

Konzeption und Implementierung eines Datenanalyse-Werkzeugs für Body-Tracking-Kameras

Bachelorarbeit von
Joel Temiz
1196670

Universität der Bundeswehr München
Fakultät für Informatik

Konzeption und Implementierung eines Datenanalyse-Werkzeugs für Body-Tracking-Kameras

Bachelorarbeit von
Joel Temiz
1196670
Joel.Temiz@unibw.de
Werner-Heisenberg-Weg 120, App. 2129
DE-85579, Neubiberg

Aufgabensteller: Prof. Dr. Michael Koch
Zweitprüfer: Prof. Dr.-Ing. Mark Minas
Betreuer: Julian Fietkau
Abgabetermin: 4. Januar 2022

Universität der Bundeswehr München
Fakultät für Informatik

Abstract

Bei der Analyse menschlichem Nutzungsverhaltens wird häufig Sensorik verwendet, um die erforderlichen Daten zu erheben. Als eine mögliche Sensorik steht die Body-Tracking-Kamera zur Auswahl, die allerdings vor allem über längere Zeiträume große Datenmengen produziert. Die Daten können dabei in Form von Log-Dateien gespeichert werden, in denen das Aufgenommene in einem festgelegten Format niedergeschrieben wird. Um von den generierten Log-Dateien zu hilfreichen Aussagen zu gelangen, bedarf eines Auswertungs- oder Analysewerkzeugs, das die Daten in eine, für den Menschen verständliche Form bringt. In der Bachelorarbeit wird die Frage bearbeitet, wie ein Datenanalyse-Werkzeug aufgebaut sein kann, um dessen User einen Nutzen stiften zu können. Dabei wird vor allem auf die Auswertung von Body-Tracking-Kameras eingegangen. Für die Beantwortung dieser Frage wird im ersten Schritt eine Analyse durchgeführt. Darauf aufbauend wird das Analysetool konzipiert und anschließend prototypisch implementiert. Zur Erfolgsermittlung der Arbeit wird das Analysetool abschließend durch einen Domänenexperten im Bereich Mensch-Computer-Interaktion und Ambient-Displays evaluiert.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
1.1	Motivation	2
1.2	Verwandte Arbeiten	2
1.3	Grundlagen und Definitionen	3
2	Analyse	7
2.1	Anforderungen	7
2.1.1	Vision	7
2.1.2	Funktion	8
2.1.3	Nutzung	8
2.1.4	Visualisierung	9
2.1.5	Filterung	10
2.2	Anwendungsfälle	13
3	Konzeption	16
3.1	GUI-Mockup	16
3.2	Programmablauf	20
3.3	Teilsysteme	21
4	Implementierung	24
4.1	Entwicklungsgrundlagen	24
4.1.1	Programmiersprache	24
4.1.2	Entwicklungsumgebung	25
4.2	Programmaufbau	26
4.3	Herausforderungen und Schwierigkeiten	27
4.3.1	Programmkomplexität	27
4.3.2	Filteralgorithmen	29
4.3.3	Berechnung der Heatmap-Daten	29
4.3.4	CORS-Policy	29
4.4	Codebeschreibung	30
4.4.1	Programmkomplexität	31
4.4.2	Filteralgorithmen	33
4.4.3	Berechnung der Heatmap-Daten	38
5	Evaluation	40
5.1	Thinking-Aloud-Test	40
5.1.1	Test-Bedingungen	40
5.1.2	Test-Ergebnis	41
5.2	Fragebogen	42
6	Fazit und Ausblick	45

Inhaltsverzeichnis

Abkürzungsverzeichnis	46
Abbildungsverzeichnis	47
Listings	48
Literaturverzeichnis	49
Anhang A. Evaluation Aufgabenblatt	51
Anhang B. Evaluation Transkript	54
Anhang C. Evaluation Fragebogen	65

1 Einleitung und Motivation

Von Fruit Ninja über Zumba und Harry Potter bis hin zur Grenzüberwachung Südkoreas – die von Microsoft hergestellte Tiefenkamera Kinect hat ein weitreichendes Einsatzgebiet. Erstmals wurde die Forschung für die 3D-Technologie durch ein von Microsoft beauftragtes Unternehmen gestartet, dem es gelang unter den Anforderungen Microsofts einen Prototyp zu entwickeln, der Kamerabilder mit Tiefendaten mit einer Frequenz von 30fps bildete. Der ursprüngliche Anwendungsfall der Kinect v2 war die Steuerung der Microsoft Spielekonsole Xbox 360. (Viereck, 2012)

Nachdem kurz nach ihrer Veröffentlichung ein Open-Source-Treiber für die Microsoft-Kinect entwickelt wurde, war das Einsatzgebiet des Systems nicht mehr auf die bloße Steuerung einer Spielekonsole begrenzt (Gieselmann, 2010). Ein Einsatzgebiet, das sehr stark vom Ursprung abweicht, ist die Unterstützung der Grenzüberwachung in Südkorea. Seit August 2013 wurde die 3D-Kamera dort eingesetzt, um Objekte zu identifizieren, die sich in der *Demilitarized Zone* bewegen. Das System könne sogar zwischen Mensch und Tier differenzieren. (Ashcraft, 2014)

Neben dem militärischen Einsatz der 3D-Kamera gibt es zahlreiche andere Verwendungsmöglichkeiten, bei denen sie einen großen Mehrwert bringen kann. Auch in dieser Arbeit hat die 3D-Kamera einen zentralen Stellenwert. In der Bachelorarbeit „Konzeption und Implementierung eines Datenanalyse-Werkzeugs für Body-Tracking-Kameras“ sollen die Entwicklungsschritte eines Auswertetools erarbeitet werden. Dabei soll dieses Tool vor allem auf die Auswertung von Body-Tracking-Daten spezialisiert sein. Ziel der Arbeit ist es, die Frage zu beantworten, wie ein Datenanalyse-Tool für Body-Tracking-Daten aufgebaut sein kann, um einen Nutzen zu bringen. Diese Fragestellung wird bearbeitet, indem ein solches Tool konzipiert und aufbauend auf den Ergebnissen der Konzeption prototypisch implementiert wird. Da, wie soeben exemplarisch gezeigt wurde, das Einsatzgebiet von Body-Tracking-Kameras sehr weitläufig sein kann, ist die Bearbeitung dieser Fragestellung für eine Vielzahl an Forschungszwecken relevant. Zum einen könnte es für Marketingforschung eingesetzt werden, um Nutzerverhalten von Werbetafeln zu beurteilen. Zum anderen ist die Auswertung von Body-Tracking-Daten im militärischen bzw. sicherheitstechnischen Bereich relevant. Als drittes, wäre ein solches Analyse-Tool ebenso für gewerbliche Zwecke nützlich. Hierbei könnten beispielsweise Personenbewegungen verfolgt werden, um eine Fabrikhalle zu optimieren und somit Lauf- und Fahrwege zu verkürzen.

Um in dem vielfältigen Einsatz des Tools einen Nutzen erbringen zu können, werden im Zuge dieser Bachelorarbeit in den Kapiteln mehrere Schritte durchlaufen. In Kapitel 2 wird die Analyse des Datenanalyse-Tools durchgeführt. In dem Kapitel werden die Anforderungen an das Tool ermittelt, die im Anschluss durch Use-Cases homogenisiert werden. Das darauffolgende Kapitel 3 behandelt die Konzeption des Tools. Dabei wird neben einem GUI-Mockup, der allgemeine Programm-Ablauf dargestellt. Zusätzlich wird die Unterteilung des Gesamtsystems in zwei Teilsysteme diskutiert. Kapitel 4 handelt von der Implementierung des Prototyps des Datenanalyse-Tools. Die Implementierung des Tools besitzt die Untergliederung in Entwicklungsgrundlagen, Programmaufbau, Herausforderungen und Schwierigkeiten und der Code-Beschreibung. Nach Abschluss des Kapitels *Implementierung* wird die Evaluation des implementierten Prototyps durchgeführt. Dabei wird anhand eines Usability-Tests und eines

Fragebogens das implementierte Tool evaluiert, um die Erfolgsermittlung der Konzeption und Implementierung durchzuführen. Am Ende der Arbeit werden die Ergebnisse anhand eines Fazits zusammengefasst.

1.1 Motivation

Im Oktober 2021 war der Kick-Off eines dreijährigen Projekts, das im Zusammenspiel der Universität der Bundeswehr München und der Hochschule für angewandte Wissenschaften in Hamburg durchgeführt wird. In dem Projekt *HoPE*¹ wird der Honey-Pot-Effekt an Ambient-Displays im (halb-)öffentlichen Raum mittels einer Langzeit-Feldstudie untersucht.

Ziel dieser Studie ist es, den Honey-Pot-Effekt an interaktiven Großbildschirmen zu untersuchen und somit den derzeitigen Kenntnisstand durch quantitative Ergebnisse weiter zu festigen. Innerhalb dieser Langzeit-Feldstudie ist die Installation solcher Ambient-Displays an mehreren öffentlichen Orten, wie Universitäten vorgesehen. Zur Erhebung von auswertbaren Daten werden die Ambient-Displays mit Tiefenkameras ausgestattet, die Nutzungsdaten der Passanten und Nutzer umfangreich aufzeichnen und dadurch eine asynchrone Auswertung der Nutzung ermöglichen. Durch die große Menge an Daten, die bei der Aufzeichnung von Nutzungsdaten entsteht, ist für die Auswertung ein Software-Tool notwendig, welches die Analyse nach bestimmten Eingabeparametern durchführt. Die Auswertung würde sich durch das manuelle Konfigurieren der Parameter stark verkomplizieren, weshalb der Bedarf an einem halb-automatischen Daten-Analyse-Tool für die Auswertung jener Daten besteht. Der derzeitige Stand der Forschung zum Thema Auswertung von Body-Tracking-Daten ist jedoch sehr gering, wodurch die Konzeption eines solchen Analyse-Tools für Body-Tracking-Daten eine Forschungslücke darstellt. Diese Forschungslücke soll in dieser Arbeit durch die Konzeption und prototypische Implementierung eines Datenanalyse-Tools für Body-Tracking-Kameras geschlossen werden.

1.2 Verwandte Arbeiten

Das Thema technische Werkzeuge zur Analyse von Body-Tracking-Daten wurde bereits in anderen Arbeiten behandelt. Blascheck et al. (2014) behandeln in Ihrer Arbeit „State-of-the-Art of Visualization for Eye Tracking Data“ die Visualisierung von Eye-Tracking-Daten. Dabei stellen sie unterschiedlich Arten der Visualisierung von Eye-Tracking gegenüber. Ebenso standen bereits andere Arbeiten vor der Problemstellung eines aufwendigen Datenanalyse-Prozesses, der durch die Konzeption eines Tools verbessert werden soll. Die Konzeption und Implementierung eines Analyse-Tools für die Auswertung von Log-Dateien wurde dabei von Hampel (2012) schon näher untersucht. In dieser Arbeit wurde eine automatisierte Aufbereitung von Daten entwickelt, die im Zusammenhang mit empirischen Experimenten erhoben wurden. In der Studienarbeit wurde der bisher bestehende Diskrepanz des mehrstufigen, fehleranfälligen und aufwendigen Prozesses der Datenauswertung der Log-Dateien aufgefasst und auf Basis des bestehenden Prozesses ein integriertes Auswerte-Tool konzeptioniert und implementiert, was diese Diskrepanz beheben soll.

Als letzte verwandte Arbeit ist der Artikel von Mäkelä et al. (2017) zu nennen. In dieser Arbeit wird ein halb-automatischer Analyseprozess von Passanteninteraktion mit gesteuerten Public-Displays dargestellt. Dabei wurde zum einen auf den Prototyp des Public

¹Vgl. <https://go.unibw.de/hope>

Displays, sowie den halb-automatischen Prozess zur Analyse der Interaktionsdaten eingegangen. Zum anderen wurden die Ergebnisse der Auswertung und die Vor- und Nachteile des halb-automatischen Analyseprozesses diskutiert.

1.3 Grundlagen und Definitionen

Body-Tracking

Für die Definition des Begriffs *Body-Tracking* ist das direkte Übersetzen sinnvoll. Im Deutschen steht dieser Begriff wörtlich übersetzt für „Körper-Verfolgung“ was die Bedeutung hinter dem Begriff bereits gut trifft. Unter Body-Tracking soll in diesem Kontext das Überwachen von Personenbewegung verstanden werden. Hierzu gehört vor allem die Körperbewegung, sowie die Bewegung innerhalb des Kamerabereichs. Die Person, die sich im Kamerabereich aufhält wird dabei im System als vereinfachtes Skelett dargestellt. Anhand dieser Skelett-Daten in Verbindung mit der Position der Person werden die Daten in Records gespeichert.

Microsoft Kinect v2

Die Microsoft Kinect v2 Kamera spielt in dieser Arbeit eine zentrale Rolle. Sie dient zur Generierung der Daten, die mittels des zu implementierenden Datenanalyse-Tools ausgewertet werden sollen. Die Kinect v2 ist eine Kamera die Farbbilder der integrierten RGB-Kamera, Audio-Daten des Mikrofons und Tiefeninformationen der Tiefen-Kamera liefert (Windows-Developer-Center, 2014, S. 6). Zur Ermittlung der Tiefeninformationen wird das Time-of-Flight (ToF)-Prinzip angewandt. Dabei besitzt die Tiefenkamera einen Infrarot-Sender, sowie einen Infrarot-Sensor, über den das ausgestrahlte Infrarotlicht wieder aufgenommen werden kann. Anhand der Zeitdifferenz zwischen Ausstrahlung und Aufnahme des Infrarotlichts, können Entfernungen ermittelt werden. Diese Entfernungen bieten die Möglichkeit eine Tiefenkarte über den Aufnahmebereich der RGB-Kamera zu erstellen. (Breuer, 2005, S. 3ff) Mittels dieser Informationen ist die Kinect-v2-Kamera fähig Personen zu erkennen und anhand von 25 Skelett-Punkten die menschliche Bewegung zu tracken (Windows-Developer-Center, 2014, S. 7ff). Eine exemplarische Skelett-Aufnahme ist in Abbildung 1.1 dargestellt.

Ambient-Display

Der Begriff *Ambient-Display* tritt häufig mit Begriffen rund um interaktive Großbildschirme im (halb-)öffentlichen Raum auf. Eine Definition des Begriffs wird von Schneider genannt. Dieser bezeichnet Ambient-Display als Bildschirme, die sich in den Alltag integrieren. Sie würden von den Passanten nicht unmittelbar wahrgenommen werden. Im Zentrum stünde dabei vielmehr die dargestellte Information und nicht der Bildschirm an sich. (Schneider, 2011, S. 11, zitiert nach Lipp (2004))

Flying Pixels

Flying Pixels sind fehlerhafte Bilddaten, die in Verbindung mit ToF Kameras auftreten können. Dabei handelt es sich um Fehler in der Tiefenberechnung der Positionen. Diese Fehler treten in der Nähe von Tiefenunterbrechungen auf. Bemerkbar macht sich dieser Effekt darin, dass die Kamera vereinzelt Pixel anzeigt, auch wenn sich an der jeweiligen Position kein Objekt befindet. Die folgende Abbildung 1.2 stellt einen Vergleich zwischen der Kinect-v1 und der

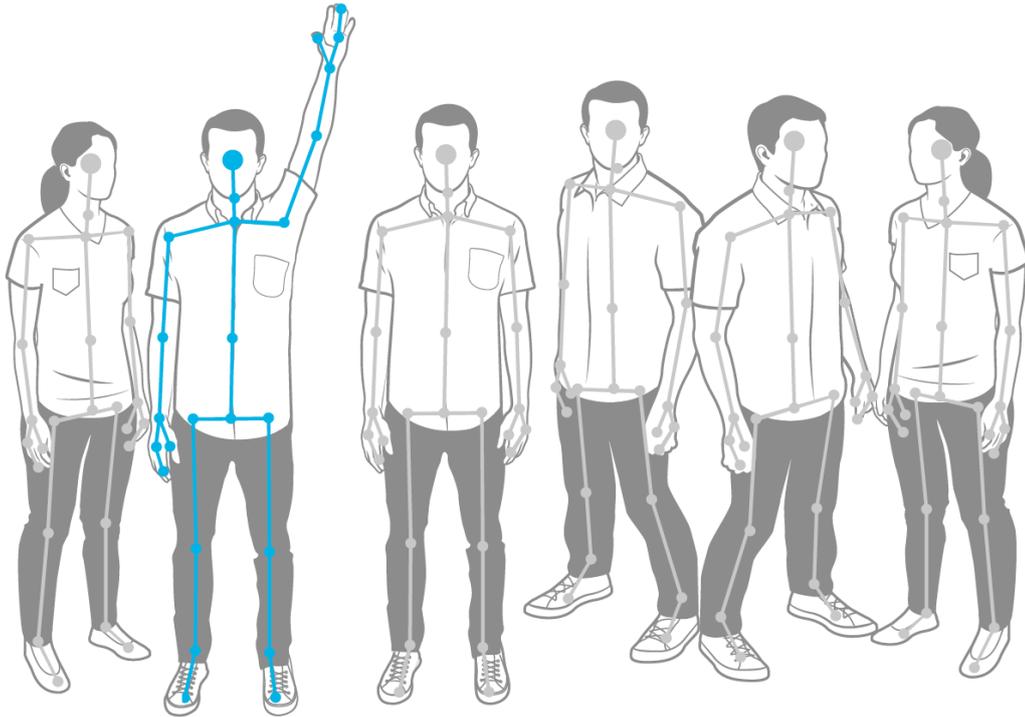


Abbildung 1.1: Skelett-Punkte Microsoft Kinect v2 (Windows-Developer-Center, 2014, S. 8)

hier verwendeten Kinect-v2-Kamera dar. Wie auf der Abbildung zu sehen ist, treten bei der Kinect-v2 deutlich häufiger Flying-Pixels auf, als bei der Kinect-v1. (Wasenmüller und Stricker, 2016, S. 8)

Record

Ein Record ist eine Aufnahme der Body-Tracking-Kamera. Jeder zulässige Record enthält mindestens einen Frame und beginnt beim Eintritt der ersten Person in den Kamerabereich. Der Record endet, wenn die letzte Person den Kamerabereich verlassen hat. In dem verwendeten Datensatz entspricht jeder Record einer Textdatei, deren Name der Zeitstempel des ersten Frames ist.

Frame

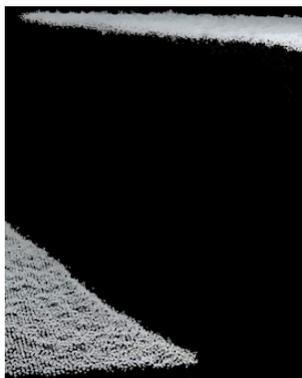
Ein Frame ist ein Log-Eintrag eines Records. Er bildet eine Momentaufnahme der Kamera ab, die im Falle der Aufnahme mehrerer Personen in mehrere Frames untergliedert sein kann. Dabei entspräche jeder Frame einer Person. Die Untergliederung der Werte erfolgt durch das Trennzeichen „###“, wobei die Werte beginnend vom Anfang bis zum Ende folgenden Definitionen zuzuordnen sind:

1. Timestamp: Zeitpunkt der Aufnahme im Format YYYY-MM-DD HH:MM:SS.mmm +HH:MM
2. Framework Body Tracking ID: Identifikationsnummer des .NET Frameworks
3. Record ID: Identifikationsnummer des Records

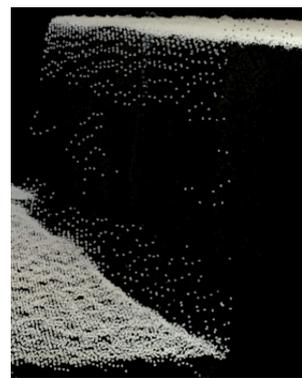
1 Einleitung und Motivation

4. Body Index: Identifikationsnummer der aufgenommenen Person
5. Body Amount: Anzahl der Personen im Bild
6. FaceProperty.Happy: Person glücklich
7. FaceProperty.Engaged: Person zeigt Aufmerksamkeit
8. FaceProperty.WearingGlasses: Person trägt Brille
9. FaceProperty.LeftEyeClosed: Person hat linkes Auge geschlossen
10. FaceProperty.RightEyeClosed: Person hat rechtes Auge geschlossen
11. FaceProperty.MouthOpen: Person hat den Mund geöffnet
12. FaceProperty.MouthMoved: Person bewegt den Mund
13. FaceProperty.LookingAway: Person sieht weg
14. body.LeftHandState: Position der linken Hand
15. body.RightHandState: Position der rechten Hand
16. x: x-Koordinate des Punktes JointPoint.SpineShoulder
17. y: y-Koordinate des Punktes JointPoint.SpineShoulder
18. z: z-Koordinate des Punktes JointPoint.SpineShoulder
19. distance: Distanz zum Punkt JointPoint.SpineShoulder

Ein konkretes Beispiel für einen Frame kann der folgenden Abbildung 1.3 entnommen werden



(a) Kinect v1



(b) Kinect v2

Abbildung 1.2: Vergleich Kinect v1 und v2 (Wasenmüller und Stricker, 2016, S. 8)

1 Einleitung und Motivation

```
2017-07-21 14:45:21.599 +02:00 ### 72057594039042469 ### 1634244614 ### 2 ###  
1 ### No ### No ### Unknown ### Yes ### Yes ### No ### Yes ### Yes ### Unknown  
### Closed ### -0,6855212 ### 0,2279052 ### 4,054336 ### 4,11819358853092
```

Abbildung 1.3: Framesatzeintrag / Frame

2 Analyse

Um ein Datenanalyse-Werkzeug zu konzipieren ist es vorerst notwendig festzustellen, was das Programm leisten können soll. Folgt man dem Wasserfallmodell, entspräche diese Phase der Analyse-Phase. Diese ist die erste Phase des Modells und ist somit das Fundament der Konzeption und Implementierung. Um dieses Fundament zu errichten, werden in Abschnitt 2.1 im ersten Schritt die Anforderungen erarbeitet, die im anschließenden Abschnitt 2.2 in Anwendungsfällen homogenisiert werden.

2.1 Anforderungen

In dem folgenden Kapitel werden die Anforderungen erarbeitet, die ein Daten-Analyse-Werkzeug für Body-Tracking-Daten vorweisen soll, um dessen User einen Nutzen bringen zu können. Dabei wird in einem mehrstufigen Prozess vorgegangen, der an die Struktur der Anforderungserhebung von Broy und Kuhmann angelehnt ist. Nach Broy und Kuhmann (2021) sei die Anforderungserhebung mindestens in die vier Kategorien *Vision*, *Funktion*, *Nutzung* und *Details* zu untergliedern. Diese Struktur wird hier dahingehend abgeändert, dass die Kategorie *Detail* in die Kategorien *Visualisierung* und *Filterung* aufgeteilt wird, da diese Punkte eine zentrale Bedeutung für die Arbeit haben.

Die Anforderungserhebung erfolgt in diesem Fall durch eine Kombination verschiedener Methoden. Neben einem Interview mit einem Experten im Bereich Ambient-Displays wurden Anforderungen durch die Inventurmethode sowie anhand der Brainstorming-Methode erhoben.

2.1.1 Vision

Ein Interview mit Dr. Jan Schwarzer am 07. Oktober 2021, der Experte im Bereich Mensch-Computer-Interaktion ist und sich bereits mit dem Thema Datenauswertung von Body-Tracking-Kameras auseinandergesetzt hat, lieferte Rückschlüsse auf die Vision des Programms. Als *Vision* versteht man die „Mission des Systems“ (Broy und Kuhmann, 2021, 202 ff). Es sollen also die zu erfüllenden Aufgaben des zu implementierenden Programms aufgezeigt werden. Ebenso enthält die *Vision* Angaben zum Kontext des Programms, was eng mit der Motivation der Implementierung verknüpft ist.

Die *Vision* des Analyse-Werkzeugs ist die Beantwortung von Fragen zum Nutzungsverhalten der Passanten der installierten Ambient-Displays. Es soll möglich sein, für den Mensch verständliche Informationen über das Verhalten der Passanten und über die Interaktion mit den Ambient-Displays zu erhalten.

Von Dr. Jan Schwarzer sind vor allem drei zu beantwortende Fragestellungen konkretisiert worden:

- An welchem Punkt im getrackten Raum haben sich Passanten am meisten aufgehalten?

- Aus welcher Position des Raumes gab es die meisten Interaktionen mit dem Ambient-Display?
- Wie erfolgten die Bewegungen der Passanten? Aus welcher Richtung kamen sie und in welche bewegten Sie sich fort?

2.1.2 Funktion

Der Untergliederungspunkt *Funktion* umfasst alle Funktionen, die vom Programm angeboten werden (Broy und Kuhrmann, 2021, S. 202 ff). Dabei müsse der Zusammenhang zu den Aufgaben, die das Programm erfüllen soll, klargestellt werden. Um die Hauptaufgabe des Analyse-Werkzeugs erfüllen zu können, werden vier primäre Funktionen benötigt.

Es wird eine Funktion benötigt, die es dem Nutzer ermöglicht einen ausgewählten Datensatz zu importieren und die Daten somit für das Analyse-Werkzeug verfügbar macht. Dabei werden zwei unterschiedliche Import-Methoden in Erwägung gezogen:

- Über eine Dateiupload-Funktion, bei dem der Nutzer eine ausgewählte, lokal verfügbare Datei per Filepicker auswählen und zur Verfügung stellen kann.
- Über eine API-Schnittstelle zwischen Datenhaltung und Analyse-Werkzeug, die einen ausgewählten Datensatz zur Verfügung stellt.

