

DIPLOMARBEIT

Advanced Parking Monitoring (APM)



ADVANCED PARKING MONITORING

Philipp Kraft, Dennis Köb und Samuel Bleiner

Dipl.-Ing. Christoph Stüttler

29. März 2021, Rankweil

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig 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 erkenntlich gemacht habe.

Rankweil, 29. März 2021

Philipp Kraft

Dennis Köb

Samuel Bleiner

Kurzfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Vorwort

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Danksagung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Ein lesenswertes Buch ist »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«¹

¹Albert Einstein. »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«. In: *Annalen der Physik* 322.10 (1905), S. 891–921. doi: <http://dx.doi.org/10.1002/andp.19053221004>.

Inhaltsverzeichnis

Eidesstattliche Erklärung	ii
Kurzfassung	iii
Abstract	iv
Vorwort	v
Danksagung	vi
1 Projektteam	8
2 Projektbetreuer	9
3 Auftragnehmer	10
4 Projektplanung	11
5 Rechtliches	12
6 Einleitung	13
7 Projektantrag	15
8 Kennzeichenerkennung	16
8.1 Anforderungen	16
8.2 Vorstudie	17
8.3 Bildverarbeitung	19
8.3.1 Einleitung	19
8.3.2 Bilaterale Filterung	19
8.3.3 Thresholding	20

8.3.4	Erosion	21
8.3.5	Farbraum	22
8.3.5.1	RGB	22
8.3.5.2	Graustufen	22
8.3.5.3	BGR	23
8.3.6	Konturerkennung	23
8.4	Kennzeichenerkennungsprogramm	24
8.4.1	Einleitung	24
8.4.2	Programmiersprache	24
8.4.3	Konzept	24
8.4.3.1	Bildaufnahme	25
8.4.3.2	Kennzeichenerfassung	25
8.4.3.3	Kennzeichensegmentierung	25
8.4.3.4	Zeichenerkennung	26
8.4.3.5	Anbindung an Datenbank	26
8.4.3.6	Ablauf	26
8.5	Raspberry Pi	27
8.5.1	Einleitung	27
8.5.2	Wahl des Raspberry Pi	28
8.5.3	Kamera	28
9	Fahrzeugerkennung	30
9.1	Anforderungen	30
9.2	Vorstudie	30
9.3	Erkennung von Metallen über Spulen	30
9.3.1	Messung der ferromagnetischer Metalle	34
9.3.1.1	RL-Oszillator mit Timer Baustein	34
9.3.1.2	Puls-Ladung mit Mikrokontroller	34

9.3.2	Messung paramagnetischer Metalle	34
9.3.2.1	Messung eines Wechselstromes	34
9.3.2.2	Messung der Resonanzfrequenz eines Oszillators	34
9.4	RS485 Bussystem	34
9.4.1	Überblick	34
9.4.2	Elektrische Spezifikation	34
9.4.3	Implementation eines eigenen Protokolls	34
9.5	Mikrokontroller Slave-Geräte	34
9.5.1	Überblick	34
9.5.2	Atmega328PB	34
9.5.3	Peripherie des Mikrokontrollers	34
9.5.3.1	Spannungswandler	34
9.5.3.2	RS485 Pegelwandler	35
9.5.3.3	Digitale Ein- und Ausgänge	35
9.5.4	Layout des Slave-Gerätes	35
9.5.5	Gehäuse	35
9.6	USB-Master	35
9.6.1	USB-Bussadapter Gerät	35
9.6.1.1	Überblick	35
9.6.1.2	FT232RL	35
9.6.1.3	Spannungsversorgung	35
9.6.1.4	USB-C Anschluss	35
9.6.1.5	Layout des Master-Geräts	35
9.6.1.6	Gehäuse	35
9.6.2	Master Programm	35
9.6.2.1	Benötigte Software	35
9.6.2.2	Adressvergabe	35

9.6.2.3	Frequenzauslesung	35
9.6.2.4	Auswertung	36
9.6.2.5	API-Post	36
9.6.3	RaspberryPi als Mastergerät	36
9.6.3.1	SSH Remote Zugriff	36
9.6.3.2	Code Deployment	36
9.6.3.3	Unittest	37
10	Webinterface	37
10.1	Einleitung	37
10.2	Anforderungen	37
10.3	Verwendete Technologien	37
10.3.1	HTML	37
10.3.1.1	Beispielhafte HTML Seite	38
10.3.2	CSS	39
10.3.3	JavaScript	41
10.3.4	PHP	42
10.3.5	TailwindCSS	43
10.3.6	Vue	43
10.3.7	Laravel	43
10.3.7.1	Routing	43
10.3.7.2	Blade Templates	43
10.3.7.3	Sessions	43
10.3.7.4	Controllers	43
10.3.7.5	Artisan	43
10.3.7.6	Migrations	43
10.3.7.7	Middleware	43
10.3.7.8	Eloquent ORM	43

10.3.7.9	Laravel Mix	43
10.3.7.10	Laravel Sanctum	43
10.4	Lokale Entwicklungsumgebung mit Laragon	43
10.4.1	Benötigte Software	44
10.4.2	Konfiguration von PHP	44
10.4.3	Installation von phpMyAdmin	47
10.5	Lokale Entwicklungsumgebung mit WSL und Docker	48
10.5.1	Benötigte Software	48
10.5.2	Installation von WSL	49
10.5.2.1	2. Schritt: Virtual Machine Aktivieren	49
10.5.2.2	3. Schritt: Linux Kernel Update	49
10.5.2.3	4. Schritt: WSL 2	49
10.5.2.4	5. Schritt: Linux Distribution herunterladen	50
10.5.3	Installation von Docker	50
10.6	Production Server	51
10.6.1	Benötigte Software	51
10.6.2	Installation des LAMP Stacks	52
10.6.2.1	Apache	52
10.6.2.2	PHP	53
10.6.2.3	MariaDB	53
10.6.2.4	phpMyAdmin	54
10.6.2.5	Webinterface Virtual Host	58
10.6.2.6	Installation von Composer	60
10.6.3	Deployment mit Github Actions	60
10.6.3.1	Git Setup	60
10.6.3.2	Deploy Script	61
10.6.3.3	Server Deploy Script	62

10.6.3.4 Github Action	63
10.7 Grundlegender Aufbau	66
10.7.1 Components	66
10.7.1.1 Anonymous Components	66
10.7.1.2 Components erstellen	67
10.7.1.3 Components verwenden	67
10.7.1.4 Attribute übergeben	68
10.7.2 Layouts	68
10.7.3 Navbar	73
10.7.4 Sidebar	73
10.7.5 Footer	73
10.8 Funktionen	73
10.8.1 Dashboard	73
10.8.2 Login- und Registersystem	73
10.8.3 News	73
10.8.4 Benutzerverwaltung	73
10.8.5 Rechte- und Rollenverwaltung	73
10.8.6 Parkplatzverwaltung	73
10.8.7 Kennzeichenverwaltung	73
10.8.8 Erkennungsverlauf	73
10.8.9 Seiten Einstellungen	73
10.8.10API Schlüssel	73
10.8.11Displays	73
10.8.12Notifications	73
10.8.13Profil	73
10.8.14Lokalisierung	73
10.9 Performance und Sicherheit	73

10.9.1 Form Validation	73
10.9.2 Authentifizierung	73
10.9.3 Authentisierung	73
10.9.4 SQL Injection	73
10.9.5 Cross-Site-Scripting	73
10.9.6 Lighthouse	73
11 Zusammenfassung und Ausblick	74
12 Anhang	76
Abbildungsverzeichnis	77
Tabellenverzeichnis	78
Codeverzeichnis	79
Abkürzungsverzeichnis	81
Literaturverzeichnis	82

1 Projektteam

2 Projektbetreuer

3 Auftragnehmer

4 Projektplanung

5 Rechtliches

6 Einleitung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu

enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

7 Projektantrag

8 Kennzeichenerkennung

8.1 Anforderungen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus

vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

8.2 Vorstudie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel

leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper,

leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

8.3 Bildverarbeitung

8.3.1 Einleitung

Die Bildverarbeitung ist ein zentrales Thema in dieser Applikation für Kennzeichenerkennung. Sie wird für die Zeichensegmentierung verwendet, sowie für die Vorbereitung von Bildern für andere Algorithmen. Im Folgenden werden die verwendeten Bildverarbeitungsfunktionen aufgelistet und deren Funktionsweise erläutert.

8.3.2 Bilaterale Filterung

Bilaterale Filterung ist eine Methode für eine kantenerhaltende Weichzeichnung eines Bildes.

Bei der Berechnung für den Farbwert des Ausgabepixels werden die benachbarten Pixel nicht nur mit ihrer Entfernung gewichtet, sondern auch mit ihrem eigenen Farbwert. Dadurch können einzelne farbliche Ausreißer herausgefiltert werden. Dies ist vor allem in der Bildverarbeitung wichtig, da dadurch die wichtigen Eigenschaften eines Bildes, wie zum Beispiel Kanten, erhalten bleiben und verarbeitet werden können, aber einzelne abweichende Pixel herausgefiltert werden wodurch unnötige Informationen

entfernt werden.

In Abbildung 1 kann man ein Bild von verschiedenen Lebensmitteln sehen. Wenn man genau hinsieht erkennt man vor allem bei den Blättern im Hintergrund und beim Brot viele detailreiche Texturen. Diese Texturen haben keine wichtige Texturen und sind deswegen unnötig. Um die Bildverarbeitung zu vereinfachen wendet man deswegen die bilaterale Filterung auf dieses Bild an, um diese detailreichen Texturen zu vereinfachen. In Abbildung 2 sieht man das Bild nach der bilateralen Filterung. Wenn man hier dann wieder genauer auf die Blätter und das Brot sieht, erkennt man, dass die detailreichen Texturen weichgezeichnet wurden, aber die Kanten sind genauso gut erkennbar wie vor der Filterung.

8.3.3 Thresholding

Das Thresholding oder auch Schwellenwertverfahren wird in der Bildverarbeitung verwendet, um Bilder zu segmentieren. Aus einem Graubild kann dadurch ein Binäres Bild erzeugt werden.

Bei diesem Verfahren wird ein bestimmter Schwellwert (En.: Threshold) definiert, welcher mit den Grauwerten der einzelnen Pixel des Bildes verglichen wird. Wenn der Grauwert den Schwellwert überschreitet, wird dieser durch einen weißen Pixel ersetzt und wenn der Grauwert kleiner als der Schwellwert ist, wird dieser durch einen schwarzen Pixel ersetzt. Dadurch erhält man ein Bild welches nur noch zwei Farben hat, Schwarz und Weiß. Dies wird deswegen eingesetzt, da dadurch viele Bildverarbeitungsalgorithmen schneller arbeiten und die Effizienz gesteigert wird.

In Abbildung 3 sieht man ein solches Graubild welches nur verschiedenen Graustufen aufweist. In Abbildung 4 sieht man das Bild nach dem Thresholding. Hier kann man

nur noch das Boot mit den Menschen erkennen. Dies ist nicht nur für schnellere Bildverarbeitungsalgorithmen wichtig, sondern wird auch zur Objekterkennung in Bildern verwendet.

