

DIPLOMARBEIT



KUNDENFREQUENZMESSER

Ausgeführt im Schuljahr 2019/20 von: Betreuer:

Jerkovic Josip	6CBKIF	Prof. Mag. Oberkersch Manfred
Mijatovic Alexander	6CBKIF	
Weinberger Alexander	6CBKIF	

Wien, im Juni 2020

EIDESSTÄTTLICHE ERKLÄRUNG

Wir erklären an Eides statt, dass wir die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht haben.

Wien, am 25.06.2020

.....

Jerkovic Josip

.....

Mijatovic Alexander

.....

Weinberger Alexander

KURZFASSUNG

Im Rahmen der Diplomarbeit sollen typische Fragen die sich Unternehmer stellen wie z.B. wie viele Kunden besuchen eigentlich mein Geschäft? Wann sind meine kundenschwächsten Tage oder Zeiträume? Wann sind meine am stärksten besuchten Tage und Zeiträume? Wann sollte ich Werbeaktionen und Sonderangebote starten, die wiederum für mehr Umsatz sorgen könnten? Wie viele Kunden hat meine letzte Werbeaktion in meinen Laden gelockt? Wie steht diese Besucherzahl in Relation zu meinen Umsätzen? Um solche Fragen zu beantworten - und um eine bessere Einschätzung der Lage zu ermöglichen - bietet es sich an, die Kundenfrequenz zu wissen.


Die technische Umsetzung des Kundenfrequenzmessers soll auf der Hardwareebene mithilfe von Kleincomputern und Kameras stattfinden. Auf der Softwareebene sollen die gesammelten Daten in einer der gängigen Datenbanklösungen gespeichert und anschließend grafisch mithilfe eines Browsers für den Kunden bereitgestellt werden.

Das Projekt-Team besteht aus drei Mitgliedern, jedes Mitglied übernimmt eigenverantwortlich einen Teilbereich und trägt so zur erfolgreichen Umsetzung dessen bei. Die Versionskontrolle der Codebase wird durch die Verwendung einer online Versionskontrolle unterstützt.

DIPLOMARBEIT

DOKUMENTATION

Namen der Verfasser/innen	Jerkovic Josip Mijatovic Alexander Weinberger Alexander
Jahrgang/ Klasse Schuljahr	2019/20 6CBKIF
Thema der Diplomarbeit	Kundenfrequenzmesser
Kooperationspartner	
Aufgabenstellung	Die Entwicklung einer möglichst kostengünstigen Lösung, um mithilfe von Gesichtserkennung die Kundenfrequenz eines Unternehmens aufzuzeichnen und diese grafisch anhand von Diagrammen darzustellen.
Realisierung	Zu diesem Zweck wurde mit dem Kleincomputer Raspberry Pi 4 und der Raspberry Pi Camera V2 ein System entwickelt, worauf eine Software läuft, die durch Gesichtserkennung die Kundenfrequenz misst und diese an eine Datenbank übermittelt. Die Visualisierung dieser Daten anhand von Diagrammen erfolgt über eine Webseite und wahlweise einer Android App.
Ergebnisse	Fertigstellung eines Kundenfrequenzmessers und einer Webseite bzw. Android App für die visuelle Darstellung der Daten.

Bildbezeichnung	
-----------------	--

Teilnahme an Wettbewerben: Auszeichnungen	Keine
--	-------


Möglichkeiten der Einsichtnahme in die Arbeit	Bibliothek bzw. Abteilungsvorstellung BI/BI
--	--

Approbation (Datum / Unterschrift) 14.05.2014	Prüfer/in	Abteilungsvorstand / Direktion DI Andreas Heinbach
---	-----------	--

DIPLOMA THESIS

DOCUMENTATION

Authors	Jerkovic Josip Mijatovic Alexander Weinberger Alexander
From Academic year	2019/20 6CBKIF
Topic	People Countier
Co-operation partners	
Assignment of tasks	The development of a cost-effective solution that uses face tracking to record the customer frequency of a company and to display this graphically using diagrams.
Realization	For this purpose, a system was developed with the small computer Raspberry Pi 4 and the Raspberry Pi Camera V2 on which a software runs that measures the customer frequency by face tracking and transmits the frequency to a database. The transmitted data is visualized using diagrams via a website and optionally an Android app.
Results	Completion of a customer frequency meter and a website and Android app for the visual display of the data.

	 KUNDENFREQUENZMESSER
--	--

Participation in competitions Awards	none

Accessibility of diploma thesis	Blocking Note This examination must not be read by a third party!
---------------------------------	--

Apporaval (Date/Sign)	Examiner	Head of Department/ College DI. Andreas Heinbach
May 14th 2020		

ABSTRACT

In the context of the diploma thesis, typical questions that entrepreneurs ask themselves, such as how many customers actually visit my shop? When are my customer's weakest days or periods? When are my most visited days and periods? When should I start promotions and special offers that could in turn generate more sales? How many customers did my last promotion attract to my shop? How is this number of visitors in relation to my sales? should be solved. In order to answer these questions - and to enable a better assessment of the situation - it makes sense to know the customer frequency.

The technical implementation of the customer frequency meter should take place on the hardware level with the help of small computers and cameras. At the software level, the data collected should be saved in one of the common database solutions and then made available to the customer graphically using a browser.

The project team consists of three members, each member is responsible for taking over a sub-area and thus contributing to the successful implementation of this. Version control of the codebase is supported by using an online version control.



1	ALLGEMEINE EINLEITUNG	- 15 -
1.1	DIE VISION.....	- 15 -
1.2	VEREINFACHTER PRODUKTENTWURF	- 16 -
1.3	ZIELE.....	- 17 -
1.4	NICHT ZIELE	- 17 -
1.5	AUFGABENAUFTEILUNG.....	- 17 -
1.6	IDIVIDUELLE ANFORDERUNGEN	- 17 -
1.7	TESTVORGÄNGE.....	- 18 -
2	RESSOURCENPLANUNG.....	- 18 -
2.1	VERWENDETE HARDWARE	- 18 -
2.1.1	<i>Raspberry Pi 4</i>	<i>- 19 -</i>
2.1.2	<i>Raspberry Pi Camera V2.....</i>	<i>- 19 -</i>
2.1.3	<i>Raspberry Pi 4 Case.....</i>	<i>- 19 -</i>
2.1.4	<i>Powerbank</i>	<i>- 19 -</i>
2.1.5	<i>Notebook Acer TravelMate TMX349-M-3373.....</i>	<i>- 20 -</i>
2.1.6	<i>Notebook Lenovo IdeaPad G50</i>	<i>- 21 -</i>
2.1.7	<i>Notebook DELL XPS 13 9360</i>	<i>- 22 -</i>
2.1.8	<i>Smartphone OnePlus 5T.....</i>	<i>- 23 -</i>
2.1.9	<i>Ipad Air 2.....</i>	<i>- 23 -</i>
2.2	VERWENDETE SOFTWARE	- 24 -
2.2.1	<i>Design</i>	<i>- 24 -</i>
2.2.1.1	<i>Adobe Photoshop</i>	<i>- 24 -</i>
2.2.1.2	<i>Adobe Illustrator.....</i>	<i>- 24 -</i>
2.2.1.3	<i>Adobe Xd.....</i>	<i>- 25 -</i>
2.2.2	<i>IDEs und Editoren</i>	<i>- 25 -</i>



2.2.2.1	Visual Studio Code	- 25 -
2.2.2.2	Thonny	- 25 -
2.2.3	<i>Datenbanken</i>	- 26 -
2.2.3.1	MySQL	- 26 -
2.2.3.2	phpMyAdmin	- 26 -
2.2.3.3	XAMPP	- 26 -
2.2.4	<i>Sonstige Software</i>	- 27 -
2.2.4.1	PuTTY	- 27 -
2.2.4.2	FileZilla	- 27 -
2.2.4.3	Git und GitHub	- 27 -
3	HARDWARE DES KUNDENFREQUENZMESSERS	- 28 -
3.1	EINLEITUNG	- 28 -
3.2	AUFGABENSTELLUNG HARDWARE	- 28 -
3.3	VERWENDETE TECHNOLOGIEN	- 29 -
3.3.1	<i>Raspberry Pi 4</i>	- 29 -
3.3.2	<i>OKdo Power-Hat</i>	- 30 -
3.3.3	<i>Raspberry Pi Camera V2</i>	- 31 -
3.3.4	<i>Ansmann Powerbank 10.8</i>	- 32 -
3.3.4.1	VNC Viewer	- 33 -
3.3.4.2	Softperfect Network Scanner	- 33 -
3.3.4.3	PuTTY	- 33 -
3.4	UMSETZUNG	- 34 -
3.4.1	<i>Bau</i>	- 34 -
3.4.2	<i>Installation</i>	- 39 -
3.4.2.1	Freigabe	- 39 -
3.4.2.2	Betriebssystem	- 41 -
3.4.2.3	Python / Numpy	- 44 -

3.4.2.4	Libraries	- 45 -
3.4.2.5	Pip-Install	- 46 -
3.4.2.6	Verworfenen Ideen	- 47 -
4	DATENBANKEN DES KUNDENFREQUENZMESSERS.....	- 50 -
4.1	VERWENDETE TECHNOLOGIEN	- 50 -
4.1.1	<i>MySQL und SQL</i>	<i>- 50 -</i>
4.2	AUFGABENSTELLUNG DATENBANKEN	- 50 -
4.3	DATENBANKSTRUKTUR	- 51 -
4.3.1	<i>Relationen</i>	<i>- 52 -</i>
4.3.2	<i>Primary und Foreign Keys.....</i>	<i>- 52 -</i>
4.3.3	<i>Constraints.....</i>	<i>- 52 -</i>
4.3.4	<i>Stored Procedures</i>	<i>- 53 -</i>
5	SOFTWARE DES KUNDENFREQUENZMESSERS.....	- 54 -
5.1	AUFGABENSTELLUNG SOFTWARE.....	- 54 -
5.2	VERWENDETE TECHNOLOGIEN	- 55 -
5.2.1	<i>Python</i>	<i>- 55 -</i>
5.2.2	<i>OpenCV</i>	<i>- 56 -</i>
5.2.3	<i>Virtualenv</i>	<i>- 57 -</i>
5.2.4	<i>PHP</i>	<i>- 57 -</i>
5.2.5	<i>Cron-Daemon</i>	<i>- 58 -</i>
5.2.6	<i>Haar Cascade Classifier</i>	<i>- 58 -</i>
5.3	UMSETZUNG	- 59 -
5.3.1	<i>Verzeichnisstruktur.....</i>	<i>- 59 -</i>
5.3.2	<i>Cronjobs</i>	<i>- 61 -</i>
5.3.3	<i>Starten des Facetracking-Programms.....</i>	<i>- 63 -</i>
5.3.4	<i>Das Facetracking Programm</i>	<i>- 63 -</i>

5.3.5	Modifikationen	- 64 -
5.4	DATENBANK	- 68 -
5.4.1	Struktur	- 68 -
5.4.2	Datensatz erstellen	- 70 -
5.4.3	Datensatz aktualisieren	- 71 -
6	ANDROID APPLIKATION	- 74 -
6.1	AUFGABENSTELLUNG ANDROID APPLIKATION	- 74 -
6.1.1	Funktionsweise	- 74 -
6.1.2	Design	- 76 -
6.2	VERWENDETE TECHNOLOGIEN	- 77 -
6.2.1	Java	- 77 -
6.2.2	JDBC	- 77 -
6.2.3	MPAndroidChart	- 77 -
6.3	UMSETZUNG	- 78 -
6.3.1	Nebenbemerkung	- 78 -
6.3.2	Activity	- 78 -
6.3.3	XML	- 79 -
6.3.4	Java Code	- 79 -
6.3.4.1	AsyncTask	- 80 -
6.3.4.2	Activity Login	- 81 -
6.3.4.3	Activity Dashboard	- 84 -
6.3.4.4	Activitys Ansichten	- 88 -
7	WEBAUFTTRITT DES KUNDENFREQUENZMESSERS	- 89 -
7.1	AUFGABENSTELLUNG WEBAUFTTRITT	- 89 -
7.1.1	User Experience	- 89 -
7.1.1.1	Accessibility	- 90 -

7.1.1.2	Responsive Webdesign	- 91 -
7.1.2	<i>Use-Cases</i>	- 91 -
7.1.2.1	Landing Page	- 92 -
7.1.2.2	Login.....	- 92 -
7.1.2.3	Kundenbereich Dashboard	- 92 -
7.1.3	<i>Verwendete Technologien</i>	- 93 -
7.1.3.1	HTML	- 93 -
7.1.3.2	CSS	- 94 -
7.1.3.3	JavaScript.....	- 95 -
7.1.3.4	Highcharts	- 95 -
7.1.3.5	AJAX	- 96 -
7.1.3.6	XAMPP.....	- 96 -
7.1.3.7	PHP.....	- 97 -
7.1.3.8	WAVE.....	- 98 -
7.1.3.9	NVDA Screen Reader.....	- 98 -
7.2	UMSETZUNG FRONTEND	- 99 -
7.2.1	<i>Aufbau HTML-Dokumente</i>	- 99 -
7.2.1.1	Generelle Strukturelemente	- 99 -
7.2.1.2	Elemente Landing Page	- 102 -
7.2.1.3	Login Formular	- 103 -
7.2.2	<i>CSS Layout</i>	- 105 -
7.2.2.1	Grid	- 105 -
7.2.2.2	Breakpoints.....	- 107 -
7.2.2.3	Flexbox	- 109 -
7.2.2.4	Wiederverwendung CSS-Code	- 110 -
7.2.3	<i>Visualisierung der Kundenfrequenz</i>	- 111 -
7.2.3.1	Highcharts	- 111 -



7.2.3.2	AJAX	- 114 -
7.2.3.3	Button Funktionalitäten	- 116 -
7.3	UMSETZUNG BACKEND	- 117 -
7.3.1	<i>Wiederverwendung HTML-Code</i>	- 117 -
7.3.2	<i>Authentifizierung</i>	- 117 -
7.3.2.1	Login / Logout	- 118 -
7.3.2.2	Account anlegen	- 120 -
7.3.3	<i>Daten für das Diagramm</i>	- 121 -
7.3.3.1	Datenbankverbindung	- 121 -
7.3.3.2	Initiales Auslesen	- 122 -
7.3.3.3	Änderung des Zeitraums	- 123 -
7.4	TESTING	- 125 -
7.4.1	<i>Lokale Entwicklungsumgebung</i>	- 125 -
7.4.2	<i>Browser Developer Tools</i>	- 125 -
7.4.3	<i>Accessibility Tests</i>	- 126 -
8	SCHLUSSWORT	- 129 -
9	ABBILDUNGSVERZEICHNIS	- 131 -
10	LITERATURVERZEICHNIS	- 136 -



1 Allgemeine Einleitung

1.1 Die Vision

Die Welt der Unternehmen wird von Zahlen regiert und viele Manager stellen sich Fragen wie, „Wie viele Kunden besuchen eigentlich mein Geschäft?“, oder „Wann ist der stärkste und wann der schwächste Tag?“.

Mit der Fertigung eines Kundenfrequenzmessers werden diese Fragen beantwortet und bieten eine Möglichkeit der Einschätzung des Geschäfts und eine Struktur für die weitere Entwicklung dessen.

Da Frequenzmessung mit Gesichtserkennung recht teuer werden kann ist es das Ziel der Diplomarbeit gewesen eine günstige Alternative zu bieten.

Die Erfassung der Frequenz erfolgt durch einen Kleincomputer, dem Raspberry Pi der im Eingangsbereich des Geschäfts angebracht wird. Eine Kamera, die an dem Kleincomputer angebracht ist, erfasst Kunden. Die gesammelten Daten werden vom Raspberry Pi an eine Datenbank übermittelt und gespeichert.

Die Kundenfrequenz sollte für Unternehmen anhand einer Website grafisch in der Form eines Diagrammes abrufbar sein. Weiters soll es dem Kunden möglich sein von überall auf die Daten zugreifen zu können und so eine größtmögliche Flexibilität im Umgang mit unserem Produkt bieten. Auch das Entwickeln einer Webseite und nicht einer App unterstreicht diesen Punkt weiter. Webseiten sind Plattform unabhängig und fordern nur eine Internetverbindung, wohingegen Applikationen für Plattformen, wie Android oder Apple entwickelt werden.

1.2 Vereinfachter Produktentwurf

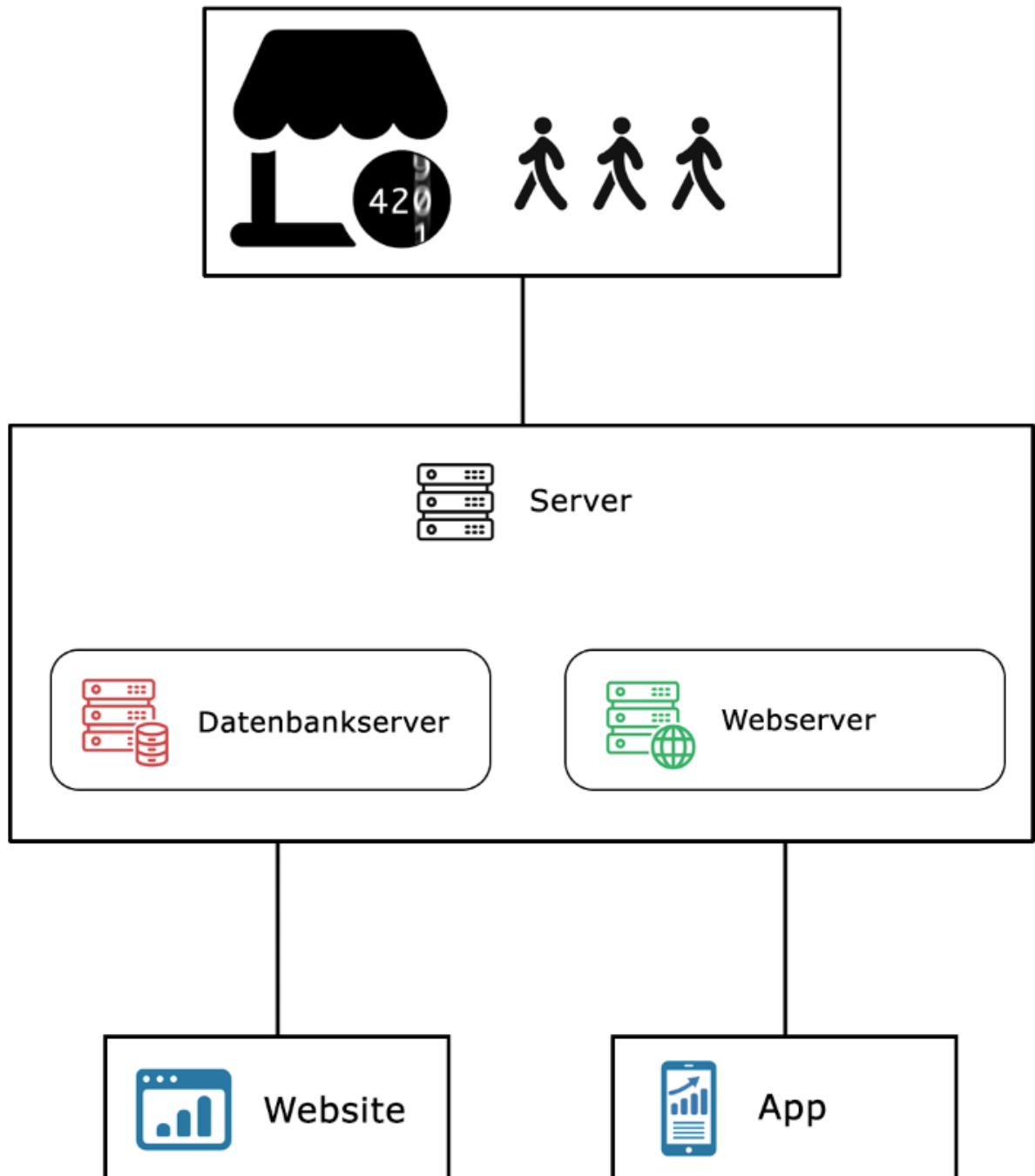


Abbildung 1: Vereinfachter Produktentwurf

1.3 Ziele

Das Ziel dieses Projektes ist die Entwicklung eines möglichst kostengünstigen Kundenfrequenzmessers. Dieser soll mithilfe von Gesichtserkennung die Frequenz der Kunden messen. Eine Webseite, auf der sich der Endkunde einloggen kann, soll eine personalisierte Darstellung der Kundenfrequenz anhand von Diagrammen bieten.

1.4 Nicht Ziele

Es ist nicht Teil des Projektes, dass die Gesichtsdaten in jeglicher Form abgespeichert werden. Ebenso wenig wird die Verweildauer der gezählten Personen gemessen oder die gemessene Frequenz in Verbindung mit dem Umsatz gebracht.

1.5 Aufgabenaufteilung

Die Aufgaben wurden so aufgeteilt, dass jeder Teilnehmer ungefähr den gleichen Arbeitsaufwand hat. Eine Abweichung des tatsächlichen Aufwandes kann natürlich abweichen da im Laufe des Projektes auch sehr viel themenübergreifend gearbeitet wurde.

1.6 Individuelle Anforderungen

Josip Jerkovic	→	Software, App Entwicklung, Datenbanken
Alexander Mijatovic	→	Hardware, Datenbanken
Alexander Weinberger	→	Webauftritt, Datenbanken



1.7 Testvorgänge

Grundsätzlich wurden im Verlauf des Projektes nach jeder abgeschlossenen Aufgabe die Komponenten und deren Zusammenspiel getestet, um bei der finalen Zusammenführung der Komponenten sicherzustellen, dass es keine Kompatibilitätsprobleme zwischen den einzelnen Komponenten gibt.

Sowohl die Website als auch die App wurden regelmäßig auf verschiedenen Geräten und von Personen die mit dem Projekt nichts zu tun getestet, um konstruktives Feedback zu erhalten.

2 Ressourcenplanung

Nachfolgend ein kurzer Überblick der verwendeten Ressourcen in diesem Projekt. Es wird in den jeweiligen Kapiteln, in denen diese Ressourcen verwendet werden, genauer auf deren Funktionalitäten eingegangen.

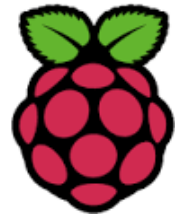
2.1 Verwendete Hardware

Es werden Komponenten eingesetzt, um ein physisches Gerät zu bauen auf welchem die Kundenzählung stattfinden soll. Die in Folge aufgelistete Hardware wird in den einzelnen Abschnitten der Teammitglieder näher beschrieben und die Verwendung und deren Nutzen näher erläutert.



2.1.1 Raspberry Pi 4

Der Raspberry Pi ist ein Einplatinencomputer von der ungefähren Größe einer Kreditkarte in der mittlerweile 4. Version auf dem die Linux-Distribution Raspbian läuft.



*Abbildung 2:
Raspberry Pi Logo,
Quelle: wikipedia.org*

2.1.2 Raspberry Pi Camera V2

Die Pi Camera V2 ist ein Kameramodul für den Kleincomputer Raspberry Pi in der mittlerweile zweiten Version. Die Kamera kann sowohl hochauflösende Videos als auch Standbilder aufnehmen und funktioniert mit allen Modellen von Raspberry Pi.



*Abbildung 3:
Raspberry Pi
Camera*

2.1.3 Raspberry Pi 4 Case

Um den Raspberry Pi physisch zu schützen wird das OKdo Power-Hat Case benutzt. Dieses Case zeichnet sich insofern aus, als dass es einen integrierten Power Button hat.



*Abbildung 4:
OKdo Logo*

2.1.4 Powerbank

Die Ansmann Powerbank 10.8 ist eine 10.000 mAh große Powerbank, welche im Falle einer Trennung der Primären Stromversorgung den Raspberry Pi mit genügend Strom versorgen sollte.



*Abbildung 5: Ansmann
Logo*

2.1.5 Notebook Acer TravelMate TMX349-M-3373

Josip Jerkovic besitzt ein TravelMate 349 Notebook von Acer aus dem Privatbestand.

Optisch macht das Gerät den Eindruck eines Business-Rechners der Oberklasse. Ausstattung und Preisniveau jedoch entsprechen eher der Einstiegsklasse, womit Acer mit dem TravelMate X349 im 14-Zoll-Format Großteils Schüler und Studenten anpeilt.



Abbildung 6: Acer TravelMate TMX349-M-3373

Eckdaten der Hardware-Spezifikationen:

CPU: Intel Core i3-6100U, 2x 2.30GHz

GPU: Intel HD Graphics 520 (IGP)

RAM: 4GB DDR4

HDD: 1x 256 GB SSD

2.1.6 Notebook Lenovo IdeaPad G50

Alexander Mijatovic besitzt ein IdeaPad G50 Notebook von Lenovo aus dem Privatbestand.

Das Gerät kann für seinen vergleichsbar niedrigen Preis mit einer dedizierten Grafikkarte und einem hochwertigen Bildschirm und guten Lautsprechern als Multimedia-Notebook überzeugen.



Abbildung 7: Lenovo IdeaPad G50

Eckdaten der Hardware-Spezifikationen:

CPU: AMD A6-6310 APU, 1.8Ghz
GPU: AMD Radeon R4 Graphics 4.6GB RAM
RAM: 8GB DDR3
HDD: 1x 256 GB HDD



2.1.7 Notebook DELL XPS 13 9360

Alexander Weinberger besitzt ein XPS 9360 13 Zoll Notebook von DELL aus dem Privatbestand.

Dells neues XPS 13 in der bereits vierten Generation mit dem Kaby-Lake-i5 und dem Full-HD-Bildschirm überzeugt mit extrem langen Akkulaufzeiten, toller Leistung und hervorragender Ergonomie.



Abbildung 8: DELL XPS 13 9360

Eckdaten der Hardware-Spezifikationen:

- CPU:** Intel Core i5-8250U, 4x 1.60GHz
- GPU:** Intel UHD Graphics 620 (IGP)
- RAM:** 8GB LPDDR3 (8GB verlötet, nicht erweiterbar)
- HDD:** 1x 256GB M.2 PCIe

2.1.8 Smartphone OnePlus 5T

Das Android Smartphone von OnePlus aus dem Privatbesitz von Josip Jerkovic wurde sowohl zum Testen der App als auch der Mobilansicht der Responsive Website genutzt.

Eckdaten der Hardware-Spezifikationen:

<i>Display:</i>	6.01", 2160x1080 Pixel, AMOLED
<i>CPU:</i>	Snapdragon 835
<i>RAM:</i>	8 GB
<i>Speicher:</i>	128 GB
<i>System:</i>	Android 9 (Pie)



Abbildung 9:
OnePlus 5T

2.1.9 Ipad Air 2

Um die Kompatibilität der Website auch auf einem Tablet und vor allem einem iOS Betriebssystem zu gewährleisten wurde zu Testzwecken aus dem Privatbesitz von Josip Jerkovic ein iPad Air 2 benutzt.

