,4	0,5

Table 5.8 Similarity for user 16

These values could also be found in the overall similarity table above. But even the smallest overall similarities of user 16 to others are comparatively high (~0,3). 0,21 is the minimal value.

In this table, it should be mentioned that the lowest and most frequent values of the coloring have taken place. So they are not exactly the same colors as in the upper table. For Video 2 the highest Values appear.

To independently judge the actual video, the similarity values for each video were viewed individually. The weighting of the similarity is based on the average fixations of that particular video. Thus, if a trailer has a very high number of fixings in total, it is weighted as if it has few fixings. In the case of a high average number, more common fixations must also appear in order to obtain a high similarity value than in the case of videos with an average of less fixations. Thereby an independent weighting of the similarity can take place. Film trailers are, of course, intended to guide the viewer's eye mostly to a point of the video. We have developed the method of individual weighting so as to be able to use well-made trailers as well as those which do not provide guidance of the view.

5.6.2. Evaluation of the BCI data

Due to software and hardware malfunctioning 11 and 12 was not recorded and wasn't saved in the database.

5.6.2.1. Accuracy of emotion recognition

In order to estimate the accuracy of the emotion recognition, the recorded emotion rates were compared to the feedback provided by users after watching each trailer. In their feedback they were asked about respective emotional states (interest, focus, stress, engagement, and relaxation) and how strong felt was the emotion they felt.

Possible answers were:

- Strongly Disagree;
- Disagree;
- Neutral;
- Agree;
- Strongly Agree.

Recorded emotions have values in the range from 0 to 1. For the evaluation they were scaled according to the answers:

- Strongly disagree = 0.000-0.199
- Disagree = 0.200-0.399
- Neutral = 0.400-0.599
- Agree = 0.600-0.799
- Strongly agree = 0.800-1.000

Percentage of matches between user feedback and recognised emotions is 53, 583

5.6.2.2. Emotion-based similarity

For the emotion-based similarity model same visualization was created. Was used the same colour encoding scheme as in Table 5.6 Rating-based similarity and Table 5.7 Gaze-based overall similarity.

Comparison with rating-based similarity resulted in 41,7 percent overlapping in 25 similar users recognition. Comparison with fixations-based similarity resulted in 7 users considered similar to same respective users in both models. Those are: User 1 having highest similarity with users 17 and 18; 2 to 14; 17 to 18 and vice versa; user 20 to users 1 and 18.

User	1	2	3	4	5	6	7	8	9	10
1		0,884	0,583	0,771	0,789	0,684	0,757	0,764	0,857	0,246
2	0,884		0,354	0,449	0,868	0,816	0,736	0,969	0,842	0,579
3	0,583	0,354		0,609	0,735	0,834	0,768	0,776	0,954	0,589
4	0,771	0,449	0,609		0,761	0,554	0,715	0,389	0,877	0,969
5	0,789	0,868	0,735	0,761		0,798	0,564	0,943	0,815	0,784
6	0,684	0,816	0,834	0,554	0,798		0,546	0,745	0,81	0,623
7	0,757	0,736	0,768	0,715	0,564	0,546		0,652	0,781	0,837
8	0,764	0,969	0,776	0,389	0,943	0,745	0,652		0,817	0,435
9	0,857	0,842	0,954	0,877	0,815	0,81	0,781	0,817		0,478
10	0,246	0,579	0,589	0,969	0,784	0,623	0,837	0,435	0,478	
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0,544	0,897	0,735	0,831	0,902	0,635	0,627	0,814	0,812	0,765
14	0,684	0,898	0,761	0,91	0,768	0,843	0,534	0,859	0,836	0,724
15	0,724	0,789	0,843	0,889	0,757	0,774	0,521	0,687	0,845	0,801
16	0,438	0,456	0,726	0,665	0,603	0,687	0,598	0,884	0,687	0,462
17	0,918	0,549	0,522	0,693	0,81	0,603	0,802	0,658	0,621	0,813
18	0,963	0,687	0,871	0,821	0,879	0,599	0,856	0,747	0,798	0,783
19	0,788	0,861	0,838	0,873	0,981	0,883	0,598	0,839	0,618	0,735
20	0,865	0,403	0,698	0,435	0,678	0,58	0,871	0,562	0,733	0,709
average	0,645	0,634	0,642	0,643	0,707	0,632	0,619	0,660	0,694	0,596
average	0,963	0,969	0,954	0,969	0,981	0,883	0,871	0,969	0,954	0,969
higest similarity	18; 17; 2	8; 13; 14	9; 19; 18	10; 14; 15	19; 8; 13	19; 14; 3	20; 18; 10	2; 5; 3	3; 4; 1	4; 7; 17