Um den Schwellwert zu bestimmen kann man diesen entweder variieren bis das gewünschte Ergebnis erscheint oder man verwendet Methoden, welche den Schwellwert automatisch bestimmen. Eine der bekanntesten Methoden zur Schwellwertbestimmung ist die Methode von Otsu, welche mit dem Schwellenwert die Pixel in Vordergrund und Hintergrund unterteilt.

8.3.4 Erosion

Erosion ist eine Funktion der Bildverarbeitung und ist in die morphologische Bildverarbeitung einzuordnen. Diese beschäftigt sich primär mit der Verarbeitung von binären Bildern, welche man nach Thresholding erhält.

Erosion benötigt zwei Eingaben, das binäre Bild und einen Kernel. Der Kernel ist dabei die Angabe, nach welcher die Erosion durchgeführt wird. Der Kernel ist auch eine binäre Struktur, welche über jeden einzelnen Pixel des binären Bildes geschoben wird. Wenn der Kernel komplett mit dem binären Bild übereinstimmt, behält dieser Pixel seinen Wert und ansonsten wird er invertiert. Dabei muss jedoch darauf geachtet werden, dass die Polarität des binären Bildes und des Kernels übereinstimmt, da sonst die Erosion nicht richtig funktioniert. Als Resultat erhält man danach ein deutlicheres Bild bei welchem einzelne Pixelfehler herausgefiltert wurden und die Konturen besser erkennbar sind.

In Abbildung 5 und 6 sieht man die Anwendung der Erosion. Die Konturen der einzelnen Zeichen im Kennzeichen sind in Abbildung 6 nach der Erosion deutlicher erkennbar

als davor.

8.3.5 Farbraum

Der Farbraum eines Bildes enthält alle möglichen Farben eines Farbmodells. Das Farbmodell beschreibt dabei die Parameter, aus welchen die einzelnen Farben gebildet werden. Dies ist in der Bildverarbeitung relevant, da verschiedene Funktionen der Bildverarbeitung, unterschiedliche Farbräume verwenden und dieser deswegen korrekt eingestellt werden muss.

In dieser spezifischen Applikation werden die folgenden Farbräume verwendet:

8.3.5.1 RGB

RGB ist einer der häufigsten und bekanntesten Farbräume. Er basiert auf den drei Grundfarben Rot, Grün und Blau und wird vor allem bei Bildschirmen und in der Fotografie genutzt. Die Farben setzen sich in diesem Modell aus dem jeweiligen Rot-, Grün- und Blauanteil der einzelnen Pixel zusammen.

8.3.5.2 Graustufen

Bei einem Graustufen-Bild, zu sehen in Abbildung 3, hat jeder Pixel einen Wert von 0 bis 255. Diese Werte erstrecken sich also von Schwarz bis Weiß und dazwischen liegen verschiedene Grautöne. Dieser Farbraum wird in der Bildverarbeitung häufig verwendet, da Konturen einfacher erkennbar sind und es nur einen Parameter gibt, welcher verarbeitet werden muss, wodurch die Effizienz diverser Algorithmen gesteigert werden kann. Zudem wird dieser Farbraum auch oft in Verbindung mit Thresholding verwendet.

8.3.5.3 BGR

Der BGR ist ein relativ unbekannter und wenig verwendeter Farbraum, da er sehr ähnlich zum RGB-Farbraum ist. Der einzige Unterschied zwischen diesen beiden liegt in der Anordnung der Parameter. Bei BGR sind die Parameter spiegelverkehrt zu RGB, das heißt es kommt zuerst der Blauanteil, dann der Grünanteil und zum Schluss der Rotanteil. Insgesamt ergibt dies für die einzelnen Pixel zwar die gleichen Farben, aber die Funktionen der Bildverarbeitung müssen trotzdem das Bild im passenden Farbraum erhalten. So verwendet zum Beispiel die Funktion „imread“ von OpenCV den BGR-Farbraum und die Funktion „imshow“ von Matplotlib verwendet den RGB-Farbraum. Wenn man diese Funktionen also nacheinander anwendet, muss dazwischen der Farbraum umgewandelt werden.

8.3.6 Konturerkennung

Die Konturerkennung ist eine wichtige Funktion in der Bildverarbeitung mit welcher Objekte in einem Bild gefunden werden können. In dieser Applikation wird sie für die Zeichensegmentierung eingesetzt.

Die Konturerkennung wird hauptsächlich bei binären Bildern verwendet. Eine Kontur kann dabei wie im Folgenden definiert werden. Man überprüft jeden einzelnen Pixel und sieht nach, ob ein benachbarter Pixel einen anderen Farbwert aufweist. Falls dies zutrifft muss der zu prüfende Pixel zu einer Kontur gehören. Wenn dies auf mehrere zusammenhängende Pixel zutrifft, bedeutet das, dass diese zusammen eine Kontur bilden.

Die Funktion „findcontours“ von OpenCV, welche in dieser Applikation verwendet wird, ist eine Funktion für Konturerkennung und kann weiße Objekte auf einem schwarzen Hintergrund erkennen. Sie basiert auf dem Algorithmus von Suzuki von 1985 und

liefert eine Liste mit allen Konturen. Die Konturen werden in der Liste als ein Array von Koordinaten abgespeichert.

8.4 Kennzeichenerkennungsprogramm

8.4.1 Einleitung

Die Software ist der wichtigste und größte Teil der Kennzeichenerkennung. Sie erhält ein Bild, in welchem ein Auto mit einem Kennzeichen enthalten ist und liefert am Ende dieses Kennzeichen und sendet dieses dann automatisch an die Datenbank. Die Software kann entweder über Bildverarbeitung oder mit Machine Learning Modellen realisiert werden. Der erste Ansatz bei dieser Anwendung war mit klassischer Bildverarbeitung, welche aber nicht die gewünschte Genauigkeit erreicht hat, weswegen dann auf Machine Learning gewechselt wurde.

8.4.2 Programmiersprache

Die verwendete Programmiersprache für die Kennzeichenerkennung ist Python. Python ist eine höhere Programmiersprache, welche übersichtlich und leicht lesbar ist. Sie ist vor allem für Bildverarbeitung und Machine Learning Anwendungen gut geeignet, da es dafür hoch optimierte und effiziente Bibliotheken gibt wie zum Beispiel OpenCV, Numpy und Tensorflow. Dadurch ist Python für diese Anwendung besser geeignet als zum Beispiel C++. Dieses wäre zwar normalerweise effizienter, bietet aber weniger optimierte Bibliotheken in diesem Bereich, wodurch es hier weniger gut geeignet ist.

8.4.3 Konzept

Das Programm für die Kennzeichenerkennung basiert auf fünf Stufen. Die erste Stufe ist die Bildaufnahme, die zweite ist die Kennzeichenerfassung mittels Machine

Learning, die dritte ist die Kennzeichensegmentierung mithilfe von Bildverarbeitung, die vierte ist die Zeichenerkennung mittels Machine Learning und die fünfte ist die Anbindung an die Datenbank.

8.4.3.1 Bildaufnahme

Um ein Bild verarbeiten zu können und aus diesem ein Kennzeichen auslesen zu können, muss zuerst ein Bild vorliegen. Dieses wird über den RaspberryPi mit der RaspberryPi-Kamera aufgenommen. Um das Bild aufzunehmen, muss einfach ein Auslöser aktiviert werden und dann wird das Bild aufgenommen und im richtigen Ordner abgespeichert. Zuvor wird noch überprüft ob sich in diesem Ordner bereits ein Bild befindet und falls eines vorhanden ist wird es gelöscht. Dadurch werden mögliche Fehler durch mehrere Bilder verhindert.

8.4.3.2 Kennzeichenerfassung

Die Kennzeichenerfassung hat die Aufgabe, das Kennzeichen im Eingabebild zu lokalisieren. Dies geschieht mittels Machine Learning mit dem Modul WPOD-NET von Sérgio Montazolli Silva und Cláudio Rosita Jung. Dieses verwendet zuerst das Modul YOLOv2 welches zur Echtzeitobjekterkennung verwendet werden kann und in dieser Anwendung zur Erkennung von Fahrzeugen verwendet wird. Danach werden die Koordinaten des Kennzeichens ermittelt und dieses aus dem Bild ausgeschnitten und abgespeichert.

8.4.3.3 Kennzeichensegmentierung

Die Kennzeichensegmentierung hat das Ziel die einzelnen Zeichen im Kennzeichen zu separieren und so zu vorbereiten, dass die darauffolgende Zeichenerkennung damit arbeiten kann. Dazu wird das Bild mit dem Kennzeichen zuerst in Graustufen konvertiert, dann mit einem bilateralen Filter gefiltert, mit Thresholding in ein binäres Bild umgewandelt und anschließend mittels Erosion besser erkennbar gemacht. Danach

werden im verarbeiteten Bild die Konturen gesucht, sortiert und anhand dieser die einzelnen Zeichen herausgefiltert.

8.4.3.4 Zeichenerkennung

Die letzte Stufe der Kennzeichenerkennung ist die Zeichenerkennung. In dieser werden die einzelnen Zeichen erkannt und als Text abgespeichert. Dies funktioniert über eine eigenes Neuronales Netz basierend auf MobileNetV2, welches mit einem Datensatz von über 35 000 Bildern auf die Erkennung von Zeichen aus Bildern trainiert wurde.

8.4.3.5 Anbindung an Datenbank

Nachdem das Kennzeichen als Text abgespeichert wurde, muss diese Information in die Datenbank übergeben werden. Dazu wird die eigene API angewandt, welcher man diese Informationen übergeben muss und als Rückgabe die Information bekommt, ob sich das Fahrzeug nun innerhalb oder außerhalb des Parkplatzes befindet.

8.4.3.6 Ablauf

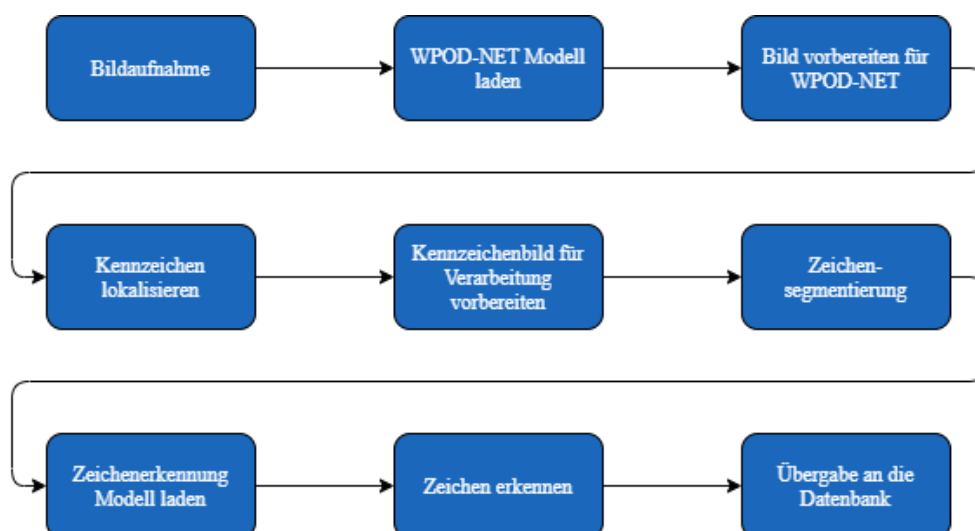


Abbildung 1: Ablaufdiagramm der Kennzeichenerkennung

Im oberen Diagramm ist der Ablauf des Programms angegeben. Zuerst wird mit einem Button der Auslöser betätigt und damit das Foto aufgenommen. Dann wird das erste Machine Learning Modell für die Kennzeichenerkennung „WPOD-NET“ geladen. Bevor das Bild diesem Modell übergeben werden kann, muss es noch angepasst werden damit das Modell damit arbeiten kann. Danach kann damit das Kennzeichen im Bild lokalisiert werden. Im Anschluss wird dieses Kennzeichenbild mit mehreren Bildverarbeitungsalgorithmen verarbeitet, um dann die einzelnen Zeichen zu segmentieren. Danach kann dann das Modell für die Zeichenerkennung geladen werden und dieses dann auch angewendet werden, um das Ergebnis zu erhalten. Dieses Ergebnis wird dann noch mit einer API an die Datenbank übergeben.