Eckdaten der Hardware-Spezifikationen:

<i>Display:</i>	9.7", 2048x1536 IPS
<i>CPU:</i>	3x 1.50GHz A8X (Typhoon)
<i>RAM:</i>	2 GB
<i>Speicher:</i>	128 GB
<i>System:</i>	iOS 13.4.2



Abbildung 10: Ipad Air 2



2.2 Verwendete Software

Die in Folge aufgelisteten Softwareanwendungen werden in den einzelnen Abschnitten der Teammitglieder näher beschrieben und die Verwendung und deren Nutzen näher erläutert.

2.2.1 Design

2.2.1.1 Adobe Photoshop

Adobe Photoshop ist wohl das bekannteste und am häufigsten benutzte Bildbearbeitungstool auf dem Markt.

Es ist Teil einer Sammlung von Grafik- und Designprogrammen von Adobe, welche unter dem Namen Adobe Creative Cloud vermarktet wird. Adobe Photoshop wurde verwendet, um die Aufgabenstellung zu visualisieren.



Abbildung 11:
Adobe
Photoshop
Logo, Quelle:
[adobe.com](https://www.adobe.com)

2.2.1.2 Adobe Illustrator

Illustrator von Adobe ist ein vektorbasiertes Grafik- und Zeichenprogramm zum Erstellen von Logos, Symbolen, Icons, und Ähnlichem. Adobe Illustrator wurde verwendet, um das Projektlogo *Simple Statistics* für die App und die Website zu erstellen.



Abbildung 12:
Adobe
Illustrator
Logo, Quelle:
[adobe.com](https://www.adobe.com)



2.2.1.3 Adobe Xd

Adobe Xd ist ein vektorbasiertes Design-Tool für das Erstellen von Designs von Webseiten, Apps und vielem mehr. Mit Adobe Xd kann man weder Webseiten an sich erstellen noch Apps programmieren, sondern sie ist lediglich dafür konzipiert, um Interaktionsdesigns zu erstellen und eine optimale User-Experience zu ermöglichen.



*Abbildung 13:
Adobe
Photoshop
Logo, Quelle:
adobe.com*

2.2.2 IDEs und Editoren

2.2.2.1 Visual Studio Code

Visual Studio Code ist ein Texteditor, dieser wird von Microsoft entwickelt und bietet von sich aus eine Integration mit Git und ist mit durch Community entwickelten Erweiterungen sehr individualisierbar und deckt eine Vielzahl an Programmiersprachen ab, unter anderen HTML, CSS, JavaScript und PHP.



*Abbildung 14:
Visual Studio
Code Logo,
Quelle:
visualstudio.com*

2.2.2.2 Thonny

Thonny ist eine Entwicklungsumgebung für Python, die alles bietet was man braucht, um in Python zu programmieren, ist frei verfügbar und zudem bei Raspbian vorinstalliert.



*Abbildung 15:
Thonny Logo,
Quelle: thonny.org*

2.2.3 Datenbanken

2.2.3.1 MySQL

MySQL ist ein relationales Datenbank Management System (DBMS), um Daten und Datenbanken zu verwalten. Es ist eines der weltweit verbreitetsten relationalen Datenbankverwaltungssysteme und ist sowohl als Open-Source-Software als auch als kommerzielle Enterprise Version verfügbar.



Abbildung 16: MySQL

Logo, Quelle:

wikipedia.org

2.2.3.2 phpMyAdmin

PhpMyAdmin ist eine Open-Source Webanwendung für die Administration von MySQL und MariaDB-Datenbanken.



Abbildung 17:

phpMyAdmin, Quelle:

wikipedia.org

2.2.3.3 XAMPP

XAMPP ist ein Zusammenschluss von mehreren Technologien und wird als ein lokaler Entwicklungsserver mit Datenbank verwendet.



Abbildung 18: XAMPP

Logo, Quelle:

apacheefriends.org

2.2.4 Sonstige Software

2.2.4.1 PuTTY

PuTTY ist eine Freeware zum Herstellen von unter anderem SSH-Verbindungen. Auf die Funktionsweise von PuTTY wird in Folge noch näher eingegangen.



*Abbildung 19: Putty Logo,
Quelle: wikipedia.org*

2.2.4.2 FileZilla

FileZilla ist eine Software, die durch Datentransferprotokolle das Verschieben von Dateien von einer lokalen Entwicklungsumgebung auf einen Live-Server ermöglicht.



*Abbildung 20:
Filezilla Logo, Quelle:
wikipedia.prg*

2.2.4.3 Git und GitHub

Git ist ein Versionskontrollsystem und wird verwendet, um den Entwicklungsverlauf eines Projektes in Versionen zu organisieren.



GitHub ist eine Online-Plattform, auf der es möglich ist an Open-Source Projekten mitzuarbeiten und kann auch als eine online Versionskontrolle für eigene Projekte genutzt werden. Mit GitHub Pages können diese sogar kostenlos mit Webhosting versorgt werden.



*Abbildung 21: Git,
Github Logos, Quelle:
wikipedia.org*

3 Hardware des Kundenfrequenzmessers

3.1 Einleitung

Es werden einige Ressourcen benötigt, welche die Gesichtserkennungssoftware unterstützen. Dieses Gerät wird mithilfe von Applikationen die Online bereitgestellt werden installiert und für das Projekt angepasst. Mit Hilfe einiger Online-Tools werden Programme wie OpenCV installiert, die für die Software des Projekts verwendet wird.

3.2 Aufgabenstellung Hardware

Um diese Idee realisieren zu können wird ein kompaktes Gerät benötigt, welches klein genug ist, um an verschiedensten Stellen in unterschiedlichen Positionen über der Türe des Geschäfts anmontiert werden zu können. Es soll außerdem auch eine gute Leistungsfähigkeit aufweisen, die die Kalkulationen der Kamera problemlos durchführen und zum Server herübersenden kann, um die Daten dann zu verarbeiten.

Ein simples Design in Verbindung mit einer kleinen Kamera soll kostengünstig und unkompliziert sein, um auch die Installation der Software an allerlei Geschäften schnell und effizient zu erledigen zu können.

Da die Gesichtserfassung mit etwas preisgünstigen Kameras einen gewissen Grad an Problematik entwickeln kann, wird die Vielseitigkeit des Geräts eine etwas größere Rolle zur Verarbeitung der Daten spielen. Es muss aber auch Acht auf die Komponenten gegeben werden, sodass diese nicht offenstehen und leicht in Sichtweite erwischt werden können, oder aber andere Gegenstände in der unmittelbaren Nähe beeinträchtigen.

Damit dieses Gerät transportabel und ausfallsicher bleiben kann, wird eine Batterie benötigt, die an einer Energiequelle verbunden werden kann, um die Gesichtserfassung auf Wunsch oder rund um die Uhr fortgesetzt in Betrieb zu halten. Daher soll es möglich sein diese gesamte Ausstattung als Erstbenutzer bei möglichen Ausfällen oder Systemfehlern leicht bedienen zu können, wie zum Beispiel mit Hilfe eines einzigen Knopfdrucks, um den gesamten Vorgang erneut in Bedienung zu setzen.

3.3 Verwendete Technologien

Im Rahmen des Projekts ersinnt sich der Raspberry Pi als die perfekte vielseitige Lösung zu den oben genannten Hindernissen. Ein Minicomputer von bescheidener Größe und einer gewissen wirksamen Leistungseffizienz erschien sich als selbstverständlich für diese Umstände. Mit den verschiedenen Modifikationsmöglichkeiten erweist sich diese Vielseitigkeit vom vorteilhaften Nutzen. Das von der Raspberry Pi Hauptseite bereitgestellte Operating System verhilft dem Projekt mehr Spielraum mit dem Umgang von diversen Programmiersprachen, die für die Gesichtserkennung verwendet werden.

3.3.1 Raspberry Pi 4

Der Raspberry Pi ist ein Dual Display Minicomputer, entworfen von Eben Upton und seinen Kollegen in der Informatik Industrie. Dieses Gerät wurde entwickelt für Einsteiger der Technikwelt aller Altersgruppen. Nun hauptsächlich in Verwendung für Schulen und als Meilenstein für das Erwecken das Interesse für zukünftige Entwickler von Software oder Hardware.

Das für das Projekt bereitgestellte RaspberryPi 4 Model B erschien am 28. Mai 2020. Auf der Platine befinden sich sowohl jeweils 2 USB2 und 3, 2 Micro HDMI Ports als auch einen USB-Type C und Gigabit Ethernet Anschluss.

Die Spezifikationen sind wie folgt:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE

Quelle: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>

Es wurde die 4GB RAM Version des Raspberry Modells verwendet.

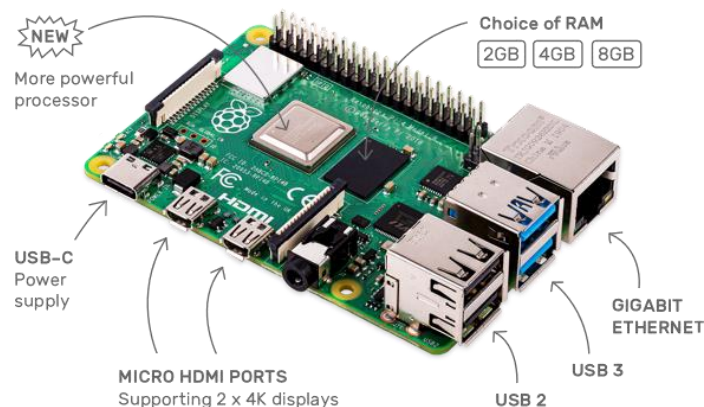


Abbildung 22: Raspberry Pi 4, Quelle: winfuture.de

3.3.2 OKdo Power-Hat

Diese Case eignet sich hervorragend für die erleichterte Bedienung des Raspberrys. Sie kapselt die Platine ein, schützt somit die Komponenten und ist gleichermaßen ästhetisch angenehm. Außerdem besitzt es Platz für

einen Lüfter, um die Platine kühl zu halten, da sie für längere Zeit Gesichter kalkulieren muss.

Je nach Knopfdruck Dauer kann mithilfe des Schalters den Raspberry entweder neustartet oder komplett herunterfahren werden. Das LED leuchtet grün, wenn es an ist und Rot im Standby oder abgeschaltetem Modus.

Im Rahmen des Projekts wurde an der Case eine kleine Modifikation durchgeführt, um die Raspberry Pi Camera V2 durch die Schnittstellen führen zu können.



Abbildung 23: OKdo Raspberry Pi 4 Power-Hat Case, Quelle: okdo.com

3.3.3 Raspberry Pi Camera V2

Diese Kamera ist vom Preis/Leistungsverhältnis ausreichend für das Testen und Ausführen von Gesichtserkennungssoftware. Sie besitzt eine 8 Megapixel Auflösung und kann Bilder von bis zu 3280 px by 2464 px Pixeln aufnehmen. Videos werden bei einer Auflösung von 1080p auf 30fps; 720p auf 60 fps und 640 x 480p auf 90 fps unterstützt. Die Linse selbst ist eine „Sony IMX219 CCD“.



Abbildung 24: Raspberry Pi Camera Module V2, Quelle: raspberrypi.org

3.3.4 Ansmann Powerbank 10.8

Um unter Umständen einen Datenverlust durch Stromausfall zu verhindern, wurde eine Powerbank in Verwendung gebracht. Dieses Gerät hat eine Kapazität von 10.000mAh und versorgt den Raspberry mit 2.1A Strom.

Es besitzt 2 Ausgangs-USB Anschlüsse und einen Eingangs Micro-USB Anschluss. Ein USB-Port hat eine 1A Stromversorgung, der gegenüberliegende Port hat einen 2A Stromdurchfluss.



Abbildung 25: Ansmann Powerbank 10.8, Wuelle: amazon.de

Um eine Remote Control Verbindung aufzubauen werden einige Programme benötigt, die aus dem Internet heruntergeladen wurden. Die verwendeten Programme in Frage lauten:

3.3.4.1 VNC Viewer

VNC = Virtual Network Computing.

Mithilfe dieses Programms kann der „Kopf“ des Raspberrys simuliert werden ohne Gebrauch von HDMI Kabel + Monitor (+ Tastatur/Maus). Dazu muss nur die IP des Raspberrys eingegeben werden. Eine Verbindung mit VNC wird über SSH bereitgestellt und ein Fenster geöffnet, welche den Desktop des Raspberrys anzeigen wird.

SSH oder „Secure Shell“ ist ein kryptografisches Protokoll, welcher benutzt wird, um eine sichere Verbindung zwischen Medien in einem ungesicherten Netzwerk bereitstellen zu können.

3.3.4.2 Softperfect Network Scanner

Dieses Programm wird benötigt, um die IP Adresse des Raspberrys herauszufinden, falls der Zugang zum Ethernet Port des Routers nicht gewährt werden kann.

Um sich auf den Raspberry allerdings verbinden zu können werden zuallererst die zugestellte IP des eigenen DHCP Servers und einige angepasste Einstellungen benötigt. Diese Einstellungen werden im Kapitel „3.4.2.1 Freigabe“ näher erläutert.

3.3.4.3 PuTTY

PuTTY ist ein kostenloser SSH/Telnet Client, der eine sichere Verbindung zum Terminal des Raspberry OS ermöglicht. Es werden Name und Passwort benötigt, um sich mit dem Client zum Raspberry zu verbinden. Die Out-of-

the-box Credentials sollten im Raspberry gelten. Deshalb ist es empfehlenswert diese Anmeldedaten im Terminal später zu ändern, um mögliche Brute-force Angriffe zu vermeiden.

Diese Programme werden jedoch nicht für die Funktionen des Gesichtserkennungsprogramms oder Server Verbindungsaufbau benötigt, diese dienen lediglich für die erleichterte Bedienung und Installation des Rasperrys und dessen Konfiguration.

3.4 Umsetzung

3.4.1 Bau



Abbildung 26: Hardware Komponenten

In der obigen Grafik werden alle nötigen Teile für den Bau angezeigt. Von Oben nach Unten, links nach rechts: Schalter und Boden des PoweHATs, die HAT Platine, Raspberry Pi 4, Raspberry Camera V2.



Abbildung 27: Raspberry Pi 4 CSI-Schnittstelle

Als Erstes wurde die schwarze Lasche - wie in der Grafik rot gekennzeichnet – leicht gehoben und der Kabel der Kamera in den Schlitz hineingeschoben.



Abbildung 28: Angeschlossenes Pi Kamera Modul

Anschließend wird die Lasche niedergedrückt, damit die Kamera festsetzt. Dabei wurde darauf geachtet, dass sich keine allzu großen Falten im Kabel

bilden. Wenn dies der Fall wäre, würde das bedeuten, dass die Kamera einen Fehler beim Test ausgibt. Da solche Probleme nicht behoben werden können, müsste die Kamera ersetzt werden. Knicke müssen daher, wenn möglich, vermieden werden.

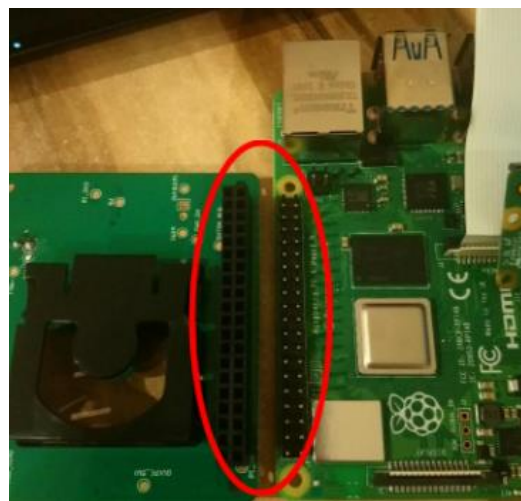


Abbildung 29: Raspberry Pi 4 GPIO Pins

Im nächsten Schritt wird die HAT Platine an die GPIO Pins angesteckt. Die Platine benützt alle Pins für die Funktion des Powerschalters und dient zu dem auch gleichzeitig als Adapter. Das bedeutet, dass man die Pins auf der HAT Platine verwenden soll, falls bei Belieben ein paar weitere Module angesteckt werden sollen.



Abbildung 30: HAT-Platine

Es wird nebenbei auch darauf geachtet, dass die Kamera wie in obiger Grafik gezeigt um die Platine geführt wird und nicht darunter sitzt. Die freien GPIO Pins werden auf der linken Seite in der Grafik der Platine angezeigt. In diesem Projekt werden diese aber nicht verwendet.



Abbildung 31: Zusammengebaute Komponenten

Anschließend wird die zusammengefügte Platine in die PowerHAT Case angesteckt. Dazu muss diese einfach an die Schnittstellen mit den Ports

angepasst und der Schalter draufgesteckt werden. Es gibt außerdem genügend Platz, um die Kamera noch durch eine der Ports zu legen, sodass diese funktionsfähig bleibt und sicher festsitzt.



Abbildung 32: Zusammengebaute Komponenten mit Case

Sowie in dieser Grafik erkannt wird, wurde die Kamera auf einen der Ports darübergerlegt. Es wurde auch spekuliert eine extra Schnittstelle an die Case für die Kamera zu modifizieren, jedoch wurde stattdessen die erstere Methode für das Projekt entschieden.

Dies ist das Endprodukt des Projekts und mit mehreren Tests erfolgreich als „funktionstüchtig“ beurteilt.

Als nächstes wird die Installation des Raspberrys am Laptop oder PC für die Software vorbereitet.

3.4.2 Installation

Nach dem Zusammenbau der Teile ist der Raspberry Pi nun einsatzbereit für die Installation. Als nächstes werden ein Ethernetkabel und ein Stromanschluss mit USB type C benötigt, um den Raspberry anzuschalten. Empfehlenswert ist der Gebrauch eines Laptops oder Desktop Computer Monitor mit HDMI Ports. Nur für alle Fälle, falls die Remote Control Verbindung aus irgendwelchen Gründen, wie zum Beispiel Zustellung der IP-Adresse vom DHCP Server nicht funktionieren sollte.

Nachdem die Materialien bereitgestellt wurden, wird alles zusammengesteckt und vorbereitet. Es wurde ein Laptop gemeinsam mit einem Desktop PC für das Projekt verwendet.

3.4.2.1 Freigabe

Nun werden die Einstellungen für den Raspberry Pi unter Windows 10 angepasst. Damit die Ethernet Verbindung gelingen kann muss unter den Netzwerkeinstellungen die Verbindung freigegeben werden. Dazu wird zuerst in den Netzwerk- und Freigabecenter navigiert.

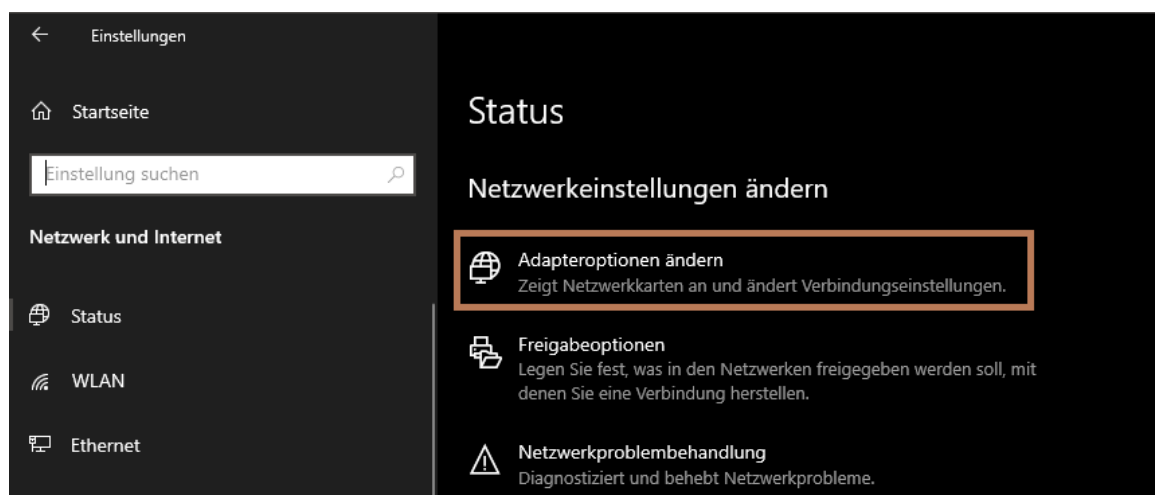


Abbildung 33: Netzwerkeinstellungen ändern

Unter den Adapteroptionen werden die verwendeten Netzwerkverbindungen angezeigt. In diesem Fall wird nur das WLAN benötigt

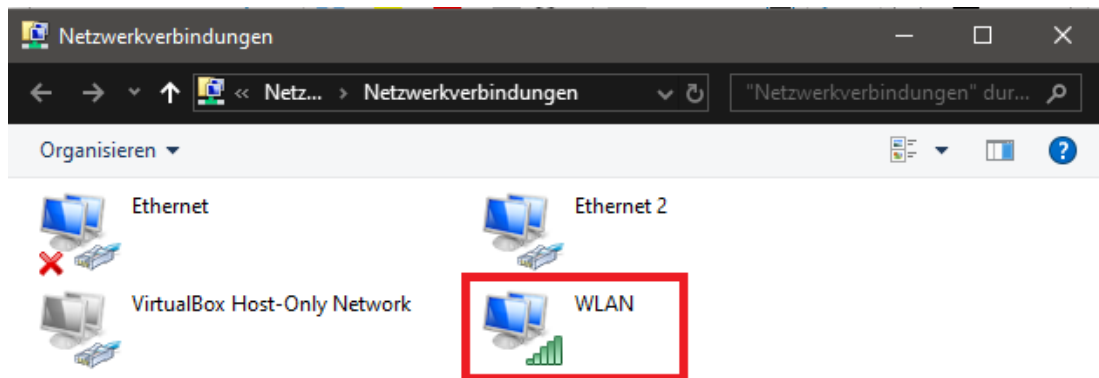


Abbildung 34: Netzwerkverbindungen

Mit einem Rechtsklick auf der WLAN-Verbindung, wird in den Eigenschaften und anschließend auf den Tab „Freigabe“ navigiert.

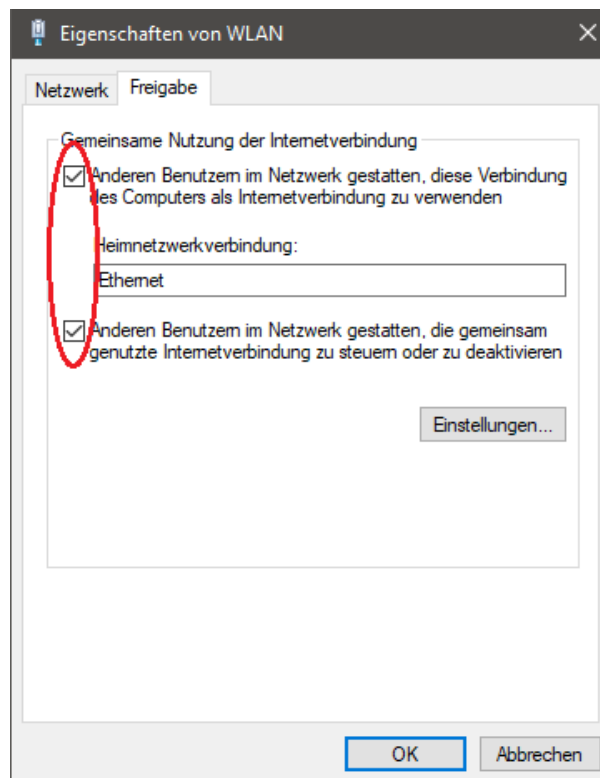


Abbildung 35: Eigenschaften von WLAN

Damit die Verbindung zum Router für ein angeschlossenes Gerät am Laptop freigegeben werden kann, müssen beide Häkchen angekreuzt werden und die Heimnetzwerkverbindung: „Ethernet“ anzeigen.

Dadurch erlaubt der Laptop den Raspberry eine Verbindung mit dem Router zu erlangen, um vom DHCP Server seine eigene IP bekommen zu können. Diese IP wird dann hauptsächlich für die Remoteverbindung im verbundenen WLAN verwendet.

3.4.2.2 Betriebssystem

Nun wird eine Mini SD-Card benötigt, die als Harddrive für den Raspberry fungiert. Auf dieser SD-Card wird die Raspbian BusterOS Image heruntergeladen, welche auf den Debian OS Buster basiert. Damit das Image auf die Card übertragen werden kann, wird ein SD-Card Adapter benötigt, welcher in den Laptop oder Computer eingesteckt werden kann, um Zugriff darauf bekommen zu können. Es muss vergewissert werden, dass die SD-Card nicht für andere Dinge verwendet wird oder persönliche Daten noch drinnen vorhanden sind. Die Card wird nämlich hauptsächlich für die Verwendung des Raspberrys reserviert, daher wird die gesamte Kapazität mithilfe eines Tools formatiert.

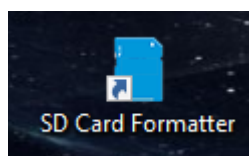


Abbildung 36: SD-Card Formatter Verknüpfung

Mithilfe des „SD-Card Formatter“ Tools, wird der gesamte Inhalt der Card sicher entfernt. Das verschafft dann den nötigen Platz für das Raspbian OS.

Als Nächstes wird das Image heruntergeladen. Diese Datei ist eine .iso extension welche mit einem weiteren Tool auf die SD-Card entpackt wird.



Abbildung 37: Debian OS Download Seite, Quelle: raspberrypi.org

Das nächste Tool ist der „balenaEtcher“.

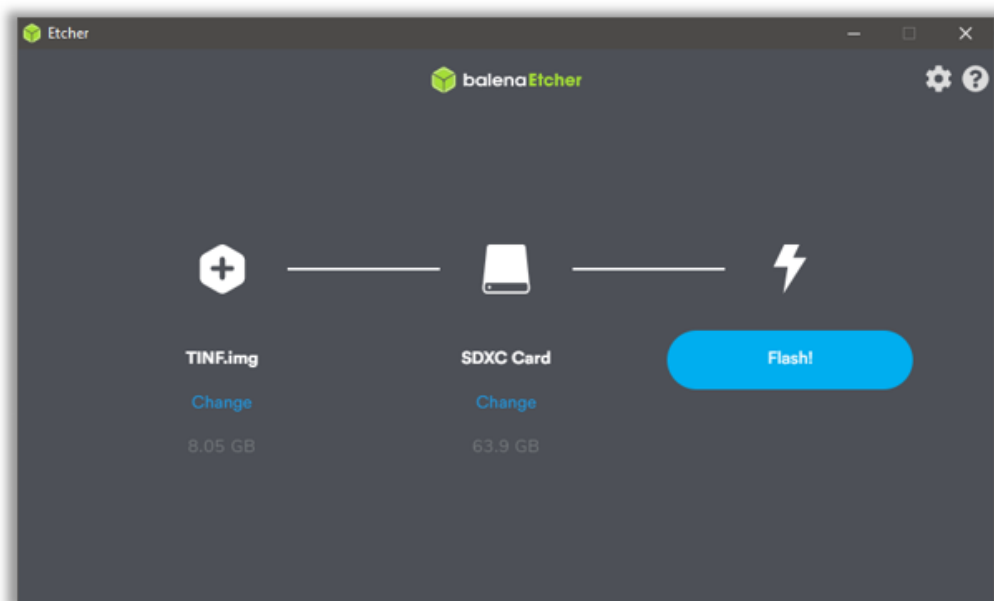


Abbildung 38: balenaEtcher - Image to SD-Karte



Dieses Programm „flasht“ die .iso Datei in den ausgewählten Datenträger. Es soll festgestellt werden, dass der Datenträger vorerst genügend Kapazität aufweist. Mindestens 8GB soll für den Raspberry ausreichen. Die für das Projekt verwendeten SD-Cards verfügen über 64GB Speicherplatz.