Um bei der Darstellung der Daten benutzerindividuell Schwerpunkte setzen zu können, ist eine Filterfunktion notwendig. Dabei sollen einerseits Daten gefiltert werden, um Ungenauigkeiten aus den Datensätzen auszufiltern. Andererseits soll der Nutzer in verschiedenen Dimensionen die darzustellenden Daten selektieren oder abgrenzen können, um selbst Schwerpunkte setzen zu können. Auf diesen Filterungsprozess wird in Unterabschnitt 2.1.5 weiter eingegangen.

Weiter wird eine Darstellungsfunktion benötigt. Diese Funktion steht im Mittelpunkt des Programms. Da die Bildverarbeitung mit rund 25 % des Gehirns im Vergleich zu allen anderen Sinnen den größten Anteil im Gehirn in Anspruch nimmt, ist die visuelle Auffassungsgabe des Menschen besser als die Verarbeitung von reinen Daten. Folglich ist es für den Mensch effizienter Daten zu verbildlichen, anstatt sie in Tabellen oder Fließtext darzustellen (Kosslyn, 1985, S. 512). Substituiert auf die Funktion des Analyse-Werkzeugs ist es also von Vorteil, die auszuwertenden Daten zu visualisieren. Da die Analyse der Daten nicht nur auf die Beantwortung einer Fragestellung ausgerichtet ist, sondern einen Gesamteindruck über das Nutzungsverhalten der Passanten geben soll, sind verschiedene Visualisierungsarten anzustreben. Die Wahl der zur Verfügung zu stellenden Arten wird in dem Untergliederungspunkt *Visualisierung* weiter aufgefasst.

Die Sicherungsfunktion ist die letzte Funktion des Programms. Nach fertiggestellter Konfiguration der Visualisierungen soll die erstellte Auswertung mittels der Sicherungsfunktion vorübergehend speicherbar sein, um sie mit anderen Visualisierungen vergleichen zu können. Die Visualisierungen sollen alle zusammen auf einer übersichtlichen Seite dargestellt werden.

2.1.3 Nutzung

Aus dem Gliederungspunkt *Nutzung* sollen die Benutzerschnittstellen hervorgehen. Die *Nutzung* soll erklären, wie der Mensch mit dem System oder Programm interagiert (Broy und

Kuhrmann, 2021, 202 ff). Für die Nutzung des Analyse-Werkzeugs wird der Webbrowser als Benutzerschnittstelle verwendet, da das Programm als Webapplikation implementiert werden soll. Dies ermöglicht zum einen, dass das Programm mit jedem gängigen Betriebssystem (z.B. Windows, MacOS, iOS, Android) kompatibel ist. Zum anderen gibt es dadurch, abgesehen von der Kompatibilität eines Webbrowsers, keine Hardwareanforderungen an das User-Endgerät, um das Analyse-Werkzeug zu nutzen. Die Webapplikation soll primär für Geräte mit Bildschirm-Größe über 10 Zoll optimiert sein. Somit wäre die Nutzung auf Smartphones zwar möglich, jedoch nicht effizient.

2.1.4 Visualisierung

Da das Thema Visualisierung in dieser Arbeit einen bedeutsamen Stellenwert besitzt, werden in diesem Unterpunkt die Anforderungen an die Visualisierungen detailliert erarbeitet. Aus der Erarbeitung soll hervorgehen, wie die Visualisierung umzusetzen ist, damit sie dem Nutzer in angemessener und zugleich ausreichend detaillierter Weise die Informationen vermittelt, die er benötigt. Um erschließen zu können, welche Anforderungen eine Visualisierung besitzt, muss festgelegt sein, wie sich der Wert einer Visualisierung bestimmen lässt. Die Qualität bzw. der Wert einer Visualisierung bestimme sich nach Schumann und Müller (2000) daraus, wie gut das Kommunikationsziel der Darstellung erfüllt wird. Subsumiert auf das vorliegende Thema kann das Kommunikationsziel die jeweilige, zu beantwortende Fragestellung sein. Je nach Erfüllungsgrad der Beantwortung lässt sich somit daraus schließen, ob die Visualisierung wertvoll ist. Neben der Fragestellung sei zusätzlich die Charakteristika der Daten ein Einflussfaktor für den Erfüllungsgrad des Kommunikationsziels. Durch diesen Einflussfaktor entsteht der Effekt, dass eine Visualisierung mit unterschiedlicher Daten-Eigenschaften (z.B. Menge der Dateneinträge, Datentyp, etc.) ebenso unterschiedlich qualitativ eine Fragestellung beantworten können. Folglich hat die Wahl der Visualisierung einen Einfluss auf ihre Zweckerfüllung, ebenso wie die Qualität der Auswertung. Um die richtige Visualisierung für die jeweilige Fragestellung zu finden, benötigt man zuerst eine Übersicht über mögliche Visualisierungen, die dann im Folgenden gegenübergestellt werden können.

Eine solche Übersicht bietet das Unternehmen Tableau. Sie zählen unter anderem die in Tabelle 2.1 aufgeführten Visualisierungsarten auf. Da die Qualität der Visualisierung von dem

Name	Beschreibung	Vorteile
Bar Chart	Balken zum Vergleich von Kategorien	Simpler Überblick / klare Struktur durch Kategorisierung von Daten
Line Chart	Linie zum Darstellen von kontinuierlichen Daten	Darstellung von kontinuierlichen Daten & klare Struktur
Pie Chart	Kreis zum Visualisieren von prozentualen Angaben	Simpler Überblick von Prozentangaben
Maps	Karte zum Darstellen der Häufigkeit eines Ereignisses / Objektes in einer Fläche	Großer Informationsgehalt & Klare Darstellung von räumlichen Variablen
Scatter Plot	Darstellen von diskreten Werten durch Punkte	Möglichkeit der Darstellung von Streuungen durch diskrete Werte

Tabelle 2.1: Auflistung Visualisierungsarten (Tableau-Software, 2021)

Erfüllungsgrad ihres Zweckes abhängt, ist es hierbei sinnvoll unterschiedliche Darstellungsarten zu wählen. Grund hierfür ist, dass sich das verfolgte Visualisierungsziel durch einzelne zu beantwortende Fragestellungen zusammensetzt. Einerseits soll der Nutzer des Daten-Analyse-Werkzeugs einen Überblick über die zu analysierenden Daten bekommen. Dabei ist es vorteilhaft, klar strukturierte Visualisierungen zu wählen, die den Nutzer nicht mit Information überschwemmen. Andererseits sollen die Visualisierungen ebenso detailliertere Auskünfte über das Passanten-Verhalten geben, wobei die Schwierigkeit besteht, dass mit steigendem Detaillierungsgrad die Konzentration der Information steigt. Diese hohe Informationskonzentration kann dazu führen, dass Visualisierungen schwer verständlich und unübersichtlich wirken.

Da bei der Analyse von Body-Tracking-Kameras die Daten über die räumliche Position (x , y , z) definiert sind, bietet sich die Darstellung dieser Daten mittels einer Map an. Betrachtet man die unterschiedlichen Fragestellungen, die sich aus der Vision des Systems ergeben, ist erkennbar, dass sich die Darstellung nicht nur auf eine einzelne Momentaufnahme, also einen Frame beziehen sollen. Die Darstellungen sollen sich auf eine gewisse Zeitspanne t beziehen, wodurch die Darstellung an einer zusätzlichen Dimension gewinnt. Diese zusätzliche Dimension wäre in einer Map durch beispielsweise die Farbe der Darstellung realisierbar. Durch die Kombination der Dimensionen in einer Heatmap wären Aussagen möglich wie 'In der Zeitspanne zwischen t_a und t_b haben sich an x_c , z_d n Personen aufgehalten'.

Schwarzer et al. (ruck) nutzen für die Auswertung der Kameradaten Heatmaps, die mit dem Inhalt auf vier Metriken ausgerichtet waren. Diese Metriken sind inhaltlich und strukturell mit den Fragestellungen dieser Arbeit vergleichbar und werden jeweils durch eine Heatmap mit eigenem Inhalt visualisiert. Aus ihnen kann man im statischen Bild die Positionen auf der x - und z -Achse, sowie das Ausmaß des Wertes anhand der Farbe erkennen. Aufgrund der Verwendung eines Browsers kann man diesen Werten durch Verwendung des Eingabe-Geräts (Maus oder Touchscreen) noch mit zusätzlichen Daten versehen, die der Nutzer auf Nachfrage präsentiert bekommt, die Übersichtlichkeit der Visualisierung jedoch nicht beeinträchtigen, da sie nur auf Nachfrage abgerufen werden. Die Heatmap bietet folglich eine Visualisierungsart, die den Zweck der Visualisierung erfüllen kann.

2.1.5 Filterung

Neben dem Thema *Visualisierung* ist die Filterung von Daten ein wichtiger Bestandteil des Datenanalyse-Tools. Die Filterung ist dafür zuständig, in der Visualisierung lediglich Daten zu berücksichtigen, die den ausgewählten Filter-Parametern entsprechen. Die Filtermethode ermöglicht es dem Nutzer folglich Daten zu selektieren. Dabei sollen vor allem zwei Filterungsarten ermöglicht werden. Zum einen soll es möglich sein, Messwert-Ungenauigkeiten oder zu vernachlässigende Daten der Kamera auszufiltern. Zum anderen soll der Nutzer nach persönlich festgelegtem Schwerpunkt die Daten selektieren können. Im ersten Schritt wird auf die Ausfilterung von ersterer Kategorie eingegangen. Für die Ausfilterung von Messwert-Ungenauigkeiten und zu vernachlässigenden Daten wenden Schwarzer et al. (ruck) einen sechsstufigen Prozess an :

- 1 Entfernen der Records an Wochenenden
- 2 Eingrenzen der Zeitspanne auf 07:00 - 18:00
- 3 Entfernen aller Records mit weniger als 15 Frames
- 4 Entfernen aller Records mit Personen mit 100% Aufmerksamkeit

5 Entfernen von Records mit flying pixels (Sprünge in der x- und z-Achse größer 0,5 m)

6 Entfernen aller Records mit Personen mit 0% Aufmerksamkeit

Da ein möglichst flexibler Einsatz des Datenanalyse-Tools erreicht werden soll, werden diese Parameter nicht fest vorgegeben. Der Nutzer soll die Möglichkeit bekommen, die Filter-Parameter selbstständig nach Bedarf wählen zu können. Um die Entfernung der Records nach Wochentagen und Arbeitszeiten durchführen zu können, wird daher ein Zeitfilter benötigt. Weiter wird ein Filter benötigt, der die Records nach Mindestgröße filtert. Um den vierten und sechsten Prozessschritt anzuwenden wird ein Filter benötigt, der die Records nach Aufmerksamkeit selektiert. Als Letztes bedarf es in dem Analyse-Tool einem Filter für die Flying Pixels. Records, die Flying-Pixels enthalten sind erkennbar an Koordinaten-Sprüngen. Ist die Distanz der Positionen eines Frames zu der Position des darauffolgenden Frames größer als der ausgewählte Wert, müsste dieser Recordteil ausgefiltert werden.

Neben der eben genannten Filterungsart soll der User zusätzlich die Möglichkeit erhalten, die Daten nach ausgewählten Personengruppen zu filtern. Um mögliche Filterungsmöglichkeiten herauszufinden bietet sich das Betrachten eines Log-Eintrags, bzw. Frames an. Einen solchen Frameeintrag, kann man der Abbildung 1.3 entnehmen. Erkennbar ist, dass neben der bloßen Position des Passanten und dem Zeitstempel der Frame-Erstellung, sowie der Identifikationsnummer des Records und des Frames weitere Daten vorhanden sind. Der Frame enthält zusätzlich Daten über die Personeneigenschaften, wie z. B. Happy, Aufmerksam, Linkes/Rechtes Auge geschlossen, Mund geöffnet / bewegt sich, sieht weg, Position der linken / rechten Hand.

Eine mögliche Filterungsoption wäre die Berücksichtigung oder das Ausschließen einer Teilmenge dieser Personen-Eigenschaften. Dem Nutzer könnte es z. B. durch Filtern nach Frames, in denen sich der Mund bewegt, ermöglicht werden, nur Personen anzuzeigen, die sich unterhalten oder essen. Ebenso könnte die Filterung nach Personenanzahl (= viertes Element im Frame) einen Nutzen bringen. Der Filter würde dem Nutzer somit lediglich Records / Frames anzeigen, in denen sich mindestens eine bestimmte Personenzahl aufhält. Weiter wäre es denkbar die Distanzen begrenzen zu können. Dem Nutzer würden folglich nur Personen angezeigt werden, die sich von der Body-Tracking-Kamera mehr oder weniger als den ausgewählten Wert aufhalten.

Eine weitere Filterungsmöglichkeit, die nicht auf direktem Weg aus dem Frame-Eintrag hervorgeht, ist die Filterung nach Personen-Klassen. Die Klassifizierung von Personen könnte in dem Fall nach dem Interaktionsmuster erfolgen. Eine solche Unterteilung nach Interaktionen wurde bereits von Michelis und Müller (2011) durchgeführt, die die Nutzerinteraktion mit dem gestengesteuerten System nach Phasen untergliedern. Diese Phasen werden unterteilt in *passing by*, *viewing*, *reacting subtle interaction*, *multiple interaction* und *follow-up action*. Die jeweiligen Eigenheiten der Personengruppe lassen sich zusammengefasst aus Tabelle 2.2 entnehmen. Wie man anhand dieser Tabelle sehen kann, ist eine vergleichbar feingliedrige Unterteilung der Personenklassen anhand der Body-Tracking-Kameras nicht zweckmäßig und teilweise nicht möglich. Ersichtlich ist, dass die ersten zwei Personenklassen, Passing-By und Viewing and Reacting realisierbar sind, anhand der verfügbaren Einträge des Frames. Jedoch ist die Klassifizierung des Subtle-Users bereits unzweckmäßig, da das Merkmal eines Subtle-Users auf der Interaktion mit einem gestengesteuerten System basiert. Da die Identifikation eines Subtle-Users nicht möglich ist, ist die Einordnung in die darauffolgende Klasse, Direct Interaction ebenso nicht möglich. Dieselbe Überlegung ist bei der Einordnung in die Klasse Multiple In-

Personengruppe	Eigenschaft
Passing By	Alle Personen, die den Aufnahmebereich der Kamera betreten
Viewing and Reacting	Personen, die irgendeine Art der Interaktion zeigen
Subtle Interaction	Personen, die aus einer gewissen Entfernung innehalten und durch Gestikulieren mit dem Bildschirm interagieren
Direct Interaction	Personen, die erst als subtil interagierten und dann durch Betreten des nahen Bereichs (1 m) die Benutzung vereinnahmen
Multiple Interaction	Personen, die nach der Direct Interaction mit einem anderen Gerät interagieren oder nach einer Unterbrechung nochmal mit demselben System interagieren
Follow-up action	Personen die ihre Interaktion mit anderen Personen teilen, z.B. durch Fotografieren

Tabelle 2.2: Phasen von User-Interaktion (Michelis und Müller, 2011, S.16-18)

teraction zu treffen. Die übrig bleibende Klasse Follow-up action benötigt das Erkennen eines fotografierenden Endgeräts oder vergleichbarem, womit diese Kategorisierung ebenso entfällt.

Eine alternative Klassifizierung der Personen in zweckmäßiger Granularität könnte *interagierende* und *nicht-interagierende* Personen sein. Mäkelä et al. (2017) entwickelten angelehnt an der Kategorisierung von Michelis und Müller (2011) eine Entscheidungsmatrix, die anhand dreier Parameter die Passanten in die drei Klassen Direct-, Suble- und Passive-User einordnet. Sofern man auch hier die Kategorie Subtle-User vernachlässigt, bietet diese Entscheidungsmatrix eine Grundlage für die Klassifizierung in interagierende und nicht-interagierende Personen. In dieser Entscheidungsmatrix weisen sie die Personen über die drei Parameter Time in Area, Hover und Trigger ihren jeweiligen Klassen zu. Während Time in Area ein verfügbarer Wert in einem Frame der Body-Tracking-Kamera ist, stehen die Werte Hover und Trigger nicht zur Verfügung. Diese beiden Werte ergeben sich durch gestenbasierte Interaktionen mit dem System.

Dennoch zielen diese beiden Werte darauf ab, die Passanten anhand ihrer Interaktion den Klassen zuzuordnen. Während Passive-User keinerlei Interaktion zeigen müssen, bedarf es zu der Einordnung in die Klassen Subtle oder Direct einem Mindestmaß an Interaktion. Diese Interaktion ist bei der Body-Tracking-Kamera mit der Aufmerksamkeit des Passanten zu vergleichen. Zeigt ein Passant keine Aufmerksamkeit, ist er folglich in die Klasse der nicht-interagierenden Personen einzuordnen. Andernfalls interagiert er mit dem System und ist der dazugehörigen Klasse zuordenbar. Angelehnt an die Entscheidungsmatrix Mäkeläs entsteht die verkleinerte Matrix, die Tabelle 2.3 zu entnehmen ist. Die erarbeitete Entscheidungsmatrix soll bei der Realisierung des Datenanalyse-Tools für Body-Tracking-Kameras dazu dienen, die Passanten in die Gruppe *Interacting* und *Not-Interaction* einzuordnen.

Tabelle 2.4 bietet schließlich ein Zusammenfassung der erarbeiteten Filterdimensionen, die für das Datenanalyse-Tool einen Mehrwert bringen.

	Time in Area	Aufmerksamkeit
Interacting	$\geq 2s$	≥ 2
Not-Interacting	$\geq 0s$	≤ 1

Tabelle 2.3: Entscheidungsmatrix Interacting & Not-Interacting

2.2 Anwendungsfälle

Im vorliegenden Kapitel werden die Anwendungsfälle (auch Use Cases oder Nutzungsfälle genannt) bearbeitet. Die Anwendungsfälle stellen die möglichen Nutzungsszenarien für das Programm dar und bieten somit eine Übersicht über die möglichen User-Interaktionen mit dem zu implementierenden System (Broy und Kuhrmann, 2021, S. 228). Diese werden vorab zur Übersicht in einem Use-Case-Diagramm dargestellt.

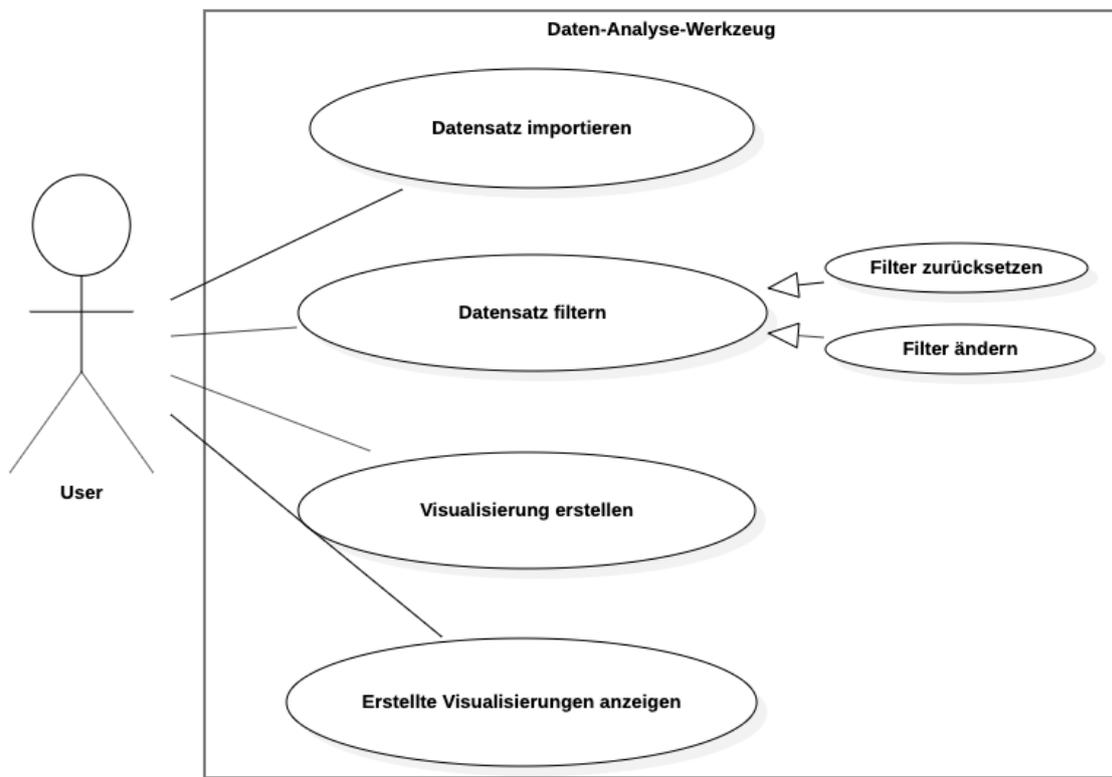


Abbildung 2.1: Use-Case-Diagramm

Das hier aufgeführte Use-Case-Diagramm beinhaltet einen Akteur, den mit dem System verbundenen User. Dieser soll nach dem Starten der App die Möglichkeit haben, Daten von seinem Endgerät zu importieren um sie dem System somit verfügbar zu machen. Ebenso soll er Filter bearbeiten können. Die Filterbearbeitung enthält zwei Ausprägungen. Es soll möglich sein, die Filter zu verändern und die eingestellten Filter auf ihren Standardwert zurückzusetzen, was einer Deaktivierung des Filters gleichkommt. Die wählbaren Filter orientieren sich an den in Tabelle 2.4 aufgezählten Filterdimensionen.

Filterdimension	Beschreibung	Wertebereich
Zeitraumen	Ausfiltern von Records, die nicht innerhalb des ausgewählten Zeitraums erstellt wurden	[00:00; 23:59]
Recordgröße	Ausfiltern von Records, die weniger Frames beinhalten als ausgewählt	[0; ∞ [
Aufmerksamkeit	Ausfiltern von Records, deren Aufmerksamkeit nicht im ausgewählten Wertebereich liegt	[0; 100]%
Flying-Pixels	Ausfiltern von Records, die Teile beinhalten, die in aufeinanderfolgenden Frames Positionssprünge haben, die größer als der ausgewählte Wert sind	[0; ∞ [cm
Distanz zur Kamera	Ausfiltern von Frames, deren Distanz zu Kamera nicht im Wertebereich liegt	[0; ∞ [cm
Personenanzahl	Ausfiltern von Frames, deren Personenanzahl nicht im Wertebereich liegt	[0; 6]
Personenattribute	Ausfiltern der Ergebnisse nach Eigenschaften, die Passanten vorgewiesen haben	Alle verfügbaren Elemente FaceProperty-Elemente der Body-Tracking-Kamera
Personenklasse	Anzeigen / Ausschließen der Nutzer die mit dem System interagieren	{Interact, Not-Interact}

Tabelle 2.4: Zusammenfassung Filterdimensionen

2 Analyse

Ein weiterer Use Case ist die Erstellung einer Visualisierung. Dabei kann der User nach Verfügbarkeit eines Datensatzes die importierten Daten mit der jeweiligen Filtereinstellung visualisiert werden. Dabei soll es verschiedene Möglichkeiten der Visualisierung geben, die der User auswählen kann. Jede Visualisierung soll von dem User gespeichert werden können. Als letzter Use Case ist die Anzeige der erstellten Visualisierungen zu nennen. Dabei kann sich der User die gespeicherten Visualisierungen anzeigen lassen, um sie zu vergleichen.

3 Konzeption

In dem Kapitel Konzeption wird ausgehend von der vorher durchgeführten Analyse das Programm modellhaft entworfen. Dabei geht es zum einen um visuelle Entwürfe des Programms, welche in Form von Wireframes und GUI-Skizzen dargestellt werden. Zum anderen die Programmabläufe modelliert. Im Gegensatz zu der Analyse ist die Konzeption näher an der Implementierung des Programms ausgerichtet, weshalb im letzten Schritt der grobe Aufbau des Datenanalyse-Tools erarbeitet wird, indem die beiden Teilsysteme Frontend und Backend dargestellt werden.

3.1 GUI-Mockup

Das Graphical User Interface (GUI)-Mockup ist ein erster Entwurf des Programms, der darauf abzielt, dem Nutzer oder Kunden vorab das „Look-and-Feel“ zu vermitteln (Böhringer et al., 2014, S.48). Die Funktionalität des Programms ist beim Mockup zweitrangig. In erster Linie sollen der Nutzer erfahren, wie er sich durch das Programm navigieren kann und wie er damit interagiert. Darauf folgend wird das Mockup durch den Nutzer evaluiert und eine weitere Iterationsschleife durchgeführt, bis das *Look-and-Feel* den Ansprüchen des Nutzers entsprechen. Für den Entwurf eines Mockups gibt es Designprogramme wie Adobe XD, die es ermöglichen eine simulierte Anwendung im Baukastenprinzip zu entwerfen. Diese Programme erleichtern das Nachbessern des Mockups nach der Nutzerevaluation, da das Verändern eines Baukasten-Elements mit vergleichsweise wenig Aufwand umsetzbar ist. Hier wurde das Mockup direkt in Hyper-Text-Markup-Language (HTML) umgesetzt und die optischen Anpassungen mit Cascading-Style-Sheet (CSS) vorgenommen, da die Übertragung von Mockup zum Prototyps auf diese Weise mit weniger Aufwand umsetzbar ist.