User	11	12	13	14	15	16	17	18	19	20
1	0	0	0,544	0,684	0,724	0,438	0,918	0,963	0,788	0,865
2	0	0	0,897	0,898	0,789	0,456	0,549	0,687	0,861	0,403
3	0	0	0,735	0,761	0,843	0,726	0,522	0,871	0,838	0,698
4	0	0	0,831	0,91	0,889	0,665	0,693	0,821	0,873	0,435
5	0	0	0,902	0,768	0,757	0,603	0,81	0,879	0,981	0,678
6	0	0	0,635	0,843	0,774	0,687	0,603	0,599	0,883	0,58
7	0	0	0,627	0,534	0,521	0,598	0,802	0,856	0,598	0,871
8	0	0	0,814	0,859	0,687	0,884	0,658	0,747	0,839	0,562
9	0	0	0,812	0,836	0,845	0,687	0,621	0,798	0,618	0,733
10	0	0	0,765	0,724	0,801	0,462	0,813	0,783	0,735	0,709
11		0	0	0	0	0	0	0	0	0
12	0		0	0	0	0	0	0	0	0
13	0	0		0,718	0,801	0,487	0,514	0,729	0,651	0,756
14	0	0	0,718		0,84	0,546	0,564	0,654	0,745	0,354
15	0	0	0,801	0,84		0,838	0,549	0,684	0,924	0,575
16	0	0	0,487	0,546	0,838		0,554	0,589	0,572	0,576
17	0	0	0,514	0,564	0,549	0,554		0,914	0,621	0,857
18	0	0	0,729	0,654	0,684	0,589	0,914		0,719	0,927
19	0	0	0,651	0,745	0,924	0,572	0,621	0,719		0,681
20	0	0	0,756	0,354	0,575	0,576	0,857	0,927	0,681	
average	0	0	0,643053	0,64410526	0,67584211	0,545684211	0,60852632	0,69578947	0,6803684	0,59263158
higest similarity	0	0	0,902	0,91	0,924	0,884	0,918	0,963	0,981	0,927
user with highest similarity			5; 2; 4	4; 2; 8	19; 4; 9	8; 15; 3	1; 18; 20	1; 20; 17	5; 15; 6	18; 7; 1

Table 5.9 Emotion-based similarities

5.6.3. Evaluation of the Demographic Questionnaire

This section contains the evaluation of the demographic questionnaire and will disclose important characteristics of the sample. Within the scope of our study the data of subjects were raised between the 19.05.2017 and the 26.05.2017. In total 20 respondents (n=20) participated in the experiment and answered both questionnaires. This sample consists of 10 women and also 10 men. Hence, the number of the genders is equal.

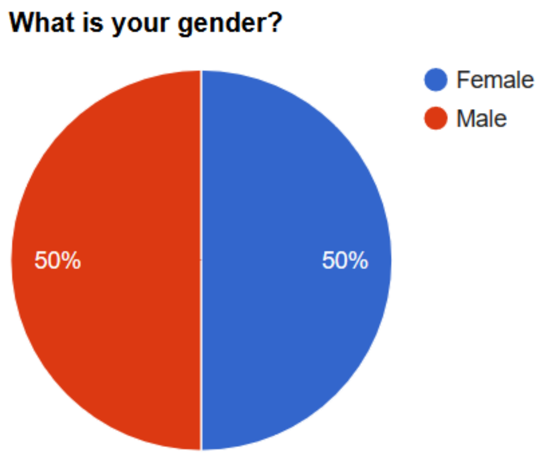

Figure 5.5 Gender Ratio

It is striking that all probands are between 20 and 29 years old. Though, this is due to the fact that we had defined our target group in advance and invited the volunteers accordingly. The average age of all subjects is 24,5 years.