8.5 Raspberry Pi

8.5.1 Einleitung

Damit die Software der Kennzeichenerkennung arbeiten kann, benötigt es eine geeignete Hardware. Diese muss dabei die folgenden Eigenschaften aufweisen, um für die Kennzeichenerkennung geeignet zu sein:

- Möglichst klein
- Nicht zu teuer
- Möglichkeit eine Kamera anzuschließen
- Schnell
- Internetanbindung

8.5.2 Wahl des Raspberry Pi

In dieser Applikation wird ein RaspberryPi 4B 2GB verwendet, da er alle zuvor genannten Eigenschaften am besten erfüllt.

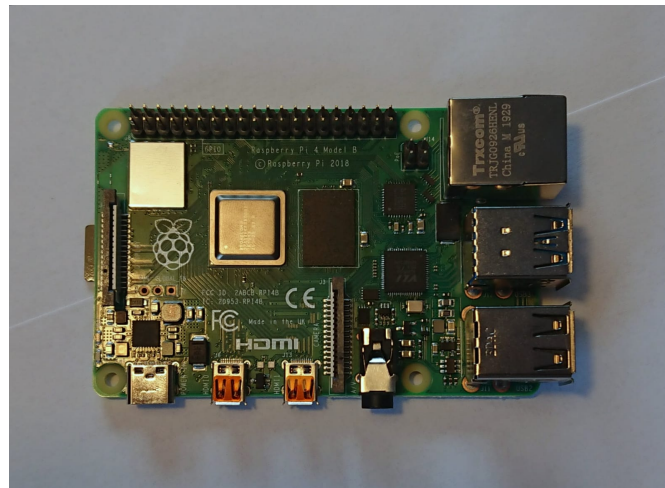


Abbildung 2: Raspberry Pi

Er hat die Größe einer Scheckkarte, wodurch es möglich ist ein kompaktes Gehäuse zu bauen, welches man einfach montieren kann. Er ist mit 35€ pro Stück nicht zu teuer. Er hat einen integrierten Kameraanschluss und es gibt unzählige Kameras, welche damit kompatibel sind. Er hat für seine Größe eine sehr gute Rechenkraft und ist damit in der Lage das komplette Programm für die Kennzeichenerkennung in einer akzeptablen Zeit abzuarbeiten. Er besitzt außerdem einen Ethernet-Anschluss und ein WLAN-Modul, wodurch er sehr einfach mit dem Internet verbunden werden kann.

8.5.3 Kamera

Für den Raspberry Pi muss zudem noch die passende Kamera ausgewählt werden, um die Bilder von den Kennzeichen aufzunehmen. Dafür gibt es Unmengen an kompatiblen Kameras, aber hier wird das Originalzubehör von RaspberryPi verwendet. Dies hat den

Grund, dass diese Kamera leicht zu verwenden, sehr klein, mit Schrauben einfach zu befestigen und günstig ist. Zudem liefert sie ein qualitativ hochwertiges Bild, welches für die Software gut verarbeitbar ist.



Abbildung 3: Raspberry Pi mit Kamera

9 Fahrzeugerkennung

9.1 Anforderungen

Das Ziel der Fahrzeugerkennung ist es Fahrzeuge auf mehrere Parklücken eines Parkplatzes zu erkennen. Die daraus gewonnen Zustände sollen an das Webinterface übermittelt und an den jeweilige Parklücken über LEDs ausgegeben werden.

9.2 Vorstudie

9.3 Erkennung von Metallen über Spulen

Grundsätzlich werden in der Realität häufig Spulen verwendet, welche unter dem Asphalt verbaut sind, um darüberliegende Fahrzeuge zu detektieren. Als Messprinzip wird die Änderung des magnetischen Widerstands R_m bei konstanter magnetischer Spannung U_m und daraus resultierende magnetischen Fluss ϕ Es gelten für diese Größen der folgende Zusammenhänge:

$$\phi = \frac{U_m}{R_m} \quad (1)$$

Wobei

ϕ = magnetischer Fluss

R_m = magnetischer Widerstand

U_m = magnetische Spannung

Der magnetische Widerstand lässt sich wiederum durch die Eigenschaften der Spule bestimmen. Die Formel hierfür lautet:

$$R_m = \frac{N \cdot (2a + 2b)}{\mu_0 \cdot \mu_r \cdot A} \quad (2)$$

Wobei

$$A = a \cdot b \quad (3)$$

N = Anzahl der Windungen der Spule

a = Breite der Spule in m

b = Länge der Spule in m

$\mu_0 = 1,2566 \cdot 10^{-6} \text{ N/A}^2$ = magnetische Feldkonstante

μ_r = relative Permeabilität

A = Fläche der Spule

Bis auf die relative Permeabilität sind alle anderen Variablen konstant. Daraus lässt sich schlussfolgern, dass der magnetische Fluss ϕ anhand der Gleichung 1 und 2 proportional zur relativen Permeabilität ist.

$$\phi \propto \mu_r \quad (4)$$

Der Fluss ϕ hängt somit auch von den Materialien ab durch die er fließt. In der nächsten Abbildung kann man erkennen, wie ein Fahrzeug über eine im Boden installierte Spule den magnetischen Fluss ϕ und somit die magnetische Flussdichte B beeinflussen kann.

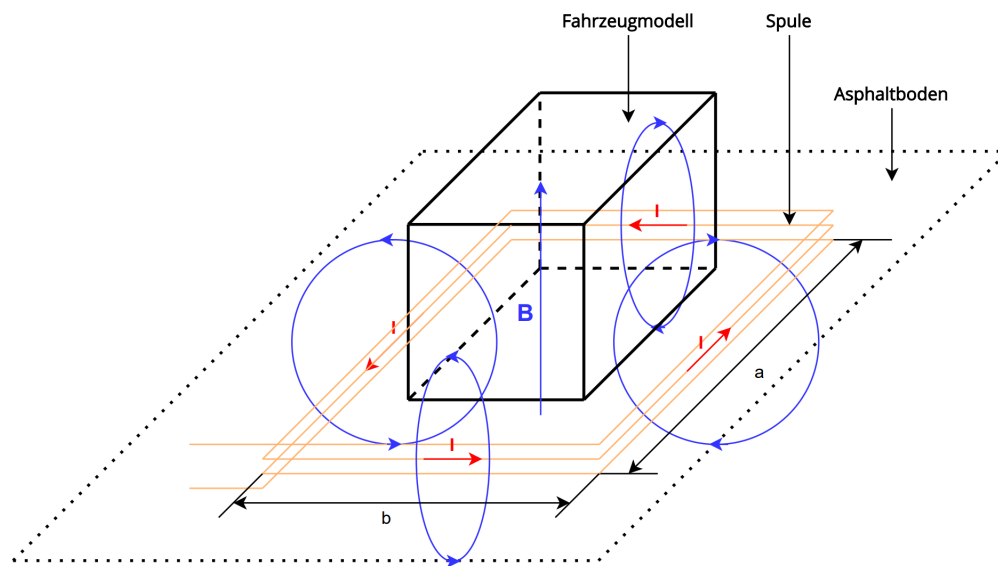


Abbildung 4: Installation der Spule

Um diese Größen auslesbar zu machen, müssen diese magnetischen bei einer direkten Messung in elektrische Größen umgewandelt werden. Hierfür gelten folgende Zusammenhänge:

$$L = \frac{N \cdot \phi}{I} = \frac{\psi}{I} \quad (5)$$

Wobei

L = Induktivität der Spule

I = Strom der durch die Spule fließt

N = Anzahl der Windungen der Spule

ψ = Verkettete Fluss

$$u(t) = L * \frac{di(t)}{dt} \quad (6)$$

Wobei

L = Induktivität der Spule

$i(t)$ = Strom der durch die Spule fließt zum Zeitpunkt t

$u(t)$ = Spannung die an der Spule anliegt zum Zeitpunkt t

So lässt sich bei Bekanntheit von Strom und Spannung auf die Induktivität und mit der Gleichung 5 und mit den Zusammenhang 4 auf die magnetischen Eigenschaften des Materials rückschließen. Diese Art der Detektion bietet viele praktische Vorteile.

- **Größerer Messbereich**

Im Vergleich zu anderen Detektionsmethoden wie einer Lichtschranke kann man einen größeren Bereich durch die Wirkfläche der Spule abdecken. So lassen sich auch kleinere Kraftfahrzeuge wie Motorräder oder Mopeds besser erkennen.

- **Schutz vor Umweltfaktoren**

Durch den Verbau im Boden ist die Messeinrichtung vor Umwelteinflüssen wie Regen, Frost, hohen beziehungsweise niedrigen Temperaturen und Korrosion besser geschützt. Dies verringert auch den Einfluss dieser Störfaktoren auf die Eigenschaften Spule und somit auf die daraus resultierenden Messergebnisse.

- **Ausschließung von Materialien**

Alle nicht metallische Stoffe werden von diesem Messprinzip nicht wahrgenommen. So können Verschmutzungen wie Blätter und Staub, welche visuelle Sensoren stören können, die Detektion nicht behindern.