Betrifft ein User die Seite gelangt dieser auf die Startseite, auf der eine leere Heatmap mit verschiedenen Einstellungsmöglichkeiten und Buttons zu sehen ist. Die Gestaltung dieser Seite ist auf Abbildung 3.1 zu sehen. Das Design ist im Allgemeinen dezent gehalten, um nicht von der Kernfunktionalität, der Visualisierung abzulenken. In dem grau abgesetzten Kasten, mittig der Seite ist eine Tab-Leiste zu finden. Um die Visualisierungsart wechseln zu können, wechselt der User zwischen den dortigen Tabs. Zusätzlich gelangt er darüber zu der Auswahl einstellbarer Filter.

In der rechten oberen Ecke kann der User einen neuen Datensatz hochladen. Dazu öffnet sich ein Modal, welches er per Drag-and-Drop oder über ein Auswahlfenster (File-Picker) mit Textdateien befüllen kann. Die dazugehörige Ansicht ist in Abbildung 3.2 zu finden. Durch einen Klick außerhalb des Modals oder auf den roten Button verlässt der User das Modal wieder und der Daten-Import wird abgebrochen. Durch einen Klick auf den grünen Bestätigungsbutton bestätigt der User den Daten-Import und die ausgewählten Daten werden dem System zur Verfügung gestellt.

Wechselt der Nutzer in den Tab *Filter* gelangt er zu der in Abbildung 3.3 dargestellten Ansicht. Die Filter sind durch unterschiedliche Eingabe-Elemente einstellbar. Dazu gehören

3 Konzeption

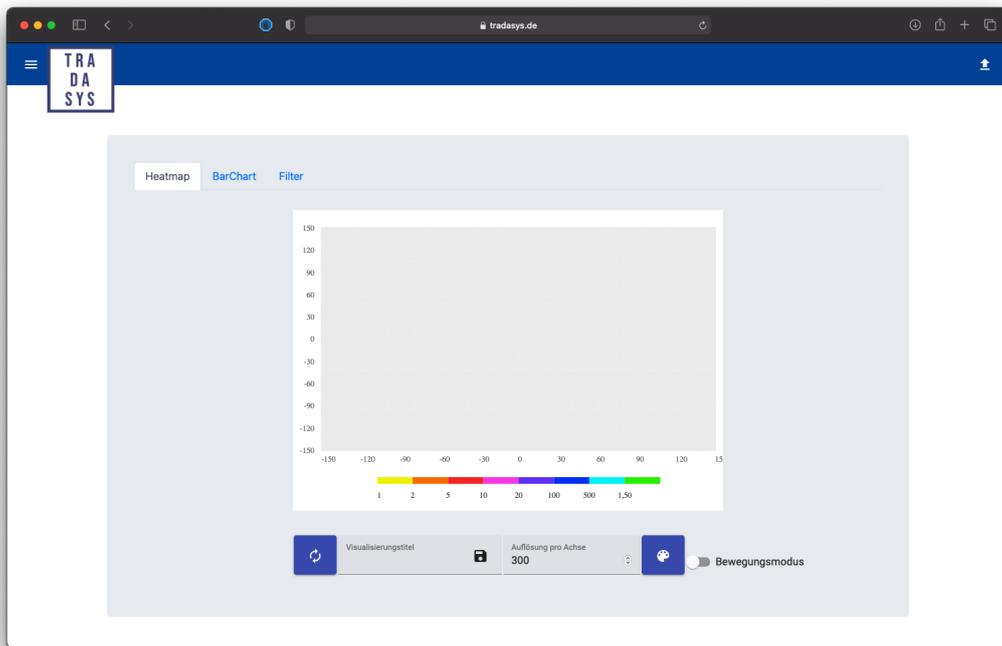


Abbildung 3.1: UI – Startseite

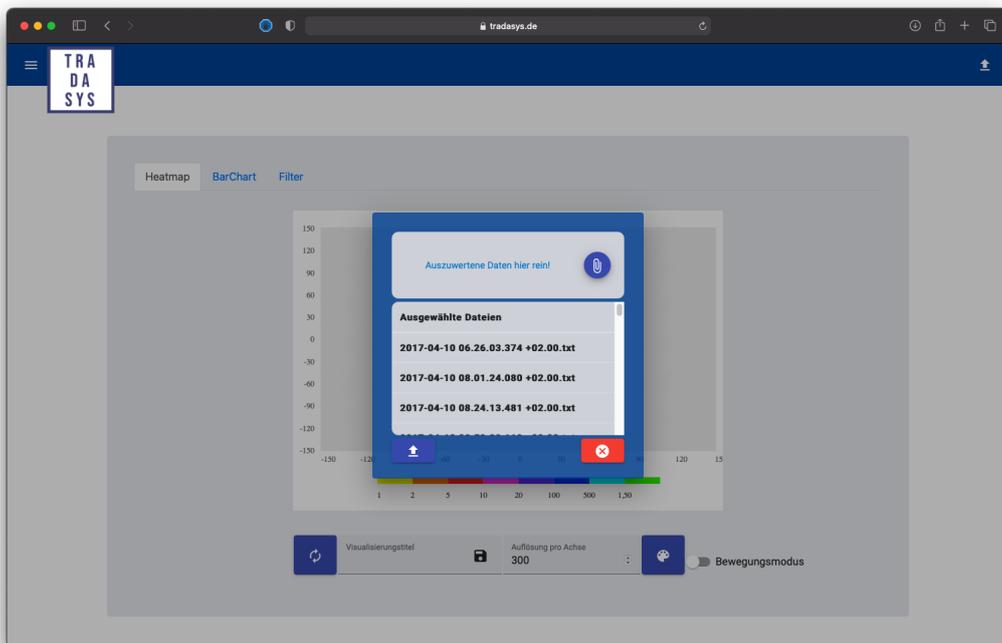


Abbildung 3.2: UI – Daten-Import

3 Konzeption

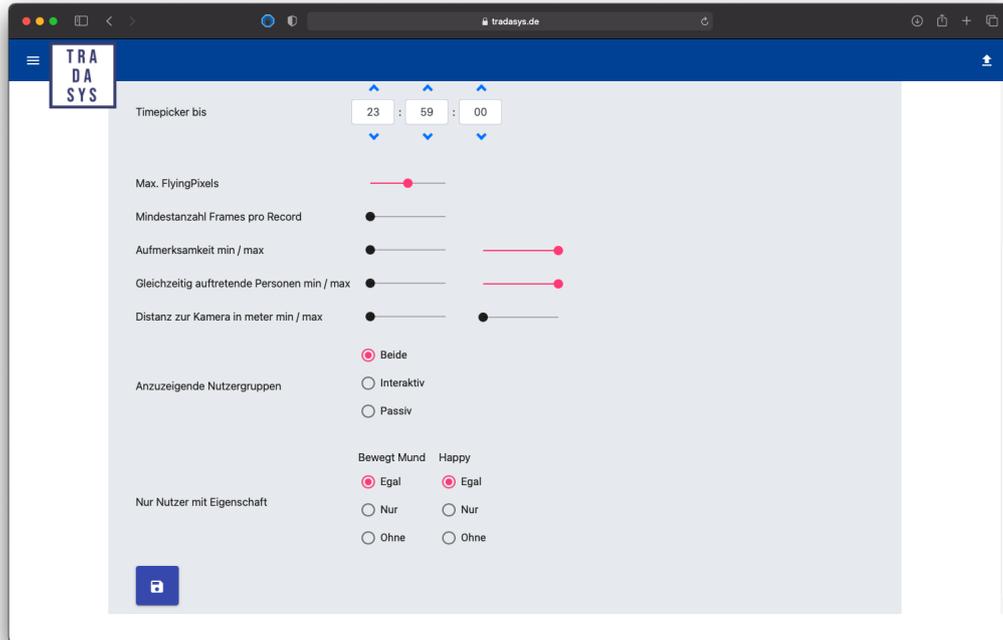


Abbildung 3.3: UI – Filter

Sliders, die dem Nutzer eine Einstellung auf einer bestimmten Skala erlauben. Der ausgewählte Wert wird während der Bearbeitung als Pop-Over über dem Slider dargestellt. Die Slider werden unter anderem für die Eingrenzung der Datensätze nach Flying-Pixels sowie dem Aufmerksamkeitswert der jeweiligen Passanten verwendet (Vgl. Tabelle 2.4). Zudem gibt es für die Einstellung von zeitlichen Parametern einen Time-Picker, der dem Nutzer die Einstellung von Stunden, Minuten und Sekunden erlaubt. Sie sollen dazu dienen, den abzubildenden Zeitbereich der Visualisierung temporär einzugrenzen, sodass man, wie in der Anforderungserhebung festgelegt, nur einen ausgewählten Uhrzeit-Bereich dargestellt bekommt. Als dritte Eingabemethode werden Checkboxes bzw. Radio-Buttons genutzt, über die der Nutzer vorgegebene Parameter an- und abwählen kann. Der Einsatz der Eingabeelemente erfolgt bei der Filterung nach Nutzer-Gruppe sowie -Eigenschaft.

Klickt der User auf das Listensymbol in der linken oberen Ecke, öffnet sich auf der linken Seite eine Navigationsbar. Die Navigation befindet sich im Standardzustand auf „Visualisierung erstellen“. In der Navigationsbar hat der Nutzer die Möglichkeit von dem genannten Standardzustand in den Anzeige-Modus zu gelangen, indem er auf das Navigationselement „Visualisierung anzeigen“ klickt. In dem Anzeige-Modus werden dem User alle gespeicherten Visualisierungen in einer Liste angezeigt, sie sind in diesem Modus jedoch nicht modifizierbar. Die Seite dient dazu alle erstellten Visualisierungen auf einer Seite vergleichen zu können. Die Vergleichsseite ist in Abbildung 3.5 zu sehen. Möchte er wieder zurück zu der Visualisierungserstellung gelangen, kann er erneut den Weg über die Navigationsbar nehmen und „Visualisierung erstellen“ auswählen.

3 Konzeption

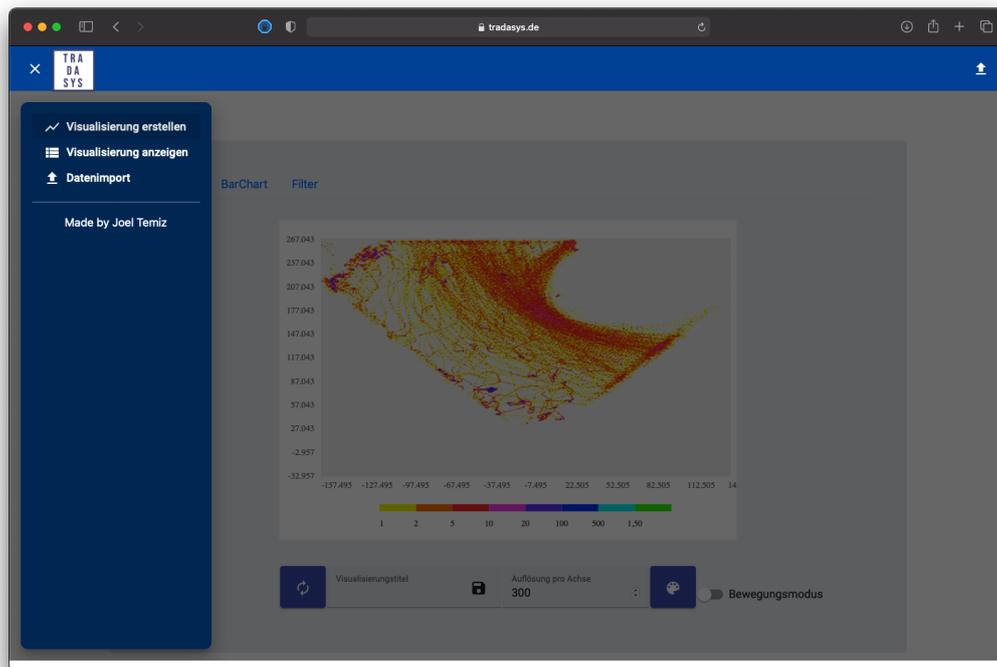


Abbildung 3.4: UI – Navigation

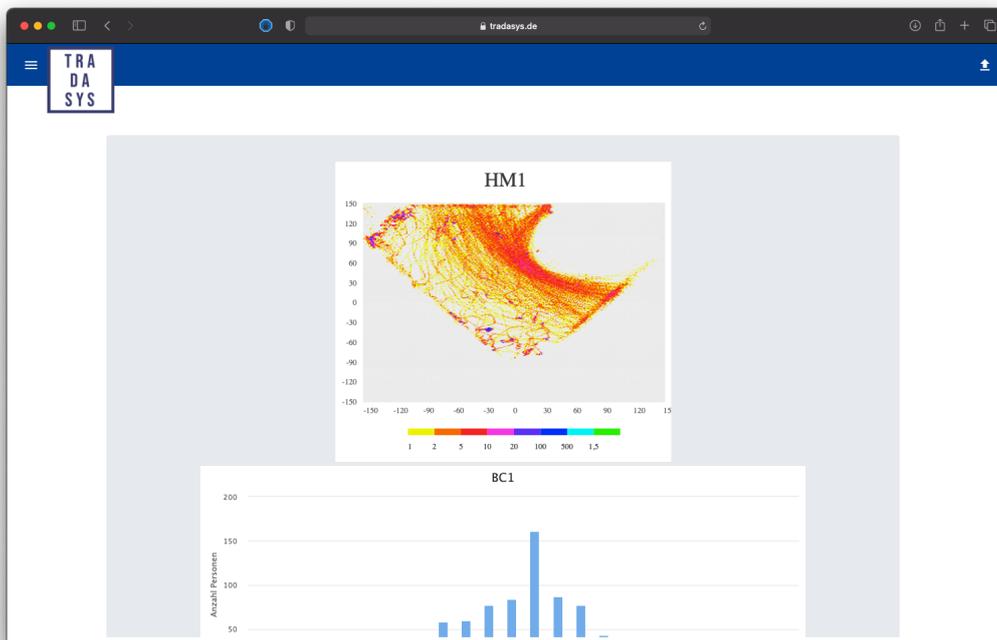


Abbildung 3.5: UI – Visualisierungsvergleich

3.2 Programmablauf

Das Kapitel Programmablauf behandelt die unterschiedlichen Programmsequenzen. Dabei soll dargestellt werden, welche Prozesse innerhalb des Analysetools durchgeführt werden, wenn eine konkrete Situation eintritt, oder eine Methode getriggert wird. Um diese Abläufe modellieren zu können, kann man sich verschiedenen Modellierungssprachen bedienen. Da die Modellierungssprache Business Process Modeling and Notation (BPMN) im Vergleich zu anderen Sprachen, wie EPK und Petrinetzen den Vorteil hat, dass es einerseits leicht verständlich ist und andererseits eine große Werkzeugvielfalt vorweist, wird der Prozess des Datenanalysewerkzeugs in BPMN modelliert (Gronau, 2016).

Für die Modellierung in BPMN stehen verschiedene Grundelemente zur Auswahl. Für die Repräsentation eines Prozessbeteiligten wird der Pool verwendet. Dieser kann durch einzelne Swimlanes unterteilt werden, die jeweils eine Untergliederung des Prozessbeteiligten darstellen. Subsumiert auf das Datenanalyse-Tool stellt der Pool die Gesamtheit des Systems dar. Die Swimlanes repräsentieren dabei den User, das Frontend und das Backend des Programms.

Die Prozesse selbst werden durch Aktivitäten, Events und Gateways dargestellt, die durch Prozess-Flüsse miteinander verbunden sind. Aktivitäten sind dabei die tatsächlichen Prozesse, in denen der jeweilige Prozessbeteiligte eine Methode durchführt. Diese Aktivitäten werden durch andere Aktivitäten oder Ereignisse getriggert. Ein Ereignis kann im BPMN drei Funktionen haben. Es kann einen Prozess auslösen, beenden und steuern. Um diese Funktionen zu erfüllen, kann es verschiedene Ausprägungen, wie eine Nachricht, einen Timer, eine Exception etc. annehmen, wobei nicht jede der Ereignisarten für jede Funktion zulässig ist.

Zusätzlich kann man in BPMN noch, aus Aktivitäten oder Events resultierende Artefakte darstellen. Diese sind dazu geeignet Ein- und Ausgabedaten zu modellieren, die im Zuge eines Prozesses entstehen.

Der Prozess der Auswertung mit dem Datenanalyse-Tool kann wie folgt modelliert werden: Das Startevent entsteht in jedem Fall durch den User und wird durch das Aufrufen der Seite getriggert. Dieses Event führt zum Initialisieren der Seite, wodurch dem User im nächsten Schritt unterschiedliche Möglichkeiten zur Auswahl stehen. Die verschiedenen Interaktionsmöglichkeiten des Users können jeweils durch ein Klick-Event getriggert werden. Hierbei stehen ihm generell vier zur Auswahl: Daten-Import, Daten-Filterung, Visualisierungserstellung und Visualisierungsübersicht. Auf diese wird im Folgenden näher eingegangen.

Daten-Import

Löst der User das Startevent des Daten-Import-Prozesses aus, wird seitens des Frontends ein Modal erzeugt, welches die Eingabe von Daten erlaubt. Bestätigt der User die Dateneingabe werden die Daten vom Frontend als Form-Data zum Backend gesendet. Das Backend filtert diese Daten vorerst nach den jeweiligen Vorab-Filter-Parametern, bevor sie die Daten in das Format eines Dict-Arrays bringt und diese für die weitere Verwendung ablegt.

Daten-Filterung

Wählt der User den Trigger für die Daten-Filterung, wird die Filterseite initialisiert und er kann die Filter modifizieren. Bestätigt er die Modifikation, werden die Filter im Dict-Format an das Backend gesendet, wo sie gespeichert werden. Nach Speicherung der Daten wird der Aktualisierungsprozess der Visualisierung angestoßen, wodurch das Visualisierungs-Modul seine

visualisierten Daten durch die neu gefilterten ersetzt und diese darstellt.

Visualisierungserstellung

Bei der Visualisierungserstellung wird zunächst eine Get-Request an das Backend gesendet, welches die vorhandenen Daten in Form eines Arrays zurückgibt. Diese Daten werden anschließend gespeichert und dem jeweiligen Visualisierungs-Modul übergeben und von ihm dargestellt.

Visualisierungsübersicht

Bei der Visualisierungsübersicht werden zunächst die gespeicherten Visualisierungen aus dem jeweiligen Service geladen. Dieses beinhaltet zum einen die Options-Datei, welche die Chart-Einstellungen beschreibt. Zum anderen wird die zu visualisierende Datenquelle geladen. Nach Fertigstellung des Ladeprozesses wird für jeden gespeicherten Eintrag ein Chart generiert, der die jeweiligen Daten unter den festgelegten Optionen visualisiert.

Da der Prozess unendlich ist, besitzt jeder der aufgeführten Prozesse eine Rückschleife auf das OR-Gateway, an dem der User sich für die verschiedenen Aktionen entscheiden kann. Der Prozess endet nur dann, wenn die Seite durch den User geschlossen wird. Der Gesamtprozess ist nachfolgend als BPMN-Diagramm in Abbildung 3.6 modelliert.

3.3 Teilsysteme

Im Folgenden wird der grobe Aufbau des Datenanalyse-Tools für Body-Tracking-Kameras erarbeitet. Dabei wird die Untergliederung in Teilsysteme des Tools erarbeitet. Wie aus den Programm-Prozessen bereits ersichtlich ist, lässt sich der Datenanalyse-Prozess durch drei parallel agierende Teilprozesse darstellen, die miteinander kommunizieren. Der Teilprozess des Users kann im Folgenden vernachlässigt werden, da dieser lediglich die Triggerevents für die Benutzerschnittstelle ausführt. Da die Benutzerschnittstelle, wie in Unterabschnitt 2.1.3 dargestellt, der Browser sein soll, ist die daraus resultierende Client-Server-Architektur des Programms als gegeben anzunehmen. Die Client-Server-Architektur bzw. das Client-Server-Modell besteht im Wesentlichen aus zwei Einheiten, die miteinander interagieren, dem Server und den Clients. Dabei ist der Client mit dem Server verbunden, hat jedoch keine Verbindung zu den anderen Clients. Die beiden Komponenten bilden zusammen ein System, in welchem eine feste Aufgabenverteilung vorherrscht. (Baun et al., 2015) Wie diese Aufteilung durchgeführt wird, ist für die Leistungsoptimierung der Applikation relevant und kann verschiedene Ausprägungen besitzen. Eine mögliche Ausprägung ist, dass der Server lediglich für die Speicherung des Source-Codes zuständig ist, den er bei einer Request durch den Client übermittelt. Die Logik der Applikation, sowie die Visualisierung würde in dem Fall seitens des Clients liegen.

Eine diametrale Aufgabenverteilung wäre, dass der Client sämtliche Verarbeitungsschritte über den Server anfragen muss. In dem Fall wäre der Server für die Logik der Applikation zuständig und würde auf Request des Clients die jeweiligen verarbeiteten Daten an den Client senden, der folglich darstellt.

Für das Datenanalyse-Tool könnte eine Zwischenlösung sinnvoll sein. Da die Auswertung von Kameradaten über Zeiträume mit großen Datensätzen verbunden sein kann, ist für die Verarbeitung eine Vielzahl an Operationen notwendig. Durch die hohe Anzahl an Operationen würde eine clientseitige Applikationslogik dazu führen, dass Nutzer mit nicht aktueller Hardware der Gebrauch der Anwendung verwehrt bliebe. Ebenso entstünde die Problematik bei

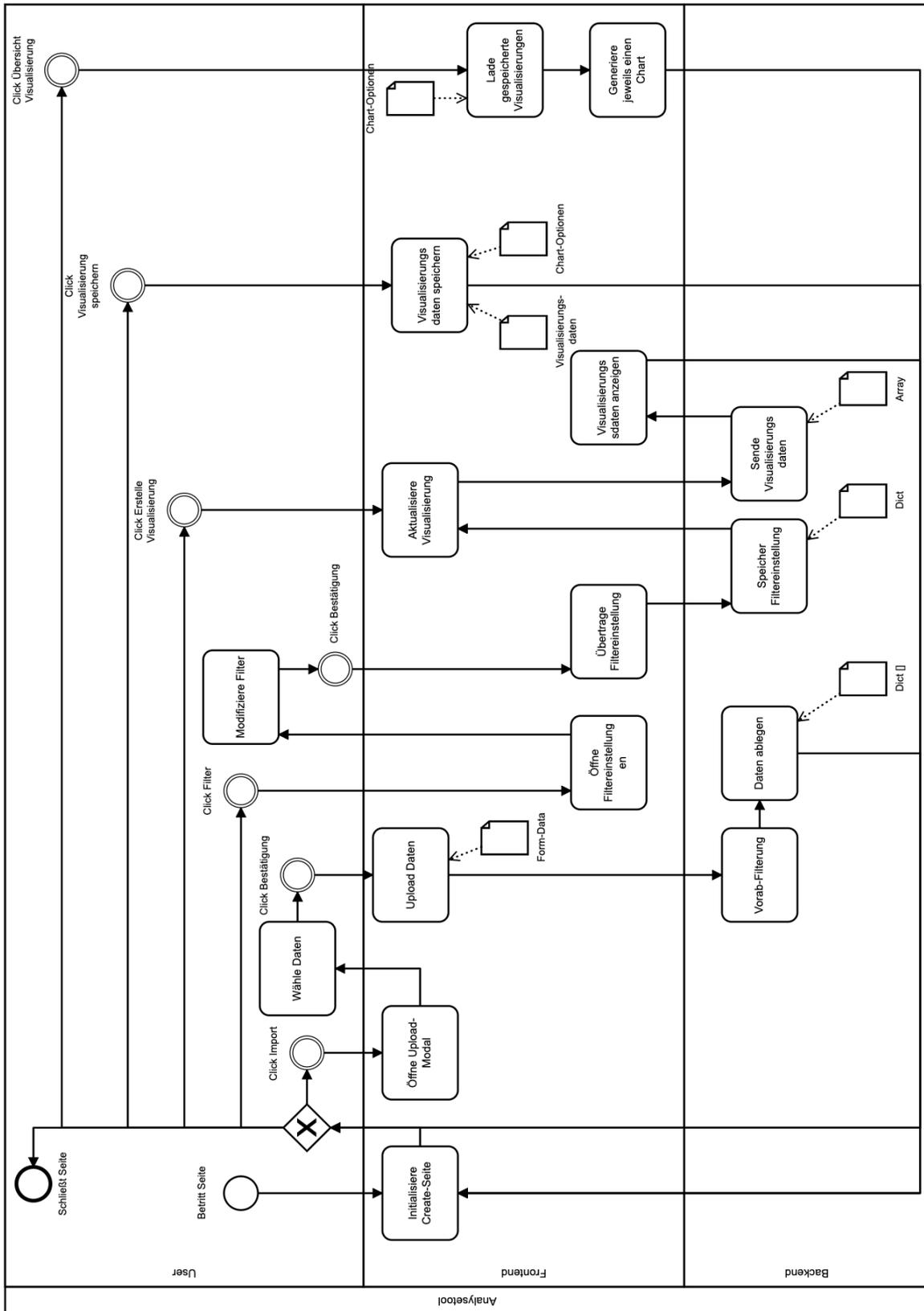


Abbildung 3.6: Programmabläufe – BPMN-Diagramm

3 Konzeption

der rein serverseitigen Verarbeitung der Daten, dass ein großer Datenverkehr herrschte, da der Client für jede Aktion die Daten bei dem Server anfragen müsste.