Nachdem die iso Datei übertragen wurde, kann die SD-Card nun vom PC oder Laptop entfernt und die Mini SD-Card in den Raspberry eingeführt werden. Vorerst soll der Raspberry aber heruntergefahren und abgesteckt werden bevor die SD-Card eingesteckt wird.

Danach wird das Gerät erneut angesteckt. Sobald der Raspberry hochgefahren ist, sollte im Hintergrund der Hochfahr-Prozess starten. Währenddessen kann mithilfe des VNC Viewers - oder einem angeschlossenen Monitor - eine Verbindung mit dem Raspberry aufgebaut werden. Es ist auch möglich sich direkt am Terminal mithilfe des SSH Tools zu verbinden, um die Installationen durchzuführen.

Im Projekt wurde eine VNC remote Verbindung verwendet und ein Terminal Fenster am Raspberry Desktop geöffnet.

Da der Raspberry Pi 4 Modul B eine WiFi Funktion hat, wird diese hauptsächlich verwendet, um sich ins Internet zu verbinden. Da aber das Passwort der Verbindung zum beliebigen Router im Raspberry Interface ungeschützt ist, wird in einer WPA configfile mithilfe einiger Kommandos das Passwort verschlüsselt und gespeichert, sodass sie nicht erneut eingegeben werden muss.

Dazu das Kommando:

```
-> sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

In der GUI:

```
network{  
    ssid:"ssid"  
    *psk:"psk"  
}
```

*Die PSK muss verschlüsselt werden

Unter „SSID“ wird der Name des WLANs eingegeben, unter „PSK“ das Passwort.

Bevor in der nano GUI jedoch das WLAN-Passwort eingegeben wird, muss sie mithilfe eines Kommandos verschlüsselt werden.

-> wpa_passphrase ssid psk

Dadurch wird ein neuer PSK Hashwert erstellt.

Der wird kopiert und in die wpa_supplicant.conf eingefügt.

Zuletzt wird der Raspberry rebooted.

Nun kann der Raspberry sich automatisch ins gespeicherte Netzwerk verbinden nach jedem reboot.

3.4.2.3 Python / Numpy

Im Folgenden werden die Commandlines für die Installation vom Python virtual environment und NumPy gezeigt, die im Terminal eingegeben wurden.

-> wget <https://bootstrap.pypa.io/get-pip.py>



```
-> sudo python get-pip.py
-> sudo python3 get-pip.py
-> sudo rm -rf ~/.cache/pip
-> sudo pip install virtualenv virtualenvwrapper
```

Nun öffnet man die bashrc file

```
-> nano ~/.bashrc
```

In diesem File werden ein paar Zeilen drunter angehängt:

```
-> export WORKON_HOME=$HOME/.virtualenvs
-> export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
-> source /usr/local/bin/virtualenvwrapper.sh
-> source ~/.bashrc
```

Zuletzt wird das virtual environment erstellt. Damit wird Python 3 und OpenCV 4 benutzt.

```
-> mkvirtualenv cv -p python3
```

Da das Projekt ein Kameramodul verwendet muss die API installiert werden.

```
-> pip install „picamera[array]“
```

3.4.2.4 Libraries

Nun werden einige Commandlines verwendet, um bestimmte Libraries herunter zu laden die die Funktionen für OpenCV unterstützen.

Aufgrund der Erstverwendung des Raspberrys werden einige Pakete als erstes geupdatet

```
-> sudo apt-get update
```

-> sudo apt-get upgrade

Danach werden I/O Pakete installiert, diese unterstützen die Bildformat Dateien.

-> sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev

Anschließend auch die Videoformat Dateien.

-> sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

-> sudo apt-get install libxvidcore-dev libx264-dev

Zuletzt werden für die GUI's eine GTK dev library installiert.

-> sudo apt-get install libfontconfig1-dev libcairo2-dev

-> sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev

-> sudo apt-get install libgtk2.0-dev libgtk-3-dev

3.4.2.5 Pip-Install

Für das Projekt und die OpenCV Installation wurde die „Pip-Install“ Methode verwendet.

-> pip install opencv-contrib-python

Damit auch alles funktioniert, wird die OpenCV 4 im virtual environment getestet.

-> cd ~

-> workon cv

-> python

>>> import cv2

```
>>> cv2.__version__
```

```
'4.1.1'
```

Die ausgegebene Version zeigt eine erfolgreiche Installation an.

Hiermit sind das Gerät und das System voll funktionstüchtig. Es kann nun eine Verbindung mit dem Server aufgebaut werden und die nötigen Programme können auf den Raspberry draufgeladen werden.

3.4.2.6 Verworfenne Ideen

Dies ist die zweite Version der Hardware.

Die erste Version wurde zusammengebaut mit einer Battery-HAT statt einer Case mit Schalter.



Abbildung 39: Strompi V3 Battery-HAT

Dies ist die zweite Version der Hardware.

Die erste Version wurde zusammengebaut mit einer Battery-HAT statt einer Case mit Schalter.



Abbildung 40: Angeschlossener Battery-HAT

Die verwendete Battery heißt: Strompi V3 Battery-HAT.

Die Idee war, dass der Raspberry unter Stromausfall auf eine Energiereserve laufen soll, die lang genug durchhält, bis der Strom wieder in den Raspberry zugeführt werden kann.

Diese Batterie hat 2 Platinen, eine mit eingebauten Funktionen und eine mit der eigentlichen Batterie, beide auf dem Vorgänger „Raspberry Pi 3“ angebracht. Nach dem Zusammenbau werden im Raspberry Terminal einige Kommandos eingetippt, welche die Batterie zum Laufen bringen sollen. Diese Kommandos starten die Hauptfunktionen der Batterie unter einem eigenen Terminal. Darunter können weitere Optionen eingestellt werden, welche hervorragend für das Projekt gedient hätten. Ein eingebauter Timer der den Raspberry automatisch an oder ausschalten soll und wenn die Haupt-Energiezufuhr aus bestimmten Gründen abgeschnitten wird, soll die Batterie wissen, wann sie automatisch weiterlaufen soll oder nicht.



Weiters hat die Batterie eine Date&Time Funktion. Diese erlaubt es der Batterie den Raspberry an bestimmten Werktagen an einem bestimmten Zeitpunkt an oder auszuschalten. Mitunter kann zum Beispiel eingestellt werden, dass der Raspberry nur montags bis freitags zwischen 8 Uhr morgens und 9 Uhr abends laufen soll.

Eine sehr hilfreiche Lösung für geschäftliche Software und Datenverarbeitung.

Jedoch ergaben sich neue Probleme mit dieser Lösung.

Wegen der addierten Größe des Geräts, passt sie in keine sichere Case, sodass diese die Batterie vor möglichen Schäden schützt. Darüber hinaus ergab sich durch einige Tests, dass der Raspberry nicht genügend Leistung für die nötigen Kalkulationen aufwies. Daher musste die Hardware komplett neu ausgedacht werden. Auch aufgrund der Veröffentlichung des neuen Raspberry Pi 4's die auch eine signifikant höhere Leistung hat, konnte die Lösung mit der Batterie nicht weitergeführt werden. Die verbesserte Leistung des Raspberry Pi 4 dient der Leistung der Gesichtserkennung ebenfalls vom Vorteil.

Die funktionstüchtige Lösung wird daher nicht mehr automatisch, sondern manuell mithilfe von Knopfdruck bedient.

Damit wurde entschieden stattdessen eine Powerbank als Batterie zu benutzen, welche den Raspberry auch viel länger mit Strom unterstützen kann.

4 Datenbanken des Kundenfrequenzmessers

Datenbanken kommen in der Softwareentwicklung zum Einsatz, wenn eine Vielzahl an Informationen verwaltet und gespeichert werden soll, wie zum Beispiel Benutzerinformationen, oder Produktkataloge von Onlinehändlern.

4.1 Verwendete Technologien

4.1.1 MySQL und SQL

MySQL ist ein Datenbankverwaltungssystem (DBMS), Daten werden in Tabellen, die in einer Relation zueinanderstehen gespeichert. SQL, die Datenbanksprache, wird eingesetzt, um mit diesen Daten arbeiten zu können und kann in drei Aufgabenbereiche geteilt werden. Das Erstellen der Datenbankstruktur (Data Definition Language), das Manipulieren der Daten zum Beispiel auslesen dieser (Data Manipulation Language) und das Kontrollieren der Zugriffe auf die Datenbank selbst durch Rechteverwaltung (Data Control Language).

4.2 Aufgabenstellung Datenbanken

Die Datenbank dient als zentrale Komponente für alle weiteren Teile des Projektes, da der Raspberry Pi die gesammelten Daten der Geschäfte an diese sendet und die Webseite auf diese zugreifen soll, um die Kundenfrequenz auszulesen. Sie soll Informationen über die Kunden, wann diese sich anmelden, deren Geschäfte und die Kundenzahl dieser speichern.

4.3 Datenbankstruktur

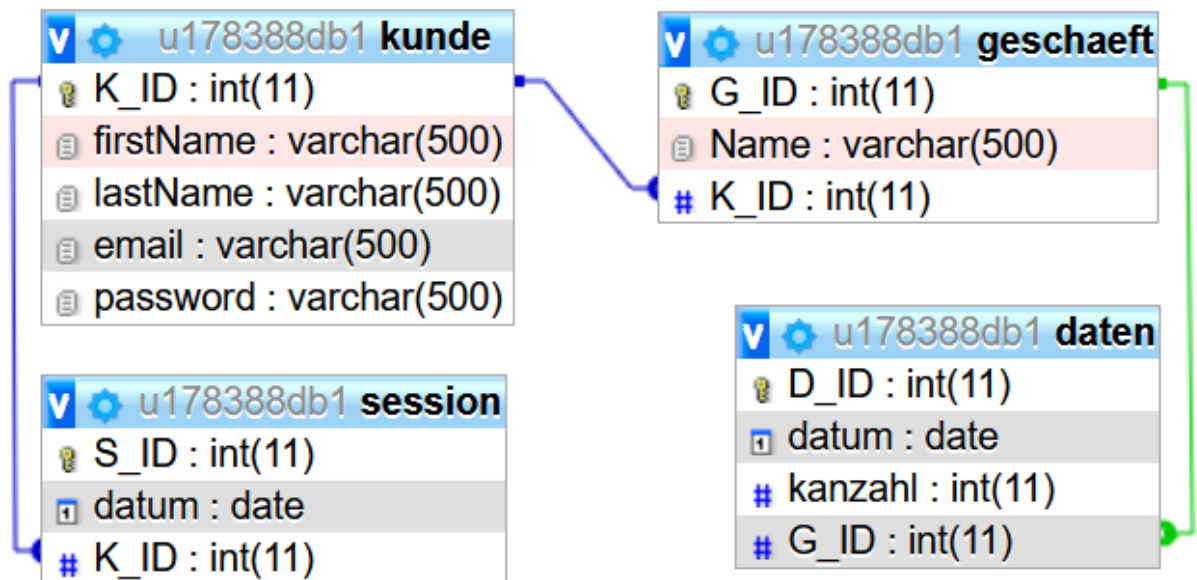


Abbildung 41: Datenbankstruktur Kundenfrequenzmesser

Für die Speicherung aller benötigten Informationen wird eine MySQL Datenbank verwendet, da das Team durch den Schulunterricht mit der Funktionsweise bereits vertraut war und die eingesetzte Serverstruktur dieses bereits unterstützte. Als eine Alternative zu MySQL kann eine Dokumenten basierende Datenbank wie NoSQL (not only SQL) mit MongoDB als einen bekannten Vertreter betrachtet werden. In NoSQL werden Daten in Dokumenten dargestellt diese werden in JSON-Syntax verfasst. Die Struktur der MySQL Datenbank besteht aus vier Tabellen, der Kunden-Tabelle, in welcher die personenbezogenen Informationen gespeichert werden wie der Name des Kunden und dessen Login Informationen. Sowie eine Geschäfts-Tabelle die Geschäfte und deren Namen abbilden soll. In einer Daten-Tabelle, in welcher die Kundenfrequenz, als *kanzahl* jeglicher Geschäfte mit dem aktuellen Datum gespeichert wird. In der Session-Tabelle werden beim Anmeldevorgang die Kunden ID und das Datum des aktuellen Tages abgespeichert.

4.3.1 Relationen

Tabellen in der Datenbank stehen in einer Relation zueinander, welche unterschiedliche Kardinalitäten aufweisen können. Die Kundentabelle ist über die `K_ID` mit den Tabellen Session und Geschäfte verbunden, da sie eine 1: n Beziehung aufweist. Der Kunde kann sich beliebig oft anmelden und ein, oder mehrere Geschäfte haben, jedoch können beide nur zu einem Kunden gehören. Die Geschäfte-Tabelle und Daten-Tabellen sind über das Feld `G_ID` verbunden und bilden wieder eine 1:n Beziehung. Ein Geschäft kann mehrere Datensätze in der Tabelle Daten besitzen, diese können jedoch immer nur zu einem Geschäft gehören.

4.3.2 Primary und Foreign Keys

Solche Relationen, oder Beziehungen sind über einen Schlüssel, der jeden Datensatz in einer Tabelle eindeutig macht, Primary Key (PK) genannt mit anderen Tabellen verbunden. In diesen Tabellen wird der PK zu einen Foreign Key (FK) und drückt so die Zugehörigkeit aus. Gemeinsam bilden PK und FK die Relation und können Beschränkungen aufweisen.

4.3.3 Constraints

Constraints, die Beschränkungen in einer Datenbank beschreiben, werden gesetzt, um die Integrität, den Zustand der Daten in der Datenbank sicherzustellen. Sie beschreiben wie sich Datensätze verhalten, wenn Änderungen in den Tabellen stattfinden, mit denen sie in einer Beziehung zueinanderstehen.



4.3.4 Stored Procedures

Stored Procedure sind vorbereitete SQL-Anweisungen, die erstellt und in der Datenbank abgespeichert werden, damit diese immer wiederholt aufgerufen und verwendet werden können. Ein Stored Procedure akzeptiert auch einen oder mehrere Eingabeparameter, sodass einzelne Stored Procedures von mehreren Clients mit unterschiedlichen Eingabedaten verwendet werden können. Solche Stored Procedures reduzieren den Netzwerkverkehr und erhöhen dessen Leistung. Wenn man ein Stored Procedure ändert, erhalten alle Clients das aktualisierte, gespeicherte Stored Procedure.

5 Software des Kundenfrequenzmessers

5.1 Aufgabenstellung Software

Es soll ein Programm geschrieben werden, welches auf einem Raspberry Pi 4 läuft. Dieses soll beim Starten des Kleincomputers automatisch ausgeführt werden, auf die angeschlossene Kamera zugreifen und den Stream dieser Kamera konstant analysieren. Sobald eine Person bzw. ein Gesicht in einem vordefinierten Bereich von dem Programm erkannt wird, soll das Programm beginnen dieses Gesicht zu verfolgen. Falls das verfolgte Gesicht Schwelle überschreitet, soll ein Zähler erhöht werden und die Person gilt als gezählt.

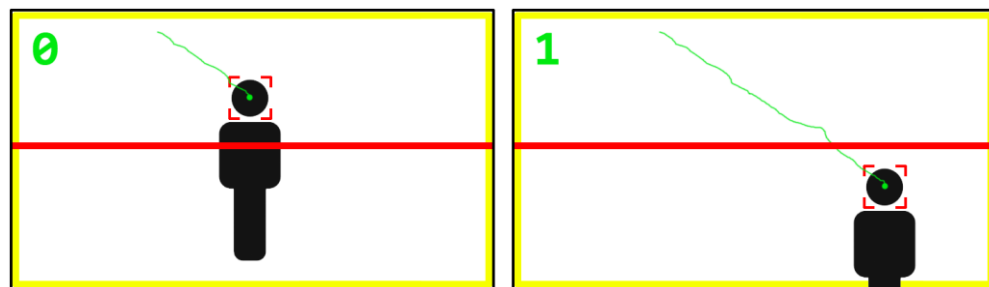


Abbildung 42: Beispielhafte Kundenzählung

Es soll auf jeden Fall sowohl der Begrenzungsrahmen, also derjenige Bereich in welchem das Programm nach Gesichtern Ausschau hält und in welchem das erkannte Gesicht verfolgt wird, als auch die Schwelle bei der die Person, nach dem Überschreiten dieser, gezählt wird, individuell einstellbar sein.

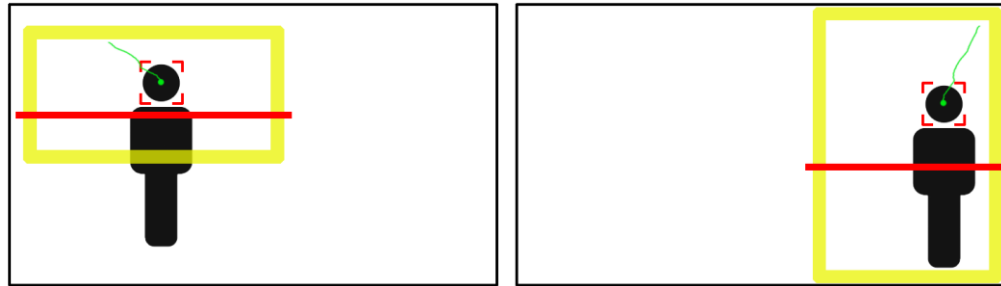


Abbildung 43: Beispielhafte Anpassungsfähigkeit des Begrenzungsrahmens

5.2 Verwendete Technologien

5.2.1 Python

Python ist eine interpretierte, objektorientierte höhere Programmiersprache. Sie wurde ursprünglich entwickelt, um das Programmieren beizubringen, weswegen die Syntax vergleichsweise so einfach zu verstehen und zu erlernen ist. Wegen der Plattformunabhängigkeit, der klaren Syntax und der guten Lesbarkeit hat Python, vor allem in den letzten Jahren, sehr an Popularität gewonnen, insbesondere bei Open-Source Projekten.



*Abbildung 44: Python
Logo, Quelle:
[wikipedia.org](https://www.wikipedia.org)*

In Python geschriebener Code ist sehr portabel was bedeutet er lässt sich in verschiedensten Betriebssystemen, wie zum Beispiel auf UNIX-basierten Systemen, Linux, MacOS, sowie verschiedensten Varianten von Microsofts Windows interpretieren. Zudem gibt es sehr viele Interfaces oder Schnittstellen für etliche Bibliotheken wie beispielsweise OpenCV.

Aus den genannten Gründen einerseits, und andererseits dem persönlichen Interesse heraus, diese Sprache zu erlernen hat man sich für Python als Programmiersprache bei diesem Projekt entschieden.

5.2.2 OpenCV

Die freie und plattformunabhängige Open-Source Programmierbibliothek OpenCV ist die am häufigsten verwendete Bibliothek zur Analyse und Verarbeitung von Bild- und Videomaterial mit einem Schwerpunkt auf Echtzeitverarbeitung.

Auch wenn OpenCV auf C und mittlerweile auf C++ basiert gibt es, zusätzlich zu den genannten Sprachen, Schnittstellen auch für Python, Java und MATLAB sowie Wrapper für C#, Ch, Perl, Haskell und Ruby. OpenCV gibt es für die Betriebssysteme von Linux, Windows, macOS, Android, iOS sowie für BlackBerry und diverse BSD-Varianten.

Zu den zahlreichen, um genauer zu sein über 2500, Funktionen und Modulen von OpenCV gehören unter anderem die Gesichtserkennung oder die Identifizierung und das Verfolgen von Objekten was sich für unser Projekt sehr anbietet.

Trotz des enormen Umfangs dieser Bibliothek ist der Umgang mit ihr vergleichsweise einfach. Das Einlesen von Informationen bzw. Bildern über zum Beispiel eine Webcam lässt sich mit nur wenigen Zeilen Python-Code realisieren. Dies ist unter Anderem einer der Gründe, weswegen man sich für die Verwendung dieser Bibliothek entschieden hat.

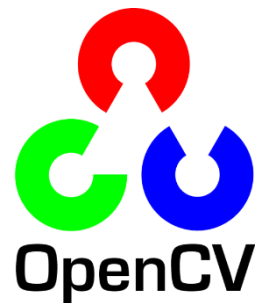


Abbildung 45:
OpenCV Logo,
Quelle:
wikipedia.com

5.2.3 Virtualenv

Ein Virtual Environment, zu Deutsch virtuelle Umgebung, ist ein Tool, mit dem die für verschiedene Projekte erforderlichen Abhängigkeiten getrennt werden können, indem für sie eine vollkommen isolierte virtuelle Python-Umgebungen erstellt wird. So sind die Abhängigkeiten jedes Projekts vom System und voneinander isoliert. Dies ist eines der wichtigsten Tools, welches die meisten Python-Entwickler verwenden.

Für dieses Projekt wird ein Modul namens virtualenv verwendet, mit welchem isolierte Python-Umgebungen erstellt werden können. Virtualenv erstellt einen Ordner, der alle erforderlichen und ausführbaren Dateien enthält, um diejenigen Pakete zu verwenden, die ein Python-Projekt benötigt.

5.2.4 PHP

Die ursprünglich 1994 entwickelte serverseitige Skriptsprache PHP (Hypertext Preprocessorist, ursprünglich Personal Home Page Tools) wird vor allem für die Webentwicklung genutzt und ist an die Syntax von C und Perl angelehnt. Die Skriptsprache zeichnet sich insbesondere durch eine breite Datenbankunterstützung aus. Eine einfache Integration von beispielsweise MySQL ermöglicht es relativ einfach Dateneinträge aus einer Datenbank abzurufen oder einzutragen und bietet sich dadurch für unser Projekt hervorragend an.



*Abbildung 46: PHP Logo,
Quelle: [wikipedia.org](https://www.wikipedia.org)*



5.2.5 Cron-Daemon

Der Cron-Daemon ist das Linux Pendant zum Task-Manager von Windows. Der Cron-Daemon läuft automatisch im Hintergrund und dient zur zeitbasierten Ausführung von Prozessen. Das bedeutet, dass zu angegebenen Zeiten bestimmte Prozesse ausgeführt werden können, was vorteilhaft ist, wenn man automatisierbare Aufgaben erstellen möchte. Diese Aufgaben werden Cronjobs genannt.

5.2.6 Haar Cascade Classifier

Haar Cascade ist ein Algorithmus zur maschinellen Lernobjekterkennung, welcher zum Identifizieren von Objekten in einem Bild oder Video verwendet wird und auf dem Konzept von Merkmalen basiert. Diese Merkmale unterscheiden sich in Abhängigkeit vom Objekt, welches zu erlernen ist.

Es ist ein auf maschinellem Lernen basierender Ansatz, bei dem eine Kaskadenfunktion aus vielen positiven und negativen Bildern trainiert wird. Weiters wird es verwendet, um Objekte in Bildern oder Videos zu erkennen, wie in unserem Falle Gesichter.

Die freie Python Bibliothek OpenCV bringt zwar bereits einige vortrainierte Haar Cascade Classifier mit, unter anderem um Gesichter von Menschen oder Katzen zu erkennen. Jedoch lassen sich auf GitHub viele verschiedene und konkretere, vortrainierte Haar Cascade Classifier finden, wie zum Beispiel welche für Autokennzeichen, den Ober- oder Unterkörper oder nur die Augen. In diesem Projekt hat man sich für den `haarcascade_frontalface_default.xml` Haar Cascade Classifier entschieden.

5.3 Umsetzung

Um die Software dieses Projekts zu realisieren war es zunächst erforderlich die Programmiersprache Python von Grund auf zu erlernen. Da während der schulischen Laufbahn mehrere Programmiersprachen unterrichtet wurden, war schon eine gewisse Basis und allgemeines Verständnis für Programmiersprachen vorhanden auf der aufgebaut werden konnte. Unter Zuhilfenahme von sowohl kostenlosen YouTube Tutorials und E-Learning-Webseiten als auch kostenpflichtigen Online-Kursen konnte zunächst ein Grundwissen in Python aufgebaut werden, um mit der Umsetzung zu beginnen.

5.3.1 Verzeichnisstruktur

Um eine grobe Übersicht zu schaffen wird hier ein wenig vorgreifen und ein paar Worte zur Verzeichnisstruktur verloren.

Alle benötigten Dateien, auf die in weiterer Folge noch genauer eingegangen wird, müssen sich auf dem Raspberry Pi 4 im selben Verzeichnis befinden. In unserem Fall ist es das Verzeichnis `/home/pi/facetracking`. Dieses beinhaltet unter anderem das Python Programm, welches für die Gesichtserkennung zuständig ist. Hierbei ist zu beachten, dass sowohl die Textdatei `FileWithCounter.txt`, in welcher die bereits gezählten Personen zwischengespeichert werden, als auch die Haar Cascade Classifier Datei `haarcascade_frontalface_default.xml`, sich im selben Verzeichnis befinden. Ebenfalls im selbigen Verzeichnis sind der Backup Ordner sowie weitere Python Programme, die dafür sorgen, dass der Kundenfrequenzmesser automatisiert wird und gegebenenfalls bei einem

Ausfall des Geräts die benötigten Programme wieder automatisch ausgeführt werden.

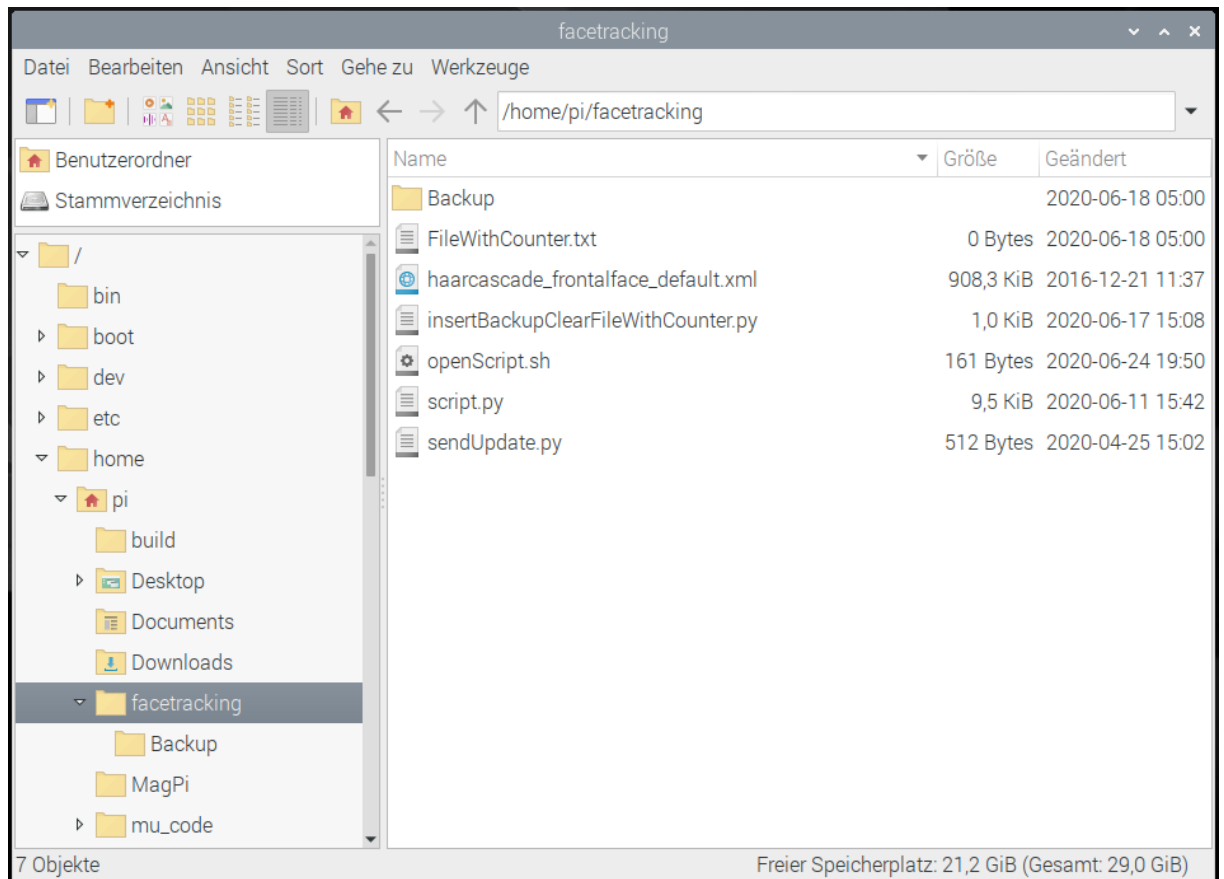


Abbildung 47: Verzeichnisstruktur Raspberry Pi

Auf die obig erwähnten Dateien und Ordner als auch deren Funktionen wird in den folgenden Kapiteln genauer eingegangen.

5.3.2 Cronjobs

Die Cronjobs werden unter Linux-Systemen in einer Crontab-Tabelle namens *crontab* verwaltet. Die systemweite Datei befindet sich unter `/etc/crontab` und kann nur mit Root-Rechten bearbeitet werden. Um die Cronjobs zu verwalten bzw. die Crontab-Datei zu editieren führt man im Terminal-Fenster den Shell-Befehl `crontab -e` aus.

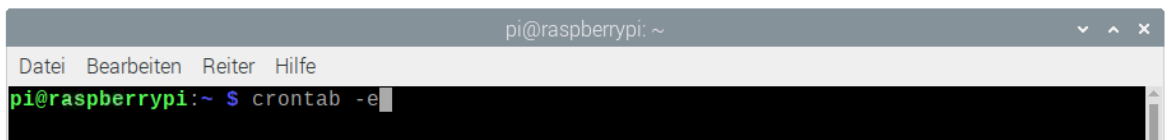


Abbildung 48: Befehl zum Abrufen der Crontab-Tabelle

Aus historischen Gründen wird eine crontab mit einem Kommandozeileneditor bearbeitet. Wurde noch kein Editor festgelegt, erscheint eine Auswahl an Editoren, welche man benutzen kann. Man hat sich hier für den Nano-Editor entschieden.

Falls vom User noch keine Cronjobs erstellt wurden, befinden sich in der Crontab-Datei lediglich Kommentare, die beschrieben wie Cronjobs zu verwenden sind. Kommentarzeilen beginnen mit einer Raute am Anfang jeder Zeile. Diese können bedenkenlos gelöscht werden.

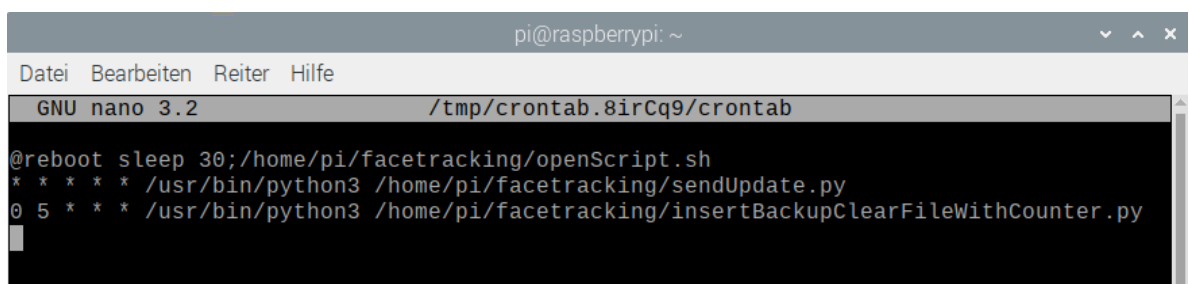
In dieser Datei werden die Zeitinformationen, als auch die auszuführenden Shell-Befehle, eingetragen. Dabei wird pro Zeile die Zeitinformation und daran anschließend in derselben Zeile das aufzurufende Programm oder der auszuführende Befehl eingetragen. Üblicherweise samt Angabe des absoluten Pfandes.

Die folgende Abbildung zeigt die obig beschriebene grundlegende Struktur des Aufbaus von crontab.