9.3.1 Messung der ferromagnetischer Metalle

Bei ferromagnetischen Metalle ist die Permeabilitätszahl μ_r sehr viel größer 1. Dies bedeutet, dass eine

9.3.1.1 RL-Oszillator mit Timer Baustein

9.3.1.2 Puls-Ladung mit Mikrokontroller

9.3.2 Messung paramagnetischer Metalle

9.3.2.1 Messung eines Wechselstromes

9.3.2.2 Messung der Resonanzfrequenz eines Oszillators

9.4 RS485 Bussystem

9.4.1 Überblick

9.4.2 Elektrische Spezifikation

9.4.3 Implementation eines eigenen Protokolls

9.5 Mikrokontroller Slave-Geräte

9.5.1 Überblick

9.5.2 Atmega328PB

9.5.3 Peripherie des Mikrokontrollers

9.5.3.1 Spannungswandler

9.5.3.2 RS485 Pegelwandler

9.5.3.3 Digitale Ein- und Ausgänge

9.5.4 Layout des Slave-Gerätes

9.5.5 Gehäuse

9.6 USB-Master

9.6.1 USB-Bussadapter Gerät

9.6.1.1 Überblick

9.6.1.2 FT232RL

9.6.1.3 Spannungsversorgung

9.6.1.4 USB-C Anschluss

9.6.1.5 Layout des Master-Geräts

9.6.1.6 Gehäuse

9.6.2 Master Programm

9.6.2.1 Benötigte Software

9.6.2.2 Adressvergabe

9.6.2.3 Frequenzauslesung

9.6.2.4 Auswertung

9.6.2.5 API-Post

9.6.3 RaspberryPi als Mastergerät

9.6.3.1 SSH Remote Zugriff

9.6.3.2 Code Deployment

9.6.3.3 Unittest

10 Webinterface

10.1 Einleitung

10.2 Anforderungen

Das Webinterface hat auf der einen Seite die Aufgabe die Kommunikation mit der Kennzeichenerkennung und der Fahrzeugerkennung sicherzustellen und auf der anderen Seite die Verwaltung und Darstellung der gewonnen Daten.

10.3 Verwendete Technologien

10.3.1 HTML

HTML steht hierbei für Hypertext Markup Language und ist eine Auszeichnungssprache welche vom World Wide Web Consortium (W3C)² entwickelt wird. Hypertext Markup Language (HTML) ist De-Facto-Standard um Inhalte in Browsern darzustellen. HTML ist dabei aber nicht für die visuelle Darstellung verantwortlich sondern nur für die semantische Struktur. Der Sinn dahinter ist, dass der Inhalt und die Vorgaben an die Darstellung möglichst gut getrennt ist. Für die Formatierung kommt die Stylesheet-Sprache Cascading Style Sheets (CSS) zum Einsatz, welche ebenfalls vom World Wide Web Consortium entwickelt wird. Die aktuellste Version der HTML Spezifikation ist HTML5³ und wurde am 28. Oktober 2014 vom W3C vorgelegt.

²<https://www.w3.org>

³<https://www.w3.org/2014/10/html5-rec.html.en>

10.3.1.1 Beispielhafte HTML Seite

Eine HTML Seite setzt sich aus einer Vielzahl von sogenannten Elementen zusammen. Ein Element besteht aus einem Start Tag und aus einem End Tag, der Inhalt wird zwischen diese Tags geschrieben. Nun folgt eine einfache HTML Seite, welche die Grundlegenden Funktionen von HTML und CSS darlegen soll.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Titel</title>
5  </head>
6  <body>
7
8  <h1>Überschrift</h1>
9  <p>Paragraph</p>
10
11 </body>
12 </html>
```

Code 1: index.html

Das Element `<!DOCTYPE html>` deklariert, dass die folgende Seite den HTML5 Standard verwendet. Danach folgt mit `<html>` das Wurzelement, dass alle anderen Elemente beinhaltet. Das `<head>` Element beinhaltet verschiedene Metadaten d.h. Daten die nicht angezeigt werden. Im oben gezeigten Beispiel Code 2 wird nur der Title des Dokuments gesetzt, dieser wird im Browser Tab angezeigt. Es können aber auch noch andere Daten gesetzt bzw. eingebunden werden:

- Character Set

- Styles
- Scripts
- Viewport
- Sonstige Metainformationen (Author, Keywords)

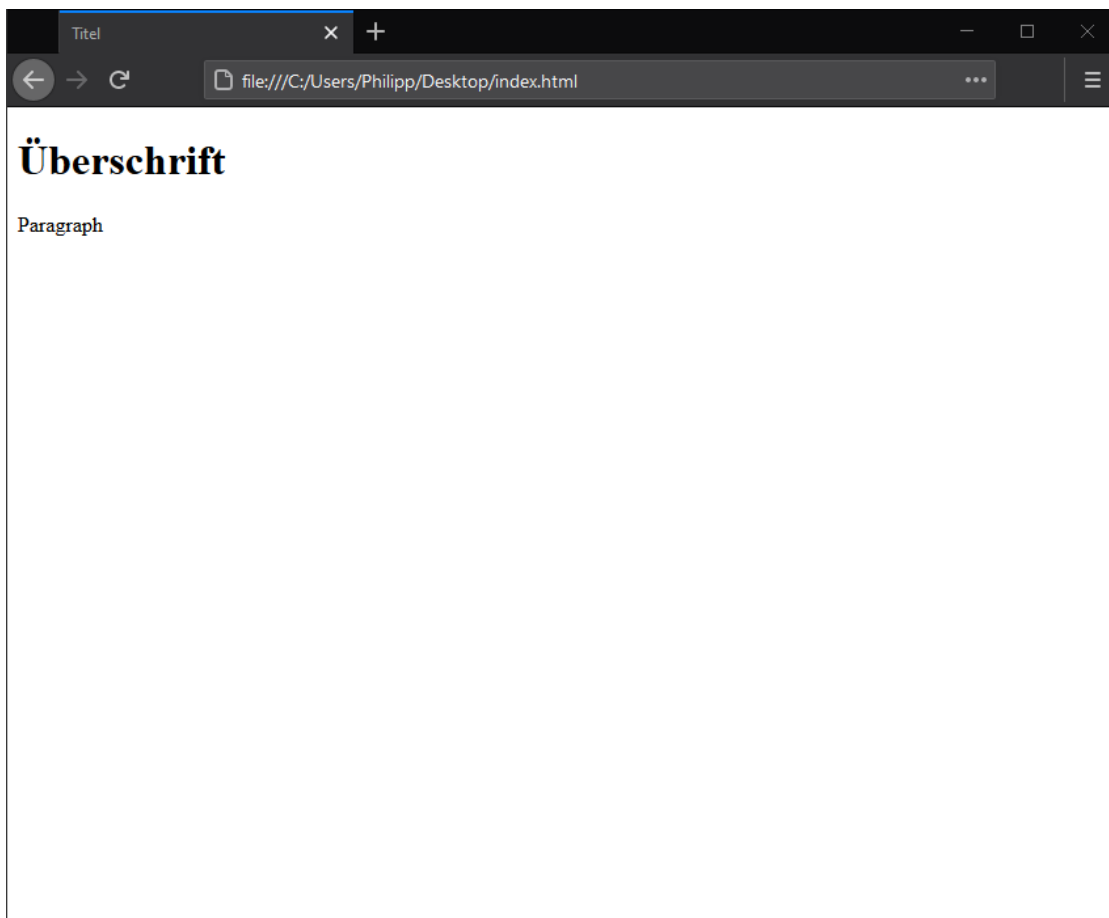


Abbildung 5: Einfache HTML Seite

10.3.2 CSS

Wie bereits angesprochen ist Cascading Style Sheets (CSS) für die Formatierung bzw. die visuelle Darstellung der einzelnen HTML-Elemente verantwortlich. Der Standard

wird wie bei HTML vom W3C spezifiziert und die aktuellste Version ist CSS3 was so viel bedeutet wie CSS Level 3, wobei nur einzelne Teile als Empfehlung durch das W3C vorgelegt wurden, beispielweise das CSS Color Module Level 3⁴. Um die Funktion darzustellen wird die vorherige HTML Seite nun mit CSS ergänzt.