Für diese Anwendung wird das System in das Teilsystem *Frontend* und das Teilsystem *Backend* gegliedert. Das Frontend wäre hierbei für die Darstellung Seite sowie die Visualisierung der verarbeiteten Daten zuständig. Das Backend wiederum wäre bei dieser Aufgabenverteilung für die Verarbeitung der Datensätze zuständig. Die Verarbeitung beinhaltet dabei die Verarbeitung der Text-Dateien zu Records, die Filterung der Records nach den ausgewählten Filteroptionen, sowie die Verarbeitung der Records zu visualisierbaren Daten. Durch die Aufteilung in Backend und Frontend kann die Verarbeitung und Visualisierung der Daten leistungsoptimiert durchgeführt werden.

4 Implementierung

In dem Kapitel *Implementierung* wird das Vorgehen, der prototypischen Umsetzung der zuvor erlangten Erkenntnisse beschrieben. Dazu wird im ersten Abschnitt auf die Grundlagen der Implementierung des Datenanalyse-Werkzeugs eingegangen. Hierbei werden vor allem die Programmiersprachen des Frontends und Backends und die Entwicklungsumgebung erläutert. In dem darauffolgenden Abschnitt wird der Aufbau der implementierten Anwendung dargestellt. Im Gegensatz zu der in Kapitel 3.3 aufgeführten Untergliederung in Teilsysteme werden hierbei die Programmiersprachen-spezifischen Eigenschaften der Programmstruktur berücksichtigt. Der Abschnitt 4.3 behandelt die Herausforderungen und Schwierigkeiten, die im Zuge der Implementierung aufgetreten sind. Davon ausgehend werden in Abschnitt 4.4 die Beschreibung des entwickelten Software-Codes erläutert. Die Schwerpunkte dabei liegen auf den vorher aufgeführten Herausforderungen und Schwierigkeiten.

4.1 Entwicklungsgrundlagen

In dem Unterkapitel Entwicklungsgrundlagen werden die Rahmenbedingungen zur Implementierung der Anwendung dargestellt. Dazu wird zuerst auf die verwendete Programmiersprache eingegangen.

4.1.1 Programmiersprache

Als eine der Entwicklungsgrundlagen gilt die Programmiersprache zu erwähnen, da diese die Grundlage der Implementierung darstellt. Wie in Kapitel 3.3 dargestellt, wird das System in Frontend und Backend unterteilt. Diese beiden Teilsysteme werden jeweils in verschiedenen Programmiersprachen entwickelt.

Frontend

Das Programm soll, wie in der Anforderungserhebung dargestellt, als Browser- bzw. Webanwendung implementiert werden. Das Webframework Angular ist ein Typescript-basiertes Framework, das aufgrund der Verwendung von *Components* für die modulare, leicht erweiterbare Entwicklung von Webanwendungen gut geeignet ist.

Mit Angular ist es möglich Webanwendungen nach dem Model-View-ViewModel (MVVM)-Pattern zu entwickeln. Dabei wird das *Model* durch die *Services* dargestellt. Services in Angular sind Singleton-Objekte, die dazu genutzt werden, Component-übergreifend Operationen durchzuführen und somit unter anderem auch zwischen Components zu kommunizieren.

Die *View* wird durch die Templates dargestellt. Diese werden in der Auszeichnungssprache HTML geschrieben und mittels CSS oder einer Modifikation dessen (z.B. SCSS, Sass, Less etc.) optisch modifiziert. Die View stellt Inhalte für den Benutzer lesbar dar und ermöglicht ihm die Ein- und Ausgabe von Daten.

Die *ModelView* wird durch das TypeScript-File der Component erzeugt. Sie dient als Bindeglied zwischen Service und Template und bietet die Möglichkeit Benutzerinteraktionen in der View zu steuern, indem den Benutzereingaben Methoden zugewiesen werden. Components können im Gegensatz zu Services instanziiert werden, beispielsweise durch das Öffnen eines neuen Tabs auf derselben URL.

Angular bietet über den Open Source Packageinstaller npmjs die Möglichkeit Module zu importieren, wodurch das Integrieren von bereits implementierten Components mit dem Befehl *npm install* einfach umsetzbar ist. Mögliche Module sind beispielsweise *Highcharts*, die zahlreiche Charts und Visualisierungsmethoden kostenfrei zur Verfügung stellen (Highcharts, 2021). Durch die Verwendung des Frameworks *Angular* kann folglich eine effiziente, leicht erweiterbare und wartungsfreundliche Web-App entwickelt werden.

Backend

Um das Programm möglichst leistungsoptimiert zu gestalten, wird zu dem Angular-basierten Frontend eine Backend-Komponente ergänzt. Das Backend soll auf dem Webserver die abzubildenden Daten verarbeiten und sie in eine visualisierbare Form bringen.

Für die Implementierung des Backends kommt das Python-Framework *Flask* in Betracht. Python bietet über den pip-Installer eine Vielzahl an Open-Source-Bibliotheken, die die Implementierung einer leistungsfähigen App vereinfachen. (Python Packaging Authority, 2021) Ein Beispiel für eine solche Bibliothek ist *Flask-RESTful*, die die Implementierung einer Representational State Transfer (REST) Application Programming Interface (API) mit wenig Aufwand ermöglicht. Diese Bibliothek bietet die Möglichkeit URLs zu definieren, die jeweils auf eine Klasse umleiten. Die jeweilige Klasse kann Methoden beinhalten, die im Falle einer Request (post, put, get, delete) ausgeführt wird und einen definierten Rückgabewert als Response zurückgibt. (FlaskRESTful, 2020) Die Kombination des Frameworks *Flask* und der Bibliothek *Flask-RESTful* ermöglicht somit, die benötigte Rechenleistung auf den Server zu verlagern und die Hardware-Anforderungen an den Nutzer weiter zu minimieren.

4.1.2 Entwicklungsumgebung

Die Entwicklungsumgebung sowie ihre Verfeinerung, die Integrated Development Environment (IDE) ist ein wichtiger Baustein der Entwicklung. Diese beinhaltet eine Sammlung von Werkzeugen, die den Entwickler bei der Implementierung unterstützen. Beispiele hierfür sind Compiler, Interpreter, Debugger, Quelltext-Formatierung und zahlreiche plattformabhängige Plugins (Stefan Luber, 2017). Als Entwicklungsumgebung wurde hier IntelliJ IDEA der Firma JetBrains verwendet.

IntelliJ IDEA bietet unter den rund 6000 auf dem Market-Place verfügbaren Plugins die Erweiterung *Angular and AngularJS*, die bei der Entwicklung Webapplikationen mit dem Framework Angular unterstützen. Unterstützt wird der Entwickler bereits beim Erstellen des Projekts, da das Plugin die notwendigen Dependencies sowie die Grundstruktur des Projektes zu Beginn initialisiert. Zusätzlich wird die Unterstützung durch das automatische Speichern und Neu-Kompilieren zuteil, da Änderungen unmittelbar übernommen werden, sodass der Entwickler nach Beenden des Tippens direkt in den Browser wechseln kann und seine Änderungen übernommen wurden. (JetBrains, 2021)

4.2 Programmaufbau

In dem Kapitel Programmaufbau wird die allgemeine Programmstruktur dargestellt. Die Modellierung der Architektur erfolgt in Form eines Klassendiagramms. Gleich wie das Use-Case-Diagramm ist das Klassendiagramm der Modellierungssprache Unified Modeling Language (UML) zuzuordnen. Es bildet die Struktur eines Programms, wodurch es auch als „architekturelle[s] Rückgrat“ der Modellierung zu werten ist (Rumpe, 2004, S. 41). Klassendiagramme bestehen im Wesentlichen aus fünf Kernelementen, den Klassen, Attributen, Methoden, Assoziationen und Kompositionen.

In der Modellierung wird die Klasse in drei Felder untergliedert, wobei dem ersten der Klassenname, dem zweiten die Attribute und dem dritten die Methoden zugeordnet werden. Die Funktion der Attribute und der Methoden entspricht der der Programmierung. Die Attribute sind dabei für die Speicherung von Daten und Zuständen verantwortlich, wohingegen Methoden Operationen oder Aufgaben erledigen. Ein weiteres Element von Klasse ist die Vererbung. Diese legt fest, dass eine Unterklasse die Methoden und Attribute ihrer Oberklasse erben und erweitern kann. Zusätzlich kann die Unterklasse Methoden der Oberklasse überschreiben. Kennlich gemacht wird diese Vererbungsbeziehung durch einen Pfeil von der Unterklasse zur Oberklasse.

Um verschiedene Klassen gemeinsam in Beziehung zu setzen, bedient man sich im Klassendiagramm Assoziationen. Diese werden durch einen einfachen Pfeil dargestellt und können sowohl Bezeichnung, als auch eine Kardinalität besitzen. Soll dargestellt werden, dass eine Klasse nur während des Existierens einer anderen Klasse existiert, so kann diese *starke Abhängigkeit* durch eine Komposition gekennzeichnet werden. Diese wird durch eine Raute an der Assoziation markiert.

Eine Teilmenge dieser Elemente ist in dem Klassendiagramm des Datenanalyse-Tools zu finden. Da das System im Wesentlichen aus zwei Komponenten besteht, wurden diese jeweils durch ein Subsystem modelliert, die im Ganzen miteinander kommunizieren. Angefangen bei dem Frontend, wird die AppComponent beim Betreten der Seite initialisiert. Diese bildet mit ihrem Template das Rahmenwerk des Frontends. Sowohl die ComparismComponent, als auch die CreateComponent besitzen eine starke Abhängigkeit von der AppComponent. Beim Betreten der Seite muss zwangsläufig erst die AppComponent initialisiert werden, bevor die genannten Komponenten erstellt werden können. Ebenso werden sie nach Verlassen der Seite wieder zerstört.

Der Zweck der CreateComponent ist das Erstellen neuer Visualisierungen. Sie soll daher sowohl eine Assoziation zu der Oberklasse ChartComponent, als auch zu der FilterComponent besitzen. Die ChartComponent ist dabei für die Erstellung von Chart-Objekten zuständig, die jeweils von ihren Unterklassen Heatmap- und BarChart-Component als Bibliothek importiert werden. Für die Speicherung der erstellten Visualisierungen kommuniziert die ChartComponent mit dem ComparismService, wo die zu speichernden Daten Charts abgelegt werden. In der ChartComponent wird die abstrakte Methode updateChart() definiert, die von jeder Unterklasse jeweils überschrieben werden muss. Durch diese Generalisierungsbeziehung zwischen den Chart-Ausprägungen und der ChartComponent ist die Implementierung eines zusätzlichen Visualisierungstyps mit wenig Aufwand umsetzbar.

Die FilterComponent hat die Aufgabe die Filter zu verwalten. Dieses beinhaltet einerseits die Eingabemöglichkeit der Filtermodifikationen, als auch die Weiterleitung der durchgeführten Modifikationen zum Backend.

Die ComparismComponent ist für die Darstellung gespeicherter Visualisierungen zuständig.

Dazu besitzt sie eine Assoziation zu dem `ComparismService`, der als Bindeglied zwischen der `Create-` und der `ComparismComponent` dient. Bei Initialisierung der `ComparismComponent` greift sie auf ihren Service zu um, die gespeicherten Charts zu erhalten und diese zu darzustellen.

Für die allgemeine Kommunikation zwischen den Frontend-Components und dem Backend-Parser ist der `DataParserService` implementiert. Dieser sendet die importierten Daten an das Backend und empfängt dessen Rückgabe-Werte.

Das in Python-Flask implementierte Backend `BackendParser` ist für die leistungsoptimierte Verarbeitung der Eingabedaten zu visualisierbaren Datentypen zuständig. Dazu besitzt es vier Klassen, die jeweils `get()`, `put()`, `post()` und `delete()`-Methoden besitzen. Während die `ImportApi`-Klasse für das Importieren der Datensätze zuständig ist, verarbeiten `Heatmap-` sowie `BarChartApi` die Daten zu, für die jeweiligen Chart-Typen visualisierbare Daten. Die `FilterApi` ist für die Filterung der Daten nach den jeweils eingegebenen Parametern zuständig. Das UML-Diagramm in Abbildung 4.1 zeigt die Struktur des Datenanalyse-Tools visuell.

4.3 Herausforderungen und Schwierigkeiten

Bei der Umsetzung des Datenanalyse-Tools für Body-Tracking-Daten entstanden vereinzelt Herausforderungen und Schwierigkeiten, auf die im Folgenden eingegangen wird.

4.3.1 Programmkomplexität

Generell lag eine allgegenwärtige Schwierigkeit in der leistungsoptimierten Umsetzung der Anwendung. Die Datenerhebung erfolgte mit einer großen Zahl an Datensätzen. Ein zum Testen zur Verfügung gestellter Datensatz beinhaltete Textdateien mit einer Gesamtgröße von 8,52 GB, die die Daten zweier Body-Tracking-Kameras über einen Zeitraum von vier Monaten (07.04.2017 - 04.08.2017) aufzeichnen. Die Tracking-Daten wurden in Hinblick auf ihren Umfang mittels eines Python-Scripts ausgewertet.

```

1 def sum_frames():
2     frameAmount = 0
3     for root, subdirs, files in os.walk(rootdir):
4         for file in files:
5             frameAmount += sum(1 for line in open(os.path.join(root, file)))
6     return frameAmount
7
8 def sum_records():
9     recordAmount = 0
10    for root, subdirs, files in os.walk(rootdir):
11        recordAmount += len(files)
12    return recordAmount

```

Listing 4.1: Mengenermittlung Records / Frames

Die Auswertung des Datensatzes ergab, dass dieser 97.626 Records mit durchschnittlich je 355 Frames beinhaltet, wodurch eine Gesamt-Frameanzahl von 34.687.630 entsteht. Durch die Größe dieses Datensatzes würde der Rechenaufwand vor allem in zwei Prozessen der Datenanalyse zu groß für eine Client-basierte Webanwendung, dem Dateiupload und der Darstellung in Heatmaps.

Bei diesen Prozessen ist das systematische Iterieren über alle Frames unumgänglich, da das Überspringen einzelner Frames oder Records die Darstellung verfälschen würde. Daher muss

4 Implementierung

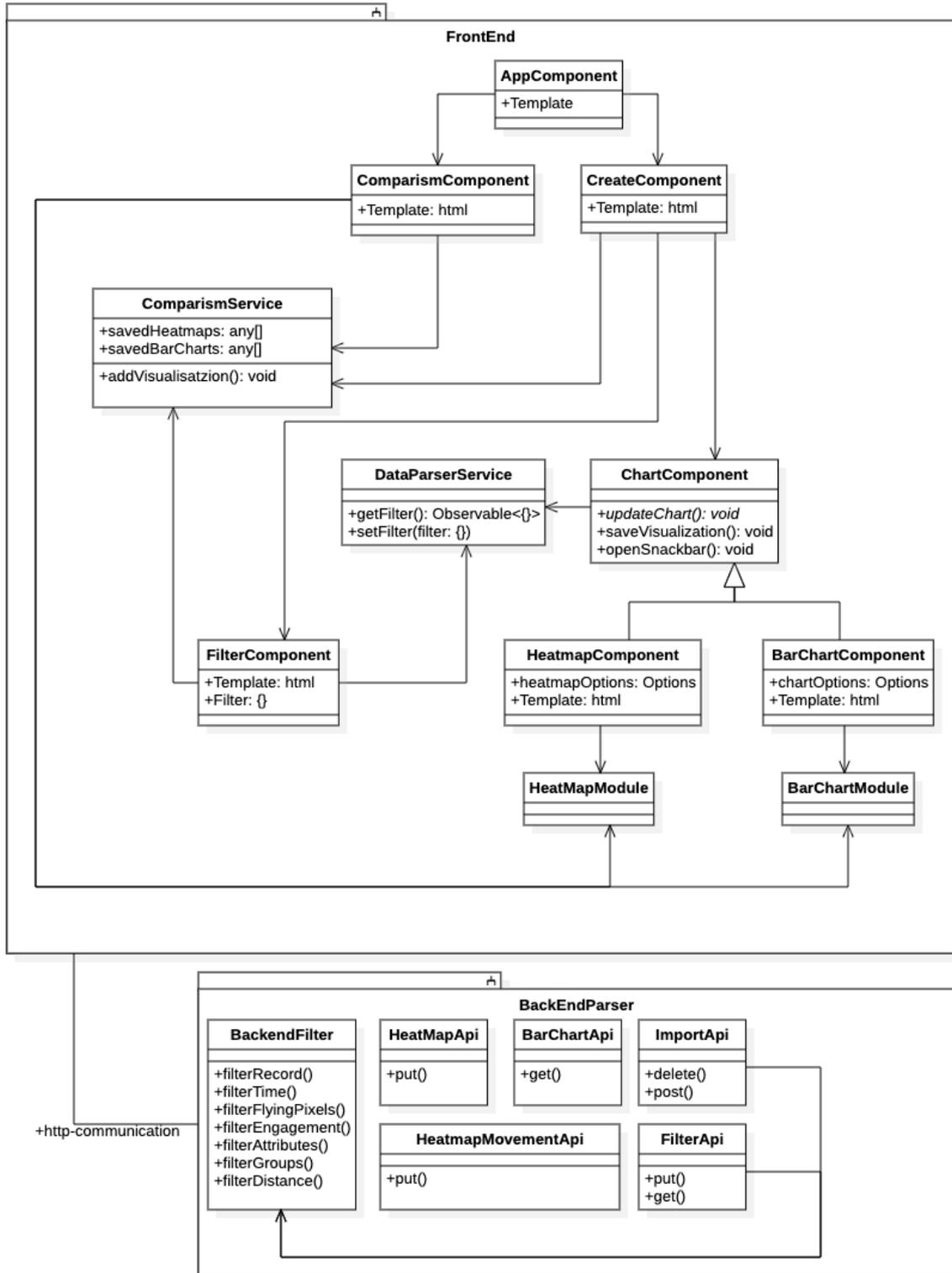


Abbildung 4.1: Programmaufbau – UML-Diagramm

die jeweilige Operation für jeden hochgeladenen bzw. hochzuladenden Frame durchgeführt werden.

Um die Durchlaufzeiten zum Zwecke der Benutzerfreundlichkeit möglichst gering zu halten, war bei diesen beiden Prozessen daher die Komplexität der Operation eine zu beachtende Komponente.

4.3.2 Filteralgorithmen

Eine weitere Herausforderung in der Implementierung des Datenanalyse-Tools war die Umsetzung der angebotenen Filter. Da die Filter für den Nutzer flexibel, für den Entwickler wartungsfreundlich und zusätzlich für das System simpel berechenbar sein sollen, war für die Implementierung eine ganzheitliche Konzeption über alle Filterdimensionen notwendig. Auf welche der Anforderungen der Schwerpunkt gesetzt werden soll, musste im Zuge der Implementierung entschieden werden.

4.3.3 Berechnung der Heatmap-Daten

Die Berechnung der Heatmaps erwies sich bei der Implementierung als eine Herausforderung. Die Herausforderung bestand darin, einen wenig komplexen Algorithmus zu entwickeln, der die x- und z-Positionen der Frames einer Position in einem zweidimensionalen Array zuordnet. Die Positionen der Frames sind als Fließkommazahlen dargestellt, während der Eintrag eines Records jeweils für einen Pixel-Punkt in der Heatmap steht. Jedem Eintrag des Arrays muss ein Integer-Wert $\epsilon [0; n]$ zugeordnet werden, der die Farbe in der Heatmap darstellt. Da auch hier die flexible Einsetzbarkeit des Programms einen sehr hohen Stellenwert besitzt, soll die Größe der Darstellung nicht statisch, sondern auf die Randpositionen (äußerste Punkte aller Records) angepasst sein.

4.3.4 CORS-Policy

Ebenso sind Schwierigkeiten mit der CORS-Policy an dieser Stelle als Herausforderung aufzuzählen. Um zwischen den Komponenten Frontend und Backend kommunizieren zu können wurde das HTTP-Protokoll gewählt. Somit wurden bei der Durchführung einer Operation, der im Zusammenhang mit der Datenverarbeitung steht, eine post-, put-, get- oder delete-Request an das Python-Backend gesendet, das die jeweilige Anfrage bearbeitete und eine Response zurückgab. Diese Kommunikation erwies sich vor allem auf dem Development-Server als unzuverlässig. Nach einer unbestimmten Nutzungsdauer ergaben sich Ausfälle bei dem Versenden der post-Request, die für die Übermittlung der Datensätze vom Frontend an das Backend zuständig ist. Bemerkbar wurde dies dadurch, dass der Prozess zwar gestartet wurde, allerdings keine Request am Backend ankam, wodurch der Prozess so lange anhielt, bis er durch einen Time-Out abgebrochen wurde. Dieser Fehler trat sporadisch auf und war nach mehreren Minuten wieder verschwunden.

Auf selber Weise sporadisch traten Fehler auf, die durch die Cross-Origin Resource Sharing (CORS) verursacht wurden. Die CORS-Funktion dient dazu, den Browser darüber zu informieren, dass ausgewählte Ressourcen der derzeit geöffneten Website an einen dritten Server geteilt werden. Tritt dieser Fall ein, handelt es sich um eine Cross-origin Request, die durch CORS kontrolliert wird. Abbildung 4.2 stellt diesen Prozess bildlich dar.

Wird ein solches Ressourcen-Sharing verwendet, wie in dem hier vorliegenden Fall, um den Leistungsbedarf auf ein Backend zu verlagern, hat der Entwickler die Möglichkeit den zu sen-

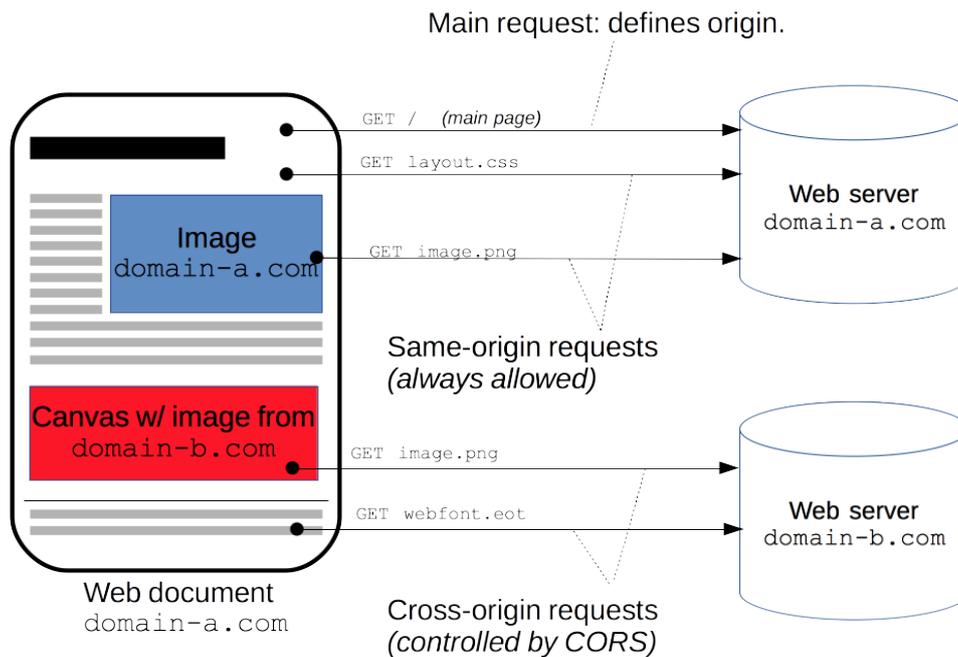


Abbildung 4.2: CORS-Prinzip (MDN-Mozilla, 2021)

denden Request einen Header mitzugeben, die das Blockieren der Request durch die CORS verhindern sollen. In der Implementierung des Datenanalyse-Tools trat es sporadisch auf, dass eine Request durch die CORS-Policy blockiert wurde. Daraus resultierte die folgende Fehlermeldung:

```
Failed to load resource: Origin https://tradasys.de is not allowed by Access-Controll-Allow-Origin.
```

Die Request wurde danach folglich abgebrochen und der Daten-Import schlug fehl. Nach einer erfolgten Fehleranalyse konnten einige Fehler behoben werden, indem den Request *Headers* angehängen wurden. Dennoch trat der Fehler überwiegend beim Daten-Import weiterhin auf. Nach Rücksprache mit dem Hostinganbieter konnte der Fehler darauf eingegrenzt werden, dass der Web-Hosting-Server eine Begrenzung der Body-Size hat. Beim Datei-Upload wurde diese Begrenzung bei einem Upload von Dateien mit mehr als 100 MB überschritten, wodurch sie durch den Server blockiert werden. Eine Lösung des Problems kann folglich nur durch den Wechsel des Hosting-Servers erfolgen. In der Realisierung des Prototyps wurde zur Steigerung der Benutzerfreundlichkeit die importierbare Datengröße auf 100 MB begrenzt.

4.4 Codebeschreibung

Für die soeben aufgezählten Herausforderungen wurden größtenteils Lösungen entwickelt und umgesetzt. Die Lösungen bestanden selten in groben Änderungen der Konzeption. In den meisten Fällen genühten für die Fehlerbehebung einzelne Änderungen in Methoden oder Algorithmen. Auf die Umsetzung der letztendlich zielführenden Implementierungsstücke wird im folgenden Teil eingegangen. Dabei wird jeweils die vorher aufgeführte Herausforderung aufgefasst und ihr Lösungsansatz dargestellt.