```
*      *      *      *      *      /Pfad/Befehl oder Programm
-      -      -      -      -
|      |      |      |      |
|      |      |      |      +----- Wochentag (0-7, Sonntag = 0 & 7)
|      |      |      +----- Monat (1-12)
|      |      +----- Tag (1-31)
|      +----- Stunde (0-23)
+----- Minute (0-59)
```

Für dieses Projekt wurden 3 Cronjobs verwendet.

1. Im ersten Cronjob wird festgelegt, dass nach einem Reboot bzw. nach dem Hochfahren vom Raspberry Pi 30 Sekunden gewartet wird und dann die Datei `openScript.sh` ausgeführt wird.
2. Im zweiten Cronjob wird jede Minute das Python Programm `sendUpdate.py` ausgeführt.
3. Die dritte Zeile setzt fest, dass jeden Tag um 5 Uhr morgens das Python Programm `insertBackupClearFileWithCounter.py` ausgeführt wird.



```
pi@raspberrypi: ~
Datei Bearbeiten Reiter Hilfe
GNU nano 3.2 /tmp/crontab.8irCq9/crontab
@reboot sleep 30;/home/pi/facetracking/openScript.sh
* * * * * /usr/bin/python3 /home/pi/facetracking/sendUpdate.py
0 5 * * * /usr/bin/python3 /home/pi/facetracking/insertBackupClearFileWithCounter.py
```

Abbildung 49: Cronjobs aus crontab

In den folgenden Kapiteln wird genauer darauf eingegangen was die Funktionalitäten dieser Programme sind.

5.3.3 Starten des Facetracking-Programms

Nach dem Hochfahren bzw. dem Rebooten des Raspberry Pis kümmert sich, wie bereits erwähnt, ein Cronjob darum, dass die Datei `openScript.sh` gestartet wird.

In dieser Datei wird zunächst ein Virtual Environment namens `cv` gestartet. Erst in dieser virtuellen Umgebung wird dann das Facetracking Programm ausgeführt.

```
#!/bin/bash

export DISPLAY=:0
source /usr/local/bin/virtualenvwrapper.sh
cd /home/pi/.virtualenvs/cv
source bin/activate
python3 /home/pi/facetracking/script.py
```

Abbildung 50: Code Snippet openScript.sh

5.3.4 Das Facetracking Programm

Gesichter zu erkennen und diese auch in Echtzeit zu verfolgen kann eine große Herausforderung für jeden Computer Vision- oder Bildverarbeitungsalgorithmus sein. Es gibt eine Vielzahl von Tricks, welche man anwenden kann, um einen solchen Algorithmus zu entwickeln damit dieser effektiv, ressourcenfreundlich und dennoch schnell genug ist um auf einem Kleincomputer wie dem Raspberry Pi laufen zu können und sowohl die Objekterkennung als auch die Objektverfolgung in Echtzeit zu übernehmen.

Der Quellcode für eine simple Verfolgung von Gesichtern mit OpenCV kann der Webseite pyimagesearch.com heruntergeladen werde. Siehe Quellverzeichnis.

Basierend auf diesem Quellcode wurden einige Modifikationen vorgenommen.

5.3.5 Modifikationen

Zunächst wurde ein Begrenzungsbereich, also der Bereich, in dem das Programm nach Gesichtern Ausschau hält, festgesetzt. Die Start- und Endpunkte `x` und `y` liegen im Bereich von 0 bis 1 und zeigen 0% bis 100% Breite oder Höhe des Begrenzungsbereichs. Dabei ist `endy` die zu überschreitende Schwelle. Das Gesicht muss über diese Schwelle verfolgt werden, um die Person zu zählen.

```
startx = 0.02  
endx = 0.98  
starty = 0.02  
endy = 0.33
```

Abbildung 51: Code Snippet scrip.py 1

Um diesen Bereich in der visuellen Ausgabe kenntlich zu machen, wird ein Rahmen basieren auf den gesetzten Werten von `startx`, `starty`, `endx` und `endy` um das Begrenzungsfeld gezeichnet. Weiters wird dem Rahmen eine Farbe mit RGB Werten und eine Dicke zugewiesen.

```
boxA = (int(startx * width), int(starty * height))  
boxB = (int(endx * width), int(endy * height))  
cv2.rectangle(image, boxA, boxB, (0, 255, 255), 2)
```

Abbildung 52: Code Snippet script.py 2

Der folgende Befehl wendet die Gesichtserkennung nur in diesem gesetzten Begrenzungsfeld an und gibt an wo genau in diesem Begrenzungsrahmen das Gesicht erkannt wurde indem es einen Rahmen um das erkannte Gesicht zeichnet.

```
if detect_face:
    faces = face_cascade.detectMultiScale(
        img_gray[int(starty * height):int(endy * height), int(startx * width):int(endx * width)], 1.2, 3, 0,
        minSize, maxSize)
    if len(faces) > 0:
        detect_face = False
        track_face = True
```

Abbildung 53: Code Snipped script.py 3

Überschreitet das verfolgte Gesicht die gesetzte Schwelle, erhöht sich die Zählervariable `count`, welche beim Ausführen des Programms zunächst mit 0 initialisiert wurde.

Jedes Mal, wenn diese Zählervariable inkrementiert wird, öffnet Python die Textdatei `FileWithCounter.txt` und der neue Wert der Variable `count` wird in diese Textdatei zwischengespeichert.

Da am Anfang des Programms bei der Deklaration der Variable `count` diese auch gleich mit 0 initialisiert wird, bedeutet dies, dass bei einem unerwarteten Absturz oder Reboot des Gerätes, und dem darauffolgenden Neustart des Programms, die Variable `count` erneut auf 0 gesetzt wird und die Zählung wieder von neu beginnt.

Um, in so einem Fall, eine gewisse Ausfallsicherung zu gewährleisten, werden bei der Zählung einige Bedingungen abgefragt, um die Anzahl der bislang gezählten Personen nicht zu verlieren.

Zunächst wird bei der ersten Zählung, also, wenn die Zählervariable `count` von 0 auf 1 inkrementiert wird, die Textdatei `FileWithCounter.txt` geöffnet, ausgelesen und abgefragt ob diese leer ist. Falls dies nicht der Fall

ist, kann davon ausgegangen werden, dass es bereits eine zwischengespeicherte Anzahl gibt. Dieser in der Textdatei Zwischengespeicherte Wert wird dann der Variable `count` zugewiesen und die Zählung wird bei dem neuen Wert fortgesetzt.

Falls der ausgelesene Wert der Textdatei `FileWithCounter.txt` jedoch leer ist, wird der gerade auf 1 inkrementierte Wert von `count` in diese Textdatei zwischengespeichert und die Zählung wird bei diesem Wert fortgesetzt.

```
if count == 1:
    readFromFile = open('/home/pi/facetracking/FileWithCounter.txt', 'r')
    readCount = readFromFile.read()
    if readCount == '':
        saveInFile = open('/home/pi/facetracking/FileWithCounter.txt', 'w')
        saveInFile.write(str(count))
        saveInFile.close()
    else:
        count = int(readCount)
        readFromFile.close()
saveInFile = open('/home/pi/facetracking/FileWithCounter.txt', 'w')
saveInFile.write(str(count))
saveInFile.close()
```

Abbildung 54: Code Snippet script.py 4

Damit diese Abfrage bzw. Ausfallsicherung funktioniert, ist es notwendig, dass am Anfang jeden Tages die Textdatei `FileWithCounter.txt`, in welcher die Anzahl der gezählten Personen zwischengespeichert wurden, auch geleert wird. Das Python Programm `insertBackupClearFileWithCounter.py` welches täglich, in unserem Fall um 5 Uhr morgens, durch unseren zweiten Cronjob ausgeführt wird, kümmert sich, unter Anderem, genau darum.



Dieses Programm führt mehrere Aktionen durch. Zunächst spricht es eine der PHP-Dateien auf den Server an, welche einen neuen Datensatz für den derzeitigen Tag erstellt.

Die genaue Funktionsweise dieser PHP-Datei wird in einem der folgenden Kapiteln genauer erläutert.

Weiters sorgt das Programm dafür, dass ein lokales Backup von der Textdatei `FileWithCounter.txt` erstellt wird. Diese wird in einen lokalen Ordner namens Backup kopiert und dann in „`Besucher [dazugehöriges Datum].txt`“ umbenannt.

```
#move File with counter into the Backup folder
shutil.move('/home/pi/facetracking/FileWithCounter.txt','/home/pi/facetracking/Backup')
#change directory to Backupfolder
os.chdir('/home/pi/facetracking/Backup')
yesterday = datetime.strftime(datetime.now() - timedelta(1), '%Y-%m-%d')
prevName = 'FileWithCounter.txt'
newName = 'Besucher_' + str(yesterday) + '.txt'
#rename moved file to "Besucher_[currentdate].txt"
os.rename(prevName, newName)
```

Abbildung 55: Code Snippet insertBackupClearFileWithCounter.py 1

Im Anschluss daran wird dann eine neue Textdatei `FileWithCounter.txt` für die Zwischenspeicherung der Anzahl gezählter Personen erstellt und mit einem leeren String befüllt.



```
#change directory to primary folder
os.chdir('/home/pi/facetracking/')
#create new empty File with counter
counterStart = ''
writeInFile = open('FileWithCounter.txt', 'w')
writeInFile.write(counterStart)
writeInFile.close()
```

Abbildung 56 Code Snippet insertBackupClearFileWithCounter.py 2

Somit ist die Textdatei `FileWithCounter.txt` leer und bereit wieder neu befüllt zu werden.

5.4 Datenbank

5.4.1 Struktur

Jeder Kunde, welcher einen Kundenfrequenzmesser besitzt, hat in der Datenbank eine eindeutige und fixe Geschäfts ID `G_ID`.

Jeder Tag von jedem Kunden hat einen eigenen Datensatz welcher aus den Spalten `D_ID` (die Datensatz ID), `datum` (das dazugehörige Datum), `kanzahl` (die Anzahl gezählter Personen für dieses Datum) und `G_ID` (der Geschäfts ID) besteht.



D_ID	datum ▾ 1	kanzahl	G_ID
2637	2020-05-26	1192	3
2636	2020-05-26	605	2
2635	2020-05-26	538	1
2634	2020-05-25	402	3
2633	2020-05-25	788	2
2632	2020-05-25	582	1
2630	2020-05-24	409	2
2629	2020-05-24	520	1

Abbildung 57: Datensätze der Tabelle daten

5.4.2 Datensatz erstellen

Das Python Programm `insertBackupClearFileWithCounter.py` kümmert sich, unter Anderem, darum, dass in der Datenbank so ein Datensatz zum täglichen befüllen der Anzahl der gezählten Personen zur Verfügung steht indem es durch das Ausführen eines `subprocesses` ein URL aufruft welche durch eine GET-Methode die PHP Datei `sendInsert.php` auf dem Server anspricht und diesem die, im Python Programm Hardcoded hinterlegte Geschäfts ID `G_ID`, übergibt.

```
import os, shutil
from datetime import datetime, timedelta
import subprocess
#hardcoded GeschäftsID
g_id = 3
#terminal command for deleting the return of curl command
v = "-so /dev/null"
#make database insert command
comm = "https://kfm.htl-ottakring.schulwebpace.at/dbZugriff/sendInsert.php?q_id="+str(g_id)
#execute command
subprocess.call(['curl', v, comm])
```

Abbildung 58: Code Snippet `insertBackupClearFileWithCounter.py` 3

Zunächst wird hier der übergebene Parameter, in diesem Fall die Geschäfts ID `G_ID`, in eine Variable `$g_id` gespeichert. Sobald die Verbindung zur Datenbank hergestellt wurde, wird mit dem SQL-Befehl `INSERT INTO` ein neuer Datensatz in der Tabelle `daten` angelegt welcher dann befüllt werden kann. Beim Erstellen eines neuen Datensatzes wird die Anzahl der Kunden `kanzahl` zunächst auf 0 gesetzt.

```
<?php
//erwartete Parameter
$g_id = $_GET['g_id'];

//Verbindungs Aufbau
try{
    $pdo = new PDO('mysql:host=localhost; dbname=u178388db1', 'u178388db1', 'ZnhX7nKW');
    //SQL Befehl in Variable schreiben
    $sql = "INSERT INTO `daten`(`datum`, `kanzahl`, `G_ID`) VALUES (CURRENT_DATE,0,$g_id)";
    //SQL Befehl
    $pdo->exec($sql);

    //Testausgabe
    echo "Geschäft: " . $g_id . "<br>";
    echo "Zeit: " . date("h:i:s") . "<br><br>";
    echo "Neuer Tag erfolgreich Erstellt.";

} catch(PDOException $e){
    die("ERROR: Could not connect. " . $e->getMessage());
}

?>
```

Abbildung 59: Code Snippet sendInsert.php

5.4.3 Datensatz aktualisieren

Um die in der Datenbank stehende Anzahl der gezählten Personen möglichst aktuell zu halten, wird die lokale zwischengespeicherte Anzahl, welche in dem Textfile `FileWithCounter.txt` steht, jede Minute ausgelesen und der dazugehörige Datensatz in der Tabelle `daten` aktualisiert.

Dies wird dadurch gewährleistet, indem minütlich durch unseren dritten Cronjob das Python Programm `sendUpdate.py` ausgeführt wird.

Hier wird zunächst die Anzahl der bislang gezählten Personen aus dem Textfile `FileWithCounter.txt` ausgelesen und dessen Inhalt der Variable `count` zugewiesen.

Durch das Ausführen eines `subprocesses` kann Python ein URL aufrufen welche durch die GET-Methode die eben ausgelesene Anzahl samt der

Geschäfts ID `g_id` übergibt. Diese URL spricht auf dem Server das PHP File `sendUpdate.php` auf.

```
#!/usr/bin/python3
import subprocess
#hardcoded GeschäftsID
g_id = 3
#terminal command for deleting the return of curl command
v = "-so /dev/null"
#open and read the file
with open('/home/pi/facetracking/FileWithCounter.txt','r') as file:
    #write content of file in variable count
    count = file.read()
# testUpdate.php
#make command
comm = "https://kfm.htl-ottakring.schulwebospace.at/dbZugriff/sendUpdate.php?kanzahl="+str(count)+"&g_id="+str(g_id)
#execute command
subprocess.call(['curl', v, comm])
```

Abbildung 60: Code Snippet `sendUpdate.py`

In diesem PHP File werden die übergebenen Parameter `g_id` der Variable `$g_id` und der ausgelesene Inhalt von `FileWithCounter.txt` der Variable `$kanzahl` zugewiesen. Weiters wird das aktuelle Datum durch die PHP-Funktion `Date()` der Variable `$datum` zugewiesen.

Nach dem Verbindungsaufbau mit der Datenbank wird der SQL-Befehl `UPDATE` ausgeführt, wobei die Spalte `kanzahl` des dazugehörigen Datensatzes der Tabelle `kunde` mit der aktuellen Kundenanzahl aktualisiert wird.


```
<?php
//erwartete Parameter
$kanzahl = $_GET['kanzahl'];
$g_id = $_GET['g_id'];
//Aktuelles Datum in Variable schreiben
$datum = date("Y-m-d");

//Verbindungsaufbau
try{
    $pdo = new PDO('mysql:host=localhost; dbname=u178388db1', 'u178388db1', 'ZnhX7nKW');
    //SQL Befehl in Variable schreiben
    $sql = "UPDATE daten SET kanzahl=$kanzahl WHERE G_ID=$g_id AND datum='$datum'";
    //SQL Befehl ausführen
    $pdo->exec($sql);

    //Testausgabe
    echo "Anzahl: " . $kanzahl . "<br>";
    echo "Geschäft: " . $g_id . "<br>";
    echo "Update um: " . date("h:i:s") . "<br><br>";
    echo "Änderung war Erfolgreich.";

} catch(PDOException $e){
    die("ERROR: Could not connect. " . $e->getMessage());
}