```
1  body {  
2      background-color: deepskyblue;  
3  }  
4  
5  h1 {  
6      color: white;  
7      text-align: center;  
8      font-family: verdana;  
9  }  
10  
11 p {  
12     color: wheat;  
13     font-family: verdana;  
14     font-size: 20px;  
15 }
```

Code 2: style.css

Nun muss dieses Stylesheet nur noch im <head> Tag mit

<link rel="stylesheet" href="mystyle.css"> eingebunden werden.

⁴<https://www.w3.org/TR/css-color-3>

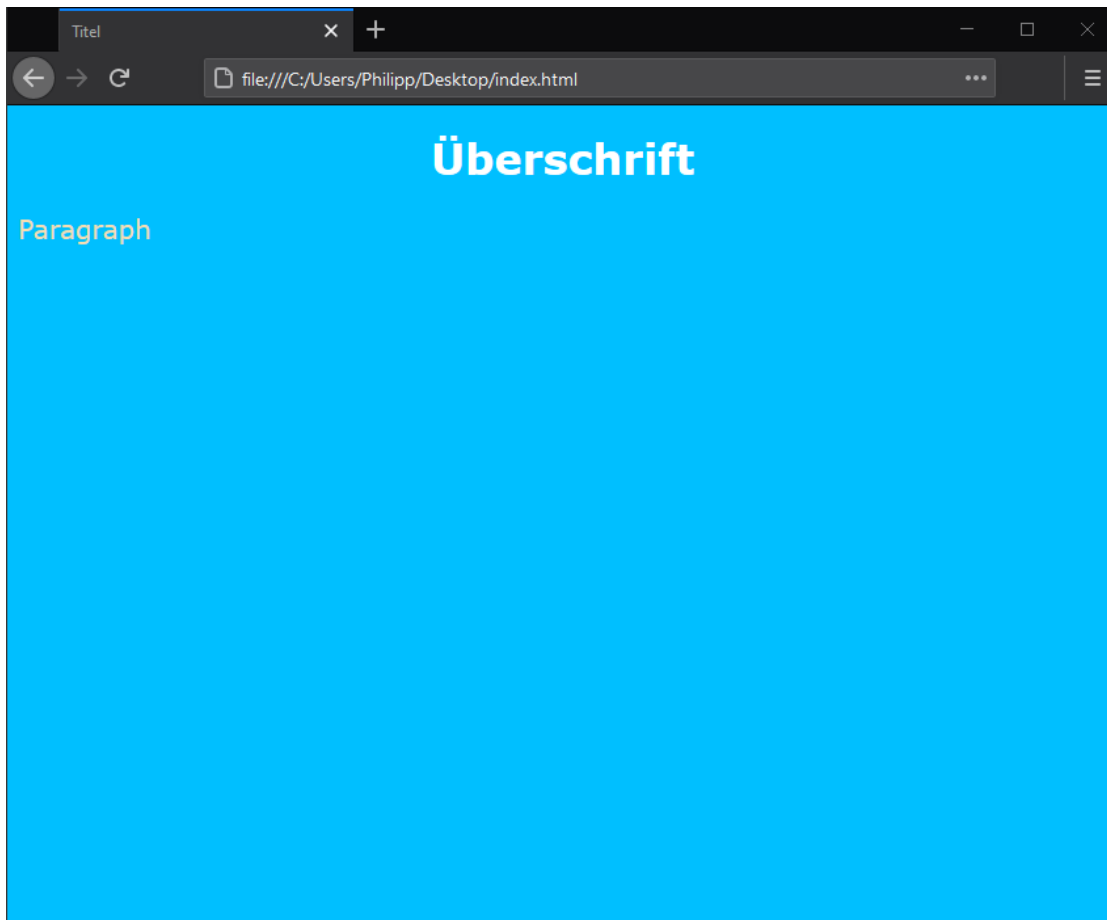


Abbildung 6: Einfache HTML Seite mit CSS

Es lässt sich nun deutlich erkennen, dass sich die Webseite stark verändert hat von den Farben bis zu der Schriftart. HTML und CSS bieten noch viel mehr Funktionen, eine Vielzahl der Funktionen sind auf der Website [w3schools](https://www.w3schools.com)⁵ zu finden.

10.3.3 JavaScript

Es folgt nun eine weitere sehr wichtige Technologie und die meist verwendete Programmiersprache überhaupt laut der Stack Overflow Developer Survey 2020⁶. JavaScript

⁵<https://www.w3schools.com/html> und <https://www.w3schools.com/css>

⁶<https://insights.stackoverflow.com/survey/2020>

(JS) ermöglicht es dynamische Webseiten zu erstellen, dabei wird der Code direkt lokal im Browser ausgeführt. Jedoch ist JavaScript nicht mehr nur auf das Frontend⁷ mehr beschränkt, es möglich mit Frameworks wie Node.js auch Backend⁸ Applikationen zu schreiben und somit ist es möglich Full-Stack⁹-Anwendungen vollständig mit JavaScript zu entwickeln. Eine der wichtigsten Anwendungsgebiete ist die Manipulation von Elementen über das Document Object Model (DOM). Der Standards wird unter dem Namen ECMA Script von der Organisation Ecma International¹⁰ veröffentlicht und die aktuellste Version ist **ECMA-262**¹¹.

10.3.4 PHP

Hypertext Preprocessor (PHP) ist eine Skriptsprache um dynamische Webseiten zu realisieren, jedoch wird nicht wie bei JavaScript der Code auf dem Client ausgeführt sondern auf dem Server, dort wird die HTML-Ausgabe generiert und dem Client zugesendet. Somit ist es nicht möglich den Code als Benutzer zu betrachten. Grundsätzlich ist PHP als synchrone Sprache geplant worden, es ist jedoch auch möglich asynchron zu Programmieren um die Performance zu steigern. Die aktuellste Version ist PHP 8¹².

⁷Grafische Benutzeroberfläche mit der der Benutzer interagiert

⁸Verarbeitung von Daten auf beispielsweise einem Server

⁹Front- und Backend

¹⁰<https://www.ecma-international.org>

¹¹<https://www.ecma-international.org/publications-and-standards/standards/ecma-262>

¹²<https://www.php.net/docs.php>

10.3.5 TailwindCSS

10.3.6 Vue

10.3.7 Laravel

10.3.7.1 Routing

10.3.7.2 Blade Templates

10.3.7.3 Sessions

10.3.7.4 Controllers

10.3.7.5 Artisan

10.3.7.6 Migrations

10.3.7.7 Middleware

10.3.7.8 Eloquent ORM

10.3.7.9 Laravel Mix

10.3.7.10 Laravel Sanctum

10.4 Lokale Entwicklungsumgebung mit Laragon

Für die Programmierung des Webinterfaces müssen zuerst einige Vorkehrungen getroffen werden, dazu zählt zu einem die Installation von benötigter Software und deren konfiguration.

10.4.1 Benötigte Software

- **Laragon** (<https://laragon.org>)

Beinhaltet mehrere Softwarepakete die für die Entwicklung notwendig sind.

- Apache HTTP Server
- MySQL
- PHP

- **phpMyAdmin** (<https://www.phpmyadmin.net>)

Webinterface für MySQL

- **Composer** (<https://getcomposer.org>)

Paketmanager für PHP

- **Git** (<https://git-scm.com>)

Versionskontrolle

- **Visual Studio Code** (<https://code.visualstudio.com>)

Quelltext-Editor

10.4.2 Konfiguration von PHP

Um PHP Befehle von der Kommandozeile auszuführen muss die Installation zuerst in den Windows Path Variables hinzugefügt werden.

Dies erfolgt durch die Advanced System Settings ► Environment Variables ► System Variables. Dort kann nun die Path Variable editiert werden und der Pfad hinzugefügt werden in welchem die `php.exe` liegt.

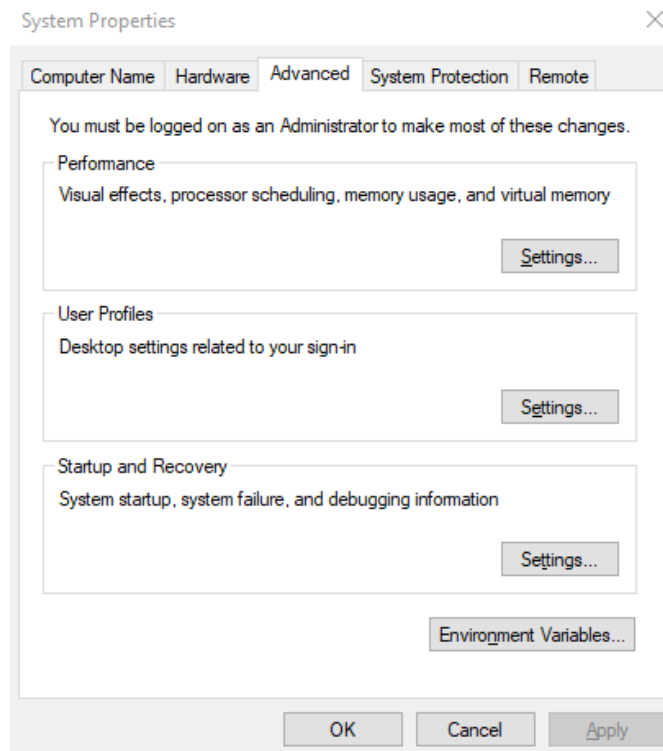


Abbildung 7: Advanced System Settings

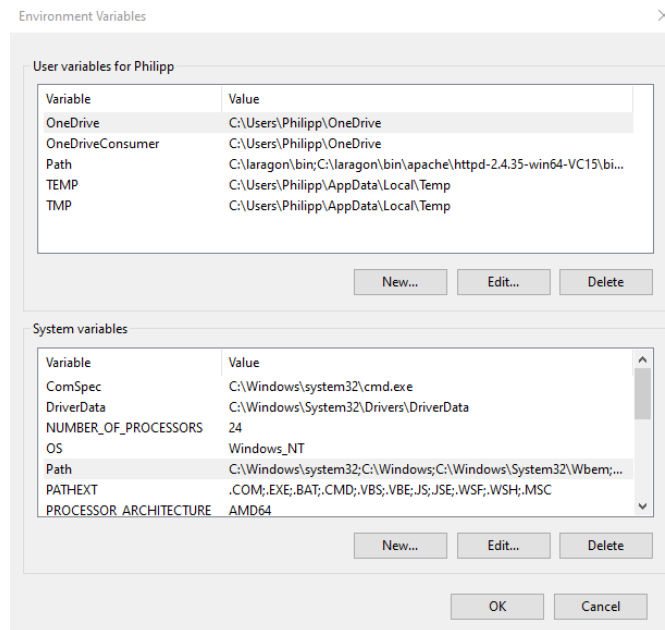


Abbildung 8: System Variables

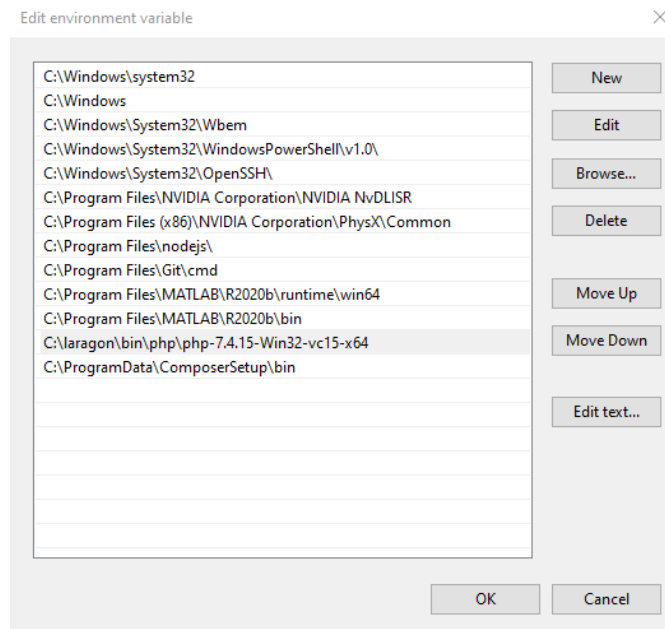


Abbildung 9: Environment Variables

Die korrekte konfiguration kann durch die Kommandozeile geprüft werden, dort muss das Befehl `php -v` ausgeführt werden. Dabei ist zu beachten, dass nach dem hinzufügen der Path Variable die gewählte Kommandozeile neu gestartet werden muss.

```
C:\Users\Philipp>php -v
PHP 7.4.15 (cli) (built: Feb  2 2021 20:47:45) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
```

Abbildung 10: PHP Version

Somit ist PHP korrekt konfiguriert.

10.4.3 Installation von phpMyAdmin

phpMyAdmin ist ein Tool, welches den Umgang mit MySQL Datenbanken mit einem Webinterface erleichtert. Die aktuellste Version lässt sich von <https://www.phpmyadmin.net/downloads> downloaden. Dieses Archiv muss entpackt werden und ausgehend vom Laragon Root Verzeichnis in das Verzeichnis `/etc/apps` kopiert werden. Um die Installation zu überprüfen muss der Apache HTTP Server und der MySQL Server gestartet werden, nun sollte bei einer korrekten Installation das Webinterface von phpMyAdmin unter `http://localhost/phpmyadmin` erreichbar sein.

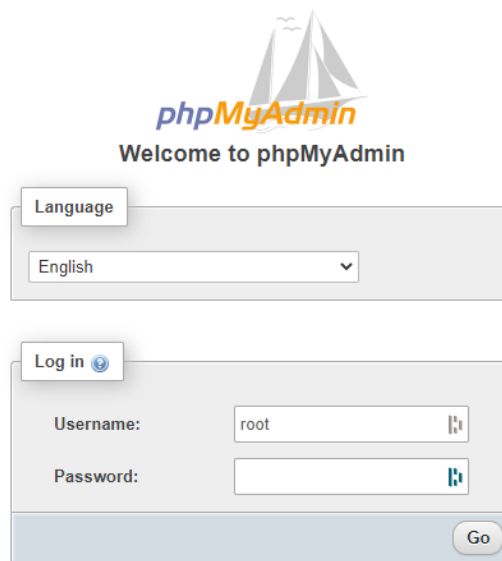


Abbildung 11: phpMyAdmin Webinterface

Es ist nicht notwendig ein Passwort einzugeben, da Standardmäßig kein Passwort gesetzt wird.

10.5 Lokale Entwicklungsumgebung mit WSL und Docker

10.5.1 Benötigte Software

- **Docker** (<https://www.docker.com>)
Ermöglicht Isolation von Anwendungen mit Containervirtualisierung
- **WSL** (<https://docs.microsoft.com/en-us/windows/wsl>)
Kompatibilitätsschicht für Linux Anwendungen unter Windows 10
- **Visual Studio Code** (<https://code.visualstudio.com>)
Quelltext-Editor

10.5.2 Installation von WSL

Zuerst müssen einige Einstellungen in Windows getroffen werden um später eine Linux Distribution herunterzuladen können. Diese Befehle können über die Kommandozeile mit Administrativen Rechten ausgeführt werden.