4.4.1 Programmkomplexität

Wie in Abschnitt 3.3 angemerkt wurde die hohe Programmkomplexität durch die Hinzunahme eines Backends kompensiert. Durch das Backend wurde es ermöglicht den Rechenaufwand, der bislang auf der Seite des Clients lag, auf den Server zu verteilen. Dazu wurden vom Client lediglich die Textdateien importiert und in ein Form-Data geladen. Das Form-Data mit der Liste an Dateien wurde im Anschluss per post-Request an das Backend gesendet. Im Einzelnen wurde der Prozess des Sendens durch die Methode *transportFile()* repräsentiert. In ihr wurde im ersten Schritt geprüft, ob der verfügbare Dateibuffer *output* leer ist. In diesem Fall ist keine weiterführende Operation notwendig. Ist dies nicht der Fall, werden über eine for-each-Schleife alle gebufferten Daten in ein Form-Data geladen, welches eine Liste der Daten repräsentiert. Nach Abschluss der Schleife wird dieses ans Backend gesendet. Hierzu wird zuerst eine delete-Request gesendet, die die bestehenden Daten des Backends entfernt. Nach Empfang der Response wird im nächsten Schritt das Form-Data übertragen. Die Verschachtelung der subscribe-Methoden ist dabei für die Wahrung der Chronologie erforderlich. Werden die subscribe-Methoden parallel zueinander ausgeführt, kann es, abhängig von der Dauer der jeweiligen Request, dazu kommen, dass der importierte Datensatz durch die delete-Request wieder gelöscht wird.

```

1 public transportFile(): void {
2   if (this.output.length == 0) {
3     this.openSnackBar('Keine_Datei_ausgewaehlt.');
```

```

4   } else {
5     try {
6       let formData = new FormData()
7       for (let file of this.output) {
8         formData.append('files[]', file, file.name)
9       }
10      this.dataParserService.deleteData()
11      .subscribe(success => this.dataParserService.importData(formData)
12      .subscribe(receiveMessage => this.openSnackBar(receiveMessage)));
13    } catch (e) {
14      this.openSnackBar(e);
15    }
16  }
17 }
```

Listing 4.2: Frontend – Datenupload

Der Empfänger dieser Request ist das Backend. Dazu wird die Request durch die verwendete Bibliothek *flask-restful* auf die Methode *post(self)* geleitet. Um die Request verarbeiten zu können, wird im ersten Schritt dem Request Parser der Parameter *files[]* übergeben, unter dem das übermittelte Form-Data erreichbar ist. Über den Übergabeparameter *action = 'append'* wird dem Request Parser mitgeteilt, dass es sich um ein Iterable-Objekt handelt, wodurch das Datenobjekt als Liste behandelt werden kann. Im nächsten Schritt wird über das Listenobjekt iteriert. In jedem Iterationsschritt wird die jeweilige Datei gelesen und von Bytes in das UTF8-Format gebracht. Daraufhin werden die Dateien über die Methode *parseFileToRecord(fileString)* verarbeitet.

```

1 def post(self):
2   global uploadedFiles
3   parser = reqparse.RequestParser()
4   parser.add_argument('files[]',
5                       type=werkzeug.datastructures.FileStorage,
```

4 Implementierung

```
6         location='files', action='append')
7     args = parser.parse_args()
8     files = args['files[]']
9     for file in files:
10        filestring = file.read().decode('utf8')
11        uploadedFiles.append(filestring)
12        parseFileToRecord(filestring)
13    return str(len(uploadedFiles)) + '_Records_mit_je_durchschnittlich_' + str(
14        int(len(flatStoredRecords) / len(uploadedFiles))) + '_Frames_hochgeladen', 200
```

Listing 4.3: Backend – Datenupload

Die Methode *parseFileToRecord(fileString)* ist für die Verarbeitung der Strings in ein Dictionary-Format, sowie die Filterung zuständig. Der Übergabeparameter ist vom Datentyp String und soll für diesen Zweck immer ein Record bzw. eine Textdatei sein. Zur Verarbeitung zu Dictionaries wird der Übergabeparameter zuerst nach Zeilenumbrüchen geteilt und in einem String-Array zwischengespeichert. Die Frames des Records werden dann als Array-Element repräsentiert, die im nächsten Schritt erneut nach dem Trennzeichen „###“ geteilt werden. Nach Fertigstellung der Schleife entsteht somit ein zweidimensionaler Array, der Strings beinhaltet, bei dem jedes Element [i][j] den Framewert an Position *j* im Frame *i* besitzt.

Im nächsten Schritt wird jeder Array *i* anhand der jeweiligen Werte *j* zu einem Dictionary formatiert zu der lokale Variable *tmp* hinzugefügt. Nach Abschluss der Formatierung des Records wird *tmp* nach den ausgewählten Parametern gefiltert. Die gefilterten Records werden als letzten Schritt zu dem globalen Array *storedRecords* hinzugefügt.

```
1 def parseFileToRecord(fileString):
2     global storedRecords
3     global flatStoredRecords
4     frames = fileString.split('\r\n')
5     tmp = []
6     if len(frames) < BackendFilter.filterConfig['minFrameAmount']:
7         return
8     # iterate through array of Frames and split each Frame on '###' to get 19 single values
9     for frame in frames:
10        # split text-file on each linebreak --> Array of Frames
11        valArray = frame.split('_###_')
12        if len(valArray) > 1:
13            frameDict = {
14                'time': datetimeMethod.strptime(valArray[0][:7], '%Y-%m-%d_%H:%M:%S.%F'),
15                'frameWorkBodyTrackingId': valArray[1],
16                'recordId': valArray[2],
17                'bodyIndex': int(valArray[3]),
18                'bodyAmount': int(valArray[4]),
19                'Happy': valArray[5],
20                'Engaged': valArray[6],
21                'Glasses': valArray[7],
22                'LeftEyeClosed': valArray[8],
23                'RightEyeClosed': valArray[9],
24                'MouthOpen': valArray[10],
25                'MouthMoved': valArray[11],
26                'LookingAway': valArray[12],
27                'HandLeftState': valArray[13],
28                'HandRightState': valArray[14],
29                'xAxis': float(valArray[15].replace(',', '.')),
30                'yAxis': float(valArray[16].replace(',', '.')),
31                'zAxis': float(valArray[17].replace(',', '.')),
```

```

32     'Distance': float(valArray[18].replace(',', '.')),
33 }
34 # append the dict to the temporary variable tmp
35 tmp.append(frameDict)
36 # when all frames of the record are appended temporary record tmp: filter record
37 filteredRecord = BackendFilter.filterRecord(tmp)
38 # if result of filtering is no empty array: append it to the global storedRecords
39 if len(filteredRecord) > 0:
40     storedRecords.append(filteredRecord)
41     flatStoredRecords.extend(filteredRecord)
42 return

```

Listing 4.4: Parsing-Methode für Erstellung von Records

4.4.2 Filteralgorithmen

Die in *parseFileToRecord(fileString)* verwendete Methode *filterRecord(record)* ist für die Filterung der generierten Records nach den eingestellten Filterdimensionen zuständig. Dabei wird eine Abfolge von einzelnen Filtermethoden ausgeführt, die sich jeweils den vorher gefilterten Record übergeben. Die aufgerufenen Methoden sind:

1. *filterTime(record)* zum Ausfiltern von Records, die außerhalb der gewünschten Zeitspanne liegen.
2. *filterFlyingPixels(record)* zum Ausfiltern von Positionen, die durch Flying-Pixels entstanden sind
3. *filterEngagement(record)* zum Ausfiltern von Personen, deren Aufmerksamkeit nicht im gewünschten Bereich liegt
4. *filterAttributes(record)* zum Ausfiltern von Personen, die nicht die gewünschten Attribute erfüllen
5. *filterGroups(record)* zum Ausfiltern von Personen, die nicht in die ausgewählte Klasse zählen
6. *filterBodyAmount(record)* zum Ausfiltern von Frames deren Personenanzahl nicht im gewünschten Bereich liegt
7. *filterDistance(record)* zum Ausfiltern von Personen, deren Position außerhalb des ausgewählten Bereichs liegt

```

1 def filterRecord(record):
2     filteredRecord = record
3     funcs = [filterTime, filterFlyingPixelsInRecord, filterEngagement, filterAttributes, filterGroups
4             , filterBodyAmount,
5             filterDistance]
6     for func in funcs:
7         if len(filteredRecord) > 0:
8             filteredRecord = func(filteredRecord)
9         else:
10            return []
11 return filteredRecord

```

Listing 4.5: Anreihung der Filtermethoden

Auf die Implementierung der einzelnen Methoden wird im Folgenden der Reihe nach eingegangen.

filterTime(record)

Die Methode *filterTime(record)* hat die Aufgabe einen Record nach der Zeit zu filtern. Dabei gibt es lediglich zwei mögliche Rückgabewerte, den gesamten übergebenen Record oder einen leeren Array. Die Entscheidung, welcher der beiden Werte zurückgegeben wird, ist abhängig von dem Parameter *time* des ersten Frames.

Befindet sich dieser Parameter im zulässigen Bereich wird der übergebene Record zurückgegeben, andernfalls ein leerer Array.

```

1 def filterTime(record):
2     time = record[0]['time']
3     if filterConfig['from'] < filterConfig['to']:
4         if filterConfig['from'] <= time.time() <= filterConfig['to']:
5             return record
6     else: # crosses midnight
7         if time >= filterConfig['from'] or time.time() <= filterConfig['to']:
8             return record
9     return []

```

Listing 4.6: Filter-Time-Methode

filterFlyingPixels(record)

Ziel der Methode *filterFlyingPixels(record)* ist das Ausfiltern von Frames mit fehlerhaften Positionen. Fehlerhafte Positionen machen sich bemerkbar, indem sie mit ihren umliegenden Positionen (z. B. des vorherigen oder nachfolgenden Frames) verglichen werden. Übersteigt die Distanz der beiden Positionen den ausgewählten Filterwert, liegt ein Fehler vor und der Frame wird zusammen mit den dazugehörigen Frames entfernt. Da durch das Entfernen eines einzelnen Frames das Ergebnis verfälschen würde, wird der gesamte Recordteil, der einer Person zuordenbar ist entfernt. Die jeweilige Person wird in der Visualisierung folglich nicht berücksichtigt.

Dazu wird im ersten Schritt geprüft, ob der Flying-Pixel-Filter deaktiviert ist. In diesem Fall wären alle Werte zulässig und es würde der übergebene Record unmittelbar zurückgegeben werden. Andernfalls wird die Methode *divideRecordInBodies(record)* aufgerufen, die den Record in Recordteile zerlegt, die jeweils einer Person (= BodyIndex) zugeordnet werden können. Im darauffolgenden Schritt wird über jeden Frame jeder Person iteriert und anhand der Methode *checkDist(frame, lastFrame)* geprüft, ob die Distanz der Positionen innerhalb des zulässigen Bereichs liegt. Ist dies nicht der Fall, liegt ein Verstoß gegen die Filterbedingung vor. Folglich wird der Recordteil, der dem jeweiligen Passanten zugeordnet wurde, entfernt und taucht nicht in der Visualisierung auf.

Sind alle Frames der jeweiligen Person zulässig, werden sie an den Array *filteredRecord* angehängen und schließlich nach Abschluss der Methode als gefilterter Array zurückgegeben.

```

1 def filterFlyingPixelsInRecord(record):
2     if filterConfig['flyingPixDist'] == -1:
3         return record
4     else:
5         framesDividedInBodies = divideRecordInBodies(record)
6         filteredRecord = []

```

```

7  # take each body and check the positionDistances
8  for body in framesDividedInBodies:
9      filteredRecordOfBodyN = []
10     lastFrame = None
11     for frame in body:
12         if lastFrame is None:
13             # than it has to be the first frame, so go on
14             lastFrame = frame
15         elif checkDist(frame, lastFrame):
16             # if check dist returns true the distances are ok
17             filteredRecordOfBodyN.append(frame)
18             lastFrame = frame
19         else:
20             # if not: The distances are to large --> So the complete body has to be thrown out of the
                record
21             filteredRecordOfBodyN.clear()
22             break
23     # if the record of one body isn't empty, we can append it to the filtered Records because
        there where no
24     # flyingPixel
25     if len(filteredRecordOfBodyN) > 0:
26         filteredRecord.extend(filteredRecordOfBodyN)
27     return filteredRecord

```

Listing 4.7: Filter-Flying-Pixels-Methode

filterEngagement(record)

Die Methode *filterEngagement(record)* ist für die Filterung der Daten nach dem Aufmerksamkeitsfilter zuständig. Ihr Übergabeparameter ist ein Record mit ungefilterten bzw. teilweise gefilterten Frames. Die Methode liefert einen Record zurück, der ausschließlich Passanten beinhaltet, deren Aufmerksamkeitswert innerhalb eines ausgewählten Bereichs liegt. Der Aufmerksamkeitswert ist dabei ein Prozentwert, der sich durch den Anteil aufmerksamer Frames an dem Record errechnen lässt.

Um die eingegebenen Records nach Aufmerksamkeit zu filtern wird in den ersten zwei Schritten gleich vorgegangen wie in der Methode *filterFlyingPixels(record)* (siehe Unterunterabschnitt 4.4.2). Dabei wird geprüft, ob der Filter angewendet wird, oder ob der eingegebene Record direkt zurückgegeben werden kann. Wird der Engagement-Filter verwendet, wird der Record in Record-Teile aufgegliedert und anschließend über jeden Frame jedes Recordteils iteriert. Ist der Frame-Parameter „Engaged“ erfüllt, hat der Passant in diesem Frame Aufmerksamkeit gezeigt und die Dauer des Frames wird zu der lokalen Variablen *engagedFrames* addiert.

Da ein Frame lediglich einen Zeitstempel und keine Zeitdauer besitzt, wird als Dauer des Frames die Differenz zwischen dem derzeitigen und dem nächsten Frame verwendet. Die Dauer des letzten Frames errechnet sich schließlich durch den Durchschnitt aller. Nach Fertigstellung eines Recordteils wird geprüft, wie groß der Anteil des Wertes *engagedFrames* an der Gesamtaufenthaltsdauer des Passanten ist. Ist dieser Anteil innerhalb des gewählten Filterbereichs, wird der Record-Teil an den zukünftigen Rückgabe-Record angehängen.

```

1  def filterEngagement(record):
2      if filterConfig['engagementFrom'] == 0 and filterConfig['engagementTo'] == 1:
3          return record
4      returnRecord = []

```

```

5 framesDividedInBodies = divideRecordInBodies(record)
6 # check engagement body by body
7 for body in framesDividedInBodies:
8     if len(body) > 1:
9         engagedFrames = 0
10        # get duration of the passengers stay
11        timeInArea = (body[-1]['time'] - body[0]['time'])
12        for index, frame in enumerate(body):
13            if frame['Engaged'] == 'Yes':
14                # get the duration of the Frame
15                # if it is the last Frame of the record --> take the average value (usually around 0.032s)
16                # else take the real duration of the Frame
17                timedelta = (timeInArea / len(body)) if index == (len(body) - 1) \
18                    else (body[index + 1]['time'] - frame['time'])
19                engagedFrames += timedelta.total_seconds()
20        if len(body) > 1:
21            if filterConfig['engagementFrom'] < (engagedFrames / timeInArea.total_seconds()) <
22                filterConfig['engagementTo']:
23                returnRecord.extend(body)
24    return returnRecord

```

Listing 4.8: Filter-Engagement-Methode

filterAttributes(record)

Die Methode *filterAttribute(record)* soll den Record nach Erfüllung der drei Filterparameter *Trägt Brille*, *Ist glücklich* und *Bewegt Mund* filtern. Die Filterparameter können die Ausprägung *deaktiviert*, *nur* und *ohne* besitzen, wodurch bei Bedarf Frames mit jeweils erfüllten Parametern ausgeschlossen oder inkludiert werden. Der Übergabeparameter der Methode ist ein Record, der gleich wie bei den anderen Filtermethoden nicht oder nur teilweise gefiltert ist. Zurückgegeben wird von der Methode ein Record der den jeweiligen Filtereinstellungen entspricht.

Ist der Filter nicht in der Standardeinstellung, wird über den übergebenen Record iteriert und die relevanten Parameter des Frames überprüft. Entspricht dieser der derzeitigen Filtereinstellung, wird die lokale Variable *filterGlasses*, *filterHappy* oder *filterMouthMoved* jeweils auf True gesetzt. Sind die Werte aller lokalen Variablen True, wird der Frame an den künftigen Rückgabe-Record angehängen. Andernfalls wird er nicht in der Visualisierung berücksichtigt.

```

1 def filterAttributes(record):
2     if filterConfig['DisplayAttributes'] == [-1, -1, -1]:
3         return record
4     returnRecord = []
5     # check the frame
6     for frame in record:
7         filterGlasses = filterConfig['DisplayAttributes'][0]
8         filterHappy = filterConfig['DisplayAttributes'][1]
9         filterMouthMoved = filterConfig['DisplayAttributes'][2]
10        filterGlassesErfuehlt = (filterGlasses == -1 or (filterGlasses == 1 and frame['Glasses'] == '
11            Yes') or (
12            filterGlasses == 0 and frame['Glasses'] == 'No'))
13        filterHappyErfuehlt = (filterHappy == -1 or (filterHappy == 1 and frame['Happy'] == 'Yes') or (
14            filterHappy == 0 and frame['Happy'] == 'No'))
15        filterMouthMovedErfuehlt = (filterMouthMoved == -1 or (filterMouthMoved == 1 and frame['
16            MouthMoved'] == 'Yes') or (
17            filterMouthMoved == 1 and frame['MouthMoved'] == 'No'))
18        if filterHappyErfuehlt and filterMouthMovedErfuehlt and filterGlassesErfuehlt:

```

```

17     returnRecord.append(frame)
18     return returnRecord

```

Listing 4.9: Filter-Attributes-Methode

filterGroups(record)

Der Filter *filterGroups(record)* klassifiziert die aufgetretenen Personen abhängig davon, in welcher Position sich der Filter *DisplayGroups* befindet. Dieser kann drei Ausprägungen annehmen, wobei -1 für deaktiviert, 0 für interactive und 1 für nicht-interaktive User 1 steht. Abhängig von der Einstellung des Filters wird der Record ungefiltert zurückgegeben oder eine der Methoden *filterGroupsInteractive(record)* oder *filterGroupsNotInteractive(record)*

```

1 def filterGroups(record):
2     if filterConfig['DisplayGroups'] == -1:
3         return record
4     elif filterConfig['DisplayGroups'] == 0:
5         return filterGroupsInteractive(record)
6     else:
7         return filterGroupsNotInteractive(record)

```

Listing 4.10: Filter-Groups-Methode

Die beiden Methoden *filterGroupsInteractive(record)* und *filterGroupsNotInteractive(record)* sind im Prinzip äquivalent aufgebaut, weshalb im Folgenden lediglich auf erstere Methode eingegangen wird.

Die Methode *filterGroupsInteractive(record)* hat zur Aufgabe den übergebenen Record nach interaktiven Nutzern zu filtern. Hierzu wurden in Tabelle 2.3 die notwendige Entscheidungsmatrix ausgearbeitet, um einen Passanten als (nicht) interaktiv einordnen zu können.

Dazu wird im ersten Schritt, der Record nach Passanten untergliedert. Im Anschluss wird als erste Voraussetzung für jeden Passanten geprüft, ob dessen Aufenthaltszeit im Kamerabereich länger als zwei Sekunden angedauert hat. Ist dies der Fall, wird im Folgenden über jeweiligen Record des Passanten iteriert und die Anzahl der Engagement-Ereignisse geprüft. Wurde bei dem User mehr als zweimal „Engaged=Yes“ festgestellt, wird der Passant als *interaktiv* eingestuft und wird an den Rückgabe-Array angehängt. Nach Terminierung der Schleife wird der Rückgabe-Array mit allen interaktiven Passanten zurückgegeben.

```

1 def filterGroupsInteractive(record):
2     framesDividedInBodies = divideRecordInBodies(record)
3     filteredRecord = []
4     for body in framesDividedInBodies:
5         if len(body) > 0:
6             if checkTimeInArea(body, 2, True):
7                 engagementCounter = 0
8                 for frame in body:
9                     if engagementCounter == 2:
10                        filteredRecord.extend(body)
11                        break
12                        elif frame['Engaged'] == 'Yes':
13                            engagementCounter += 1
14     return filteredRecord

```

Listing 4.11: Filter-Groups – Interactive

filterDistance(record) und filterBodyAmount(record)

Die beiden Methoden *filterDistance(record)* und *filterBodyAmount(record)* sind prinzipiell gleich aufgebaut wie die Methode *filterAttributes(record)* (Vgl. Unterunterabschnitt 4.4.2). Sie bestehen aus einer Abfolge von if-else abfragen, anhand derer die Erfüllung der Filterbedingungen geprüft wird. Ist die Filterbedingung erfüllt, wird jeweils der Record bzw. der Recordteil zurückgegeben. Andernfalls wird ein leerer Record zurückgegeben. Auf diese Methoden wird aufgrund der Vergleichbarkeit mit vorherigen Methoden nicht weiter eingegangen.

4.4.3 Berechnung der Heatmap-Daten

Bei der Berechnung der Heatmap-Daten war ein Algorithmus notwendig, eine flexible und skalierbare Übertragung der Fließkomma-Tupel (x- und z-Position) einem Element eines zweidimensionalen Arrays zuordnet. Der Ablauf der Verarbeitung ist vergleichbar mit dem einer zweidimensionalen Quantisierung. Bei der Quantisierung wird einer physikalischen Größe ein begrenzter Skalenwert zugewiesen. Da dieses hier sowohl bei der x-Achse, als auch bei der z-Achse der Positionen angewendet wird, kann von einer zweidimensionalen Quantisierung gesprochen werden. Um die Begrenzung der Skalen zu erreichen, wurden anhand der übergebenen Auflösung *xAxisPixel* und *zAxisPixel* und der Randpositionen des Records (*maxX*, *maxZ*, *minX*, *minZ*) die Größe eines Skalen-Schritts berechnet. Als nächsten Schritt wird der Anteil negativer Werte bei der Visualisierung ermittelt, um eine Verschiebung der Werte in den positiven Bereich durchzuführen. Diese Verschiebung ist notwendig, da die Positionen sonst negativ sein können. Da die Indizes n des zu Beginn initialisierten Arrays jedoch den Wertebereich $n \in \mathbb{N}, n \geq 0$ besitzen, wäre die Position keinem Array-Element zuordenbar. Diese Zuordnung wird im nächsten Schritt gemacht. Dazu wird der Index der Array-Spalte als *idxX* und der Index der Array-Zeile als *idxZ* berechnet. Nach der Anwendung dieses Schritts auf beiden Achsen ergibt sich das Element, dem die Position zuordenbar ist. Der Wert dieses Elements wird folglich um eins erhöht. Schließlich werden zum Zweck der besseren Visualisierbarkeit alle Werte des Arrays, die mit 0 belegt sind, durch None ersetzt.

```

1 def getHeatMapArray(xAxisPixel, zAxisPixel):
2     # initialize the return-array. Size of the Array depends on the amount of pixels
3     hmArray = [[0 for i in range(args['yaxis'])] for j in range(args['yaxis'])]
4     # get all the max and min values of x and yAxis to get the size of each pixel
5     maxX = max(flatStoredRecords, key=lambda x: x['xAxis'])['xAxis']
6     maxZ = max(flatStoredRecords, key=lambda x: x['zAxis'])['zAxis']
7     minX = min(flatStoredRecords, key=lambda x: x['xAxis'])['xAxis']
8     minZ = min(flatStoredRecords, key=lambda x: x['zAxis'])['zAxis']
9     xAreaSize = (abs(maxX) + abs(minX)) / xAxisPixel
10    zAreaSize = (abs(maxZ) + abs(minZ)) / zAxisPixel
11    # look for the negative area of the visualization to avoid OutOfBounds-Exceptions
12    negAreaX = math.floor(xAxisPixel * abs(minX) / (abs(maxX) + abs(minX)))
13    negAreaZ = math.floor((zAxisPixel * abs(minZ) / (abs(maxZ) + abs(minZ))))
14    # iterate through all values, check in which array element the position should be assigned to.
15    # add 1 to those element.
16    for record in storedRecords:
17        for frame in record:
18            idxX = math.floor(frame['xAxis'] / xAreaSize) + negAreaX
19            idxZ = math.floor(frame['zAxis'] / zAreaSize) + negAreaZ
20            hmArray[idxX][idxZ] += 1
21    for row in range(len(hmArray)):
22        for col in range(len(hmArray[row])):

```

4 Implementierung

```
23     if hmArray[row][col] == 0:  
24         hmArray[row][col] = None  
25     return {'minX': minX, 'maxX': maxX, 'minY': minZ, 'maxY': maxZ, 'dataSet': hmArray}
```

Listing 4.12: Generierung Heatmap

5 Evaluation

Der Begriff Evaluation stammt ursprünglich aus dem Lateinischen (lat *valere* = stark, wert sein) und bedeutet per Definition eine „sach- und fachgerechte Bewertung.“ (Dudenredaktion, o.J.) Das folgende Kapitel verfolgt somit das Ziel den implementierten Prototypen sach- und fachgerecht zu bewerten. Die Evaluation eines Prototyps kann man durch verschiedene Methoden angehen. Im Zuge dieser Arbeit wird der Prototyp anhand eines Usability-Tests evaluiert. Die Begrifflichkeit *Usability* setzt sich aus den beiden englischen Wörtern *use* (= Gebrauch) und *utility* (= Nutzen) zusammen. Die Unterteilung der Begriffe lässt darauf schließen, dass die Usability die Güte der Nutzung beschreibt. Stapelkamp (2007) beschreibt die Usability eines Softwaretools als die Benutzerfreundlichkeit. Diese ginge jedoch über die Einfachheit des Gebrauchs hinweg. Weiter berücksichtige sie die allgemeine Effizienz, um eine bestimmte Aufgabe zu erfüllen. Dazu gehöre beispielsweise auch der Weg zum Erhalt des Produkts oder dessen Installation.