?>
```

Abbildung 61: Code Snippet sendUpdate.php

Zu Testzwecken werden die Parameter welche den PHP-Dateien `sendInsert.php` und `sendUpdate.php` übergebenen werden, nach dem ausführen des SQL Befehls, durch den Befehl `echo` ausgegeben. Jedoch ist diese nur zu sehen, falls die PHP-Dateien mit einem Browser aufgerufen werden. Diese Ausgaben werden beim Aufrufen der URL durch die `subprocess` Funktion der Python Programme `sendUpdate.py` bzw. `insertBackupClearFileWithCounter.py` nicht aufgerufen da diese bewusst vom von dem Prefix `curl` verworfen werden.



6 Android Applikation

Das Programmieren einer App war zunächst nicht Gegenstand dieser Diplomarbeit. Jedoch entwickelte sich im Laufe des Projektes immer mehr das Interesse an App Entwicklungen, weswegen man sich entschied dem Benutzer zusätzlich die Möglichkeit zu geben seine Kundenfrequenz auch über eine App angezeigt zu bekommen.

6.1 Aufgabenstellung Android Applikation

Die Applikation soll auf jedem Android Gerät verwendet werden können welches Googles Betriebssystemversion 7 namens Nougat oder höher besitzt. Diese soll auf unserer Website heruntergeladen werden können.

6.1.1 Funktionsweise

Die App soll optisch einfach und selbsterklärend gestaltet werden.

Beim Starten der App soll ein Login Fenster erscheinen, welche nach der Emailadresse und einem Passwort verlangt.

Nach einem erfolgreichen Einloggen wird dem Benutzer ein übersichtliches Dashboard angezeigt. In diesem werden im oberen Bereich der Benutzeroberfläche der Benutzername, seine Tätigkeit im Geschäft und der Geschäftsname angezeigt.

Auf dem Dashboard angelangt sollen im unteren Bereich der Benutzeroberfläche vier Auswahlmöglichkeiten in Form von Buttons angezeigt werden, welche jeweils zu einer der folgenden Ansichten führt:



- Statistik der letzten 30 Tage
- Statistik der letzten 7 Tage
- Statistik eines selbst ausgewählten Datums
- Besucheranzahl Heute

Um eine noch bessere Übersicht zu gewährleisten sollen am Dashboard auf dem Button, welcher zu der Ansicht der letzten 30 Tage führt, die durchschnittlichen Besucher der letzten 30 Tage angezeigt werden. Selbiges gilt für den Button, welcher einen zur Ansicht der letzten 7 Tage führt.

Auf dem Button, welcher zur Heutigen Ansicht führt, soll das heutige Datum, gekürzt im Format [Tag-Monat] angezeigt werden.

6.1.2 Design

Um sich die Benutzeroberfläche auch vor Augen führen zu können, wurde ein UI-Design gestaltet. Ein grobes Mockup der App wurde in Folge der Aufgabenstellung mittels Adobes Xd visualisiert. Diese Visualisierung zeigt die Funktionalitäten aller Buttons an, also alle Verbindungen der Buttons zu den jeweiligen Ansichten.

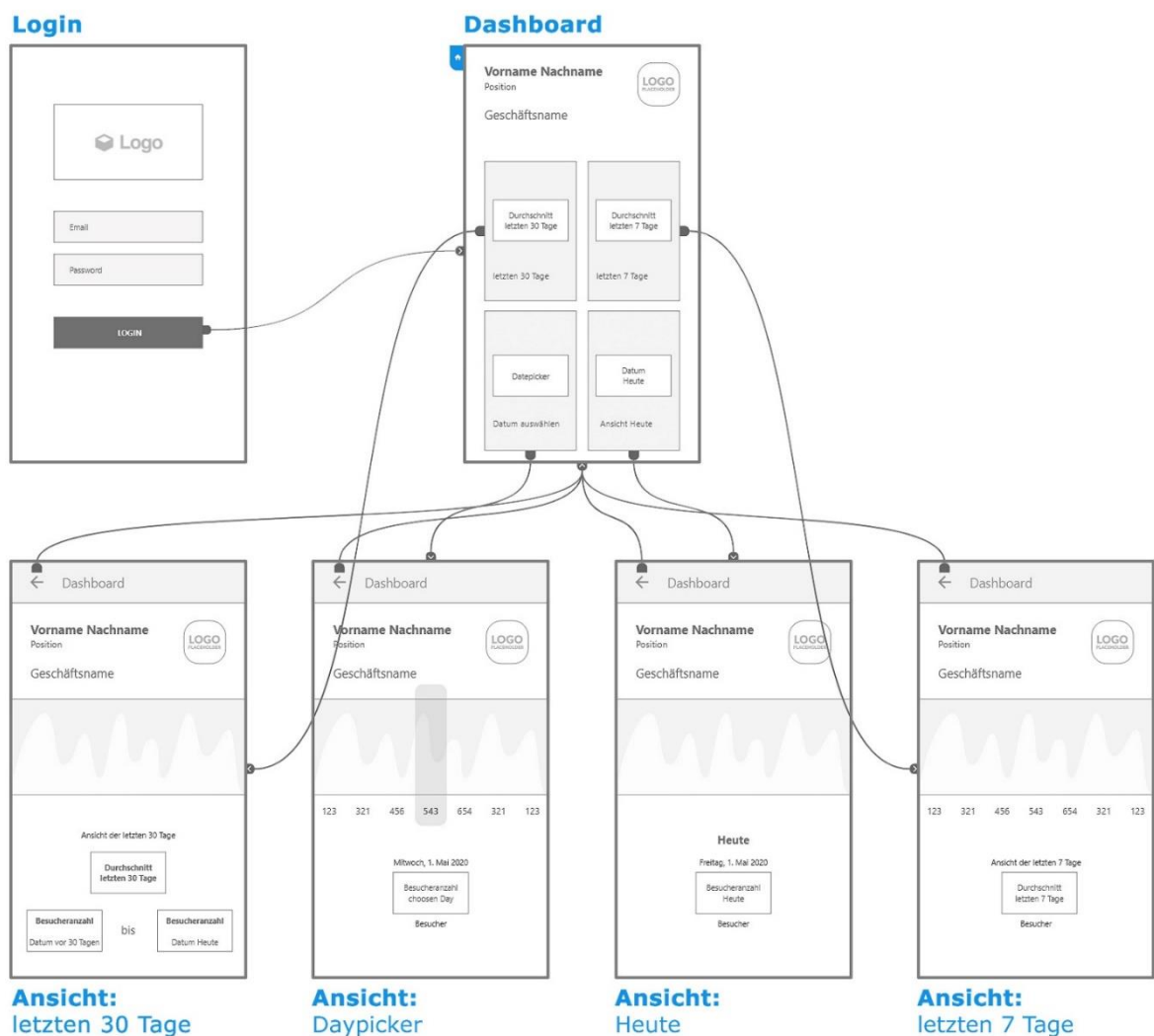


Abbildung 62: App Design Mockup

6.2 Verwendete Technologien

6.2.1 Java

Java ist eine objektorientierte Programmiersprache. Sie wurde von Sun Microsystems entwickelt und gehört mittlerweile Oracle. Sie ist eine der meist verbreitetsten Programmiersprachen und wurde daher im Laufe unserer Ausbildung in den letzten paar Semestern unterrichtet. Java wird verwendet, um mobile Apps, Web-Apps, Desktop-Apps, Spiele und vieles mehr zu entwickeln.



Abbildung 63:
Java Logo,
Quelle:
wikipedia.org

6.2.2 JDBC

JDBC steht für Java Database Connectivity und ist eine einheitliche Datenbank-Schnittstelle für Java. Da die App in Java programmiert wird und mit einer relationalen Datenbank kommunizieren soll, bietet sich JDBC für die Datenbankverbindung hervorragend an.



Abbildung 64:
JDBC Abbildung,
Quelle: oracle.com

6.2.3 MPAndroidChart

MPAndroidChart ist eine leistungsstarke und benutzerfreundliche Diagrammbibliothek für Android. Sie steht unter der Lizenz von Apache-License Version 2.0 und wurde von Philipp Jahoda für Android entwickelt. MPAndroidChart wurde ausgewählt, weil sie dank ihrer hervorragenden und umfangreichen Dokumentation einfach zu erlernen und einzusetzen ist.



Abbildung 65: MPAndroidCharts
Logo, Quelle: github.com



6.3 Umsetzung

Googles Android Apps werden grundsätzlich in Java oder Kotlin programmiert. Da in der schulischen Laufbahn Java als Programmiersprache unterrichtet wurde bietet sich diese besonders an weswegen sich auch auf Java geeinigt wurde.

Damit die Applikation Abfragen aus der Datenbank tätigen kann bietet sich JDBC als Datenbankschnittstelle für Java an.

Die grafische Aufbereitung der Statistiken in Form von Diagrammen soll unter Zuhilfenahme der Android Bibliothek MPAndroidChart erfolgen.

Auf die eben genannten Technologien wird in den folgenden Kapiteln genauer eingegangen

6.3.1 Nebenbemerkung

Da, wie bereits erwähnt, die Entwicklung einer Applikation anfänglich nicht Gegenstand des Projektes war, sondern sich diese zusätzliche Möglichkeit im Laufe des Projektes ergeben hat wird hier die Entwicklung nur oberflächlich erläutert.

Ein genaueres Eingehen auf die Entwicklung einer Android Applikation würde den Rahmen des Projektes sprengen.

Außerdem wird von einem gewissen Grundverständnis in objektorientierter und Android Programmierung ausgegangen.

6.3.2 Activity

Die Activity-Klasse ist eine wichtige Komponente einer Android-App. Die Art und Weise, wie Activitys gestartet und zusammengestellt werden, ist ein

wesentlicher Bestandteil des Anwendungsmodells der Plattform. Im Gegensatz zu Programmierparadigmen, bei denen Apps mit einer `main()` Methode gestartet werden, initiiert das Android-System Code in einer Activity-Instanz, indem bestimmte Rückrufmethoden aufgerufen werden, die bestimmten Phasen seines Lebenszyklus entsprechen.

Eine Activity ist ein Bildschirm der Android App. Es ist dem Fenster unter Windows sehr ähnlich. Eine Android-App kann eine oder mehrere Aktivitäten enthalten.

Eine Activity ist eine Kombination aus einer XML-Datei und einer Java Code-Datei.

6.3.3 XML

XML steht für Extensible Markup Language und ist eine Markup Language, ähnlich wie HTML.

Eine Markup Language ist eine Computersprache, die Tags verwendet, um Elemente in einem Dokument zu definieren. XML als solches ist sowohl für Menschen als auch für Maschinen gut lesbar, das heißt Markup-Dateien enthalten Standardwörter und keine typische Programmiersyntax. Außerdem ist es skalierbar und einfach zu entwickeln.

In Android verwendet man XML zum Schreiben von UI-Code, um mit diesen Benutzeroberflächen zu entwerfen.

6.3.4 Java Code

Grundsätzlich sollte man keine umfangreichen Operationen und Vorgänge wie zum Beispiel größere Datenbankabfragen oder Netzwerkzugriffe auf

dem UI-Thread, also dem Haupt-Thread, ausführen, da dies die Benutzeroberfläche blockieren könnte und die App nicht mehr reagieren würde.

Solche Operationen müssen in Hintergrund-Threads ausgeführt werden. Das Problem dabei ist jedoch, dass man nicht über den Hintergrund-Thread auf Ansichten zugreifen kann. Um also die Benutzeroberfläche zu aktualisieren, muss man wieder zurück zum UI-Thread.

6.3.4.1 AsyncTask

AsyncTask ist eine Android-Klasse, die es einfacher macht, am Hintergrund-Thread zu arbeiten und die Ergebnisse im UI-Thread zu veröffentlichen.

Eine AsyncTask erleichtert diesen gesamten Prozess. Sie ist nur für kurze Operationen geeignet, die höchstens einige Sekunden dauern, was sich hervorragend für diese App anbietet.

Der Ablauf der Methodenausführung wird in mehrere Teile gegliedert:

- `onPreExecute()`
- `doInBackground()`
- `onProgressUpdate()`
- `onPostExecute()`

Der Zweck der Verwendung von AsyncTask besteht normalerweise darin, die Benutzeroberfläche entsprechend dem Ergebnis der Hintergrundaufgabe zu aktualisieren.

Die folgende Abbildung zeigt anhand eines Sequenzdiagramms die vier Schritte des Ablaufs der Methodenausführung.

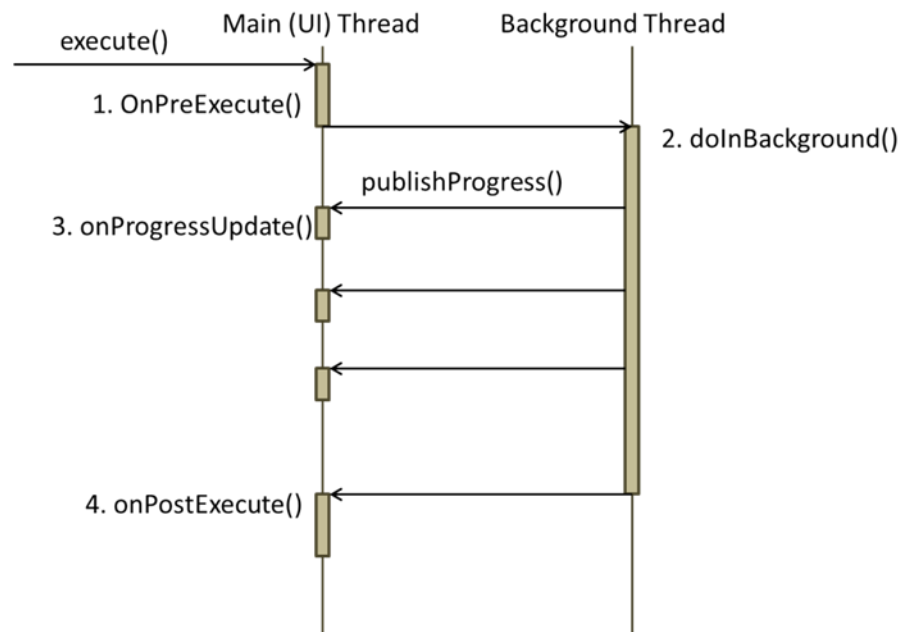


Abbildung 66: AsyncTask Sequenzdiagramm, Quelle: corochann.com

6.3.4.2 Activity Login

Beim Starten der App erscheint zunächst immer die `LoginActivity` bei welcher zwei Eingabefelder, E-Mail und Passwort, auf Input warten. Ergänzend dazu gibt es noch einen Login Button dem ein `onClickListener` zugewiesen ist welcher einen AsyncTask ausführt.

```

buttonLogin.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        new LoginTask().execute();
    }
});

```

Abbildung 67: Code Snippet LoginActivity.java. 1

Nach dem Tippen auf den Login Button durchläuft der AsyncTask dann seine üblichen Prozesse. In der `doInBackground()` Methode wird zunächst der JDBC Treiber geladen, eine Verbindung zur Datenbank aufgebaut und ein

SQL-Befehl initialisiert und ausgeführt. Der SQL-Befehl ruft in der Datenbank ein Stored Procedure auf, welches auf die Benutzerdaten zugreift, und übergibt diesem den eingelesenen Parameter, welcher in das Email-Eingabefeld eingegeben wurde.

In Folge wird das Resultat der Abfrage durchlaufen, und der Inhalt der einzelnen Felder werden in Variablen gespeichert.

```
class LoginTask extends AsyncTask<Void, Void, Void>{
    @Override
    protected Void doInBackground(Void... voids) {
        try{
            Class.forName(JDBC_DRIVER);
            Connection connection = DriverManager.getConnection(DB_URL,USER,PASS);
            Statement statement = connection.createStatement();
            String sql = "CALL login('"+textViewEmail.getText().toString()+"')";
            ResultSet rs = statement.executeQuery(sql);
            while(rs.next()){
                db_vorname = rs.getString( columnName: "firstName");
                db_nachname = rs.getString( columnName: "lastName");
                db_password = rs.getString( columnName: "password");
                db_geschaefte = rs.getString( columnName: "Name");
            }
        } catch (ClassNotFoundException | SQLException e) {
            error = e.toString();
        }
        return null;
    }
}
```

Abbildung 68: Code Snippet LoginActivity.java. 2

In der `onPostExecute()` Methode wird dann sichergestellt, dass sowohl eine Emailadresse als auch ein Passwort eingegeben wurde und diese Eingabefelder nicht ausgelassen wurden. Wurde eines der Felder nicht ausgefüllt, erscheint eine Fehlermeldung und die Login Activity startet sich neu.

Falls beide Eingabefelder einen Input hatten, wird das eingegebene Passwort mit dem aus der Datenbank ausgelesenen Passwort verglichen. Fällt der Vergleich negativ aus erscheint eine Fehlermeldung und die Login Activity startet sich neu. Ist dieser jedoch positiv, so startet die `MainActivity` welche unser Dashboard ist.

```
protected void onPostExecute(Void aVoid) {  
    if (textViewEmail.getText().toString().isEmpty() || textViewPassword.getText().toString().isEmpty()){  
        AlertDialog dialog = new AlertDialog.Builder( context: LoginActivity.this)  
            .setTitle("Leere Eingabe")  
            .setMessage("Bitte geben Sie etwas ein.")  
            .setPositiveButton( text: "OK", listener: null)  
            .show();  
        Button positivButton = dialog.getButton(AlertDialog.BUTTON_POSITIVE);  
        positivButton.setOnClickListener((v) → {  
            Intent intent = getIntent();  
            finish();  
            startActivity(intent);  
        });  
    }  
    else {  
        if (textViewPassword.getText().toString().equals(db_password)){  
            GlobalClass globalClass = (GlobalClass) getApplicationContext();  
            globalClass.setUsername(db_vorname+" "+db_nachname);  
            globalClass.setGeschaeft(db_geschaeft);  
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
            startActivity(intent);  
            Toast.makeText(getApplicationContext(), text: "Willkommen "+db_vorname.toString(), Toast.LENGTH_LONG).show();  
        }else{  
            AlertDialog dialog = new AlertDialog.Builder( context: LoginActivity.this)  
                .setTitle("Fehler")  
                .setMessage("Login Fehlgeschlagen. Versuchen Sie es erneut.")  
                .setPositiveButton( text: "OK", listener: null)  
                .show();  
            Button positivButton = dialog.getButton(AlertDialog.BUTTON_POSITIVE);  
            positivButton.setOnClickListener((v) → {  
                Intent intent = getIntent();  
                finish();  
                startActivity(intent);  
            });  
        }  
        if (!error.equals(""))  
            textViewError.setText(error);  
        super.onPostExecute(aVoid);  
    }  
}
```

Abbildung 69: Code Snippet LoginActivity.java. 3

6.3.4.3 Activity Dashboard

Unmittelbar nach dem Start der Activity „Dashboard“ wird der AsyncTask `TaskDashboard` ausgeführt.

```
new TaskDashboard().execute();
```

Abbildung 70: Code Snippet MainActivity.java. 1

In der `doInBackground()` Methode dieses AsyncTasks wird zunächst wieder der JDBC Treiber geladen, eine Verbindung zur Datenbank aufgebaut und ein SQL-Befehl initialisiert und ausgeführt. Der SQL-Befehl ruft in der Datenbank ein Stored Procedure auf, welches die durchschnittliche Besucheranzahl der letzten 30 Tage und die durchschnittliche Besucheranzahl der letzten 7 Tage ausliefert. Diese Werte werden dann in die jeweilig eigenen Variable gespeichert.

```
class TaskDashboard extends AsyncTask<Void, Void, Void>{
    @Override
    protected Void doInBackground(Void... voids) {
        try{
            Class.forName("com.mysql.jdbc.Driver");//.newInstance();
            Connection connection = DriverManager.getConnection(DB_URL,USER,PASS);
            Statement statement = connection.createStatement();

            String sql = "CALL avg()";
            ResultSet rs = statement.executeQuery(sql);
            while(rs.next()){
                avg_month = rs.getString( columnLabel: "avg_month");
                avg_week = rs.getString( columnLabel: "avg_week");
            }
        } catch (ClassNotFoundException | SQLException e) {
            error = e.toString();
        }
        return null;
    }
}
```

Abbildung 71: Code Snippet MainActivity.java. 2

In der `onPostExecute()` Methode werden die ausgelesenen Werte den jeweiligen Views zugewiesen, wodurch ein personalisiertes, übersichtliches und dynamisches Dashboard erstellt wird.

```
protected void onPostExecute(Void aVoid) {  
    GlobalClass globalClass = (GlobalClass) getApplicationContext();  
    textViewUsername.setText(globalClass.getUsername());  
    textViewGeschaeft.setText(globalClass.getGeschaeft());  
    textViewAVG_month.setText(avg_month);  
    textViewAVG_week.setText(avg_week);  
    setLogo();  
    currentDate();  
    if (!error.equals(""))  
        textViewAktualisiertUm.setText(error);  
    super.onPostExecute(aVoid);  
}
```

Abbildung 72: Code Snippet MainActivity.java. 3

Weiters gibt es auf dem Dashboard die schon erwähnten Buttons für die vier verschiedenen Ansichten. Jedem Button wurde ein `onClickListener` zugewiesen welcher jeweils ein eigenes `AsyncTask` ausführt.

```
buttonMonat.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new TaskMonth().execute();
    }
});

buttonWoche.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new TaskWeek().execute();
    }
});

buttonDatepicker.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DialogFragment datePicker = new DatePickerFragment();
        datePicker.show(getSupportFragmentManager(), tag: "date picker");
    }
});

buttonHeute.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new TaskHeute().execute();
    }
});
```

Abbildung 73: Code Snippet MainActivity.java. 4

Bis auf den `buttonDaypicker` führt jeder dieser Button einen eigenen AsyncTask aus.

Bei den AsyncTasks `TaskWeek()` und `TaskMonth()` wird in der jeweiligen `doInBackground()` Methode wieder der JDBC Treiber geladen, eine

Verbindung aufgebaut und ein MySQL-Befehl initialisiert und ausgeführt wobei wieder ein Stored Procedure ausgeführt wird, welches die Besucheranzahlen der letzten 30 bzw. der letzten 7 Tage zurückgibt.

Zusätzlich wird hier noch eine Methode `openMonthActivity()` bzw. `openWeekActivity()` aufgerufen welche die jeweilige Activity startet und die eingelesenen Werte dieser gestarteten Activity übergibt.

```
public void openMonthActivity(){
    Intent intent = new Intent( packageContext: this, MonthActivity.class);
    intent.putExtra(TAG1, tag1);
    :
    intent.putExtra(TAG32, tag32);
    intent.putExtra(AVG_M, avg_month);
    startActivity(intent);
    this.overridePendingTransition( enterAnim: 0, exitAnim: 0);
}
```

Abbildung 74: gekürztes Code Snippet MainActivity.java. 1