```
1  dism.exe /online /enable-feature  
    ↪ /featurename:Microsoft-Windows-Subsystem-Linux /all  
    ↪ /norestart
```

Code 3: WSL Feature Feature aktivierens

10.5.2.1 2. Schritt: Virtual Machine Aktivieren

```
1  dism.exe /online /enable-feature  
    ↪ /featurename:VirtualMachinePlatform /all /norestart
```

Code 4: Virtual Machine Feature aktivieren

Nach diesem Schritt ist ein Neustart des Computers notwendig.

10.5.2.2 3. Schritt: Linux Kernel Update

Nun muss ein Linux Kernel Update installiert werden, die aktuelle Version ist unter <https://aka.ms/wsl2kernel> zu finden.

10.5.2.3 4. Schritt: WSL 2

Nach dem Neustart des Computers sollte es nun möglich sein WSL 2 als Version auszuwählen.

```
1 wsl --set-default-version 2
```

Code 5: WSL 2 auswählen

10.5.2.4 5. Schritt: Linux Distribution herunterladen

Zuletzt kann eine Linux Distribution aus dem Windows Store heruntergeladen werden, in diesem Fall Debian (<https://www.microsoft.com/de-de/p/debian>).

10.5.3 Installation von Docker

Die aktuellste Version von Docker Desktop für Windows lässt sich am einfachsten über die offizielle Website von Docker herunterladen (<https://docker.com>). Nach der Installation muss noch die WSL Integration aktiviert werden. Dazu muss in den Einstellungen unter Resources ► WSL Integration und dort muss der Haken bei Enable integration with my default WSL distro gesetzt werden und die installierte Linux Distribution muss unten aktiviert werden.

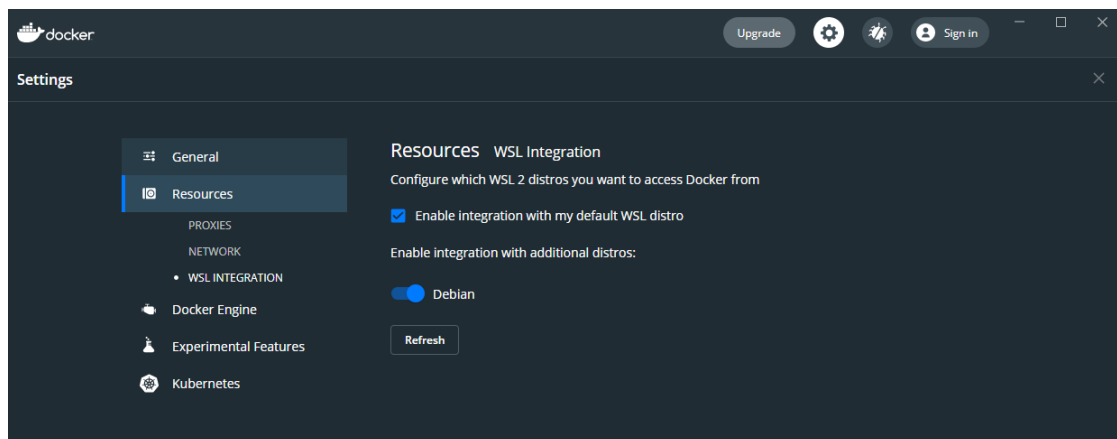


Abbildung 12: Docker WSL Integration

Somit ist die Lokale Entwicklungsumgebung mit WSL und Docker abgeschlossen,

die benötigte Software wird später automatisch durch Laravel Sail in einem Docker Container installiert.

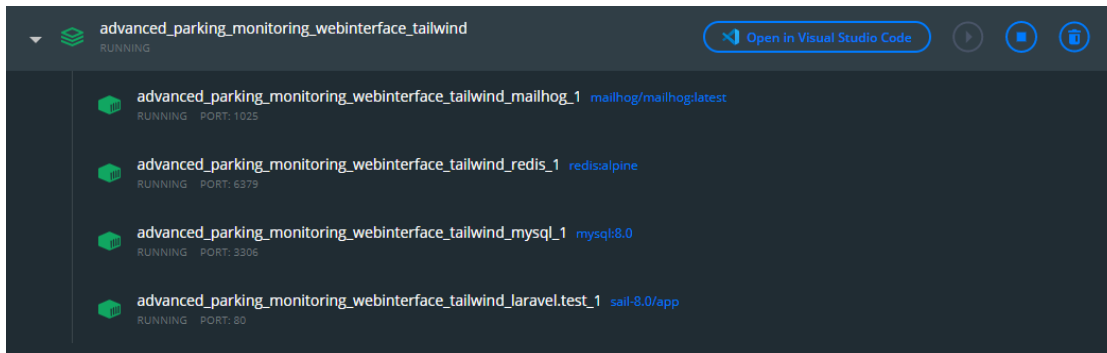


Abbildung 13: Docker Container Steuerung

Es ist somit möglich die Services welche im Container in der Linux Distribution laufen über die Docker Desktop Anwendung zu steuern.

10.6 Production Server

Der Production Server bzw. der Live Server ist der Server wo sich die Webanwendung befindet und die Endbenutzer zugreifen, dieser Server wird auch einfach mit Production abgekürzt. Der Production Server ist in diesem Fall ein Virtual Private Server mit dem Betriebssystem Debian 10, welcher bei einem Internet-Hosting Unternehmen mit Sitz in Deutschland gehostet wird.

10.6.1 Benötigte Software

Für den Live Server wird der sogenannte „LAMP“ Stack verwendet. LAMP steht dabei für die Software **L**inux, **A**pache, **M**ySQL und **P**HP.

- **Apache Web Server** (<https://httpd.apache.org>)
HTTP Server

- **MariaDB** (<https://mariadb.org>)
Fork von MySQL
- **PHP** (<https://mariadb.org>)
Fork von MySQL
- **phpMyAdmin** (<https://www.phpmyadmin.net>)
Webinterface für MySQL
- **Composer** (<https://getcomposer.org>)
Paketmanager für PHP
- **Git** (<https://git-scm.com>)
Versionskontrolle Quelltext-Editor

10.6.2 Installation des LAMP Stacks

Bevor die Software Pakete installiert werden sollte die Linux Software/Update Repository geupdatet werden.

```
1 apt-get update && apt-get upgrade
```

Code 6: Repositories updaten

10.6.2.1 Apache

Nun kann der Apache Web Server installiert werden.

```
1 apt install apache2
```

Code 7: Apache installieren

Die Installation kann nun leicht überprüft werden indem man im Browser die IP bzw. die dazugehörige Domain öffnet, in diesem Fall: (<http://dev.philipp-kraft.com>).

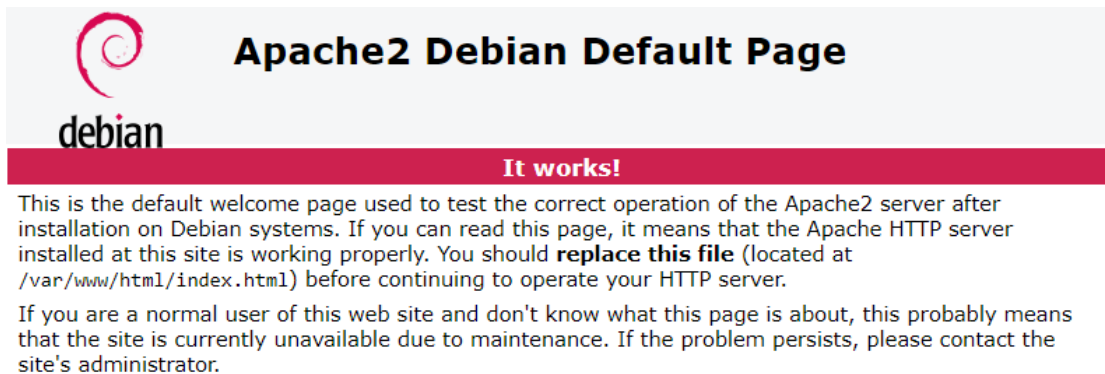


Abbildung 14: Debian Default Page

Erscheint die Debian Default Page ist Apache korrekt installiert.

10.6.2.2 PHP

Neben PHP werden auch einige PHP Extensions benötigt.

```
1 apt install wget php php-cgi php-mysqli php-pear php-mbstring  
  ↪ php-gettext libapache2-mod-php php-common php-phpseclib  
  ↪ php-mysql
```

Code 8: PHP installieren

Die Installation kann einfach mit dem Befehl `php -v` überprüft werden.

10.6.2.3 MariaDB

```
1 apt install mariadb-server
```

Code 9: MariadB installieren

Nun muss MariaDB noch konfiguriert werden.

```
1 mysql_secure_installation
```

Code 10: MariaDB Secure Installation

Dabei wird dem root MySQL User ein Passwort gesetzt, Anonyme Benutzer gelöscht und es werden Test Datenbanken gelöscht.

Nun wird ein neuer Benutzer mit root Berechtigungen erstellt.

```
1 mysql
2 GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED
3 BY 'password' WITH GRANT OPTION;
4 flush privileges;
5 exit
```

Code 11: MariaDB konfiguration

10.6.2.4 phpMyAdmin

Die aktuelle Version von phpMyAdmin kann von (<https://www.phpmyadmin.net/downloads>) bezogen werden und mit dem wget Befehl heruntergeladen werden.

```
1  wget https://files.phpmyadmin.net/phpMyAdmin/5.0.4
2  /phpMyAdmin-5.0.4-all-languages.tar.gz
```

Code 12: phpMyAdmin Download

Anschließend muss das Archiv entpackt werden.

```
1  tar xvf phpMyAdmin-5.0.4-all-languages.tar.gz
```

Code 13: phpMyAdmin Entpacken

Als nächstes muss das entpackte Archiv in einen anderen Pfad verschoben werden und zusätzlich müssen einige Rechte und Verzeichnisse angepasst werden.

```
1  mv phpMyAdmin-5.0.4-all-languages /usr/share/phpmyadmin
2  mkdir -p /var/lib/phpmyadmin/tmp
3  chown -R www-data:www-data /var/lib/phpmyadmin
4  mkdir /etc/phpmyadmin/
```

Code 14: phpMyAdmin Rechte und Verzeichnisse

Nun muss eine Konfigurations Datei erstellt werden und dort muss ein Blowfish Secret¹³ angegeben werden und den Pfad für ein Temporäres Verzeichnis.

¹³32 Zeichen String für Cookie-Authentifizierung

```
1 cp /usr/share/phpmyadmin/config.sample.inc.php  
  ↪ /usr/share/phpmyadmin/config.inc.php
```

Code 15: phpMyAdmin Konfigurationsdatei erstellen

und am Ende dieser Datei müssen folgende zwei Zeilen eingefügt werden.

```
1 $cfg['blowfish_secret'] = 'H20xcGXflSd8JwrwVlh6KW6s2rER63i';  
2 $cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';
```

Code 16: phpMyAdmin Blowfish Secret und TempDir

Zuletzt muss der Apache Web Server konfiguriert werden.

Im Verzeichnis `/etc/apache2/sites-available` muss eine neue Konfiguration angelegt werden `phpmyadmin.conf`.