Eine weit verbreitete Methode um die Usability eines Tools zu ermitteln ist der Nutzertest. Als Test-Verfahren bietet sich hierbei das Thinking-Aloud-Verfahren (= Laut Denken) an. Ebenso wäre als Verfahren die Interaktionsaufzeichnung denkbar, diese setzt jedoch den Einsatz eines Tracking-Tools (Eye-Tracking, Maus-/Tastatur-Tracking, etc.) voraus. (Stapelkamp, 2007, S. 514) Im Zuge dieser Arbeit wird der Usability-Test im Thinking-Aloud-Verfahren durchgeführt. Zusätzlich wird dem Testteilnehmer ein Fragebogen zur Verfügung gestellt, den er nach der Durchführung des Tests ausfüllen kann.

5.1 Thinking-Aloud-Test

Ziel des Thinking-Aloud-Tests ist, die spontanen Gedanken des Probanden bei der Nutzung des Produkts zu erfahren. Anhand der Gedanken kann der Entwickler des Produkts Rückschlüsse darauf ziehen, was er an seiner Anwendung verbessern kann und wie der Nutzer die Anwendung wahrnimmt. Da es für den Menschen unüblich ist, seine Gedanken prompt zu äußern, ist es notwendig, den Probanden dahingehend zu instruieren, dass er seine Gedanken ausspricht. (Stapelkamp, 2007, S. 540)

5.1.1 Test-Bedingungen

Die Bedingungen, unter denen der Test durchgeführt wurde, waren mit wenigen Abweichungen so gewählt, wie sie unter Realbedingungen eintreten könnten. Der Proband Dr. Jan Schwarzer wurde gezielt ausgewählt, da er sich bereits mit dem Thema Datenauswertung von Body-Tracking-Kameras umfangreich befasst hat. Weiter ist er ein Domänenexperte im Bereich Mensch-Computer-Interaktion und Public Displays. Da das Tool vielmehr im professionellen Bereich eingesetzt werden kann, ist die Durchführung eines Test mit zufällig ausgewählten Probanden weniger zweckmäßig. Sinnvoller ist es die Usability des Tools durch Personen zu testen, die Erfahrung in der Analyse von Body-Tracking-Daten haben.

Der Test wurde online über die Kommunikationsplattform *Zoom* durchgeführt. Der Proband nahm an dem Test mit seinem eigenen Endgerät (Desktop-PC) teil. Die Videoübertragung erfolgte dabei über ein zweites Endgerät des Probanden (Laptop). Der Proband filmte sich während der gesamten Testdurchführung mit Ton und Bild, während die Experimentatoren sowohl die Kamera, als auch den Ton, mit wenigen Ausnahmen deaktiviert hatten. Das Datenanalyse-Tool wurde auf einem öffentlich zugänglichen Webhosting-Server gehostet, sodass es dem Probanden mit der üblichen Ladezeit zur Verfügung stand. Abweichend von den Realbedingungen erhielt der Proband ein Aufgabenblatt, sowie einen ausgewählten Datensatz, der die Kontrollierbarkeit des Tests ermöglichte. Das entwickelte Aufgabenblatt zielte darauf ab den Teilnehmer über alle Kernfunktionen des Programms zu begleiten, sodass dieser einen Überblick über die möglichen Funktionen und Eigenschaften erhält und mit allen Use Cases konfrontiert wird. Das entwickelte Aufgabenblatt bearbeitet die Auswertung dreier Datensätze. In jeder der Auswertung werden dabei verschiedene Filtereinstellungen verwendet, die auf die Unterteilung der Passanten in Klassen und nach Eigenschaften abzielt. Ebenso abweichend von den Realbedingungen wurde der Proband vorher instruiert, dass er seine Gedanken ausspricht, um seine spontanen Eindrücke protokollieren zu können. Es wurde vor der Durchführung des Tests bewusst von einer Einführung in das Tool abgesehen, da mitunter bewertet werden sollte, wie intuitiv sich die Bedienung des Tools gestaltet. Stieß der Proband jedoch auf ein Hindernis bei der Aufgabenstellung oder bewegte sich die Durchführung des Tests in die falsche Richtung, griffen die Experimentatoren in den Test ein, indem sie dem Probanden Tipps gaben.

5.1.2 Test-Ergebnis

Anhand des Transkripts, das während des Tests erstellt wurde, kann das Ergebnis des Thinking-Aloud-Tests ermittelt werden. Das Ergebnis war in manchen Fällen eindeutig durch den Probanden ausgesprochen, häufig kam es bei verbesserungswürdigen Funktionen jedoch vor, dass der Proband nicht wusste wie er vorgehen soll. Ergebnisse, die von dem Probanden eindeutig ausgesprochen worden, waren unter anderem:

- Erkennen des Datei-Upload-Buttons (Transkript 1:03 min)
- Verständlichkeit der Legende unterhalb der Heatmap (Transkript 7:50 min)
- Verständlichkeit der Achsen-Einheiten der Heatmap (Transkript 7:36 min)
- Erkennen der Reset-Funktion der Filter (Transkript 12:27 min)
- Fehlen einer Funktion zum Exportieren der Visualisierungen (Transkript 27:46 min)
- Fehlen einer Zoom-Funktion zum Vergrößern der Heatmap (Transkript 35:33 min)

Zusätzlich trat häufig die Situation auf, dass der Proband nicht wusste, wie er eine Aufgabe lösen konnte, da ihm die eine Funktion des Tools nicht ersichtlich war. Diese Situationen traten unter anderem in folgenden Zusammenhängen auf:

- Verständlichkeit, wann ein Filter auf Standardeinstellung ist, explizit bei den Time-Pickern (Transkript 9:21 min)
- Erkennen des Drag-and-Drop-Fields in der Upload-Funktion (Transkript 10:03 min)

- Erkennen der Funktionalitäten der Farbpaletten-Einstellung bei der Heatmap (Transkript 15:47 min)
- Erkennen der Speicherfunktion der Visualisierungen (Transkript 15:47)
- Erkennen des Side-Menüs zum Wechseln auf die Vergleichsseite (Transkript 20:13 min)
- Verständlichkeit der Unterschiede zwischen dem temporären Deaktivieren von Farbeinträgen und dem Ändern der Farbpaletteneinträge in der Heatmap (Transkript 23:45 min)

Analysiert man die soeben aufgezählten Punkte in Hinblick auf den Ursprung des Defizits, stellt man fest, dass ein Großteil der Mängel aus der Bedienung des User-Interfaces stammen. Sehr oft trat es auf, dass Funktionen oder Buttons durch den Nutzer nicht erkannt werden konnten, da das dazugehörige Eingabeelement (z. B. Button) vom Benutzer nicht gefunden wurden.

Als Lösungsansatz könnte dabei eine Überarbeitung des Designs der GUI in Betracht kommen. Bei der Überarbeitung sollte konkret die farbliche Abhebung der Buttons sowie deren Positionierung überdacht werden. Derzeit sind die Buttons oftmals nur als Icons oder in einer dezenten Farbe dargestellt, die sich nicht vom Rest der Applikation abhebt. Die Verwendung einer herausragenden Farbe, z. B. rot könnte dabei Abhilfe schaffen. Zusätzlich wäre es in Bezug auf die Positionierung anzudenken eine Toolbar zu entwerfen, die alle möglichen Buttons enthält. Diese würde als großes Element in der GUI hervorstechen, sodass der User nicht nach ihr suchen muss.

Ein weiterer großer Teil der angesprochenen Mängel bezieht sich auf die Verständlichkeit einzelner Elemente der Heatmap. Dabei gäbe es zwei Lösungsansätze. Einerseits wäre es möglich, die Verständlichkeit der Elemente durch Unterrichtung der Nutzer zu verbessern. Hierbei käme die Erstellung eines Handbuchs infrage, das der Nutzer jederzeit innerhalb der Applikation abrufen kann. Andererseits wäre es möglich, die unverständlichen Elemente verständlicher zu gestalten, indem sie modifiziert werden. Da es sich bei der Heatmap jedoch um eine externe Bibliothek handelt, sind die Elemente nur bedingt modifizierbar. Ein Lösungsansatz wäre dabei die Bibliothek zu wechseln, oder die Heatmap von Grund auf zu implementieren, um alle Elemente modifizieren zu können.

Ein Mangel, der vergleichsweise selten während des Thinking-Aloud-Tests auftrat, ist, dass benötigte Funktionalitäten tatsächlich fehlten. Dieses Fehlen der Funktionalitäten bezog sich jedoch in den meisten Fällen erneut auf eine Modalität der Heatmap-Bibliothek, was den Lösungsansatz bekräftigt, diese Bibliothek auszutauschen.

5.2 Fragebogen

Um neben den spontanen Gedanken des Teilnehmers noch die, im Nachgang entstandene, überdachte Meinung zu der Anwendung zu erfahren, kann der Testteilnehmer nach der Durchführung des Usability-Tests einen Fragebogen ausfüllen. Die Fragen des Fragebogens sind in fünf Kategorien unterteilbar:

- Allgemeine Meinung: Dieser Teil zielt auf die allgemeine Bewertung des Prototyps ab. Es soll den Gesamteindruck der Anwendung in Hinblick auf die Optik, die Bedienung, Erfüllung der Nutzeransprüche und die Navigation darstellen. Ein negatives Ergebnis dieser Fragen sollte ein grundlegendes Überdenken der Anwendung zur Folge haben. Ein

positives Ergebnis kann dahingegen den Erfolg der Konzeption und Implementierung bestätigen.

- **Verständlichkeit:** Die Fragen zu der Verständlichkeit beabsichtigen herauszufinden, ob der Teilnehmer die Inhalte der Applikation versteht. Hierbei geht es vielmehr um die Formulierung der Filter und Darstellung der Visualisierungen. Entstände in diesem Fragebogen ein negatives Ergebnis, wäre es notwendig, die Funktionalitäten in den jeweiligen Teilen besser zu erklären, oder ein umfassendes Handbuch zu erstellen, das den Benutzer bei der Verwendung des Tools unterstützt.
- **Ladezeiten:** Dieser Teil des Fragebogens soll auf die Performance der Anwendung eingehen. Da bei einer Webanwendung zwischen der tatsächlichen Leistung der Anwendung und der Übertragungszeit zwischen Client und Server differenziert werden muss, wurden die Fragen in die einzelnen Schritte des Prozesses untergliedert (Upload, Filterung, Visualisierung). Ein negatives Ergebnis in allen Teilen, wäre sowohl auf die Leistungsfähigkeit des Backends, als auch auf die Übertragungsgeschwindigkeit zwischen von Client auf Server zurückzuführen. Dahingegen spräche ein gutes Ergebnis in den Fragen zu Filterung und Visualisierung zusammen mit einem schlechten Ergebnis beim Upload dafür, dass das Defizit in der Übertragungsgeschwindigkeit läge.
- **Umfang der Inhalte:** Der Umfang der Inhalte soll die Frage beantworten, ob das Tool die Datenauswertung von Bodytracking-Kameras nur oberflächlich erfüllt, oder ob es auch die Ansprüche erfahrener Nutzer erfüllt. Ein schlechtes Ergebnis hätte zur Folge, dass für die Anwendung im professionellen Bereich Funktionen nachgerüstet werden müssten.
- **Potenzial:** Die Fragen zu dem Potenzial des Tools beabsichtigen Auskunft über die Zukunftsträchtigkeit zu beantworten. Es soll herausgefunden werden, ob es sinnvoll ist, die Anwendung basierend auf dem bestehenden Fundament weiterzuentwickeln, oder ob grundlegende Elemente, wie die Benutzerschnittstelle *Webbrowser* für weiterführende Arbeiten an dem System geändert werden sollten.

Das Ergebnis des Fragebogens deckte sich in großen Teilen mit dem Ergebnis des Thinking-Aloud-Tests. Die Fragen, die auf die allgemeine Meinung zum Tool eingehen, wurden mit mittelmäßigen Werten bewertet. So empfand der Proband die Navigation, sowie die Anwendung im Allgemeinen als gut (entspricht 3/5 Punkten). In Hinblick auf die optische Aufmachung bewertete er die Anwendung mit 4/5 Punkten.

Im Gegensatz zu dem Thinking-Aloud-Test erreichte die Bewertung der Fragen zur Verständlichkeit der Anwendung gute Ergebnisse. Der Proband bewertete die Verständlichkeit der Visualisierungen und Filter mit jeweils mit 4/5 Punkten. Dahingegen war ein Resultat des Thinking-Aloud-Tests, dass die Verständlichkeit der Anwendung verbessert werden müsste. Diese unterschiedlichen Ergebnisse könnten sich darauf zurückführen lassen, dass der Proband während der Nutzung das Tool besser kennengelernt hat und im Nachgang mit der jeweiligen Eigenart zurechtkommt. Dies deutet weiter darauf hin, dass die Einführung eines Handbuchs für die Anwendung sinnvoll wäre, damit präventiv eine anfängliche Frustration künftiger Nutzer vermieden werden kann.

Im Bereich der Ladezeiten wurden gute bis sehr gute Ergebnisse erzielt. Vor allem die Fragen, die die Leistungsfähigkeit des Programms abzielen ist das Ergebnis positiv. Der Proband gab an, dass die Ladezeiten beim Aktualisieren der Filter, wie auch beim Aktualisieren der Visualisierung sehr gut (4/4 Punkte) sei. Lediglich die Ladezeit beim Hochladen eines neuen

Datensatzes sei nur zufriedenstellend (3/4 Punkte). Die vergleichsweise hohe Ladezeit lässt sich auf die Übertragung der Daten von Client auf Server zurückführen. Ein Lösungsansatz zur Verbesserung dieses Punktes könnte der Wechsel des Webhosting-Anbieters sein. Alternativ wäre es denkbar, den Import-Prozess neu zu organisieren. Eine mögliche Organisationsstruktur könnte dabei sein, eine Datenbank zu aktivieren und den umfassenden Datensatz auf dort abzulegen. Der User könnte bei Bedarf den gewünschten Tag des Datensatzes auswählen, der dann unmittelbar von der Datenbank in das Backend geladen wird. Die Übertragung von Client auf Server entfiere in diesem Fall.

Der Umfang der Inhalte wurde insgesamt mittelmäßig bewertet. In der darauffolgenden Freitext-Eingabe merkte der Proband weitere Visualisierungs- und Filterungsmöglichkeiten an, die für die Weiterentwicklung des Analyse-Tools förderlich wären.

Dieses Ergebnis deckt sich auch mit den Fragen zum Potenzial des Tools. Daraus geht hervor, dass eine Weiterentwicklung der Anwendung auf diesem Fundament sinnvoll wäre.

Alles in Einem ist das Ergebnis des Tests für einen Prototyp akzeptabel. Es geht vor allem aus dem Thinking-Aloud-Test und dem Fragebogen hervor, dass die Anwendung basierend auf der existierenden Lösung weiterentwickelt werden sollte. Zudem wäre die Nachrüstung zusätzlicher Funktionen wie Filter, Visualisierungen, Speicherfunktionen und Interaktionen mit den Visualisierungen nützlich. Ein wichtiges Artefakt, was aus der Weiterentwicklung hervorgehen sollte, ist ein Benutzerhandbuch, das den Nutzer durch die ersten Schritte der Anwendung führt. Ebenso wäre eine Nachbesserung des Userinterfaces sinnvoll. Dabei könnte die Positionierung sowie die farbliche Abhebung einzelner Interaktionselemente (z. B. Buttons) verbessert werden.

6 Fazit und Ausblick

Ziel der Bachelorarbeit war es, zu erarbeiten, wie ein nutzen-stiftendes Datenanalyse-Tool für Body-Tracking-Kameras aufgebaut sein kann. Um die Frage zu erarbeiten wurde für das Tool anfänglich eine Analyse durchgeführt, in der die Anforderungen vorerst erarbeitet und im Anschluss durch Use-Cases homogenisiert wurden. Bei der Erarbeitung der Anforderungen erschloss sich, dass der Schwerpunkt des Tools auf dem Visualisierungs- und Filterungsprozess liegt. Im Anschluss dessen wurde das Tool konzipiert. Die Konzeption beinhaltete einerseits den Entwurf eines User-Interfaces, andererseits wurden die Programmabläufe und die Teilsysteme des Tools erarbeitet. Das Ergebnis der Konzeption war ein grober Entwurf des Tools, anhand dessen im darauffolgenden Kapitel *Implementierung* die Realisierung der Software durchgeführt werden konnte. In der Implementierung wurde überwiegend auf die entstandenen Herausforderungen, sowie den jeweiligen Lösungsansatz eingegangen. Nach der Durchführung der Implementierung wurde der entwickelte Prototyp abschließend durch einen Domänenexperten evaluiert.

Wie die Evaluation des Prototyps zeigte, ist das Ergebnis der Arbeit akzeptabel. Zwar besteht an einigen Stellen Nachbesserungsbedarf, was jedoch in der Entwicklung eines Systems kein ungewöhnliches Ergebnis ist. Aus zeitlichen Gründen kann die Nachbesserung und Weiterentwicklung des Datenanalyse-Tools für Body-Tracking-Kameras nicht im Rahmen dieser Bachelorarbeit durchgeführt werden. Dennoch wären folgende Weiterentwicklungen für die Optimierung von Nutzen und Usability sinnvoll:

Das Integrieren einer MySQL-Datenbank, die erstmalig hochgeladene Daten in Tabellen ablegt und somit die Uploadzeiten minimiert. Weiter könnte das Tool um zusätzliche Visualisierungs- und Filterarten erweitert werden. Diese könnten dem Nutzer zusätzliche Informationen über das Passantenverhalten liefern. Als letzte Weiterführung ist die Überarbeitung des User-Interfaces sowie die Entwicklung eines Handbuchs zu nennen. Wie bereits in der Evaluation erarbeitet, wird die Bedienung des Tools zu Teilen nicht als intuitiv wahrgenommen. Dieser Mangel könnte durch die Weiterführung des Projektes behoben werden und somit das Tool vom Prototyp zu einem sinnvollen, benutzerfreundlichen und nutzenstiftenden Software-Tool verwandeln.

Dennoch wurde die anfängliche Fragestellung, wie ein Datenanalyse-Werkzeug für Body-Tracking-Kameras aufgebaut sein kann, umfassend bearbeitet und beantwortet. Die Evaluation zeigte letztendlich, dass die Weiterentwicklung auf Basis des derzeitigen Standes auf jeden Fall weitergeführt werden soll, was für die Erfüllung des Forschungsziels spricht.

Abkürzungsverzeichnis

UML	Unified Modeling Language
BPMN	Business Process Modeling and Notation
MVVM	Model-View-ViewModel
CSS	Cascading-Style-Sheet
HTML	Hyper-Text-Markup-Language
GUI	Graphical User Interface
IDE	Integrated Development Environment
CORS	Cross-Origin Resource Sharing
REST	Representational State Transfer
API	Application Programming Interface
ToF	Time-of-Flight

Abbildungsverzeichnis

1.1	Skelett-Punkte Microsoft Kinect v2 (Windows-Developer-Center, 2014, S. 8)	4
1.2	Vergleich Kinect v1 und v2 (Wasenmüller und Stricker, 2016, S. 8)	5
1.3	Framesatzeintrag / Frame	6
2.1	Use-Case-Diagramm	13
3.1	UI – Startseite	17
3.2	UI – Daten-Import	17
3.3	UI – Filter	18
3.4	UI – Navigation	19
3.5	UI – Visualisierungsvergleich	19
3.6	Programmabläufe – BPMN-Diagramm	22
4.1	Programmaufbau – UML-Diagramm	28
4.2	CORS-Prinzip (MDN-Mozilla, 2021)	30

Listings

4.1	Mengenermittlung Records / Frames	27
4.2	Frontend – Datenupload	31
4.3	Backend – Datenupload	31
4.4	Parsing-Methode für Erstellung von Records	32
4.5	Anreihung der Filtermethoden	33
4.6	Filter-Time-Methode	34
4.7	Filter-Flying-Pixels-Methode	34
4.8	Filter-Engagement-Methode	35
4.9	Filter-Attributes-Methode	36
4.10	Filter-Groups-Methode	37
4.11	Filter-Groups – Interactive	37
4.12	Generierung Heatmap	38

Literaturverzeichnis

- Ashcraft, B. (2014). Microsoft's Kinect Is Now Guarding the Korean Border. <https://kotaku.com/microsofts-kinect-is-now-guarding-the-korean-border-1514792443>. last-checked: 20.12.2021.
- Baun, C., Bengel, G., Kunze, M., und Stucky, K.-U. (2015). *Client-Server-Modell und Web-Services*. Springer Fachmedien Wiesbaden.
- Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., und Ertl, T. (2014). State-of-the-Art of Visualization for Eye Tracking Data. In Borgo, R., Maciejewski, R., und Viola, I., Herausgeber, *EuroVis - STARs*. The Eurographics Association.
- Breuer, P. (2005). Entwicklung einer prototypischen Gestenerkennung in Echtzeit unter Verwendung einer IR- Tiefenkamera. Master's thesis, Universität Koblenz.
- Broy, M. und Kuhrmann, M. (2021). *Einführung in die Softwaretechnik*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Böhringer, J., Bühler, P., Schlaich, P., und Sinner, D. (2014). *Webdesign*, Seiten 1–81. Springer Berlin Heidelberg.
- Dudenredaktion (o.J.). Evaluation. <https://www.duden.de/node/43225/revision/484099>. last-checked: 11.12.2021.
- FlaskRESTful (2020). FlaskRESTful Documentation. <https://flask-restful.readthedocs.io/en/latest/>. last-checked: 17.11.2021.
- Gieselmann, H. (2010). Open-Source-Treiber für Kinect veröffentlicht. <https://www.heise.de/newsticker/meldung/Open-Source-Treiber-fuer-Kinect-veroeffentlicht-1135025.html>. last-checked: 20.12.2021.
- Gronau, N. (2016). Modellierung von Geschäftsprozessen mit EPK und BPMN. Technischer bericht, Lehrstuhl für Wirtschaftsinformatik und Electronic Government Universität Potsdam.
- Hampel, T. (2012). Konzeption und Implementation eines Analysetools zur Auswertung von Logfiles einer komplexen, dynamischen Versuchsumgebung. Master's thesis, Humboldt Universität zu Berlin.
- Highcharts (2021). Highcharts Homepage. <https://www.highcharts.com>.
- JetBrains (2021). IntelliJ IDEA. <https://www.jetbrains.com/de-de/idea/>. last-checked: 24.11.2021.
- Kosslyn, S. M. (1985). Graphics and Human Information Processing: A Review of Five Books. *Journal of the American Statistical Association*, 80(391):499–512.

- Lipp, L. (2004). *Interaktion zwischen Mensch und Computer im ubiquitous computing: alternative Ein- und Ausgabemöglichkeiten für allgegenwärtige Informationstechnologien*. Publizistik (Lit). Lit.
- MDN-Mozilla (2021). Cross-Origin Resource Sharing (CORS). <https://developer.mozilla.org/de/docs/Web/HTTP/CORS>. last-checked: 03.12.2021.
- Michelis, D. und Müller, J. (2011). The Audience Funnel: Observations of Gesture Based Interaction With Multiple Large Displays in a City Center. *International Journal of Human-Computer Interaction*, 27(6):562–579.
- Mäkelä, V., Heimonen, T., und Turunen, M. (2017). Semi-Automated, Large-Scale Evaluation of Public Displays. *International Journal of Human-Computer Interaction*, 34(6):491–505.
- Python Packaging Authority (2021). An Overview of Packaging for Python. <https://packaging.python.org/overview/>. last-checked: 17.11.2021.
- Rumpe, B. (2004). *Modellierung mit UML*. Springer Berlin Heidelberg.
- Schneider, F. (2011). Mobile Interaktion mit Public Displays im öffentlichen Verkehr. Master's thesis, Technische Universität Dresden.
- Schumann, H. und Müller, W. (2000). *Visualisierung*. Springer Berlin Heidelberg.
- Schwarzer, J., Draheim, S., und von Luck, K. (im Druck). Behavior of Professional Agile Practitioners Around Semi-Public Ambient Displays. Unveröffentlichtes Manuskript.
- Stapelkamp, T. (2007). *Screen- und Interfacedesign*. Springer Berlin Heidelberg.
- Stefan Luber, S. A. (2017). Definition „Integrierte Entwicklungsumgebung“. <https://www.dev-insider.de/was-ist-eine-ide-a-600703/>. last-checked: 24.11.2021.
- Tableau-Software (2021). The Best Visualization for You. <https://www.tableau.com/learn/get-started/desktop-viz-design>. last-checked: 05.11.2021.
- Viereck, S. (2012). Microsofts technische Entwicklungsgeschichte der Kinect. <https://sebastianviereck.de/microsofts-technische-entwicklungsgeschichte-der-kinect/>. last-checked: 20.12.2021.
- Wasenmüller, O. und Stricker, D. (2016). Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision. In *Asian Conference on Computer Vision Workshop*.
- Windows-Developer-Center (2014). *Human Interface Guidelines v2.0*. Microsoft, Redmond, USA.