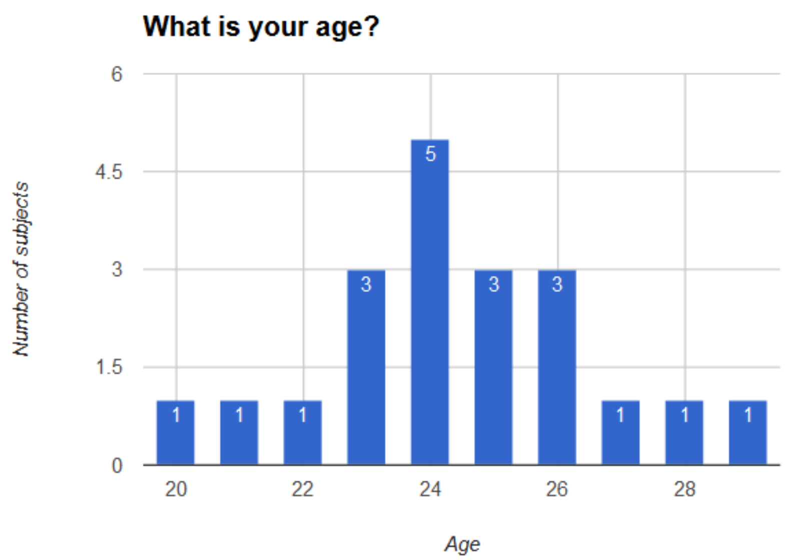

Figure 5.6 Gender histogram

A crosstab is the tabular representation of the frequencies that occur when the variables of two or more variables are combined. In our cause we combined the variables - woman and man. The values of the variables A (age) are entered from left to right in columns and the values of the variables B (gender) are entered in rows from top to bottom. In each individual cell, the specific frequency of the respective combination expression of the variable A with the expression of the variable B is then noted. The results can be interpreted more easily, even if the relative frequencies are determined. In our case, the crosstab only serves as an overview. This is why we renounce on the calculation of the frequencies and the verification of the correlations with the Chi-square test. This would be inappropriate at this point.

		Age										Total
		20	21	22	23	24	25	26	27	28	29	
Gender	w[30]	-	-	-	2	3	2	2	1	-	-	10
	m[31]	1	1	1	1	2	1	1	-	1	1	10
Total		1	1	1	3	5	3	3	1	1	1	20

Table 5.10 Gender distribution

Each subject has indicated that he is studying computer science. Here we can separate between the following courses of study - web science, computer science, computational visualistics and computer science/mathematics.

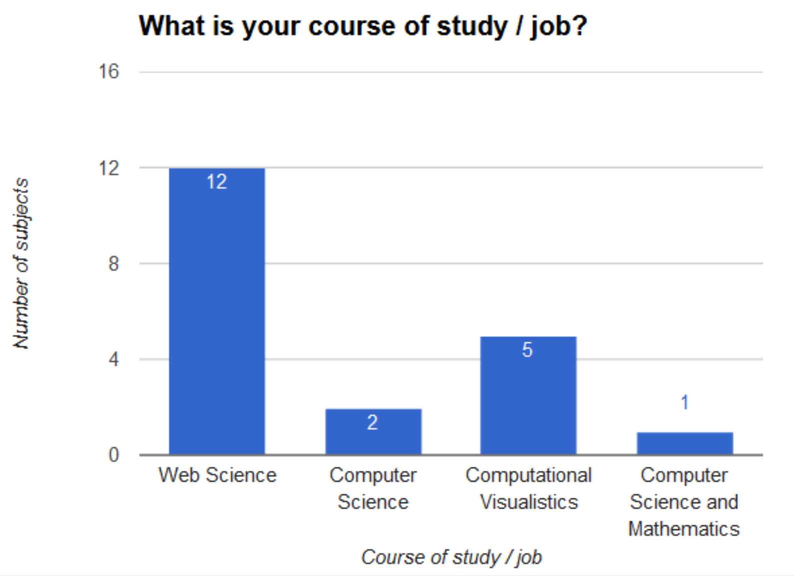

Figure 5.7 Job's histogram

Therefore, we can assume that each volunteer is technically knowledgeable and possess good IT user knowledge. But this does not mean that he is an expert in controlling an application through the gaze only. Therefore, it was necessary to ask them, if they used an eye tracker before the experiment. With

[30] Women
[31] Men

these answers we can draw a conclusion about their experience. Unfortunately we have not asked how often they have already used an eye tracker. Therefore it is only possible to determine whether they are novices or more experienced persons in this area.

Nearly half of all probands (45%, 9 subjects) have already used an eye tracker. In comparison, exactly 50% (10 subjects) have never used one and only one person (5%) does not know if he has ever used one before.