```
1 Listen 9000
2
3 <VirtualHost *:9000>
4     ServerName localhost
5
6     <Directory /usr/share/phpmyadmin>
7         AllowOverride None
8         Require all granted
9     </Directory>
10
11     DocumentRoot /usr/share/phpmyadmin
12
13     ErrorLog ${APACHE_LOG_DIR}/phpmyadmin.error.log
14     CustomLog ${APACHE_LOG_DIR}/phpmyadmin.access.log
15         ↪ combined
16 </VirtualHost>
```

Code 17: phpmyadmin.conf

Nun kann diese Virtual Host Konfigurations Datei aktiviert werden und danach muss der Apache Web Server neu gestartet werden.

```
1 a2ensite phpmyadmin
2 systemctl restart apache2
```

Code 18: Virtual Host aktivieren

Diese Konfiguration ermöglicht es, dass das Webinterface von phpMyAdmin über

den Port 9000 (<http://dev.philipp-kraft.com:9000>) erreichbar ist und nicht wie Standardmäßig vorgesehen über das Verzeichnis /phpmyadmin (<http://dev.philipp-kraft.com/phpmyadmin>), dies bietet einen Sicherheitsvorteil.

10.6.2.5 Webinterface Virtual Host

Nun muss noch eine Virtual Host Konfiguration für das Webinterface selbst erstellt werden.


```
1 <VirtualHost *:80>
2   ServerAdmin webmaster@localhost
3   DocumentRoot /var/www/apm/public
4
5   <Directory />
6     Options FollowSymLinks
7     AllowOverride All
8   </Directory>
9
10  <Directory /var/www/apm>
11    Options Indexes FollowSymLinks MultiViews
12    AllowOverride All
13    Order allow,deny
14    allow from all
15  </Directory>
16
17  ErrorLog ${APACHE_LOG_DIR}/error.log
18  CustomLog ${APACHE_LOG_DIR}/access.log combined
19 </VirtualHost>
```

Code 19: apm.conf

Diesmal wird auf den Standard HTTP Port 80 gehört und dieser führt in das Verzeichnis /var/www/apm/public. Zusätzlich werden noch Directives gesetzt, damit das Standard .htaccess File von Laravel richtig funktionieren kann.

10.6.2.6 Installation von Composer

Die Installation von Composer gestaltet sich relativ einfach.

```
1 wget -O composer-setup.php https://getcomposer.org/installer
```

Code 20: Download Composer Installer

Nun muss das Setup ausgeführt werden und damit das `composer` Befehl Global verfügbar ist wird Composer in den Pfad `/usr/local/bin` verschoben.

```
1 php composer-setup.php --install-dir=/usr/local/bin  
  ↪ --filename=composer
```

Code 21: Composer Setup

10.6.3 Deployment mit Github Actions

Unter Deployment versteht man die automatische Installation von Software, in diesem Fall auf einem Linux Server. Erreicht wird das durch zwei Bash Scripts und mit Github Actions (<https://github.com/features/actions>).

10.6.3.1 Git Setup

Sollte auf dem Server noch kein Git installiert sein, lässt sich das wie folgt installieren.

```
1 apt install git
```

Code 22: Git Installation

Im Verzeichnis `/var/www/apm` soll sich später das Webinterface befinden, deshalb muss in diesem Pfad Git konfiguriert werden. Dazu wird die Remote URL konfiguriert.

```
1 git config remote.origin.url 'https://{TOKEN}@github.com/  
2 Philipp-Kraft/Advanced_Parking_Monitoring_Webinterface.git'
```

Code 23: Git Remote Origin

Da beim Github Account eine Zwei-Faktor-Authentisierung verwendet wird muss die Authentifizierung mit einem Personal access token erfolgen. Dieser kann unter <https://github.com/settings/tokens> erstellt werden, der erstellte Token kann dann einfach in der URL eingefügt werden.

10.6.3.2 Deploy Script

Das Deploy-Script wird auf der Lokalen Entwicklermaschine ausgeführt. Das Script wechselt in den Production Branch und merged mit dem Main Branch, dieser Push in den Production Branch löst dann die Github Action aus.

```
1  #!/bin/sh
2  set -e
3
4  #vendor/bin/phpunit
5
6  (git push) || true
7
8  git checkout production
9  git merge main
10
11 git push origin production
12
13 git checkout main
```

Code 24: phpmyadmin.conf

10.6.3.3 Server Deploy Script

Das Server Deploy Script `server_deploy.sh` versetzt die Laravel Applikation in den Wartungsmodus und lädt sich vom deploy Branch den Code auf den Server herunter, danach werden einige Befehle ausgeführt.

```
1  #!/bin/sh
2  set -e
3
4  echo "Deploying application ..."
5
```

```
6  # Enter maintenance mode
7  php artisan down
8
9  # Update codebase
10 git fetch origin deploy
11 git reset --hard origin/deploy
12
13 # Install dependencies based on lock file
14 composer install --no-interaction --prefer-dist
    ↳ --optimize-autoloader
15
16 # Migrate database
17 php artisan migrate:refresh --seed
18
19 # Clear cache
20 php artisan optimize
21
22 # Exit maintenance mode
23 php artisan up
24
25 echo "Application deployed!"
```

Code 25: serverdeploy.sh

10.6.3.4 Github Action

Github Actions ist ein Projekt von Github, welches es ermöglicht Automatisierungen in den Bereichen Entwicklung, Testing und Deployment durchzuführen.

Als erstes muss ein Workflow erstellt werden, dieser wird im Root-Verzeichnis des Projekts erstellt APM\.github\workflows\main.yml.

```
1  name: Deploy Laravel app
2
3  on:
4    push:
5      branches: [ production ]
6
7  jobs:
8    deploy:
9      runs-on: ubuntu-latest
10     steps:
11       - uses: actions/checkout@v2
12         with:
13           token: ${ secrets.PUSH_TOKEN }
14       - name: Set up Node
15         uses: actions/setup-node@v1
16         with:
17           node-version: '12.x'
18       - run: npm install
19       - run: npm run production
20       - name: Commit built assets
21         run: |
22           git config --local user.email "action@github.com"
23           git config --local user.name "GitHub Action"
24           git checkout -B deploy
```

```
25     git add -f public/  
26     git commit -m "Build front-end assets"  
27     git push -f origin deploy  
28 -   name: Deploy to production  
29     uses: appleboy/ssh-action@master  
30     with:  
31         username: root  
32         host: dev.philipp-kraft.com  
33         password: ${ secrets.SSH_PASSWORD }  
34         script: 'cd /var/apm && ./server_deploy.sh && chown -R  
           ↪ www-data:www-data /var/apm && chmod -R 755 /var/apm  
           ↪ && chmod -R 777 /var/apm/storage'
```

Code 26: main.yml

Dieser Workflow setzt einen Ubuntu Server in der Cloud auf und baut dort die Assets zusammen, damit die Downtime auf dem Production Server möglichst gering ist. Danach wird eine SSH Verbindung zum Production Server aufgebaut und dort wird das Bash-Skript `server_deploy.sh` ausgeführt. Gleichzeitig werden einige Berechtigungen angepasst.

In der Repository muss nun noch das SSH-Passwort und der Personal access token hinterlegt werden. Dies geschieht über Settings ► Secrets. Dort können nun über New repository secrets die Secrets hinterlegt werden.



Abbildung 15: Action Secrets

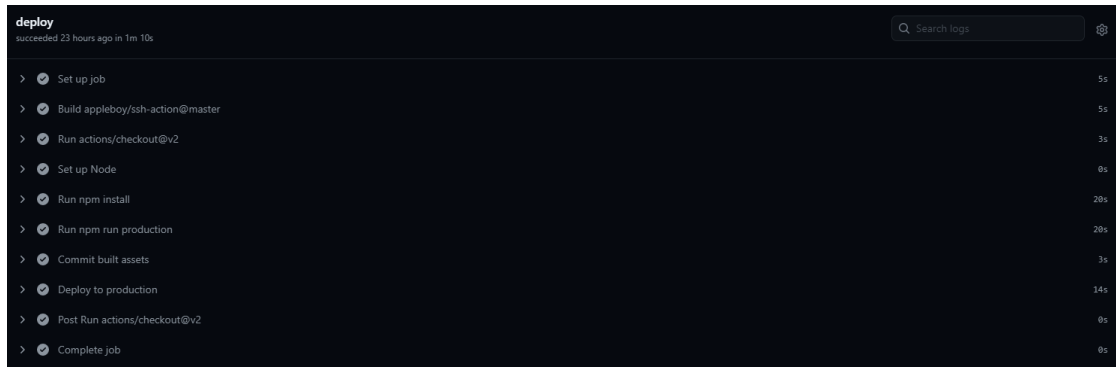


Abbildung 16: Github Action Übersicht

10.7 Grundlegender Aufbau

Da das Ziel ist, dass das Frontend¹⁴ des Webinterfaces möglichst Modular aufgebaut ist und so wenig Code wie möglich wiederholt wird. Ermöglicht wird dies durch das aufbauen der Seite mithilfe von Components.

10.7.1 Components

Components sind praktisch kleine Bausteine aus denen die komplette Seite aufgebaut ist. Components sind kein natives Feature von HTML/CSS oder PHP, diese Funktion wird von der Template Engine Blade bereitgestellt, deshalb werden diese auch oft Blade Components genannt.

10.7.1.1 Anonymous Components

Es gibt viele verschiedene Möglichkeiten Components zu erstellen und verschiedene Konventionen an, einfachsten sind aber die Anonymous Components, diese haben

¹⁴Präsentationsebene in Form der grafischen Benutzeroberfläche

den Vorteil, dass diese in einer Datei verwaltet werden können und somit sehr einfach zu handhaben sind.

10.7.1.2 Components erstellen

Das Erstellen von einem Component wird nun anhand eines Buttons gezeigt. Da dieser als Anonymous Component angelegt wird muss dieser mit keiner Klasse assoziiert werden, es wird einfach im Pfad `resources\views\components` ein Ordner mit dem Namen `buttons` angelegt und darin ein Blade File mit dem Namen `primary.blade.php`. Dort kann nun der gewünschte HTML Code platziert werden.

```
1 <button type="submit" class="inline-flex items-center px-4 py-2  
  ↳ bg-apm-blue...">  
2   {{ $slot }}  
3 </button>
```

Code 27: primary.blade.php

Im Code 27 ist eine Variable mit dem Namen `slot` verwendet worden. Diese Variable wird später beim verwenden automatisch mit dem Inhalt zwischen dem HTML Element ersetzt.

10.7.1.3 Components verwenden

Es stellt sich nun die Frage wie man dieses erstellte Component nun verwendet. Die Blade Components verwenden den gleichen Syntax wie ein normales HTML Element, mit dem Unterschied dass ein `x-` vor dem Namen des Components angeführt werden muss. Da sich der vorhin erstellte Component in einem Ordner befindet muss das

auch angegeben werden, dabei wird kein Slash wie üblich um einen Pfad anzugeben verwendet sondern ein Punkt, es muss auch keine Extensions angegeben werden.