Anhang A. Evaluation Aufgabenblatt

Evaluation TraDAsys

Joel Temiz

11. Dezember 2021

Allgemeine Hinweise

- Im Folgenden wird die Bearbeitung eines Usability-Test durchgeführt.
- Die Teilnahme an dem Test erfolgt freiwillig und unentgeltlich.
- Der Teilnehmer kann den Test ohne Entstehung von Nachteilen und ohne Angabe von Gründen jederzeit abbrechen.
- Die Bearbeitung des Test wird im Think-Aloud-Verfahren durchgeführt. Dieses sieht vor, dass der Teilnehmer des Tests all seine Gedanken in Bezug auf die Usability des Tools *TraDAsys* kundtut um sie protokollierbar zu machen.
- Der Test wird unter Zustimmung des Teilnehmers aufgezeichnet und schriftlich mitprotokolliert. Aufgezeichnet wird dabei der Bildschirm sowie die Tonspur aller Anwesenden.
- Die Video- und Audio-Aufnahmen werden bis zum 05.01.2021 vernichtet.
- Bedingt durch die Konzeption der Evaluation ist eine anonymisierte Auswertung der Daten nicht möglich. Der Proband erklärt sich einverstanden, dass die erhobenen Daten im Rahmen der späteren Veröffentlichung namentlich mit ihm in Verbindung gebracht werden können. Der Proband kann der Auswertung einzelner Datenpunkte und Aussagen widersprechen, auch noch im Nachhinein. Dazu kann er sich bis spätestens 7 Tage nach der Evaluation an den Experimentator wenden um einzelne erhobene Aussagen aus der Auswertung zu streichen.
- Zur verbesserten Steuerung des Testverfahrens werden dem Teilnehmer Aufgaben gestellt, die er der Reihe nach abarbeiten kann.
- Von einer einleitenden Einweisung wird bewusst abgesehen, da sich die Befähigung zur Nutzung des Tools aus der Güte der Usability des Tools ergeben sollen.
- Zur Ermittlung des Gesamteindrucks ist die Bearbeitung eines nachträglichen Fragebogens erwünscht.

1 Analyse des ersten Datensatzes

- Laden Sie den Datensatz des 10.04.2017 in das System
- Verschaffen Sie sich anhand des BarCharts eine grobe Übersicht über das Personenaufkommen des Tages
- Grenzen Sie den Datensatz anhand der Filter auf die Kernzeiten des Auftretens ein
- Verschaffen Sie sich einen Überblick über die Visualisierung und treffen Sie allgemeingültige Aussage zu dem Angezeigten (z.B. Angezeigt werden nun alle Personen..., ein Großteil der Personen hielt sich... auf, ...)

2 Analyse des zweiten Datensatzes

- Laden Sie den Datensatz des 11.04.2017 und 12.04.2017 in das System und entfernen Sie den zuvor gewählten Zeitfilter.
- Wählen Sie anhand des BarCharts eine geeignete Halbierungsmarke bei der in beiden Teilen des Datensatzes gleich viele Personen erfasst werden.
- Führen Sie die Halbierung des Datensatzes zuerst für die erste Hälfte in den Filtern durch
- Passen Sie in der Farbpalette der Heatmap die Skalierung der Darstellung nach eigenem Ermessen an und sichern Sie die beiden Heatmaps unter geeigneter Bezeichnung. Um überflüssige Farbpaletten-Einträge zu erkennen, nutzen Sie die Legende der Heatmap.
- Führen Sie den letzten Schritt für die verbleibende Hälfte durch
- Vergleichen Sie die beiden gespeicherten Visualisierungen und treffen Sie eine Aussage zu dem Vergleich.

3 Analyse des dritten Datensatzes

- Laden Sie den Datensatz vom 02. - 05.05.2017 in das System ohne Filterung in das System
- Verbessern Sie die Auflösung der Heatmap indem Sie die Pixel pro Achse um 100 erhöhen.
- Finden Sie den Punkt in der Heatmap, an dem am meisten Frames aufgezeichnet wurden und sichern Sie die Visualisierung unter geeignetem Namen. Filtern Sie anschließend Records aus, derer Koordinatensprünge größer als 50cm sind und

überprüfen Sie die vorher ermittelten Koordinaten im Vergleich. (Sollte der Browser mit der Auflösung nicht zurecht kommen, kann der wert auf 200 verringert werden)

- Filtern Sie die Personen nun nach Happy und nicht Happy. Speichern Sie jeweils den BarChart und vergleichen Sie. Was fällt Ihnen auf?

4 Umfrage

Wie anfänglich angemerkt kann die Evaluation des Tools zusätzlich durch die Beantwortung von 16 Fragen gestützt werden. Die Bearbeitungszeit des Fragebogens beläuft sich auf circa 5 - 10 Minuten. Der Fragebogen ist erreichbar unter folgendem Link:

<https://www.survio.com/survey/d/C4A5C9Z1J5F6U7X3U>

Oder per QR-Code:



Anhang B. Evaluation Transkript

Joel Temiz (00:16)

Ja, es wäre schön, wenn du sagst, an welcher Aufgabe du bist. Das kannst du auch beiläufig machen. Ich bin jetzt gerade bei Aufgabe 1a. Aufgabe ist Punkt. Punkt, Punkt. Und dann ist es ja wie gesagt, dieser Thinking-Aloud-Test. Das heißt alles was du denkst zu der Usability des Tools, dass du das kundtust, also aussprichst. Ja, das finde ich jetzt blöd organisiert, oder? Das finde ich super. Die Farben sind toll, ... Weißt du worauf ich hinauswill?

Dr. Jan Schwarzer (00:49)

Okay, ich versuch's noch einmal und dann lege ich mal los. Ich will mal kurz das... Ist es wichtig, dass du mich von vorne siehst, weil ich jetzt hier sozusagen mein Set-Up habe?

Joel Temiz (01:00)

Nein, Das ist okay.

Dr. Jan Schwarzer (01:03)

Okay, dann versuche ich es mal. Also dann bin ich jetzt bei der Aufgabe 1, Analyse des ersten Datensatzes und da ist dann der erste Punkt. (Liest vor) Laden Sie den Datensatz des 10.01.2017. Ich mal vorab dann erst mal die Ordner runtergeladen. Das erste was mir aufgefallen ist, Joel: Ich habe erst mal suchen müssen, wo ich den Datenupload mache. Das war oben-links, der Datenimport. So, dann gehe ich da kurz drauf. So nächstes Problem, was ich jetzt habe ist, ... dass ich grad nicht weiß, was ich da auswählen soll ich habe jetzt einfach mal alle Textdateien ausgewählt.

Dr. Jan Schwarzer (01:53)

Jetzt habe ich hier die Möglichkeit wahrscheinlich hochzuladen mit dem linken Button und abzurechnen mit dem rechten Button und das wird mir auch jetzt nicht so richtig klar. Jetzt habe ich mal die linke Taste gedrückt, also den Upload. Jetzt sehe ich einen Ladebalken... So heißt ich habe 667 Records reingeladen. So, der zweite Punkt: (Liest vor) verschaffen Sie sich anhand des Barcharts eine grobe Übersicht über das Personenaufkommen des Tages. So also ich sehe im Barchart jetzt noch nichts. Auch nicht in der Heatmap. Ich weiß nicht, ob das so sein soll. Ich kann momentan bei mir lokal nicht mit dem Barchart-Diagramm Sachen angucken wie in Punkt 2 steht.

Joel Temiz (02:51)

Jawoll! Bitte einmal unter dem Barchart ist ein blauer Button mit einem Reload-Zeichen.

Dr. Jan Schwarzer (02:59)

Ja, da hätte ich jetzt auch drauf kommen können. Da wäre vielleicht die Idee, dass man das dann automatisch macht. Ok, genau. Also ich sehe hier auf jeden Fall das, was ich auch schon in den Interaktionsdaten gesehen habe. Das eben um 12 Uhr 13 Uhr, also um die Mittagszeit herum auch um 11 Uhr, da die meisten Passanten erkannt wurden, das ist auf jeden Fall ein plausibles Ergebnis. Ebenso um 9 Uhr wie auch um 18 Uhr, das habe ich beobachtet im Feld selber. Auch das sind erst einmal plausible Ergebnisse und stützen im Prinzip die Beobachtungen, die ich auch während meiner Doktorarbeit gemacht habe. Das sind so die Dinge, die ich jetzt hier erkennen kann in den Daten. So, jetzt ist der dritte Punkt: (Liest vor) Schränken Sie den Datensatz anhand der Filter auf die Kernzeiten des Auftretens ein. Okay, anhand der

Filter, dann geh ich jetzt auf den Reiter Filter... Time- Picker von, Time-Picker bis... Genau da regle ich dann jetzt mal die Kerzeiten des Auftretens.

Natürlich ist da jetzt die Frage was sind Kernzeiten? Wenn da jetzt die Arbeitszeiten gemeint sind, dann sag ich jetzt einfach mal zum Beispiel von 7 Uhr bis 17 Uhr. Dann soll ich wahrscheinlich unten-links das speichern. So, jetzt macht er hier was. Okay, es heißt jetzt nur noch knapp 81 Prozent der Records jetzt noch berücksichtigt. Jetzt geht zurück ins Barchart-Diagramm. Aktualisiere noch mal, und sehe dann die Aktualisierung. Ich sehe jetzt bei 17 Uhr gar keine mehr. Okay, weil ich wahrscheinlich bis 17 Uhr das eingegrenzt habe. Okay, dann der letzte Punkt bei der Aufgabe 1: (Liest vor) Verschaffen sich einen Überblick über die Visualisierung und treffen Sie allgemeingültige Aussagen dazu. Zum Beispiel angezeigt werden alle Personen..., ein Großteil der Personen, die sich auf... Über die Visualisierung.

Ok, das ist ja im Prinzip das, was ich schon gesagt habe, oder? Also zu den Uhrzeiten, so verstehe ich das jetzt.

Joel Temiz (05:24)

Entschuldigung, kannst du vielleicht noch mal wiederholen.

Dr. Jan Schwarzer (05:27)

Ich verstehe jetzt den vierten Punkt bei der ersten Aufgabe so im Prinzip das, was ich auch schon gesagt habe. Ich soll mir einen Überblick verschaffen und treffen Sie allgemeingültige Aussagen zu dem angezeigten. Und diese Patterns hatte ich ja kurz beschrieben, dass ich das deckt mit meinen früheren Beobachtungen. Das ist ja damit gemeint, oder?

Joel Temiz (05:46)

Eine Sekunde bitte ich schaue mir kurz die Aufgabe selber.

Dr. Jan Schwarzer (05:55)

Also bei mir steht ein Großteil der Person hielt sich Punkt Punkt Punkt auf meinst du...

Joel Temiz (06:01)

Das bezieht sich jetzt auf die Heatmaps. Das hätte besser beschrieben sein sollen, also nicht anhand des Barcharts analysieren. Du hattest es ja im vorherigen anhand des Barcharts analysiert und jetzt sollst du das anhand der Heatmap.

Dr. Jan Schwarzer (06:12)

Dann verstehe ich was du meinst. Ok, also anhand der Heatmap sehe ich auf jeden Fall das, was ich jetzt auch gerade aktuell selber untersucht habe. Man hat eine sehr starke Betonung auf den Laufweg, man sieht halt sozusagen... Nochmal vielleicht einen Hinweis, Joel, ich weiß nicht, welche Kamera das von den beiden ist. Das steht nicht dabei. Nur der Datensatz. Ich vermute, dass das die Kamera 2 ist, weil die Dichte an der Wand ist. Dadurch kommt diese Kurve zustande.

Joel Temiz (06:38)

Jawoll, das ist Kamera 2

Dr. Jan Schwarzer (06:40)

Genau. Ich sehe wie gesagt diese Betonung auf die Laufwege. Ich sehe insbesondere Betonung dieses, ... Im Prinzip der rosafarbenen, roten Bereiche. Das ist eben der Bereich, wo Leute sich gerade Getränke geholt hatten oder auch gerade von der Toilette gekommen sind. Also dass der sehr, sehr hoch frequentierte Bereich gewesen vor den Systemen. Ich sehe links bei mir, da sind ja sogar auch blaue, lila farbene Spots. Das ist halt im Prinzip der Bereich gewesen, wo die Leute zum Teil Meetings gehalten haben. Da sind dann Kartons zum Beispiel, also so Pinnwände, wo die Mitarbeitenden etwas anbringen könnten. Das ist so in der Essenz das, was ich jetzt sehen kann. Ich sehe auch vereinzelt ein paar Pfade, die anscheinend Richtung Ambient-Surfaces gehen, was ich anhand der Achsen allerdings nicht erkennen kann, ist was

die Achsen mir hier überhaupt zeigen. Was sind das jetzt? Meter oder was? Die verwirren mich gerade ein bisschen die Zahlen. Das sind erst mal so Dinge, die ich jetzt herauslesen kann, aus der Heatmap.

Joel Temiz (07:48)

Jawoll! So ausführlich jetzt gar nicht sein müssen.

Dr. Jan Schwarzer (07:50)

Aber Joel, noch als letztes vielleicht: unten diese Farbskala, da weiß ich nicht, was das für Einheiten sind.

Joel Temiz (08:01)

Ehm, soll ich dir kurz erklären. Da die Farbskala unten, das sind halt die die Werte, also jede Position, jeder Frame wird ja oder die Anzahl an Frames, die in einem bestimmten Bereich auf dieser Map markiert wurden. Und die Farbskala sagt dir dann, wie viel Frames da drin sind.

Dr. Jan Schwarzer (08:22)

Und das muss immer dran stehen, weil ich weiß, ich bin da jetzt mehr drin in dem Thema. Wenn das jemand gar nicht weiß, dann durch viele Fragezeichen und du hast auch dann da 1,5 stehen hinter 500 Personen für wahrscheinlich 500 stehen soll, oder?

Joel Temiz (08:35) Jawoll.

Dr. Jan Schwarzer (08:37)

Das würde ich auch noch dahingehend ändern. Okay, aber dass war inhaltlich das, was ich dazu sagen kann, zu der ersten Aufgabe zum vierten Punkt.

Joel Temiz (08:49)

Perfekt, vielen Dank! Lass mich kurz eine Sekunde mit dem Schreiben hinterherkommen. ... Perfekt. Wegen mir dürfen wir gerne zur nächsten Aufgabe gehen.

Dr. Jan Schwarzer (09:21)

Dann die zweite Analyse des zweiten Datensatzes. So, da soll ich jetzt den Datensatz vom und 11.04. und vom 12.04. laden und (Liest vor) entfernen Sie den zuvor gewählten Zeit-Filter. Dann entferne ich erst kurz den Filter. Okay, da ist jetzt, finde ich, das Problem, so ein bisschen, dass ich mir nicht gemerkt habe, was bei dem „bis“ drin stand. Da stand nämlich glaube ich 59 für Minuten drin. Wahrscheinlich 23:59 weiß ich gar nicht mehr. Und beim anderen geh ich jetzt wieder auf 0 Uhr runter und dann auf speichern. So, jetzt habe ich 98 prozent der Records. Nur das kann ja gar nicht sein. Okay, ich habe es mal beides, also Time-Picker-von Time-Picker-bis, beides auf alles auf Null. Trotzdem habe ich beim Speichern... jetzt sagte mir gar nichts mehr. Achso ich muss wahrscheinlich eine Änderung machen, damit ich was sehe. Okay, ich weiß jetzt nicht, ob es funktioniert hat. Ich weiß grad nicht wie viele Daten er geladen hat. Dann gehe ich jetzt aber trotzdem mal kurz in den Daten-Import. Jetzt ist die Frage... Ich soll jetzt zwei Datensätzen hochladen. Okay, 11. und 12.04. öffnen. Okay, da ist tatsächlich jetzt ein bisschen die Schwierigkeit... Ich bin jetzt gerade beim 11.04. Mach „Öffnen“. Soll ich das jetzt hintereinander hochladen? Also kann ich beide Datensätze parallel laden, das kann ich machen. Okay, jetzt habe ich gerade den 11.04. Das ist wahrscheinlich falsch.

Joel Temiz (11:11)

Ja, das ist ein bisschen blöd markiert. Das ist ein Drag-and-Drop-Field und wenn man mehr als einen Datensatz, also mehr als einen Ordner hochladen möchte, dann müsste man entweder beide Dateien oder alle Dateien in einen Ordner verschieben oder per Drag-and-Drop beide Ordner zusammen reinziehen.

Dr. Jan Schwarzer (11:29)

Ja okay, dann mach ich das mal eben kurz. Das Problem ist, ich habe jetzt gar keine Rückmeldung mehr davon. Also er hat anscheinend den 11.04. fälschlicherweise geladen. Aber er

hat mir jetzt endlich mal eine Bestätigung gegeben, wie viel Daten geladen wurden. Ich drücke vielleicht mal Control-F5. Um vielleicht den Cache zu leeren. Keine Daten verfügbar sagt mir jetzt die Meldung. Das ist ja schon mal super. Dann mache ich es jetzt wie du es gerade beschrieben hast.

Joel Temiz (11:55)

Dann eine Sekunde... Gehe mal bitte auf die Filter und klicke neben dem Time-Picker, also neben den beiden Time-Pickern links, wo die die Beschreibung ist Time-Picker-von und Time-Picker-bis... da mal mit einem Doppelklick drauf und schau, was passiert.

Dr. Jan Schwarzer (12:13)

Ah okay, also was ist das? (Lacht)

Joel Temiz (12:15)

Das ist eine eine Geheim-Funktion, die von mir nicht kenntlich gemacht wurde. Und dann nochmal speichern.

Dr. Jan Schwarzer (12:21)

Okay, habe ich dann gespeichert jetzt.

Joel Temiz (12:24)

Funktioniert übrigens bei jeder Zeile.

Dr. Jan Schwarzer (12:27)

Also da kenne ich das so, dass du immer so ein „Stupsi“ dann hast, also default Settings oder sowas. Für so eine Maske hast irgendwo immer einen Button „Restore Default Settings“ oder so was, weil das weiß man ja nicht. Ich meine das ist für dich jetzt klar. Du weißt, dass es geht, aber das sieht man ja als Benutzerin / Benutzer dann ja gar nicht, dass das funktioniert. Ja okay, dann gehe ich jetzt wieder auf Daten-Import und ziehe die beiden Ordner dort rein.

Dr. Jan Schwarzer (12:54)

Also vom 11.04. und dem 12.04. So, die werden jetzt anscheinend alle angezeigt. Ich sage jetzt einfach hochladen... Er lädt.. 1081 Records wurden hochgeladen. Dann soll ich jetzt ins Barchart-Diagramm, dann aktualisiere ich da jetzt. So, jetzt habe ich da wieder eine Anzeige über den Tag verteilt. So, wählen Sie anhand des Barcharts eine geeignete Halbierungsmarke, bei der in beiden Teilen des Datensatzes gleich viele Personen erfasst werden. So okay ist das jetzt dann unter Filter?... Okay, ich muss sagen, dass ich da gerade auf dem Schlauch stehe. Also ich sehe jetzt hier irgendwie einen Punkt unter 11Uhr. Ich weiß jetzt nicht, ob das so eine Halbierungsmarke sein soll, oder ob ich da irgendwas machen können soll

Joel Temiz (14:17)

Ne, also die Halbierungsmarke... das ist jetzt wieder dieser Punkt dem Verständnis für die Aufgabe. Du sollst praktisch einfach sagen „Okay, die Hälfte der Person, die jemals in diesem Raum waren, haben sich zwischen sechs und zwölf Uhr aufgehalten. Die andere Hälfte am Nachmittag sozusagen“.

Dr. Jan Schwarzer (14:40)

Okay, aber dann würde ich zum Beispiel die Halbierung dann bei zwölf Uhr setzen, weil da augenscheinlich anhand des Barcharts ungefähr die Mitte ist, im Sinne von „Danach,, , vielleicht ist es 12:30 oder so. Vielleicht müsste man den Punkt da einsetzen. So meinst du das oder?

Joel Temiz (15:07)

Genau richtig, so ist es perfekt.

Dr. Jan Schwarzer (15:09)

So um den Dreh... Zwischen 12 und 13 Uhr müsste man das wahrscheinlich halbieren, damit man quasi die Anzahl der Benutzer tatsächlich halbiert. Und dann komme ich zum Dritten Punkt der Aufgabe 2: Führen Sie die Halbierung des Datensatzes zuerst für die erste Hälfte in

den Filtern durch. Okay. Damit meinst du dann jetzt sozusagen wahrscheinlich, dass ich jetzt, da wir die Zeit einschränke bis 12:30 zum Beispiel.

Joel Temiz (15:43)

Jawoll.

Dr. Jan Schwarzer (15:47)

Also das heißt, ich lasse jetzt den Time-Picker-von unverändert, den Time-Picker-bis mache ich jetzt auf 12:30 Speicher. 43 Prozent bleiben noch über. Gehe jetzt in den Barchart rein, aktualisiere nochmal. Und da wäre das ja dann sozusagen, ich würde es mal schätzen. Also ungefähr vielleicht bei dem, was mit 10 Uhr, also zwischen 9 und 12 Uhr sind relativ gleich viel, wenn man jetzt im Schnitt zwischen 10 und 11 macht um 10:30 wäre das gleich am Vormittag sind noch ein paar mehr Benutzer, das heißt, ich würde den auch leicht nach links verschieben. Vielleicht. Vielleicht sogar bei 10 Uhr... Um den Dreh... Dann der vierte Punkt Bei Aufgabe 2: (Liest vor) Passen Sie in der Farbpalette der Heatmap die Skalierung der Darstellung nach einem eigenen Ermessen an und sichern Sie die beiden Heatmaps unter geeigneter Bezeichnung. Um überflüssige Farbpaletteneinträge zu erkennen, nutzen die Legende der Heatmap.

Okay also Heatmap aktualisieren erst mal. Okay, (liest vor) passen die Farbpalette der Heatmap, die Darstellung nach eigenem Ermessen an. Okay, so ich klicke jetzt einfach mal auf die Farbpalette, jetzt habe ich die Sachen grau gemacht. Ach so, okay. Okay, also ich nehme mal kurz grün, türkis, türkis, blau, lila, rosa raus. Das heißt, ich kann jetzt sehen... Das ist auf jeden Fall eine sehr nette Idee, dass man jetzt zum Beispiel im... Anhand der der Engagement, eh Entschuldigung, das sind ja die Frames... Also anhand der Anzahl der Frames jetzt wirklich gucken kann, wo tauchen jetzt wie viele Frames in welchem Bereich auf und man sieht tatsächlich, dass der Großteil ja sich bei der Skalierung auf gelb, orange und rot verteilt und ich jetzt sogar die vierte Schwelle mit rein nehme von der von der Heatmap, also rosa, dann sieht man, dass das eigentlich nur wenige Punkte sind. Dann würde ich jetzt mal diese mit gelb, orange und rot markierte Heatmap speichern... (liest vor) Und sichern Sie die beiden Heatmaps. Okay, Auflösung pro Achse, Bewegungsmodus... Okay, alles klar. Ich kann also auch noch die Farben ändern. Dann mache ich jetzt mal aus gelb eine Art türkisfarben, obwohl wir das schon haben, aber das kenne ich zum Beispiel vom Wetterbericht. Dann machen wir mal für orange ein dunkleres Blau und dann für die letzte Rot nehme ich dann vielleicht mal einen Lila-Ton. Also ich merk gerade, dass ich die schon wieder ein bisschen falsch verstanden hab.. Speichern. Okay, jetzt sehe ich quasi mit meinen eigenen Farben tatsächlich die Heatmap. Er hat allerdings nicht meine Einstellung in Bezug auf die Skala und er hat jetzt wieder alle Farbteile wieder drin. Ich hatte vorher ja nur drei drin, die sind jetzt weg. Ich habe auch die erste nicht gespeichert, fällt mir auf die kann ich jetzt versuchen zu speichern. Wenn ich wüsste, wo das geht und. (Liest vor) Und sichern Sie beiden Heatmaps unter geeigneter Bezeichnung. Genau das habe ich ja gemacht.

Joel Temiz (20:03)

Die Aufgabe hast du bislang sehr gut verstanden, also genau richtig eigentlich. und das Speichern kannst du unterhalb der Visualisierung machen.

Dr. Jan Schwarzer (20:13)

Ja, habe ich jetzt auch herausgefunden. Dort sieht „Visualisierung gespeichert“. Ich kann da offensichtlich auch was eingeben, aber er sagt mir nicht wo er sie speichert. Also nenne ich sie einfach mal „Jan-Test“. Ist das im Browsercache oder wo hast du die jetzt hin geschmissen, weil das seh ich jetzt nicht.

Joel Temiz (20:33)

Ehm, in der linken oberen Ecke hast du einen ein Menü also ein Hamburger Menü.

Dr. Jan Schwarzer (20:43)

Ja. Okay, okay, okay, okay. Alles klar. Jetzt habe ich das Problem, dass man Browser mit dem Leben tatsächlich hapert. Okay, jetzt habe ich tatsächlich dadurch dass ich 3 mal geklickt habe, habe ich jetzt drei Abbildungen. Verstehe. Also ich hab jetzt 3 Heatmaps gespeichert. Okay, dann würde ich jetzt zur vorletzten Aufgabe in Aufgabe zwei übergehen. Führen Sie den letzten Schritt für die verbleibende Hälfte durch. Okay, den letzten Schritt. Okay, mir ist jetzt nicht klar... Visualisierungen erstellen, also über „Visualisierung erstellen“ komme ich dann wieder zurück.