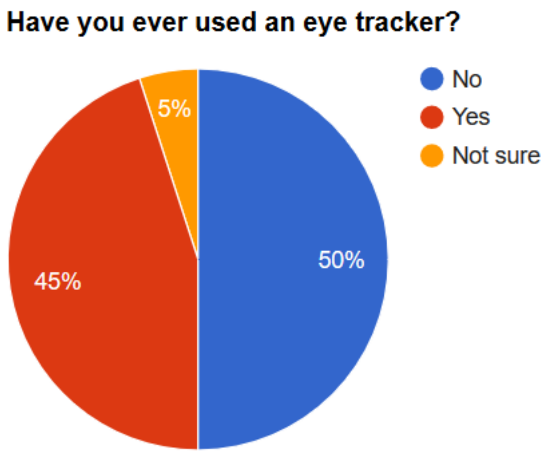

Figure 5.8 Eye-tracker user's ratio

About 55,6% (5 subjects) of the persons who have used one, specified that they used SMI only and ca.11,1% (1 subjects) of them specified that they used Tobi and SMI. One subject (ca. 11,1%) used one before, but was not able to identify it. The remaining two subjects (ca. 22,2%) have used an eye tracker, but did not specify which one(s).

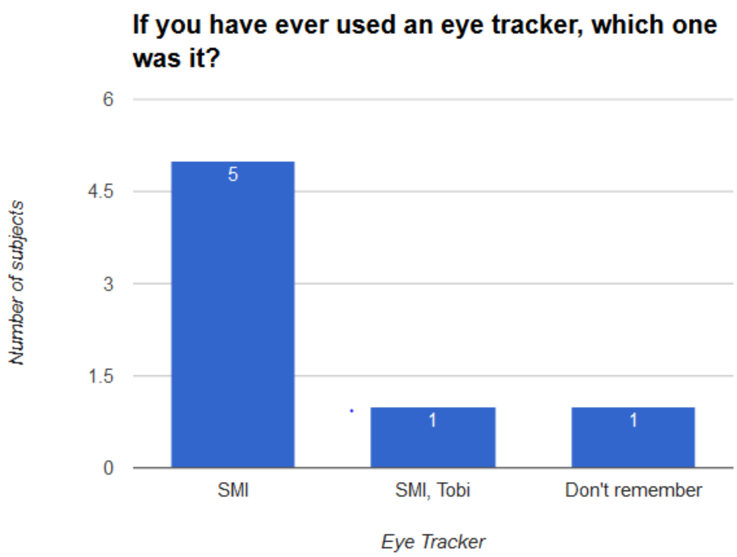

Figure 5.9 Eye-tracker user's histogram

For the calibration of the eye tracker and for problems e. g. with the calculation of fixation points or similarity during the experiment, it was also important to know, if a person has eye diseases (e. g.

squint). Only 20% (4 subjects) answered that they suffer from a visual weakness (short-sightedness). Three of four short-sighted persons are wearing contact lenses. It is noticeable that three people have indicated that they wear glasses or contact lenses but have not registered any eye diseases.

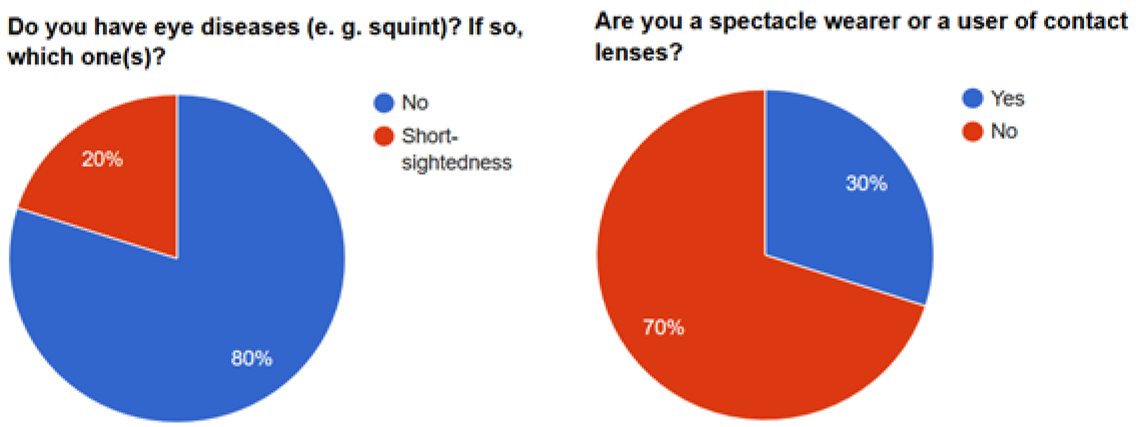

Figure 5.10 Eye diseases andspecracle wearer ratios

13 subjects have indicated that they do not wear glasses or contact lenses and have no eye problems either.