```
public void openWeekActivity(){
    Intent intent = new Intent( packageContext: this, WeekActivity.class);
    intent.putExtra(TAG1, tag1);
    :
    intent.putExtra(TAG9, tag9);
    intent.putExtra(AVG_W, avg_week);
    startActivity(intent);
    this.overridePendingTransition( enterAnim: 0, exitAnim: 0);
}
```

Abbildung 75: gekürztes Code Snippet MainActivity.java. 2

Beim triggern des Buttons `buttonDaypicker` wird zunächst ein Widget aufgerufen, wo man das Datum auswählen muss. Nach dieser Auswahl wird der AsyncTask `TaskDaypicker` gestartet wobei wieder in der `doInBackground()` Methode die Prozesse für die Verbindung und das Auslesen aus der Datenbank durchlaufen werden und in weiterer Folge die `openDayPickerActivity()` Methode ausgeführt wird welche die `DaypickerActivity` startet und die aus der Datenbank ausgelesenen Werte an diese Activity übergeben werden.

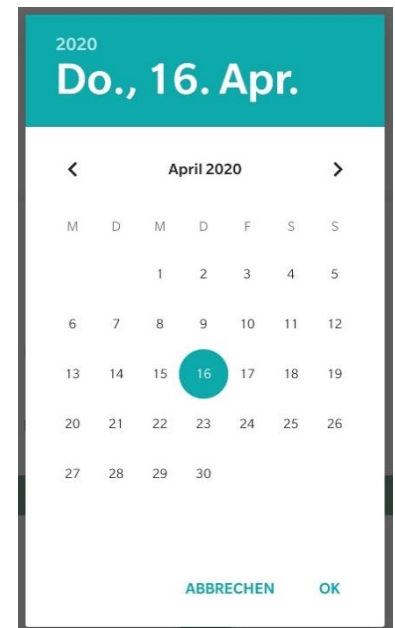


Abbildung 76: App Screenshot Datepicker

6.3.4.4 Activitys Ansichten

In jeder der vier möglichen Ansichten wird zunächst eine Toolbar mit einem Back-Button hinzugefügt, womit man bei Bedarf zurück zum Dashboard navigieren kann.

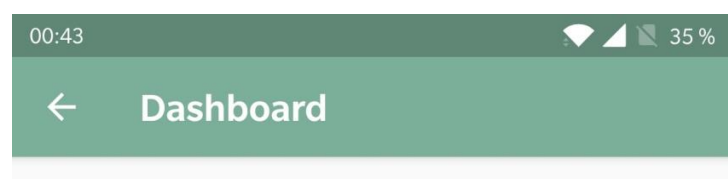


Abbildung 77: App Screenshot Toolbar

Weiters werden je nach Ansicht die übergebenen Werte ausgelesen und den jeweiligen Views zugewiesen. Ebenso wird bei jeder Ansicht, wo ein Diagramm erforderlich ist, ein Objekt vom Typ `LineChart` erzeugt. In weiterer Folge wird die Methode `makeLineChart()` ausgeführt wo die übergebenen Werte dem Diagramm zugewiesen werden und in weiterer Folge das Diagramm ausgegeben wird.

7 Webauftritt des Kundenfrequenzmessers

Der Webauftritt dient als Repräsentation des Kundenfrequenzmessers im Internet welcher den Benutzer sowohl über diesen informiert als auch die Kundenfrequenz visuell anhand von Diagrammen darstellt.

7.1 Aufgabenstellung Webauftritt

Als Frontend Ziele werden eine grundlegende Barrierefreiheit für Benutzer mit Einschränkungen und ein anpassungsfähiges Design der Seiten welches sich auf verschiedene Größen eines Bildschirms anpassen kann als Anforderungen an den Webauftritt betrachtet. Die Bedeutung von Accessibility und Responsive Web Design wird in späteren Kapiteln erläutert.

Zum Abfragen der Kundenfrequenz soll ein Kundenbereich mit übersichtlichem und minimalistischem Dashboard für diese erstellt werden und eine Wahlmöglichkeit aus verschiedenen Zeiträumen geboten werden.

Als Backend Ziele werden die Möglichkeiten der Abfrage der Kundenfrequenz durch die Webseite sowie das Anmelden eines Kunden und die Zuordnung der Daten zu diesem betrachtet.

7.1.1 User Experience

Zu Beginn der Entwicklungen wurde eine mögliche User Experience versucht zu beschreiben, um die Entwicklungsarbeiten zu unterstützen und eine ungefähre Vorstellung zu erlangen, wie der Umgang eines Kunden mit dem Kundenfrequenzmesser aussehen könnte.

7.1.1.1 Accessibility

Manche Benutzer einer Webseite oder Webanwendung sind in ihren physischen oder psychischen Fähigkeiten eingeschränkt, sei es durch ihr Alter, einen Unfall oder anderen Ursachen. Daher ist die Betrachtung und die Bedienung einer Webseite sowie Webanwendung, wie sie viele Menschen kennen und täglich vollziehen, für diese Personen erschwert.

Accessibility beschreibt diese Barrierefreiheit und versucht für alle Benutzer Webseiten und Webanwendungen konsumierbarer zu machen.

Richtlinien zum Umsetzen dieser Überlegungen wurden durch das World Wide Web Consortium W3C und deren Mitgliedern in den Web Content Accessibility Guidelines WCAG festgehalten. Das W3C ist ein Zusammenschluss an Firmen, Organisationen und Personen, die an der Ausarbeitung von Richtlinien arbeiten.

Personen, die ein schlechtes oder kein Sehvermögen aufweisen, können Webseiten und Webanwendungen wie sie sehende Personen erleben und bedienen nicht oder nur unvollständig nachvollziehen.

Dadurch entstehen für diese Benutzergruppe Problempunkte in der Navigation der Seite und schließlich im Verständnis der Informationen auf dieser, wie zum Beispiel die Frage auf welcher Seite man sich befindet oder welche Informationen auf dieser zu finden sind. In solchen Fällen kommen unterstützende Technologien wie ein Screen Reader der Inhalte auf der Seite vorliest und bei der Navigation durch diese hilft zum Einsatz.

Daher sollte soweit wie möglich semantische HTML-Syntax geschrieben werden, welche die Bedeutung über deren Inhalt vermittelt und einen guten und meist auch einfach umzusetzenden Ausgangspunkt für mehr Barrierefreiheit bildet, damit Hilfstechnologien, die auf ihn angewiesen sind, wie zum Beispiel die Navigation, zuverlässiger funktionieren.



7.1.1.2 Responsive Webdesign

Responsive Web Design ist ein Konzept und löst die Frage, wie mit der großen Zersplitterung der webdarstellenden Geräte und derer verschiedenen Bildschirmgrößen umgegangen wird.

Responsive Web Design hat sich vor allem im letzten Jahrzehnt als der Standard für Webentwicklung & -design etabliert. Um es zu ermöglichen braucht es weniger technische Hilfsmittel und verlässt sich dabei eher auf die Disziplin von Designern und Entwicklern.

Das Web ist von Natur aus flüssig und passt sich soweit seinen Umgebungsbedingungen, wie zum Beispiele die Bildschirmgröße und Bildschirmauflösung, an. Ein wenig muss dann noch nachgeholfen werden, um sicher zu stellen, dass das Dokument auf einem mobilen Gerät ebenfalls die gesamte Breite des Bildschirms nutzt.

Das letzte technische Hilfsmittel, dass Responsive Web Design mit bringt sind sogenannte Breakpoints, die definieren, dass das Layout ab einer bestimmten Größe „bricht“. Dadurch lässt sich CSS spezifisch für bestimmte Bildschirmgrößen schreiben.

7.1.2 Use-Cases

Mit Use-Cases sind bei der Webentwicklung Anwendungsfälle gemeint, welche beschreiben sollen, welche Benutzeroberflächen der Benutzer auf der Webseite erreichen kann.

7.1.2.1 Landing Page

Der erste Kontaktpunkt des Webauftritts ist die Landing Page, wo der Kunde über die Problematik, die der Kundenfrequenzmesser lösen soll, informiert wird, welche Services geboten werden sowie mögliche Einsatzgebiete des Kundenfrequenzmesser liegen. Auf dieser kann sowohl die Diplomarbeit als PDF als auch die Android Applikation heruntergeladen werden. Weiters kann man sich über die an der Umsetzung beteiligten Personen informieren. Mit dem Anmeldebutton kommt der Benutzer weiter in den Login-Bereich.

7.1.2.2 Login

Im Login-Bereich kann sich der Kunde mit seinen Login-Informationen anmelden, oder mit dem Zurückbutton den Bereich wieder verlassen, um auf die Landing Page zu gelangen.

Bei falschen oder keinen Eingaben in das Formular sollen Fehlermeldungen angezeigt werden.

7.1.2.3 Kundenbereich Dashboard

Im Kundenbereich wird für das jeweilige Geschäft am Dashboard des Kunden die Kundenfrequenz angezeigt, wobei dieser zwischen verschiedenen Zeiträumen für die Darstellungsmöglichkeit der Frequenz wählen kann.

Verlassen werden kann der Kundenbereich wieder über einen Zurückbutton, der auf den Login-Bereich weiterleitet.



7.1.3 Verwendete Technologien

In der Web-Entwicklung können Technologien grob in zwei Bereiche eingeteilt werden. In Frontend, welche im Browser ausgeführt werden, und in Backend welche auf dem Server ausgeführt werden. Jedoch verschwimmt diese Grenze zusehends, da die Weiterentwicklung von JavaScript in den letzten Jahren auch in das Backend vordringt.

Unterstützt wird diese Verschiebung weiter durch die an Bedeutung gewinnende Verbreitung des JAMstack (JavaScript, APIs, Markup) Prinzips und des weiter an Bedeutung gewinnenden Cloud-Computing welche jedoch nicht Teil dieser Diplomarbeit ist.

7.1.3.1 HTML

HTML steht für Hypertext Markup Language und ist der Grundstein zur Erstellung einer Webseite, dessen Aufgabe es ist Inhalt einer Seite zu strukturieren und diesem eine Semantik zu geben. Aufgrund der schnell wachsenden Popularität gilt HTML heute als offizieller Webstandard. Die HTML-Spezifikationen werden vom World Wide Web Consortium W3C verwaltet und entwickelt.

HTML wurde von Tim Berners-Lee, einem Physiker am CERN-Forschungsinstitut in der Schweiz, erfunden. Er veröffentlichte 1991 die erste Version von HTML, bestehend aus 18 HTML-Tags. Seitdem hat jede neue Version der HTML-Sprache neue Tags und Attribute, sogenannte Tag-Modifikatoren, zum Markup hinzugefügt. Laut der HTML-Elementreferenz von Mozilla Developer Network gibt es derzeit 140 HTML-Tags, von denen einige bereits veraltet sind, welche von



Abbildung 78:
HTML Logo,
Quelle:
[wikipedia.org](https://en.wikipedia.org/wiki/HTML)

modernen Browsern nicht mehr unterstützt werden. Das größte Upgrade der Sprache war die Einführung von HTML5 im Jahr 2014.

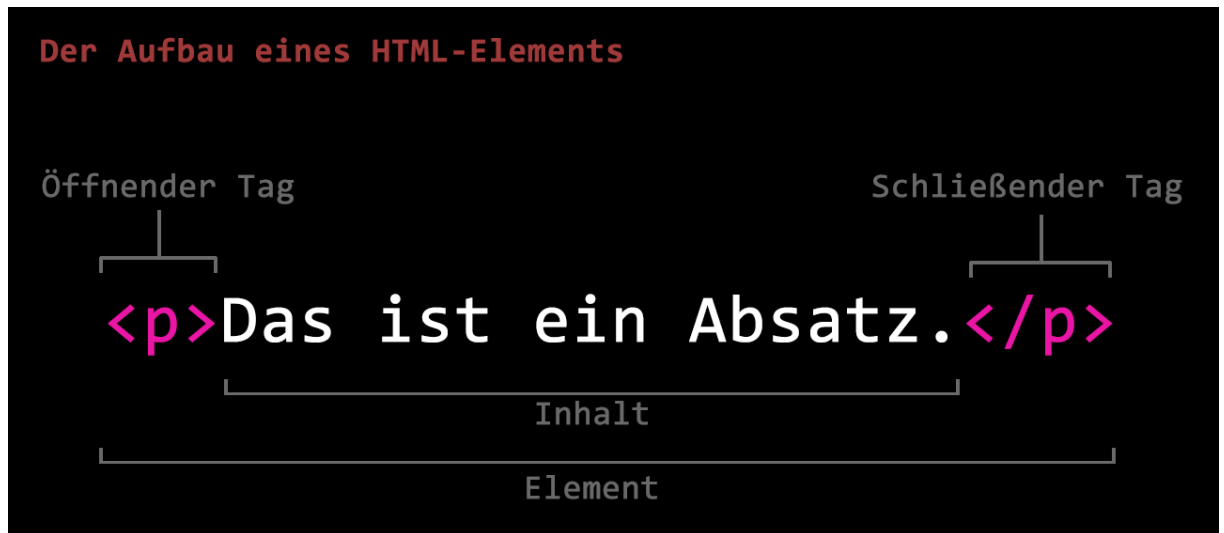


Abbildung 79: HTML Element Aufbau

7.1.3.2 CSS

CSS steht für Cascading Style Sheets und bildet den zweiten Grundstein in der Erstellung einer Webseite und wird verwendet, um das Layoutverhalten von HTML-Elementen auf einer Webseite zu verändern sowie das Aussehen dieser. CSS wurde erstmals 1994 vom Norwegen Håkon Wium Lie vorgeschlagen als dieser mit dem HTML-Erfinder Tim Berners-Lee am CERN arbeitete. Zu dieser Zeit wurden mehrere andere StyleSheet-Sprachen für das Web vorgeschlagen jedoch setzte sich CSS durch und wurde 1996 von W3C veröffentlicht.



Abbildung 80: CSS
Logo, Quelle:
[wikipedia.org](https://de.wikipedia.org/wiki/Cascading_Style_Sheets)

7.1.3.3 JavaScript

JavaScript ist der dritte Grundstein in der Erstellung einer Webseite und ist eine Programmiersprache, die nicht mit Java zu verwechseln ist. Sie wird im Web als Skriptsprache verwendet und verhilft Webseiten interaktiv zu werden, um auf Benutzereingaben reagieren zu können. Der Standard für JavaScript ist ECMAScript.



Abbildung 81:
JavaScript
Logo, Quelle:
wikipedia.org

Durch die Entwicklung von node.js wird es möglich JavaScript auch im Backend einzusetzen, da die Serverstruktur PHP unterstützt wurde nicht mit node.js gearbeitet.

JavaScript wird anders als viele Programmiersprachen interpretiert und just-in-time kompiliert. Obwohl JS mit Objekten arbeiten werden keine Klassen für die Erzeugung dieser wie in Java benötigt.

7.1.3.4 Highcharts

Highcharts ist eine Softwarebibliothek für Diagramme, die in reinem JavaScript geschrieben. Sie wird vom Unternehmen Highsoft, welches in Norwegen ansässig ist, entwickelt. Hauptsächlich wurde diese Softwarebibliothek von Torstein Hønsi, dem Chief Product Officer von Highsoft, ausgedacht, entwickelt und 2009 erstmals veröffentlicht. Seit der Einführung wird, die in internationalen Branchenmedien erwähnt, wo sie wiederholt als eine der besten JS-Chartbibliotheken bezeichnet wird.



HIGHCHARTS

Abbildung 82:
Highcharts Logo,
Quelle: Highsoft.com

Highcharts ist für den nichtkommerziellen Gebrauch kostenlos. Für die Verwendung in kommerziellen Anwendungen sind jedoch Lizenzen erforderlich.

7.1.3.5 AJAX

AJAX beschreibt ein Verfahren, wie Informationen von einem Server auf eine Webseite übertragen werden. Dabei sind mehrere Technologien unter dem Akronym Ajax zusammengefasst, dieses steht für Asynchronous JavaScript and XML. Der asynchrone Teil in Ajax beschreibt das Verhalten, wie Webseiten Daten laden, so, dass kein neu laden der Seite notwendig wird. Die Übertragung von diesen wurde in früheren Anwendungen durch XML ermöglicht, eine heute weiter verbreiterte Art der Datenübertragung ist JSON (JavaScript Object Notation), da es eine ähnliche Syntax zu JavaScript aufweist.



Abbildung 83:
AJAX Logo, Quelle:
dvglogo.com

Viele Programmiersprachen bieten eine Integration des Formates an und erleichtern so das Arbeiten mit verschiedenen Sprachen. Beim Informationsübertragen von einem Server auf eine Webseite wird der Inhalt der Übertragung zuerst codiert und am Zielort wieder decodiert.

7.1.3.6 XAMPP

XAMPP ist die Kombination des Apache Webserver, der Datenbank MySQL, wurde jedoch in MariaDB geändert, und der Skriptsprachen Perl und PHP. Daraus ergibt sich auch Akronym. Mit XAMPP erhält man die Basis, um



Abbildung 84: XAMPP
Logo, Quelle:
apacheefriends.org

eigene PHP-Skripten lokal auf dem eigenen Rechner laufen zu lassen und zu testen. Offiziell beabsichtigten die Designer von XAMPP, es nur als Entwicklungswerkzeug zu verwenden, damit Website-Designer und Programmierer ihre Arbeit auf ihren eigenen Computern ohne Zugang zum Internet testen können. Um dies so einfach wie möglich zu machen, sind viele wichtige Sicherheitsfunktionen standardmäßig deaktiviert.

7.1.3.7 PHP

PHP ist eine Skriptsprache und wird auf dem Server ausgeführt. Sie wird häufig in der Entwicklung von dynamischen Webseiten eingesetzt.



Der eigentliche PHP-Code wird nicht im Browser gelesen. Es ist möglich HTML und PHP in eine gemeinsame Datei zu schreiben. PHP wird auch, wie JavaScript interpretiert während das Skript Zeile für Zeile abgearbeitet wird und nicht kompiliert.

*Abbildung 85: PHP Logo,
Quelle: wikipedia.org*

Ähnlich zu Java können auch mit PHP-Klassen erstellt und objektorientierte Konzepte verwendet werden. Zum Einsatz kommt die Skriptsprache unter anderen, wenn Webseiten eine Datenbankbindung benötigen, oder Formulare behandelt werden müssen.

Es können auch komplette Bereiche einer Webseite auf dem Server gebaut oder einzelne Abschnitte dynamisch mit Inhalt versorgt werden. Das ist hilfreich, wenn sich der Inhalt des Abschnittes verändern kann.

7.1.3.8 WAVE

WAVE ist ein online Werkzeug mit denen Webentwickler ihre Webinhalte für Menschen mit Behinderungen und Einschränkungen zugänglicher machen können. WAVE kann viele Fehler in Bezug auf Barrierefreiheit und Richtlinien zur Barrierefreiheit von Webinhalten (WCAG) identifizieren

Dies kann über WAVEs eigene Webseite oder den Browsererweiterungen für Chrome und Firefox erfolgen.



Abbildung 86:
WAVE Logo,
Quelle:
webaim.org

7.1.3.9 NVDA Screen Reader

Screen Reader, sind Vorlesehilfen die Inhalte auf einer Webseite vorlesen, diese werden von Menschen, die nur unzureichend oder nicht sehen können, verwendet. Windows 10 bietet einen rudimentären Screen Reader. In

der Praxis werden jedoch meist von Drittanbietern entwickelte Reader verwendet. Einer von diesen ist der NVDA Screen Reader, er wird von NV Access als ein Open-Source Software entwickelt und wird auf Windows

Computer eingesetzt. Der NVDA Reader weist zwei unterschiedliche Modi auf, den Lesemodus und den Fokusmodus. Zum Beispiel werden einfache Textpassagen normal vorgelesen. Überschriften werden jedoch als solche und deren Ebenen und ihr Textinhalt vorgelesen. So werden Inhalt und die Bedeutung vermittelt. Ist der Fokusmodus ausgewählt wird es möglich Eingaben auf einer Seite zu tätigen, wenn zum Beispiel ein Formular ausgefüllt werden soll.



Abbildung 87:
NVDA Logo,
Quelle:
nvaccess.org

7.2 Umsetzung Frontend

In diesem Kapitel wird die Umsetzung der Aufgabenstellung aus der Sicht des Frontend beschrieben sowie Aufbau der einzelnen HTML Dokumente, die den Webauftritt bilden und das Erstellen dessen Layouts.

Weiters wird darauf eingegangen, wie AJAX eingesetzt wurde, um Daten von einem Server an die Webseite zu übertragen. Wie diese durch die externe Bibliothek Highcharts als ein Diagramm der Kundenfrequenz dargestellt werden und wie der Zeitraum der Frequenz durch User Interaktion verändert werden kann.

7.2.1 Aufbau HTML-Dokumente

Die Aufgabe von Webseiten ist es Informationen an ihre Besucher zu vermitteln, wie es durch klassische Informationmedien, wie zum Beispiel Zeitungen und anderen Dokumenten erfolgt wobei diese Informationen grafisch und inhaltlich organisiert werden. In Dokumenten für Webseiten wird dies durch den Einsatz von HTML-Elemente realisiert und durch optisch CSS gestaltet.

Ein besonderes Merkmal von HTML-Dokumenten im Gegensatz zu anderen Programmen ist, dass HTML-Dokumente trotz Fehlerhafte Syntax ausführbar sind.

7.2.1.1 Generelle Strukturelemente

In der Erstellung der HTML-Dokumente die gemeinsam den Webauftritt bilden wurden Grundlegende HTML-Elemente eingesetzt, die ein jedes

Dokument aufweisen muss, diese Elemente sind `<!DOCTYPE html>`, `<html>`, `<head>`, `<title>` und `<body>`.

```
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="css/style.css">
7      <title>Simple Statistics</title>
8  </head>
9  <body>
10 >      <header class="main-header"> ...
17 </header>
18 >      <main> ...
89 </main>
90 >      <footer> ...
115 </footer>
116 </body>
117 </html>
```

Abbildung 88: Grundstruktur HTML Dokumente

Das Erstellen des Dokuments wird mit `<!DOCTYPE html>` begonnen. Fortgesetzt wird es mit dem `<html>` Element samt Attribut `lang`, welches dem Browser signalisiert in welcher Sprache der Inhalt einer Webseite geschrieben ist und auch von Screen Readern als die vorlesende Sprache erkannt und verwendet wird. Ein aussagekräftiger Titel `<title>` einer Seite ist ebenfalls für Screen Reader wichtig jener wird beim Weiterleiten auf andere Seiten des Webauftrittes vorgelesen und führt zu mehr Verständnis.

```
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="css/indexStyle.css">
7      <title>Simple Statistics</title>
8  </head>
```

Abbildung 89: Metainformationen im `<head>` Element

Innerhalb des HTML-Elements `<head>` werden Informationen über das Dokument angegeben, die den Browser in der Ausführung der Webseite

unterstützten. Zu einen wurde festgelegt welcher Zeichensatz `charset` zum Einsatz kommt und wie das Verhalten der Webseiten auf kleinen Bildschirmen, Viewports genannt, erfolgen soll. Der Viewport beschreibt den aktuell angezeigten Bereich einer Seite und wird vom Browserfenster oder die Größe des Bildschirmes eines mobilen webfähigen Gerätes festgelegt. Weiters werden jegliche CSS-Stylesheets, JavaScript oder andere externe Ressourcen, die in die Seiten eingebunden werden innerhalb des `<head>` Elements geschrieben.

Die generelle Grundstruktur der HTML Dokumente ist durch die einzelnen Seiten konsistent. Jede verfügt über ein `<header>` Element, innerhalb dem sich das Logo und die Navigation befindet. Ein `<main>` Element, das den Hauptinhalt der jeweiligen Seite und ein `<footer>` Element, dass zusätzliche Informationen wie Social Media Links beinhaltet. Und werden beim Vorlesen der Seite durch den Screen Reader als eine Sprungmarke vorgelesen, so wird `<header>` als Bannersprungmarke und `<main>` als Hauptsprungmarke bezeichnet, welche zu mehr Verständnis beim Navigieren führt.

Diese und weitere HTML-Elemente bilden die Schnittstelle zwischen semantischer Syntax und Unterstützungstechnologien, wie einen Screen Reader in der Form eines Accessibility Trees AOM der durch den Browser erstellt wird.

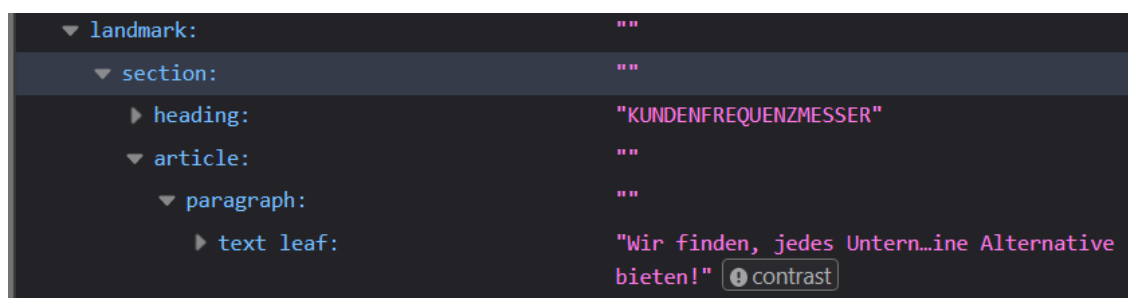


Abbildung 90: Accessibility Tree

Daher ist es nicht notwendig und nicht sinnvoll alle Informationen in den Dokumenten für Webseiten oder Webanwendung nur mit den HTML-Elementen `<div>` und `` zu gliedern. Beide Elemente bieten keinen bedeutungsvollen Mehrwert an sich, da sie eine Allzweckrolle einnehmen und eignen sich dann, wenn kein anderes HTML-Element für die Lösung der Problemstellung geeignet ist.

7.2.1.2 Elemente Landing Page

Die Landing Page `index.php` unterscheidet sich im Aufbau zu anderen Seiten des Webauftritts in der Sicht, dass Informationen, die einen gleichen inhaltlichen Kontext aufweisen in Sektionen mit `<section>` und `<article>` gegliedert werden. Jede dieser Sektionen verfügen über eigene Überschriften, welche für einen Screen Reader zu wichtigen Punkten in der Seite werden und stehen diesem für das Navigieren durch die Seite zur Verfügung.

Weiters kommen Bilder zum Einsatz welche Informationen transportieren und die für seheingeschränkte User nicht wahrnehmbar sind. Wie zum Beispiel die Downloadbuttons für die Diplomarbeit und die Android Applikation.



Abbildung 91: Screen Snippet Downloadbuttons Landing Page

Um diese Informationen vermitteln zu können werden für `` Elemente, mit dem HTML-Attribut `alt`, ein alternativer Text, welcher das Bild beschreibt, geschrieben und durch den Screen Reader vorgelesen. Der Text des `alt` Attributs wird, auch wenn Bilder nicht geladen werden, für sehende Benutzer auf der Seite angezeigt.

```

```

Abbildung 92: Code Snippet `` Element

7.2.1.3 Login Formular

Der Anmeldebereich `anmelden.php` unterscheidet sich im Aufbau zu den anderen Seiten des Webauftrittes anhand das Login-Formular.

Über dieses werden eingegebene Informationen an den Server zur Verarbeitung gesendet. Das HTML-Attribut `action` des `<form>` Elements legt fest welches Skript am Server die versendeten Informationen verarbeitet. Das HTML-Attribut `method` legt fest, dass die Übertragung durch POST stattfindet. Die Methode POST wurde verwendet, um keine sichtbaren Parameter in der URL bei deren Übermittlung anzuzeigen und bietet mehr Sicherheit. Erfolgen inkorrekte Eingaben durch den User werden Fehlermeldungen angezeigt.

Bitte Accountdaten eintragen

Email:

Passwort:

ANMELDEN

Abbildung 93: Screen Snippet Login Formular ohne Fehlermeldung

Email:

Bitte gültige E-Mail-Adresse
eintragen!

Passwort:

Bitte Ihr Passwort eintragen!

Abbildung 94: Screen Snippet Login Formular mit Fehlermeldung

Um die Benutzung eines Screen Readers beim Ausfüllen des Formulars zu erleichtern werden die jeweiligen Eingabefelder mit ihren zugehörigen Beschreibungen verbunden. Durch dieses Verbinden der HTML-Elemente `<input>` und `<label>` über deren Attribute `id` und `for` ist es dem Reader möglich das Eingabefeld und dessen Beschriftung gemeinsam vorzulesen.

```
<label for="email">Email:</label>  
<input type="email" id="email" name="email" size="30" value="<?php echo $email;?>">
```

Abbildung 95: Code Snippet `<label>` und `<input>` Elemente

7.2.2 CSS Layout

Es kann nicht kontrolliert werden, wie eine Webseite oder Webanwendung konsumiert wird und daher muss deren Layout anpassungsfähig designet werden.

Die Aufteilung und Abgrenzung von Inhalten in einzelne Layout-Elemente erleichtert das spätere Layout-Design mit den CSS-Techniken wie Grid und Flexbox, da je nach Grad der Aufteilung der Informationen in Bruchstücke ein flexibles System entsteht, um diese wie einzelne Bausteine in das Layout einzubauen.

Es gibt CSS-Frameworks, wie Bootstrap die Entwickler bei der Umsetzung von Responsive Web Design unterstützen, um jedoch bereits gewonnene Erfahrung mit den Layouttechniken Grid und Flexbox zu vertiefen und zu prüfen, ob es auch ohne die Verwendung eines solchen Frameworks möglich ist ein Design umzusetzen wurde kein CSS-Framework verwendet.



Abbildung 96: Beispielhafte Grid-Abbildung

7.2.2.1 Grid

Mit dem Befehl `display: grid;` wurde einem HTML-Element, welches als Grid-Container für weitere Elemente dient signalisiert, dass ein Layout

durch Grid erstellt und mit `grid-template-columns` und `grid-template-rows` in Spalten und Reihen eingeteilt wird.

```
19  header{
20      display: grid;
21      grid-template-columns: 5vw 1fr 2fr 1fr 5vw;
22      grid-template-rows: auto;
23      margin: 5vh 0 5vh 0;
24      row-gap: 5vh;
25      justify-items: center;
26  }
```

Abbildung 97: Code Snippet CSS Grid

Die Größe erzeugter Spalten und Reihen können mit fixen oder flexiblen Werten, wie den Fraction-Einheiten `fr` oder Viewport-Einheiten `vw` festgelegt werden. In der Umsetzung des Designs wurden diese Einheiten für Spalten verwendet, um so auf die Veränderung der Fenstergröße zu reagieren und ein anpassungsfähiges Gitter entstehen zu lassen, wobei die eigentliche Größe der Spalten und Reihen durch den Browser berechnet werden.

Auf dieses Gitter können Layout-Elemente nach Bereichen oder anhand von Linien verteilt werden. Sollte es vorkommen das mehr Elemente als erzeugte Reihen und Spalten vorhanden sind wird das Gitter automatisch von Grid um diese Anzahl erweitert, ohne sie vorher festlegen zu müssen. Diese Flexibilität bildet eine Stärke von Grid.

```
19  grid-template-areas:
20      "logo . . nav"
21      "welcome welcome welcome welcome"
```

Abbildung 98: Code Snippet Zuordnung der Interface Element im Grid 1

Mit den Befehlen `grid-template-areas` und `grid-area` wurde eine Zugehörigkeit zwischen dem Grid-Container und dem zu positionierenden Element geschaffen welches schließlich durch die Eintragung des Namens in `grid-template-areas` auf der Seite positioniert.

```
62 .navigation{  
63     grid-area: nav;  
64     justify-self: center;  
65     align-self: center;  
66 }
```

Abbildung 99: Code Snippet Zuordnung der Interface Element im Grid 2

Diese Methode entspricht der Positionierung nach Bereichen, jedoch wurde die Positionierung von Layout-Elementen durch Linien in der Umsetzung des Designs bevorzugt, da sie praktikabler als die Positionierung über Bereiche ist und zu weniger Code führte.

Bei dieser wurden mit den Befehlen `grid-column` und `grid-row` die Start-Linie, von der das Element ausgeht, sowie die End-Linie bis zu der sich das Element ausdehnt mit Zahlen festgelegt.

```
302     grid-column: 4 / 5;  
303     grid-row: 1 / 2;
```

Abbildung 100: Code Snippet Zuordnung der Interface Element im Grid 3

7.2.2.2 Breakpoints

Trotz des Einsatzes von flexiblen Größeneinheiten für ein Seitenlayout entsteht die unausweichliche Situation, in der dieses Seitenlayout und dessen Inhalt nicht mehr korrekt für die aktuelle Fenstergröße des Browsers optimiert ist und wie abgeschnitten wirkt das nennt sich Breakpoint. Um

das Seitenlayout wieder zu optimieren wurden Media Queries geschrieben, welche vergleichbar zu `if`-Anweisungen in Programmiersprachen sind. Diese Media Queries ermöglichen ab einer festgelegten Größe des Browserfensters einen Breakpoint im Design, damit neue CSS-Regeln in dieser Situation zum Einsatz kommen können.

```
41  @media screen and (max-width: 700px){  
42      .main-header > a{  
43          grid-column: 1 / 5;  
44          grid-row: 1 / 2;  
45      }  
46  }
```

Abbildung 101: Code Snippet Media Query

Media Queries können für verschiedene Gerätearten geschrieben werden und unterstützen ein breites Spektrum an Möglichkeiten, auf welche geprüft werden können. Die am häufigsten zum Einsatz kommenden Media Queries sind jene, welche Geräte mit Bildschirmen und deren Breite prüfen. So auch in der Umsetzung des Responsive Web Designs des Kundenfrequenzmessers.

Um ein Media Query zu erstellen wird der Befehl `@media screen and (max-width: [Wert])` verwendet, wobei die Anzahl an verwendeten Media Queries je nach Komplexität des Designs variiert kann. In diesen werden die vorhandenen Layout-Elemente und deren Position auf dem Gitter neu ausgerichtet, um den neuen Gegebenheiten eines kleineren Bildschirmes, wie sie bei Smartphones oder Tablets vorkommen, angepasst.

7.2.2.3 Flexbox

Nach Erstellung des Seitenlayouts wird Flexbox verwendet, um in bestimmten Sektionen ein spezifisches Design von Layout-Elementen unabhängig von dem Seitenlayout umzusetzen. Wie zum Beispiel die Abschnitte *Service*, *Einsatzbereiche* oder *Team* auf der Landing Page des Webauftritts des Kundenfrequenzmessers oder die Zeitraumbuttons am Dashboard im Kundenbereich.

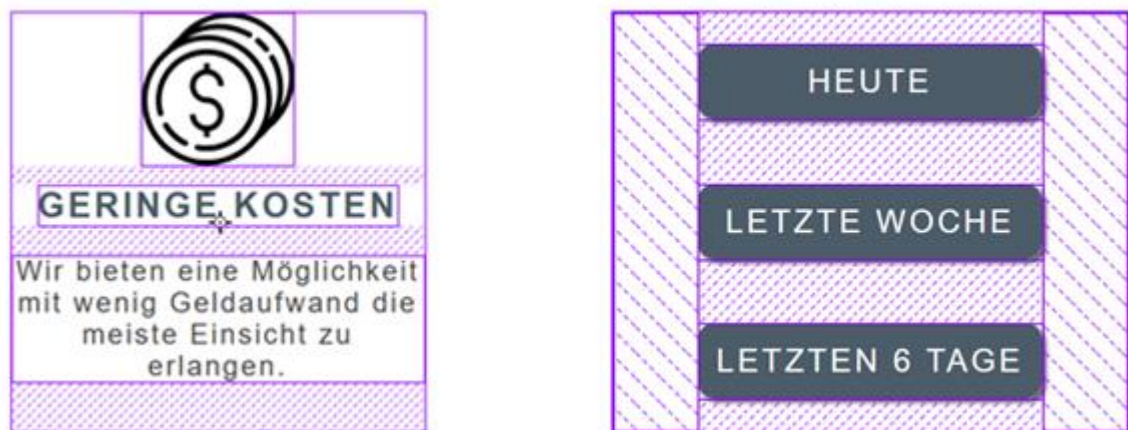


Abbildung 102: Flexitems auf Landing Page und Dashboard

Ähnlich zu Grid wurde auch Flexbox auf ein Container HTML-Element angewandt und mit dem Befehl `display: flex;` aktiviert.

```
89  display: flex;
90  flex-direction: row;
91  flex-wrap: wrap;
```

Abbildung 103: Code Snippet Flexbox

Die Stärke von Flexbox liegt in der Ausrichtung von Elementen in einer Reihe oder einer Spalte. Diese wird durch den Befehl `flex-direction` bestimmt. Mit dem Befehl `flex-wrap` wird festgelegt, wie sich Elemente

Verhalten sollen, falls sich der Platz für die Anzahl und / oder die Größe dieser Elemente innerhalb des Flex-Container, durch die Skalierung des Browserfensters verändert. Dadurch wird bestimmt, ob sich diese Elemente auf eine weitere Zeile aufteilen, oder nicht. Die Größe der Flexbox-Elemente richtet sich nach dem Inhalt, den sie fassen.

Sowohl die Technik Flexbox als auch die Technik Grid erlauben es, Layout-Elemente in ihren Containern einfach vertikal und horizontal auszurichten. Dies kann eine herausfordernde Aufgabe mit früheren Layouttechniken sein.

7.2.2.4 Wiederverwendung CSS-Code

Um CSS leichter kontrollieren zu können sowie wartungsfreundlicher zu gestalten, werden für CSS-Regeln, welche sich im Stylesheet wiederholen, CSS-Variablen verwendet. Diese CSS-Variablen werden auch Custom Properties genannt und werden wie in einer Programmiersprache verwendet, um Werte zu halten, wie zum Beispiel die Textgröße auf den Webseiten oder die Grundfarben dieser.

In diesem Fall ist es möglich jegliche CSS-Regel zu speichern, was spätere Änderungen im Design der Webseite erleichtert.