Dr. Jan Schwarzer (21:41)

Barchart, okay, dann gehe ich jetzt wieder in den Filter rein, setze jetzt das auf 12:30 bis.. Ach ne.. Doppelklick auf Time-Picker-bis, Speicher. So, geh nochmal kurz zur Kontrolle ins Barchart-Diagramm, aktualisier noch mal, so jetzt habe ich die andere Hälfte, gehe ich in die Heatmap rein, aktualisiere dort noch mal! Okay, hier ich sehe eigentlich keine gravierenden Änderungen zu der Heatmap zuvor, was jetzt die Daten an sich anbelangt, außer ein paar Nuancen vielleicht. Dann würde ich da auch nochmal kurz die Farben anpassen. Zum Beispiel bei Gelb würde ich dann wieder ein Türkis nehmen, ein Bisschen ein anderes, aber dann für Orange nehme ich dann wieder Blau. Und für rot nehme ich jetzt anstatt dessen auch wieder Lila. Speicher das Ergebnis. Jetzt mal aus Spaß, nimm ich mal alles raus außer Türkis, Blau und Lila, Klicke also jetzt unten in die die Farbpalette rein unter der unter der Heatmap. So habe jetzt alles bis auf diese drei Farben dort repräsentiert. Gib jetzt hier einen Namen ein... Jan Test 2, speicher das. Geh zur Kontrolle nochmal auf das Menü oben, geh auf Visualisierung anzeigen. So, sehe die von vorhin. Das dauert bei mir tatsächlich ein bisschen bis das geladen hat.

Joel Temiz (23:37)

Das ist bei mir aber auch so, also die „Visualisierung Anzeigen“-Seite, ist sehr leistungsintensiv.

Dr. Jan Schwarzer (23:45)

Okay, also er hat anscheinend... Er hat gespeichert, aber er hat tatsächlich nicht meine Selektierung angenommen. Ich habe tatsächlich in der Farbpalette nur türkis, blau und lila gehabt und trotzdem noch alle anderen wieder mit drin. Da scheint irgendwas nicht hingehauen zu haben, glaube ich. Genau so scheint es mir. Okay, dann die letzte Aufgabe in Aufgabe 2: (liest vor) vergleichen Sie die beiden gespeicherten Visualisierung und treffen Sie eine Aussage zu dem Vergleich. Das ist ein bisschen schwierig dahingehend, da ich jetzt das Gefühl habe, dass die gespeicherten Daten... Also das Problem ist ich habe jetzt zwei Mal lila (lacht) jetzt in der Visualisierung, weil er meine ursprünglichen Farbcodes sozusagen nicht nur die gespeichert hat, sondern auch die alten mit drin hat. Also okay, ganz grundsätzlich, was mir auffällt ist, dass die Fragmentierung des Bereichs unmittelbar vor der Wand, also da, wo dieses diese Aussparung ist zum Nachmittag hin anscheinend stärker ist. Mir scheinen auch in diesem Datensatz mehrere Fälle zu sein, die sich in Richtung der beiden Ambient-Surfaces bewegt haben. Wenn ich das jetzt richtig interpretiere: die Kamera nimmt ja sozusagen die Perspektive aus Sicht des Benutzers auf.

Dr. Jan Schwarzer (25:16)

Das heißt also, man müsste es drehen, was wir jetzt hier sehen. Das heißt also, was wir in der zweiten Tageshälfte sehen, wo ich eben gerade meinte, da scheinen mir Leute von den Ambient-Surfaces zu stehen, das sieht ein bisschen so aus, als wären die Leute zu dem System rechts von Kamera 2 gegangen, dass sie zum Beispiel am Vormittag gar nicht wissen, was ich nach wie vor der beiden Abbildungen, aber das sehe ich auch gerade so bei meinen Analysen ist nach wie vor der Main-Walking-Path, also der Hauptpfad der Leute ist unverändert immer

zu sehen, mit mehr oder weniger starker Ausprägung. Und das ist im Prinzip auch das, was ich im Kern jetzt dazu zu sagen hab. Ja, das wäre dann Aufgabe 2.

Joel Temiz (26:05) Perfekt, Vielen Dank!

Dr. Jan Schwarzer (26:07)

Okay, dann mal zu Aufgabe 3, oder?

Joel Temiz (26:09)

Jawoll! Aufgabe 3 ist eh relativ kurz

Dr. Jan Schwarzer (26:13)

Dann... (liest vor) Laden Sie den Datensatz vom 02. Bis 05.05. Okay, also ein Paar Ordner, dann gehe ich jetzt wieder sozusagen zu Daten-Import. Und ziehe jetzt dann diese, diese vier Ordner per Drag-and-Drop rein, ok, hat er angezeigt. Sage hochladen... Er lädt... Er lädt immer noch. 2489 Records wurden geladen mit durchschnittlich 50 Frames. Dann gehe ich erst mal wieder auf Visualisierung erstellen. Gehe dann erst mal auf die Filter und stell die aus. Ich warte gerade dass das Tool reagiert.

Dr. Jan Schwarzer (27:46)

Okay jetzt habe ich mir nicht die andere Seite angesehen. Ich sehe immer noch die abgespeicherten Bilder, die ich eben gesehen hab. Mir fällt gerade noch ein als Ergänzung: es wär natürlich super, wenn man die exportieren könnte, die Bilder. Am besten auch vektorbasiert. Also ich gehe in die Browser URL und gebe nochmal die Hauptseite ein damit ich nochmal zurück komm. Und jetzt bin ich zurück, gehe auf Filter, mach wieder auf Time-Picker-von/-bis. Kurz Doppelklick, Speicher. Das heißt, den ersten Punkt habe ich.

Joel Temiz (28:27)

Kurze Ergänzung, da ist auch gemeint, dass man die Flying-Pixel-Filter zurücksetzt.

Dr. Jan Schwarzer (28:43)

Ja, aber die haben noch gar nicht genutzt, oder?

Joel Temiz (28:45)

Ja, der ist standardmäßig auf 50 Zentimeter

Dr. Jan Schwarzer (28:50)

Okay, aber das ist jetzt bei bei Drittens. Also den habe ich ja nie angefasst.

Joel Temiz (29:00)

Genau. Also der Gedanke hier war, dass man es jetzt ändert und später dann wieder zurück ändert, sozusagen.

Dr. Jan Schwarzer (29:09)

Okay, also das wird aus dem ersten Punkt nicht klar. Also ich soll jetzt den Schieber bei Flying-Pixels- Filter ganz links auf Null?

Joel Temiz (29:17)

Nein, einfach mit Doppelklick auch wieder. Dann deaktiviert er sich wieder.

Dr. Jan Schwarzer (29:28)

Dann habe ich den jetzt auf Null, also mit Doppelklick, und hab gespeichert. So... (liest vor) Verbessern Sie die Auflösung der Heatmap indem sie Pixel pro Achse um 100 erhöhen. Was heißt verbessern bei Aufgabe 3?

Joel Temiz (30:04)

Auf dem Reiter Heatmap, also unter der Heatmap. Da steht was von wegen „Pixel pro Achse“.

Dr. Jan Schwarzer (30:13)

Ach okay, alles klar. Sorry, aber hier auf. Okay, ich verstehe diese Anleitung nicht. Ich gebe da jetzt einfach mal 500 ein. Muss ich wahrscheinlich aktualisieren drücken und was macht er jetzt? Okay. Verstehe. Okay, okay, das ich mach jetzt mal tausend Mal, weil das finde ich

eigentlich ganz gut, dass das granularer werden kann. Aber wie gesagt, die Achsen-Beschriftung also... Das ist wirklich ein bisschen verwirrend, weil zum Beispiel seh unten auf der Achse nur noch -269 Punkt Punkt Punkt und -236 Punkte Punkt Punkt und das ein bisschen verwirrend. Und dann zum Beispiel kleinere Zahlen haben da ganz lange Nachkommastellen. Okay, für tausend lädt immer noch.

Joel Temiz (31:12)

Ja, das kann sein, dass der die Tausend nicht so gern mag, weil es ja dann eine Auflösung von 1000 zum Quadrat ist.

Dr. Jan Schwarzer (31:20)

Also ich habe es jetzt auf jeden Fall drin und ich sehe auch ganz deutlich diese Muster in der Kamera. (liest vor) Verbessern Sie die Auflösung. Das ist wirklich die Frage, was du mit Verbessern meinst. Ich kann das jetzt fein-granularer machen. Sagen wir mal, ich gehe jetzt mal von tausend auf ein hundert, also bei tausend sich sehr deutlich bei 500 schon die Muster, die sehe ich bei 100 nicht.

Joel Temiz (31:41)

„Um“ 100, „indem sie die Pixel pro Achse um 100 erhöhen“, aber es geht auch mit tausend, ist kein Problem. Ist es nur die Frage, ob der Browser dabei mitmacht. Der ist dabei die limitierende Komponente.

Dr. Jan Schwarzer (31:57)

Dass man um 100 erhöhen soll habe ich tatsächlich überlesen. Also tausend funktionierte. Mit 1000 habe ich diese Muster gesehen. Jetzt habe ich gerade 100 getestet. Da sieht man diese Effekte nicht. Jetzt geh noch mal zurück auf 300 auf den Default Wert. Da sehe ich die Effekte ein bisschen, dann gehe ich jetzt mal 400. Also ich tu mich ein bisschen schwer mit dem Wort verbessern. Also die Frage ist ein bisschen, was du halt erreichen willst mit denen mit der feineren Graduierung. Was jetzt das Prinzip ein bisschen hilft, ist sozusagen dieses Phänomen. Vielleicht noch mal deutlicher zu sehen bin ich jetzt auf 500 oder 600, das einfach diese Muster deutlicher werden. Das hilft vielleicht dabei. Bei 600 lädt er auch schon ein bisschen und genau jetzt finde ich bei 600 fängt das langsam an auch wirklich deutlich zu werden. Probieren wir mal noch 700. Er lädt und lädt. Also ich sehe das ja mit zunehmender Auflösung, diese Sand-Dünnen-Effekte, sehe ich halt besser dadurch, dass ich diese Auflösung pro Achse höher stelle. Okay, der dritte Punkt bei Aufgabe 3: (liest vor) finden Sie den Punkt in der Heatmap, an dem die meisten Frames aufgezeichnet wurden. Und sichern Sie die Visualisierung unter geeignetem Namen und... Okay. Finden Sie den Punkt in der Heatmap, an dem die meisten Frames aufgezeichnet wurden? Okay, das heißt jetzt natürlich die Frage, bei welcher Auflösung sie jetzt mal auftreten zu. So genau und da ist jetzt mit lila bei mir markiert gerade der Bereich vor den Toiletten und der Laufpfad also wieder diese Kurve, die hat hier die meisten Nutzer mit lilafarbenen und rosafarbenen Bereichen und dann davon weggehend, dann Richtung Ambient-Surfaces, werden die Bereiche dann rot, orange, gelb. Genau das sind im Prinzip die, so verstehe ich zumindest jetzt, die Bereiche, wo am meisten aufgezeichnet wurde.

Joel Temiz (34:43)

Ergänzend gibt es in dieser Visualisierung einen ganz konkreten Punkt, an dem ein extrem hoher Wert vorhanden ist. Das sind nur ein oder zwei Pixel, vielleicht drei Pixel.

Dr. Jan Schwarzer (35:00)

Ich bin jetzt bei der Auflösung pro Achse von 300 und wenn ich jetzt auf diese Punkt hover, also das ist dann türkis. Der ist bei mir jetzt gerade bei der Auflösung von 300 bei minus 42. Und dann bei 225. Doppelpunkt 45. Das meinst du wahrscheinlich. Da wäre natürlich hilfreich, dass man sozusagen rein zoomen könnte, weil wenn du das siehst.

Dr. Jan Schwarzer (35:33)

Ich bin jetzt mal bei der Auflösung auf 200 runter und jetzt sehe ich diesen Türkis-Ton vielleicht ein bisschen besser, aber da wäre natürlich super, wenn man sozusagen mit dem Mausrad so reinscrollen könnte zum Beispiel.

Joel Temiz (35:46)

Es gibt aber meines Wissens noch einen Punkt der noch größere Werte hat. Das sollte bei 200 genau so wie bei 3 und 400 noch vorhanden sein. Dazu kannst du mal unten auf der Legende wie du vorhin auch schon rausgefunden hast, kannst du temporär. Das ist nämlich die Funktion von dieser Legende, also von diesen Balken, von den kleinen Farb-Balken. Da kannst du von links nach rechts mal anfangen, die Elemente weg zu klicken und dann fällt auf, dass dann am Ende noch dieser 1500 Balken übrig ist. Und dann sieht man eigentlich die Höhe des verbleibenden Werts.

Dr. Jan Schwarzer (36:19)

Ah, der ist in der Wand, oder? Genau. Also der ist jetzt sozusagen in der Kurve. Okay, ich nehme jetzt alles raus bis auf grün. Genau jetzt weiß ich, was du meinst und der ist jetzt bei 28, 120: 6. Da sehe ich den jetzt. Okay, dann verstehe ich jetzt, was gemeint ist. So.. (liest vor) Und sicher Sie die Visualisierung unter geeignetem Namen. Okay, dann nenne ich das jetzt mal einfach „meiste Frames“ und speicher das. So... (liest vor) filtern Sie anschließend Records aus, derer Koordinatensprünge größer als 50 Zentimeter sind und überprüfen Sie die vorher ermittelten Koordinaten im Vergleich. Okay, das heißt Filter, da gehe ich jetzt auf max. Flying Pixels, das schiebe ich jetzt hoch aus 0,50 und klicke auf Speichern.

Dr. Jan Schwarzer (37:42)

Jetzt da lädt er. So gehe wieder zurück auf die Heatmap, aktualisiere. Mach wieder die Datenpunkte links von grün aus. Aber ich sehe auch schon keine türkisfarbene mehr, ach doch. Türkis ist auch noch da. Aber wenn ich nur noch grün drin habe, habe ich tatsächlich nichts mehr. Okay, ich schreibe da mal aus Spaß „meiste Frames 2“ rein und mach speichern.

Dr. Jan Schwarzer (38:23)

Guck mir die kurz an. Es dauert es wahrscheinlich einen Moment. Ach nein okay, sind jetzt nur noch die zwei. So, Ja, das Problem ist ein bisschen wieder: er hat jetzt wieder, obwohl ich ja beide Abbildungen gespeichert habe mit grün, also wo ich nur die Farbskala grün vertreten hatte, hat er mir jetzt trotzdem wieder mit allen Farben abgespeichert. Also ich sehe zwar jetzt wo der Grüne war. Aber das hilft mir jetzt nicht, weil ich tatsächlich diese Sachen miteinander vergleichen kann in dem Fall. Das ist jetzt natürlich schwierig und ich stelle auch fest, wenn ich in der oberen von den beiden gespeicherten Abbildungen mit der Maus jetzt unterwegs bin, dann zeigt er mir die Koordinaten in der zweiten Abbildung darunter an. Wenn ich jetzt unten... Ich habe jetzt quasi diese beiden Abbildung und wenn ich in der unteren hover zeigt er mir das an, und wenn ich das in der Oberen mache, dann zeigt er mir das auch nach wie vor in der Unteren an.

Joel Temiz (39:28)

Okay, das scheint ein Bug zu sein. Das liegt wahrscheinlich daran, weil TypeScript da Shallow-Copies macht was er eigentlich nicht machen sollte.

Dr. Jan Schwarzer (39:39)

Gut, dann gehe ich jetzt zurück zum Erstellen. Dann soll ich jetzt, jetzt ist das wieder anscheinend ohne jegliche Art von Filterung, Auflösung ist wieder auf 300, und ich sehe wieder alle Farben in der Heatmap. Und jetzt soll ich nach Personen filtern, die Happy und Nicht Happy sind. Okay, also ich filter die Personen nach Happy und Nicht Happy, (liest vor) speichern sie jeweils den Barchart. Okay, alles andere lasse ich jetzt so, wie es ist. (liest vor) Egal, nur ohne..

Okay. Also ja da würde ich jetzt etwas erwarten wie Ja, Nein. Aber Okay. Dann mache ich jetzt mal „nur“, Speichern. Okay... 4,5 Prozent der Daten, gehe ins Barchart-Diagramm aktualisiere. Okay, ich speicher den jetzt mal unter „happy yes“ und speichern. Gehe dann zurück in die Filter-Einstellungen, wähle Happy Ohne aus, speichern. 78 Prozent... Oh, da sind aber viele Leute unglücklich, nicht gut. So, dann „Happy No“, speichern. Dann gehe ich wieder auf Visualisierung anzeigen. Ach, ich habe einen Fehler gemacht. Okay, ich habe nicht aktualisiert. Visualisierung erstellen.

Dr. Jan Schwarzer (41:48)

Ich muss das wahrscheinlich wieder von vorne beginnen, so wie es aussieht. Okay, sorry. Also, ohne, speichern. Okay, Barchart aktualisieren, so „Happy No 2“, Speichern. Er lädt jetzt wieder gerade. Ach ne, ich muss jetzt gerade ein bisschen umständlich hoch und runter scrollen in der Anzeige, weil ich jetzt natürlich noch ein „Happy No“ drüber habe, was nicht richtig war. Okay, also was fällt mir auf... Also Leute sind eigentlich den ganzen Tag über unhappy. Also fangen von Anfang an: am Mittag sehe ich keine Unterschiede. Obwohl Entschuldigung, ich gucke gerade jetzt auf die Achse, auf die y-Achse links. Wir haben natürlich massiv, also fast zehnmal so viele Leute mit einem No-Happy, oder was auch immer.

Dr. Jan Schwarzer (43:29)

Wahrscheinlich sind das ja auch alle anderen Attribute, außer „yes“ also viel mehr. Da ist natürlich jetzt die Frage, da das nicht nur Ja und Nein ist in dem Framework: was sind jetzt da die anderen Werte? Sind alle Werte außer „Yes“ oder tatsächlich „Nein“?

Joel Temiz (43:47)

Das ist dann tatsächlich „Nein“ Werte. Also wenn du die beiden aufsummiert, „nur“ und „ohne“, dann kommen ja auch keine hundert Prozent raus, meines Wissens.

Dr. Jan Schwarzer (43:56)

Das kann sein. Ich habe mir die Prozente gerade nicht gemerkt.

Joel Temiz (44:00)

Beim Ersten waren es glaube ich 4 Prozent, beim Zweiten waren es 78 Prozent. Wenn mich nicht alles täuscht und der Rest ist dann „unknown“.

Dr. Jan Schwarzer (44:10)

Man kann dann auf jeden Fall daraus schließen, oder zumindest sagen, dass die Kamera meint, dass ein viel, viel größerer Anteil, beispielsweise um 12 Uhr um Faktor 10 zumindest nicht den Anschein macht, dass sie glücklich sind. Das erstreckt sich dann von der absoluten Zahl der der Person tatsächlich über den ganzen Tag wieder. Wenn man jetzt die Zahl relativ betrachtet, unabhängig von der absoluten Zahl, dann sieht man auf jeden Fall Ähnlichkeiten am Mittag. Man sieht insofern Ähnlichkeiten, dass um elf, die auch um 13 Uhr ähnliche Ausschläge sind.

Dr. Jan Schwarzer (44:47)

Aber sonst ist tatsächlich einfach diese Erkennung bei „Happy No“ deutlich besser und mehr über den Tag verteilt, als es bei „Yes“ der Fall ist. Das ist auf jeden Fall interessant. Die Frage ist natürlich, wie gut das funktioniert. Man könnte jetzt daraus schließen, dass die Leute, wenn sie zu Mittag gehen, gar nicht fröhlich sind, obwohl da auch ein paar dabei sind, die als fröhlich erkannt wurden. Ich finde es schwer, das wirklich zu beurteilen, weil ich mit diesem Attribut selber nicht gearbeitet habe.

Dr. Jan Schwarzer (45:19)

Ich habe es zwar in den Daten drin aber auf jeden Fall interessant ist die Frage, inwiefern das tatsächlich stimmt und auch, was daraus die Implikationen sind, weil man jetzt tatsächlich über den Campus läuft. Bei euch oder so ist man ja auch nicht immer am Grinsen. Das ist nicht, wie das im Detail funktioniert. Das ist im Prinzip das, was mir jetzt zu den beiden

Abbildungen einfällt.

Joel Temiz (45:42)

Jawoll perfekt. Das war es auch schon. Dann würde ich die Aufnahme jetzt wieder beenden.

Anhang C. Evaluation Fragebogen

1. Wie gut erfüllt die Applikation Ihre Ansprüche im Allgemeinen?*

Wählen Sie eine Antwort

Extrem gut

Sehr gut

Gut

Nicht so gut

Gar nicht gut

2. Wie empfanden Sie die Navigation auf der Seite?*

Wählen Sie eine Antwort

Extrem gut

Sehr gut

Gut

Nicht so gut

Gar nicht gut

3. Wie ansprechend fanden Sie die Applikation aus optischer Sicht?*

Wählen Sie eine Antwort

Extrem ansprechend

Sehr ansprechend

ansprechend

Nicht so ansprechend

Gar nicht ansprechend

4. Wie verständlich war der Inhalt der Visualisierung?*

Wählen Sie eine Antwort

Extrem verständlich

Sehr verständlich

verständlich

Nicht so verständlich

Gar nicht verständlich

5. Wie verständlich waren die Filterungsmöglichkeiten?*

Wählen Sie eine Antwort

Extrem verständlich

Sehr verständlich

verständlich

Nicht so verständlich

Gar nicht verständlich

6. Wie empfanden Sie die Ladezeiten der Applikation im Allgemeinen?*

Wählen Sie eine Antwort

Viel zu lang

Etwas zu lang

Zufriedenstellend

Sehr gut

7. Wie empfanden Sie die Ladezeiten beim Hochladen eines neuen Datensatzes?*

Wählen Sie eine Antwort

Viel zu lang

Etwas zu lang

Zufriedenstellend

Sehr gut

8. Wie empfanden Sie die Ladezeiten beim Aktualisieren der Filter?*

Wählen Sie eine Antwort

Viel zu lang

Etwas zu lang

Zufriedenstellend

Sehr gut

9. Wie empfanden Sie die Ladezeiten beim Aktualisieren der Visualisierung?*

Wählen Sie eine Antwort

Viel zu lang

Etwas zu lang

Zufriedenstellend

Sehr gut

10. Wie umfangreich empfanden Sie die Auswahl an Filterungsmöglichkeiten?*

Wählen Sie eine Antwort

Alle benötigten Filter vorhanden

Besteht Verbesserungspotential

Unzureichend

11. Wie umfangreich empfanden Sie die Auswahl an Visualisierungen?*

Wählen Sie eine Antwort

Alle benötigten Visualisierungstypen vorhanden

Besteht Verbesserungspotential

Unzureichend

12. Welche Visualisierungsart könnte dem Nutzer noch einen Mehrwert bringen?

Beantwortung der Frage in der Form "Visualisierungsart --> Visualisierungsziel"

Beispiel: PiChart --> Darstellung der Aufteilung in Subtle, Direct und Passive User

(1) Scatter Plott -> Laufwege; (2) Clustering -> Explorativ Ähnlichkeiten in den Daten erkennen können.

500

13. Welche Filterungsmöglichkeiten könnte dem Nutzer noch einen Mehrwert bringen?

(1) Alle Kinect Property Features wie Engaged, Happy etc. (2) Zusätzlich Auswahl verschiedener Modi für die Heat Maps. (3) Mehr Unterscheidungen für "Anzuzeigende Nutzergruppen". (4) Filter für Distanzzonen in Ergänzung zu "Distanz zur Kamera in meter min / max".

14. Für wie sinnvoll halten Sie die Applikation als Web-App?*

Wählen Sie eine Antwort

Sehr sinnvoll

Sinnvoll

Gleichermaßen

Weniger sinnvoll

Überhaupt nicht sinnvoll

15. Wie bewerten Sie den Gesamteindruck der Applikation?*



1

2

3

4

5

6

7

8

9

10

16. Wie viel Potential sehen Sie in der Weiterentwicklung der Applikation*

Wählen Sie eine Antwort

Zwecklos

Zur zweckmäßigen Nutzung muss viel überarbeitet werden

Sie bietet ein gutes Fundament und sollte auf jeden Fall weiter entwickelt werden

Sie hat bis auf wenigen Anpassungen wenige verbesserungswürdiger Punkte

Sie ist perfekt

17. Wie wahrscheinlich ist es, dass Sie diese App verwenden oder weiterempfehlen?*

Wählen Sie eine Antwort

Sehr wahrscheinlich

Wahrscheinlich

Eventuell

Eher unwahrscheinlich

Sehr unwahrscheinlich

18. Abschließende Anmerkungen?

Haben Sie weitere Punkte, die bislang ungesagt geblieben sind, teilen Sie mir diese gerne hier mit

Hi Joel. Ich habe versucht, hier meine ehrliche Rückmeldung zu geben und hoffe, dass es in Deinen Ohren nicht zu negativ klingt. Wir besprachen heute im Prinzip alles; es gibt hier und da Verbesserungspotenzial. Wir müssten perspektivisch die Anwendung noch weiterentwickeln aber Stand heute ist diese ein gutes Fundament. Tolle Arbeit, auch mit Blick auf die kurze Bearbeitungszeit! Grüße