The following table describes the combination of the results of the two questions and it is separated into these cases:

1. Spectacle wearer / wearer of contact lenses
2. Eye diseases
3. Spectacle wearer / wearer of contact lenses, but no eye diseases
4. No spectacle wearer / wearer of contact lenses, but eye diseases
5. No spectacle wearer / wearer of contact lense and no eye diseases
6. Spectacle wearer / wearer of contact lenses and eye diseases

		1.	2.	3.	4.	5.	6.
Gender	w	8	2	-	1	7	2
	m	6	1	3	0	6	1
Total		14	3	3	1	13	3

The cases three and four are questionable, because normally a person wears glasses or contact lenses if a weakness or eye disease has been observed. Unless it is a fashion glasses. And it is not clear why somebody do not wear glasses even though he has eye diseases. On the basis of these facts it can be stated that the questions may have been formulated unclearly and the result are misunderstandings.

At least the analysis of the last question is missing. We asked them what kind of genres they prefer. For this, we have previously made a selection, which is limited to six categories (action, adventure, comedy, horror, science fiction, westerns). We asked that question as multiple choice, because a person can like several genres and not specially one. The following diagram shows the answers.

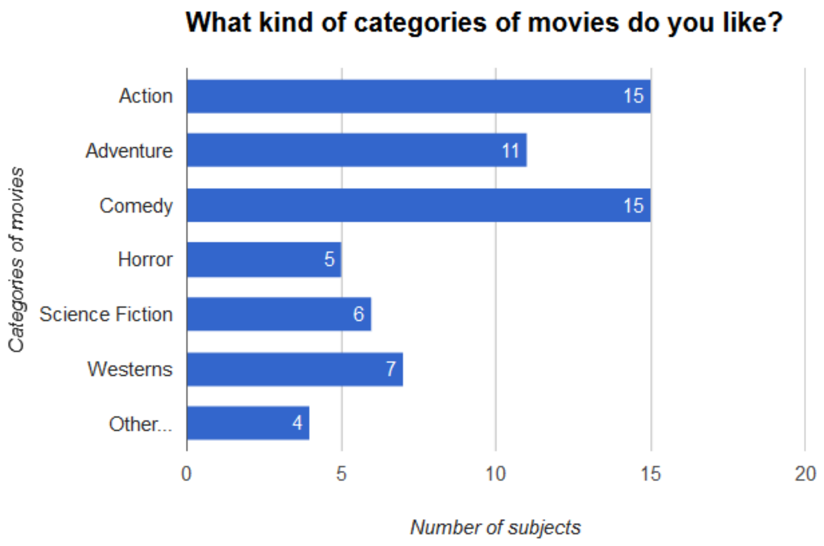

Figure 5.11 Movie preferences ratio

		Gender		Total
		w	m	
Genre	Action	6	9	15
	Adventure	6	5	11
	Comedy	10	5	15
	Horror	2	3	5
	Science Fiction	3	3	6
	Westerns	0	7	7
	Other…:			
	• Fantasy	1	2	3
	• Drama	-	1	1
	• Thriller	-	1	1
	• Detective	-	1	1

Figure 5.12 Movie preferences

The most popular genre of our target group are Action and Comedy with 15 answers each. The second popular category is Adventure with 11 answers. Only four subjects have added other genres. One of these four persons has added the three categories Thriller, Detective and Drama and the other three subjects have added Fantasy only.

5.7. Discussion

Both new types of data being used for similarity measurement had low correlation with each other or control data. Multiple factors, including the equipment malfunctioning affected the quality of the recording and caused partial or complete loss of data of three test subjects.
Reason for that is a very big range of possible values each variable in both given datatypes can have comparing to the rating-based system. In order to overcome this problem additional pre and post processing steps required.
However, both new types of user-data can be used for measuring similarity and evaluating the specific behaviour of users.