```
6      --defaultFontSizeText: calc(1em + 0.1vw);  
7      --defaultFontSizeHeadingLarge: calc(1.5em + 0.4vw);  
8      --defaultFontSizeHeadingSmall: calc(1.2em + 0.4vw);
```

Abbildung 104: Code Snippet Custom Properties 1

```
475 font-size: var(--defaultFontSizeHeadingSmall);  
476 color: var(--defaultColorHeadingLarge);
```

Abbildung 105: Code Snippet Custom Properties 2

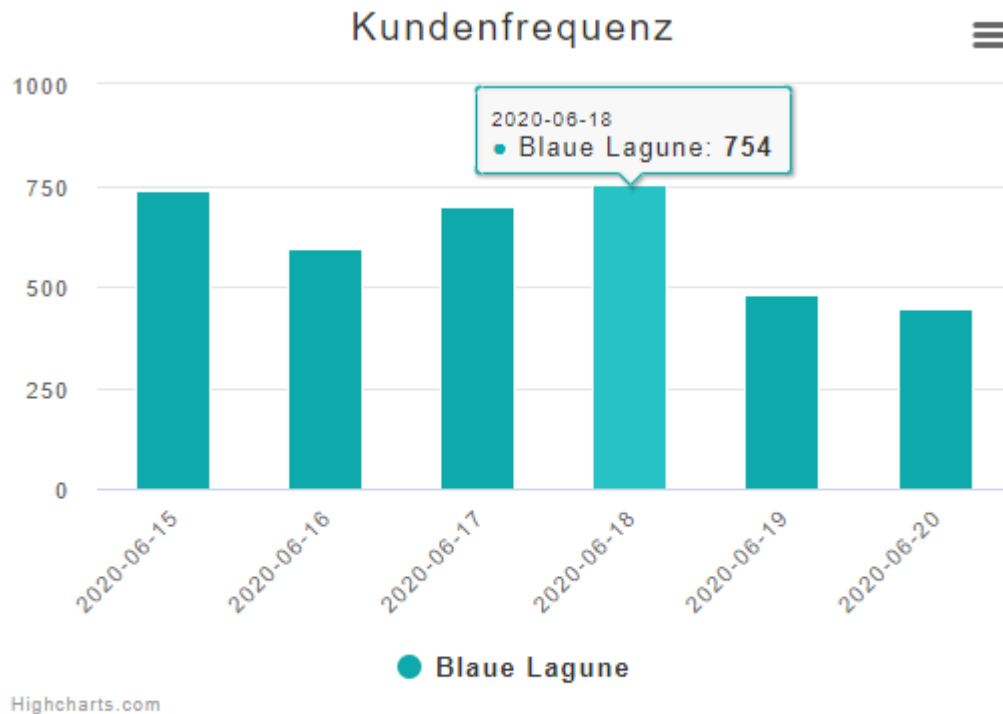
Ohne CSS-Variablen müsste jede Regel einzeln umgeschrieben werden. Durch die Verwendung von diesen wird der Wert jedoch nur einmal umgeschrieben.

7.2.3 Visualisierung der Kundenfrequenz

Nachfolgend wird beschrieben, wie das Konzept AJAX eingesetzt wird, um Daten von einem Server an die Webseite zu übertragen und wie sowohl diese Daten, durch die externe Bibliothek Highcharts, als ein Diagramm der Kundenfrequenz dargestellt als auch wie der Zeitraum der Frequenz durch User Interaktion verändert wird.

7.2.3.1 Highcharts

Um eine grafische Darstellung, in Form von Diagrammen, auf der Website umzusetzen, wurde Highcharts verwendet, da diese mit jeder Art von Backend Struktur zusammenarbeiten kann und umfangreiche Zusatzfunktionen wie verschiedene Arten von Diagrammen, Unterstützung von Accessibility und Responsive Design bietet. Man hat sich bei der Darstellung der Kundenfrequenz auf der Webseite für ein Balkendiagramm entschieden, da die Darstellung vergleichsweise klarer ausfällt als bei anderen Typen von Diagrammen.



Kundenzahl - Gesamt
3710

Abbildung 106: Diagramm der Kundenfrequenz am Dashboard

Um Highcharts verwenden zu können, müssen die JavaScript Ressourcen `highcharts.js`, das Exporting Modul und das Accessibility Modul sowie `chartScript.js` eingebunden.

Damit es zu keiner Blockade beim Aufbau der Webseite kommt wird die Ausführung von `chartScript.js` mit dem HTML-Attribut `defer` verzögert, um sicherzustellen, dass das Container-Element für das Diagramm durch den Browser erzeugt wurde.


```
18 <script src="https://code.highcharts.com/highcharts.js"></script>
19 <script src="https://code.highcharts.com/modules/exporting.js"></script>
20 <script src="https://code.highcharts.com/modules/accessibility.js"></script>
21 <script src="javascript/chartScript.js" defer></script>
```

Abbildung 107: Code Snippet Highcharts Module

Highcharts erlaubt es Optionen an den Konstruktor zu übergeben und sie während der Laufzeit zu verändern.

Damit eine Grafik erstellt wird, werden Optionen, wie die Art der Grafik und welche Achsen sie aufweisen sollen, vor der Erstellung festgelegt. Die Optionen werden in der Variable `drawChartOptions` gespeichert und mit dem HTML-Element, in welches die Grafik gezeichnet wird, an den Konstruktor von Highcharts übergeben, damit ein leeres Diagramm erstellt wird.

Mit den Highcharts eigenen Methoden `setCategories`, `setData` und `update` werden Datum, Kundenzahl und der Name des Geschäfts in die Objekte `xAxis` und `series` geschrieben.

```
44 barChart.xAxis[0].setCategories(dataArray[1]);
45 barChart.series[0].setData([parseInt(dataArray[2])]);
46 barChart.series[0].update({name: dataArray[0]});
```

Abbildung 108: Code Snippet Highcharts Methoden

Um die Verwendung von Highcharts rechtlich abzusichern muss ein Objekt namens `credits` bei jeder Erzeugung eines Diagrammes verwendet werden, welches einen Hyperlink auf die Highcharts Webseite anzeigt. Die wichtigsten Optionen für das Darstellen des Diagramms sind die beiden Objekte `xAxis` und `series`. Dabei bildet das `xAxis` Objekt und dessen `categories` Array den gewählten Zeitraum und das `series` Objekt den

Datensatz ab, welcher die Kundenfrequenz darstellt. Die Eigenschaft `name` und das `data` Array von `series` beinhalten den Geschäftsnamen und die Kundenanzahl.

```
2  var drawChartOptions = {
3      title: {
4          text: 'Kundenfrequenz'
5      },
6      chart:{
7          type: 'column'
8      },
9      credits: {
10         position: {
11             align: 'left',
12             x: 10
13         }
14     },
15     xAxis: {
16         categories: []
17     },
18     yAxis: [{
19         title: '',
20         allowDecimals: false
21     }],
22     series:[{
23         name: '',
24         data: []
25     }
26 ]
27 }
28
29 // Diagramm beim Laden der Seite erstellen
30 Highcharts.setOptions({colors:['#0ea9ab']});
31 var barChart = Highcharts.chart('container', drawChartOptions);
```

Abbildung 109: Code Snippet Highcharts Optionen

7.2.3.2 AJAX

Damit Daten für das Erstellen des Diagrammes und der Kundenfrequenz zur Verfügung stehen wird ein Ajax-Request für eine der auf dem Server befindlichen PHP-Skripten genutzt.

```
33 const xmlhttp = new XMLHttpRequest();
34
35 xmlhttp.onload = function (){
36     if(xmlhttp.status === 200){
37         dataArray = xmlhttp.response;
38         barChart.series[0].update({name: dataArray[0]});
39         barChart.xAxis[0].setCategories(dataArray[1]);
40         barChart.series[0].setData([parseInt(dataArray[2])]);
41         document.getElementById('kundenzahlGesamtFeld').innerHTML = dataArray[3];
42     }
43 }
44
45 xmlhttp.responseType = "json";
46 // Aufruf localhost
47 xmlhttp.open('GET', "http://localhost/kfm/phpScripts/kfAktuell.php");
48 // Aufruf Live-Server
49 // xmlhttp.open('GET', "https://kfm.htl-ottakring.schulwebpace.at/phpScripts/kfAktuell.php");
50 xmlhttp.send();
```

Abbildung 110: Code Snippet AJAX Request

Zuerst wird ein Objekt namens `XMLHttpRequest` erzeugt und in einer anonymen Funktion wird festgelegt wie mit der Antwort des Requests weiter verfahren wird.

In dieser Funktion wird zunächst auf eine Antwort des Servers geprüft. Bei einer positiven Antwort sollen anschließend die gelieferten Daten mit Highcharts in die Grafik eingetragen werden.

Mit der Methode `open()` beginnt ein Request. In dieser Methode wird festgelegt, dass der Request mit `GET` durchgeführt und an welche Adresse er gesendet werden soll. Die Funktion wird durchlaufen, sobald die `send()` Methode ausgeführt wird.

Der `responseType` für den Ajax-Request wird auf `json` gesetzt, um anzuzeigen, dass die übertragenen Daten in diesem Format übermittelt werden, diese findet in JSON statt damit die vom Server gelieferten Daten auch für JavaScript benutzbar werden.

7.2.3.3 Button Funktionalitäten

Das Dashboard bietet dem Benutzer die Möglichkeit aus verschiedenen Zeiträumen zu wählen, in welcher die Kundenfrequenz dargestellt werden soll.



Abbildung 111: Screenshot Snippet Userinterface Buttons Dashboard

Um das zu realisieren werden `<button>` Elemente auf der Webseite erstellt und mit dem HTML-Attribut `id` für eine Ansprache derer durch JavaScript versehen.

```
40 <button id="btnKfAktuell">Heute</button>
41 <button id="btnKfWoche">Letzte Woche</button>
42 <button id="btnKfSechsTage">Letzten 6 Tage</button>
```

Abbildung 112: Code Snippet Button Elemente

In der `chartScript.js` Datei werden die vorher erstellten `<button>` Elemente mit der Methode `getElementById` an konstante Variablen zugewiesen, da sich ihr Inhalt nicht verändert.

```
52 const btnKfAktuell = document.getElementById('btnKfAktuell');
53 const btnKfWoche = document.getElementById('btnKfWoche');
54 const btnKfSechsTage = document.getElementById('btnKfSechsTage');
```

Abbildung 113: Code Snippet Zuweisung Buttons

Mit der Methode `addEventListener` wird auf ein Klick-Event, welches durch den Benutzer ausgelöst wird, gewartet um anschließend einen AJAX-Request durchzuführen, welcher das benötigte PHP-Skript auslöst. Dadurch



wird die Kundenfrequenz neu dargestellt. Um das neu laden der Webseite nach dem Klick-Event zu verhindern wird die Methode `preventDefault` ausgeführt.

7.3 Umsetzung Backend

In diesem Kapitel wird die Wiederverwendung von HTML-Code, das Login-System und der Auslesevorgang der Kundenfrequenz aus der Datenbank beschrieben sowie das Anlegen eines neuen Accounts für einen Kunden.

7.3.1 Wiederverwendung HTML-Code

Um schneller und leichter arbeiten zu können, wird wiederkehrender HTML-Code in den Dokumenten, wie die Meta-Informationen zu Zeichensatz, Viewport sowie die Einbindung des CSS-Stylesheets und der `<footer>` Bereich, in eigene PHP-Dateien abgespeichert und mit dem Befehl `require()` in die Seiten geladen. Dies führt zu einem konsistenten Aussehen der Seiten, die den Webaufttritt bilden, da auch dieselben CSS-Regeln für die Seiten zum Tragen kommen.

7.3.2 Authentifizierung

Benutzer werden erst in den User-Bereich mit Dashboard weitergeleitet, wenn diese sich erfolgreich angemeldet haben. Eine Session speichert Informationen über den angemeldeten Kunden auf dem Server und bietet eine sichere Art mit diesen Informationen zu arbeiten.

Mit der Funktion `session_start()` wird eine Session begonnen und eine Session-Variable mit der E-Mail-Adresse des Benutzers angelegt. Die

Session-Variable wird benötigt, um zu bestimmen welches Geschäft und dessen Kundenfrequenz zu dem angemeldeten Benutzer gehört, damit die zugehörigen Daten diesem Benutzer zugeordnet werden kann. Die Session-Variable ist global verfügbar, um bei der Ausführung weiterer PHP-Skripten verfügbar zu sein.

7.3.2.1 Login / Logout

Kommt es zu keiner fehlerhaften Eingabe der Login-Informationen, wird eine SQL-Abfrage an die Datenbank abgesetzt, welche nach der E-Mail-Adresse des Benutzers sucht und prüft, ob ein bestehender User-Account vorliegt.

```
43 $login = $pdo->prepare("SELECT email, password FROM kunde WHERE email=:email");
44 $login->bindParam(':email', $email);
45 $login->execute();
46 if($login->rowCount() == 1){
```

Abbildung 114: Code Snippet Login

Bei keiner oder falsch Eingabe von Login-Informationen in das Formular werden Fehlermeldungen generiert, die den Benutzer darauf hinweisen, dass es zu einer fehlerhaften Eingabe kam.

Alle Benutzereingaben werden durch die Funktion `pruefen()` auf Leerzeichen sowie spezielle Zeichen, wie HTML-Zeichen, geprüft und erst bei Richtigkeit weiterverwendet. Da die Verarbeitung und Eingabe der Login-Informationen über das gleiche Dokument geregelt wird ist das Entfernen von HTML-Zeichen in den Eingaben wichtig, da sonst auf diesem Weg eine Cross-side-scripting Attacke durchgeführt werden kann. Unter Cross-side-scripting XSS Angriffen wird die Ausführung von potenziell

gefährlichem Code durch eine Eingabe dessen über ein Formular verstanden.

Ein weiterer Angriffspunkt sind SQL-Injections. Diese sind SQL-Anweisungen, welche über Eingabeformulare an die Datenbank versucht übermittelt zu werden. Um SQL-Injections zu verhindern wird die SQL-Abfrage, für das Anmelden auf der Seite, nach einem bestehenden User-Account mit Befehl `prepare()` vorbereitet und anschließend erst mit den Eingaben aus dem Formular als Parameter in diese geschrieben und ausgeführt. Bei übereinstimmender E-Mail-Adresse wird das zugehörige verschlüsselte Passwort aus der Datenbank gelesen und mit dem eingegebenen Passwort durch die Methode `password_verify()` verglichen.

Bei einem positiven Ergebnis wird die Session-Variable gesetzt, die passende Kunden ID zu dem angemeldeten Account gesucht und mit dem heutigen Datum in die Datenbank eingetragen. Der User wird mittels HTTP-Request in den User-Bereich weitergeleitet. Ansonsten wird bei nicht Übereinstimmung eine Fehlermeldung ausgegeben. Versucht ein nicht angemeldeter Benutzer den User-Bereich aufzurufen wird dieser auf die Anmeldeseite umgeleitet und darüber informiert, dass nur bereits angemeldete Accounts diesen betreten dürfen. Ob bereits eine aktive Session besteht wird durch den `loggedin` Parameter überprüft, welcher an den HTTP-Request angehängt ist.

```
3  if (isset($_REQUEST['loggedin'])) {  
4      session_start();  
5  
6  } else {  
7      // localhost Weiterleitung  
8      header('Location: http://localhost/kfm/anmelden.php?notauthorized');  
9  
10     // Live-Server Weiterleitung  
11     header('Location: https://kfm.htl-ottakring.schulwebpace.at/anmelden.php?notauthorized');  
12 }
```

Abbildung 115: Code Snippet Session Start

Meldet sich ein Benutzer aus dem Kundenbereich ab wird wieder der Parameter, welcher an den HTTP-Request angehängen ist, geprüft, ob dieser gesetzt ist und beendet die Session durch `session_destroy()`.

```
4      if(isset($_REQUEST['logout'])){  
5          session_destroy();  
6      }
```

Abbildung 116: Code Snippet Session End

7.3.2.2 Account anlegen

Wenn sich Kunden für den Einsatz des Kundenfrequenzmessers entscheiden wird durch uns ein neuer Account in der Datenbank angelegt. Für diesen Prozess wurde eine eigene Seite `create.php`, die nicht Teil des Webauftretes ist, geschrieben, welche nur durch das Team genutzt wird.

Über ein Formular werden alle benötigte Information des neuen Kunden eingetragen, auf keine Eingaben, Leerzeichen und HTML-Zeichen geprüft und durch den Server bearbeitet. Das eingegeben Passwort wird mit der PHP-Funktion `password_hash()` verschlüsselt, es wird der Standard-Algorithmus namens `bcrypt` von PHP angewandt, dieser wird durch die Entwickler an die neuersten Entwicklungsstand angepasst, daher ist es wichtig, dass das Passwort haltende Feld in der Datenbank mit ausreichender Speicherkapazität angelegt ist.

7.3.3 Daten für das Diagramm

7.3.3.1 Datenbankverbindung

Damit die für das Darstellen der Frequenz im Dashboard benötigten Daten mit den PHP Skripten aus der Datenbank ausgelesen werden können wird zuerst eine Verbindung zu der Datenbank aufgebaut. Es wurden zwei unterschiedliche Verbindungen geschrieben, da eine lokale sowie eine externe Datenbank angesprochen werden. Ein Auslagern der Verbindung in externe Dateien und späteres Einbinden dieser in die benötigten PHP Skripten reduziert zu schreibenden Code und erleichtert ein späteres Wechseln der zu Grunde liegenden Datenbank, falls die Weiterentwicklung des Projektes dies verlangt, da die Verbindungsdatei nur einmalige Veränderung werden muss.

```
6      define('dbServer', 'localhost');
7      define('dbUsername', 'root');
8      define('dbPassword', '');
9      define('dbName', 'u178388db1');
10
11     try{
12         // $pdo = new PDO("mysql:host=$servername; dbname=$dbName", $username, $password);
13         $pdo = new PDO("mysql:host=".dbServer."; dbname=".dbName."; charset=utf8", dbUsername,
14             dbPassword);
15         $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
16     }catch(PDOException $e){
17
18         die("Verbindung zu DB nicht möglich. " . $e->getMessage());
19     }
20 }
```

Abbildung 117: Code Snippet Datenbankverbindung

Für den Aufbau der Verbindung werden zunächst notwendige Werte wie der Datenbankserver, der Benutzer, ein Passwort und der Datenbankname definiert um anschließend mit PDO die Verbindung aufzubauen. Dritt ein Problem beim Aufbau der Verbindung auf wird eine Fehlermeldung generiert und angezeigt.

In der Diplomarbeit wurde mit der PDO-Erweiterung von PHP eine Verbindung zu der Datenbank aufgebaut, da sie eine Vielzahl an unterschiedlichen Datenbanken unterstützt. PDO ist eine von vielen Möglichkeiten mit PHP eine Verbindung zu einer Datenbank zu etablieren.

Je nach dem mit welcher Datenbank gearbeitet wird können oder müssen andere Verbindungsarten verwendet werden. Wie schon erwähnt liegt die Stärke von PDO in ihrer vielseitigen Anwendungsmöglichkeit. Dieser Vorteil wird ersichtlich, wenn eine andere Methode, wie zum Beispiel MySQLi, für das Erstellen einer Verbindung an Datenbanken betrachtet wird. Mit MySQLi ist es nur möglich MySQL-Datenbanken anzusprechen im Gegensatz zu PDO, daher hat sich das Team für die zweite Lösung entschieden. Eine frühere Vorgängerversion von MySQLi ist MySQL und wird nur noch selten in der Entwicklung genutzt.

7.3.3.2 Initiales Auslesen

Nach erfolgreichem Login des Users und Weiterleitung dessen in den User-Bereich wird initial die Kundenfrequenz für den aktuellen Tag angezeigt dies erfolgt durch einen Ajax-Request aus dem Frontend für das jeweilige PHP-Skript, in dem Fall `kfAktuell.php`.

Es wird die benötigte Datenbankverbindung inkludiert und eine Session für dieses Skript gestartet. So kann auf die zuvor gespeicherte Mail-Adresse des eingeloggtten Users zugegriffen werden, um ihn mit seinem Geschäft zu verbinden und die zugehörigen Daten anzuzeigen. Dieses zugehörige Geschäft wird als `G_ID` an die frequenzauslesende SQL-Abfragen übergeben.