```
1 <x-buttons.primary>Press me!</x-buttons.primary>
```

Code 28: Verwendung eines Button Components

10.7.1.4 Attribute übergeben

Auch wenn viele Components ohne Problem überall ohne Veränderung verwendet werden können, ist es gewünscht bei manchen Components beispielsweise eine zusätzliche Klasse anzugeben um die Größe des Elements zu verändern. Erreicht wird das mit der `attributes` Variable im Blade File des Components, da aber oft schon Attribute definiert sind ist es möglich mit der `merge` Methode die Attribute zusammenzuführen.

```
1 <button {{ $attributes->merge(['type' => 'submit', 'class' =>
   ↳ 'inline-flex items-center px-4 py-2 bg-apm-blue...']) }}>
2   {{ $slot }}
3 </button>
```

Code 29: Modularer Button Component

Somit ist dieser Button Component nun vollständig Modular.

10.7.2 Layouts

Da auf den meisten Seiten des Webinterfaces fast das gleiche Layout beibehaltet ist es sinnvoll diesen Inhalt in ein Component umzuwandeln. Auch Layouts sind

Components.

Es gibt im Webinterface folgende Layouts:

- **admin.blade.php**

Layout für die Administrativen Seiten mit einer Sidebar und Page Header

- **app.blade.php**

Layout für sonstige Seiten ohne Sidebar

- **display.blade.php**

Besonderes Layout für die Display Seiten

Admin Layout

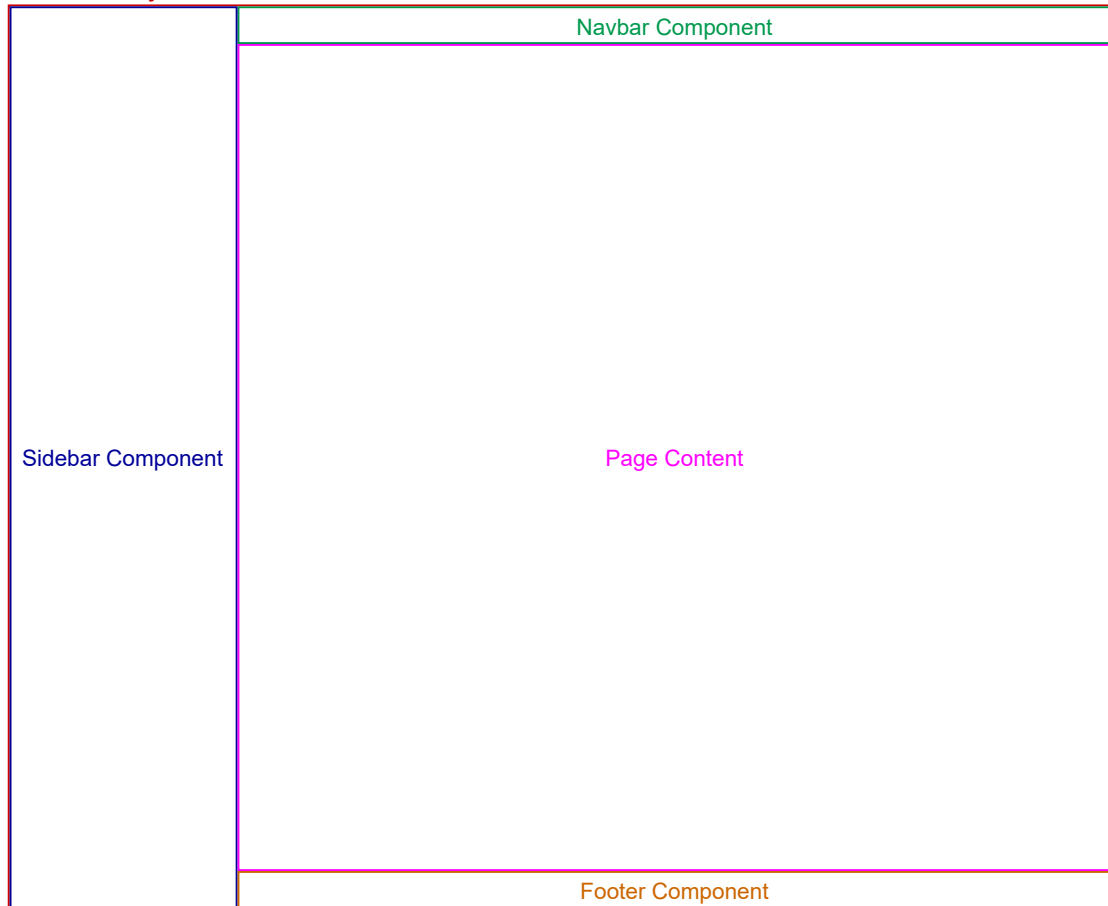


Abbildung 17: Admin Layout

App Layout

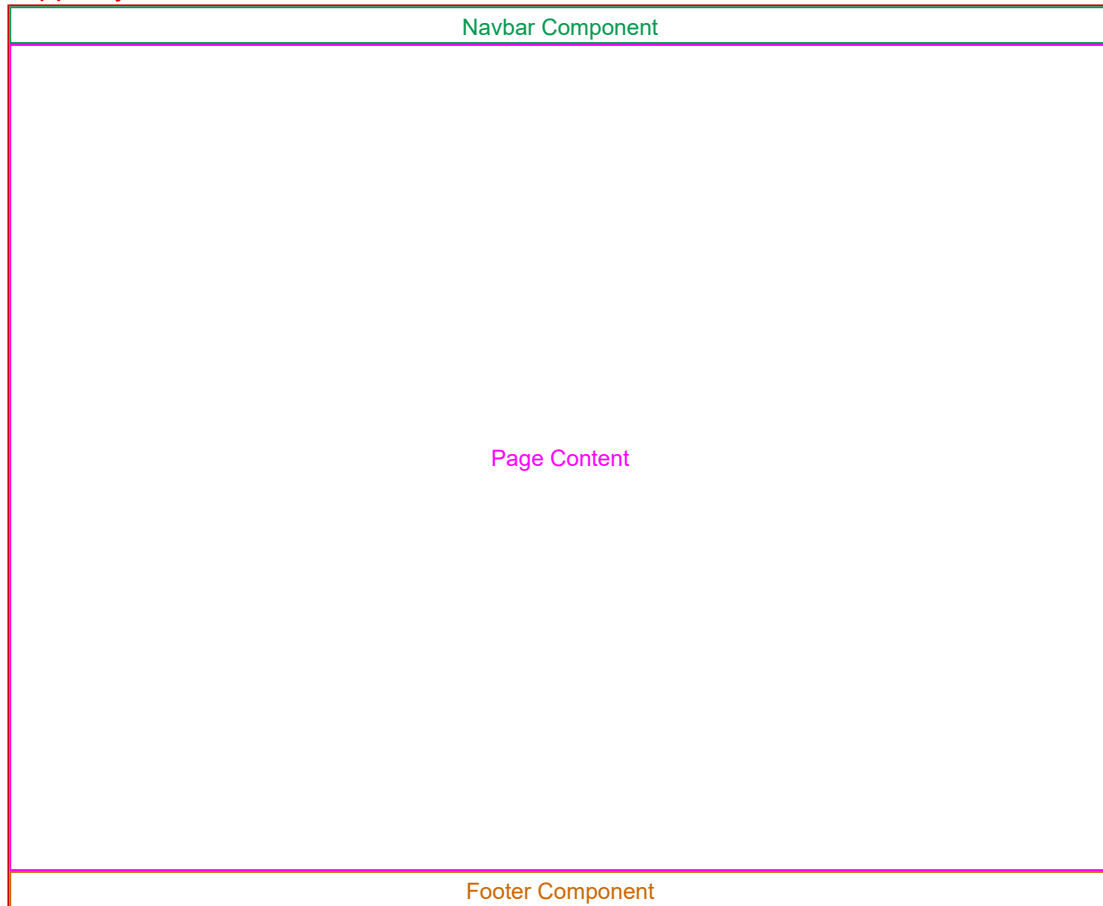


Abbildung 18: App Layout

10.7.3 Navbar

10.7.4 Sidebar

10.7.5 Footer

10.8 Funktionen

10.8.1 Dashboard

10.8.2 Login- und Registersystem

10.8.3 News

10.8.4 Benutzerverwaltung

10.8.5 Rechte- und Rollenverwaltung

10.8.6 Parkplatzverwaltung

10.8.7 Kennzeichenverwaltung

10.8.8 Erkennungsverlauf

10.8.9 Seiten Einstellungen

10.8.10 API Schlüssel

10.8.11 Displays

10.8.12 Notifications

10.8.13 Profil

10.8.14 Lokalisierung

10.9 Performance und Sicherheit

10.9.1 Form Validation

Philipp Kraft

10.9.2 Authentifizierung

10.9.3 Authentisierung

10.9.4 SQL Injection

10.9.5 Cross Site Scripting

11 Zusammenfassung und Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu

enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

12 Anhang

Abbildungsverzeichnis

1	Ablaufdiagramm der Kennzeichenerkennung	26
2	Raspberry Pi	28
3	Raspberry Pi mit Kamera	29
4	Installation der Spule	32
5	Einfache HTML Seite	39
6	Einfache HTML Seite mit CSS	41
7	Advanced System Settings	45
8	System Variables	46
9	Environment Variables	46
10	PHP Version	47
11	phpMyAdmin Webinterface	48
12	Docker WSL Integration	50
13	Docker Container Steuerung	51
14	Debian Default Page	53
15	Action Secrets	65
16	Github Action Übersicht	66
17	Admin Layout	70
18	App Layout	71

Tabellenverzeichnis

Codeverzeichnis

1	index.html	38
2	style.css	40
3	WSL Feature Feature aktivieren	49
4	Virtual Machine Feature aktivieren	49
5	WSL 2 auswählen	50
6	Respositorys updaten	52
7	Apache installieren	52
8	PHP installieren	53
9	Mariadb installieren	54
10	MariaDB Secure Installation	54
11	MariaDB konfiguration	54
12	phpMyAdmin Download	55
13	phpMyAdmin Entpacken	55
14	phpMyAdmin Rechte und Verzeichnisse	55
15	phpMyAdmin Konfigurationsdatei erstellen	56
16	phpMyAdmin Blowfish Secret und TempDir	56
17	phpmyadmin.conf	57
18	Virtual Host aktivieren	57
19	apm.conf	59
20	Download Composer Installer	60
21	Composer Setup	60
22	Git Installation	60
23	Git Remote Origin	61
24	phpmyadmin.conf	62
25	serverdeploy.sh	63
26	main.yml	65

27	primary.blade.php	67
28	Verwendung eines Button Components	68
29	Modularer Button Component	68

Abkürzungsverzeichnis

PK Philipp Kraft

DK Dennis Köb

SB Samuel Bleiner

APM Advanced Parking Monitoring

HTML Hypertext Markup Language

CSS Cascading Style Sheets

PHP Hypertext Preprocessor

JS JavaScript

ORM Object–relational mapping

W3C World Wide Web Consortium

DOM Document Object Model

Literaturverzeichnis

Einstein, Albert. »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«. In: *Annalen der Physik* 322.10 (1905), S. 891–921.
DOI: <http://dx.doi.org/10.1002/andp.19053221004>.