5.7.1. Ideas

In the beginning we had the idea to compare not only time and position of overlapping fixations. We wanted to check, if the participants have the same feelings as well. In the end it is not possible to detect only one feeling at one timestamp correctly. It is more a time range and so we could not use this idea to compare and analyse user similarity.
The selection of the videos turned out to be very difficult. A huge amount of movie trailers have been viewed, but it is difficult to assign these to the actual genre. Thus, emotionally emotive films are both sad, but can also contain happy emotional parts. Comedies are often paired with action (at least in the trailer). To produce in just two minutes a real deeper emotion in a laboratory is somewhat difficult. However, it was decided to use the trailers, considering that they can usually produce a change of emotions in a short period of time.

6. Conclusion

6.1. Evaluation Results

The analysis of the recorded gaze fixations and emotions had shown that both similarity models have low match rate when compared to the rating-based similarity model. Besides, the emotions recorded by the BCI had a low match rate with the respective emotions which the test subjects stated in the questionnaires. Multiple factors, including the equipment malfunctioning affected the quality of the recording and caused partial or complete loss of data of three test subjects. However, both new sources of user-data can be used for measuring similarity and evaluating the specific behaviour of users.

6.2. Further Improvements

Various aspects and stages of the EBCI data collection and processing can be improved.

1) Development and implementation of alternative algorithms for raw gaze-data processing. Additional study and tuning of the fixation points recognition technologies required.
2) Emotions recognition was based on the proprietary algorithms developed by Emotiv. Development of alternative algorithms may improve the accuracy of recognition.
3) Advancement of tracking devices reliability.
4) Applying different strategies of similarity calculation for all three types of data.
5) Possibility to advance the accuracy of emotion recognition using patterns in gaze movements caused by particular emotions.
6) Alternative methods of emotion’s validation should be explored since the feedback provided by users themselves may not always be accurate.
7) In order to assess the full potential of the similarity models larger sets of test subjects and videos should be involved in the experiment.
8) Further development supposes generation of recommendations based on measured similarities including videos previously unseen.
9) Development of scaling recommender system based on gaze and emotion data.

7. List of Figures

8. Appendix

8.1. Demographic Questionnaire

1. Demographic Information

*Required

1. **Test Subject ID** *

2. **What is your gender?** *
   Mark only one oval.
   - Male
   - Female

3. **What is your age?** *

4. **Are you currently...?** *
   Mark only one oval.
   - a student
   - an employee
   - an employeer
   - unable to work

5. **What is your course of study/job?** *

6. **Are you a spectacle wearer or a user of contact lenses?** *
   Mark only one oval.
   - Yes
   - No

7. **Do you have eye diseases (e. g. squint)? If so, which one(s)?** *

8. **Have you ever used an eye tracker?** *
   Mark only one oval.
   - Yes
   - No
   - Not sure

9. **If you have ever used an eye tracker, which one was it?**

10. **What kind of categories of movies do you like?** *
Tick all that apply.

- [ ] Action
- [ ] Adventure
- [ ] Comedy
- [ ] Horror
- [ ] Science Fiction
- [ ] Westerns
- [ ] Other: ______________________

8.2. Post-Task Questionnaire

Experiment - BCI and EyeTracking

*Required

1. **Test Subject ID** *

____

1. Video

2. **What trailer have you watched?** *
Mark only one oval.

- ( ) Going in Style
- ( ) Suicide squad
- ( ) Love is all you need?
- ( ) Wolves
- ( ) Lights out
- ( ) Ouija

3. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...	( )	( )	( )	( )	( )

4. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- ( ) Yes
- ( ) No
- ( ) Not sure

5. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- ( ) Yes
- ( ) No
- ( ) Not sure

6. **How do you feel after watching the video?** *
Please read all the feelings bellow and mark how much you disagree or agree.
Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.	⬭	⬭	⬭	⬭	⬭
Stress (Frustration) - you felt overwhelmed, had negative impression.	⬭	⬭	⬭	⬭	⬭
Engagement - you were highly immersed in what was happening in the video.	⬭	⬭	⬭	⬭	⬭
Focus - you didn't switch to thinking about other things while watching.	⬭	⬭	⬭	⬭	⬭
Relaxation (Meditation) - you felt calm wasn't nervous about anything.	⬭	⬭	⬭	⬭	⬭

2. Video

7. **What trailer have you watched?** *
Mark only one oval.