```
20      $email = $_SESSION['email'];
21
22      $geschaeft = "SELECT g.G_ID FROM session s
23                  INNER JOIN kunde k ON s.K_ID=k.K_ID
24                  INNER JOIN geschaeft g ON k.K_ID=g.K_ID
25                  WHERE s.datum = '$aktuellesDatum' AND k.email = '$email'";
```

Abbildung 118: Code Snippet SQL-Abfrage

Weiters werden Arrays angelegt, welche die zurückgelieferten Werte aus der Datenbank abspeichern. Für die Abspeicherung bieten sich hier Arrays an, da diese eine beliebige Anzahl an Werten halten können und eine spätere Erweiterung des Projektes erleichtern. Mit der SQL-Abfrage, welche die Kundenfrequenz ausliest, werden die benötigten Daten für den Namen des Geschäfts, den abzubildenden Zeitraum und die Anzahl der Kunden aus der Datenbank ausgelesen und mit der Funktion `array_push()` in die vorher erwähnten Arrays geschrieben.

Mit der Funktion `json_encode()` werden die generierten Werte in den Arrays an den aufrufenden Ajax-Request zurückgeliefert.

7.3.3.3 Änderung des Zeitraums

Wird im Dashboard ein neuer Zeitraum durch den User ausgewählt erfolgen erneut Ajax-Requests für die jeweils benötigten PHP-Skripten `kfAktuell.php`, `kfWoche.php` und `kfLetztenSechsTage.php`.

Es wird eine Session für das jeweilige Skript gestartet und die Datenbankverbindung inkludiert sowie die Geschäfte dem angemeldeten User zu geordnet. Alle generierten Daten werden wieder mit JSON an den Request übertragen.



Für die Ansicht der Kundenfrequenz der letzten Woche wird mit den Funktionen `date()` und `strtotime()` die Zeitspanne berechnet. Dabei wird das Datum in Variablen die den Beginn sowie das Ende des Zeitraums darstellen gespeichert und bilden in dem Where-Clause der SQL-Abfrage den Filter, nach dem die Datensätze in der Datenbank durchsucht werden.

Für die Ansicht der Kundenfrequenz der letzten sechs Tage wird ausgehen vom aktuellen Datum fünf Tage zurück gerechnet. Die Daten werden wieder in Variablen, welche den Beginn und das Ende des Zeitraums darstellen, geschrieben und für die SQL-Abfrage der Frequenz als Filter verwendet.

Kommt in dem Zeitraum ein Sonntag vor oder wird die Abfrage an einem Sonntag ausgeführt, würden dadurch am Dashboard leeren Lücken in der Darstellung der Kundenfrequenz entstehen, da sonntags keine Aufzeichnungen stattfinden. Um das zu vermeiden werden in dem Array `$zeitraumArray` die letzten sechs Tage abgespeichert. Im nächsten Schritt wird mit der Methode `array_search()` das Array durchsucht, ob der Tag, an dem die Frequenz abgefragt wird, ein Sonntag ist, oder sich ein Sonntag in den letzten sechs Tage befindet.

Ergibt die Suche ein positives Resultat wird das Startdatum der Abfrage neu angepasst und um sechs Tage zurückverlegt.



7.4 Testing

Während der Entwicklungsarbeiten werden in verschiedenen Browsern mit lokalen sowie online Werkzeugen Tests der Webseite und deren Funktionen durchgeführt.

Dieses Verfahren ist als Cross-Browser-Testing bekannt und zeigt das Verhalten der Webseite in diesen Umgebungen, um mögliche Problempunkte zu erkennen. Bei den Testungen hat sich gezeigt, dass der Browser Microsoft Edge, der Nachfolger des früheren Browsers Internet Explorer, die meisten Problempunkte aufweist.

7.4.1 Lokale Entwicklungsumgebung

Um die bestehenden Funktionalitäten des Webauftritts durch Verbesserungen und Veränderung dieser nicht zu behindern wird in einer lokalen Entwicklungsumgebung gearbeitet.

Eine lokale Entwicklungsumgebung wird mit dem Softwarepaket XAMPP simuliert. Dieser Entwicklungsserver bietet den Vorteil, dass in relativer Sicherheit an Funktionen und an der Webseite gearbeitet werden kann, ohne die live Webseite zu beeinträchtigen.

7.4.2 Browser Developer Tools

Mit den Developer Tools wurde der Browser, die Barrierefreiheit und das Responsive Design getestet.

Developer Tools sind in Browser integrierte Werkzeuge, die den Entwickler beim Testen von Webseiten bei verschiedenen Problempunkten unterstützen.



Auch wie sich das Design der Webseite auf verschiedene Größen von Bildschirmen verhält wurde mit den Developer Tools von Browsern getestet.

7.4.3 Accessibility Tests

Das Testen der Accessibility wird mit dem online Tool WAVE und dem NVDA Screen Reader sowie den Developer Tools der Browser durchgeführt.

Mit WAVE wurden die Seiten des Webauftrittes auf ein breites Spektrum an Problemen getestet. Diese getesteten Probleme reichen von fehlenden Beschreibungstexten eines Bildes bis hin zu Kontrastproblemen bei Farben, welche für Texte und das visuelle Design der Seiten eingesetzt werden.

Dabei erstellt WAVE eine automatisierte Zusammenfassung, welche einen Überblick über die bestehenden Probleme auf der Seite gibt und erleichtert dadurch die Lösung dieser.

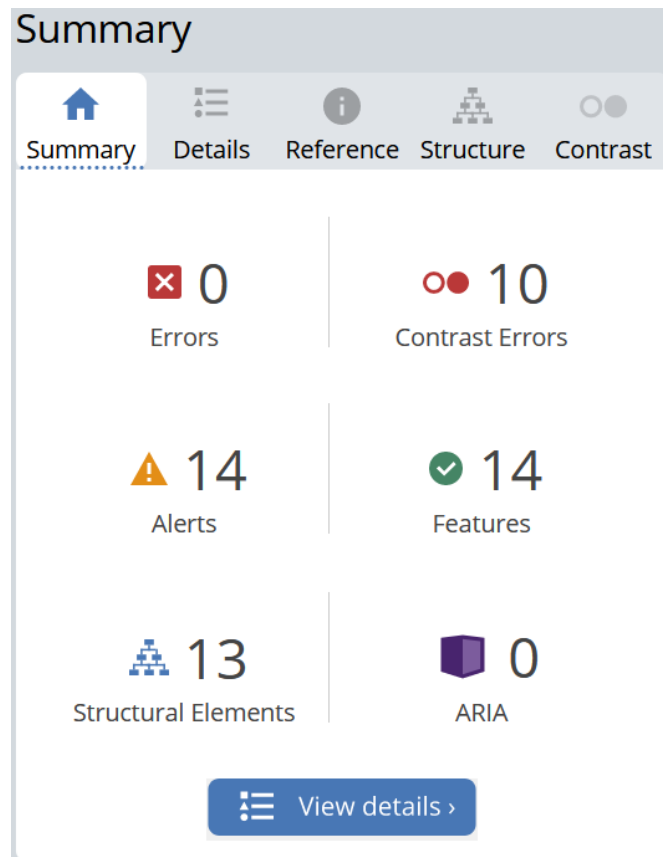


Abbildung 119: Screenshot Snippet Zusammenfassung WAVE Analyse

In der generierten Zusammenfassung werden sowohl bestehende Probleme und mögliche problematische Bereiche als auch Bereiche, die bereits den Web Content Accessibility Guidelines WCAG entsprechen gelistet. Nach dem Einsatz von WAVE zeigt sich, dass die Seiten des Webauftritts bereits einige WCAG-Richtlinien erfüllen es jedoch noch zu Problemen bei Farbkontrast für Text kommt.

Durch die automatisierte Generierung des Testberichtes sollte jedoch auch immer ein Blick auf die Warnungen geworfen werden und nicht einfach annehmen die Seite sei vollkommen barrierefrei.



Weiters wurde die Bedienung der Webseite für sehingeschränkte Benutzer mit dem NVDA Screen Reader simuliert, wobei nicht vergessen werden sollte den Bildschirm abzudecken, um akkurate Testergebnisse zu erlangen.

Screen Reader erfüllen zwei Aufgaben für die sehingeschränkte Benutzergruppe, das Vorlesen von Informationen auf der Seite und das Navigieren durch die Seite. Tests mit dem Screen Reader zeigen, dass es zu Problemen beim Ausfüllen des Formulars kommen kann, wenn inkorrekte Eingaben erfolgen, da Fehlermeldungen nicht immer vorgelesen werden. Ein Weiterer Problembereich stellt auch die Darstellung der Kundenfrequenz mit Highcharts dar. Durch die Verwendung des Highcharts eigene Accessibility Moduls wird die Grafik in Englisch vorgelesen.

Es zeigt sich auch das Screen Reader Tests ohne Testpersonen, welche einen Screen Reader täglich benutzen, eine wirkliche Feststellung der Barrierefreiheit der Seiten fast unmöglich macht.

8 Schlusswort

Nach Abschluss der Arbeit zeigt sich, dass es möglich ist mit einem Raspberry Pi 4 und Pi Camera V2 eine kostengünstige Version eines Messers für die Kundenfrequenz zu bauen.

Jedoch ist dieser für den Einsatz in der Praxis untauglich, da die Kamera zu wenig Leistung aufweist und schlechte Lichtverhältnisse die Zählung sehr stark beeinflusst. Der Grund für diese Anfälligkeit auf unterschiedliche Lichtverhältnisse liegt am Design der Pi Camera und der Größe des Sensors sowie der fixen Linse. Durch diese Problematik wird ein akkurates und vor allem zu verlässliches verfolgen von Personen, welche das Geschäft betreten, unmöglich.

Ein Hoffnungsschimmer für eine noch relativ kostengünstige Lösung, verglichen mit derzeitigen noch weitaus teureren Lösungen, könnte die erst kürzlich vorgestellte Raspberry Pi High Quality Kamera darstellen.



Abbildung 120: Raspberry Pi High Quality Camera mit Objektiv, Quelle: [raspberrypi.org](https://www.raspberrypi.org)

Im Vergleich zu der Pi Camera V2 bietet diese einen größeren und daher stärkeren Sensor sowie tauschbare 6 und 16mm Linsen. Für die Zählung von Kunden könnte die 6mm Linse mit Weidwinkel und einstellbaren Fokusring eine geeignete Lösung sein, da sie einen breiten Bereich abdecken. Die High Quality Kamera so gut wie jeden Raspberry Pi unterstützt. Eine Ausnahme bilden die Raspberry Pi Zero Modelle diese benötigen einen FPC Adapter.

9 Abbildungsverzeichnis

Abbildung 1: Vereinfachter Produktentwurf	16 -
Abbildung 2: Raspberry Pi Logo, Quelle: wikipedia.org	19 -
Abbildung 3: Raspberry Pi Camera	19 -
Abbildung 4: OKdo Logo	19 -
Abbildung 5: Ansmann Logo.....	19 -
Abbildung 6: Acer TravelMate TMX349-M-3373	20 -
Abbildung 7: Lenovo IdeaPad G50	21 -
Abbildung 8: DELL XPS 13 9360	22 -
Abbildung 9: OnePlus 5T	23 -
Abbildung 10: Ipad Air 2	23 -
Abbildung 11: Adobe Photoshop Logo, Quelle: adobe.com	24 -
Abbildung 12: Adobe Illustrator Logo, Quelle: adobe.com	24 -
Abbildung 13: Adobe Photoshop Logo, Quelle: adobe.com	25 -
Abbildung 14: Visual Studio Code Logo, Quelle: visualstudio.com	25 -
Abbildung 15: Thonny Logo, Quelle: thonny.org	25 -
Abbildung 16: MySQL Logo, Quelle: wikipedia.org.....	26 -
Abbildung 17: phpMyAdmin, Quelle: wikipedia.org.....	26 -
Abbildung 18: XAMPP Logo, Quelle: apacheefriends.org	26 -
Abbildung 19: Putty Logo, Quelle: wikipedia.org	27 -
Abbildung 20: Filezilla Logo, Quelle: wikipedia.prg	27 -
Abbildung 21: Git, Github Logos, Quelle: wikipadia.org	27 -
Abbildung 22: Raspberry Pi 4, Quelle: winfuture.de	30 -
Abbildung 23: OKdo Raspberry Pi 4 Power-Hat Case, Quelle: okdo.com.....	31 -
Abbildung 24: Raspberry Pi Camera Module V2, Quelle: raspberrypi.org	32 -
Abbildung 25: Asmann Powerbank 10.8, Wuelle: amazon.de.....	32 -

Abbildung 26: Hardware Komponenten	- 34 -
Abbildung 27: Raspberry Pi 4 CSI-Schnittstelle.....	- 35 -
Abbildung 28: Angeschlossenes Pi Kamera Modul	- 35 -
Abbildung 29: Raspberry Pi 4 GPIO Pins	- 36 -
Abbildung 30: HAT-Platine	- 37 -
Abbildung 31: Zusammengebaute Komponenten	- 37 -
Abbildung 32: Zusammengebaute Komponenten mit Case	- 38 -
Abbildung 33: Netzwerkeinstellungen ändern	- 39 -
Abbildung 34: Netzwerkverbindungen	- 40 -
Abbildung 35: Eigenschaften von WLAN.....	- 40 -
Abbildung 36: SD Card Formatter Verknüpfung	- 41 -
Abbildung 37: Debian OS Download Seite, Quelle: raspberrypi.org	- 42 -
Abbildung 38: balenaEtcher - Image to USB & SD Karte, Quelle: chip.de	- 42 -
Abbildung 39: Strompi V3 Battery-HAT	- 47 -
Abbildung 40: Angeschlossener Battery-HAT	- 48 -
Abbildung 42: Datenbankstruktur Kundenfrequenzmesser	- 51 -
Abbildung 43: Beispielhafte Kundenzählung	- 54 -
Abbildung 44: Beispielhafte Anpassungsfähigkeit des Begrenzungsrahmens	- 55 -
Abbildung 45: Python Logo, Quelle: wikipedia.org.....	- 55 -
Abbildung 46: OpenCV Logo, Quelle: wikipedia.com	- 56 -
Abbildung 47: PHP Logo, Quelle: wikipedia.org.....	- 57 -
Abbildung 48: Verzeichnisstruktur Raspberry Pi	- 60 -
Abbildung 49: Befehl zum Abrufen der Crontab-Tabelle	- 61 -
Abbildung 50: Cronjobs aus crontab.....	- 62 -
Abbildung 51: Code Snippet openScript.sh.....	- 63 -
Abbildung 52: Code Snippet scrip.py 1	- 64 -

Abbildung 53: Code Snippet script.py 2	- 64 -
Abbildung 54: Code Snippet script.py 3	- 65 -
Abbildung 55: Code Snippet script.py 4	- 66 -
Abbildung 56: Code Snippet insertBackupClearFileWithCounter.py 1	- 67 -
Abbildung 57: Code Snippet insertBackupClearFileWithCounter.py 2	- 68 -
Abbildung 58: Datensätze der Tabelle daten	- 69 -
Abbildung 59: Code Snippet insertBackupClearFileWithCounter.py 3	- 70 -
Abbildung 60: Code Snippet sendInsert.php	- 71 -
Abbildung 61: Code Snippet sendUpdate.py	- 72 -
Abbildung 62: Code Snippet sendUpdate.php	- 73 -
Abbildung 63: App Design Mockup	- 76 -
Abbildung 64: Java Logo, Quelle: wikipedia.org	- 77 -
Abbildung 65: JDBC Abbildung, Quelle: oracle.com	- 77 -
Abbildung 66: MPAndroidCharts Logo, Quelle: github.com	- 77 -
Abbildung 67: AsyncTask Sequenzdiagramm, Quelle: corochann.com	- 81 -
Abbildung 68: Code Snippet LoginActivity.java. 1	- 81 -
Abbildung 69: Code Snippet LoginActivity.java. 2	- 82 -
Abbildung 70: Code Snippet LoginActivity.java. 3	- 83 -
Abbildung 71: Code Snippet MainActivity.java. 1	- 84 -
Abbildung 72: Code Snippet MainActivity.java. 2	- 84 -
Abbildung 73: Code Snippet MainActivity.java. 3	- 85 -
Abbildung 74: Code Snippet MainActivity.java. 4	- 86 -
Abbildung 75: gekürztes Code Snippet MainActivity.java. 1	- 87 -
Abbildung 76: gekürztes Code Snippet MainActivity.java. 2	- 87 -
Abbildung 77: App Screenshot DatePicker	- 88 -
Abbildung 78: App Screenshot Toolbar	- 88 -

Abbildung 79: HTML Logo, Quelle: wikipedia.org	- 93 -
Abbildung 80: HTML Element Aufbau.....	- 94 -
Abbildung 81: CSS Logo, Quelle: wikipedia.org	- 94 -
Abbildung 82: JavaScript Logo, Quelle: wikipedia.org	- 95 -
Abbildung 83: Highcharts Logo, Quelle: Highsoft.com	- 95 -
Abbildung 84: AJAX Logo, Quelle: dvglogo.com	- 96 -
Abbildung 85: XAMPP Logo, Quelle: apacheefriends.org	- 96 -
Abbildung 86: PHP Logo, Quelle: wikipedia.org.....	- 97 -
Abbildung 87: WAVE Logo, Quelle: webaim.corg.....	- 98 -
Abbildung 88: NVDA Logo, Quelle: nvaccess.org	- 98 -
Abbildung 89: Grundstruktur HTML Dokumente	- 100 -
Abbildung 90: Metainformationen im <head> Element	- 100 -
Abbildung 91: Accessibility Tree	- 101 -
Abbildung 92: Screen Snippet Downloadbuttons Landing Page	- 102 -
Abbildung 93: Code Snippet Element.....	- 103 -
Abbildung 94: Screen Snippet Login Formular ohne Fehlermeldung.....	- 104 -
Abbildung 95: Screen Snippet Login Formular mit Fehlermeldung	- 104 -
Abbildung 96: Code Snippet <lable> und <input> Elemente	- 104 -
Abbildung 97: Beispielhafte Grid-Abbildung.....	- 105 -
Abbildung 98: Code Snippet CSS Grid	- 106 -
Abbildung 99: Code Snippet Zuordnung der Interface Element im Grid 1	- 106 -
Abbildung 100: Code Snippet Zuordnung der Interface Element im Grid 2	- 107 -
Abbildung 101: Code Snippet Zuordnung der Interface Element im Grid 3	- 107 -
Abbildung 102: Code Snippet Media Query	- 108 -
Abbildung 103: Flexitems auf Landing Page und Dashboard	- 109 -
Abbildung 104: Code Snippet Flexbox	- 109 -

Abbildung 105: Code Snippet Custom Properties 1.....	- 110 -
Abbildung 106: Code Snippet Custom Properties 2.....	- 111 -
Abbildung 107: Diagramm der Kundenfrequenz am Dashboard	- 112 -
Abbildung 108: Code Snippet Highcharts Module	- 113 -
Abbildung 109: Code Snippet Highcharts Methoden	- 113 -
Abbildung 110: Code Snippet Highcharts Optionen	- 114 -
Abbildung 111: Code Snippet AJAX Request.....	- 115 -
Abbildung 112: Screenshot Snippet Userinterface Buttons Dashboard	- 116 -
Abbildung 113: Code Snippet Button Elemente.....	- 116 -
Abbildung 114: Code Snippet Zuweisung Buttons	- 116 -
Abbildung 115: Code Snippet Login	- 118 -
Abbildung 116: Code Snippet Session Start	- 119 -
Abbildung 117: Code Snippet Session End	- 120 -
Abbildung 118: Code Snippet Datenbankverbindung	- 121 -
Abbildung 119: Code Snippet SQL-Abfrage	- 123 -
Abbildung 120: Screenshot Snippet Zusammenfassung WAVE Analyse	- 127 -
Abbildung 121: Raspberry Pi High Quality Camera mit Objektiv, Quelle: raspberrypi.org .-	129 -



10 Literaturverzeichnis

XAMPP Produktseite (2020):

<https://www.apachefriends.org/index.html> - Download vom: 06.05.2020

FileZilla Produktseite (2020):

<https://filezilla-project.org> - Download vom: 06.05.2020

Visual Studio Code Produktseite (2020):

<https://code.visualstudio.com/> - Download vom 06.05.2020

Highcharts Produktseite:

<https://www.highcharts.com/blog/products/highcharts> - Download vom: 06.05.2020

Mozilla Developer Network web docs (2020): XML

<https://developer.mozilla.org/en-US/docs/Glossary/XML> - Download vom: 06.05.2020

Mozilla Developer Network web docs (2020): Ajax

<https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX> - Download vom: 06.05.2020

MySQL Manual (2020): Data Definition Language

https://dev.mysql.com/doc/refman/8.0/en/glossary.html#glos_ddl - Download vom: 11.05.2020

MySQL Manual (2020): Data Manipulation Language

https://dev.mysql.com/doc/refman/8.0/en/glossary.html#glos_dml - Download vom: 11.05.2020

MySQL Manual (2020): Data Control Language

https://dev.mysql.com/doc/refman/8.0/en/glossary.html#glos_dcl - Download vom: 11.05.2020

Mozilla Developer Network web docs (2020): What is Accessibility?

https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility - Download vom: 11.05.2020

Web Hypertext Application Technology Working Group (2020): HTML Semantics

<https://html.spec.whatwg.org/dev/dom.html#semantics-2> - Download vom: 23.05.2020

W3C (2020): Evaluating Web Accessibility

<https://www.w3.org/WAI/test-evaluate/> - Download am: 03.06.2020

WAVE Produktseite (2020): WAVE web accessibility evaluation tool

<https://wave.webaim.org/> - Download vom: 03.06.2020

Stanford Online Accessibility Program (2020): Screen Reader

<https://soap.stanford.edu/tips/screen-reader-testing> - Download vom: 03.06.2020

NV Access (2020): NVDA Screen Reader

<https://www.nvaccess.org/> - Download vom: 03.06.2020

World Wide Web Consortium (2020): HTML Sectioning Elements

<https://w3c.github.io/aria-practices/examples/landmarks/HTML5.html> - Download vom: 14.06.2020

WebAIM (2020): Visual Disabilities

<https://webaim.org/articles/visual/blind> - Download vom: 15.06.2020

Mozilla Developer Network web docs (2020): Accessibility Tree

<https://developer.mozilla.org/en-US/docs/Glossary/AOM> - Download vom: 15.06.2020

Mozilla Developer Network web docs (2020): Basic concepts of grid

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout -

Download vom: 11.05.2020

Mozilla Developer Network web docs (2020): Basic concepts of flexbox

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox -

Download vom: 11.05.2020

Mozilla Developer Network web docs (2020): XSS Angriffe

https://developer.mozilla.org/en-US/docs/Web/Security/Types_of_attacks#Cross-site_scripting_XSS -

Download vom: 15.06.2020

W3schools (2020): XSS Angriffe

https://www.w3schools.com/php/php_form_validation.asp - Download vom: 15.06.2020

PHP Manual (2020): Manual

<https://www.php.net/manual/en/index.php> - Download vom: 17.06.2020

Tutorial Republic (2020): Anhaltspunkte für das Login System

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php> - Download vom: 10.06.2020

Notebookcheck.com (2020): Test Acer TravelMate X349 Notebook

<https://www.notebookcheck.com/Test-Acer-TravelMate-X349-G2-Laptop.212722.0.html> - Download vom
12.04.2020

Notebookcheck.com (2020): Test Lenovo G50-80 Notebook

<https://www.notebookcheck.com/Test-Lenovo-G50-80-Notebook.158173.0.html> - Download vom 12.04.2020

Notebookcheck.com (2020): Test Dell XPS 13 9360 FHD i5 Notebook

<https://www.notebookcheck.com/Test-Acer-TravelMate-X349-G2-Laptop.212722.0.html> - Download vom
12.04.2020

Notebookcheck.com (2020): Test OnePlus 5T Smartphone

<https://www.notebookcheck.com/Test-Apple-iPad-Air-2-A1567-128-GB-LTE-Tablet.129027.0.html> - Download
vom 12.04.2020

Notebookcheck.com (2020): Test Apple iPad Air 2 Tablet

<https://www.notebookcheck.com/Test-Acer-TravelMate-X349-G2-Laptop.212722.0.html> - Download vom
12.04.2020

Raspberry Pi Produktseite (2020): Spezifikationen Raspberry Pi

<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/> - Download vom: 13.05.2020

Raspberry Pi Produktseite (2020): Spezifikationen Camera

<https://www.raspberrypi.org/products/camera-module-v2/> - Download vom: 13.05.2020



Testberichte.de (2020): Ansmann AG PowerBank 10.8 Datenblatt

<https://www.testberichte.de/p/ansmann-tests/powerbank-10-8-testbericht.html#produkt-datenblatt> –

Download vom 13.05.2020

Adobe Produktseite (2020): Photoshop

<https://www.adobe.com/at/products/photoshop.html> – Download vom: 12.04.2020

Wikipedia (2020): Photoshop

https://de.wikipedia.org/wiki/Adobe_Photoshop – Download vom: 12.04.2020

Adobe Produktseite (2020): Illustrator

<https://www.adobe.com/at/products/illustrator.html> – Download vom: 12.04.2020

Wikipedia (2020): Illustrator

https://de.wikipedia.org/wiki/Adobe_Illustrator – Download vom: 12.04.2020

Adobe Produktseite (2020): Xd

<https://www.adobe.com/at/products/xd.html> – Download vom: 12.04.2020

Wikipedia (2020): Xd

https://de.wikipedia.org/wiki/Adobe_XD – Download vom: 12.04.2020

Thonny Produktseite (2020): Thonny

<https://thonny.org/> – Download vom: 12.04.2020

Wikipedia (2020): Thonny

<https://en.wikipedia.org/wiki/Thonny> – Download vom 12.04.2020

MySQL Produktseite (2020): MySQL

<https://www.mysql.com/de/> – Download vom: 12.04.2020

Wikipedia (2020): MySQL

<https://de.wikipedia.org/wiki/MySQL> – Download vom 12.04.2020

MySQLTutorial (2020): MySQL Tutorial

<https://www.mysqltutorial.org/> – Download vom 12.04.2020

w3schools (2020): SQL Tutorial

<https://www.w3schools.com/sql/default.asp> – Download vom 12.04.2020

PhpMyAdmin Produktseite (2020): phpMyAdmin

<https://www.phpmyadmin.net/> – Download vom: 12.04.2020

w3schools (2020): PHP Tutorial

<https://www.w3schools.com/php/default.asp> – Download 12.04.2020

Wikipedia (2020): phpMyAdmin

<https://de.wikipedia.org/wiki/PhpMyAdmin> – Download vom: 12.04.2020

Putty Produktseite (2020): Putty

<https://www.putty.org/> – Download vom 12.04.2020



Wikipadia (2020): Putty

<https://de.wikipedia.org/wiki/PuTTY> – Download vom 12.04.2020

GitHub Produktseite (2020): GitHub

<https://github.com/> – Download vom 12.04.2020

Wikipedia (2020): GitHub

<https://de.wikipedia.org/wiki/GitHub> – Download vom 12.04.2020

Raspberry Pi Produktseite (2020)

<https://www.raspberrypi.org/> – Download vom 12.04.2020

Wikipadia (2020): Raspberry Pi 4

https://de.wikipedia.org/wiki/Raspberry_Pi – Download vom 12.04.2020

Python Produktseite (2020): Python

<https://www.python.org/> – Download vom 12.04.2020

Wikipadia (2020): Python

[https://de.wikipedia.org/wiki/Python_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Python_(Programmiersprache)) – Download vom 12.04.2020

OpenCV Produktseite (2020): OpenCV

<https://opencv.org/> – Download vom 12.04.2020

codecentric.de (2020): Einführung in Computer Vision

<https://blog.codecentric.de/2017/06/einfuehrung-in-computer-vision-mit-opencv-und-python/> – Download vom 12.04.2020

Youtube (2020): Einführung in OpenCV

https://www.youtube.com/watch?v=Q_2tbDCJTnU – Download vom 12.04.2020

Wikipedia (2020): OpenCV

<https://de.wikipedia.org/wiki/OpenCV> – Download vom 12.04.2020

Geeksforgeeks (2020): Virtual Enviroment

<https://www.geeksforgeeks.org/python-virtual-environment> – Download 12.04.2020

Python Documentation (2020): venv

<https://docs.python.org/3/library/venv.html> – Download vom 12.04.2020

Tutorialspoint (2020): Python Tutorial

<https://www.tutorialspoint.com/python/index.htm> – Download vom 12.04.2020

Stackoverflow (2020): Troubleshooting Programmieren

<https://stackoverflow.com/> – Download vom 12.04.2020

Linux-für-Neulinge (2020): Cron

<http://www.linux-neuling.de/anleitungen/cron.html> – Download vom 12.04.2020

Wikipadia (2020): Cron

<https://de.wikipedia.org/wiki/Cron> – Download vom 12.04.2020



medium.com (2020): How to Schedule a Python Script Cron Job

<https://medium.com/@gavinwiener/how-to-schedule-a-python-script-cron-job-dea6cbf69f4e> – Download

12.04.2020

GitHub (2020): Haarcascades

<https://github.com/opencv/opencv/tree/master/data/haarcascades> – Download vom 12.04.2020

Willberger (2020): Haarcascade

<http://www.willberger.org/cascade-haar-explained/> – Download vom 12.04.2020

Pyimagesearch (2020): Install OpenCV 4 on Raspberry Pi 4

<https://www.pyimagesearch.com/2019/09/16/install-opencv-4-on-raspberry-pi-4-and-raspbian-buster/> -

Download vom 12.04.2020

Pyimagesearch (2020): Gesichtverfolgung Quellcode

<https://www.pyimagesearch.com/2018/08/15/simple-face-tracking-with-opencv/> - Download vom 12.04.2020

Pyimagesearch (2020): OpenCV People Counter

<https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/> - Download vom 12.03.2020

Pyimagesearch (2020): Simple object tracking with OpenCV

<https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/> - Download vom

12.03.2020

w3schools (2020): Python Tutorial

<https://www.w3schools.com/python/default.asp> – Download vom 14.04.2020

Udemy (2020): Python for Computer Vision with OpenCV and Deep Learning (kostenpflichtig)

<https://www.udemy.com/course/python-for-computer-vision-with-opencv-and-deep-learning/> – Download vom

12.04.2020

Udemy (2020): Python für Computer Vision und Data Science mit OpenCV (kostenpflichtig)

<https://www.udemy.com/course/computer-vision-mit-opencv-und-deep-learning/> – Download vom 12.04.2020

Udemy (2020): Python – Grundlagen Bootcamp (kostenpflichtig)

<https://www.udemy.com/course/originale-python-bootcamp/> – Download vom 12.04.2020

Udemy (2020): Learn Python Face Detection (kostenpflichtig)

<https://www.udemy.com/course/learn-python-face-detection/> – Download vom 12.04.2020

Udemy (2020): Learn Computer Vision with OpenCV and Python (kostenpflichtig)

<https://www.udemy.com/course/learn-computer-vision-with-opencv-and-python/> – Download vom 12.04.2020

Udemy (2020): Python Programming Bible (kostenpflichtig)

<https://www.udemy.com/course/pythonv3/> - Download vom 12.04.2020

Udemy (2020): Java Programmieren – Der umfassende Einsteigerkurs

<https://www.udemy.com/course/java-programmieren-der-umfassende-einsteigerkurs/> - Download vom

12.04.2020

Udemy (2020): Java Programmieren – Der Fortgeschrittenen Kurs

<https://www.udemy.com/course/java-programmieren-der-fortgeschrittenen-kurs/> - Download vom 12.04.2020



Raspberrypi (2020): New aspberry Pi High Quality Camera

<https://www.raspberrypi.org/blog/new-product-raspberry-pi-high-quality-camera-on-sale-now-at-50/> -

Download vom 12.04.2020

Developers Android (2020): Android Studio IDE

<https://developer.android.com/studio> - Download vom 12.04.2020

Developers Android (2020): Documentation for app developers

<https://developer.android.com/docs> - Download vom 12.04.2020

Oracle Java (2020): Java Platform Documentation

<https://docs.oracle.com/javase/8/docs/api/> - Download vom 12.04.2020

Java Programmieren (2020): Java Programmieren lernen

<http://www.java-programmieren.com/> - Download vom 12.04.2020

w3schools (2020): Java Tutorial

<https://www.w3schools.com/java/default.asp> - Download vom 12.04.2020

Youtube (2020): Android Studio Tutorial

https://www.youtube.com/playlist?list=PLGI5h6V3K_kAPMv8ty-1YJzPbRucnjCE9 - Download vom 12.04.2020

Youtube (2020): MPAndroidCharts Tutorial

<https://www.youtube.com/playlist?list=PLc2rvfiptPSR7NliWRLwweX55XwrsrTOO> - Download vom 12.04.2020

w3schools (2020): XML Tutorial

<https://www.w3schools.com/xml/default.asp> - Download vom 12.04.2020

Jon Duckett, JavaScript & JQuery, Wiley, 2014, ISBN 978-1-118-53164-8

Ethan Marcotte, Responsive Web Design, A Book Apart, Second Edition, 2014, ISBN 9781937557188

Karen McGrane, Going Responsive, A Book Apart, 2015, ISBN 9781937557300

David Demaree, Git for Humans, A Book Apart, 2016, ISBN 9781937557386

Eben Upton und Gareth Halfacree, Raspberry Pi User Guide, Wiley, 4th Edition ,2016, ISBN 978-1-119-26436-1

Tanja Möller, Gesichtsdetektion mit OpenCV zur Erkennung eines Gesprächspartners: Robustheitsanalyse und Implementierung, AV Akademikerverlag, 2016, ISBN 9783330507197