- ⬭ Going in Style
- ⬭ Suicide squad
- ⬭ Love is all you need?
- ⬭ Wolves
- ⬭ Lights out
- ⬭ Ouija

8. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...	⬭	⬭	⬭	⬭	⬭

9. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

10. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

11. **How do you feel after watching the video?** *
Please read all the feelings bellow and mark how much you disagree or agree.
Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.					
Stress (Frustration) - you felt overwhelmed, had negative impression.					
Engagement - you were highly immersed in what was happening in the video.					
Focus - you didn't switch to thinking about other things while watching.					
Relaxation (Meditation) - you felt calm wasn't nervous about anything.					

3. Video

12. **What trailer have you watched?** *
Mark only one oval.

- Going in Style
- Suicide squad
- Love is all you need?
- Wolves
- Lights out
- Ouija

13. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...					

14. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- Yes
- No
- Not sure

15. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- Yes
- No
- Not sure

16. **How do you feel after watching the video?** *
Please read all the feelings bellow and mark how much you disagree or agree.
Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.					
Stress (Frustration) - you felt overwhelmed, had negative impression.					
Engagement - you were highly immersed in what was happening in the video.					
Focus - you didn't switch to thinking about other things while watching.					
Relaxation (Meditation) - you felt calm wasn't nervous about anything.					

4. Video

17. **What trailer have you watched?** *
Mark only one oval.

- Going in Style
- Suicide squad
- Love is all you need?
- Wolves
- Lights out
- Ouija

18. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...					

19. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- Yes
- No
- Not sure

20. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- Yes
- No
- Not sure

21. **How do you feel after watching the video?** *
Please read all the feelings bellow and mark how much you disagree or agree.
Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.	⬭	⬭	⬭	⬭	⬭
Stress (Frustration) - you felt overwhelmed, had negative impression.	⬭	⬭	⬭	⬭	⬭
Engagement - you were highly immersed in what was happening in the video.	⬭	⬭	⬭	⬭	⬭
Focus - you didn't switch to thinking about other things while watching.	⬭	⬭	⬭	⬭	⬭
Relaxation (Meditation) - you felt calm wasn't nervous about anything.	⬭	⬭	⬭	⬭	⬭

5. Video

22. **What trailer have you watched?** *
Mark only one oval.

- ⬭ Going in Style
- ⬭ Suicide squad
- ⬭ Love is all you need?
- ⬭ Wolves
- ⬭ Lights out
- ⬭ Ouija

23. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...	⬭	⬭	⬭	⬭	⬭

24. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

25. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

26. **How do you feel after watching the video?** *
Please read all the feelings bellow and mark how much you disagree or agree.
Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.	⬭	⬭	⬭	⬭	⬭
Stress (Frustration) - you felt overwhelmed, had negative impression.	⬭	⬭	⬭	⬭	⬭
Engagement - you were highly immersed in what was happening in the video.	⬭	⬭	⬭	⬭	⬭
Focus - you didn't switch to thinking about other things while watching.	⬭	⬭	⬭	⬭	⬭
Relaxation (Meditation) - you felt calm wasn't nervous about anything.	⬭	⬭	⬭	⬭	⬭

6. Video

27. **What trailer have you watched?** *
Mark only one oval.

- ⬭ Going in Style
- ⬭ Suicide squad
- ⬭ Love is all you need?
- ⬭ Wolves
- ⬭ Lights out
- ⬭ Ouija

28. **How much do you like the video?** *
Please rate it.
Mark only one oval per row.

	1 Star (bad)	2 Stars	3 Stars	4 Stars	5 Stars (excellent)
I give ...	⬭	⬭	⬭	⬭	⬭

29. **Did you see the trailer or movie before this experiment?** *
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

30. **If you haven't seen the video/movie before, would you like to see the whole movie in the future?**
Mark only one oval.

- ⬭ Yes
- ⬭ No
- ⬭ Not sure

31. **How do you feel after watching the video?** *

Please read all the feelings bellow and mark how much you disagree or agree.

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Interest - you liked the video and felt curious about it.					
Stress (Frustration) - you felt overwhelmed, had negative impression.					
Engagement - you were highly immersed in what was happening in the video.					
Focus - you didn't switch to thinking about other things while watching.					
Relaxation (Meditation) - you felt calm wasn't nervous about anything.					